aws

Guida per l'utente

CloudWatch Registri Amazon



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

CloudWatch Registri Amazon: Guida per l’utente

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

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

Table of Contents

Che cos'è Amazon CloudWatch Logs? 1
Funzionalità 1
Servizi correlati AWS 3
Prezzi 4
Concetti 4
Fatturazione e costi 6
Classi di registro 7
Funzionalità supportate 7
Ottimizza i costi di storage con Amazon CloudWatch Logs Intelligent Tiering 11
CloudWatch Livelli di accesso Logs Intelligent Tiering 11
Azioni che costituiscono l'accesso 12
Abilitazione dei CloudWatch log: Intelligent Tiering 13
Autorizzazioni richieste 13
Abilita l'utilizzo della console 13
Abilita usando il AWS CLI 13
Visualizzazione della configurazione attuale 14
Monitoraggio dello storage per livello 14
Prezzi di CloudWatch Logs Intelligent Tiering 15
Nozioni di base 16
Prerequisiti 16
Registrati per un Account AWS 16
Impostazione dell'interfaccia a riga di comando 17
Utilizzo dell'agente unificato CloudWatch 17
Utilizzando l'agente precedente CloudWatch 17
CloudWatch Registra i prerequisiti dell'agente 18
Quick Start: installazione dell'agente su un'istanza Linux EC2 in esecuzione 19
Quick Start: installazione dell'agente su un'istanza Linux EC2 all'avvio 26
Avvio rapido: usa i CloudWatch log con le istanze di Windows Server 2016 30
Avvio rapido: usa CloudWatch i log con le istanze di Windows Server 2012 e Windows Server 2008 41
Segnala lo stato dell' CloudWatch agente Logs 52
Avvia l'agente CloudWatch Logs 52
Arresta l'agente CloudWatch Logs 53
CloudWatch Registra il riferimento dell'agente 53

Avvio rapido con CloudFormation ... 65
Inserimento dei log tramite endpoint HTTP ... 67
Comportamento comune ... 67
Autenticazione con token Bearer ... 68
Opzione 1: avvio rapido a utilizzare la AWS console ... 69
Opzione 2: configurazione manuale ... 71
Controlla le autorizzazioni per la generazione e l'utilizzo delle chiavi API CloudWatch Logs ... 74
Chiavi API rotanti ... 76
Risposta a una chiave API compromessa ... 77
Le migliori pratiche di sicurezza per le chiavi API ... 78
Registrazione dell'utilizzo delle chiavi API con CloudTrail ... 79
Endpoint OTLP ... 80
Formato della richiesta ... 80
Richiesta di esempio ... 80
Risposte ... 81
Comportamenti specifici di OTLP ... 82
Endpoint HLC ... 82
Modalità di input ... 82
Campo evento (obbligatorio) ... 83
Campo orario (opzionale) ... 83
Content-Type ... 84
Tipi di valori JSON accettati ... 84
Formato dell'endpoint ... 84
Formato della richiesta ... 85
Richiesta di esempio ... 86
Best practice ... 86
Limitazioni ... 87
Endpoint NDJSON ... 87
Formato della richiesta ... 88
Tipi di valori JSON accettati ... 88
Campo Timestamp ... 89
Righe non valide ... 90
Richiesta di esempio ... 90
Risposte ... 90
Best practice ... 91

Limitazioni ... 92
Endpoint JSON strutturato ... 92
Formato della richiesta ... 92
Tipi di valori JSON accettati ... 93
Campo Timestamp ... 93
Richiesta di esempio ... 94
Risposte ... 94
Best practice ... 95
Limitazioni ... 95
Confronto degli endpoint di ingestione HTTP ... 96
Scelta di un endpoint ... 97
Ingestione di Syslog ... 98
Protocolli e porte supportati ... 98
Formati syslog supportati ... 99
Campi estratti da RFC 5424 ... 99
Campi estratti in RFC 3164 ... 100
Campi FTD/ASA estratti da Cisco ... 101
Consegna dei messaggi ... 101
Quote e limiti ... 102
Configurazione ... 103
Prerequisiti ... 103
Fase 1: Creare o identificare un VPC ... 104
Fase 2: creazione di un gruppo di sicurezza ... 106
Fase 3: creazione dell'endpoint VPC ... 107
Fase 4: Creare un gruppo di log ... 109
Fase 5: Aggiungere una politica delle risorse ... 109
Fase 6: Creare la configurazione syslog ... 111
Fase 7: Verifica della consegna e dell'estrazione sul campo ... 112
Configurazione del dispositivo ... 113
rsyslog ... 113
syslog-ng ... 115
Validazione rapida ... 116
Apparecchiature di rete ... 117
Attendibilità dei certificati TLS ... 118
Monitoraggio ... 118
Metriche Syslog ... 118

Elimina i motivi 119
Motivi del rifiuto della connessione 120
Allarmi raccomandati 120
Visualizzazione delle metriche syslog sulla dashboard automatica 121
Query di esempio 121
Sicurezza e monitoraggio degli accessi 121
Visibilità operativa 122
Interrogazioni su firewall e dispositivi di rete 123
Policy di endpoint VPC 124
Chiavi di condizione supportate 124
Limita a un gruppo di log specifico 125
Nega tutto il traffico 126
Ripristina la politica predefinita 126
Propagazione delle politiche 126
Risoluzione dei problemi 127
Messaggi che non compaiono nel gruppo di log 127
Errori di handshake TLS 128
Messaggi UDP persi 129
Connessione rifiutata 129
Messaggi interrotti a causa della limitazione della velocità 129
Lavorare con AWS gli SDK 131
Gestione dei registri 133
Individuazione e gestione delle fonti di dati 133
Che cos'è CloudWatch Logs Data Sources? 134
Come iniziare 134
Campi di sistema 136
Accesso alle origini dati 136
Relazione con i gruppi di log 137
Funzionalità abilitate dalle fonti di dati 137
Origini dati supportate 138
Supportata Servizi AWS per le fonti di dati 138
Fonti di terze parti supportate per le fonti di dati 143
Origini personalizzate 161
Analisi dei dati di registro con CloudWatch Log Analytics 162
Cosa puoi fare con Log Analytics 162
Inizia a usare Log Analytics 162

Analisi dei dati di registro con CloudWatch Logs Insights 164
Linguaggi di interrogazione supportati 167
CloudWatch Linguaggio di interrogazione Logs Insights (Logs Insights QL) 167
OpenSearch Piped Processing Language (PPL) 262
OpenSearch Structured Query Language (SQL) 274
Usa il linguaggio naturale per generare e aggiornare CloudWatch le query di Logs Insights 287
Query di esempio 288
Rifiuto esplicito all'utilizzo dei dati volto al miglioramento del servizio 290
Registri supportati e campi rilevati 291
Campi nei registri JSON 293
Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione .. 296
Campi indicizzati automaticamente 299
Sintassi e quote dell'indice di campo 301
Crea una politica di indicizzazione dei campi a livello di account 306
Crea una politica di indicizzazione dei campi a livello di gruppo di log 307
Registra le opzioni di selezione dei gruppi durante la creazione di un'interrogazione 308
Effetti dell'eliminazione di una politica di indicizzazione dei campi 309
Usa le sfaccettature per raggruppare ed esplorare i log 310
Per eseguire una query basata su facet 311
Per salvare un'interrogazione basata su facet 311
Per creare un facet a livello di account 311
Gestione dei facet tramite API 312
Visualizzazione dei log circostanti in Logs Insights CloudWatch 312
Come funzionano i registri circostanti 313
Visualizza i registri circostanti 313
Cerca nei log circostanti 314
Analisi del modello 315
Guida introduttiva all'analisi dei pattern 316
Dettagli sul comando pattern 319
Salvataggio e riesecuzione di query 319
Utilizzo di interrogazioni salvate con parametri 325
Aggiunta di query a pannello di controllo o esportazione dei risultati della query 328
Visualizzazione di query in esecuzione o cronologia delle query 328
Crittografa i risultati delle interrogazioni con AWS Key Management Service 329
Limits 330
Passaggio 1: creare un AWS KMS key 330

Fase 2: Impostazione delle autorizzazioni sulla chiave KMS 331
Fase 3: associazione di una chiave KMS ai risultati della query 332
Fase 4: Dissociazione di una chiave dai risultati della query nell'account 333
Genera un riepilogo in linguaggio naturale dai risultati delle query di CloudWatch Logs Insights 333
Come funziona 333
Disponibilità regionale ed elaborazione dei dati 333
Nozioni di base 334
Permissions 335
Privacy dei dati 335
Automatizzazione dell'analisi dei log con interrogazioni pianificate 336
Comprensione dei concetti relativi alle interrogazioni pianificate 336
Separazione dei ruoli IAM 337
Cross-region e utilizzo su più account 339
Pianifica le espressioni e la gestione dei fusi orari 341
Scelta di un linguaggio di interrogazione 341
Selezione della destinazione e casi d'uso 342
Formato e struttura dei risultati delle query 344
Informazioni su RecordsScanned, RecordsMatched e ResultCount 346
Riferimento alle espressioni di pianificazione 349
Best practice 353
Guida introduttiva alle query pianificate 354
Creazione di un'interrogazione pianificata 355
Visualizzazione e gestione delle interrogazioni pianificate 359
Visualizzazione della cronologia delle esecuzioni pianificate delle query 360
Aggiornamento di una query pianificata 362
Configurazione delle destinazioni S3 per le interrogazioni pianificate 364
Fornire risultati a un bucket Amazon S3 nello stesso account 364
Fornitura dei risultati a un bucket Amazon S3 in un altro account 365
Crittografia dei risultati con un cliente gestito AWS KMS Chiave 367
Configurazione delle destinazioni delle tabelle di ricerca per le interrogazioni pianificate 369
Risoluzione dei problemi relativi alle interrogazioni pianificate 370
L'esecuzione della query non riesce a causa di errori di autorizzazione 370
Scade il timeout delle query 371
L'elaborazione della destinazione non riesce 372
Errori di interrogazione non validi 372

Errori di concorrenza nelle query 373
Rilevamento delle anomalie nei registri 374
Gravità e priorità delle anomalie e dei modelli 375
Tempo di visibilità dell'anomalia 376
Soppressione di un'anomalia 376
Domande frequenti 376
Utilizzo del rilevamento delle anomalie in Logs Insights CloudWatch 377
Abilita il rilevamento delle anomalie su un gruppo di log 378
Visualizza le anomalie rilevate 379
Crea allarmi sui rilevatori di anomalie di registro 382
Metriche pubblicate dai rilevatori di anomalie di registro 385
Crittografa un rilevatore di anomalie e i relativi risultati con AWS KMS 385
Limits 386
Risolvi i problemi con CloudWatch Logs Live Tail 391
Avvia una sessione Live Tail utilizzando il AWS CLI 391
solo per la stampa 392
interattivo 392
Avvia una sessione Live Tail nella console 394
Cross-account Centralizzazione dei log interregionali 398
Concetti di centralizzazione dei dati 398
Configurazione della centralizzazione dei log 400
Prerequisiti 400
Personalizzazione dei nomi dei gruppi di log di destinazione 403
Creazione di una regola di centralizzazione 404
Modifica di una regola di centralizzazione 407
Visualizzazione di una regola di centralizzazione 408
Eliminazione di una regola di centralizzazione 408
Monitoraggio e risoluzione dei problemi relativi alle regole di centralizzazione 409
Stato di integrità delle regole di centralizzazione 409
Monitoraggio delle chiamate API di centralizzazione con AWS CloudTrail 410
Consigli sul monitoraggio 411
Utilizzo di gruppi di log e flussi di log 414
Creazione di un gruppo di log 414
Invio di log a un gruppo di log 415
Visualizzazione dati di log 415
Ricerca di dati di log utilizzando i modelli dei filtri 416

Ricerca di voci di log utilizzando la console ... 416
Cerca nelle voci del registro utilizzando il AWS CLI ... 417
Cambiare da parametri a log ... 417
Risoluzione dei problemi ... 418
Modifica della conservazione dei dati di log ... 418
Proteggi i gruppi di log dall'eliminazione ... 419
Proteggi i gruppi di log dall'eliminazione ... 420
Attivazione della protezione dall'eliminazione ... 420
Assegnazione di tag ai gruppi di log ... 421
Nozioni di base sui tag ... 421
Monitoraggio dei costi mediante l'assegnazione di tag ... 422
Limitazioni applicate ai tag ... 422
Taggare i gruppi di log utilizzando il AWS CLI ... 423
Etichettatura dei gruppi di log utilizzando l' CloudWatch API Logs ... 423
Crittografa i dati di registro utilizzando AWS KMS ... 424
Limits ... 425
Passaggio 1: creare una chiave AWS KMS ... 330
Fase 2: Impostazione delle autorizzazioni sulla chiave KMS ... 426
Passaggio 3: imposta le autorizzazioni sul principale IAM chiamante ... 429
Fase 4: Associare una chiave KMS a un gruppo di log ... 390
Passo 5: Dissocia la chiave da un gruppo di log ... 390
Chiavi KMS e contesto di crittografia ... 432
Incremento della protezione dei dati di log sensibili con il mascheramento ... 435
Informazioni sulle policy di protezione dei dati ... 439
Autorizzazioni IAM necessarie per la creazione o l'uso di una policy di protezione dei dati ... 441
Creazione di una policy di protezione dei dati a livello di account ... 447
Creazione di una policy di protezione dei dati per un singolo gruppo di log ... 450
Visualizzazione di dati senza mascheramento ... 453
Report sui risultati della verifica ... 454
Tipi di dati che è possibile proteggere ... 456
Trasforma i log durante l'ingestione ... 501
Crea e gestisci trasformatori di log ... 503
Crea una politica di trasformazione a livello di account ... 504
Modifica o elimina una politica di trasformazione a livello di account ... 505
Crea un trasformatore di log-group-level log partendo da zero ... 506
Crea un log-group-level trasformatore copiandone uno esistente ... 507

Modifica un trasformatore log-group-level ... 508
Eliminare un trasformatore log-group-level ... 509
Processori configurabili di tipo parser ... 509
parseJSON ... 510
grok ... 511
csv ... 544
parseKeyValue ... 546
Processori integrati per i AWS log venduti ... 548
ParseWAF ... 549
parsePostGres ... 551
parseCloudFront ... 552
parseRoute53 ... 554
ParseVPC ... 555
parseToOCSF ... 556
Processori di mutazione delle stringhe ... 558
lowerCaseString ... 558
upperCaseString ... 559
SplitString ... 560
Stringa sostitutiva ... 562
TrimString ... 566
Processori JSON con mutazione ... 567
AddKeys ... 568
DeleteKeys ... 569
moveKeys ... 570
RenameKeys ... 572
CopyValue ... 574
listToMap ... 576
Processori di conversione di tipi di dati ... 581
TypeConverter ... 582
DateTimeConverter ... 583
Metriche ed errori di trasformazione ... 585
Accedi ai log con S3 Tables Integration ... 586
Comprendere l'integrazione delle tabelle S3 ... 586
Componenti principali ... 586
Flusso di dati verso le tabelle S3 ... 586
Quando utilizzare S3 Tables Integration ... 587

Prerequisiti ... 587
autorizzazioni IAM ... 588
Politica delle chiavi KMS (per dati crittografati) ... 589
Nozioni di base ... 591
Integrazione delle tabelle Amazon Amazon S3 con AWS servizi di analisi - Utilizzo della console Amazon S3 (Link) ... 592
Configura i permessi di Lake Formation ... 592
Connect con gli strumenti di analisi ... 593
Filtri di parametri ... 595
Concetti ... 596
Sintassi del modello di filtro per i filtri di parametri ... 598
Configurazione di valori di parametri per un filtro di parametri ... 599
Pubblicazione di dimensioni con parametri dal log eventi ... 600
Utilizzo di valori nei log eventi per incrementare il valore di un parametro ... 603
Creazione di filtri di parametri ... 604
Creazione di un filtro di parametri per un gruppo di log ... 604
Esempio: conteggio di log eventi ... 606
Esempio: conteggio delle occorrenze di un termine ... 607
Esempio: conteggio di codici HTTP 404 ... 609
Esempio: conteggio di codici HTTP 4xx ... 611
Esempio: estrazione di campi da un log di Apache e assegnare dimensioni ... 613
Elencazione di filtri di parametri ... 615
Eliminazione di un filtro di parametri ... 616
Filtri di sottoscrizione ... 617
Concetti ... 619
Filtri di sottoscrizione a livello di gruppo di log ... 620
Esempio 1: filtri di abbonamento con Amazon Kinesis Data Streams ... 621
Esempio 2: filtri di abbonamento con AWS Lambda ... 627
Esempio 3: filtri di abbonamento con Amazon Data Firehose ... 631
Esempio 4: filtri di abbonamento con Amazon Service OpenSearch ... 638
Account-level filtri di abbonamento ... 639
Esempio 1: filtri di abbonamento con Amazon Kinesis Data Streams ... 640
Esempio 2: filtri di abbonamento con AWS Lambda ... 646
Esempio 3: filtri di abbonamento con Amazon Data Firehose ... 651
Cross-account Abbonamenti interregionali ... 658

Cross-account Condivisione dei dati di registro tra regioni utilizzando Amazon Kinesis Data Streams ... 660
Cross-account Condivisione dei dati di registro tra regioni tramite Firehose ... 680
Cross-account Abbonamenti a livello di account interregionali che utilizzano Amazon Kinesis Data Streams ... 695
Cross-account Abbonamenti a livello di account interregionale che utilizzano Firehose ... 714
Prevenzione del "confused deputy" ... 726
Prevenzione della ricorsione dei log ... 728
Sintassi del modello di filtro ... 730
Espressioni regolari supportate ... 731
Corrispondenza dei termini usando espressioni regolari ... 734
Corrispondenza dei termini in log eventi non strutturati ... 734
Corrispondenza dei termini nei log eventi JSON ... 738
Corrispondenza dei termini in log eventi delimitati da spazi ... 747
Abilita la registrazione da AWS services ... 752
Destinazioni e autorizzazioni di registro supportate ... 755
Registrazione che richiede autorizzazioni aggiuntive [V1] ... 772
Registri inviati a CloudWatch Logs ... 772
Log inviati ad Amazon S3 ... 776
Log inviati a Firehose ... 780
Registrazione che richiede autorizzazioni aggiuntive [V2] ... 782
Esempi di configurazione della consegna dei log ... 783
Registri inviati a CloudWatch Logs ... 787
Log inviati ad Amazon S3 ... 791
Log inviati a Firehose ... 798
Tracce inviate a X-Ray ... 801
Service-specific autorizzazioni ... 804
Console-specific autorizzazioni ... 805
Cross-account esempio di consegna ... 806
Crea una fonte di consegna ... 807
Configurare la consegna a un bucket Amazon S3 ... 807
Configurare la consegna a uno stream Firehose ... 811
Cross-service confusa prevenzione sostitutiva ... 813
Automatizza l'abilitazione dei log con le regole di abilitazione della telemetria ... 814
Gerarchia di valutazione delle regole ... 815
Creazione di una regola di abilitazione ... 815

Abilita la registrazione da fonti di terze parti 816
Integrazioni dirette di terze parti 816
Fonti aggiuntive tramite AWS Security Hub CSPM 816
Esportazione di dati di log in Amazon S3 822
Concetti 823
Esportazione di dati di log in Amazon S3 tramite la console 824
Same-account esportazione (console) 824
Cross-account esporta (console) 832
Esporta i dati di registro su Amazon S3 utilizzando il AWS CLI 841
Same-account esportazione (CLI) 842
Cross-account esportazione (CLI) 849
Descrivere le attività di esportazione (CLI) 859
Annullare un'attività di esportazione (CLI) 860
Streaming di dati su Service OpenSearch 862
Prerequisiti 862
Sottoscrivi un gruppo di log a OpenSearch Service 863
Esempi di codice 865
Nozioni di base 866
Ciao Logs CloudWatch 867
Azioni 868
Scenari 943
Configura la connettività dei servizi container 944
Creazione della tua prima funzione serverless 961
Scopri le nozioni di base sull'ingestione di syslog 972
Eseguire una query di grandi dimensioni 984
Utilizzo degli eventi pianificati per invocare una funzione Lambda 1021
Crittografa le tabelle di ricerca utilizzando AWS KMS 1025
Come viene utilizzato CloudWatch Logs per le tabelle di ricerca AWS KMS 1026
Autorizzazioni richieste 1026
Passaggio 1: creare una chiave AWS KMS 1027
Fase 2: Impostazione delle autorizzazioni sulla chiave KMS 1028
Fase 3: Associare una chiave KMS a una tabella di ricerca 1029
Considerazioni 1030
Sicurezza 1031
Protezione dei dati 1032
Crittografia dei dati a riposo 1033

Crittografia dei dati in transito 1033
Gestione dell'identità e degli accessi 1033
Autenticazione 1034
Controllo accessi 1034
Panoramica sulla gestione degli accessi 1034
Utilizzo di policy basate su identità (policy IAM) 1040
Convalida della conformità 1069
Resilienza 1070
Sicurezza dell'infrastruttura 1070
Endpoint VPC di interfaccia 1070
Disponibilità 1071
Creazione di un endpoint VPC per i log CloudWatch 1071
Verifica della connessione tra il tuo VPC e Logs CloudWatch 1071
Controllo dell'accesso all'endpoint VPC CloudWatch Logs 1072
Supporto delle chiavi di contesto VPC 1073
Registrazione delle operazioni dell'API e della console con AWS CloudTrail 1074
Informazioni sulla generazione di query in CloudTrail 1079
Informazioni sulle voci dei file di log di 1080
Monitoraggio dell'utilizzo con metriche CloudWatch 1082
CloudWatch Parametri dei log 1082
Dimensioni per le metriche di Logs CloudWatch 1087
Metriche e dimensioni del trasformatore di log 1088
Metriche e dimensioni di centralizzazione 1089
CloudWatch Registra le metriche di utilizzo del servizio 1093
Service Quotas 1096
Gestione delle quote del servizio CloudWatch Logs 1108
Cronologia dei documenti 1110
AWS Glossario 1127
..... mcxxviii

Che cos'è Amazon CloudWatch Logs?

Puoi utilizzare Amazon CloudWatch Logs per monitorare, archiviare e accedere ai tuoi file di log da istanze Amazon Elastic Compute Cloud EC2 (Amazon) AWS CloudTrail, Route 53 e altre fonti.

CloudWatch Logs ti consente di centralizzare i log di tutti i sistemi, le applicazioni e i AWS servizi che utilizzi, in un unico servizio altamente scalabile. Potrai quindi visualizzarli facilmente, cercarli per codici o modelli di errore specifici, filtrarli in base a campi specifici o archiviarli in modo sicuro per analisi future. CloudWatch I registri consentono di visualizzare tutti i registri, indipendentemente dalla loro origine, come un flusso unico e coerente di eventi ordinati per ora.

CloudWatch Logs supporta anche l'interrogazione dei log con un potente linguaggio di query, il controllo e il mascheramento dei dati sensibili nei log e la generazione di metriche a partire dai log utilizzando filtri o un formato di registro incorporato.

CloudWatch Logs supporta due classi di log. I gruppi di log nella classe di CloudWatch log Logs Standard supportano tutte le funzionalità CloudWatch Logs. I gruppi di log della classe di CloudWatch log Logs Infrequent Access comportano costi di ingestione inferiori e supportano un sottoinsieme delle funzionalità della classe Standard. Per ulteriori informazioni, consulta Classi di registro.

Funzionalità

- Due classi di log per una maggiore flessibilità: CloudWatch Logs offre due classi di log in modo da poter disporre di un'opzione conveniente per i log a cui si accede di rado. È inoltre disponibile un'opzione completa per i log che richiedono il monitoraggio in tempo reale o altre funzionalità. Per ulteriori informazioni, consulta Classi di registro.
- Interroga i dati di registro: puoi utilizzare CloudWatch Logs Insights per cercare e analizzare in modo interattivo i dati di registro. È possibile eseguire interrogazioni per rispondere in modo più efficiente ed efficace ai problemi operativi. CloudWatch Logs Insights include un linguaggio di interrogazione creato appositamente con pochi comandi semplici ma potenti. Per iniziare, forniamo query di esempio, descrizioni dei comandi, completamento automatico delle query e individuazione dei campi di log. Sono incluse query di esempio per diversi tipi di registri di servizio. AWS Per iniziare, consulta Analisi dei dati di registro con CloudWatch Logs Insights.
- Crea indici di campo per rendere le query più efficienti: puoi creare indici di campo dei campi nei tuoi eventi di registro. Quando poi si utilizza un indice di campo in una query di CloudWatch Logs Insights, la query tenta di ignorare l'elaborazione degli eventi di registro che sono noti per non includere il campo indicizzato. Questa query riduce il volume di scansione delle query, rendendo

possibile la restituzione dei risultati più rapidamente. Per iniziare, consulta Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione.

- Rileva ed esegui il debug con Live Tail: puoi usare Live Tail per risolvere rapidamente gli incidenti visualizzando un elenco di streaming dei nuovi log eventi man mano che vengono importati. Puoi visualizzare, filtrare ed evidenziare i log importati quasi in tempo reale, in modo da poter rilevare e risolvere rapidamente i problemi. Puoi filtrare i log in base ai termini specificati e anche evidenziare quelli che contengono termini specifici per in modo da poter trovare rapidamente ciò che stai cercando. Per ulteriori informazioni, consulta Risolvi i problemi con CloudWatch Logs Live Tail.
- Monitora i log dalle EC2 istanze Amazon: puoi utilizzare CloudWatch Logs per monitorare applicazioni e sistemi utilizzando i dati di registro. Ad esempio, CloudWatch Logs può tenere traccia del numero di errori che si verificano nei log delle applicazioni e inviarti una notifica ogni volta che il tasso di errori supera una soglia specificata. CloudWatch Logs utilizza i dati di registro per il monitoraggio, quindi non sono necessarie modifiche al codice. Ad esempio, è possibile monitorare i registri delle applicazioni per termini letterali specifici (come "NullReferenceException«) o contare il numero di occorrenze di un termine letterale in una posizione particolare nei dati di registro (come i codici di stato «404" in un log di accesso Apache). Quando viene trovato il termine che state cercando, CloudWatch Logs riporta i dati in base a una metrica specificata dall'utente. CloudWatch I dati di log vengono crittografati durante il transito e mentre sono a riposo. Per iniziare, consulta Guida introduttiva ai CloudWatch registri.
- Monitora gli eventi AWS CloudTrail registrati: puoi creare allarmi CloudWatch e ricevere notifiche relative a particolari attività API acquisite CloudTrail e utilizzare la notifica per eseguire la risoluzione dei problemi. Per iniziare, consulta Invio di CloudTrail eventi ai CloudWatch registri nella Guida per l'AWS CloudTrail utente.
- Verifica e maschera i dati sensibili: se hai dati sensibili nei tuoi log, puoi contribuire a salvaguardarli con policy di protezione dei dati. Queste policy consentono di controllare e mascherare i dati di log sensibili. Se abiliti la protezione dei dati, i dati sensibili che corrispondono agli identificatori dei dati che hai selezionato vengono mascherati per impostazione predefinita. Per ulteriori informazioni, consulta Incremento della protezione dei dati di log sensibili con il mascheramento.
- Conservazione dei log: per impostazione predefinita, i log vengono conservati a tempo indeterminato e non scadono mai. Puoi modificare la policy di conservazione per ogni gruppo di log mantenendo la conservazione a tempo indeterminato o scegliendo periodi di conservazione compresi tra 10 anni e un giorno.
- Protezione dall'eliminazione: una protezione che impedisce l'eliminazione accidentale dei gruppi di log e dei relativi flussi di log. Se abilitata su un gruppo di log, la protezione dall'eliminazione blocca tutte le operazioni di eliminazione finché non viene disabilitata in modo esplicito. Per impostazione

predefinita, la protezione da eliminazione non è abilitata. Questa funzionalità opzionale aiuta a proteggere i dati operativi e di conformità critici dalla rimozione involontaria, come i gruppi di log che contengono dati di controllo e i registri delle applicazioni di produzione per la risoluzione dei problemi e l'analisi.

- Archivia i dati di registro: è possibile utilizzare CloudWatch Logs per archiviare i dati di registro in uno storage altamente durevole. L'agente CloudWatch Logs semplifica l'invio rapido di dati di registro ruotati e non ruotati da un host al servizio di registro. Puoi quindi accedere ai dati di log grezzi in caso di necessità.
- Registra le interrogazioni DNS di Route 53: puoi utilizzare CloudWatch Logs per registrare le informazioni sulle query DNS ricevute da Route 53. Per ulteriori informazioni, consulta Registrazione delle query DNS nella Guida per sviluppatori di Amazon Route 53.
- Centralizza i log tra account e regioni: puoi utilizzare CloudWatch Logs Centralization per definire regole di centralizzazione tra account e regioni che replicano i dati di registro acquisiti in ambienti con più account e regioni in un'area e un account centrali. È possibile configurare un'area di backup all'interno dell'account centrale per una maggiore resilienza, definire le impostazioni di crittografia per i gruppi di log appena creati nell'account centrale e creare filtri metrici e di abbonamento sugli eventi di registro da account e regioni di origine specifici con funzionalità di filtraggio avanzate.

Servizi correlati AWS

I seguenti servizi vengono utilizzati insieme ai registri: CloudWatch

- AWS CloudTrailè un servizio web che consente di monitorare le chiamate effettuate all'API CloudWatch Logs per l'account, incluse le chiamate effettuate da Console di gestione AWS, AWS Command Line Interface (AWS CLI) e altri servizi. Quando CloudTrail la registrazione è attivata, CloudTrail acquisisce le chiamate API nel tuo account e invia i file di registro al bucket Amazon S3 specificato. Ogni file di log può contenere uno o più record, in base al numero di operazioni da eseguire per soddisfare una richiesta. Per ulteriori informazioni su, consulta What AWS CloudTrail Is? AWS CloudTrail nella Guida AWS CloudTrail per l'utente. Per un esempio del tipo di dati che vengono CloudWatch scritti nei file di CloudTrail registro, vedereLogging CloudWatch Logs, API e operazioni della console in AWS CloudTrail.
- AWS Identity and Access Management (IAM) è un servizio web che ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse per i tuoi utenti. Utilizza IAM per controllare chi può utilizzare le tue risorse AWS (autenticazione), quali risorse e in che modo (autorizzazione). Per ulteriori informazioni, consulta Che cos'è IAM? nella Guida per l'utente di IAM.

- Amazon Kinesis Data Streams è un servizio Web che puoi utilizzare per l'acquisizione e l'aggregazione di dati rapidamente e continuamente. Il tipo di dati utilizzato include dati di logo dell'infrastruttura IT, log di applicazioni, social media, feed di dati di mercato e dati clickstream Web. Poiché il tempo di risposta per il consumo e l'elaborazione dei dati è in tempo reale, l'elaborazione è in genere leggera. Per ulteriori informazioni, consulta Cos'è Amazon Kinesis Data Streams? nella Guida per gli sviluppatori di Amazon Kinesis Data Streams.
- AWS Lambda è un servizio Web che puoi utilizzare per creare applicazioni che rispondono rapidamente a nuove informazioni. Carica il codice dell'applicazione come funzioni Lambda e Lambda eseguirà il codice in un'infrastruttura di calcolo ad alta disponibilità ed eseguirà tutte le attività di amministrazione delle risorse di calcolo, tra cui la manutenzione del server e del sistema operativo, il provisioning della capacità e la scalabilità automatica, la distribuzione di patch di sicurezza e per il codice e il monitoraggio e la registrazione del codice. Tutto quello che occorre fare è fornire il proprio codice in una delle lingue supportate da Lambda. Per ulteriori informazioni, consulta Cos'è AWS Lambda? nella Guida per gli AWS Lambda sviluppatori.

Prezzi

Quando ti registri AWS, puoi iniziare a usare CloudWatch Logs gratuitamente utilizzando il piano AWS gratuito.

Le tariffe standard si applicano ai log archiviati da altri servizi che utilizzano CloudWatch Logs (ad esempio, i log di flusso di Amazon VPC e i log Lambda).

Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Per ulteriori informazioni su come analizzare i costi e l'utilizzo di CloudWatch Logs e CloudWatch per le best practice su come ridurre i costi, consulta CloudWatch Fatturazione e costi.

Concetti di Amazon CloudWatch Logs

La terminologia e i concetti fondamentali per la comprensione e l'uso di CloudWatch Logs sono descritti di seguito.

Classe di registro

CloudWatch Logs offre due classi di gruppi di log. La classe di registro Standard è un'opzione completa per i registri che richiedono il monitoraggio in tempo reale o per i registri a cui si accede

frequentemente. La classe di log Infrequent Access è un'opzione a basso costo per i log a cui si accede meno frequentemente. Supporta un sottoinsieme delle funzionalità della classe di registro Standard.

Eventi di log

Un evento di log è un record di alcune attività registrate dall'applicazione o dalla risorsa monitorata. Il record degli eventi di registro che CloudWatch Logs comprende contiene due proprietà: il timestamp di quando si è verificato l'evento e il messaggio non elaborato dell'evento. I messaggi di evento devono essere in codifica UTF-8.

Flussi di log

Un flusso di log è una sequenza di log eventi che condividono la stessa origine. Più precisamente, un flusso di log in genere è destinato a rappresentare la sequenza di eventi provenienti dall'istanza dell'applicazione o dalla risorsa monitorata. Ad esempio, un flusso di log può essere associata a un log di accesso Apache in un determinato host. Quando non è più necessario un flusso di log, è possibile eliminarlo utilizzando il comando aws logs. delete-log-stream

Gruppi di log

I gruppi di log definiscono i gruppi dei flussi di log che condividono le stesse impostazioni di conservazione, monitoraggio e controllo degli accessi. Ogni flusso di log deve appartenere a un gruppo di log. Ad esempio, se disponi di un flusso di log separato per i log di accesso Apache da ogni host, puoi raggruppare tali flussi di log in una singola chiamata di gruppi di log `MyWebsite.com/Apache/access_log`.

Non vi è alcun limite al numero di flussi di log che possono appartenere a un gruppo di log.

Filtri di parametri

Puoi utilizzare filtri parametri per estrarre osservazioni sui parametri dagli eventi acquisiti e trasformarle in punti dati in un parametro CloudWatch . I filtri di parametro vengono assegnati ai gruppi di log e tutti i filtri assegnati a un gruppo di log vengono applicati ai relativi flussi di log.

Impostazioni della conservazione

Le impostazioni di conservazione possono essere utilizzate per specificare per quanto tempo gli eventi di registro vengono conservati nei CloudWatch registri. Gli eventi di log scaduti vengono eliminati automaticamente. Proprio come i filtri parametro, anche le impostazioni della conservazione vengono assegnate ai gruppi di log e la conservazione assegnata a un gruppo di log viene applicata ai relativi flussi di log.

Protezione da eliminazione

La protezione dall'eliminazione è una protezione che impedisce l'eliminazione accidentale dei gruppi di log e dei relativi flussi di log. Se abilitata su un gruppo di log, la protezione dall'eliminazione blocca tutte le operazioni di eliminazione finché non viene disabilitata esplicitamente. Per impostazione predefinita, la protezione da eliminazione non è abilitata. Questa funzionalità opzionale aiuta a proteggere i dati operativi e di conformità critici dalla rimozione involontaria, come i gruppi di log che contengono dati di controllo e i registri delle applicazioni di produzione per la risoluzione dei problemi e l'analisi.

Fatturazione e costi di Amazon CloudWatch Logs

Per informazioni dettagliate su come analizzare i costi e l'utilizzo di CloudWatch Logs e CloudWatch per le migliori pratiche su come ridurre i costi, consulta CloudWatch fatturazione e costi.

Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Quando ti registri AWS, puoi iniziare a usare CloudWatch Logs gratuitamente utilizzando il piano AWS gratuito.

Le tariffe standard si applicano ai log archiviati da altri servizi che utilizzano CloudWatch Logs (ad esempio, i log di flusso di Amazon VPC e i log Lambda).

Classi di registro

CloudWatch Logs offre due classi di gruppi di log:

- La classe di log CloudWatch Logs Standard è un'opzione completa per i log che richiedono il monitoraggio in tempo reale o per i log a cui si accede frequentemente.
- La classe di CloudWatch log Logs Infrequent Access è una nuova classe di log che è possibile utilizzare per consolidare i registri in modo conveniente. Questa classe di log offre un sottoinsieme di funzionalità di CloudWatch Logs, tra cui acquisizione gestita, archiviazione, analisi dei log tra account e crittografia, con un prezzo di ingestione inferiore per GB. La classe di log Infrequent Access è ideale per interrogazioni ad hoc e analisi forensi su log a cui si accede raramente. after-the-fact

> **Note**
>
> Per quanto riguarda i costi, le classi di registro Standard e Infrequent Access si differenziano solo per i costi di ingestione. I costi di storage e gli addebiti di CloudWatch Logs Insights sono gli stessi in ogni classe di log.

Per ulteriori informazioni sui prezzi di CloudWatch Logs, consulta la pagina dei CloudWatch prezzi di Amazon.

> **Important**
>
> Dopo aver creato un gruppo di log, la relativa classe di log non può essere modificata.

Funzionalità supportate

La tabella seguente elenca le funzionalità per ogni classe di log.

Funzionalità	Standard	Accesso poco frequente	
Inserimento e archiviazione dei log completamente gestiti	Sì ✓	Sì ✓	
Funzionalità per più account	Sì ✓	Sì ✓	
Crittografia con AWS KMS	Sì ✓	Sì ✓	
CloudWatch Comandi di interrogazione di Logs Insights	Sì ✓	Sì ✓ (La maggior parte dei comandi, vediI comandi QL di Logs Insights sono supportati nelle classi di log.)	
CloudWatch Registra i campi rilevati da Insights	Sì ✓	Sì ✓	
Facet	Sì ✓	No	
Utilizzo di CloudWatch Pipeline per trasformare i log	Sì ✓	Sì ✓	
Esportazione su Amazon S3	Sì ✓	Sì ✓	
Integrazione con tabelle S3	Sì ✓	Sì ✓	
Interrogazioni pianificate	Sì ✓	Sì ✓	
Utilizzo di OpenSearch PPL o OpenSearch SQL per eseguire interrogazioni in CloudWatch Logs Insights;	Sì ✓	Sì ✓	

Funzionalità	Standard	Accesso poco frequente	
Assistenza alle interrogazioni in linguaggio naturale	Sì ✓	No	
CloudWatch Rilevamento di anomalie nei registri	Sì ✓	No	
Live Tail	Sì ✓	No	
Indicizzazione dei campi	Sì ✓	No	
Confronta con l'intervallo di tempo precedente	Sì ✓	No	
Filtri di abbonamento	Sì ✓	No	
GetLogEvents e operazioni FilterLogEvents API	Sì ✓	Non supportato. Utilizza CloudWatch Logs Insights per visualizzare gli eventi di registro memorizzati in gruppi di log nella classe di log Infrequent Access.	
Filtri metrici	Sì ✓	No	
Inserimento dei log di Container Insights	Sì ✓	No	
Inserimento di log Lambda Insights	Sì ✓	No	

Funzionalità	Standard	Accesso poco frequente	
Protezione dei dati sensibili con mascheramento	Sì ✓	Sì ✓	
Formato di metriche incorporato	Sì ✓	No	

> **Note**
>
> Oltre a queste due classi di log, esiste una classe di `Delivery` log. Utilizzate la classe `Delivery` log solo per fornire AWS Lambda registri da archiviare in Amazon S3 o Amazon Data Firehose. Gli eventi di registro nei gruppi di log della classe Delivery vengono conservati CloudWatch Logs per due giorni. Questo periodo di conservazione è fisso e non può essere modificato. Questa classe di log non offre CloudWatch Logs funzionalità avanzate come le query CloudWatch Logs Insights.

Ottimizza i costi di storage con Amazon CloudWatch Logs Intelligent Tiering

Amazon CloudWatch Logs Intelligent Tiering classifica automaticamente i dati di log su tre livelli di accesso in base ai modelli di accesso: Standard, Infrequent Access e Archive Instant Access. Con Intelligent Tiering, puoi conservare registri ad alto volume e a lungo termine a livelli a basso costo senza alcun sovraccarico operativo. Invece di filtrare i log o trasferirli su soluzioni di storage separate, puoi conservarli in modo nativo in CloudWatch Logs e beneficiare della stessa esperienza di interrogazione indipendentemente dal livello in cui risiedono i dati.

CloudWatch Logs monitora i modelli di accesso e riclassifica automaticamente i dati a cui non si accede da 30 giorni al livello Infrequent Access e i dati a cui non si accede per 90 giorni al livello Archive Instant Access. Quando si accede a dati meno recenti, CloudWatch Logs li riconduce automaticamente al livello Standard per 30 giorni. Consolidando tutti i tuoi CloudWatch log in Logs, ottieni la piena visibilità in un unico strumento, eliminando il sovraccarico operativo della gestione di più soluzioni di storage e riducendo il tempo medio di risoluzione (MTTR) eseguendo interrogazioni, analizzando e inviando avvisi su tutti i log in un unico posto.

Per iniziare a ottimizzare i costi di archiviazione, abilita l'Intelligent Tiering a livello di account all'interno di una regione utilizzando la console o chiamando. CloudWatch `PutStorageTierPolicy` Una volta abilitato, Intelligent Tiering si applica a tutti i gruppi di log del tuo account all'interno di quella regione.

Amazon CloudWatch Logs Intelligent Tiering è disponibile in tutte le regioni AWS commerciali ad eccezione delle regioni del Medio Oriente (Bahrain) e del Medio Oriente (Emirati Arabi Uniti).

CloudWatch Livelli di accesso Logs Intelligent Tiering

L'esperienza di interrogazione rimane la stessa indipendentemente dal livello di accesso in cui si trovano i dati di registro.

Standard (impostazione predefinita)

> Questo è il livello di default. Tutti i dati di registro appena inseriti iniziano nel livello Standard. I dati di registro rimangono nel livello Standard finché vi si accede attivamente (da 0 a 30 giorni dall'ultimo accesso). Il livello Standard offre bassa latenza e prestazioni ad alto rendimento per query, esportazioni, live tail e recupero dei log.

Accesso poco frequente

Se non si accede ai dati di registro per 30 giorni consecutivi, si passa al livello Infrequent Access. Questo livello contiene i dati a cui è stato effettuato l'ultimo accesso da 31 a 90 giorni fa. Il livello Infrequent Access offre la stessa latenza delle query e lo stesso throughput del livello Standard, a un costo di archiviazione inferiore.

Accesso istantaneo all'archivio

Se non si accede ai dati di registro per 90 giorni consecutivi, si passa al livello Archive Instant Access. Questo livello contiene i dati a cui è stato effettuato l'ultimo accesso più di 90 giorni fa. Il livello Archive Instant Access fornisce la stessa latenza di interrogazione in millisecondi degli altri livelli, al minor costo di storage.

Azioni che costituiscono l'accesso

I seguenti modelli di accesso ripristinano automaticamente i dati di registro dal livello Infrequent Access o Archive Instant Access al livello Standard. Quando una di queste operazioni legge gli eventi di registro in un livello a costo inferiore, tali eventi di registro tornano al livello Standard per 30 giorni. Il timer si azzera a ogni accesso successivo.

> **Classificazione a livelli per la centralizzazione dei log**
>
> L'accesso ai log replicati in un account di destinazione tramite la centralizzazione dei log tra più account in più regioni promuove solo i dati di registro nell'account di destinazione.

- Uscita con configurazione di interrogazione o allarme: utilizza `StartQuery` o `CreateScheduledQuery` per interrogare i dati di registro con CloudWatch Logs Insights o per configurare gli allarmi nei log
- Filtraggio e recupero degli eventi di registro: utilizza `FilterLogEvents` o per leggere gli eventi di registro `GetLogEvents`
- Esportazione: esporta i dati di registro in Amazon S3 con `CreateExportTask`

Abilitazione dei CloudWatch log: Intelligent Tiering

Autorizzazioni richieste

Per gestire il livello di archiviazione del tuo account, hai bisogno delle seguenti autorizzazioni IAM:

- `logs:PutStorageTierPolicy`— Necessario per impostare o modificare il livello di archiviazione del tuo account.
- `logs:GetStorageTierPolicy`— Necessario per recuperare il livello di archiviazione attualmente configurato. Necessario anche per abilitare Intelligent Tiering dalla console.

Gli utenti con la policy `CloudWatchLogsFullAccess` gestita dispongono già di entrambe le autorizzazioni. Gli utenti con la policy `CloudWatchLogsReadOnlyAccess` gestita dispongono dell'`logs:GetStorageTierPolicy`autorizzazione.

L'esempio seguente mostra una dichiarazione politica che concede entrambe le autorizzazioni:

```
{
    "Effect": "Allow",
    "Action": [
        "logs:PutStorageTierPolicy",
        "logs:GetStorageTierPolicy"
    ],
    "Resource": "*"
}
```

Abilita l'utilizzo della console

Per abilitare Intelligent Tiering utilizzando la CloudWatch console:

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Impostazioni.
3. Nella pagina Impostazioni, scegli la scheda Registri.
4. Nella sezione Intelligent Tiering, scegli Abilita.

Abilita usando il AWS CLI

Per abilitare Intelligent Tiering, usa il comando: `put-storage-tier-policy`

```
aws logs put-storage-tier-policy --storage-tier INTELLIGENT_TIERING
```

Il comando restituisce il livello di archiviazione impostato e l'ora dell'ultimo aggiornamento:

```
{
    "storageTier": "INTELLIGENT_TIERING",
    "lastUpdatedTime": 1751328000000
}
```

Per disattivare Intelligent Tiering, reimposta il livello di storage su`STANDARD`:

```
aws logs put-storage-tier-policy --storage-tier STANDARD
```

Quando imposti nuovamente il livello di archiviazione su`STANDARD`, tutti i gruppi di log del tuo account vengono fatturati alla tariffa di archiviazione Standard.

Visualizzazione della configurazione attuale

Per visualizzare il livello di archiviazione corrente del tuo account, usa il `get-storage-tier-policy` comando:

```
aws logs get-storage-tier-policy
```

Il comando restituisce il livello di archiviazione corrente e l'ora dell'ultimo aggiornamento:

```
{
    "storageTier": "INTELLIGENT_TIERING",
    "lastUpdatedTime": 1751328000000
}
```

Monitoraggio dello storage per livello

Quando Intelligent Tiering è abilitato, CloudWatch Logs pubblica una metrica venduta che mostra la quantità di dati archiviati in ogni livello per ogni gruppo di log:

- Namespace: `AWS/Logs`
- Nome della metrica: `StoredBytes`

- Dimensioni: `LogGroupName`, `StorageType` (valori:`Standard`,`Infrequent_Access`,`Archival_Instant_Access`)

Puoi utilizzare questa metrica per creare CloudWatch dashboard che mostrino la suddivisione dello storage per livello, impostare allarmi sulle dimensioni dei livelli o monitorare l'ottimizzazione dei costi nel tempo.

> **Note**
>
> Questa metrica viene pubblicata una volta al giorno e potrebbe non riflettere le transizioni di livello in tempo reale.

Prezzi di CloudWatch Logs Intelligent Tiering

Con Intelligent Tiering, paghi tariffe di archiviazione diverse in base al livello in cui risiedono i tuoi dati. Non sono previsti costi aggiuntivi per l'attivazione di Intelligent Tiering e nessun costo per le transizioni di livello. Paghi solo per lo spazio di archiviazione consumato in ogni livello.

Per ulteriori informazioni sui prezzi, consulta i CloudWatch prezzi di Amazon.

Per visualizzare i costi di storage per livello, utilizza Athena e filtra in base ai tipi di utilizzo `TimedStorage-ByteHrs` (Standard), `TimedStorage-IA-ByteHrs` (Infrequent Access) e `TimedStorage-AIA-ByteHrs` (Archive Instant Access). Per ulteriori informazioni sui tipi di utilizzo dei CloudWatch log nella fattura, vedi Analisi, ottimizzazione e riduzione dei costi. CloudWatch

Guida introduttiva ai CloudWatch registri

Per raccogliere i log dalle istanze Amazon EC2 e dai server locali CloudWatch in Logs, utilizza l'agente unificato. CloudWatch Consente di raccogliere sia i log che i parametri avanzati con un agente. Offre il supporto tra sistemi operativi, inclusi i server che eseguono Windows Server. Questo agente fornisce inoltre prestazioni migliori.

Se utilizzi l' CloudWatch agente unificato per raccogliere i parametri, abilita la raccolta di CloudWatch parametri di sistema aggiuntivi, per la visibilità degli ospiti. Inoltre, supporta la raccolta dei parametri personalizzati utilizzando `StatsD` o `collectd`.

Per ulteriori informazioni, consulta Installazione dell' CloudWatch agente nella Amazon CloudWatch User Guide.

Il vecchio agente CloudWatch Logs, che supporta solo la raccolta di log dai server che eseguono Linux, è obsoleto e non è più supportato. Per informazioni sulla migrazione dal vecchio agente CloudWatch Logs all'agente unificato, consulta Creare il file di configurazione dell'agente con la procedura guidata. CloudWatch

Indice

- Prerequisiti
- Usa l' CloudWatch agente unificato per iniziare a usare Logs CloudWatch
- Usa l' CloudWatch agente precedente per iniziare a usare Logs CloudWatch
- Avvio rapido: CloudFormation da utilizzare per iniziare a utilizzare i log CloudWatch

Prerequisiti

Per utilizzare Amazon CloudWatch Logs è necessario un AWS account. Il tuo AWS account ti consente di utilizzare i servizi (ad esempio Amazon EC2) per generare log che puoi visualizzare nella CloudWatch console, un'interfaccia basata sul web. Inoltre, puoi installare e configurare AWS Command Line Interface ()AWS CLI.

Registrati per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedere Guida introduttiva a Account AWS nella Guida Gestione dell'account AWS di riferimento.

Impostazione dell'interfaccia a riga di comando

È possibile utilizzare il AWS CLI per eseguire operazioni di CloudWatch log.

Per informazioni su come installare e configurare AWS CLI, vedere Getting Setup with the AWS Command Line Interface nella Guida per l'AWS Command Line Interface utente.

Usa l' CloudWatch agente unificato per iniziare a usare Logs CloudWatch

Per ulteriori informazioni sull'utilizzo dell' CloudWatch agente unificato per iniziare a usare i log, consulta Collect Metrics and CloudWatch Logs from Amazon EC2 Instances and On-Premises Servers with the Agent nella Amazon User Guide. CloudWatch CloudWatch Per installare, configurare e avviare l'agente, completa la procedura di seguito riportata. Se non utilizzi l'agente per raccogliere anche i parametri, puoi ignorare tutte le sezioni che fanno riferimento ai CloudWatch parametri.

Se attualmente utilizzi il vecchio agente CloudWatch Logs e desideri passare al nuovo agente unificato, ti consigliamo di utilizzare la procedura guidata inclusa nel nuovo pacchetto dell'agente. Questa procedura guidata può leggere il file di configurazione corrente dell'agente CloudWatch Logs e configurare l' CloudWatchagente per raccogliere gli stessi registri. Per ulteriori informazioni sulla procedura guidata, consulta Create the CloudWatch Agent Configuration File with the Wizard nella Amazon CloudWatch User Guide.

Usa l' CloudWatch agente precedente per iniziare a usare Logs CloudWatch

> **Important**
>
> CloudWatch include un CloudWatch agente unificato in grado di raccogliere sia log che metriche dalle istanze EC2 e dai server locali. Il vecchio agente che utilizza solo i log è obsoleto e non è più supportato.
> Per informazioni sulla migrazione dal vecchio agente logs-only all'agente unificato, consulta Creare il file di configurazione dell'agente con la procedura guidata. CloudWatch
> Il resto di questa sezione spiega l'uso del vecchio agente CloudWatch Logs per i clienti che lo utilizzano ancora.

Utilizzando l'agente CloudWatch Logs, puoi pubblicare i dati di registro dalle istanze Amazon EC2 che eseguono Linux o Windows Server e gli eventi registrati da. AWS CloudTrail Ti consigliamo invece di utilizzare l'agente CloudWatch unificato per pubblicare i dati di registro. Per ulteriori informazioni sul nuovo agente, consulta Collect metrics and logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent nella Amazon User Guide. CloudWatch

Indice

- CloudWatch Registra i prerequisiti dell'agente
- Avvio rapido: installa e configura l'agente CloudWatch Logs su un'istanza Linux EC2 in esecuzione
- Avvio rapido: installa e configura l'agente CloudWatch Logs su un'istanza Linux EC2 al momento del lancio
- Avvio rapido: consenti alle istanze Amazon EC2 che eseguono Windows Server 2016 di inviare log a Logs utilizzando l'agente CloudWatch Logs CloudWatch
- Avvio rapido: consenti alle istanze Amazon EC2 che eseguono Windows Server 2012 e Windows Server 2008 di inviare log a Logs CloudWatch
- Segnala lo stato dell'agente CloudWatch Logs
- Avvia l'agente Logs CloudWatch
- Arresta l'agente CloudWatch Logs
- CloudWatch Registra il riferimento dell'agente

CloudWatch Registra i prerequisiti dell'agente

L'agente CloudWatch Logs richiede la versione Python 2.7, 3.0 o 3.3 e una delle seguenti versioni di Linux:

- Amazon Linux 2014.03.02 o versioni successive. Amazon Linux 2 non è supportato
- Server Ubuntu versione 12.04, 14.04 o 16.04
- CentOS versione 6, 6.3, 6.4, 6.5 o 7.0
- Red Hat Enterprise Linux (RHEL) versione 6.5 o 7.0
- Debian 8.0

Avvio rapido: installa e configura l'agente CloudWatch Logs su un'istanza Linux EC2 in esecuzione

Important

Il vecchio logs agent è obsoleto. CloudWatch include un agente unificato in grado di raccogliere sia i log che le metriche dalle istanze EC2 e dai server locali. Per ulteriori informazioni, consulta Guida introduttiva ai CloudWatch registri.
Per informazioni sulla migrazione dal vecchio agente CloudWatch Logs all'agente unificato, consulta Creare il file di configurazione dell'agente con la procedura guidata. CloudWatch
L'agente di log più vecchio supporta solo le versioni da 2.6 a 3.5 di Python. Inoltre, il vecchio agente CloudWatch Logs non supporta Instance Metadata Service versione 2 (). IMDSv2
Se il server utilizza IMDSv2, è necessario utilizzare l'agente unificato più recente anziché il precedente agente Logs. CloudWatch
Il resto di questa sezione spiega l'uso del vecchio agente CloudWatch Logs per i clienti che lo utilizzano ancora.

Tip

CloudWatch include un nuovo agente unificato in grado di raccogliere sia i log che le metriche dalle istanze EC2 e dai server locali. Se non stai già utilizzando il vecchio agente CloudWatch Logs, ti consigliamo di utilizzare il nuovo agente unificato. CloudWatch Per ulteriori informazioni, consulta Guida introduttiva ai CloudWatch registri.
Inoltre, l'agente precedente non supporta Instance Metadata Service versione 2 (). IMDSv2
Se il server utilizza IMDSv2, è necessario utilizzare l'agente unificato più recente anziché il vecchio CloudWatch agente Logs.
Il resto di questa sezione spiega l'uso del vecchio CloudWatch agente Logs.

Configura il vecchio agente CloudWatch Logs su un'istanza EC2 Linux in esecuzione

Puoi utilizzare il programma di installazione dell'agente CloudWatch Logs su un'istanza EC2 esistente per installare e configurare l'agente Logs. CloudWatch Dopo aver completato l'installazione, i log cominciano a fluire automaticamente dall'istanza al flusso di log creato durante l'installazione dell'agente. L'agente conferma l'avvio e continua l'esecuzione fino a quando non lo disabiliterai.

Oltre a utilizzare l'agente, puoi anche pubblicare i dati di registro utilizzando CloudWatch Logs SDK o l' AWS CLI API Logs. CloudWatch AWS CLI È più adatto per la pubblicazione di dati sulla riga di comando o tramite script. L'SDK CloudWatch Logs è più adatto per pubblicare i dati di registro direttamente dalle applicazioni o per creare un'applicazione di pubblicazione dei log personalizzata.

Passaggio 1: configura il tuo ruolo o utente IAM per Logs CloudWatch

L'agente CloudWatch Logs supporta i ruoli e gli utenti IAM. Se l'istanza dispone già di un ruolo IAM a essa associato, assicurati di includere la policy IAM sottostante. Se non disponi già di un ruolo IAM assegnato alla tua istanza, puoi utilizzare le tue credenziali IAM per le fasi successive o puoi assegnare un ruolo IAM a tale istanza. Per ulteriori informazioni, consulta la sezione relativa al collegamento di un ruolo IAM a un'istanza.

Per configurare il tuo ruolo o utente IAM per Logs CloudWatch

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Scegliere il ruolo selezionando il nome del ruolo (non selezionare la casella di controllo accanto al nome).
4. Scegliere Attach Policies (Collega policy), Create Policy (Crea policy).

   Verrà aperta una nuova scheda o finestra del browser.
5. Scegliere la scheda JSON e digitare il documento di policy JSON seguente.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "*"
    ]
  }
 ]
```

```
}
```

6. Al termine, selezionare Review policy (Rivedi policy). In Policy Validator (Validatore di policy) vengono segnalati eventuali errori di sintassi.
7. Nella pagina Review policy (Rivedi policy) digitare i valori per Name (Nome) e Description (Descrizione) (facoltativa) per la policy che si sta creando. Consulta il Riepilogo della policy per visualizzare le autorizzazioni concesse dalla policy. Seleziona Create policy (Crea policy) per salvare il proprio lavoro.
8. Chiudere la scheda o la finestra del browser e tornare alla pagina Add permissions (Aggiungi autorizzazioni) per il ruolo. Scegliere Refresh (Aggiorna) e quindi scegliere la nuova policy per collegarla al ruolo.
9. Scegli Attach Policy (Collega policy).

Fase 2: installare e configurare CloudWatch i log su un'istanza Amazon EC2 esistente

Il processo di installazione dell'agente CloudWatch Logs varia a seconda che l'istanza Amazon EC2 esegua Amazon Linux, Ubuntu, CentOS o Red Hat. Utilizza la procedura appropriata per la versione di Linux della tua istanza.

Per installare e configurare CloudWatch i log su un'istanza Amazon Linux esistente

A partire dall'AMI Amazon Linux 2014.09, l'agente CloudWatch Logs è disponibile come installazione RPM con il pacchetto awslogs. Le versioni precedenti di Amazon Linux possono accedere al pacchetto awslogs aggiornando l'istanza con il comando `sudo yum update -y`. Installando il pacchetto awslogs come RPM anziché utilizzare il programma di installazione CloudWatch Logs, l'istanza riceve aggiornamenti e patch regolari del pacchetto senza dover reinstallare manualmente l'agente Logs. AWS CloudWatch

> **Warning**
>
> Non aggiornate l'agente CloudWatch Logs utilizzando il metodo di installazione RPM se in precedenza avete utilizzato lo script Python per installare l'agente. Ciò potrebbe causare problemi di configurazione che impediscono all'agente CloudWatch Logs di inviare i log a. CloudWatch

1. Connessione a un'istanza Amazon Linux. Per ulteriori informazioni, consulta Connect to Your Instance nella Amazon EC2 User Guide.

Per ulteriori informazioni sui problemi di connessione, consulta Troubleshooting Connecting to Your Instance nella Amazon EC2 User Guide.

2. Aggiornare l'istanza Amazon Linux per prelevare le modifiche più recenti nei repository del pacchetto.

```
sudo yum update -y
```

3. Installare il pacchetto `awslogs`. Questo è il metodo consigliato per l'installazione di awslogs nelle istanze Amazon Linux.

```
sudo yum install -y awslogs
```

4. Modificare il file `/etc/awslogs/awslogs.conf` per configurare i log da monitorare. Per ulteriori informazioni sulla modifica di questo file, consulta CloudWatch Registra il riferimento dell'agente.
5. Per impostazione predefinita, `/etc/awslogs/awscli.conf` punta alla Regione us-east-1. Per portare i log in un'altra Regione, modificare il file `awscli.conf` e specificare la Regione.
6. Avviare il servizio `awslogs`.

```
sudo service awslogs start
```

Se si esegue Amazon Linux 2, avviare il servizio `awslogs` con il comando seguente.

```
sudo systemctl start awslogsd
```

7. Controlla che il file `/var/log/awslogs.log` non contenga errori registrati all'avvio del servizio (facoltativo).
8. Esegui il comando seguente per avviare il servizio `awslogs` a ogni avvio del sistema (facoltativo).

```
sudo chkconfig awslogs on
```

Se si esegue Amazon Linux 2, utilizzare il comando seguente per avviare il servizio a ogni avvio del sistema.

```
sudo systemctl enable awslogsd.service
```

9. Il gruppo di log e il flusso di log appena creati dovrebbero venire visualizzati nella console CloudWatch dopo qualche minuto di esecuzione dell'agente.

   Per ulteriori informazioni, consulta Visualizza i dati di registro inviati ai registri CloudWatch.

Per installare e configurare CloudWatch i log su un'istanza esistente di Ubuntu Server, CentOS o Red Hat

Se utilizzi un'AMI che esegue Ubuntu Server, CentOS o Red Hat, utilizza la seguente procedura per installare manualmente l'agente CloudWatch Logs sulla tua istanza.

1. Connettiti all'istanza EC2. Per ulteriori informazioni, consulta Connect to Your Instance nella Amazon EC2 User Guide.

   Per ulteriori informazioni sui problemi di connessione, consulta Troubleshooting Connecting to Your Instance nella Amazon EC2 User Guide.
2. Esegui il programma di installazione dell'agente CloudWatch Logs utilizzando una delle due opzioni. È possibile eseguirlo direttamente da Internet o scaricare i file ed eseguirlo autonomamente.

   > **Note**
   >
   > Se si esegue CentOS 6.x o Red Hat 6.x o Ubuntu 12.04, utilizzare la procedura di download ed esecuzione del programma di installazione in maniera autonoma. L'installazione dell'agente CloudWatch Logs direttamente da Internet non è supportata su questi sistemi.

   > **Note**
   >
   > Su Ubuntu, esegui `apt-get update` prima di eseguire i comandi sottostanti.

   Per eseguirlo direttamente da Internet, utilizza i comandi seguenti e segui le istruzioni:

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
   setup.py -O
   ```

```
sudo python ./awslogs-agent-setup.py --region us-east-1
```

Se il comando precedente non funziona, prova il seguente:

```
sudo python3 ./awslogs-agent-setup.py --region us-east-1
```

Per scaricarlo ed eseguirlo autonomamente, utilizza i comandi seguenti e segui le istruzioni:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py -O
```

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/
AgentDependencies.tar.gz -O
```

```
tar xvf AgentDependencies.tar.gz -C /tmp/
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1 --dependency-path /tmp/
AgentDependencies
```

È possibile installare l'agente CloudWatch Logs specificando us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-northeast-1, ap-northeast-1, ap-southeast-1, ap-southeast-2, eu-Regioni eu-central-1, eu-west-1 o sa-east-1.

> **Note**
>
> Per ulteriori informazioni sulla versione attuale e sulla cronologia delle versioni di `awslogs-agent-setup`, consulta CHANGELOG.txt.

Il programma di installazione dell'agente Logs richiede determinate informazioni durante la configurazione. CloudWatch Prima di iniziare, è necessario sapere quale file di log monitorare e il suo formato timestamp. È inoltre necessario disporre delle informazioni seguenti.

Elemento	Description
AWS ID della chiave di accesso	Premi invio se utilizzi un ruolo IAM. Altrimenti, inserisci l'ID della tua chiave di AWS accesso.
AWS chiave di accesso segreta	Premi invio se utilizzi un ruolo IAM. Altrimenti, inserisci la tua chiave di accesso AWS segreta.
Nome della Regione predefinito	Premere Invio. La Regione di default è us-east-2. Si può impostare questo a us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, o sa-east-1.
Formato di output predefinito	Lascia questo campo vuoto e premi invio.
Percorso del file di log da caricare	Posizione del file che contiene i dati di log da inviare. Il programma di installazione ti suggerisce un percorso.
Nome del gruppo di log di destinazione	Nome del gruppo di log. Il programma di installazione ti suggerisce un nome per il gruppo di log.
Nome del flusso di log di destinazione	Per impostazione predefinita, questo è il nome dell'host. Il programma di installazione ti suggerisce un nome host.
Formato timestamp	Specifica il formato del timestamp all'interno del file di log specificato. Scegli il valore di personalizzazione per specificare il tuo formato.
Posizione iniziale	In che modo i dati vengono caricati. Imposta questo valore su start_of_file per caricare tutto il contenuto del file. Imposta su end_of_file per caricare solo i dati appena aggiunti.

Dopo aver completato questa procedura, il programma di installazione chiede se intendi configurare un altro file di log. Puoi eseguire il processo quante volte lo desideri per ogni file di log. Se non hai altri file di log da monitorare, scegli N quando richiesto dal programma di

installazione per configurare un altro log. Per ulteriori informazioni sulle impostazioni all'interno del file di configurazione dell'agente, consulta CloudWatch Registra il riferimento dell'agente.

> **Note**
>
> La configurazione di più origini di log per l'invio di dati a un unico flusso di log non è supportata.

3. Il gruppo di log e il flusso di log appena creati dovrebbero venire visualizzati nella console CloudWatch dopo qualche minuto di esecuzione dell'agente.

   Per ulteriori informazioni, consulta Visualizza i dati di registro inviati ai registri CloudWatch.

Avvio rapido: installa e configura l'agente CloudWatch Logs su un'istanza Linux EC2 al momento del lancio

> **Tip**
>
> Il vecchio agente CloudWatch Logs discusso in questa sezione sta per diventare obsoleto. Ti consigliamo vivamente di utilizzare invece il nuovo CloudWatch agente unificato in grado di raccogliere sia i log che le metriche. Inoltre, il vecchio agente CloudWatch Logs richiede Python 3.3 o versioni precedenti e queste versioni non sono installate su nuove istanze EC2 per impostazione predefinita. Per ulteriori informazioni sull'agente unificato, consulta Installazione CloudWatch dell'agente. CloudWatch
> Il resto di questa sezione spiega l'uso del vecchio agente CloudWatch Logs.

Installazione del vecchio agente CloudWatch Logs su un'istanza EC2 Linux all'avvio

Puoi utilizzare i dati utente di Amazon EC2, una funzionalità di Amazon EC2 che consente il trasferimento di informazioni parametriche all'istanza all'avvio, per installare e configurare CloudWatch l'agente Logs su quell'istanza. Per trasmettere le informazioni di installazione e configurazione dell'agente CloudWatch Logs ad Amazon EC2, puoi fornire il file di configurazione in una posizione di rete, ad esempio un bucket Amazon S3.

La configurazione di più origini di log per l'invio di dati a un unico flusso di log non è supportata.

Prerequisito

Crea un file di configurazione dell'agente che descriva tutti i gruppi di log e i flussi di log. Si tratta di un file di testo che descrive i file di log da monitorare e i gruppi di log e i flussi di log sui quali caricarli. L'agente utilizza questo file di configurazione e inizia a monitore e a caricare tutti i file di log in esso descritti. Per ulteriori informazioni sulle impostazioni all'interno del file di configurazione dell'agente, consulta CloudWatch Registra il riferimento dell'agente.

Di seguito viene illustrato un esempio di file di configurazione dell'agente per Amazon Linux 2

```
[general]
state_file = /var/lib/awslogs/state/agent-state

[/var/log/messages]
file = /var/log/messages
log_group_name = /var/log/messages
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Di seguito un esempio di file di configurazione dell'agente per Ubuntu

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/syslog]
file = /var/log/syslog
log_group_name = /var/log/syslog
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Configurazione di un ruolo IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione scegli Policies (Policy), quindi Create Policy (Crea policy).
3. Nella pagina Create Policy (Crea policy), in Create Your Own Policy (Crea la tua policy), scegli Select (Seleziona). Per ulteriori informazioni sulla creazione di policy personalizzate, consulta IAM Policies for Amazon EC2 nella Amazon EC2 User Guide.
4. Nella pagina Review Policy (Rivedi policy), in Policy Name (Nome policy) digita un nome per la policy.
5. In Policy Document (Documento policy), copia la policy seguente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

6. Scegli Crea policy.
7. Nel riquadro di navigazione selezionare Roles (Ruoli), quindi Create new role (Crea nuovo ruolo).
8. Nella pagina Set Role Name (Imposta nome ruolo), digita un nome per il ruolo e scegli Next Step (Fase successiva).
9. Nella pagina Select Role Type (Seleziona tipo di ruolo), scegli Select (Seleziona) accanto ad Amazon EC2.
10. Nella pagina Attach Policy (Collega policy), nell'intestazione della tabella, scegli Policy Type (Tipo di policy), Customer Managed (Gestito dal cliente).
11. Seleziona la policy IAM creata, quindi scegli Next Step (Fase successiva).
12. Selezionare Crea ruolo.

Per ulteriori informazioni sugli utenti e sulle policy, consulta Utenti e gruppi IAM e Gestione delle policy IAM nella Guida per l'utente IAM.

Per avviare una nuova istanza e abilitare i log CloudWatch

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Scegliere Launch Instance (Avvia istanza).

   Per ulteriori informazioni, consulta Launching an Instance in Amazon EC2 User Guide.
3. Nella pagina Step 1: Choose an Amazon Machine Image (AMI) (Fase 1: scegli un'Amazon Machine Image (AMI)), seleziona il tipo di istanza Linux da avviare, quindi, nella pagina Step 2: Choose an Instance Type (Fase 2: scegli un tipo di istanza), scegli Next: Configure Instance Details (Successivo: configura i dettagli dell'istanza).

   Assicurati che cloud-init sia incluso nell'Amazon Machine Image (AMI). Amazon Linux AMIs e AMIs per Ubuntu e RHEL includono già cloud-init, ma CentOS e altri potrebbero non esserlo. AMIs Marketplace AWS
4. Nella pagina Step 3: Configure Instance Details (Fase 3: configura i dettagli dell'istanza), in IAM role (Ruolo IAM), seleziona il ruolo IAM creato.
5. Sotto a Advanced Details (Dettagli avanzati), in User data (Dati utente), incolla il seguente script nella casella. In seguito aggiorna tale script cambiando il valore dell'opzione -c nella posizione del file di configurazione dell'agente.

   ```
   #!/bin/bash
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
   setup.py -O
   chmod +x ./awslogs-agent-setup.py
   ./awslogs-agent-setup.py -n -r us-east-1 -c s3://amzn-s3-demo-bucket/my-config-file
   ```
6. Apporta tutte le altre modifiche all'istanza, controlla le impostazioni di avvio, quindi scegli Launch (Avvia).
7. Dovresti vedere il gruppo di log e il flusso di log appena creati nella CloudWatch console dopo che l'agente è stato in esecuzione per alcuni istanti.

   Per ulteriori informazioni, consulta Visualizza i dati di registro inviati ai registri CloudWatch.

Avvio rapido: consenti alle istanze Amazon EC2 che eseguono Windows Server 2016 di inviare log a Logs utilizzando l'agente CloudWatch Logs CloudWatch

> **Tip**
>
> CloudWatch include un nuovo agente unificato in grado di raccogliere sia i log che le metriche dalle istanze EC2 e dai server locali. Ti consigliamo di utilizzare il nuovo agente CloudWatch unificato. Per ulteriori informazioni, consulta Guida introduttiva ai CloudWatch registri.
> Il resto di questa sezione spiega l'uso del vecchio agente Logs. CloudWatch

Consenti alle istanze Amazon EC2 che eseguono Windows Server 2016 di inviare log a Logs utilizzando il vecchio CloudWatch agente Logs CloudWatch

È possibile utilizzare diversi metodi per consentire alle istanze che eseguono Windows Server 2016 di inviare i log ai registri. CloudWatch Nella procedura di questa sezione viene utilizzato Run Command di Systems Manager. Per informazioni sugli altri metodi possibili, consulta Invio di registri, eventi e contatori delle prestazioni ad Amazon. CloudWatch

Fasi

- Download del file di configurazione di esempio
- Configura il file JSON per CloudWatch
- Creare un ruolo IAM per Systems Manager
- Verifica dei prerequisiti di Systems Manager
- Verifica dell'accesso a Internet
- Abilita CloudWatch i log utilizzando il comando Systems Manager Run

Download del file di configurazione di esempio

Download il seguente file di esempio su computer: `AWS.EC2.Windows.CloudWatch.json`.

Configura il file JSON per CloudWatch

È possibile determinare a quali log inviare i dati CloudWatch specificando le proprie scelte in un file di configurazione. Il processo di creazione di questo file e l'indicazione delle scelte possono richiedere

30 minuti o più per il completamento. Una volta completata questa attività una volta, puoi riutilizzare il file di configurazione su tutte le istanze.

Fasi

- Fase 1: Abilitare i log CloudWatch
- Fase 2: Configurare le impostazioni per CloudWatch
- Fase 3: configurazione dei dati da inviare
- Fase 4: configurazione del controllo di flusso
- Fase 5: salvataggio del contenuto JSON

Fase 1: Abilitare i log CloudWatch

Nella parte superiore del file JSON, modifica "falso" in "vero" per `IsEnabled`:

```
"IsEnabled": true,
```

Fase 2: Configurare le impostazioni per CloudWatch

Specifica le credenziali, la Regione, un nome per il gruppo di log e uno spazio dei nomi del flusso di log. Ciò consente all'istanza di inviare i dati di registro ai CloudWatch registri. Per inviare gli stessi dati di registro a posizioni diverse, puoi aggiungere sezioni aggiuntive con sezioni uniche IDs (ad esempio, "CloudWatchLogs2" e CloudWatchLogs 3") e una regione diversa per ogni ID.

Per configurare le impostazioni per l'invio dei dati di registro ai registri CloudWatch

1. Nel file JSON, individuare la sezione `CloudWatchLogs`.

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. Lasciare vuoti i campi `AccessKey` e `SecretKey`. Puoi configurare le credenziali tramite un ruolo IAM.
3. In `Region` digitare la Regione alla quale inviare dati di log (ad esempio, `us-east-2`).
4. Per `LogGroup` digita il nome del gruppo di log. Questo nome viene visualizzato nella schermata Log Groups (Gruppi di log) nella console CloudWatch.
5. Per `LogStream` digita il flusso di log di destinazione. Questo nome viene visualizzato nella schermata Log Groups > Streams della CloudWatch console.

   Se si utilizza `{instance_id}`, il nome del flusso di log predefinito è l'ID istanza di tale istanza.

   Se specifichi un nome di log stream che non esiste già, CloudWatch Logs lo crea automaticamente per te. È possibile definire un nome per il flusso di log utilizzando una stringa letterale, le variabili predefinite `{instance_id}`, `{hostname}` e `{ip_address}` oppure una combinazione di queste.

Fase 3: configurazione dei dati da inviare

È possibile inviare i dati del registro degli eventi, i dati di Event Tracing for Windows (ETW) e altri dati di registro ai registri. CloudWatch

Per inviare i dati del registro degli eventi dell'applicazione Windows a Logs CloudWatch

1. Nel file JSON, individuare la sezione `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Per `Levels`, specifica il tipo di messaggio da caricare. È possibile specificare uno dei seguenti valori:

   - **1**: vengono caricati solo i messaggi di errore.
   - **2**: vengono caricati solo i messaggi di avviso.

- **4**: vengono caricati solo i messaggi informativi.

È possibile combinare i valori per includere diversi tipi di messaggio. Per esempio, il valore **3** carica ad esempio i messaggi di errore (**1**) e i messaggi di avviso (**2**). Il valore **7** carica i messaggi di errore (**1**), i messaggi di avviso (**2**) e i messaggi informativi (**4**).

Per inviare i dati del registro di sicurezza a CloudWatch Logs

1. Nel file JSON, individuare la sezione `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Per `Levels`, digita **7** per caricare tutti i messaggi.

Per inviare i dati del registro degli eventi di sistema a CloudWatch Logs

1. Nel file JSON, individuare la sezione `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. Per `Levels`, specifica il tipo di messaggio da caricare. È possibile specificare uno dei seguenti valori:

- **1**: vengono caricati solo i messaggi di errore.

- **2**: vengono caricati solo i messaggi di avviso.
- **4**: vengono caricati solo i messaggi informativi.

È possibile combinare i valori per includere diversi tipi di messaggio. Per esempio, il valore **3** carica ad esempio i messaggi di errore (**1**) e i messaggi di avviso (**2**). Il valore **7** carica i messaggi di errore (**1**), i messaggi di avviso (**2**) e i messaggi informativi (**4**).

Per inviare altri tipi di dati del registro degli eventi a CloudWatch Logs

1. Nel file JSON, aggiungere una nuova sezione. Ogni sezione deve avere un `Id` univoco.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Per `Id` digitare un nome per il log da caricare (ad esempio, **`WindowsBackup`**).
3. Per `LogName`, digita il nome del log da caricare. È possibile trovare il nome del log nel modo seguente.

   a. Aprire il visualizzatore di eventi.

   b. Nel riquadro di navigazione, selezionare Applications and Services Logs (Log di applicazioni e servizi).

   c. Andare al log, quindi scegliere Actions (Operazioni) e Properties (Proprietà).
4. Per `Levels`, specifica il tipo di messaggio da caricare. È possibile specificare uno dei seguenti valori:

   - **1**: vengono caricati solo i messaggi di errore.
   - **2**: vengono caricati solo i messaggi di avviso.
   - **4**: vengono caricati solo i messaggi informativi.

È possibile combinare i valori per includere diversi tipi di messaggio. Per esempio, il valore **3** carica ad esempio i messaggi di errore (**1**) e i messaggi di avviso (**2**). Il valore **7** carica i messaggi di errore (**1**), i messaggi di avviso (**2**) e i messaggi informativi (**4**).

Per inviare i dati di Event Tracing for Windows ai registri CloudWatch

ETW (traccia eventi per Windows) fornisce meccanismi di log efficienti e dettagliati ai quali le applicazioni possono scrivere log. Ogni ETW è controllato da un gestore di sessione che può avviare e terminare la sessione di log. Ogni sessione dispone di un fornitore e di uno o più consumatori.

1. Nel file JSON, individuare la sezione `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Per `LogName`, digita il nome del log da caricare.
3. Per `Levels`, specifica il tipo di messaggio da caricare. È possibile specificare uno dei seguenti valori:

   - **1**: vengono caricati solo i messaggi di errore.
   - **2**: vengono caricati solo i messaggi di avviso.
   - **4**: vengono caricati solo i messaggi informativi.

   È possibile combinare i valori per includere diversi tipi di messaggio. Per esempio, il valore **3** carica ad esempio i messaggi di errore (**1**) e i messaggi di avviso (**2**). Il valore **7** carica i messaggi di errore (**1**), i messaggi di avviso (**2**) e i messaggi informativi (**4**).

Per inviare registri personalizzati (qualsiasi file di registro basato su testo) a Logs CloudWatch

1. Nel file JSON, individuare la sezione `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Per `LogDirectoryPath`, digita il percorso in cui sono memorizzati i log nell'istanza.
3. In `TimestampFormat`, digita il formato timestamp da utilizzare. Per ulteriori informazioni sui valori supportati, consulta l'argomento relativo alle stringhe di formato di data e ora personalizzate su MSDN.

> **Important**
>
> Il file di log di origine deve contenere il timestamp all'inizio di ogni riga di log e uno spazio dopo il timestamp.

4. in `Encoding`, digita la codifica del file da utilizzare (ad esempio, UTF-8). Per un elenco dei valori supportati, consulta l'argomento relativo alla classe di codifica su MSDN.

> **Note**
>
> Utilizza il nome di codifica, non il nome di visualizzazione.

5. (Facoltativo) Per `Filter`, digitare il prefisso dei nomi di log. Lascia questo parametro bianco per monitorare tutti i file. Per ulteriori informazioni sui valori supportati, vedere l'argomento FileSystemWatcherFilter Proprietà su MSDN.
6. In `CultureName`, digita le impostazioni locali in cui viene registrato il timestamp (facoltativo). Se `CultureName` è vuoto, si utilizza per impostazione predefinita la stessa configurazione locale utilizzata dall'istanza di Windows. Per ulteriori informazioni relative a questo argomento, consulta la colonna `Language tag` nella tabella dell'argomento comportamento del prodotto su MSDN.

> **Note**
>
> I valori `div`, `div-MV`, `hu` e `hu-HU` non sono supportati.

7. (Facoltativo) Per `TimeZoneKind`, digitare `Local` o `UTC`. Puoi impostare questo valore per fornire delle informazioni sul fuso orario se non sono incluse nel timestamp del log. Se questo parametro viene lasciato vuoto e se il timestamp non include informazioni sul fuso orario, per impostazione predefinita CloudWatch Logs utilizza il fuso orario locale. Questo parametro viene ignorato se il timestamp contiene già informazioni sul fuso orario.
8. (Facoltativo) Per `LineCount`, digitare il numero di righe dell'intestazione per identificare il file di log. Ad esempio, i file di log IIS dispongono di intestazioni praticamente identiche. È possibile immettere **5** per leggere le prime tre righe dell'intestazione del file di log per identificarlo. Nei file di log IIS, la terza riga è costituita dalla data e dal timestamp, ma non è sempre garantito che il timestamp sia diverso tra i diversi file di log. Per questo motivo, è consigliabile includere almeno una riga di dati di log reali per identificare in modo univoco il file di log.

Per inviare i dati di registro IIS a Logs CloudWatch

1. Nel file JSON, individuare la sezione `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Per `LogDirectoryPath`, digita la cartella in cui sono memorizzati i log IIS per un sito individuale (ad esempio, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

> **Note**
>
> È supportato solo il formato di log W3C. I formati IIS, NCSA e quelli personalizzati non sono supportati.

3. In `TimestampFormat`, digita il formato timestamp da utilizzare. Per ulteriori informazioni sui valori supportati, consulta l'argomento relativo alle stringhe di formato di data e ora personalizzate su MSDN.
4. in `Encoding`, digita la codifica del file da utilizzare (ad esempio, UTF-8). Per ulteriori informazioni sui valori supportati, consulta l'argomento relativo alla classe di codifica su MSDN.

> **Note**
>
> Utilizza il nome di codifica, non il nome di visualizzazione.

5. (Facoltativo) Per `Filter`, digitare il prefisso dei nomi di log. Lascia questo parametro bianco per monitorare tutti i file. Per ulteriori informazioni sui valori supportati, vedere l'argomento FileSystemWatcherFilter Proprietà su MSDN.
6. In `CultureName`, digita le impostazioni locali in cui viene registrato il timestamp (facoltativo). Se `CultureName` è vuoto, si utilizza per impostazione predefinita la stessa configurazione locale utilizzata dall'istanza di Windows. Per ulteriori informazioni relative ai valori supportati, consulta la colonna `Language tag` nella tabella dell'argomento comportamento del prodotto su MSDN.

> **Note**
>
> I valori `div`, `div-MV`, `hu` e `hu-HU` non sono supportati.

7. In `TimeZoneKind`, inserisci `Local` o `UTC` (facoltativo). Puoi impostare questo valore per fornire delle informazioni sul fuso orario se non sono incluse nel timestamp del log. Se questo parametro viene lasciato vuoto e se il timestamp non include informazioni sul fuso orario, per impostazione predefinita CloudWatch Logs utilizza il fuso orario locale. Questo parametro viene ignorato se il timestamp contiene già informazioni sul fuso orario.
8. (Facoltativo) Per `LineCount`, digitare il numero di righe dell'intestazione per identificare il file di log. Ad esempio, i file di log IIS dispongono di intestazioni praticamente identiche. È possibile inserire **5** per leggere le prime cinque righe dell'intestazione del file di log per identificarlo. Nei file di log IIS, la terza riga è costituita dalla data e dal timestamp, ma non è sempre garantito

che il timestamp sia diverso tra i diversi file di log. Per questo motivo, ti consigliamo di includere almeno una riga di dati di log reali per identificare univocamente il file di log.

Fase 4: configurazione del controllo di flusso

Ogni tipo di dati deve avere una destinazione corrispondente nella sezione `Flows`. Ad esempio, per inviare il registro personalizzato, il registro ETW e il registro di sistema a CloudWatch Logs, aggiungete alla sezione. `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` `Flows`

> **Warning**
>
> Aggiungere una fase non valida blocca il flusso. Ad esempio, se aggiungi una fase relativa ai parametri del disco, ma la tua istanza non dispone di un disco, tutte le fasi del flusso verranno bloccate.

Puoi inviare lo stesso file di log a più destinazioni. Ad esempio, per inviare il log dell'applicazione a due destinazioni differenti, definite nella sezione `CloudWatchLogs`, aggiungi `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` alla sezione `Flows`.

Per configurare il controllo di flusso

1. Nel file `AWS.EC2.Windows.CloudWatch.json`, individuare la sezione `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. In `Flows`, aggiungere ciascun tipo di dati da caricare (ad esempio `ApplicationEventLog`) e la sua destinazione (ad esempio `CloudWatchLogs`).

Fase 5: salvataggio del contenuto JSON

Hai ora terminato la modifica del file JSON. Salvarlo e nella fase successiva incollare i contenuti del file in un editor di testo in un'altra finestra. I contenuti del file saranno necessari in un secondo momento nel corso di questa procedura.

Creare un ruolo IAM per Systems Manager

È necessario un ruolo IAM per ottenere le credenziali dell'istanza quando utilizzi Run Command per Systems Manager. Questo ruolo permette a Systems Manager di eseguire operazioni sull'istanza. Per ulteriori informazioni, consulta Configurazione dei ruoli di sicurezza per Systems Manager nella Guida per l'utente di AWS Systems Manager . Per informazioni su come collegare un ruolo IAM a un'istanza esistente, consulta Attaching an IAM Role to an Instance nella Amazon EC2 User Guide.

Verifica dei prerequisiti di Systems Manager

Prima di utilizzare Systems Manager Run Command per configurare l'integrazione con CloudWatch Logs, verificate che le istanze soddisfino i requisiti minimi. Per ulteriori informazioni, consulta Prerequisti di Systems Manager nella Guida per l'utente di AWS Systems Manager .

Verifica dell'accesso a Internet

Le tue istanze Amazon EC2 Windows Server e le istanze gestite devono disporre di accesso a Internet in uscita per poter inviare dati di log ed eventi a. CloudWatch Per ulteriori informazioni su come configurare l'accesso a Internet, consulta Gateway Internet nella Guida per l'utente di Amazon VPC.

Abilita CloudWatch i log utilizzando il comando Systems Manager Run

Run Command ti permette di gestire la configurazione delle istanze on demand. Puoi specificare un documento Systems Manager, specificare i parametri ed eseguire il comando in una o più istanze. Il SSM Agent sull'istanza elabora il comando e configura l'istanza come specificato.

Per configurare l'integrazione con CloudWatch Logs utilizzando Run Command

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Apri la console SSM all'indirizzo. https://console.aws.amazon.com/systems-manager/
3. Nel riquadro di navigazione seleziona Esegui comando.
4. Scegli Esegui comando.

5. Per il documento Command, scegli AWS- ConfigureCloudWatch.
6. Per le istanze Target, scegli le istanze da integrare con CloudWatch Logs. Se non viene visualizzata un'istanza nell'elenco, potrebbe non essere configurata per Run Command. Per ulteriori informazioni, consulta Systems Manager Prerequisites nella Amazon EC2 User Guide.
7. In Status (Stato), scegliere Enabled (Abilitato).
8. In Properties (Proprietà) copiare e incollare il contenuto JSON creato nelle attività precedenti.
9. Completare i restanti campi opzionali e scegliere Esegui.

Utilizza la procedura seguente per visualizzare i risultati dell'esecuzione del comando nella console Amazon EC2.

Per visualizzare l'output del comando nella console

1. Selezionare un comando.
2. Selezionare la scheda Output.
3. Scegliere View Output (Visualizza output). La pagina di output del comando mostra i risultati di esecuzione del comando.

Avvio rapido: consenti alle istanze Amazon EC2 che eseguono Windows Server 2012 e Windows Server 2008 di inviare log a Logs CloudWatch

> **Tip**
>
> CloudWatch include un nuovo agente unificato in grado di raccogliere sia i log che le metriche dalle istanze EC2 e dai server locali. Ti consigliamo di utilizzare il nuovo agente CloudWatch unificato. Per ulteriori informazioni, consulta Guida introduttiva ai CloudWatch registri.
> Il resto di questa sezione spiega l'uso del vecchio agente Logs. CloudWatch

Consenti alle istanze Amazon EC2 che eseguono Windows Server 2012 e Windows Server 2008 di inviare log a Logs CloudWatch

Utilizza la procedura seguente per consentire alle istanze che eseguono Windows Server 2012 e Windows Server 2008 di inviare i log ai registri. CloudWatch

Download del file di configurazione di esempio

Download il seguente file JSON di esempio su computer: `AWS.EC2.Windows.CloudWatch.json`. Puoi modificarlo seguendo le fasi sotto riportate.

Configurare il file JSON per CloudWatch

Puoi determinare a quali log inviare i dati CloudWatch specificando le tue scelte nel file di configurazione JSON. Il processo di creazione di questo file e l'indicazione delle scelte possono richiedere 30 minuti o più per il completamento. Una volta completata questa attività una volta, puoi riutilizzare il file di configurazione su tutte le istanze.

Fasi

- Passaggio 1: abilitare i log CloudWatch
- Fase 2: Configurare le impostazioni per CloudWatch
- Fase 3: configurazione dei dati da inviare
- Fase 4: configurazione del controllo di flusso

Passaggio 1: abilitare i log CloudWatch

Nella parte superiore del file JSON, modifica "falso" in "vero" per `IsEnabled`:

```
"IsEnabled": true,
```

Fase 2: Configurare le impostazioni per CloudWatch

Specifica le credenziali, la Regione, un nome per il gruppo di log e uno spazio dei nomi del flusso di log. Ciò consente all'istanza di inviare i dati di registro ai CloudWatch registri. Per inviare gli stessi dati di registro a posizioni diverse, puoi aggiungere sezioni aggiuntive con sezioni uniche IDs (ad esempio, "CloudWatchLogs2" e CloudWatchLogs 3") e una regione diversa per ogni ID.

Per configurare le impostazioni per l'invio dei dati di registro ai registri CloudWatch

1. Nel file JSON, individuare la sezione `CloudWatchLogs`.

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
```

```
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. Lasciare vuoti i campi `AccessKey` e `SecretKey`. Puoi configurare le credenziali tramite un ruolo IAM.
3. In `Region` digitare la Regione alla quale inviare dati di log (ad esempio, `us-east-2`).
4. Per `LogGroup` digita il nome del gruppo di log. Questo nome viene visualizzato nella schermata Log Groups (Gruppi di log) nella console CloudWatch.
5. Per `LogStream` digita il flusso di log di destinazione. Questo nome viene visualizzato nella schermata Log Groups > Streams della CloudWatch console.

   Se si utilizza `{instance_id}`, il nome del flusso di log predefinito è l'ID istanza di tale istanza.

   Se specifichi un nome di log stream che non esiste già, CloudWatch Logs lo crea automaticamente per te. È possibile definire un nome per il flusso di log utilizzando una stringa letterale, le variabili predefinite `{instance_id}`, `{hostname}` e `{ip_address}` oppure una combinazione di queste.

Fase 3: configurazione dei dati da inviare

È possibile inviare i dati del registro degli eventi, i dati di Event Tracing for Windows (ETW) e altri dati di registro ai registri. CloudWatch

Per inviare i dati del registro degli eventi dell'applicazione Windows a Logs CloudWatch

1. Nel file JSON, individuare la sezione `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
```

```
},
```

2. Per `Levels`, specifica il tipo di messaggio da caricare. È possibile specificare uno dei seguenti valori:

   - **1**: vengono caricati solo i messaggi di errore.
   - **2**: vengono caricati solo i messaggi di avviso.
   - **4**: vengono caricati solo i messaggi informativi.

   È possibile combinare i valori per includere diversi tipi di messaggio. Per esempio, il valore **3** carica ad esempio i messaggi di errore (**1**) e i messaggi di avviso (**2**). Il valore **7** carica i messaggi di errore (**1**), i messaggi di avviso (**2**) e i messaggi informativi (**4**).

Per inviare i dati del registro di sicurezza a CloudWatch Logs

1. Nel file JSON, individuare la sezione `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Per `Levels`, digita **7** per caricare tutti i messaggi.

Per inviare i dati del registro degli eventi di sistema a CloudWatch Logs

1. Nel file JSON, individuare la sezione `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
```

```
        "Levels": "7"
    }
},
```

2. Per `Levels`, specifica il tipo di messaggio da caricare. È possibile specificare uno dei seguenti valori:

   - **1**: vengono caricati solo i messaggi di errore.
   - **2**: vengono caricati solo i messaggi di avviso.
   - **4**: vengono caricati solo i messaggi informativi.

   È possibile combinare i valori per includere diversi tipi di messaggio. Per esempio, il valore **3** carica ad esempio i messaggi di errore (**1**) e i messaggi di avviso (**2**). Il valore **7** carica i messaggi di errore (**1**), i messaggi di avviso (**2**) e i messaggi informativi (**4**).

Per inviare altri tipi di dati del registro degli eventi a CloudWatch Logs

1. Nel file JSON, aggiungere una nuova sezione. Ogni sezione deve avere un `Id` univoco.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Per `Id` digitare un nome per il log da caricare (ad esempio, **`WindowsBackup`**).
3. Per `LogName`, digita il nome del log da caricare. È possibile trovare il nome del log nel modo seguente.

   a. Aprire il visualizzatore di eventi.

   b. Nel riquadro di navigazione, selezionare Applications and Services Logs (Log di applicazioni e servizi).

   c. Andare al log, quindi scegliere Actions (Operazioni) e Properties (Proprietà).

4. Per `Levels`, specifica il tipo di messaggio da caricare. È possibile specificare uno dei seguenti valori:

   - **1**: vengono caricati solo i messaggi di errore.
   - **2**: vengono caricati solo i messaggi di avviso.
   - **4**: vengono caricati solo i messaggi informativi.

   È possibile combinare i valori per includere diversi tipi di messaggio. Per esempio, il valore **3** carica ad esempio i messaggi di errore (**1**) e i messaggi di avviso (**2**). Il valore **7** carica i messaggi di errore (**1**), i messaggi di avviso (**2**) e i messaggi informativi (**4**).

Per inviare i dati di Event Tracing for Windows ai registri CloudWatch

ETW (traccia eventi per Windows) fornisce meccanismi di log efficienti e dettagliati ai quali le applicazioni possono scrivere log. Ogni ETW è controllato da un gestore di sessione che può avviare e terminare la sessione di log. Ogni sessione dispone di un fornitore e di uno o più consumatori.

1. Nel file JSON, individuare la sezione `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Per `LogName`, digita il nome del log da caricare.
3. Per `Levels`, specifica il tipo di messaggio da caricare. È possibile specificare uno dei seguenti valori:

   - **1**: vengono caricati solo i messaggi di errore.
   - **2**: vengono caricati solo i messaggi di avviso.
   - **4**: vengono caricati solo i messaggi informativi.

È possibile combinare i valori per includere diversi tipi di messaggio. Per esempio, il valore **3** carica ad esempio i messaggi di errore (**1**) e i messaggi di avviso (**2**). Il valore **7** carica i messaggi di errore (**1**), i messaggi di avviso (**2**) e i messaggi informativi (**4**).

Per inviare registri personalizzati (qualsiasi file di registro basato su testo) a Logs CloudWatch

1. Nel file JSON, individuare la sezione `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Per `LogDirectoryPath`, digita il percorso in cui sono memorizzati i log nell'istanza.
3. In `TimestampFormat`, digita il formato timestamp da utilizzare. Per ulteriori informazioni sui valori supportati, consulta l'argomento relativo alle stringhe di formato di data e ora personalizzate su MSDN.

> **Important**
>
> Il file di log di origine deve contenere il timestamp all'inizio di ogni riga di log e uno spazio dopo il timestamp.

4. in `Encoding`, digita la codifica del file da utilizzare (ad esempio, UTF-8). Per ulteriori informazioni sui valori supportati, consulta l'argomento relativo alla classe di codifica su MSDN.

> **Note**
>
> Utilizza il nome di codifica, non il nome di visualizzazione.

5. (Facoltativo) Per `Filter`, digitare il prefisso dei nomi di log. Lascia questo parametro bianco per monitorare tutti i file. Per ulteriori informazioni sui valori supportati, vedere l'argomento FileSystemWatcherFilter Proprietà su MSDN.
6. In `CultureName`, digita le impostazioni locali in cui viene registrato il timestamp (facoltativo). Se `CultureName` è vuoto, si utilizza per impostazione predefinita la stessa configurazione locale utilizzata dall'istanza di Windows. Per ulteriori informazioni relative ai valori supportati, consulta la colonna `Language tag` nella tabella dell'argomento comportamento del prodotto su MSDN.

> **Note**
>
> I valori `div`, `div-MV`, `hu` e `hu-HU` non sono supportati.

7. (Facoltativo) Per `TimeZoneKind`, digitare `Local` o `UTC`. Puoi impostare questo valore per fornire delle informazioni sul fuso orario se non sono incluse nel timestamp del log. Se questo parametro viene lasciato vuoto e se il timestamp non include informazioni sul fuso orario, per impostazione predefinita CloudWatch Logs utilizza il fuso orario locale. Questo parametro viene ignorato se il timestamp contiene già informazioni sul fuso orario.
8. (Facoltativo) Per `LineCount`, digitare il numero di righe dell'intestazione per identificare il file di log. Ad esempio, i file di log IIS dispongono di intestazioni praticamente identiche. È possibile immettere **5** per leggere le prime tre righe dell'intestazione del file di log per identificarlo. Nei file di log IIS, la terza riga è costituita dalla data e dal timestamp, ma non è sempre garantito che il timestamp sia diverso tra i diversi file di log. Per questo motivo, è consigliabile includere almeno una riga di dati di log reali per identificare in modo univoco il file di log.

Per inviare i dati di registro IIS a Logs CloudWatch

1. Nel file JSON, individuare la sezione `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
```

```
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Per `LogDirectoryPath`, digita la cartella in cui sono memorizzati i log IIS per un sito individuale (ad esempio, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

> **Note**
>
> È supportato solo il formato di log W3C. I formati IIS, NCSA e quelli personalizzati non sono supportati.

3. In `TimestampFormat`, digita il formato timestamp da utilizzare. Per ulteriori informazioni sui valori supportati, consulta l'argomento relativo alle stringhe di formato di data e ora personalizzate su MSDN.
4. in `Encoding`, digita la codifica del file da utilizzare (ad esempio, UTF-8). Per ulteriori informazioni sui valori supportati, consulta l'argomento relativo alla classe di codifica su MSDN.

> **Note**
>
> Utilizza il nome di codifica, non il nome di visualizzazione.

5. (Facoltativo) Per `Filter`, digitare il prefisso dei nomi di log. Lascia questo parametro bianco per monitorare tutti i file. Per ulteriori informazioni sui valori supportati, vedere l'argomento FileSystemWatcherFilter Proprietà su MSDN.
6. In `CultureName`, digita le impostazioni locali in cui viene registrato il timestamp (facoltativo). Se `CultureName` è vuoto, si utilizza per impostazione predefinita la stessa configurazione locale utilizzata dall'istanza di Windows. Per ulteriori informazioni relative ai valori supportati, consulta la colonna `Language tag` nella tabella dell'argomento comportamento del prodotto su MSDN.

> **Note**
>
> I valori `div`, `div-MV`, `hu` e `hu-HU` non sono supportati.

7. In `TimeZoneKind`, inserisci `Local` o `UTC` (facoltativo). Puoi impostare questo valore per fornire delle informazioni sul fuso orario se non sono incluse nel timestamp del log. Se questo parametro viene lasciato vuoto e se il timestamp non include informazioni sul fuso orario, per impostazione predefinita CloudWatch Logs utilizza il fuso orario locale. Questo parametro viene ignorato se il timestamp contiene già informazioni sul fuso orario.
8. (Facoltativo) Per `LineCount`, digitare il numero di righe dell'intestazione per identificare il file di log. Ad esempio, i file di log IIS dispongono di intestazioni praticamente identiche. È possibile inserire **5** per leggere le prime cinque righe dell'intestazione del file di log per identificarlo. Nei file di log IIS, la terza riga è costituita dalla data e dal timestamp, ma non è sempre garantito che il timestamp sia diverso tra i diversi file di log. Per questo motivo, ti consigliamo di includere almeno una riga di dati di log reali per identificare univocamente il file di log.

Fase 4: configurazione del controllo di flusso

Ogni tipo di dati deve avere una destinazione corrispondente nella sezione `Flows`. Ad esempio, per inviare il registro personalizzato, il registro ETW e il registro di sistema a CloudWatch Logs, aggiungete alla sezione. `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` `Flows`

> **Warning**
>
> Aggiungere una fase non valida blocca il flusso. Ad esempio, se aggiungi una fase relativa ai parametri del disco, ma la tua istanza non dispone di un disco, tutte le fasi del flusso verranno bloccate.

Puoi inviare lo stesso file di log a più destinazioni. Ad esempio, per inviare il log dell'applicazione a due destinazioni differenti, definite nella sezione `CloudWatchLogs`, aggiungi `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` alla sezione `Flows`.

Per configurare il controllo di flusso

1. Nel file `AWS.EC2.Windows.CloudWatch.json`, individuare la sezione `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. In `Flows`, aggiungere ciascun tipo di dati da caricare (ad esempio `ApplicationEventLog`) e la sua destinazione (ad esempio `CloudWatchLogs`).

Hai ora terminato la modifica del file JSON. Lo utilizzerai in una fase successiva.

Avvia l'agente

Per consentire a un'istanza Amazon EC2 che esegue Windows Server 2012 o Windows Server 2008 di inviare log a Logs, utilizza il CloudWatch servizio EC2 Config (. `EC2Config.exe`) L'istanza deve avere EC2 Config 4.0 o versione successiva ed è possibile utilizzare questa procedura.

Per configurare CloudWatch utilizzando EC2 Config 4.x

1. Controllare la codifica del file `AWS.EC2.Windows.CloudWatch.json` precedentemente modificato in questa procedura. È supportata solo la codifica UTF-8 senza BOM. Salvare quindi il file nella cartella seguente nell'istanza R2 di Windows Server 2008 - 2012: `C:\Program Files \Amazon\SSM\Plugins\awsCloudWatch\`.
2. Avvia o riavvia l'agente SSM (`AmazonSSMAgent.exe`) utilizzando il pannello di controllo dei servizi di Windows o utilizzando il seguente comando: PowerShell

```
PS C:\> Restart-Service AmazonSSMAgent
```

Dopo il riavvio, l'agente SSM rileva il file di configurazione e configura l'istanza per l'integrazione. CloudWatch Se modifichi i parametri e le impostazioni nel file di configurazione locale, sarà necessario riavviare il SSM Agent per trovare le modifiche. Per disabilitare CloudWatch l'integrazione sull'istanza, apporta `false` e salva `IsEnabled` le modifiche nel file di configurazione.

Segnala lo stato dell'agente CloudWatch Logs

Utilizza la seguente procedura per segnalare lo stato dell'agente CloudWatch Logs sulla tua istanza EC2.

Indicare lo stato dell'agente

1. Connettiti all'istanza EC2. Per ulteriori informazioni, consulta Connect to Your Instance nella Amazon EC2 User Guide.

   Per ulteriori informazioni sui problemi di connessione, consulta Troubleshooting Connecting to Your Instance nella Amazon EC2 User Guide
2. Al prompt dei comandi, digita il comando seguente:

   ```
   sudo service awslogs status
   ```

   Se esegui Amazon Linux 2, digita il comando seguente:

   ```
   sudo service awslogsd status
   ```
3. Controlla il file/var/log/awslogs.log per eventuali errori, avvisi o problemi con l' CloudWatch agente Logs.

Avvia l'agente Logs CloudWatch

Se l'agente CloudWatch Logs sulla tua istanza EC2 non si è avviato automaticamente dopo l'installazione, o se hai interrotto l'agente, puoi utilizzare la seguente procedura per avviare l'agente.

Avvio dell'agente

1. Connettiti all'istanza EC2. Per ulteriori informazioni, consulta Connect to Your Instance nella Amazon EC2 User Guide.

   Per ulteriori informazioni sui problemi di connessione, consulta Troubleshooting Connecting to Your Instance nella Amazon EC2 User Guide.
2. Al prompt dei comandi, digita il comando seguente:

   ```
   sudo service awslogs start
   ```

Se esegui Amazon Linux 2, digita il comando seguente:

```
sudo service awslogsd start
```

Arresta l'agente CloudWatch Logs

Utilizza la seguente procedura per arrestare l'agente CloudWatch Logs sulla tua istanza EC2.

Arresto dell'agente

1. Connettiti all'istanza EC2. Per ulteriori informazioni, consulta Connect to Your Instance nella Amazon EC2 User Guide.

   Per ulteriori informazioni sui problemi di connessione, consulta Troubleshooting Connecting to Your Instance nella Amazon EC2 User Guide.
2. Al prompt dei comandi, digita il comando seguente:

   ```
   sudo service awslogs stop
   ```

   Se esegui Amazon Linux 2, digita il comando seguente:

   ```
   sudo service awslogsd stop
   ```

CloudWatch Registra il riferimento dell'agente

Important

Questa sezione è un riferimento per coloro che utilizzano l'agente Logs obsoleto CloudWatch . Se utilizzi Instance Metadata Service Version 2 (IMDSv2), devi usare il nuovo agente unificato. CloudWatch Tuttavia, anche se non utilizzi IMDSv2, ti consigliamo vivamente di utilizzare il nuovo agente unificato anziché l'agente Logs obsoleto. CloudWatch CloudWatch Per informazioni sul nuovo CloudWatch agente unificato, consulta Raccolta di metriche e log dall'istanza Amazon EC2 e dai server locali con l'agente. CloudWatch Per informazioni sulla migrazione dall'agente CloudWatch Logs obsoleto all'agente unificato, crea il file di configurazione dell'agente con la procedura guidata. CloudWatch

L'agente CloudWatch Logs fornisce un modo automatizzato per inviare dati di log a CloudWatch Logs da istanze Amazon EC2. L'agente include i componenti seguenti:

- Un plug-in AWS CLI che invia i dati di registro a Logs. CloudWatch
- Uno script (demone) che avvia il processo per inviare i dati ai registri. CloudWatch
- Un processo cron che garantisce che il daemon sia sempre in esecuzione.

File di configurazione dell'agente

Il file di configurazione dell'agente CloudWatch Logs descrive le informazioni necessarie all'agente Logs. CloudWatch La sezione [general] del file di configurazione dell'agente definisce le configurazioni comuni applicabili a tutti i flussi di log. La sezione [logstream] definisce le informazioni necessarie per l'invio di un file locale a un flusso di log in remoto. Puoi disporre di più sezioni [logstream], ma ciascuna deve avere un nome univoco all'interno del file di configurazione, ad esempio [logstream1], [logstream2] e così via. Il valore [logstream] e la prima riga di dati nel file di log definiscono l'identità del file di log.

```
[general]
state_file = value
logging_config_file = value
use_gzip_http_content_encoding = [true | false]

[logstream1]
log_group_name = value
log_stream_name = value
datetime_format = value
time_zone = [LOCAL|UTC]
file = value
file_fingerprint_lines = integer | integer-integer
multi_line_start_pattern = regex | {datetime_format}
initial_position = [start_of_file | end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = integer
batch_count = integer
batch_size = integer

[logstream2]
...
```

state_file

Specifica dove è archiviato il file di stato.

logging_config_file

(Opzionale) Specifica la posizione del file di configurazione di registrazione dell'agente. Se non specifichi in questa pagina un file di configurazione di registrazione dell'agente, verrà utilizzato il file predefinito awslogs.conf. La posizione del file di default è `/var/awslogs/etc/awslogs.conf` se hai installato l'agente con uno script oppure `/etc/awslogs/awslogs.conf` se hai installato l'agente con rpm. Il file è in formato file di configurazione Python (https://docs.python.org/2/library/logging.config.html#logging -config-fileformat). I logger con i nomi seguenti possono essere personalizzati.

```
cwlogs.push
cwlogs.push.reader
cwlogs.push.publisher
cwlogs.push.event
cwlogs.push.batch
cwlogs.push.stream
cwlogs.push.watcher
```

L'esempio seguente modifica il livello di lettore ed editore a WARNING, mentre il valore predefinito è INFO.

```
[loggers]
keys=root,cwlogs,reader,publisher

[handlers]
keys=consoleHandler

[formatters]
keys=simpleFormatter

[logger_root]
level=INFO
handlers=consoleHandler

[logger_cwlogs]
level=INFO
handlers=consoleHandler
qualname=cwlogs.push
```

```
propagate=0

[logger_reader]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.reader
propagate=0

[logger_publisher]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.publisher
propagate=0

[handler_consoleHandler]
class=logging.StreamHandler
level=INFO
formatter=simpleFormatter
args=(sys.stderr,)

[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(process)d - %(threadName)s -
 %(message)s
```

use_gzip_http_content_encoding

Se impostato su true (impostazione predefinita), abilita la codifica del contenuto http gzip per inviare payload compressi ai registri. CloudWatch Ciò riduce l'utilizzo della CPU, riduce e diminuisce la latenza di put NetworkOut. Per disabilitare questa funzionalità, aggiungi use_gzip_http_content_encoding = false alla sezione [general] del file di configurazione dell'agente Logs, quindi riavvia l'agente. CloudWatch

> **Note**
>
> Questa impostazione è disponibile solo in awscli-cwlogs versione 1.3.3 e successive.

log_group_name

Specifica il gruppo di log di destinazione. Un gruppo di log viene creato automaticamente se non è già esistente. I nomi dei gruppi di log possono essere lunghi da 1 a 512 caratteri. I caratteri consentiti includono a-z, 0-9 A-Z, '_' (trattino basso), '-' (trattino), '/' (barra) e '.' (periodo).

log_stream_name

Specifica il flusso di log di destinazione. Per definire un nome per il flusso di log, puoi utilizzare una stringa letterale o variabili predefinite ({instance_id}, {hostname} e {ip_address}) oppure una combinazione di entrambe. Un flusso di log viene creato automaticamente se non è già esistente.

datetime_format

Specifica il modo in cui il timestamp viene estratto dai log. Il timestamp viene utilizzato per recuperare eventi di log e generare di parametri. L'ora corrente viene utilizzata per ciascun evento di log se il valore di datetime_format non è fornito. Se il valore di datetime_format fornito non è valido per un determinato messaggio di log, verrà utilizzato il timestamp dell'ultimo evento di log con timestamp analizzato correttamente. Se non esistono eventi di log precedenti, viene utilizzata l'ora corrente.

I codici datetime_format comuni sono elencati di seguito. Puoi inoltre utilizzare qualsiasi codice datetime_format supportato da Python, datetime.strptime(). L'offset del fuso orario (%z) è supportato sebbene non sia supportato fino a Python 3.2, [+-]HHMM senza i due punti (:). Per ulteriori informazioni, consulta la pagina relativa al comportamento di strftime() e strptime().

%y: anno senza secolo come numero decimale a cui è aggiunto uno zero. 00, 01, ..., 99

%Y: anno con secolo come numero decimale. 1970, 1988, 2001, 2013

%b: mese come nome abbreviato nella lingua locale. Jan, Feb, ..., Dec (en_US);

c%B: mese come nome completo nella lingua locale. January, February, ..., December (en_US);

%m: mese come numero decimale a cui è aggiunto uno zero. 01, 02, ..., 12

%d: giorno del mese come numero decimale a cui è aggiunto uno zero. 01, 02, ..., 31

%H: ora (formato di 24 ore) come numero decimale a cui è aggiunto uno zero. 00, 01, ..., 23

%I: ora (formato di 12 ore) come numero decimale a cui è aggiunto uno zero. 01, 02, ..., 12

%p: equivalente locale di AM o PM.

%M: minuto come numero decimale a cui è aggiunto uno zero. 00, 01, ..., 59

%S: secondo come numero decimale a cui è aggiunto uno zero. 00, 01, ..., 59

%f: microsecondo come numero decimale, a cui è aggiunto uno zero a sinistra. 000000, ..., 999999

%z: differenza dall'ora UTC nel formato +HHMM o -HHMM. +0000, -0400, +1030

Formati di esempio:

```
Syslog: '%b %d %H:%M:%S', e.g. Jan 23 20:59:29

Log4j: '%d %b %Y %H:%M:%S', e.g. 24 Jan 2014 05:00:00

ISO8601: '%Y-%m-%dT%H:%M:%S%z', e.g. 2014-02-20T05:20:20+0000
```

time_zone

Specifica il fuso orario del timestamp dell'evento di log. I due valori supportati sono UTC e LOCAL. Il valore predefinito è LOCAL, utilizzato nel caso in cui il fuso orario non può essere dedotto sulla base di datetime_format.

file

Specificare i file di registro che si desidera inviare ai CloudWatch registri. Il file può puntare a uno o più file specifici (utilizzando caratteri jolly come/var/log/system.log *). Solo il file più recente viene inviato ai CloudWatch registri in base all'ora di modifica del file. Ti consigliamo di utilizzare i caratteri jolly per specificare una serie di file dello stesso tipo, ad esempio ccess_log.2014-06-01-01, access_log.2014-06-01-02 e così via, ma non file di più tipi, ad esempio access_log_80 e access_log_443. Per specificare più tipi di file, aggiungi un'altra voce di flusso di log al file di configurazione in modo che ogni tipo di file di log vada in un flusso di log distinto. I file compressi non sono supportati.

file_fingerprint_lines

Specifica l'intervallo di righe per identificare un file. I valori validi sono un numero o due numeri delimitati da trattino, ad esempio "1", "2-5". Il valore predefinito è "1" in modo da utilizzare la prima riga per calcolare l'impronta. Le righe di impronte digitali non vengono inviate ai CloudWatch registri a meno che tutte le righe specificate non siano disponibili.

multi_line_start_pattern

Specifica il modello per identificare l'inizio di un messaggio di log. Un messaggio di log è composto da una riga corrispondente al modello e da tutte le righe successive non corrispondenti al modello. I valori validi sono espressioni regolari o {datetime_format}. Quando utilizzi

{datetime_format}, devi specificare l'opzione datetime_format. Il valore predefinito è "^[^\s]", in modo tale che le righe che iniziano con caratteri diversi da spazi vuoti chiudono il messaggio di log precedente e iniziano un nuovo messaggio di log.

initial_position

Specifica il punto da cui iniziare a leggere i dati (start_of_file o end_of_file). Il valore predefinito è start_of_file. È utilizzato solo se non vi è uno stato reso persistente per tale flusso di log.

encoding

Specifica la codifica del file di log in modo che il file può essere letto correttamente. Il valore predefinito è utf_8. Possono qui essere utilizzate le codifiche supportate da Python codecs.decode().

> **Warning**
>
> La specificazione di una codifica non corretta potrebbe causare una perdita di dati, in quanto i caratteri che non possono essere decodificati saranno sostituiti da altri caratteri.

Sono elencate di seguito alcune codifiche comuni:

```
ascii, big5, big5hkscs, cp037, cp424, cp437, cp500, cp720, cp737,
cp775, cp850, cp852, cp855, cp856, cp857, cp858, cp860, cp861, cp862,
cp863, cp864, cp865, cp866, cp869, cp874, cp875, cp932, cp949, cp950,
cp1006, cp1026, cp1140, cp1250, cp1251, cp1252, cp1253, cp1254, cp1255,
cp1256, cp1257, cp1258, euc_jp, euc_jis_2004, euc_jisx0213, euc_kr,
gb2312, gbk, gb18030, hz, iso2022_jp, iso2022_jp_1, iso2022_jp_2,
iso2022_jp_2004, iso2022_jp_3, iso2022_jp_ext, iso2022_kr, latin_1,
iso8859_2, iso8859_3, iso8859_4, iso8859_5, iso8859_6, iso8859_7,
iso8859_8, iso8859_9, iso8859_10, iso8859_13, iso8859_14, iso8859_15,
iso8859_16, johab, koi8_r, koi8_u, mac_cyrillic, mac_greek, mac_iceland,
mac_latin2, mac_roman, mac_turkish, ptcp154, shift_jis, shift_jis_2004,
shift_jisx0213, utf_32, utf_32_be, utf_32_le, utf_16, utf_16_be,
utf_16_le, utf_7, utf_8, utf_8_sig
```

buffer_duration

Specifica la durata del raggruppamento di eventi di log. Il valore minimo è 5.000 ms e il valore predefinito è 5.000 ms.

batch_count

Specifica il numero massimo di eventi di log in un batch, fino a un massimo di 10.000. Il valore predefinito è 10000.

batch_size

Specifica la dimensione massima di eventi di log in un batch in byte, fino a un massimo di 1.048.576 byte. Il valore predefinito è 1048576 byte. Questa dimensione viene calcolata come la somma di tutti i messaggi di evento inseriti UTF-8, più 26 byte per ogni evento di registro.

Utilizzo dell'agente CloudWatch Logs con proxy HTTP

È possibile utilizzare l'agente CloudWatch Logs con proxy HTTP.

> **Note**
>
> I proxy HTTP sono supportati in awslogs-agent-setup.py versione 1.3.8 o successive.

Per utilizzare l'agente CloudWatch Logs con proxy HTTP

1. Esegui una delle seguenti operazioni:

   a. Per una nuova installazione dell'agente CloudWatch Logs, esegui i seguenti comandi:

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
   ```

   ```
   sudo python awslogs-agent-setup.py --region us-east-1 --http-proxy http://your/proxy --https-proxy http://your/proxy --no-proxy 169.254.169.254
   ```

   Per mantenere l'accesso al servizio di metadati Amazon EC2 in istanze EC2, utilizza --no-proxy 169.254.169.254 (consigliato). Per ulteriori informazioni, consulta Metadati e dati utente dell'istanza nella Guida per l'utente di Amazon EC2.

   Nei valori per `http-proxy` e `https-proxy`, è necessario specificare l'URL nella sua interezza.

   b. Per un'installazione esistente dell'agente CloudWatch Logs, modifica/var/awslogs/etc/proxy.conf e aggiungi i tuoi proxy:

```
HTTP_PROXY=
HTTPS_PROXY=
NO_PROXY=
```

2. Riavvia l'agente per rendere effettive le modifiche:

```
sudo service awslogs restart
```

Se utilizzi Amazon Linux 2, utilizza il comando seguente per riavviare l'agente:

```
sudo service awslogsd restart
```

Compartimentazione dei file di configurazione dell'agente Logs CloudWatch

Se utilizzi awslogs-agent-setup.py versione 1.3.8 o successiva con awscli-cwlogs 1.3.3 o successiva, puoi importare diverse configurazioni di flusso per vari componenti indipendentemente l'una dall'altra creando file di configurazione aggiuntivi nella directory///. var/awslogs etc/config All'avvio, l'agente CloudWatch Logs include tutte le configurazioni di flusso in questi file di configurazione aggiuntivi. Le proprietà di configurazione nella sezione [generale] devono essere definite nel file di configurazione principale (/var/awslogs/etc/awslogs.conf) e vengono ignorate in tutti i file di configurazione aggiuntivi presenti in//. var/awslogs etc/config

Se non si dispone di una directory//var/awslogsetc/config/perché l'agente è stato installato con rpm, è possibile utilizzare invece la directory//config/. etc/awslogs

Riavvia l'agente per rendere effettive le modifiche:

```
sudo service awslogs restart
```

Se utilizzi Amazon Linux 2, utilizza il comando seguente per riavviare l'agente:

```
sudo service awslogsd restart
```

CloudWatch Domande frequenti sull'agente Logs

Quali tipi di rotazione di file sono supportati?

Sono supportati i seguenti meccanismi di rotazione di file:

- Ridenominazione di file di log esistenti con un suffisso numerico, quindi nuova creazione del file di log vuoto originale. Ad esempio,/var/log/syslog.log viene rinominato//syslog.log.1. var/log Se/var/log/syslog.log.1 esiste già da una rotazione precedente, viene rinominato//syslog.log.2. var/log
- Troncamento del file di log originale in vigore dopo la creazione di una copia. Ad esempio,//syslog.log viene copiato in//syslog.log.1 e//syslog.log viene troncato. var/log var/log var/log In questo caso potrebbe verificarsi una perdita di dati, quindi fai attenzione con l'uso di questo meccanismo di rotazione di file.
- Creazione di un nuovo file con un modello comune come il precedente. Ad esempio,//syslog.log.2014-01-01 rimane e//syslog.log.2014-01-02 viene creato. var/log var/log

L'impronta (ID origine) del file viene calcolata sottoponendo a hashing la chiave di flusso di log e la prima riga del contenuto del file. Per sovrascrivere questo comportamento, puoi utilizzare l'opzione file_fingerprint_lines. Quando si verifica la rotazione del file, il nuovo file deve avere un nuovo contenuto e il vecchio file non deve avere contenuto aggiuntivo; l'agente invia il nuovo file dopo aver completato la lettura del vecchio file.

Come posso determinare quale versione di agente sto utilizzando?

Se è stato utilizzato uno script di installazione per installare l'agente Logs, è possibile utilizzare//-version.sh per verificare quale versione dell'agente si sta utilizzando. CloudWatch var/awslogs bin/awslogs Verrà stampata la versione dell'agente e le sue dipendenze principali. Se hai usato yum per installare l'agente CloudWatch Logs, puoi usare «yum info awslogs» e «yum info aws-cli-plugin-cloudwatch-logs» per verificare la versione dell'agente e del plugin Logs. CloudWatch

Come vengono convertite le voci di log in eventi di log?

Gli eventi di log contengono due proprietà: il timestamp del momento in cui si è verificato l'evento e il messaggio di log non elaborato. Per impostazione predefinita, le righe che iniziano con caratteri diversi da spazi vuoti chiudono il messaggio di log precedente se ne esiste uno e iniziano un nuovo messaggio di log. Per sovrascrivere questo comportamento, puoi utilizzare multi_line_start_pattern e le righe che corrispondono al modello iniziano un nuovo messaggio di log. Il modello può essere qualsiasi espressione regolare o "{datetime_format}". Ad esempio, se la prima riga di ogni messaggio di registro contiene un timestamp come '2014-01-02 T13:13:01Z ', allora il multi_line_start_pattern può essere impostato su'\ d {4} -\ d {2} -\ d {2} T\ d {2}:\ d {2}:\ d {2} Z'. Per semplificare la configurazione, puoi utilizzare la variabile "{datetime_format}" se hai specificato l'opzione datetime_format. Per lo stesso esempio, se datetime_format è impostato su "%Y-%m-%dT%H:%M:%S%z", multi_line_start_pattern può essere semplicemente "{datetime_format}".

L'ora corrente viene utilizzata per ciascun evento di log se il valore di datetime_format non è fornito. Se il valore di datetime_format fornito non è valido per un determinato messaggio di log, verrà utilizzato il timestamp dell'ultimo evento di log con timestamp analizzato correttamente. Se non esistono eventi di log precedenti, viene utilizzata l'ora corrente. Viene registrato un messaggio di avviso quando un evento di log utilizza l'ora corrente o l'ora dell'evento di log precedente.

I timestamp vengono utilizzati per recuperare eventi di log e generare parametri, perciò se specifichi il formato errato, gli eventi di log potrebbero diventare non recuperabili e generare i parametri errati.

Come vengono raggruppatigli eventi di log?

Un batch diviene completo e viene pubblicato quando si verifica qualsiasi delle condizioni seguenti:

1. La quantità di tempo di buffer_duration è trascorsa a partire dall'aggiunta del primo evento di log.
2. È stato accumulato un valore inferiore di batch_size log di eventi di log, ma l'aggiunta del nuovo evento di log supera il valore di batch_size.
3. Il numero di eventi di log ha raggiunto il valore di batch_count.
4. Gli eventi di log dal batch non si estendono per più di 24 ore, ma l'aggiunta del nuovo evento di log supera il vincolo di 24 ore.

Cosa può causare l'omissione o il troncamento di voci di log, eventi di log o batch?

Per seguire il vincolo dell'operazione `PutLogEvents`, i seguenti problemi potrebbero causare l'omissione di un evento di log o di un batch.

> **Note**
>
> L'agente Logs scrive un avviso nel suo registro quando i dati vengono ignorati. CloudWatch

1. Se la dimensione di un evento di log supera 256 KB, l'evento di log viene completamente ignorato.
2. Se il timestamp dell'evento di log è superiore a 2 ore successive all'ora attuale, l'evento di log viene ignorato.
3. Se il timestamp dell'evento di log è superiore a 14 giorni antecedenti il giorno attuale, l'evento di log viene ignorato.

4. Se qualsiasi evento di log è precedente al periodo di conservazione del gruppo di log, l'intero batch viene ignorato.
5. Se il batch di eventi di log in una singola richiesta `PutLogEvents` si estende per più di 24 ore, l'operazione `PutLogEvents` ha esito negativo.

L'interruzione dell'agente causa dati? loss/duplicates

No, purché il file di stato sia disponibile e non si sia verificata alcuna rotazione del file a partire dall'ultima esecuzione. L'agente CloudWatch Logs può iniziare dal punto in cui si è fermato e continuare a inviare i dati di registro.

Posso indirizzare diversi file di log dallo stesso host o da host diversi allo stesso flusso di log?

La configurazione di più origini di log per l'invio di dati a un unico flusso di log non è supportata.

Quali chiamate API effettua l'agente (o quali operazioni devo aggiungere alla mia policy IAM)?

L'agente CloudWatch Logs richiede l'autorizzazione per eseguire`CreateLogGroup`, `CreateLogStream` `DescribeLogStreamsDescribeLogGroups`, `PutLogEvents` e azioni. `PutRetentionPolicy` Se utilizzi l'ultimo agente, `DescribeLogStreams` non è necessario. Consulta la policy IAM di esempio riportata di seguito.

JSON

```
{
"Version":"2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams",
      "logs:DescribeLogGroups",
      "logs:PutRetentionPolicy"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Non voglio che l'agente CloudWatch Logs crei automaticamente gruppi o flussi di log. Come posso impedire all'agente di ricreare gruppi di log e flussi di log?

Nella tua policy IAM, puoi limitare l'agente alle sole operazioni seguenti: `DescribeLogStreams`, `PutLogEvents`.

Prima di revocare le autorizzazioni `CreateLogStream` e `CreateLogGroup` all'agente, assicurati di creare sia i gruppi di log che i flussi di log che devono essere utilizzati dall'agente. L'agente di log non può creare flussi di log in un gruppo di log creato a meno che non disponga delle autorizzazioni `CreateLogStream` e `CreateLogGroup`.

Quali log dovrei esaminare durante la risoluzione dei problemi?

Il log di installazione dell'agente si trova in `/var/log/awslogs-agent-setup.log` e il log dell'agente si trova in `/var/log/awslogs.log`.

Avvio rapido: CloudFormation da utilizzare per iniziare a utilizzare i log CloudWatch

AWS CloudFormation consente di descrivere e fornire le AWS risorse in formato JSON. I vantaggi di questo metodo includono la possibilità di gestire una raccolta di AWS risorse come singola unità e di replicare facilmente AWS le risorse tra le regioni.

Quando si esegue il provisioning AWS utilizzando CloudFormation, si creano modelli che descrivono le AWS risorse da utilizzare. L'esempio seguente è un frammento di codice di un modello che crea un gruppo di log e un filtro parametri, il quale conta 404 occorrenze e invia questa quantità al gruppo di log.

```
"WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},

"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "WebServerLogGroup"
```

```
        },
        "FilterPattern": "[ip, identity, user_id, timestamp, request, status_code =
 404, size, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    }
}
```

Questo è un esempio di base. È possibile configurare distribuzioni di CloudWatch Logs molto più ricche utilizzando. CloudFormationPer ulteriori informazioni sugli esempi di modelli, consulta Amazon CloudWatch Logs Template Snippets nella Guida per l'AWS CloudFormation utente. Per ulteriori informazioni sulle nozioni di base, consulta Nozioni di base di AWS CloudFormation nella Guida per l'utente di AWS CloudFormation .

Inserimento di log tramite endpoint HTTP

Amazon CloudWatch Logs fornisce endpoint HTTP che consentono di inviare i log direttamente ai CloudWatch log utilizzando semplici richieste HTTP POST. Questi endpoint supportano sia l'autenticazione SigV4 che quella con token bearer.

> **Important**
>
> Consigliamo di utilizzare l'autenticazione SigV4 per tutti i carichi di lavoro di produzione in cui è possibile l'integrazione con SDK. AWS SigV4 utilizza credenziali a breve termine e offre la posizione di sicurezza più solida. L'autenticazione con token Bearer (chiave API) è destinata a scenari in cui SigV4 non è fattibile, come i log forwarder di terze parti che non supportano l'integrazione SDK. AWS Per ulteriori informazioni, consulta Alternative alle chiavi di accesso a lungo termine nella Guida per l'utente IAM.

CloudWatch Logs supporta i seguenti endpoint di ingestione HTTP:

Endpoint	Path	Content-Type	Formato
OpenTelemetry Logs	`/v1/logs`	`application/json` o `application/x-protobuf`	JSON OTLP o Protobuf
HLC Logs	`/services/collector/event`	`application/json`	Formato HLC
ND-JSON Logs	`/ingest/bulk`	`application/json` o `application/x-ndjson`	JSON delimitato da nuove righe
Structured JSON Logs	`/ingest/json`	`application/json`	oggetto o array JSON

Comportamento comune

Tutti gli endpoint di ingestione HTTP condividono il seguente comportamento:

Autenticazione

Tutti gli endpoint supportano sia l'autenticazione SigV4 che quella con token bearer:

- SigV4 (consigliato): firma con firma standard versione 4. AWS Usa SigV4 ogni volta che l'applicazione o l'infrastruttura supporta l' AWS SDK o può firmare richieste. SigV4 utilizza credenziali a breve termine ed è il metodo di autenticazione più sicuro.
- Token Bearer: utilizza l'intestazione. `Authorization: Bearer <ACWL token>`
  - Il token deve essere un token portatore ACWL valido. Per le istruzioni di configurazione, vedere. the section called "Autenticazione con token Bearer"
  - Richiede le autorizzazioni `logs:PutLogEvents` e `logs:CallWithBearerToken` IAM.

Gruppo di log e flusso di log

- Fornito tramite header: `x-aws-log-group` e `x-aws-log-stream`
- I parametri di interrogazione `?logGroup=<name>&logStream=<name>` sono supportati anche su tutti gli endpoint tranne OTLP.
- Non è possibile utilizzare sia i parametri di query che le intestazioni per lo stesso parametro.
- Sono obbligatori sia il gruppo di log che il flusso di log.

Risposta

- Successo: `HTTP 200` con corpo `{}`
- Errori di convalida: `HTTP 400`
- Errori di autenticazione: `HTTP 401`

Configurazione dell'autenticazione con token al portatore

Prima di poter inviare i log utilizzando l'autenticazione con token bearer con uno qualsiasi degli endpoint di ingestione HTTP, devi:

- Creare un utente IAM con autorizzazioni Logs CloudWatch
- Genera credenziali specifiche per il servizio (bearer token)
- Crea un gruppo di log e un flusso di log
- Abilita l'autenticazione con token al portatore sul gruppo di log

> **Important**
>
> Consigliamo di utilizzare l'autenticazione SigV4 con credenziali a breve termine per tutti i carichi di lavoro laddove ciò sia possibile. SigV4 offre la posizione di sicurezza più solida. Limita l'uso delle chiavi API (bearer token) agli scenari in cui l'autenticazione a breve termine basata su credenziali non è possibile. Quando sei pronto per incorporare i CloudWatch log in applicazioni con requisiti di sicurezza più elevati, dovresti passare alle credenziali a breve termine. Per ulteriori informazioni, consulta Alternative alle chiavi di accesso a lungo termine nella Guida per l'utente IAM.

Opzione 1: avvio rapido a utilizzare la AWS console

La console AWS di gestione offre un flusso di lavoro semplificato per generare chiavi API per l'accesso agli endpoint HTTP.

Per configurare l'accesso agli endpoint HTTP utilizzando la console

1. Accedi alla console di AWS gestione.
2. Vai a CloudWatch> Impostazioni > Registri.
3. Nella sezione Chiavi API, scegli Genera chiave API.
4. Per Scadenza della chiave API, esegui una delle seguenti operazioni:
   - Seleziona una durata di scadenza della chiave API di 1, 5, 30, 90 o 365 giorni.
   - Scegli Durata personalizzata per specificare una data di scadenza della chiave API personalizzata.
   - Seleziona Non scade mai (scelta non consigliata).
5. Scegli Genera chiave API.

   La console automaticamente:

   - Crea un nuovo utente IAM con le autorizzazioni appropriate
   - Allega la policy gestita di CloudWatchLogsAPIKeyAccess (include `logs:PutLogEvents` e `logs:CallWithBearerToken` autorizzazioni)
   - Genera credenziali specifiche del servizio (chiave API)
6. Copia e salva in modo sicuro le credenziali visualizzate:

- ID chiave API (ID credenziale specifico del servizio)
- Chiave API segreta (token Bearer)

> **Important**
>
> Salva immediatamente l'API Key Secret. Non può essere recuperata più tardi. Se la perdi, dovrai generare una nuova chiave API.

7. Crea il gruppo di log e il flusso di log in cui verranno archiviati i log:

```
# Create the log group
aws logs create-log-group \
    --log-group-name /aws/hlc-logs/my-application \
    --region us-east-1

# Create the log stream
aws logs create-log-stream \
    --log-group-name /aws/hlc-logs/my-application \
    --log-stream-name application-stream-001 \
    --region us-east-1
```

8. Abilita l'autenticazione con token al portatore sul gruppo di log:

```
aws logs put-bearer-token-authentication \
    --log-group-identifier /aws/hlc-logs/my-application \
    --bearer-token-authentication-enabled \
    --region us-east-1
```

   Verifica la configurazione:

```
aws logs describe-log-groups \
    --log-group-name-prefix /aws/hlc-logs/my-application \
    --region us-east-1
```

Autorizzazioni incluse: l'utente IAM creato automaticamente avrà le seguenti autorizzazioni:

- `logs:PutLogEvents`— Invia eventi di registro a Logs CloudWatch
- `logs:CallWithBearerToken`— Autenticazione utilizzando il token bearer

- `kms:Describe*`,`kms:GenerateDataKey*`, `kms:Decrypt` — Accedi ai gruppi di log crittografati con KMS (con condizioni limitate al servizio di log)

Opzione 2: configurazione manuale

Se preferisci un maggiore controllo sulla configurazione IAM o devi personalizzare le autorizzazioni, puoi configurare manualmente l'accesso agli endpoint HTTP.

Passaggio 1: creare un utente IAM

Crea un utente IAM che verrà utilizzato per l'inserimento dei log:

1. Accedi alla console di AWS gestione e accedi a IAM.
2. Nel riquadro di navigazione a sinistra, seleziona Users (Utenti).
3. Selezionare Create user (Crea utente).
4. Inserisci un nome utente (ad esempio,`cloudwatch-logs-hlc-user`).
5. Scegli Next (Successivo).
6. Allega una delle seguenti politiche IAM:

   Opzione A: utilizza la policy gestita (consigliata)

   Allega la policy gestita di CloudWatchLogsAPIKeyaccesso.

   Opzione B: creare una politica personalizzata

   Crea e allega la seguente policy IAM:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "LogsAPIs",
            "Effect": "Allow",
            "Action": [
                "logs:CallWithBearerToken",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "KMSAPIs",
            "Effect": "Allow",
            "Action": [
                "kms:Describe*",
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "logs.*.amazonaws.com"
                    ]
                }
            },
            "Resource": "arn:aws:kms:*:*:key/*"
        }
    ]
}
```

7. Scegli Avanti, quindi Crea utente.

> **Note**
>
> Le autorizzazioni KMS sono necessarie se prevedi di inviare registri a gruppi di log crittografati con KMS. La condizione limita l'accesso KMS alle sole chiavi utilizzate tramite il servizio Logs. CloudWatch

Fase 2: Generazione di credenziali specifiche per il servizio (chiave API)

Genera la chiave API CloudWatch Logs utilizzando l'API. CreateServiceSpecificCredential Puoi anche usare il comando create-service-specific-credentialCLI. Per l'età delle credenziali, è possibile specificare un valore compreso tra 1 e 36600 giorni. Se non specifici l'età delle credenziali, la chiave API non scadrà.

Per generare una chiave API con una scadenza di 30 giorni:

```
aws iam create-service-specific-credential \
    --user-name cloudwatch-logs-hlc-user \
    --service-name logs.amazonaws.com \
```

```
    --credential-age-days 30
```

La risposta è un ServiceSpecificCredentialoggetto. Il `ServiceCredentialSecret` valore è la tua chiave API CloudWatch Logs (token bearer).

> **Important**
>
> Archivia il valore `ServiceCredentialSecret` in modo sicuro, poiché non potrai recuperarlo in un secondo momento. Se la perdi, dovrai generare una nuova chiave API.

Fase 3: Creare un gruppo di log e un flusso di log

Crea il gruppo di log e il flusso di log in cui verranno archiviati i log:

```
# Create the log group
aws logs create-log-group \
    --log-group-name /aws/hlc-logs/my-application \
    --region us-east-1

# Create the log stream
aws logs create-log-stream \
    --log-group-name /aws/hlc-logs/my-application \
    --log-stream-name application-stream-001 \
    --region us-east-1
```

Passaggio 4: abilitare l'autenticazione con token al portatore

Abilita l'autenticazione con token al portatore nel gruppo di log:

```
aws logs put-bearer-token-authentication \
    --log-group-identifier /aws/hlc-logs/my-application \
    --bearer-token-authentication-enabled \
    --region us-east-1
```

Verifica la configurazione:

```
aws logs describe-log-groups \
    --log-group-name-prefix /aws/hlc-logs/my-application \
    --region us-east-1
```

Controlla le autorizzazioni per la generazione e l'utilizzo delle chiavi API CloudWatch Logs

La generazione e l'utilizzo delle chiavi API CloudWatch Logs sono controllati da azioni e chiavi di condizione nei servizi CloudWatch Logs e IAM.

Controllo della generazione di chiavi API CloudWatch Logs

L'CreateServiceSpecificCredentialazione iam: controlla la generazione di una chiave specifica del servizio (come una chiave API CloudWatch Logs). È possibile questa azione agli utenti IAM come una risorsa per limitare gli utenti per cui è possibile generare una chiave.

Per imporre condizioni all'autorizzazione per l'azione `iam:CreateServiceSpecificCredential`, è possibile utilizzare le chiavi di condizione seguenti:

- iam: ServiceSpecificCredentialAgeDays — Consente di specificare, nella condizione, il tempo di scadenza della chiave in giorni. Ad esempio, si può utilizzare questa chiave di condizione per consentire solo la creazione di chiavi API che scadono entro 90 giorni.
- iam: ServiceSpecificCredentialServiceName — Consente di specificare, nella condizione, il nome di un servizio. Ad esempio, è possibile utilizzare questa chiave di condizione per consentire solo la creazione di chiavi API per CloudWatch Logs e non per altri servizi.

Controllo dell'utilizzo delle chiavi API CloudWatch Logs

L'`logs:CallWithBearerToken`azione controlla l'uso di una chiave API CloudWatch Logs. Per impedire a un'identità di utilizzare le chiavi API CloudWatch Logs, allega una policy che neghi l'`logs:CallWithBearerToken`azione all'utente IAM associato alla chiave.

Policy di esempio

Impedisci a un'identità di generare e utilizzare le chiavi API CloudWatch Logs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyCWLAPIKeys",
            "Effect": "Deny",
            "Action": [
                "iam:CreateServiceSpecificCredential",
```

```
                "logs:CallWithBearerToken"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Warning**
>
> Questa politica impedirà la creazione di credenziali per tutti i AWS servizi che supportano la creazione di credenziali specifiche del servizio. Per ulteriori informazioni, consulta Credenziali specifiche del servizio per gli utenti IAM.

Impedisci a un'identità di utilizzare le chiavi API Logs CloudWatch

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "logs:CallWithBearerToken",
            "Resource": "*"
        }
    ]
}
```

Consenti la creazione di chiavi CloudWatch Logs solo se scadono entro 90 giorni

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceSpecificCredential",
            "Resource": "arn:aws:iam::123456789012:user/username",
            "Condition": {
                "StringEquals": {
                    "iam:ServiceSpecificCredentialServiceName": "logs.amazonaws.com"
                },
                "NumericLessThanEquals": {
                    "iam:ServiceSpecificCredentialAgeDays": "90"
```

```
                }
            }
        }
    ]
}
```

Chiavi API rotanti

La rotazione regolare delle chiavi API riduce il rischio di accessi non autorizzati. Ti consigliamo di stabilire un programma di rotazione in linea con le politiche di sicurezza dell'organizzazione.

Processo di rotazione

Per ruotare una chiave API senza interrompere la consegna dei log, segui questa procedura:

1. Crea una nuova credenziale (secondaria) per l'utente IAM:

   ```
   aws iam create-service-specific-credential \
       --user-name cloudwatch-logs-hlc-user \
       --service-name logs.amazonaws.com \
       --credential-age-days 90
   ```

2. (Facoltativo) Archivia la nuova credenziale AWS Secrets Manager per il recupero sicuro e la rotazione automatica.
3. Importa la nuova credenziale nel portale del fornitore o aggiorna la configurazione dell'applicazione per utilizzare la nuova chiave API.
4. Imposta la credenziale originale su inattiva:

   ```
   aws iam update-service-specific-credential \
       --user-name cloudwatch-logs-hlc-user \
       --service-specific-credential-id ACCA1234EXAMPLE1234 \
       --status Inactive
   ```

5. Verifica che la consegna dei log non sia influenzata dal monitoraggio della `IncomingBytes` metrica relativa al tuo gruppo di log in. CloudWatch Per ulteriori informazioni, consulta Monitoraggio con CloudWatch metriche.
6. Dopo aver confermato l'avvenuta consegna con la nuova chiave, elimina la credenziale precedente:

   ```
   aws iam delete-service-specific-credential \
   ```

```
    --service-specific-credential-id ACCA1234EXAMPLE1234
```

Monitoraggio della scadenza delle chiavi

Per verificare la data di creazione e lo stato delle chiavi API esistenti, usa il list-service-specific-credentialscomando:

```
aws iam list-service-specific-credentials \
    --user-name cloudwatch-logs-hlc-user \
    --service-name logs.amazonaws.com
```

La risposta include `CreateDate` e `Status` per ogni credenziale. Utilizza queste informazioni per identificare le chiavi che stanno per scadere o che sono state attive più a lungo di quanto consentito dalla politica di rotazione.

Risposta a una chiave API compromessa

Se sospetti che una chiave API sia stata compromessa, procedi immediatamente come segue:

1. Disattiva immediatamente la chiave per evitare ulteriori utilizzi non autorizzati:

   ```
   aws iam update-service-specific-credential \
       --user-name cloudwatch-logs-hlc-user \
       --service-specific-credential-id ACCA1234EXAMPLE1234 \
       --status Inactive
   ```

2. CloudTrail Esamina i log per determinare l'ambito dell'accesso non autorizzato. Scopri the section called "Registrazione dell'utilizzo delle chiavi API con CloudTrail" come abilitare il controllo dell'utilizzo delle chiavi API.
3. Crea una chiave sostitutiva seguendo il processo di rotazione descritto inthe section called "Processo di rotazione".
4. Eliminare la chiave compromessa dopo aver effettuato la sostituzione:

   ```
   aws iam delete-service-specific-credential \
       --service-specific-credential-id ACCA1234EXAMPLE1234
   ```

5. Allega una politica di rifiuto se devi bloccare immediatamente l'accesso a tutti i token portatori per l'utente IAM durante le indagini:

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "logs:CallWithBearerToken",
        "Resource": "*"
    }
}
```

> **Note**
>
> Per eseguire queste azioni tramite l'API, devi autenticarti con AWS le credenziali e non con una CloudWatch chiave API Logs.

Puoi anche utilizzare le seguenti operazioni dell'API IAM per gestire le chiavi compromesse:

- ResetServiceSpecificCredential— Reimposta la chiave per generare una nuova password senza eliminare la credenziale. La chiave non deve essere scaduta.

Le migliori pratiche di sicurezza per le chiavi API

Segui queste best practice per proteggere le tue chiavi API CloudWatch Logs:

- Non incorporare mai le chiavi API nel codice sorgente. Non codificate le chiavi API nel codice dell'applicazione né affidatele a sistemi di controllo della versione. Se una chiave viene salvata accidentalmente in un archivio pubblico, la scansione AWS automatica potrebbe contrassegnarla ed è necessario ruotarla immediatamente.
- Usa un gestore di segreti. Archivia le chiavi API in una soluzione di gestione dei segreti equivalente AWS Secrets Manager o in una soluzione di gestione dei segreti equivalente. Ciò consente il controllo centralizzato degli accessi, la registrazione degli audit e la rotazione automatizzata.
- Imposta una scadenza per tutte le chiavi. Specificate sempre un `--credential-age-days` valore durante la creazione delle chiavi API. Per imporre una durata massima delle chiavi in tutta l'organizzazione, utilizza la chiave di condizione `iam:ServiceSpecificCredentialAgeDays` IAM. Per vedere un esempio, consulta the section called "Consenti la creazione di chiavi CloudWatch Logs solo se scadono entro 90 giorni".

- Applica le autorizzazioni con privilegi minimi. Ambita le autorizzazioni dell'utente IAM solo per i gruppi di log e le azioni richieste. Utilizza la policy di CloudWatchLogsAPIKeyaccesso gestito come punto di partenza e limita ulteriormente se necessario.
- Abilita la CloudTrail registrazione. Verifica l'utilizzo delle chiavi API abilitando gli eventi di CloudTrail dati per`AWS::Logs::LogGroupAuthorization`. Per informazioni, consulta the section called "Registrazione dell'utilizzo delle chiavi API con CloudTrail".
- Monitora con IAM Access Analyzer. Utilizza IAM Access Analyzer per identificare le credenziali non utilizzate e le policy eccessivamente permissive associate agli utenti IAM della tua chiave API.
- Ruota i tasti regolarmente. Stabilite un programma di rotazione e seguite la procedura descritta inthe section called "Chiavi API rotanti".

Registrazione dell'utilizzo delle chiavi API con CloudTrail

È possibile utilizzare AWS CloudTrail per registrare gli eventi relativi ai dati per l'utilizzo della chiave API CloudWatch Logs. CloudWatch Logs emette eventi di `AWS::Logs::LogGroupAuthorization` dati per `CallWithBearerToken` le chiamate, consentendoti di verificare quando e come le chiavi API vengono utilizzate per inviare i log.

Per abilitare la CloudTrail registrazione per l'utilizzo delle chiavi API CloudWatch Logs:

> **Note**
>
> Il bucket S3 che specifichi per il percorso deve avere una policy sui bucket che CloudTrail consenta di scrivere file di registro al suo interno. Per ulteriori informazioni, consulta la policy sui bucket di Amazon S3 per. CloudTrail

1. Crea un percorso:

```
aws cloudtrail create-trail \
    --name cloudwatch-logs-api-key-audit \
    --s3-bucket-name my-cloudtrail-bucket \
    --region us-east-1
```

2. Configura i selettori di eventi avanzati per acquisire gli eventi di autorizzazione dei gruppi di CloudWatch log log:

```
aws cloudtrail put-event-selectors \
```

```
    --region us-east-1 \
    --trail-name cloudwatch-logs-api-key-audit \
    --advanced-event-selectors '[{
        "Name": "CloudWatch Logs API key authorization events",
        "FieldSelectors": [
            { "Field": "eventCategory", "Equals": ["Data"] },
            { "Field": "resources.type", "Equals":
 ["AWS::Logs::LogGroupAuthorization"] }
        ]
    }]'
```

3. Inizia la registrazione dei percorsi:

```
aws cloudtrail start-logging \
    --name cloudwatch-logs-api-key-audit \
    --region us-east-1
```

Invio di log utilizzando l'endpoint OTLP (Logs) OpenTelemetry

L'endpoint OpenTelemetry Logs (`/v1/logs`) accetta i dati di log del OpenTelemetry protocollo (OTLP) con codifica JSON o Protobuf. Per informazioni dettagliate sull'endpoint OTLP, inclusi configurazione e utilizzo, consulta Inviare metriche e tracce a with. CloudWatch OpenTelemetry

Se utilizzi l'autenticazione con token al portatore, completa i passaggi di configurazione prima di procedere. the section called "Autenticazione con token Bearer"

Formato della richiesta

- Metodo: `POST`
- Tipo di contenuto: o `application/json` `application/x-protobuf`
- Gruppo di log: solo `x-aws-log-group` intestazione (parametro di query non supportato)
- Flusso di registro: intestazione `x-aws-log-stream`

Richiesta di esempio

```
curl -X POST "https://logs.<region>.amazonaws.com/v1/logs" \
  -H "Authorization: Bearer ACWL<token>" \
  -H "Content-Type: application/json" \
  -H "x-aws-log-group: MyLogGroup" \
```

```
  -H "x-aws-log-stream: MyLogStream" \
  -d '{
  "resourceLogs": [
    {
      "resource": {
        "attributes": [
          {
            "key": "service.name",
            "value": { "stringValue": "my-service" }
          }
        ]
      },
      "scopeLogs": [
        {
          "scope": {
            "name": "my-library",
            "version": "1.0.0"
          },
          "logRecords": [
            {
              "timeUnixNano": "1741900000000000000",
              "severityNumber": 9,
              "severityText": "INFO",
              "body": {
                "stringValue": "User logged in successfully"
              },
              "attributes": [
                {
                  "key": "user.id",
                  "value": { "stringValue": "12345" }
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}'
```

Risposte

Operazione riuscita (tutti gli eventi sono accettati):

```
HTTP 200 OK
{}
```

Successo parziale (alcuni eventi rifiutati):

```
{
  "partialSuccess": {
    "rejectedLogRecords": 5,
    "errorMessage": "{\"tooOldLogEventCount\": 3, \"tooNewLogEventCount\": 1,
 \"expiredLogEventCount\": 1}"
  }
}
```

Quando la richiesta Content-Type è`application/x-protobuf`, la risposta viene restituita come messaggio `ExportLogsServiceResponse` protobuf serializzato con gli stessi campi.

Comportamenti specifici di OTLP

I seguenti comportamenti sono specifici dell'endpoint OTLP e non sono presenti negli altri endpoint di ingestione HTTP:

- Intestazione Retry-After: inclusa nelle risposte 503 e 429 per indicare quando il client deve riprovare.

Invio di registri utilizzando l'endpoint HLC (registri HLC)

L'endpoint HLC Logs () si basa sul formato HTTP Log Collector (HLC`/services/collector/event`).

Se utilizzi l'autenticazione con token bearer, completa i passaggi di configurazione prima di procedere. the section called “Autenticazione con token Bearer”

Modalità di input

Ogni evento è un oggetto JSON con un `"event"` campo obbligatorio. Campi di metadati opzionali:`"time"`,,`"host"`,`"source"`,`"sourcetype"`. `"index"`

Evento singolo:

```
{"event":"Hello world!","time":1486683865.0}
```

Matrice di eventi JSON:

```
[
  {"event":"msg1","time":1486683865.0},
  {"event":"msg2","time":1486683866.0}
]
```

Eventi concatenati/in batch (nessun array wrapper):

```
{"event":"msg1","time":1486683865.0}{"event":"msg2","time":1486683866.0}
```

Campo evento (obbligatorio)

Il `"event"` campo è obbligatorio. Il suo valore può essere di qualsiasi tipo JSON:

```
{"event":"a string message"}
{"event":{"message":"structured data","severity":"INFO"}}
{"event":42}
{"event":true}
```

Gli oggetti senza `"event"` campo vengono ignorati silenziosamente:

```
{"message":"this is skipped — no event field"}
```

Campo orario (opzionale)

Il `"time"` campo è in secondi epocali (non in millisecondi), con decimali opzionali per una precisione inferiore al secondo.

Formato	Esempio	Interpretato come
Float	`"time":1486683865.500`	1486683865500 ms
Numero intero	`"time":1486683865`	1486683865000 ms
Stringa (float)	`"time":"1486683865.500"`	1486683865500 ms

Formato	Esempio	Interpretato come
Stringa (numero intero)	`"time":"1486683865"`	1486683865000 ms
Mancante	(nessun campo temporale)	Ora corrente del server
Non valido	`"time":"invalid"`	Ora corrente del server

Content-Type

`application/json`È accettata solo.

Tipi di valori JSON accettati

Tipo di primo livello	Comportamento
Oggetto con `"event"`	Accettato
Oggetto senza `"event"`	Saltato
Matrice di oggetti	Ogni elemento è stato elaborato singolarmente
Oggetti concatenati	Ogni oggetto viene elaborato singolarmente
Primitiva (stringa, numero, booleano, nullo)	Saltato

Formato dell'endpoint

L'URL dell'endpoint HLC segue questo formato:

```
https://logs.<region>.amazonaws.com/services/collector/event?
logGroup=<name>&logStream=<name>[&entityName=<name>&entityEnvironment=<environment>]
```

Parametri richiesti:

- `<region>`— AWS Regione (ad esempio`us-east-1`,`eu-west-1`)

- `logGroup`— nome del gruppo di log con codifica URL
- `logStream`— nome del flusso di log con codifica URL

Parametri opzionali:

Facoltativamente, è possibile associare gli eventi di registro a un'`Service`entità includendo i seguenti parametri di query. Poiché i log inviati tramite l'endpoint HLC sono telemetria personalizzata, non vengono associati automaticamente a un'entità. Fornendo questi parametri, CloudWatch Logs crea un'entità con `KeyAttributes.Type` set to `Service` e la associa ai tuoi eventi di registro. Ciò consente alla funzionalità relativa a Explore di CloudWatch correlare questi log con altri dati di telemetria (metriche, tracce e registri) dello stesso servizio, semplificando la risoluzione dei problemi e il monitoraggio delle applicazioni su diversi tipi di segnale. Per ulteriori informazioni sulle entità e sulla relativa telemetria, vedere Aggiungere informazioni correlate alla telemetria personalizzata.

- `entityName`— Il nome dell'entità di servizio da associare agli eventi del registro. Questo valore viene memorizzato come entità `KeyAttributes.Name` (ad esempio, `my-application oapi.myservice.com`).
- `entityEnvironment`— L'ambiente in cui è ospitato il servizio o a cosa appartiene. Questo valore viene memorizzato come entità `KeyAttributes.Environment` (ad esempio`production`,`ec2:default`, `oeks:my-cluster/default`).

Formato della richiesta

Invia i log utilizzando HTTP POST con le intestazioni e il testo seguenti:

Intestazioni:

- `Authorization: Bearer <your-bearer-token>`
- `Content-Type: application/json`

Formato del corpo:

Il corpo della richiesta deve essere in formato JSON con una serie di eventi:

```
{
    "event": [
        {
```

```
            "time": 1730141374.001,
            "event": "Application started successfully",
            "host": "web-server-1",
            "source": "application.log",
            "severity": "info"
        },
        {
            "time": 1730141374.457,
            "event": "User login successful",
            "host": "web-server-1",
            "source": "auth.log",
            "user": "john.doe"
        }
    ]
}
```

Descrizioni dei campi:

- `time`— Timestamp Unix epoch in secondi, con decimale opzionale per una precisione inferiore al secondo (opzionale)
- `event`— Il messaggio di registro o i dati dell'evento (obbligatorio)
- `host`— Nome host o identificatore di origine (opzionale)
- `source`— Identificatore di origine del registro (opzionale)

Se necessario, è possibile includere campi personalizzati aggiuntivi.

Richiesta di esempio

```
curl -X POST "https://logs.<region>.amazonaws.com/services/collector/event?
logGroup=MyLogGroup&logStream=MyStream" \
  -H "Authorization: Bearer ACWL<token>" \
  -H "Content-Type: application/json" \
  -d '{"event":{"message":"User logged
 in","user_id":"u-123"},"time":1486683865.0,"host":"web-01","source":"auth-service"}'
```

Best practice

Eventi di raggruppamento

Per prestazioni ed efficienza migliori:

- Batch di più eventi in un'unica richiesta, quando possibile
- Dimensione del batch consigliata: 10—100 eventi per richiesta
- Dimensione massima della richiesta: 1 MB

Gestione degli errori

Implementa una corretta gestione degli errori nell'applicazione. Codici di stato HTTP comuni:

- `200 OK`— Registri inseriti con successo
- `400 Bad Request`— Formato o parametri di richiesta non validi
- `401 Unauthorized`— Token al portatore non valido o scaduto
- `403 Forbidden`— Autorizzazioni insufficienti
- `404 Not Found`— Il gruppo o lo stream di log non esiste
- `429 Too Many Requests`— Limite di velocità superato
- `500 Internal Server Error`— Errore di servizio (nuovo tentativo con backoff esponenziale)

Limitazioni

- Dimensione massima dell'evento: 256 KB per evento
- Dimensione massima della richiesta: 1 MB
- Numero massimo di eventi per richiesta: 10.000
- I nomi dei gruppi di log devono seguire le convenzioni di denominazione CloudWatch dei log
- L'autenticazione con token portatore deve essere abilitata nel gruppo di log se viene utilizzata l'autenticazione con token portatore.

Invio di log utilizzando l'endpoint NDJSON (ND-JSON Logs)

L'endpoint ND-JSON Logs () accetta i log in formato NDJSON (Newline Delimited JSON). `/ingest/bulk` Ogni riga contiene esattamente un valore JSON, separato da caratteri di nuova riga.

Se utilizzi l'autenticazione con token al portatore, completa i passaggi di configurazione indicati prima di procedere. the section called “Autenticazione con token Bearer”

Formato della richiesta

Invia un valore JSON per riga, separato da `\n` (LF) o `\r\n` (CRLF). Le righe vuote vengono ignorate silenziosamente.

```
{"timestamp":1771007942000,"message":"event one","level":"INFO"}
{"timestamp":1771007943000,"message":"event two","level":"ERROR"}
{"timestamp":1771007944000,"message":"event three","level":"DEBUG"}
```

Entrambi i tipi `application/json` `application/x-ndjson` sono accettati come Content-Type.

Tipi di valori JSON accettati

In base alle specifiche NDJSON (RFC 8259), qualsiasi valore JSON valido viene accettato su ogni riga.

Oggetti JSON (i più comuni):

```
{"timestamp":1771007942000,"message":"User logged in","service":"auth"}
{"timestamp":1771007943000,"error":"Connection timeout","service":"api"}
```

Matrici JSON (appiattite in singoli eventi):

```
[{"timestamp":1000,"message":"a"},{"timestamp":2000,"message":"b"}]
```

Questa singola riga produce 2 eventi. Ogni elemento dell'array diventa un evento di registro separato.

Valori primitivi:

```
"a plain string log message"
42
true
null
```

Ogni primitiva diventa un evento a sé stante con il timestamp corrente del server.

Tipi misti:

```
{"timestamp":1771007942000,"message":"structured event"}
```

```
"unstructured string message"
42
{"timestamp":1771007943000,"error":"something failed"}
```

Tutte e 4 le righe sono accettate come eventi validi.

Contenuto della riga	Comportamento
Oggetto JSON	Accettato, timestamp estratto se presente
Array JSON	Appiattito: ogni elemento diventa un evento separato
Matrice vuota `[]`	Accettato, produce 0 eventi
Stringa JSON	Accettato come messaggio di evento
Numero JSON	Accettato come messaggio di evento
Booleano JSON	Accettato come messaggio di evento
JSON nullo	Accettato come messaggio di evento
JSON non valido	Ignorato (conteggiato, l'elaborazione continua)
Riga vuota	Ignorata (non contata come ignorata)

Campo Timestamp

Il `"timestamp"` campo è espresso in millisecondi epocali (non secondi).

Formato	Esempio	Interpretato come
Numerico (millis)	`"timestamp":1771007942000`	1771007942000 ms
Mancante	(nessun campo timestamp)	Ora corrente del server
Non numerico	`"timestamp":"invalid"`	Ora corrente del server
Linea non oggetto	`"hello"`, `42`, `true`	Ora corrente del server

Righe non valide

Le righe che non sono JSON valide vengono ignorate e contate silenziosamente. L'elaborazione continua con la riga successiva.

```
{"message":"valid event"}
this is not valid json
{"message":"another valid event"}
```

Risultato: 2 eventi inseriti, 1 ignorato. Restituisce `HTTP 200`.

Se tutte le righe non sono valide, restituisce con. `HTTP 400` `"All events were invalid"`

Richiesta di esempio

```
curl -X POST "https://logs.<region>.amazonaws.com/ingest/bulk?
logGroup=MyLogGroup&logStream=MyStream" \
  -H "Authorization: Bearer ACWL<token>" \
  -H "Content-Type: application/x-ndjson" \
  -d '{"timestamp":1771007942000,"message":"User logged in","level":"INFO"}
{"timestamp":1771007943000,"message":"Query took 42ms","level":"DEBUG"}
{"timestamp":1771007944000,"error":"Connection refused","level":"ERROR"}'
```

Risposte

Operazione riuscita (tutti gli eventi sono accettati):

```
HTTP 200 OK
{}
```

Successo parziale (alcuni eventi rifiutati):

```
{
  "partialSuccess": {
    "rejectedLogRecords": 5,
    "errorMessage": "{\"tooOldLogEventCount\": 3, \"tooNewLogEventCount\": 1,
 \"expiredLogEventCount\": 1}"
  }
}
```

Il `rejectedLogRecords` campo indica il numero totale di eventi rifiutati. Il `errorMessage` campo contiene una suddivisione con codifica JSON in base al motivo del rifiuto:

- `tooOldLogEventCount`— Eventi con timestamp precedenti al periodo di conservazione
- `tooNewLogEventCount`— Eventi con timestamp troppo lontani nel futuro
- `expiredLogEventCount`— Eventi scaduti durante l'elaborazione

Best practice

Eventi di raggruppamento

Per prestazioni ed efficienza migliori:

- Batch di più eventi in un'unica richiesta, quando possibile
- Dimensione del batch consigliata: 10—100 eventi per richiesta
- Dimensione massima della richiesta: 1 MB

Gestione degli errori

Implementa una corretta gestione degli errori nell'applicazione. Codici di stato HTTP comuni:

- `200 OK`— Registri inseriti con successo
- `400 Bad Request`— Formato o parametri di richiesta non validi
- `401 Unauthorized`— Token al portatore non valido o scaduto
- `403 Forbidden`— Autorizzazioni insufficienti
- `404 Not Found`— Il gruppo o lo stream di log non esiste

- `429 Too Many Requests`— Limite di velocità superato
- `500 Internal Server Error`— Errore di servizio (nuovo tentativo con backoff esponenziale)

Limitazioni

- Dimensione massima dell'evento: 256 KB per evento
- Dimensione massima della richiesta: 1 MB
- Numero massimo di eventi per richiesta: 10.000
- I nomi dei gruppi di log devono seguire le convenzioni di denominazione CloudWatch dei log
- L'autenticazione con token portatore deve essere abilitata nel gruppo di log se viene utilizzata l'autenticazione con token portatore.

Invio di log utilizzando l'endpoint JSON strutturato (Structured JSON Logs)

L'endpoint Structured JSON Logs (`/ingest/json`) accetta lo standard JSON: un singolo oggetto JSON o un array di oggetti JSON. Questo endpoint è progettato per dati di registro strutturati in cui ogni evento è un oggetto JSON.

Se utilizzi l'autenticazione con token al portatore, completa i passaggi di configurazione indicati prima di procedere. the section called “Autenticazione con token Bearer”

Formato della richiesta

Solo `application/json` viene accettato come Content-Type.

Oggetto JSON singolo:

```
{"timestamp":1771007942000,"message":"single event","level":"INFO"}
```

Matrice di oggetti JSON:

```
[
  {"timestamp":1771007942000,"message":"event one","level":"INFO"},
  {"timestamp":1771007943000,"message":"event two","level":"ERROR"}
]
```

Tipi di valori JSON accettati

Questo endpoint è rigoroso: solo gli oggetti JSON vengono accettati come eventi.

Input	Comportamento
Oggetto JSON singolo	Accettato come evento unico
Matrice di oggetti JSON	Ogni oggetto diventa un evento separato
Matrice vuota `[]`	Accettato, produce 0 eventi
Non è un oggetto nell'array (stringa, numero, ecc.)	Saltato
Primitiva di primo livello (,) `"hello"` `42`	Saltato
Oggetti concatenati `{...}{...}`	Solo il primo oggetto è stato analizzato

Esempio: array con tipi misti:

```
[
  {"timestamp":1771007942000,"message":"valid object"},
  "just a string",
  42,
  {"timestamp":1771007943000,"message":"another valid object"}
]
```

Risultato: 2 eventi inseriti (gli oggetti), 2 ignorati (la stringa e il numero).

Campo Timestamp

Il `"timestamp"` campo è espresso in millisecondi di epoca, come l'endpoint NDJSON.

Formato	Esempio	Interpretato come
Numerico (millis)	`"timestamp":1771007942000`	1771007942000 ms

Formato	Esempio	Interpretato come
Mancante	(nessun campo timestamp)	Ora corrente del server
Non numerico	`"timestamp":"invalid"`	Ora corrente del server

Richiesta di esempio

```
curl -X POST "https://logs.<region>.amazonaws.com/ingest/json?
logGroup=MyLogGroup&logStream=MyStream" \
  -H "Authorization: Bearer ACWL<token>" \
  -H "Content-Type: application/json" \
  -d '[{"timestamp":1771007942000,"message":"User logged in","user_id":"u-123"},
{"timestamp":1771007943000,"message":"Order placed","order_id":"o-456"}]'
```

Risposte

Operazione riuscita (tutti gli eventi sono accettati):

```
HTTP 200 OK
{}
```

Successo parziale (alcuni eventi rifiutati):

```
{
  "partialSuccess": {
    "rejectedLogRecords": 5,
    "errorMessage": "{\"tooOldLogEventCount\": 3, \"tooNewLogEventCount\": 1,
 \"expiredLogEventCount\": 1}"
  }
}
```

Il `rejectedLogRecords` campo indica il numero totale di eventi rifiutati. Il `errorMessage` campo contiene una suddivisione con codifica JSON in base al motivo del rifiuto:

- `tooOldLogEventCount`— Eventi con timestamp precedenti al periodo di conservazione
- `tooNewLogEventCount`— Eventi con timestamp troppo lontani nel futuro
- `expiredLogEventCount`— Eventi scaduti durante l'elaborazione

Best practice

Eventi di raggruppamento

Per prestazioni ed efficienza migliori:

- Batch di più eventi in un'unica richiesta, quando possibile
- Dimensione del batch consigliata: 10—100 eventi per richiesta
- Dimensione massima della richiesta: 1 MB

Gestione degli errori

Implementa una corretta gestione degli errori nell'applicazione. Codici di stato HTTP comuni:

- `200 OK`— Registri inseriti con successo
- `400 Bad Request`— Formato o parametri di richiesta non validi
- `401 Unauthorized`— Token al portatore non valido o scaduto
- `403 Forbidden`— Autorizzazioni insufficienti
- `404 Not Found`— Il gruppo o lo stream di log non esiste
- `429 Too Many Requests`— Limite di velocità superato
- `500 Internal Server Error`— Errore di servizio (nuovo tentativo con backoff esponenziale)

Limitazioni

- Dimensione massima dell'evento: 256 KB per evento
- Dimensione massima della richiesta: 1 MB
- Numero massimo di eventi per richiesta: 10.000
- I nomi dei gruppi di log devono seguire le convenzioni di denominazione CloudWatch dei log
- L'autenticazione con token portatore deve essere abilitata nel gruppo di log se viene utilizzata l'autenticazione con token portatore.

Confronto degli endpoint di ingestione HTTP

Funzionalità	Registri HLC	Registri ND-JSON	Registri JSON strutturati	OpenTelemetry Registri
Path	`/services/collector/event`	`/ingest/bulk`	`/ingest/json`	`/v1/logs`
Content-Type	`application/json`	`application/json` o `application/x-ndjson`	`application/json`	`application/json` o `application/x-protobuf`
Campo Timestamp	`"time"`(secondi)	`"timestamp"` (millisecondi)	`"timestamp"` (millisecondi)	`"timeUnixNano"` (nanosecondi)
Campi obbligatori	`"event"`	Nessuno	Nessuno	Struttura OTLP () `"resourceLogs"`
Risposta parziale di successo	No	Sì	Sì	Sì
Supporto per i parametri di interrogazione	Sì	Sì	Sì	No (solo intestazioni)
Metadati dell'entità	Sì	Sì	Sì	No
Accetta primitive	No	Sì	No	No
Analisi basata su linee	No	Sì	No	No
Supporto Protobuf	No	No	No	Sì
Intestazione Retry-After	No	No	No	Sì

Scelta di un endpoint

- Stai usando il formato HLC? Usa i registri HLC. I tuoi payload HLC esistenti funzionano con modifiche minime.
- Registri di streaming line-by-line? Usa i registri ND-JSON. Ideale per pipeline di log che emettono un evento per riga. Il più flessibile: accetta qualsiasi tipo di valore JSON.
- Invio di payload JSON strutturati? Usa log JSON strutturati. Ideale per applicazioni che producono oggetti o array JSON ben formati.
- Lo stai già utilizzando? OpenTelemetry Usa OpenTelemetry i registri. Accetta il formato OTLP JSON o Protobuf e supporta risposte di successo parziali con semantica dei tentativi.

Ingestione di syslog

L'inserimento di CloudWatch syslog gestito da Amazon Logs consente di inviare messaggi syslog (RFC 5424, RFC 3164 e Cisco FTD/ASA) da firewall, router, switch e server Linux direttamente in Logs, senza installare o gestire alcun agente. CloudWatch Le tue fonti syslog inviano messaggi tramite TCP, TCP+TLS o UDP a un endpoint VPC nel tuo account. Il traffico viene incanalato AWS PrivateLink verso il servizio CloudWatch syslog Logs, che analizza automaticamente i messaggi in arrivo ed estrae campi strutturati come struttura, gravità, nome host e nome dell'applicazione, eliminando la necessità di pipeline di analisi personalizzate. Puoi quindi interrogare questi campi utilizzando CloudWatch Logs Analytics per analizzare gli eventi di sicurezza o risolvere i problemi di connettività. Ciò consente di centralizzare la visibilità dei log dell'infrastruttura, semplificare i flussi di lavoro operativi e ridurre il sovraccarico derivante dall'implementazione e dalla gestione degli agenti di raccolta dei log in ambienti distribuiti.

Se le tue fonti syslog sono già all'interno del tuo Amazon VPC, possono inviare messaggi direttamente all'endpoint VPC. Se le tue fonti sono esterne AWS (data center locali, filiali o strutture di co-location), possono raggiungere l'endpoint VPC tramite la tua VPN o la connessione Direct Connect.

Protocolli e porte supportati

Protocollo	Porta	Note
TCP + TLS	6514	Crittografato in transito. Consigliato per i requisiti di conformità.
Testo in chiaro TCP	1514	Testo in chiaro terminato (isolato dalla rete AWS PrivateLink).
UDP	514	Best-effort consegna.

Il protocollo TLS sulla porta 6514 viene terminato dal sistema di bilanciamento del carico di rete utilizzando un certificato AWS gestito emesso da Amazon Trust Services. I tuoi client syslog si fidano automaticamente di questo certificato senza alcuna configurazione speciale.

> **Note**
>
> UDP è il protocollo best-effort. I messaggi potrebbero andare persi a causa delle condizioni della rete. Usa TCP per una consegna affidabile.

Formati syslog supportati

CloudWatch Logs rileva e analizza automaticamente i seguenti formati syslog:

- RFC 5424 (il formato più recente): include dati strutturati, timestamp ISO 8601 e campi espliciti relativi al nome dell'applicazione e all'ID del processo.
- RFC 3164 (BSD syslog, il formato precedente): include timestamp e un campo TAG. BSD-style Ancora ampiamente utilizzato da dispositivi di rete come firewall, router e switch.
- Cisco FTD/ASA: il formato syslog utilizzato dai dispositivi Cisco Firepower Threat Defense (FTD) e Adaptive Security Appliance (ASA). I messaggi sono identificati dal tag o nel corpo del `%FTD-` messaggio. `%ASA-`

I messaggi vengono archiviati nel gruppo di log nel formato raw originale. CloudWatch Logs estrae automaticamente i campi strutturati da ogni formato, che puoi interrogare utilizzando CloudWatch Logs Insights.

Campi estratti da RFC 5424

Campo	Description
`facility`	Nome della categoria di registro (ad esempio,`kern`,`auth`). `local0`
`facilityCode`	Codice numerico della struttura (0—23).
`severity`	Nome del livello di gravità (ad esempio,`emerg`,`err`). `info`
`severityCode`	Codice di gravità numerico (0—7).
`timestamp`	Timestamp del messaggio in formato ISO 8601.
`hostname`	Nome host del dispositivo di origine.

Campo	Description
`appName`	Nome applicazione.
`procId`	ID processo.
`msgId`	Identificatore del messaggio.
`structuredData`	Elementi di dati strutturati RFC 5424 (metadati chiave-valore).
`message`	Corpo del messaggio.

Campi estratti in RFC 3164

Campo	Description
`facility`	Nome della categoria di registro.
`facilityCode`	Codice numerico della struttura (0—23).
`severity`	Nome del livello di gravità.
`severityCode`	Codice di gravità numerico (0—7).
`timestamp`	Timestamp del messaggio (convertito dal formato BSD a ISO 8601).
`hostname`	Nome host del dispositivo di origine.
`appName`	Nome dell'applicazione (estratto dal campo TAG).
`procId`	ID del processo (estratto dal campo TAG, se presente).
`message`	Corpo del messaggio.

Campi FTD/ASA estratti da Cisco

Campo	Description
`device`	Tipo di dispositivo (`FTD`,`ASA`, o`FMC-AUDIT-LOG`).
`timestamp`	Timestamp del messaggio (formato RFC 3164 o RFC 5424, a seconda della configurazione del dispositivo).
`deviceId`	Nome host del dispositivo (presente quando `device-id` la registrazione è configurata sull'appliance).
`severity`	Nome del livello di gravità (ad esempio, `informational warning`,`critical`).
`severityLevel`	Livello di gravità numerico (0—7).
`messageId`	Identificatore numerico del messaggio Cisco (ad esempio,). `106023` `302013`
`subsystem`	Nome del sottosistema (presente per alcuni tipi di messaggi).
`message`	Corpo del messaggio (testo normale) o singoli campi chiave-valore quando il corpo utilizza il formato strutturato di Cisco.

Consegna dei messaggi

A differenza dell' HTTP-based inserimento in cui il server restituisce un codice di stato per ogni richiesta, syslog non fornisce una conferma di successo per messaggio al mittente. Una volta ricevuti dal servizio, i messaggi vengono memorizzati nel buffer e recapitati al gruppo di log con tentativi per individuare eventuali errori transitori. Le garanzie di consegna dipendono dal protocollo di trasporto scelto:

- TCP (porte 6514 e 1514): fornisce una consegna affidabile in condizioni operative normali.

  Quando la consegna non è possibile, il servizio ripristina la connessione TCP per segnalare l'errore al client. I messaggi in corso su quella connessione potrebbero essere eliminati, ma il ripristino della connessione genera una contropressione immediata in modo che il cliente possa rilevare

il problema e archiviare i messaggi in buffer a livello locale fino alla risoluzione della condizione. In condizioni di scarsa capacità, il servizio rifiuta tempestivamente le nuove connessioni TCP, fornendo lo stesso segnale di contropressione.

Le condizioni che causano il ripristino della connessione includono:

- La politica degli endpoint VPC nega l'accesso
- La quota del tuo account è stata superata `PutLogEvents`
- Il gruppo di log di destinazione non esiste
- La politica delle risorse sul gruppo di log nega l'accesso

- UDP (porta 514): consegna. Best-effort I messaggi che non possono essere recapitati vengono eliminati senza alcun feedback al mittente. I messaggi possono anche andare persi a causa della congestione della rete o dei limiti di capacità. Usa il protocollo TCP se la distribuzione affidabile è importante per il tuo caso d'uso.

Per rilevare e rispondere ai problemi di consegna, monitora la `SyslogMessagesDropped` metrica in. CloudWatch La `Reason` dimensione indica il motivo per cui i messaggi sono stati eliminati in modo da poter intraprendere azioni correttive. Per ulteriori informazioni, consulta the section called "Monitoraggio".

Quote e limiti

Limite	Valore	Note
Dimensione massima del messaggio (TCP)	64 KB	I messaggi syslog standard sono in genere ben al di sotto di questo limite. Se hai un caso d'uso che richiede messaggi più grandi, contatta AWS l'assistenza.
Dimensione massima dei messaggi (UDP)	8 KB	I messaggi syslog standard sono in genere ben al di sotto di questo limite. Se hai un caso d'uso che richiede messaggi più grandi, contatta AWS l'assistenza.

Limite	Valore	Note
Throughput di ingestione	Condiviso con `PutLogEvents`	L'inserimento di Syslog viene calcolato sulla `PutLogEvents` quota del tuo account (5.000 richieste al secondo per account per regione per impostazione predefinita).

La quota `PutLogEvents` è modificabile. Se il traffico syslog richiede un throughput più elevato, richiedi un aumento della quota tramite Service Quotas. https://docs.aws.amazon.com/servicequotas/latest/userguide/request-quota-increase.html

Configurazione dell'ingestione di syslog

Questa sezione illustra i passaggi per configurare l'ingestione di syslog in Logs. CloudWatch Creerai un endpoint VPC per il servizio syslog, un gruppo di log per ricevere i messaggi, una politica delle risorse per autorizzare il servizio syslog e una configurazione syslog che indirizza il traffico dall'endpoint VPC al tuo gruppo di log.

Puoi eseguire tutti questi passaggi utilizzando la Console di gestione, o gli SDK. AWS AWS CLI AWS Le seguenti istruzioni forniscono sia la console che gli AWS CLI esempi.

Prerequisiti

L'identità IAM (utente o ruolo) che usi per configurare l'ingestione di syslog deve disporre delle autorizzazioni per creare endpoint VPC, gruppi di log, policy di risorse e configurazioni syslog. La seguente policy di esempio mostra le autorizzazioni minime richieste:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:PutResourcePolicy",
        "logs:DeleteResourcePolicy",
        "logs:PutSyslogConfiguration",
```

```
        "logs:ListSyslogConfigurations",
        "logs:DeleteSyslogConfiguration"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVpc",
        "ec2:ModifyVpcAttribute",
        "ec2:CreateSubnet",
        "ec2:CreateSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateVpcEndpoint",
        "ec2:ModifyVpcEndpoint",
        "ec2:DescribeVpcEndpoints"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> Se disponi già di un VPC, di una sottorete e di un gruppo di sicurezza, sono necessari solo i e i `ec2:CreateVpcEndpoint` `ec2:DescribeVpcEndpoints` permessi per le azioni EC2. `ec2:ModifyVpcEndpoint`

Fase 1: Creare o identificare un VPC

È necessario un VPC raggiungibile dalla rete locale (tramite VPN o Direct Connect) dove risiedono i dispositivi che generano syslog. Se hai già un VPC collegato al tuo data center, salta questo passaggio e usa gli ID VPC e subnet esistenti.

Console

Per creare un VPC (console)

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione, scegli I tuoi VPC, quindi scegli Crea VPC.

3. Per Risorse da creare scegli Solo VPC.
4. Per il blocco CIDR IPv4, inserisci `10.0.0.0/16` (o un CIDR che non sia in conflitto con la rete locale).
5. Seleziona Crea VPC.
6. Seleziona il VPC appena creato, scegli Azioni, Modifica impostazioni VPC. Abilita sia la risoluzione DNS che i nomi host DNS, quindi scegli Salva.
7. Crea una sottorete per l'endpoint VPC:
   a. Nel pannello di navigazione, scegli Sottoreti, quindi scegli Crea sottorete.
   b. Per VPC ID, seleziona il VPC che hai creato.
   c. Per Zona di disponibilità, scegli una zona di disponibilità.
   d. Per il blocco CIDR della sottorete IPv4, immettere. `10.0.1.0/24`
   e. Scegliere Create subnet (Crea sottorete).

AWS CLI

```
REGION=us-east-1

# Create VPC
VPC_ID=$(aws ec2 create-vpc \
  --cidr-block 10.0.0.0/16 \
  --region $REGION \
  --query 'Vpc.VpcId' --output text)

aws ec2 modify-vpc-attribute --vpc-id $VPC_ID --enable-dns-support --region $REGION
aws ec2 modify-vpc-attribute --vpc-id $VPC_ID --enable-dns-hostnames --region
 $REGION

# Create a subnet for the VPC endpoint
SUBNET_ID=$(aws ec2 create-subnet \
  --vpc-id $VPC_ID \
  --cidr-block 10.0.1.0/24 \
  --availability-zone ${REGION}a \
  --region $REGION \
  --query 'Subnet.SubnetId' --output text)

echo "VPC: $VPC_ID, Subnet: $SUBNET_ID"
```

> **Note**
>
> L'endpoint VPC crea un'interfaccia di rete elastica (ENI) con un IP privato nella sottorete. I dispositivi locali raggiungono questo IP tramite la connessione VPN o Direct Connect. Assicurati che il routing di rete consenta il traffico dai tuoi dispositivi alla sottorete CIDR.

Fase 2: creazione di un gruppo di sicurezza

Crea un gruppo di sicurezza per l'endpoint VPC che consenta il traffico syslog in entrata dal tuo VPC. Questo controlla quali risorse possono inviare syslog all'endpoint.

Console

Per creare un gruppo di sicurezza (console)

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel riquadro di navigazione, scegli Gruppi di sicurezza, quindi scegli Crea gruppo di sicurezza.
3. In Security group name (Nome gruppo di sicurezza) immettere `syslog-vpce-sg`.
4. Per Descrizione, inserisci `Allow syslog traffic to VPC endpoint`.
5. Per VPC, seleziona il VPC che hai creato o identificato nel passaggio 1.
6. Nella sezione Regole in entrata, scegli Aggiungi regola e aggiungi le seguenti regole:
   - Regola 1: Tipo = TCP personalizzato, Intervallo di porte =`6514`, Source = `10.0.0.0/16` (il tuo VPC CIDR)
   - Regola 2: Tipo = TCP personalizzato, Intervallo di porte =, Origine = `1514` `10.0.0.0/16`
   - Regola 3: Tipo = UDP personalizzato, Intervallo di porte =`514`, Fonte = `10.0.0.0/16`
7. Scegliere Create Security Group (Crea gruppo di sicurezza).

AWS CLI

```
VPCE_SG_ID=$(aws ec2 create-security-group \
  --group-name syslog-vpce-sg \
  --description "Allow syslog traffic to VPC endpoint" \
  --vpc-id $VPC_ID \
```

```
  --region $REGION \
  --query 'GroupId' --output text)

# Allow TCP+TLS (port 6514), TCP plaintext (port 1514), and UDP (port 514)
aws ec2 authorize-security-group-ingress --group-id $VPCE_SG_ID \
  --protocol tcp --port 6514 --cidr 10.0.0.0/16 --region $REGION
aws ec2 authorize-security-group-ingress --group-id $VPCE_SG_ID \
  --protocol tcp --port 1514 --cidr 10.0.0.0/16 --region $REGION
aws ec2 authorize-security-group-ingress --group-id $VPCE_SG_ID \
  --protocol udp --port 514 --cidr 10.0.0.0/16 --region $REGION
```

> **Tip**
>
> Se prevedi di utilizzare un solo protocollo (ad esempio, TCP+TLS sulla porta 6514), devi solo aprire quella porta nel gruppo di sicurezza.

Fase 3: creazione dell'endpoint VPC

Crea un endpoint VPC di interfaccia che punti al servizio syslog. AWS PrivateLink Ciò fornisce al VPC un punto di accesso privato al servizio CloudWatch Logs syslog.

> **Note**
>
> È possibile specificare più ID di sottorete in diverse zone di disponibilità durante la creazione dell'endpoint. L'endpoint crea un ENI in ogni sottorete, fornendo una maggiore disponibilità senza la necessità di endpoint VPC separati per zona di disponibilità. È sufficiente un singolo endpoint VPC con sottoreti in più zone di disponibilità.

Console

Per creare l'endpoint VPC (console)

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel riquadro di navigazione, scegli Endpoints, quindi scegli Crea endpoint.
3. Per il tag Nome, inserisci un nome per l'endpoint (ad esempio,). `syslog-vpce`
4. Per Service category (Categoria servizio), scegli AWS services.

5. Nel campo di ricerca Servizi, inserisci `syslog-logs` e seleziona il servizio`com.amazonaws.`*`Region`*`.syslog-logs`.
6. Per VPC, seleziona il VPC che hai creato o identificato nel passaggio 1.
7. Nella sezione Subnet, seleziona una o più zone di disponibilità e scegli le sottoreti in cui desideri creare le interfacce di rete degli endpoint.
8. Per i gruppi di sicurezza, selezionate il gruppo di sicurezza creato nel passaggio 2 (). `syslog-vpce-sg`
9. (Facoltativo) Se desideri limitare il traffico consentito attraverso l'endpoint, configura una policy per gli endpoint VPC. Per ulteriori informazioni, consulta the section called "Policy di endpoint VPC".
10. Seleziona Crea endpoint.
11. Dopo che lo stato dell'endpoint è passato a Disponibile, seleziona l'endpoint e annota il valore dei nomi DNS. Questo è l'indirizzo a cui verranno inviati i dispositivi syslog.

AWS CLI

```
VPCE_ID=$(aws ec2 create-vpc-endpoint \
  --vpc-id $VPC_ID \
  --service-name com.amazonaws.${REGION}.syslog-logs \
  --vpc-endpoint-type Interface \
  --subnet-ids $SUBNET_ID \
  --security-group-ids $VPCE_SG_ID \
  --region $REGION \
  --query 'VpcEndpoint.VpcEndpointId' --output text)

echo "VPC Endpoint: $VPCE_ID"
```

Attendi che l'endpoint diventi disponibile (circa 60 secondi), quindi recupera il nome DNS:

```
VPCE_DNS=$(aws ec2 describe-vpc-endpoints --vpc-endpoint-ids $VPCE_ID \
  --region $REGION --query 'VpcEndpoints[0].DnsEntries[0].DnsName' --output text)

echo "Endpoint DNS: $VPCE_DNS"
```

Salva il VPCE_DNS valore: configurerai i tuoi dispositivi syslog per l'invio a questo indirizzo.

Fase 4: Creare un gruppo di log

Crea il gruppo di log CloudWatch Logs in cui verranno recapitati i tuoi messaggi syslog. È possibile utilizzare qualsiasi nome di gruppo di log. Si consiglia di utilizzare un `/syslog/` prefisso per motivi di chiarezza organizzativa.

Console

Per creare un gruppo di log (console)

1. Apri la console CloudWatch Logs all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, in Gestione dei registri, scegli Gruppi di log.
3. Scegliere Crea gruppo di log.
4. Per il nome del gruppo di log, inserisci`/syslog/my-devices`.
5. (Facoltativo) Configurate le impostazioni di conservazione e la crittografia in base alle esigenze.
6. Scegli Create (Crea).

AWS CLI

```
aws logs create-log-group \
  --log-group-name /syslog/my-devices \
  --region $REGION
```

Non è necessario creare un flusso di log. Il servizio syslog crea automaticamente un flusso di log denominato *`VPCE_ID`*`_Syslog_`*`Region`* (ad esempio`vpce-0abc123def456_Syslog_us-east-1`) quando viene recapitato il primo messaggio.

Fase 5: Aggiungere una politica delle risorse

Il servizio CloudWatch Logs syslog scrive nel gruppo di log utilizzando Service Principal. `syslog.logs.amazonaws.com` È necessario concedergli l'autorizzazione tramite una politica delle risorse sul gruppo di log. La `aws:SourceArn` condizione garantisce che solo il traffico proveniente dal tuo endpoint VPC specifico possa scrivere su questo gruppo di log.

```
ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)
```

```
aws logs put-resource-policy \
  --policy-name syslog-ingestion \
  --policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "syslog.logs.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "arn:aws:logs:'$REGION':'$ACCOUNT_ID':log-group:/syslog/my-
devices:*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "'$ACCOUNT_ID'"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:ec2:'$REGION':'$ACCOUNT_ID':vpc-
endpoint/'$VPCE_ID'"
        }
      }
    }
  ]
}' \
  --region $REGION
```

Le condizioni contenute nella politica delle risorse forniscono le seguenti protezioni:

- `aws:SourceAccount`— Previene gli attacchi confusi dei vice agenti. Viene accettato solo il traffico del tuo account.
- `aws:SourceArn`— Ambita l'accesso a un endpoint VPC specifico. Se disponi di più endpoint VPC, aggiungili ciascuno come ARN separato nella condizione.

Per consentire a più endpoint VPC di scrivere sullo stesso gruppo di log, usa l'operatore `ArnLike` condition con un wildcard:

```
"ArnLike": {
```

```
    "aws:SourceArn": "arn:aws:ec2:us-east-1:123456789012:vpc-endpoint/*"
}
```

Fase 6: Creare la configurazione syslog

Questo passaggio indica al servizio CloudWatch Logs syslog che il traffico in arrivo dall'endpoint VPC deve essere indirizzato al gruppo di log. Senza questa configurazione, il traffico proveniente dall'endpoint viene rifiutato.

Console

Per creare la configurazione syslog (console)

1. Apri la console CloudWatch Logs all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione, in Gestione dei registri, scegli Gruppi di log.
3. Scegli il gruppo di log che hai creato nel passaggio 4 (ad esempio,`/syslog/my-devices`).
4. Nei dettagli del gruppo di log, individua la sezione Syslog Ingestion.
5. Scegli Configura.
6. Dal menu a discesa VPC endpoint, seleziona l'endpoint VPC che hai creato nel passaggio 3.
7. Scegli Create (Crea).

AWS CLI

```
aws logs put-syslog-configuration \
  --log-group-identifier /syslog/my-devices \
  --vpc-endpoint-id $VPCE_ID \
  --region $REGION
```

Verifica la configurazione:

```
aws logs list-syslog-configurations \
  --log-group-identifier /syslog/my-devices \
  --region $REGION
```

La tua pipeline di ingestione di syslog è ora attiva. Tutti i messaggi syslog inviati all'endpoint VPC verranno recapitati al gruppo di log. `/syslog/my-devices`

Fase 7: Verifica della consegna e dell'estrazione sul campo

Invia un messaggio di prova da qualsiasi host o dispositivo EC2 in grado di raggiungere l'endpoint VPC, quindi CloudWatch utilizza Log Analytics per verificare che il messaggio sia stato recapitato e che i campi strutturati siano stati estratti correttamente. I messaggi vengono in genere visualizzati entro 10-20 secondi.

Invia un messaggio di prova (testo semplice TCP):

```
echo "<134>1 $(date -u +%Y-%m-%dT%H:%M:%SZ) myhost myapp 1234 - - Hello from syslog" |
 \
  nc $VPCE_DNS 1514
```

Verifica la consegna e i campi estratti:

Console

Per verificare la consegna utilizzando Log Analytics (console)

1. Apri la console CloudWatch Logs all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, in Logs, scegli Log Analytics.
3. Nel selettore del gruppo di log, scegli il tuo gruppo di log (ad esempio,`/syslog/my-devices`).
4. Immettete la seguente interrogazione e scegliete Esegui interrogazione:

   ```
   fields @timestamp, facility, severity, hostname, appName, procId, message
   | sort @timestamp desc
   | limit 10
   ```

5. Verifica che venga visualizzato il messaggio di prova e che i campi estratti siano compilati correttamente. Per il messaggio di prova riportato sopra, dovresti vedere `facility` `severity` = `local0info`, `hostname` = `myhostmyapp`, `appName` = e `procId` =. `1234`

AWS CLI

Avvia una query di Log Analytics per verificare la consegna e l'estrazione dei campi:

```
QUERY_ID=$(aws logs start-query \
```

```
  --log-group-name /syslog/my-devices \
  --start-time $(date -d '5 minutes ago' +%s 2>/dev/null || echo $(date -v-5M +%s))
 \
  --end-time $(date +%s) \
  --query-string 'fields @timestamp, facility, severity, hostname, appName, procId,
 message | sort @timestamp desc | limit 10' \
  --region $REGION \
  --query 'queryId' --output text)

# Wait a few seconds for the query to complete, then retrieve results
aws logs get-query-results \
  --query-id $QUERY_ID \
  --region $REGION
```

Verifica che venga visualizzato il messaggio di prova e che i campi estratti siano compilati correttamente. Per il messaggio di prova riportato sopra, dovresti vedere `facility` `severity` = `local0info`, `hostname` = `myhostmyapp`, `appName` = e `procId` =. `1234`

Configurazione dei dispositivi syslog

Dopo aver configurato l'ingestione di syslog, configura le tue fonti syslog per inviare messaggi al nome DNS dell'endpoint VPC. I tuoi dispositivi raggiungono l'endpoint tramite la connessione VPN o Direct Connect al VPC.

> **Note**
>
> Le seguenti configurazioni sono fornite a titolo di esempio. Per le opzioni di configurazione complete e le migliori pratiche, fate riferimento alla documentazione ufficiale per l'implementazione di syslog (rsyslog, syslog-ng o il fornitore del dispositivo).

Negli esempi seguenti, sostituiscilo *VPCE_DNS* con il nome DNS dell'endpoint VPC della procedura di configurazione (ad esempio,). `vpce-0abc123def456.syslog-logs.us-east-1.vpce.amazonaws.com`

rsyslog

rsyslog è il demone syslog predefinito nella maggior parte delle distribuzioni Linux. Aggiungere la seguente `/etc/rsyslog.d/50-cwl-syslog.conf` configurazione a.

TCP + TLS:

```
# Forward all logs to CloudWatch Logs via TCP+TLS
action(type="omfwd"
  target="VPCE_DNS"
  port="6514"
  protocol="tcp"
  StreamDriver="gtls"
  StreamDriverMode="1"
  StreamDriverAuthMode="x509/certvalid"
  template="RSYSLOG_SyslogProtocol23Format"
  queue.type="LinkedList"
  queue.filename="cwl_fwd"
  queue.saveOnShutdown="on"
  action.resumeRetryCount="-1")
```

Testo semplice TCP:

```
# Forward all logs to CloudWatch Logs via plaintext TCP
action(type="omfwd"
  target="VPCE_DNS"
  port="1514"
  protocol="tcp"
  template="RSYSLOG_SyslogProtocol23Format"
  queue.type="LinkedList"
  queue.filename="cwl_fwd"
  queue.saveOnShutdown="on"
  action.resumeRetryCount="-1")
```

UDP:

```
# Forward all logs to CloudWatch Logs via UDP
action(type="omfwd"
  target="VPCE_DNS"
  port="514"
  protocol="udp"
  template="RSYSLOG_SyslogProtocol23Format")
```

Dopo aver aggiunto la configurazione, riavvia rsyslog:

```
sudo systemctl restart rsyslog
```

> **Note**
>
> Le `queue.saveOnShutdown="on"` direttive `queue.type="LinkedList"` and abilitano l'accodamento assistito da disco. Se la connessione all'endpoint VPC viene temporaneamente interrotta, rsyslog memorizza i messaggi nel buffer locale e riprova il recapito.

Filtraggio per struttura o gravità:

Per inviare solo messaggi specifici, aggiungi un filtro prima dell'azione:

```
# Send only auth messages via TCP+TLS
auth,authpriv.* action(type="omfwd"
  target="VPCE_DNS"
  port="6514"
  protocol="tcp"
  StreamDriver="gtls"
  StreamDriverMode="1"
  StreamDriverAuthMode="x509/certvalid"
  template="RSYSLOG_SyslogProtocol23Format")
```

syslog-ng

syslog-ng è comunemente usato su dispositivi di rete e sistemi Linux aziendali. Aggiungere la seguente configurazione a. `/etc/syslog-ng/conf.d/cwl-syslog.conf`

TCP + TLS:

```
destination d_cwl_syslog {
  network("VPCE_DNS"
    port(6514)
    transport("tls")
    tls(
      peer-verify(required-trusted)
      ca-dir("/etc/ssl/certs")
    )
  );
};

log {
```

```
  source(s_sys);
  destination(d_cwl_syslog);
};
```

Testo semplice TCP:

```
destination d_cwl_syslog {
  network("VPCE_DNS"
    port(1514)
    transport("tcp")
  );
};

log {
  source(s_sys);
  destination(d_cwl_syslog);
};
```

UDP:

```
destination d_cwl_syslog {
  network("VPCE_DNS"
    port(514)
    transport("udp")
  );
};

log {
  source(s_sys);
  destination(d_cwl_syslog);
};
```

Dopo aver aggiunto la configurazione, riavvia syslog-ng:

```
sudo systemctl restart syslog-ng
```

Validazione rapida

Utilizza i seguenti comandi da qualsiasi host in grado di raggiungere l'endpoint VPC per verificare la connettività.

Testo semplice TCP (porta 1514):

```
echo "<134>1 $(date -u +%Y-%m-%dT%H:%M:%SZ) myhost myapp 1234 - - Hello from syslog" |
 \
  nc VPCE_DNS 1514
```

TCP + TLS (porta 6514):

```
echo "<134>1 $(date -u +%Y-%m-%dT%H:%M:%SZ) myhost myapp 1234 - - Hello TLS syslog" | \
  openssl s_client -connect VPCE_DNS:6514 -quiet -no_ign_eof
```

UDP (porta 514):

```
echo "<134>1 $(date -u +%Y-%m-%dT%H:%M:%SZ) myhost myapp 1234 - - Hello UDP syslog" | \
  nc -u -w1 VPCE_DNS 514
```

Dopo aver inviato i messaggi di prova, verifica la consegna nei CloudWatch registri (di solito entro 10-20 secondi):

```
aws logs filter-log-events \
  --log-group-name /syslog/my-devices \
  --start-time $(date -d '5 minutes ago' +%s000 2>/dev/null || echo $(($(date -v-5M +
%s) * 1000))) \
  --region $REGION
```

Apparecchiature di rete

La maggior parte dei firewall, dei router e degli switch supporta la configurazione di un server syslog esterno specificando un indirizzo IP e una porta di destinazione. Utilizza l'indirizzo IP privato dell'endpoint VPC ENI (visibile nella console Amazon VPC sotto Endpoints) e una delle porte supportate (6514, 1514 o 514).

Consulta la documentazione del tuo dispositivo per la sintassi di configurazione specifica. Le impostazioni principali da configurare sono:

- Indirizzo server: il nome DNS dell'endpoint VPC o l'indirizzo IP privato ENI.
- Porta: 6514 (TCP+TLS), 1514 (TCP) o 514 (UDP).
- Protocollo: TCP (consigliato) o UDP.
- Formato: RFC 5424 (preferito) o RFC 3164 (precedente). Entrambi sono supportati.

Attendibilità dei certificati TLS

La connessione TLS sulla porta 6514 utilizza un certificato AWS gestito emesso da Amazon Trust Services. La maggior parte dei sistemi operativi e dei daemon syslog si fida automaticamente di questo certificato perché i certificati root di Amazon Trust Services sono inclusi negli archivi trust standard di CA.

Se il tuo dispositivo non considera attendibile il certificato per impostazione predefinita, scarica i certificati root di Amazon Trust Services da https://www.amazontrust.com/repository/e aggiungili al CA Trust Store del dispositivo.

Monitoraggio dell'ingestione di syslog

Il servizio di immissione syslog pubblica le metriche nel namespace. CloudWatch `AWS/Logs` Queste metriche offrono visibilità sulla pipeline di importazione di syslog: cosa è stato ricevuto, cosa è stato eliminato e perché.

Per visualizzare queste metriche nella CloudWatch console, accedi a Metriche > Tutte le metriche > e filtra in base ai nomi delle metriche elencati di seguito. AWS/Logs

Metriche Syslog

Metrica	Dimensioni	Description
`SyslogMessagesReceived`	`LogGroupName`	Il numero di messaggi syslog inseriti con successo nel tuo gruppo di log.
`SyslogMessagesDropped`	`LogGroupName` , `Reason`	Il numero di messaggi syslog che non è stato possibile recapitare. Per informazioni dettagliate, vedi the section called "Elimina i motivi".
`SyslogConnectionsRejected`	`Reason`	Il numero di connessioni TCP rifiutate.
`SyslogConnectionsEstablished`	—	Il numero di connessioni TCP accettate correttamente.

Metrica	Dimensioni	Description
`SyslogConnectionsClosed`	—	Il numero di connessioni TCP chiuse.

Elimina i motivi

La `SyslogMessagesDropped` metrica include una `Reason` dimensione che indica il motivo per cui i messaggi sono stati eliminati.

Motivo	Description
`MessageRateLimitExceeded`	La `PutLogEvents` quota del tuo account è stata superata. Prendi in considerazione la possibilità di richiedere un aumento della quota.
`MessageSizeExceeded`	Un datagramma UDP ha superato la dimensione massima del messaggio.
`ServiceUnavailable`	Capacità interna o limite di velocità superato. Si tratta in genere di un fenomeno transitorio.
`ResourceNotFound`	Il gruppo di log di destinazione non esiste. Verificare che il gruppo di log non sia stato eliminato.
`AccessDenied`	La politica delle risorse sul gruppo di log non concede l'accesso al servizio syslog. Verifica che la politica delle risorse sia corretta.
`VpcePolicyDenied`	La politica degli endpoint VPC ha rifiutato la richiesta. Rivedi la tua policy sugli endpoint VPC.
`InternalError`	Si è verificato un errore interno imprevisto. Se il problema persiste, contatta l' AWS assistenza.

Motivi del rifiuto della connessione

La `SyslogConnectionsRejected` metrica include una `Reason` dimensione.

Motivo	Description
`VpcePolicyDenied`	La politica degli endpoint VPC ha negato la connessione.
`ServiceUnavailable`	Connessione interrotta a causa di un errore interno.

Allarmi raccomandati

Consigliamo di creare CloudWatch allarmi nelle seguenti condizioni per rilevare tempestivamente i problemi:

Allarme	Condizione	Azione suggerita
Messaggi eliminati	`SyslogMessagesDropped` > 0 per 5 minuti	Analizzate la `Reason` dimensione per determinarne la causa.
Accesso negato	`SyslogMessagesDropped` con `Reason=AccessDenied`	Verifica che la politica delle risorse nel gruppo di log sia configurata correttamente.
Gruppo di log mancante	`SyslogMessagesDropped` con `Reason=ResourceNotFound`	Verificare che il gruppo di log esista e non sia stato eliminato .
Nessun messaggio ricevuto	`SyslogMessagesReceived` = 0 per 15 minuti (quando normalmente è superiore a 0)	Verifica che i dispositivi stiano ancora inviando e che la connettività di rete all'endpoint VPC sia intatta.

Per informazioni sulla creazione di CloudWatch allarmi, consulta CloudWatch Creazione di allarmi nella Guida per l'utente. CloudWatch

Visualizzazione delle metriche syslog sulla dashboard automatica

CloudWatch fornisce una dashboard automatica per CloudWatch i log che include una sezione Syslog Ingestion. Questa dashboard viene visualizzata automaticamente quando il tuo account pubblica le metriche syslog: non è richiesta alcuna configurazione manuale. La dashboard include un grafico per ogni metrica descritta in. the section called "Metriche Syslog" Ogni grafico mostra la frequenza delle metriche suddivisa per le dimensioni pertinenti, ad esempio il gruppo di log o il motivo.

Accesso al pannello di controllo

Per visualizzare la dashboard automatica di syslog:

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, scegli Dashboard, quindi scegli Dashboard automatici.
3. Scegli la dashboard dei CloudWatch log.
4. Scorri fino alla sezione Syslog Ingestion.

Query di esempio per i dati syslog

Le seguenti query di CloudWatch Logs Insights illustrano i modelli di analisi comuni per i dati syslog. Usano i campi strutturati che CloudWatch Logs estrae automaticamente dai messaggi RFC 5424, RFC 3164 e Cisco. FTD/ASA Per l'elenco completo dei campi estratti, vedere. the section called "Formati syslog supportati"

Queste interrogazioni sono disponibili anche nella CloudWatch console. In CloudWatch Logs Insights, scegli Queries, quindi espandi la categoria Syslog.

Sicurezza e monitoraggio degli accessi

Eventi Sudo e di escalation dei privilegi

La seguente query elenca le recenti attività di escalation dei privilegi sudo sugli host Linux. Filtra in base al nome dell'`sudo`applicazione e funziona con i messaggi RFC 3164 e RFC 5424.

```
filter appName="sudo"
| fields @timestamp, hostname, message
| sort @timestamp desc
```

```
| limit 50
```

I 50 principali host con autenticazione SSH fallita

La seguente query conta i tentativi di autenticazione SSH falliti per ogni host. Corrisponde ai messaggi sshd che contengono «Password fallita» o «Chiave pubblica fallita». Usalo per identificare gli host bersaglio di attacchi di forza bruta.

```
filter appName="sshd" and (message like /Failed password/ or message like /Failed
 publickey/)
| stats count(*) as failedAttempts by hostname
| sort failedAttempts desc
| limit 50
```

Visibilità operativa

Volume di log per host nel tempo

La seguente query conta i messaggi per ogni host a intervalli di 5 minuti. Usalo per misurare l'attività della flotta, identificare i picchi di volume o rilevare gli host che hanno interrotto l'invio. Questa query funziona con tutti i tipi di formato syslog.

```
stats count(*) as messages by hostname, bin(5m)
| sort messages desc
```

Volume dei messaggi per struttura e gravità

La seguente interrogazione conta i messaggi raggruppati per struttura e livello di gravità. Usalo per capire quali sottosistemi generano la maggior parte del traffico di log e con quale livello di gravità.

```
stats count(*) as messageCount by facility, severity
| sort messageCount desc
```

High-severity messaggi per applicazione

La seguente query filtra i messaggi ad alta priorità (SeverityCode 0—3, che corrisponde a emerg, alert, crit ed err) e li conta per applicazione. Utilizzatelo per identificare quali applicazioni generano il maggior numero di errori.

```
filter severityCode<=3
```

```
| stats count(*) as errorCount by appName
| sort errorCount desc
```

Warning-and-above tendenza nel tempo

La seguente query conta i messaggi con gravità dell'avviso o superiore (SeverityCode 0—4) a intervalli di 5 minuti. Usalo con la funzionalità di visualizzazione di Logs Insights per tracciare i picchi di errore nel tempo.

```
filter severityCode<=4
| stats count(*) as events by bin(5m)
| sort @timestamp
```

Interrogazioni su firewall e dispositivi di rete

Connessioni negate da Cisco (ACL 106023)

La seguente query riguarda i firewall Cisco ASA e FTD. Conta ACL-denied le connessioni (ID messaggio Cisco 106023) raggruppate per dispositivo, aiutando a identificare i dispositivi con il traffico più negato.

```
filter messageId="106023"
| stats count(*) as denies by deviceId
| sort denies desc
| limit 20
```

Eventi Cisco ASA/FTD per ID messaggio

La seguente query riguarda i firewall Cisco FTD e ASA. Conta gli eventi raggruppati per ID e gravità dei messaggi Cisco per aiutarti a identificare i tipi di eventi più frequenti nella tua flotta di firewall.

```
filter ispresent(messageId)
| stats count(*) as eventCount by messageId, severity
| sort eventCount desc
```

Il firewall ha eliminato i pacchetti in base all'IP di origine

La seguente query analizza i messaggi DROP del firewall (ad esempio, provenienti da iptables o nftables) per contare i pacchetti persi in base all'indirizzo IP di origine. Usalo per identificare le principali fonti di traffico bloccato.

```
filter message like /DROP/
| parse message /SRC=(?<srcAddr>\S+)/
| stats count(*) as drops by srcAddr
| sort drops desc
| limit 20
```

Policy degli endpoint VPC per syslog

Puoi allegare una policy di endpoint VPC al tuo endpoint VPC syslog per controllare il traffico consentito attraverso di esso. Ciò fornisce un ulteriore livello di controllo degli accessi oltre ai gruppi di sicurezza e alle politiche delle risorse.

Per impostazione predefinita, un endpoint VPC ha una politica di accesso completo che consente tutto il traffico. Puoi limitarlo allegando una policy personalizzata. Per ulteriori informazioni sulle policy degli endpoint VPC, consulta Controllare l'accesso utilizzando le policy degli endpoint nella Guida.AWS PrivateLink

> **Important**
>
> Il traffico Syslog arriva come principale anonimo (nessuna identità IAM in rete). Le policy devono utilizzare `"Principal": "*"` e definire l'accesso utilizzando chiavi di condizione Action, Resource o VPC. Principal-scoped le politiche (ad esempio, le politiche che consentono solo un ruolo IAM specifico) negheranno tutto il traffico syslog.

Chiavi di condizione supportate

Le seguenti chiavi di condizione sono supportate nelle politiche degli endpoint VPC per syslog:

Chiave di condizione	Description	Valore di esempio
`aws:SourceVpc`	Il VPC da cui proviene il traffico.	`vpc-111bbb22`
`aws:SourceVpce`	L'endpoint VPC attraverso il quale arriva il traffico.	`vpce-0abc123def456`
`aws:VpcSourceIp`	L'indirizzo IP privato del mittente all'interno del VPC.	`10.0.1.50`

Chiave di condizione	Description	Valore di esempio
`aws:ResourceAccount`	L'account proprietario del gruppo di log di destinazione.	`123456789012`
`aws:ResourceOrgID`	L'ID dell'organizzazione dell'account proprietario del gruppo di log di destinazione.	`o-aa111bb222`
`aws:ResourceOrgPaths`	Il percorso dell'unità organizzativa dell'account proprietario del gruppo di log di destinazione.	`o-aa111bb222/r-ab12/ou-ab12-11111111/*`

Le seguenti chiavi di condizione non sono supportate perché il traffico syslog non ha un'identità principale IAM:

- `aws:PrincipalArn`
- `aws:PrincipalAccount`
- `aws:PrincipalOrgID`
- `aws:PrincipalOrgPaths`
- `aws:PrincipalTag/*`

Le policy che utilizzano chiavi di condizione non supportate comporteranno un rifiuto implicito, bloccando tutto il traffico syslog attraverso l'endpoint.

Limita a un gruppo di log specifico

La seguente politica consente il traffico syslog solo verso un gruppo di log specifico:

```
aws ec2 modify-vpc-endpoint \
  --vpc-endpoint-id $VPCE_ID \
  --policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "logs:PutLogEvents",
```

```
      "Resource": "arn:aws:logs:Region:account-id:log-group:/syslog/my-devices"
    }
  ]
}' \
  --region $REGION
```

Nega tutto il traffico

La seguente policy blocca tutto il traffico syslog attraverso l'endpoint. Ciò è utile per interrompere temporaneamente l'ingestione senza eliminare l'endpoint:

```
aws ec2 modify-vpc-endpoint \
  --vpc-endpoint-id $VPCE_ID \
  --policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "*",
      "Resource": "*"
    }
  ]
}' \
  --region $REGION
```

Ripristina la politica predefinita

Per ripristinare la politica di autorizzazione predefinita:

```
aws ec2 modify-vpc-endpoint \
  --vpc-endpoint-id $VPCE_ID \
  --reset-policy \
  --region $REGION
```

Propagazione delle politiche

Dopo aver creato o modificato una policy per gli endpoint VPC, possono essere necessari diversi minuti prima che la modifica abbia effetto. Durante questo periodo di propagazione, la politica precedente (o la politica predefinita) continua ad essere applicata.

Se una policy degli endpoint VPC nega il traffico syslog, vedrai i seguenti indicatori:

- Le connessioni TCP vengono chiuse immediatamente dopo la valutazione della policy.
- I datagrammi UDP vengono eliminati.
- Viene emessa la `SyslogMessagesDropped` metrica with. `Reason=VpcePolicyDenied`
- La `SyslogConnectionsRejected` metrica with `Reason=VpcePolicyDenied` viene emessa per le connessioni TCP.

Risoluzione dei problemi relativi all'ingestione di syslog

Questa sezione descrive i problemi più comuni relativi all'ingestione di syslog e come risolverli.

Messaggi che non compaiono nel gruppo di log

Se hai inviato messaggi syslog ma questi non compaiono nel tuo gruppo di log, controlla quanto segue nell'ordine:

1. Verifica che la configurazione syslog esista.

   Esegui `list-syslog-configurations` per confermare che l'endpoint VPC sia registrato nel gruppo di log:

   ```
   aws logs list-syslog-configurations \
     --log-group-identifier /syslog/my-devices \
     --region $REGION
   ```

   Se non viene restituita alcuna configurazione, creane una con. `put-syslog-configuration`
2. Verifica la politica delle risorse.

   Assicurati che la politica delle risorse sul gruppo di log `logs:CreateLogStream` conceda `logs:PutLogEvents` e verso `syslog.logs.amazonaws.com` e che la `aws:SourceArn` condizione corrisponda esattamente all'ARN dell'endpoint VPC:

   ```
   aws logs describe-resource-policies --region $REGION
   ```

   Controlla la `SyslogMessagesDropped` metrica con `Reason=AccessDenied` per confermare se il problema è l'autorizzazione.

3. Verifica il gruppo di sicurezza.

   Assicurati che il gruppo di sicurezza collegato all'endpoint VPC consenta il traffico in entrata sulla porta che stai utilizzando (6514, 1514 o 514) dal CIDR di origine:

   ```
   aws ec2 describe-security-groups \
     --group-ids $VPCE_SG_ID \
     --region $REGION \
     --query 'SecurityGroups[0].IpPermissions'
   ```

4. Verifica la connettività di rete.

   Dal dispositivo o da un host sulla stessa rete, verifica la connettività TCP all'endpoint:

   ```
   # TCP connectivity test
   nc -zv $VPCE_DNS 6514
   nc -zv $VPCE_DNS 1514
   ```

   Se la connessione viene rifiutata o scade, verifica il routing VPN/Direct Connect e che il dispositivo possa raggiungere la sottorete degli endpoint VPC.

5. Controlla la policy degli endpoint VPC.

   Se hai allegato una policy personalizzata per gli endpoint VPC, verifica che consenta le azioni richieste. Controlla la `SyslogMessagesDropped` metrica con. `Reason=VpcePolicyDenied`

Errori di handshake TLS

Se il client syslog non riesce a stabilire una connessione TLS sulla porta 6514:

- Attendibilità dei certificati: assicurati che CA Trust Store del tuo dispositivo includa i certificati root di Amazon Trust Services. La maggior parte dei sistemi operativi li include di default. Scaricali da https://www.amazontrust.com/repository/se necessario.
- Versione TLS: il servizio supporta TLS 1.2 e TLS 1.3. Assicurati che il tuo client syslog supporti almeno TLS 1.2.
- Verifica del nome host: se il client verifica il nome host del server, assicurati che sia configurato per la connessione utilizzando il nome DNS dell'endpoint VPC (non un indirizzo IP).

Verifica manualmente la connessione TLS con OpenSSL:

```
openssl s_client -connect $VPCE_DNS:6514 -brief
```

Una connessione riuscita mostra la versione TLS e la suite di crittografia. Se fallisce, il messaggio di errore indica la causa.

Messaggi UDP persi

UDP è un protocollo ottimale. Una certa perdita di messaggi è un comportamento previsto nelle seguenti condizioni:

- Congestione della rete tra il dispositivo e l'endpoint VPC.
- I messaggi superano la dimensione massima del datagramma UDP.
- Limiti temporanei di capacità di servizio (indicati da with). `SyslogMessagesDropped Reason=ServiceUnavailable`

Se è richiesta una distribuzione affidabile, utilizza invece TCP (porta 1514 o 6514). TCP fornisce la conferma della consegna e la ritrasmissione automatica.

Connessione rifiutata

Se il dispositivo segnala «connessione rifiutata» durante la connessione all'endpoint VPC:

- Verifica che l'endpoint VPC sia nello stato: `available`

  ```
  aws ec2 describe-vpc-endpoints --vpc-endpoint-ids $VPCE_ID \
    --region $REGION --query 'VpcEndpoints[0].State'
  ```

- Verifica di connetterti su una porta supportata (6514, 1514 o 514).
- Il routing Verify VPN o Direct Connect consenta il traffico dalla rete del dispositivo alla sottorete degli endpoint VPC.
- Verifica che il gruppo di sicurezza consenta il traffico in entrata sulla porta che stai utilizzando.

Messaggi interrotti a causa della limitazione della velocità

Se la `SyslogMessagesDropped` metrica lo mostra`Reason=MessageRateLimitExceeded`, il traffico syslog supera la quota del tuo account. `PutLogEvents`

Per risolvere questo problema:

1. Controlla la tua quota attuale nella console Service Quotas in Amazon CloudWatch Logs.
2. Richiedi un aumento della quota se il throughput di syslog lo richiede. Vedi Richiesta di aumento della quota nella Guida per l'utente di Service Quotas.

Usare Logs con un CloudWatch 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
AWS SDK per SAP ABAP	AWS SDK per SAP ABAP esempi di codice
AWS SDK per Swift	AWS SDK per Swift esempi di codice

Per esempi specifici per CloudWatch Logs, vedereEsempi di codice per l'utilizzo di CloudWatch Logs AWS SDK.

> **Esempio di disponibilità**
>
> Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Fornisci un feedback nella parte inferiore di questa pagina.

Gestione dei registri

CloudWatch Logs offre funzionalità avanzate di gestione dei log che consentono di organizzare, trasformare e analizzare i dati di registro in modo più efficace. Queste funzionalità includono la centralizzazione dei dati tra account e regioni diverse, il rilevamento automatico dei dati e la gestione degli schemi, la trasformazione dei log durante le acquisizionie l'analisi avanzata con sfaccettature per l'esplorazione interattiva dei log.

Argomenti

- Individuazione e gestione delle fonti di dati
- Funzionalità abilitate dalle fonti di dati
- Origini dati supportate

Individuazione e gestione delle fonti di dati

CloudWatch Logs rileva e classifica automaticamente i dati di registro in base alla fonte e al tipo di dati, semplificando la comprensione e la gestione dei log su larga scala. Questa funzionalità consente l'individuazione dello schema per sorgenti AWS vendute come Amazon VPC Flow Logs e Route 53 CloudTrail, oltre a strumenti di sicurezza di terze parti.

La console di gestione dei log offre una visualizzazione di alto livello dei log organizzati per origine e tipo di dati, anziché solo per gruppi di log. Questa organizzazione ti aiuta a:

- Visualizza i log classificati per AWS servizi, fonti di terze parti (come Okta o CrowdStrike) e fonti personalizzate
- Comprendi automaticamente lo schema e la struttura dei tuoi dati di registro
- Crea politiche di indicizzazione dei campi basate sui campi dello schema rilevati
- Gestisci i log in modo più efficiente su diverse fonti di dati
- Interroga i log da diverse fonti di dati

Quando abiliti la registrazione CloudWatch dei log per i AWS servizi supportati, CloudWatch Logs applica automaticamente lo schema appropriato ai tuoi log. Questa applicazione automatica dello schema aiuta a mantenere la coerenza e fornisce informazioni immediate sulla struttura dei log.

Che cos'è CloudWatch Logs Data Sources?

CloudWatch Logs Data Sources è una funzionalità che offre un nuovo modo di organizzare e classificare i dati dei log in base alla fonte che li genera. Sebbene CloudWatch Logs utilizzi tradizionalmente gruppi di log per organizzare i log, Data Sources offre un ulteriore livello di organizzazione che raggruppa i log in base al servizio e al tipo di registro di origine.

Come funzionano le fonti di dati

Le fonti di dati forniscono un'organizzazione dei log basata sui servizi e un'individuazione semplificata in tutta l'infrastruttura. AWS È possibile individuare facilmente i log di servizi specifici e filtrarli per tipo di registro senza dover conoscere i nomi o le strutture dei singoli gruppi di log.

Per le fonti di terze parti e, facoltativamente, per le fonti dei log delle applicazioni, Data Sources utilizza le CloudWatch pipeline per classificare i log. Quando configuri una pipeline per importare e trasformare i log, specifichi il nome e il tipo dell'origine dati. CloudWatch Logs classifica quindi automaticamente tutti i log elaborati dalla pipeline. Per ulteriori informazioni, consulta le CloudWatch pipeline nella Amazon CloudWatch User Guide.

Le fonti di dati classificano i log utilizzando due identificatori chiave:

- Nome origine dati: il AWS servizio, l'origine di terze parti o l'applicazione che genera i log (ad esempio, Route 53, Amazon VPC CloudTrail, Okta SSO o Falcon). CrowdStrike
- Tipo di origine dati: il tipo specifico di log generato da quel servizio.

Uno schema definisce la struttura dei dati di registro, inclusi i campi presenti e l'organizzazione delle informazioni. Un'unica fonte di dati può produrre più tipi di log con schemi e scopi diversi. Ad esempio, l'origine AWS CloudTrail dati ha due tipi: eventi di gestione (che tengono traccia delle operazioni del piano di controllo come la creazione o l'eliminazione di risorse) ed eventi di dati (che tengono traccia delle operazioni del piano dati come l'accesso agli oggetti S3). Ogni tipo ha uno schema diverso perché acquisiscono diversi tipi di informazioni.

Come iniziare

CloudWatch Logs classifica i log in fonti di dati in base alla loro origine. Il metodo dipende dal tipo di log con cui stai lavorando:

Servizio AWS log

I log dei file supportati Servizi AWS vengono raggruppati automaticamente per origine dati senza che sia richiesta alcuna configurazione. CloudWatch Logs riconosce questi registri e applica il nome e il tipo di origine dati appropriati in base al servizio di origine.

Third-party registri

Third-party i log richiedono pipeline per la categorizzazione delle fonti di dati. Quando configuri una pipeline per importare log da fonti di terze parti supportate come Microsoft Office 365, Okta o Palo Alto Networks CrowdStrike, specifichi il nome e il tipo dell'origine dati nella configurazione della pipeline. CloudWatch Logs classifica automaticamente tutti i log che la pipeline elabora utilizzando tali identificatori.

Le pipeline possono facoltativamente trasformare i log di terze parti in formato Open Cybersecurity Schema Framework (OCSF) per l'analisi standardizzata degli eventi di sicurezza. Quando la trasformazione OCSF è abilitata, il nome e il tipo dell'origine dati vengono determinati automaticamente in base alla mappatura dello schema OCSF. Senza la trasformazione OCSF, è possibile specificare il nome e il tipo dell'origine dati nella configurazione della pipeline.

Log di applicazioni

Per i registri delle applicazioni personalizzati, è possibile classificarli in base all'origine dati utilizzando uno di questi metodi:

- Tag del gruppo di log: aggiungi tag ai tuoi gruppi di log utilizzando le chiavi `cw:datasource:name` e `cw:datasource:type` specificando rispettivamente il nome e il tipo dell'origine dati per tutti i log inseriti nel gruppo di log. I valori dei tag possono contenere fino a 64 caratteri e contenere solo lettere minuscole, numeri e caratteri di sottolineatura. Devono iniziare con una lettera o un numero e non possono contenere caratteri di sottolineatura doppi (__).
- Configurazione della pipeline: configura le informazioni sull'origine dei dati tramite pipeline di elaborazione dei log durante l'acquisizione dei log delle applicazioni.

> **Note**
>
> I nomi delle fonti di dati non possono iniziare con «aws» o «amazon» per evitare conflitti con i log di servizio. AWS

Campi di sistema

CloudWatch I registri aggiungono automaticamente tre campi di sistema ai registri classificati in base all'origine dati. Questi campi fungono da sfaccettature predefinite:

- `@data_source_name`- Contiene il nome della fonte di dati o «Sconosciuto» se non è specificato
- `@data_source_type`- Contiene il tipo di fonte dei dati o «Sconosciuto» se non è specificato
- `@data_format`- Indica il formato dei dati di registro

Quando non è possibile determinare il nome o il tipo dell'origine dati, questi campi sono impostati su «Sconosciuto». Le fonti di dati con valori «Sconosciuti» sono ancora visibili nei facet e nella tabella delle fonti di dati in «Gestione dei log» in Console, e consentono di identificare i log non categorizzati e da quale gruppo di log provengono.

Il `@data_format` campo può contenere uno dei seguenti valori:

- `Default`- Registri ingeriti senza modifiche.
- `Custom`- Registri elaborati tramite processori di pipeline o registri inseriti in gruppi di log con tag di origine dati. name/type
- `OCSF-<version>`- Registri elaborati con processori OCSF (Open Cybersecurity Schema Framework) nelle pipeline.
- `AWS-OTEL-LOG-V<version>`- OpenTelemetry registri importati tramite l'endpoint OTLP. CloudWatch
- `AWS-OTEL-TRACE-V<version>`- OpenTelemetry tracce ingerite tramite l'endpoint OTLP. CloudWatch

Questi campi di sistema consentono di filtrare e interrogare i log in base alla fonte e al formato, semplificando l'utilizzo di log di origini e pipeline di elaborazione diverse.

Accesso alle origini dati

Console

Nella console CloudWatch Logs, utilizzi la scheda Log Management per accedere alle tue fonti di dati. CloudWatch Logs consolida automaticamente i dati di registro in base alle fonti e ai tipi di dati, scoprendo continuamente i nuovi dati acquisiti. Dall'elenco delle fonti di dati, puoi creare pipeline, definire indici e sfaccettature di campo.

AWS CLI

Usa il seguente comando per elencare fonti di dati e tipi di log distinti nel tuo account:

```
aws logs list-aggregate-log-group-summaries --group-by DATA_SOURCE_NAME_AND_TYPE
```

Relazione con i gruppi di log

Le fonti di dati integrano anziché sostituire i gruppi di log. I log continuano a essere archiviati in gruppi di log come in precedenza, ma ora vengono anche contrassegnati automaticamente con le informazioni sulla fonte dei dati. Questa doppia organizzazione ti consente di:

- Utilizza i gruppi di log per politiche granulari di controllo e conservazione degli accessi
- Utilizza fonti di dati per l'individuazione e l'analisi dei log basate sui servizi
- Interroga i log utilizzando entrambi i metodi organizzativi, a seconda delle esigenze

Le fonti di dati semplificano l'utilizzo dei log su larga scala fornendo una visualizzazione incentrata sui servizi dei dati di registro in tutta l'infrastruttura. AWS

Funzionalità abilitate dalle fonti di dati

Le fonti di dati consentono funzionalità avanzate di elaborazione e analisi dei log attraverso la scoperta sul campo e strutture di dati coerenti.

- Facet: i facet sono campi di log indicizzati che forniscono filtri e analisi interattivi senza scrivere query. CloudWatch I log creano automaticamente i facet per il nome e il tipo di origine dei dati ed è possibile creare policy di facet sui campi rilevati per accelerare la risoluzione dei problemi. I facet mostrano le distribuzioni e i conteggi dei valori in CloudWatch Logs Insights, semplificando l'identificazione dei modelli attraverso l'esplorazione «punta e scegli».
- Pipeline: crea pipeline di trasformazione che si applicano a tutti i log in base al nome e al tipo di origine dati specifici. Ciò consente di definire regole di elaborazione coerenti per i log provenienti dalla stessa fonte.
- Scoperta dei campi: CloudWatch Logs rileva automaticamente i campi e i relativi tipi di dati per ogni combinazione di nomi e tipi di origine dati in base ai processori di pipeline. Per i log AWS gestiti, le strutture dei campi sono predefinite. Per i log delle applicazioni, consigliamo di mantenere formati di log coerenti per massimizzare la compatibilità con strumenti di analisi come le tabelle Amazon S3 che richiedono strutture di campo ben definite.

Puoi visualizzare l'elenco completo dei campi e dei relativi tipi per qualsiasi fonte di dati utilizzando l'API: `GetLogFields`

```
aws logs get-log-fields --data-source-name <name>  --data-source-type <type>
```

L'individuazione e la coerenza dei campi consentono analisi e integrazioni avanzate, poiché gli strumenti esterni possono funzionare con strutture di campo prevedibili durante l'elaborazione dei dati di registro.

Origini dati supportate

CloudWatch Logs supporta un'ampia gamma di fonti di dati tra cui AWS servizi, strumenti IT e di sicurezza di terze parti e fonti di applicazioni personalizzate. Questa pagina fornisce un catalogo completo di tutte le fonti di dati supportate. Per informazioni su come vengono scoperte e gestite le fonti di dati, consultathe section called “Individuazione e gestione delle fonti di dati”.

Argomenti

- Supportata Servizi AWS per le fonti di dati
- Fonti di terze parti supportate per le fonti di dati
- Origini personalizzate

Supportata Servizi AWS per le fonti di dati

La tabella seguente elenca quelle Servizi AWS che vengono automaticamente classificate dai CloudWatch registri come fonti di dati:

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
amazon_api_gateway	access
amazon_bedrock_agentcore	browser_usage
amazon_bedrock_agentcore	code_interpreter_application
amazon_bedrock_agentcore	code_interpreter_usage

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
amazon_bedrock_agentcore	gateway_application
amazon_bedrock_agentcore	identity_workload_application
amazon_bedrock_agentcore	memory_application
amazon_bedrock_agentcore	online_evaluation_config
amazon_bedrock_agentcore	runtime_application
amazon_bedrock_agentcore	runtime_usage
amazon_bedrock_agents	application
amazon_bedrock_agents	event
amazon_bedrock_knowledge_bases	application
amazon_cloudfront	access
amazon_cloudfront	connection
amazon_cloudwatch	rum_app_monitor
amazon_cognito	user_pool
amazon_ec2	verified_access
amazon_eks	api_server
amazon_eks	audit
amazon_eks	authenticator
amazon_eks	controller_manager
amazon_eks	scheduler
amazon_elasticache	cluster

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
amazon_eventbridge	eventbus_error
amazon_eventbridge	eventbus_info
amazon_eventbridge	pipes_execution
amazon_interactive_video_service	chat
amazon_managed_prometheus	scraper
amazon_managed_prometheus	workspace
amazon_msk	broker
amazon_msk	connect
amazon_opensearch_service	pipeline
amazon_q_business	events
amazon_q_business	sync_job
amazon_q_connect	events
amazon_route53	global_resolver_query
amazon_route53	hosted_zones
amazon_route53	profiles_resolver_query
amazon_route53	resolver_query
amazon_sagemaker	workteam_activity
amazon_ses	ingress_endpoints
amazon_ses	rule_sets
amazon_ses	traffic_policy

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
amazon_vpc	flow
amazon_vpc	route_server_peer
amazon_vpc_lattice	access
amazon_vpc_lattice	resource_access
amazon_workmail	access_control
amazon_workmail	authentication
amazon_workmail	personal_access
amazon_workmail	workmail_access
amazon_workmail	workmail_availability
aws_b2b_data_interchange	execution
aws_backup	data_access
aws_backup	hypervisor
aws_clean_rooms	analysis
aws_client_vpn	connection
aws_client_vpn	event
aws_cloudtrail	data
aws_cloudtrail	management
aws_elemental_mediapackage	egress_access
aws_elemental_mediapackage	ingress_access
aws_elemental_mediatailor	ad_decision

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
aws_elemental_mediatailor	manifest
aws_elemental_mediatailor	transcode
aws_entity_resolution	id_mapping_workflow
aws_entity_resolution	matching_workflow
aws_iot_fleetwise	error
aws_mainframe_modernization	batch_job
aws_mainframe_modernization	config
aws_mainframe_modernization	console
aws_mainframe_modernization	dataset_import
aws_network_firewall	alert
aws_network_firewall	flow
aws_network_firewall	tls
aws_nlb	access
aws_pcs	job_completion
aws_pcs	scheduler
aws_security_hub	compliance_finding
aws_security_hub	data_security_finding
aws_security_hub	detection_finding
aws_security_hub	vulnerability_finding
aws_security_hub_cspm	asff_finding

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
aws_shield	protection_flow
aws_step_functions	express
aws_step_functions	standard
aws_transfer_family	server
aws_waf	access

Fonti di terze parti supportate per le fonti di dati

La tabella seguente elenca le fonti di terze parti che vengono automaticamente classificate da CloudWatch Logs come fonti di dati quando vengono importate tramite pipeline:

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
akamai_datastream_2	base_event
akamai_datastream_2	dns_activity
akamai_datastream_2	http_activity
box	account_change
box	authentication
box	detection_finding
box	entity_management
box	file_hosting_activity
box	group_management
box	user_inventory_info

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
broadcom_carbonblack	authentication
broadcom_carbonblack	detection_finding
broadcom_carbonblack	file_system_activity
broadcom_carbonblack	module_activity
broadcom_carbonblack	network_activity
broadcom_carbonblack	process_activity
broadcom_carbonblack	registry_key_activity
broadcom_carbonblack	registry_value_activity
broadcom_carbonblack	script_activity
checkpoint_nextgenerationfirewall	data_security_finding
checkpoint_nextgenerationfirewall	detection_finding
checkpoint_nextgenerationfirewall	entity_management
checkpoint_nextgenerationfirewall	network_activity
cisco_duo	authentication
cisco_duo	base_event
cisco_duo	device_inventory_info
cisco_duo	entity_management

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
cisco_duo	user_inventory_info
cisco_ftd	authentication
cisco_ftd	authorize_session
cisco_ftd	detection_finding
cisco_ftd	network_activity
cisco_ftd	tunnel_activity
cisco_ise	account_change
cisco_ise	api_activity
cisco_ise	application_error
cisco_ise	application_lifecycle
cisco_ise	authentication
cisco_ise	entity_management
cisco_ise	network_activity
cisco_ise	tunnel_activity
cisco_ise	web_resources_activity
cisco_meraki	api_activity
cisco_meraki	detection_finding
cisco_meraki	network_activity
cisco_umbrella	data_security_finding
cisco_umbrella	dns_activity

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
cisco_umbrella	entity_management
cisco_umbrella	network_activity
crowdstrike_falcon	detection_finding
crowdstrike_falcon	process_activity
drupal_core	application_lifecycle
drupal_core	authentication
drupal_core	entity_management
drupal_core	http_activity
entrust_idaas	authentication
entrust_idaas	entity_management
f5_bigip	http_activity
f5_bigip	network_activity
github_auditlogs	account_change
github_auditlogs	api_activity
github_auditlogs	entity_management
gitlab	group_management
gitlab	software_inventory_info
gitlab	user_inventory_info
gitlab	vulnerability_finding
gitlab	web_resources_activity

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
google_workspace	account_change
google_workspace	api_activity
google_workspace	authentication
google_workspace	base_event
google_workspace	data_security_finding
google_workspace	detection_finding
google_workspace	email_activity
google_workspace	entity_management
google_workspace	file_hosting_activity
google_workspace	user_access_management
google_workspace	web_resources_activity
hashicorp_vault	api_activity
jamf_pro	authentication
jamf_pro	device_inventory_info
jamf_pro	entity_management
jamf_protect	detection_finding
jamf_protect	device_inventory_info
jamf_protect	dns_activity
jamf_protect	software_inventory_info
jamf_protect	vulnerability_finding

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
jumpcloud	account_change
jumpcloud	authentication
jumpcloud	datastore_activity
jumpcloud	detection_finding
jumpcloud	entity_management
microsoft_entraid	account_change
microsoft_entraid	authentication
microsoft_entraid	entity_management
microsoft_entraid	user_access_management
microsoft_office365	account_change
microsoft_office365	application_lifecycle
microsoft_office365	authentication
microsoft_office365	compliance_finding
microsoft_office365	detection_finding
microsoft_office365	email_activity
microsoft_office365	file_hosting_activity
microsoft_office365	group_management
microsoft_office365	incident_finding
microsoft_office365	user_access_management
microsoft_office365	vulnerability_finding

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
microsoft_office365	web_resources_activity
microsoft_windows	account_change
microsoft_windows	authentication
microsoft_windows	entity_management
microsoft_windows	event_log_activity
microsoft_windows	file_system_activity
microsoft_windows	group_management
microsoft_windows	kernel_activity
netskope_cloudexchange	account_change
netskope_cloudexchange	authentication
netskope_cloudexchange	data_security_finding
netskope_cloudexchange	detection_finding
netskope_cloudexchange	device_inventory_info
netskope_cloudexchange	entity_management
netskope_cloudexchange	file_hosting_activity
netskope_cloudexchange	network_activity
okta_auth0	api_activity
okta_auth0	authentication
okta_sso	api_activity
okta_sso	authentication

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
okta_sso	detection_finding
okta_sso	entity_management
onelogin_identity	account_change
onelogin_identity	authentication
onelogin_identity	entity_management
paloaltonetworks_nextgenerationfirewall	authentication
paloaltonetworks_nextgenerationfirewall	detection_finding
paloaltonetworks_nextgenerationfirewall	network_activity
paloaltonetworks_nextgenerationfirewall	process_activity
paloaltonetworks_prismacloud	base_event
paloaltonetworks_prismacloud	cloud_resources_inventory_info
paloaltonetworks_prismacloud	compliance_finding
paloaltonetworks_prismacloud	detection_finding
paloaltonetworks_prismacloud	device_inventory_info
paloaltonetworks_prismacloud	entity_management
paloaltonetworks_prismacloud	scan_activity
paloaltonetworks_prismacloud	vulnerability_finding

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
pingidentity_pingaccess	api_activity
pingidentity_pingaccess	http_activity
pingidentity_pingfederate	authentication
pingidentity_pingfederate	entity_management
pingidentity_pingone	account_change
pingidentity_pingone	authentication
pingidentity_pingone	entity_management
proofpoint_tap	detection_finding
proofpoint_tap	email_activity
proofpoint_tap	osint_inventory_info
qualys_csam	device_inventory_info
qualys_csam	osint_inventory_info
qualys_csam	software_inventory_info
qualys_csam	vulnerability_finding
qualys_policyaudit	compliance_finding
qualys_vmdr	account_change
qualys_vmdr	api_activity
qualys_vmdr	authentication
qualys_vmdr	device_inventory_info
qualys_vmdr	entity_management

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
qualys_vmdr	vulnerability_finding
sailpoint_isc	account_change
sailpoint_isc	authentication
sailpoint_isc	entity_management
sailpoint_isc	user_access_management
salesforce_elf	account_change
salesforce_elf	api_activity
salesforce_elf	authentication
salesforce_elf	datastore_activity
salesforce_elf	detection_finding
salesforce_elf	entity_management
salesforce_elf	file_hosting_activity
salesforce_elf	http_activity
salesforce_rtem	api_activity
salesforce_rtem	authentication
salesforce_rtem	datastore_activity
salesforce_rtem	file_hosting_activity
sentinelone_endpointsecurity	dns_activity
sentinelone_endpointsecurity	file_system_activity
sentinelone_endpointsecurity	http_activity

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
sentinelone_endpointsecurity	process_activity
servicenow_cmdb	api_activity
servicenow_cmdb	datastore_activity
servicenow_cmdb	entity_management
slack_auditlogs	account_change
slack_auditlogs	authentication
slack_auditlogs	detection_finding
slack_auditlogs	entity_management
slack_auditlogs	file_hosting_activity
slack_auditlogs	user_access_management
slack_auditlogs	web_resources_activity
snowflake_platform	authentication
snowflake_platform	datastore_activity
snowflake_platform	entity_management
snowflake_platform	group_management
snowflake_platform	user_access_management
snowflake_platform	user_inventory_info
tanium_endpointmanagement	account_change
tanium_endpointmanagement	authentication
tanium_endpointmanagement	compliance_finding

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
tanium_endpointmanagement	detection_finding
tanium_endpointmanagement	device_inventory_info
tanium_endpointmanagement	entity_management
tanium_endpointmanagement	group_management
tanium_endpointmanagement	vulnerability_finding
tenable_vulnerabilitymanagement	account_change
tenable_vulnerabilitymanagement	api_activity
tenable_vulnerabilitymanagement	authentication
tenable_vulnerabilitymanagement	compliance_finding
tenable_vulnerabilitymanagement	device_inventory_info
tenable_vulnerabilitymanagement	scan_activity
tenable_vulnerabilitymanagement	vulnerability_finding
wazuh_platform	api_activity
wazuh_platform	application_lifecycle
wazuh_platform	authentication
wazuh_platform	compliance_finding
wazuh_platform	detection_finding
wazuh_platform	device_inventory_info
wazuh_platform	file_system_activity
wazuh_platform	remediation_activity

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
wazuh_platform	vulnerability_finding
wiz_cnapp	api_activity
wiz_cnapp	authentication
wiz_cnapp	compliance_finding
wiz_cnapp	detection_finding
wiz_cnapp	vulnerability_finding
zeek	authentication
zeek	base_event
zeek	detection_finding
zeek	dhcp_activity
zeek	dns_activity
zeek	email_activity
zeek	ftp_activity
zeek	http_activity
zeek	network_activity
zeek	rdp_activity
zeek	smb_activity
zeek	software_inventory_info
zeek	ssh_activity
zeek	tunnel_activity

Nome dell'origine dati (campo @data_source_name)	Tipo di origine dati (campo @data_source_type)
`zscaler_internetaccess`	`authentication`
`zscaler_internetaccess`	`dns_activity`
`zscaler_internetaccess`	`http_activity`
`zscaler_internetaccess`	`network_activity`

Altre fonti di terze parti tramite AWS Security Hub CSPM

Ulteriori risultati di sicurezza di terze parti sono disponibili tramite l'integrazione CSPM AWS di Security Hub. I seguenti partner inviano i risultati a Security Hub CSPM, che sono quindi disponibili come fonti di dati in CloudWatch Logs. Per dettagli completi su queste integrazioni, consulta le integrazioni dei Third-party prodotti con Security Hub CSPM nella Guida per l'utente di AWS Security Hub.

Partner	Integrazione
3CoreSEC — NTA	Invia i risultati tramite Security Hub CSPM
Alert Logic: gestione delle minacce senza SIEM	Invia i risultati tramite Security Hub CSPM
Aqua Security: piattaforma di sicurezza nativa per il cloud	Invia i risultati tramite Security Hub CSPM
Aqua Security — Kube-bench	Invia i risultati tramite Security Hub CSPM
Armatura: armatura ovunque	Invia i risultati tramite Security Hub CSPM
AttackIQ	Invia i risultati tramite Security Hub CSPM

Partner	Integrazione
Barracuda Networks — Cloud Security Guardian	Invia i risultati tramite Security Hub CSPM
BigID — BigID Enterprise	Invia i risultati tramite Security Hub CSPM
Esagono blu	Invia i risultati tramite Security Hub CSPM
Check Point — CloudGuard IaaS	Invia i risultati tramite Security Hub CSPM
Check Point — Gestione della postura CloudGuard	Invia i risultati tramite Security Hub CSPM
Clarity — XDome	Invia i risultati tramite Security Hub CSPM
Sicurezza dell'archiviazione nel cloud — Antivirus per Amazon S3	Invia i risultati tramite Security Hub CSPM
Contrast Security — Contrast Assessment	Invia i risultati tramite Security Hub CSPM
CrowdStrike — Falco CrowdStrike	Invia i risultati tramite Security Hub CSPM
CyberArk — Analisi privilegiata delle minacce	Invia i risultati tramite Security Hub CSPM
Teorema dei dati	Invia i risultati tramite Security Hub CSPM
Dati	Invia i risultati tramite Security Hub CSPM
Forcepoint — CASB	Invia i risultati tramite Security Hub CSPM

Partner	Integrazione
Forcepoint — Gateway di sicurezza nel cloud	Invia i risultati tramite Security Hub CSPM
Forcepoint — DLP	Invia i risultati tramite Security Hub CSPM
Forcepoint — NGFW	Invia i risultati tramite Security Hub CSPM
Fuga	Invia i risultati tramite Security Hub CSPM
Guardicore — Centra	Invia i risultati tramite Security Hub CSPM
HackerOne — Informazioni sulla vulnerabilità	Invia i risultati tramite Security Hub CSPM
JFrog — Xray	Invia i risultati tramite Security Hub CSPM
Juniper Networks: firewall vSRx di nuova generazione	Invia i risultati tramite Security Hub CSPM
k9 Security — Analizzatore di accesso	Invia i risultati tramite Security Hub CSPM
Merletti	Invia i risultati tramite Security Hub CSPM
McAfee — MVISION CNAPP	Invia i risultati tramite Security Hub CSPM
NETSCOUT — Investigatore informatico	Invia i risultati tramite Security Hub CSPM
Orca — Piattaforma di sicurezza cloud	Invia i risultati tramite Security Hub CSPM

Partner	Integrazione
Palo Alto Networks — Prisma Cloud Compute	Invia i risultati tramite Security Hub CSPM
Palo Alto Networks — Prisma Cloud Enterprise	Invia i risultati tramite Security Hub CSPM
Plerion — Piattaforma di sicurezza cloud	Invia i risultati tramite Security Hub CSPM
che si aggira	Invia i risultati tramite Security Hub CSPM
Qualys — Gestione delle vulnerabilità	Invia i risultati tramite Security Hub CSPM
Rapid7 — InsightVM	Invia i risultati tramite Security Hub CSPM
SentinelOne	Invia i risultati tramite Security Hub CSPM
Snyk	Invia i risultati tramite Security Hub CSPM
Sonrai Security — Sonrai Dig	Invia i risultati tramite Security Hub CSPM
Sophos — Protezione dei server	Invia i risultati tramite Security Hub CSPM
StackRox — Sicurezza Kubernetes	Invia i risultati tramite Security Hub CSPM
Sumo Logic — Analisi dei dati delle macchine	Invia i risultati tramite Security Hub CSPM
Symantec — Protezione dei carichi di lavoro nel cloud	Invia i risultati tramite Security Hub CSPM

Partner	Integrazione
Tenable.io	Invia i risultati tramite Security Hub CSPM
Trend Micro — Cloud One	Invia i risultati tramite Security Hub CSPM
Vectra — Cognito Detect	Invia i risultati tramite Security Hub CSPM
Wiz	Invia i risultati tramite Security Hub CSPM
Caveonix — Caveonix Cloud	Invia e riceve i risultati tramite Security Hub CSPM
Cloud Custodian	Invia e riceve i risultati tramite Security Hub CSPM
DisruptOps	Invia e riceve i risultati tramite Security Hub CSPM
Kion	Invia e riceve i risultati tramite Security Hub CSPM
Turbot	Invia e riceve i risultati tramite Security Hub CSPM

Note

Questo elenco riflette le integrazioni dei partner di Security Hub che inviano i risultati al momento della stesura. Poiché AWS Security Hub aggiunge regolarmente nuove integrazioni per i partner, consulta le integrazioni di Third-party prodotti con Security Hub CSPM nella AWS Security Hub User Guide per l'elenco più aggiornato dei partner disponibili.

Origini personalizzate

Per i log delle applicazioni personalizzati, è possibile definire la propria categorizzazione delle fonti di dati utilizzando uno dei seguenti metodi:

- Tag del gruppo di log: aggiungi tag ai tuoi gruppi di log utilizzando le chiavi `cw:datasource:name` e `cw:datasource:type` specificando il nome e il tipo dell'origine dati per tutti i log inseriti nel gruppo di log.
- Configurazione della pipeline: configura le informazioni sull'origine dati tramite pipeline di elaborazione dei log durante l'acquisizione dei log delle applicazioni.

> **Note**
>
> I nomi delle origini dati personalizzati non possono iniziare con «aws» o «amazon» per evitare conflitti con i log di servizio. AWS

Analisi dei dati di registro con CloudWatch Log Analytics

Log Analytics è un'esperienza di console unificata in Amazon CloudWatch Logs che riunisce le funzionalità di analisi dei log in un unico posto. Puoi interrogare e analizzare i dati di log con CloudWatch Logs Insights, trasmettere i log in streaming in tempo reale con Live Tail e identificare i tuoi collaboratori principali con Contributor Insights. CloudWatch Log Analytics è l'esperienza predefinita per l'analisi dei log nella console Logs. CloudWatch

Cosa puoi fare con Log Analytics

Log Analytics riunisce le seguenti funzionalità:

- Interroga e analizza i dati di registro: esegui query sui tuoi gruppi di log con CloudWatch Logs Insights. Puoi eseguire più query in diverse schede, salvare e riutilizzare query con parametri, esplorare i dati con sfaccettature, generare query utilizzando il linguaggio naturale e visualizzare i risultati. Per ulteriori informazioni, consulta Analisi dei dati di registro con CloudWatch Logs Insights.
- Trasmetti i log in streaming quasi in tempo reale: utilizza Live Tail per visualizzare un elenco in streaming degli eventi di registro man mano che vengono inseriti e filtrare o evidenziare gli eventi che ti interessano. Per ulteriori informazioni, consulta Risolvi i problemi con CloudWatch Logs Live Tail.
- Identifica i contributori principali: utilizza CloudWatch Contributor Insights per analizzare i dati di registro ad alta cardinalità e vedere i contributori che hanno il maggiore impatto sul tuo sistema. Per ulteriori informazioni, consulta Utilizzare Contributor Insights per analizzare dati ad alta cardinalità.

Inizia a usare Log Analytics

Per utilizzare Log Analytics, accedi alla CloudWatch console e scegli Log Analytics in Logs nel riquadro di navigazione.

Log Analytics è l'esperienza predefinita. Se disattivi, puoi continuare a utilizzare CloudWatch Logs Insights, Live Tail e Contributor Insights come esperienze distinte insieme a Log Analytics e puoi tornare a Log Analytics in qualsiasi momento.

Log Analytics utilizza gli stessi prezzi delle funzionalità di base: le query di CloudWatch Logs Insights, Live Tail e Contributor Insights. Per i dettagli sui prezzi, vedere Prezzi di CloudWatch .

Analisi dei dati di registro con CloudWatch Logs Insights

Con CloudWatch Logs Insights, puoi cercare e analizzare in modo interattivo i dati di log in Amazon CloudWatch Logs. Puoi eseguire interrogazioni per aiutarti a rispondere in modo più efficiente ed efficace ai problemi operativi. Oltre a eseguire interrogazioni utilizzando gruppi di log, è possibile eseguire query utilizzando facet, origine dati e tipo di dati. Se si verifica un problema, puoi utilizzare CloudWatch Logs Insights per identificare le cause potenziali e convalidare le correzioni implementate. Sei limitato a 100 CloudWatch Logs Insights QL simultanei per account, comprese le query aggiunte ai dashboard. Inoltre, puoi eseguire 15 query simultanee per Service PPL o OpenSearch Service SQL. OpenSearch

CloudWatch Logs Insights supporta 3 linguaggi di query che puoi utilizzare per le tue query:

- Un linguaggio di query Logs Insights appositamente creato (Logs Insights QL) con pochi comandi semplici ma potenti.
- OpenSearch Service Piped Processing Language (PPL). OpenSearch PPL consente di analizzare i log utilizzando una serie di comandi delimitati da pipe (|).

  Con OpenSearch PPL è possibile recuperare, interrogare e analizzare i dati utilizzando comandi collegati tra loro, semplificando la comprensione e la composizione di query complesse. La sintassi consente il concatenamento di comandi per trasformare ed elaborare i dati. Con PPL, è possibile filtrare e aggregare i dati e utilizzare un ricco set di funzioni matematiche, di stringhe, di data, condizionali e di altro tipo per l'analisi.
- OpenSearch Service Structured Query Language (SQL). Con le query OpenSearch SQL, puoi analizzare i log in modo dichiarativo. È possibile utilizzare comandi come SELECT, FROM, WHERE, GROUP BY, HAVING e vari altri comandi e funzioni disponibili in SQL. È possibile eseguire JOIN tra gruppi di log, correlare i dati tra i log utilizzando sottoquery e utilizzare il ricco set di funzioni SQL JSON, Mathematical, String, Conditional e di altro tipo per eseguire analisi approfondite sui log.

  Quando usi i comandi SQL o PPL, assicurati di racchiudere i campi con caratteri speciali (non alfabetici e non numerici) tra i tasti rovesciati per interrogarli correttamente. Ad esempio, racchiudi e tra segni di spunta rovesciati. `@message` `Operation.Export` `Test::Field` Non è necessario racchiudere campi con nomi puramente alfabetici nei backtick.

CloudWatch Logs Insights offre le seguenti funzionalità disponibili per l'uso con qualsiasi linguaggio di query.

- Rilevamento automatico dei campi di log nei log di AWS servizi come Amazon Route 53 e Amazon VPC e di qualsiasi applicazione o registro personalizzato che emette eventi di registro in formato JSON. AWS Lambda AWS CloudTrail
- Creazione di indici di campo per ridurre i costi e velocizzare i risultati, in particolare per le query relative a un numero elevato di gruppi di log o di eventi di log. Dopo aver creato gli indici di campo dei campi comuni negli eventi di registro, è possibile utilizzarli in una query. La query ignora l'elaborazione degli eventi di registro che sono noti per non includere il campo indicizzato ed elabora meno dati.

  > **Note**
  >
  > Il `filterIndex` comando è disponibile solo in Logs Insights QL.

- Rilevamento e analisi dei pattern negli eventi di registro. Un pattern è una struttura di testo condivisa che ricorre tra i campi di registro. Quando visualizzi i risultati di una query, puoi scegliere la scheda Patterns per vedere i modelli trovati da CloudWatch Logs in base a un campione dei tuoi risultati.
- Salvare le interrogazioni, visualizzare la cronologia delle interrogazioni, rieseguire le interrogazioni salvate e utilizzare le interrogazioni salvate con parametri.
- Aggiungere interrogazioni alle dashboard.
- Crittografia dei risultati delle query con. AWS Key Management Service
- La generazione di query con linguaggio naturale consente di utilizzare il linguaggio naturale per creare query CloudWatch Logs Insights. Puoi porre domande o descrivere i dati che stai cercando, quindi l'intelligenza artificiale genera una query in base alla tua richiesta e fornisce una spiegazione riga per riga di come funziona la query.
- Usa le sfaccettature per raggruppare, filtrare ed esplorare in modo interattivo i tuoi log.
- I log circostanti consentono di visualizzare le righe di registro prima e dopo ogni record di registro specifico nei risultati della query per ottenere un contesto istantaneo sugli eventi di registro critici. È possibile configurare l'intervallo per visualizzare 5, 10, 20, 50 o 100 righe di registro prima e dopo un record selezionato e cercare parole chiave specifiche nei log circostanti.

Le seguenti funzionalità di CloudWatch Logs Insights sono supportate solo quando si utilizza Logs Insights QL.

- Query di confronto che confrontano gli eventi di registro in un gruppo di log con gli eventi di registro di un periodo di tempo precedente.

> **Important**
>
> CloudWatch Logs Insights non può accedere agli eventi di registro con timestamp precedenti all'ora di creazione del gruppo di log.

Se hai effettuato l'accesso a un account configurato come account di monitoraggio in osservabilità CloudWatch tra account, puoi eseguire query CloudWatch Logs Insights sui gruppi di log negli account di origine collegati a questo account di monitoraggio. È possibile eseguire una query che interroga più gruppi di log situati in account diversi. Per maggiori informazioni, consulta la sezione Osservabilità su più account di CloudWatch .

Quando crei query utilizzando Logs Insights QL, puoi anche utilizzare il linguaggio naturale per creare query Logs Insights. CloudWatch Per farlo, descrivi o fai domande sui dati che stai cercando. Questa AI-assisted funzionalità genera una query in base alla richiesta e fornisce una spiegazione riga per riga del funzionamento della query. Per ulteriori informazioni, consulta Utilizzare il linguaggio naturale per generare e aggiornare CloudWatch le query di Logs Insights.

Le interrogazioni che utilizzano uno dei linguaggi di query supportati scadono dopo 60 minuti, se non sono state completate. I risultati delle query sono disponibili per 7 giorni.

CloudWatch Le query di Logs Insights comportano addebiti in base alla quantità di dati di registro non compressi scansionati, indipendentemente dal linguaggio di interrogazione. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

Puoi utilizzare CloudWatch Logs Insights per cercare i dati di log inviati a CloudWatch Logs il 5 novembre 2018 o successivamente.

> **Important**
>
> Se il team addetto alla sicurezza della rete non consente l'uso di socket Web, al momento non puoi accedere alla parte CloudWatch Logs Insights della console. CloudWatch Puoi utilizzare le funzionalità di interrogazione di CloudWatch Logs Insights utilizzando le API. Per ulteriori informazioni, StartQueryconsulta Amazon CloudWatch Logs API Reference.

Indice

- Linguaggi di interrogazione supportati

- Usa il linguaggio naturale per generare e aggiornare CloudWatch le query di Logs Insights
- Registri supportati e campi rilevati
- Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione
- Usa le sfaccettature per raggruppare ed esplorare i log
- Visualizzazione dei log circostanti in Logs Insights CloudWatch
- Analisi del modello
- Salva ed esegui nuovamente le query di Logs Insights CloudWatch
- Aggiunta di query a pannello di controllo o esportazione dei risultati della query
- Visualizzazione di query in esecuzione o cronologia delle query
- Crittografa i risultati delle interrogazioni con AWS Key Management Service
- Genera un riepilogo in linguaggio naturale dai risultati delle query di CloudWatch Logs Insights

Linguaggi di interrogazione supportati

Nelle sezioni seguenti sono elencati i comandi supportati in ogni linguaggio di query. Descrivono inoltre il formato della sintassi e forniscono interrogazioni di esempio.

Argomenti

- CloudWatch Linguaggio di interrogazione Logs Insights (Logs Insights QL)
- OpenSearch Piped Processing Language (PPL)
- OpenSearch Structured Query Language (SQL)

CloudWatch Linguaggio di interrogazione Logs Insights (Logs Insights QL)

Questa sezione include la documentazione completa dei comandi e delle funzioni QL di Logs Insights. Include anche esempi di query per questo linguaggio.

Per informazioni sugli altri linguaggi di query che è possibile utilizzare, vedere OpenSearch Service PPL, OpenSearch Service SQL e. CloudWatch Metrics Insights

Argomenti

- CloudWatch Sintassi delle interrogazioni in linguaggio Logs Insights
- Inizia a usare Logs Insights QL: tutorial sulle query
- Query di esempio

- Confronta (diff) con gli intervalli di tempo precedenti
- Visualizzazione dei dati di log nei grafici

CloudWatch Sintassi delle interrogazioni in linguaggio Logs Insights

Questa sezione fornisce dettagli su Logs Insights QL. La sintassi delle query supporta funzioni e operazioni diverse, incluse, a titolo esemplificativo ma non esaustivo, funzioni generali, operazioni aritmetiche e di confronto ed espressioni regolari.

> **Important**
>
> Per evitare di incorrere in costi eccessivi a causa dell'esecuzione di query di grandi dimensioni, tieni presente le seguenti best practice:
>
> - Seleziona solo i gruppi di log necessari per ogni query.
> - Specificate sempre l'intervallo di tempo più ristretto possibile per le vostre interrogazioni.
> - Quando usi la console per eseguire query, annulla tutte le query prima di chiudere la pagina della console di Logs Insights. CloudWatch In caso contrario, le query continueranno a essere eseguite fino al completamento.
> - Quando aggiungi un widget CloudWatch Logs Insights a una dashboard, assicurati che la dashboard non si aggiorni ad alta frequenza, poiché ogni aggiornamento avvia una nuova query.

Per creare query che contengono più comandi, separali con la barra verticale (|).

Per creare query che contengono commenti, imposta i commenti con il carattere cancelletto (#).

> **Note**
>
> CloudWatch Logs Insights rileva automaticamente i campi per diversi tipi di log e genera campi che iniziano con il carattere @. Per ulteriori informazioni su questi campi, consulta la sezione Log supportati e campi rilevati nella Amazon CloudWatch User Guide.

La seguente tabella descrive brevemente ogni comando. Di seguito è riportata una descrizione più completa di ogni comando, con esempi.

> **Note**
>
> Tutti i comandi di query QL di Logs Insights sono supportati nei gruppi di log nella classe di log Standard. I gruppi di log nella classe di log Infrequent Access supportano tutti i comandi di query QL di Logs Insights tranne`pattern`, `diff` e. `unmask`

addtotals	Calcola i totali delle righe e delle colonne per i campi numerici.
anomaly	Identifica modelli insoliti nei dati di registro utilizzando l'apprendimento automatico.
display	Mostra uno o più campi specifici nei risultati della query.
fields	Mostra campi specifici nei risultati della query e supporta funzioni e operazioni che puoi utilizzare per modificare i valori dei campi e creare nuovi campi da utilizzare nella query.
filter	Filtra la query per restituire solo i log eventi che soddisfano una o più condizioni.
filterIndex	Forza una query a tentare di scansionare solo i gruppi di log che sono entrambi indicizzati nel campo menzionato in un indice di campo e contengono anche un valore per quell'indice di campo. Ciò riduce il volume scansionato tentando di analizzare solo gli eventi di registro di questi gruppi di log che contengono il valore specificato nella query per questo indice di campo. Questo comando non è supportato per i gruppi di log nella classe di log Infrequent Access.
pattern	Raggruppa automaticamente in cluster i dati di log in pattern. Un pattern è una struttura di testo condivisa che ricorre tra i campi di registro. CloudWatch Logs Insights consente di analizzare i modelli rilevati negli eventi di registro. Per ulteriori informazioni, consulta Analisi del modello.
diff	Confronta gli eventi di registro trovati nel periodo di tempo richiesto con gli eventi di registro di un periodo di tempo precedente di uguale durata,

	in modo da poter cercare le tendenze e scoprire se determinati eventi di registro sono nuovi.
parse	Estrae i dati da un campo di log per creare un campo estratto che puoi elaborare nella query. **parse** supporta sia la modalità glob con i caratteri jolly che le espressioni regolari.
sort	Mostra i log eventi restituiti in ordine crescente (asc) o decrescente (desc).
SOURCE	L'inclusione SOURCE in una query è un modo utile per specificare una grande quantità di gruppi di log da includere in una query basata sul prefisso del nome del gruppo di log, sugli identificatori di account, sulla classe del gruppo di log, sulle fonti di dati o sui tag del gruppo di log. Questo comando è supportato solo quando si crea una query in AWS CLI o a livello di codice, non nella console. CloudWatch
stats	Calcola le statistiche aggregate utilizzando i valori dei campi di log.
limit	Specifica il numero massimo di log eventi che la query deve restituir e. Utile con **sort** per restituire i "primi 20 risultati" o i "20 risultati più recenti". limit anyDa utilizzare per interrompere tempestivamente la scansione non appena vengono trovati risultati sufficienti.
dedup	Rimuove i risultati duplicati in base a valori specifici nei campi indicati.
unmask	Mostra tutto il contenuto di un log eventi nel quale alcuni contenuti sono mascherati a causa di una policy di protezione dei dati. Per ulteriori informazioni sulla protezione dei dati nei gruppi di log, consulta la sezione Incremento della protezione dei dati di log sensibili con il mascheramento.
unnest	Appiattisce un elenco preso come input per produrre più record con un singolo record per ogni elemento dell'elenco.
lookup	Arricchisce gli eventi di registro con i dati di una tabella di ricerca abbinando i valori dei campi. Utilizza le tabelle di ricerca per aggiungere dati di riferimento come dettagli utente, nomi di applicazioni o informazi oni sui prodotti ai risultati delle query.

join	Combina gli eventi di registro di un gruppo di log di origine con gli eventi di un altro gruppo di log o il risultato di una query in base a un campo corrispondente. Utilizzate il comando join per correlare gli eventi di registro correlati tra diverse fonti utilizzando chiavi comuni tra loro, come gli identificatori di richiesta o gli ID delle transazioni corrispondenti.
subqueries	Una sottoquery è una query annidata di Logs Insights che può essere utilizzata come input per un'altra query. Le sottoquery possono essere utilizzate per derivare set di risultati intermedi che vengono poi utilizzati dai comandi successivi.
autoregress	Crea copie ritardate (riga precedente) dei valori di un campo.
logcompare	Confronta la finestra temporale corrente con una finestra di base spostata indietro di una durata.
filldown	Porta avanti l'ultimo valore non nullo per colmare gli spazi vuoti.
fillmissing	Inserisce le righe per i contenitori di tempo vuoti dopo stats by bin (), riempiendo facoltativamente i campi con una costante.
cidrlookup	Arricchisce gli eventi confrontando un campo IP con gli intervalli CIDR in una tabella di ricerca.
outlier	Rileva i valori anomali statistici in base all'intervallo interquartile (IQR) e può rimuovere o trasformare le righe anomale.
accum	Calcola la somma cumulativa corrente di un campo numerico.
appendcols	Aggiunge le colonne di una sottoquery ai risultati correnti mediante la corrispondenza posizionale delle righe.
sessionize	Raggruppa gli eventi in sessioni in base ai campi di identità e al divario di inattività.
countFrequent	Restituisce un conteggio approssimativo di ogni combinazione univoca di campo-valore, in ordine decrescente.

where	Funziona come alias grammaticale per il comando e accetta una sintassi identica. `filter`
stimare	Restituisce i byte stimati che la query scansionerebbe nei gruppi di log e nell'intervallo di tempo selezionati, senza eseguire la query.
Altre operazioni e funzioni	CloudWatch Logs Insights supporta anche molte funzioni e operazioni di confronto, aritmetiche, datetime, numeriche, stringhe, indirizzi IP e funzioni e operazioni generali.

Le sezioni seguenti forniscono maggiori dettagli sui comandi di query di Logs Insights. CloudWatch

Argomenti

- I comandi QL di Logs Insights sono supportati nelle classi di log
- aggiungi totali
- anomalia
- display
- campi
- filter
- FilterIndex
- SOURCE
- pattern
- diff
- parse
- campi pertinenti
- espansione
- sort
- stats
- limit
- dedup
- unmask

- unnest
- cercare
- join
- subqueries
- regressione automatica
- logcompare
- riempimento
- riempimento mancante
- cidrlookup
- anomalo
- accum
- appendici
- sessionizzare
- Conta Frequenti
- dove
- stimare
- Funzioni booleane, di confronto, numeriche, datetime e altre
- Funzioni di hashing
- Time-series funzioni
- Campi che contengono caratteri speciali
- Utilizzo di alias e commenti nelle query

I comandi QL di Logs Insights sono supportati nelle classi di log

Tutti i comandi di query QL di Logs Insights sono supportati nei gruppi di log nella classe di log Standard. I gruppi di log nella classe di log Infrequent Access supportano tutti i comandi di query tran`pattern`,, `diffilterIndex`, e. `unmask`

aggiungi totali

Si usa `addtotals` per calcolare i totali delle righe e delle colonne per i campi numerici. I totali delle righe vengono calcolati per blocco. I totali delle colonne (`col=true`) vengono calcolati dopo l'unione

per verificarne la correttezza tra i blocchi distribuiti. È possibile utilizzare la selezione personalizzata dei campi, un nome di campo personalizzato e disabilitare i totali delle righe. `row=false`

Sintassi

```
| addtotals [fieldname=Name] [row=bool] [col=bool] [field1, field2, ...]
```

- `fieldname=`*`Name`*— Imposta il nome della colonna relativa al totale delle righe. Il valore predefinito è `Total`.
- `row=`*`bool`*— Abilita o disabilita i totali delle righe. Il valore predefinito è `true`.
- `col=`*`bool`*— Abilita o disabilita i totali delle colonne, che aggiunge una riga di riepilogo nella parte inferiore. Il valore predefinito è `false`.
- *`field1, field2, ...`*— Elenco opzionale di campi specifici da sommare. Se omesso, tutti i campi numerici vengono sommati.

Esempi

```
# Row totals (default) - adds a Total column summing all numeric fields
fields a, b
| addtotals
```

```
# Sum only specific fields
fields price, tax, qty
| addtotals price, tax
```

```
# Custom total column name
fields val
| addtotals fieldname=RowSum
```

```
# Column totals - adds a Summary row at the bottom
fields x
| addtotals col=true x
```

anomalia

Utilizzalo `anomaly` per identificare automaticamente modelli insoliti e potenziali problemi all'interno dei dati di registro utilizzando l'apprendimento automatico.

Il `anomaly` comando estende le `pattern` funzionalità esistenti e utilizza analisi avanzate per aiutare a identificare potenziali anomalie nei dati di registro. È possibile utilizzarlo `anomaly` per ridurre il tempo necessario per identificare e risolvere i problemi operativi evidenziando automaticamente modelli o comportamenti insoliti nei log.

Il `anomaly` comando funziona con il `pattern` comando per identificare innanzitutto gli schemi di registro, quindi rilevare le anomalie all'interno di tali modelli. Puoi anche combinarlo `anomaly` con `sort` i comandi `filter` o per concentrare il rilevamento delle anomalie su sottoinsiemi specifici dei tuoi dati.

Anomaly Command Input

Il `anomaly` comando viene in genere utilizzato dopo il `pattern` comando per analizzare i modelli identificati nei dati di registro. Il comando non richiede parametri aggiuntivi e analizza l'output dei comandi precedenti nella query.

Tipi di anomalie identificate

Il `anomaly` comando identifica cinque tipi distinti di anomalie:

- Anomalie della frequenza dei pattern: frequenze insolite di pattern di log specifici, ad esempio quando un'applicazione inizia a generare più messaggi di errore del solito.
- Nuovi modelli di anomalie: modelli di registro mai visti in precedenza che possono indicare nuovi tipi di errori o messaggi che appaiono nei log.
- Anomalie dovute alla variazione dei token: modifiche impreviste nel contenuto dei messaggi di registro che possono indicare variazioni insolite nei formati di registro previsti.
- Anomalie numeriche dei token: modifiche insolite nei valori numerici all'interno dei log che possono aiutare a rilevare potenziali problemi di prestazioni o variazioni metriche impreviste.
- Anomalie del codice di errore HTTP: modelli relativi alle risposte di errore HTTP, particolarmente utili per il monitoraggio di applicazioni Web e API.

Anomaly Command Output

Il `anomaly` comando preserva tutti i campi dai dati di input e aggiunge i risultati del rilevamento delle anomalie per aiutare a identificare modelli insoliti nei dati di registro.

Esempi

Il comando seguente identifica i modelli nei dati di registro e quindi rileva le anomalie all'interno di tali modelli:

```
fields @timestamp, @message
| pattern @message
| anomaly
```

Il `anomaly` comando può essere utilizzato con il filtraggio per concentrarsi su tipi di log specifici:

```
fields @timestamp, @message
| filter @type = "REPORT"
| pattern @message
| anomaly
```

Il `anomaly` comando può essere combinato con l'ordinamento per organizzare i risultati:

```
fields @timestamp, @message
| filter @type = "ERROR"
| pattern @message
| anomaly
| sort @timestamp desc
```

display

Utilizza `display` per mostrare uno o più campi specifici nei risultati della query.

Il comando `display` mostra solo i campi specificati. Se la query contiene più comandi `display`, i risultati della query mostrano solo il campo o i campi specificati nel comando `display` finale.

Esempio: visualizzazione di un campo

Il frammento di codice mostra un esempio di query che utilizza il comando parse per estrarre dati da `@message` per creare i campi estratti `loggingType` e `loggingMessage`. La query restituisce i log eventi in cui i valori per `loggingType` sono ERROR. `display` e mostra solo i valori per `loggingMessage` nei risultati della query.

```
fields @message
| parse @message "[*] *" as loggingType, loggingMessage
| filter loggingType = "ERROR"
| display loggingMessage
```

> **Tip**
>
> Utilizzo di `display` solo una volta in una query. Se utilizzi `display` più volte in una query, i risultati della query mostrano i campi specificati nell'ultima occorrenza del comando `display` usato.

campi

Utilizza `fields` per mostrare campi specifici nei risultati della query.

Se la query contiene più comandi `fields` e non include un comando `display`, i risultati mostrano tutti i campi specificati nei comandi `fields`.

Esempio: visualizzazione di campi specifici

Il seguente esempio mostra una query che restituisce 20 log eventi e li visualizza in ordine decrescente. I valori per `@timestamp` e `@message` sono mostrati nei risultati della query.

```
fields @timestamp, @message
| sort @timestamp desc
| limit 20
```

`fields`Utilizzalo invece di `display` quando desideri utilizzare le diverse funzioni e operazioni supportate da `fields` per modificare i valori dei campi e creare nuovi campi da utilizzare nelle query.

Puoi utilizzare il comando `fields` con la parola chiave as per creare campi estratti che utilizzano campi e funzioni presenti nel log eventi. Ad esempio, `fields ispresent as isRes` crea un campo estratto denominato `isRes` che può essere utilizzato nel resto della query.

filter

Utilizza `filter` per ottenere log eventi che corrispondono a una o più condizioni.

Esempio: filtrare i log eventi utilizzando una condizione

Il frammento di codice mostra un esempio di query che restituisce tutti i log eventi in cui il valore per `range` è maggiore di 3000. La query limita i risultati a 20 log eventi e li ordina per `@timestamp` e in ordine decrescente.

```
fields @timestamp, @message
```

```
| filter (range>3000)
| sort @timestamp desc
| limit 20
```

Esempio: filtrare i log eventi utilizzando più di una condizione

È possibile utilizzare le parole chiave `and` e `or` per combinare più condizioni.

Il frammento di codice mostra un esempio di query che restituisce tutti i log eventi in cui il valore per `range` è maggiore di 3000 e il valore per `accountId` è uguale a 123456789012. La query limita i risultati a 20 log eventi e li ordina per `@timestamp` e in ordine decrescente.

```
fields @timestamp, @message
| filter (range>3000 and accountId=123456789012)
| sort @timestamp desc
| limit 20
```

Campi indicizzati e comando filter

Se sono stati creati indici di campo per un gruppo di log, è possibile utilizzare tali indici di campo per rendere le `filter` query più efficienti e ridurre il volume di scansione. Ad esempio, supponiamo di aver creato un indice di campo per. `requestId` Quindi, qualsiasi query di CloudWatch Logs Insights su quel gruppo di log che includa `filter requestId = value` o `filter requestId IN [value, value, ...]` tenterà di ignorare l'elaborazione di eventi di registro di cui è noto che non includono il campo indicizzato. Tentando di scansionare solo gli eventi di registro che sono noti per contenere quel campo indicizzato, è possibile ridurre il volume di scansione e velocizzare la query.

Per ulteriori informazioni sugli indici di campo e su come crearli, vedere. Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione

> **Important**
>
> Solo le interrogazioni che prevedono `filter fieldName =...` e `filter fieldName IN...` trarranno vantaggio dai miglioramenti apportati all'indice di campo. Le query con `filter fieldName like` non utilizzano indici e analizzano sempre tutti gli eventi di registro nei gruppi di log selezionati.

Esempio: trova gli eventi di registro correlati a un determinato ID di richiesta, utilizzando gli indici

L'esempio presuppone che sia stato creato un indice di campo su. `requestId` Per i gruppi di log che utilizzano questo indice di campo, la query utilizzerà gli indici di campo per tentare di analizzare il minor numero di eventi di registro per trovare eventi `requestId` con un valore di `123456`

```
fields @timestamp, @message
| filter requestId = "1234656"
| limit 20
```

Corrispondenze ed espressioni regolari nel comando di filtro

Il comando di filtro supporta l'uso di espressioni regolari. Puoi utilizzare i seguenti operatori di confronto (=, `!=`, <, <=, >, >=) e operatori booleani (`and`, `or` e `not`).

Puoi utilizzare la parola chiave `in` per verificare l'appartenenza impostata e la presenza di elementi in una matrice. Per verificare la presenza di elementi in una matrice, posiziona la matrice subito dopo `in`. Puoi utilizzare l'operatore booleano `not` con `in`. Puoi creare query che utilizzano `in` per restituire log eventi in cui i campi sono le stringhe corrispondenti. I campi devono essere stringhe complete. Ad esempio, il seguente frammento di codice mostra una query che utilizza `in` per restituire log eventi in cui il campo `logGroup` è la stringa completa `example_group`.

```
fields @timestamp, @message
| filter logGroup in ["example_group"]
```

Puoi utilizzare le frasi di parole chiave `like` e `not like` per restituire le sottostringhe corrispondenti. Puoi utilizzare l'operatore di espressione regolare =~ per restituire le sottostringhe corrispondenti. Per restituire una sottostringa corrispondente con `like` e `not like`, racchiudi la sottostringa tra virgolette singole o doppie. Puoi utilizzare modelli di espressioni regolari con `like` e `not like`. Per restituire una sottostringa corrispondente con l'operatore dell'espressione regolare, racchiudi la sottostringa tra le barre. Gli esempi seguenti contengono frammenti di codice che mostrano come è possibile restituire le sottostringhe corrispondenti utilizzando il comando `filter`.

Esempi: corrispondenza di sottostringhe

I tre esempi seguenti restituiscono log eventi in cui `f1` contiene la parola Exception. I tre esempi fanno distinzione tra lettere maiuscole e minuscole.

Il primo esempio restituisce una sottostringa corrispondente con `like`.

```
fields f1, f2, f3
| filter f1 like "Exception"
```

Il secondo esempio restituisce una sottostringa corrispondente con `like` e un modello di espressione regolare.

```
fields f1, f2, f3
| filter f1 like /Exception/
```

Il terzo esempio restituisce una sottostringa corrispondente con un'espressione regolare.

```
fields f1, f2, f3
| filter f1 =~ /Exception/
```

Esempio: corrispondenza di sottostringhe con caratteri jolly

Puoi utilizzare il simbolo dei due punti (`.`) come carattere jolly nelle espressioni regolari per restituire le sottostringhe corrispondenti. Nell'esempio seguente, la query restituisce le corrispondenze in cui il valore per `f1` inizia con la stringa `ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog./
```

Puoi posizionare un asterisco dopo il simbolo del punto (`.*`) per creare un quantificatore greedy che restituisca il maggior numero possibile di corrispondenze. Ad esempio, la query seguente restituisce i risultati in cui il valore per `f1` non inizia solo con la stringa `ServiceLog`, ma include anche la stringa `ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog.*/
```

Le possibili corrispondenze possono essere formattate come segue:

- `ServiceLogSampleApiLogGroup`
- `SampleApiLogGroupServiceLog`

Esempio: esclusione di sottostringhe dalle corrispondenze

Il seguente esempio mostra una query che restituisce log eventi in cui `f1` non contiene la parola Exception. L'esempio fa distinzione tra maiuscole e minuscole.

```
fields f1, f2, f3
```

```
| filter f1 not like "Exception"
```

Esempio: corrispondenza di sottostringhe con modelli che non fanno distinzione tra maiuscole e minuscole

Puoi associare le sottostringhe che non fanno distinzione tra maiuscole e minuscole con espressioni `like` e regolari. Posiziona il parametro (?i) prima della sottostringa corrispondente che desideri restituire. Il seguente esempio mostra una query che restituisce log eventi in cui `f1` contiene la parola Exception o exception.

```
fields f1, f2, f3
| filter f1 like /(?i)Exception/
```

FilterIndex

Utilizzato `filterIndex` per restituire solo dati indicizzati, forzando una query a scansionare solo i gruppi di log indicizzati su un campo specificato nella query. Per questi gruppi di log indicizzati in questo campo, ottimizza ulteriormente la query ignorando i gruppi di log che non contengono eventi di registro contenenti il campo specificato nella query per il campo indicizzato. Riduce ulteriormente il volume scansionato tentando di analizzare solo gli eventi di registro di questi gruppi di log che corrispondono al valore specificato nella query per questo indice di campo. Per ulteriori informazioni sugli indici di campo e su come crearli, vedere. Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione

L'utilizzo `filterIndex` con campi indicizzati può aiutarti a interrogare gruppi di log che includono petabyte di dati di registro in modo efficiente, limitando lo spazio di ricerca effettivo ai gruppi di log e agli eventi di registro con indici di campo.

Ad esempio, supponete di aver creato un indice di campo per alcuni gruppi di log del vostro `IPaddress` account. È quindi possibile creare la seguente query e scegliere di interrogare tutti i gruppi di log dell'account per trovare gli eventi di registro che includono il valore `198.51.100.0` nel `IPaddress` campo.

```
fields @timestamp, @message
| filterIndex IPaddress = "198.51.100.0"
| limit 20
```

Il `filterIndex` comando fa sì che questa query tenti di ignorare tutti i gruppi di log per i quali non sono indicizzati. `IPaddress` Inoltre, all'interno dei gruppi di log indicizzati, la query ignora gli eventi di

registro che hanno un `IPaddress` campo ma non sono stati osservati `198.51.100.0` come valore per quel campo.

Utilizzate l'`IN`operatore per espandere i risultati a uno qualsiasi dei valori multipli per i campi indicizzati. L'esempio seguente trova gli eventi di log che includono il valore `198.51.100.0` o `198.51.100.1` nel campo. `IPaddress`

```
fields @timestamp, @message
| filterIndex IPaddress in ["198.51.100.0", "198.51.100.1"]
| limit 20
```

CloudWatch Logs fornisce indici di campo predefiniti per tutti i gruppi di log della classe Standard log. Gli indici di campo predefiniti sono automaticamente disponibili per i seguenti campi:

- `@logStream`
- `@aws.region`
- `@aws.account`
- `@source.log`
- `@data_source_name`
- `@data_source_type`
- `@data_format`
- `traceId`
- `severityText`
- `attributes.session.id`

CloudWatch Logs fornisce indici di campo predefiniti anche per determinate combinazioni di nomi e tipi di origini dati. Gli indici di campo predefiniti sono automaticamente disponibili per le seguenti combinazioni di nomi e tipi di origini dati:

Nome e tipo di origine dati	Indici di campo predefiniti

`amazon_vpc.flow`	`action`
	`logStatus`
	`region`

Nome e tipo di origine dati	Indici di campo predefiniti
	flowDirection type
amazon_route53.resolver_query	query_type transport rcode
aws_waf.access	action httpRequest.country
aws_cloudtrail.data aws_cloudtrail.management	eventSource eventName awsRegion userAgent errorCode eventType managementEvent readOnly eventCategory requestId

Gli indici di campo predefiniti si aggiungono a tutti gli indici di campo personalizzati definiti all'interno della politica. Gli indici di campo predefiniti non vengono conteggiati ai fini della quota dell'indice di campo.

FilterIndex rispetto al filtro

Per illustrare la differenza tra `filterIndex` e`filter`, considera le seguenti query di esempio. Si supponga di aver creato un indice di campo per`IPaddress`, per quattro dei gruppi di log, ma non per un quinto gruppo di log. L'utilizzo della seguente query `filterIndex` salterà la scansione del gruppo di log in cui il campo non è indicizzato. Per ogni gruppo di log indicizzato, tenta di analizzare solo gli eventi di registro che hanno il campo indicizzato e restituisce inoltre solo i risultati ottenuti dopo la creazione dell'indice del campo.

```
fields @timestamp, @message
| filterIndex IPaddress = "198.51.100.0"
| limit 20
```

Al contrario, se si utilizza `filter` invece di `filterIndex` eseguire una query degli stessi cinque gruppi di log, la query tenterà di analizzare non solo gli eventi di registro che contengono il valore nei gruppi di log indicizzati, ma analizzerà anche il quinto gruppo di log che non è indicizzato e analizzerà ogni evento di registro in quel quinto gruppo di log.

```
fields @timestamp, @message
| filter IPaddress = "198.51.100.0"
| limit 20
```

SOURCE

L'inclusione `SOURCE` in una query è un modo utile per specificare le fonti di and/or dati dei gruppi di log da includere in una query quando si utilizza l'API AWS CLI o per creare una query. Il `SOURCE` comando è supportato solo nell'API AWS CLI and, non nella CloudWatch console. Quando si utilizza la CloudWatch console per avviare una query, si utilizza l'interfaccia della console per specificare i gruppi di log.

Interroga i gruppi di log

`SOURCE`Per specificare i gruppi di log da interrogare, è possibile utilizzare le seguenti parole chiave:

- `namePrefix`esegue la query su gruppi di log i cui nomi iniziano con la stringa specificata. Se si omette questa opzione, viene eseguita una query su tutti i gruppi di log.

  È possibile includere fino a 5 prefissi nell'elenco.
- `accountIdentifier`esegue la query sui gruppi di log nell'account specificato AWS . Funziona solo quando si esegue la query in un account di monitoraggio. Se si omette questa opzione,

l'impostazione predefinita prevede l'interrogazione di tutti gli account di origine collegati e dell'account di monitoraggio corrente. Per ulteriori informazioni sull'osservabilità tra account, consulta CloudWatch osservabilità tra account.

Puoi includere fino a 20 identificatori di account nell'elenco.

- `logGroupClasse`segue la query sui gruppi di log che si trovano nella classe di log specificata, Standard o Infrequent Access. Se si omette questa impostazione, viene utilizzata l'impostazione predefinita della classe di registro Standard. Per ulteriori informazioni sulle classi di log, vedereClassi di registro.

Poiché è possibile specificare un numero elevato di gruppi di log su cui eseguire query in questo modo, si consiglia di utilizzarli `SOURCE` solo nelle query che utilizzano gli indici di campo creati dall'utente. Per ulteriori informazioni sull'indicizzazione dei campi nei gruppi di log, vedere Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione

L'esempio seguente seleziona tutti i gruppi di log dell'account. Se si tratta di un account di monitoraggio, verranno selezionati i gruppi di log tra gli account di monitoraggio e tutti gli account di origine. Se il numero totale di gruppi di log supera 10.000, verrà visualizzato un errore che richiede di ridurre il numero di gruppi di log utilizzando un metodo di selezione dei gruppi di log diverso.

```
SOURCE logGroups()
```

L'esempio seguente seleziona i gruppi di log nell'account di `111122223333` origine. Se si avvia una query in un account di monitoraggio in modalità osservabilità CloudWatch tra account, per impostazione predefinita vengono selezionati i gruppi di log in tutti gli account di origine e nell'account di monitoraggio.

```
SOURCE logGroups(accountIdentifier:['111122223333'])
```

L'esempio successivo seleziona i gruppi di log in base ai prefissi dei nomi.

```
SOURCE logGroups(namePrefix: ['namePrefix1', 'namePrefix2'])
```

L'esempio seguente seleziona tutti i gruppi di log nella classe di log Infrequent Access. Se non si include l'`class`identificatore, la query si applica solo ai gruppi di log della classe di log Standard, che è l'impostazione predefinita.

```
SOURCE logGroups(class: ['INFREQUENT_ACCESS'])
```

L'esempio successivo seleziona i gruppi di log nell'account 111122223333 che iniziano con prefissi di nome specifici e appartengono alla classe di log Standard. La classe non è menzionata nel comando perché Standard è il valore predefinito della classe di log.

```
SOURCE logGroups(accountIdentifier:['111122223333'], namePrefix: ['namePrefix1',
 'namePrefix2'])
```

L'ultimo esempio mostra come utilizzare il SOURCE comando con il `start-query` AWS CLI comando.

```
aws logs start-query
--region us-east-1
--start-time 1729728200
--end-time 1729728215
--query-string "SOURCE logGroups(namePrefix: ['Query']) | fields @message | limit 5"
```

Interroga le fonti di dati

SOURCEPer specificare le fonti di dati da interrogare, puoi usare la `dataSource` parola chiave. È possibile includere fino a dieci fonti di dati nell'elenco.

L'esempio seguente seleziona l'origine `amazon_vpc.flow` dati.

```
SOURCE dataSource(['amazon_vpc.flow'])
```

L'esempio seguente seleziona l'origine `amazon_vpc.flow` dati e limita i gruppi di log in base al prefisso del nome del gruppo di log.

```
SOURCE dataSource(['amazon_vpc.flow']) logGroups(namePrefix: ['namePrefix1'])
```

Interroga i gruppi di log per tag

Per SOURCE filtrare i gruppi di log in base ai relativi tag, usa la `logGroupTags` funzione. Specificate i tag come elenco di filtri di tag, ciascuno con un `key values` array opzionale.

- Più filtri di tag con chiavi diverse vengono combinati con la logica AND.
- Più valori all'interno dello stesso filtro di tag vengono combinati con la logica OR.

- Utilizzatelo * per abbinare i caratteri jolly. Ad esempio, `payment*` corrisponde a valori che iniziano con`payment`.
- Utilizza `!` come prefisso per la negazione. Ad esempio, `!production` corrisponde a valori che non lo sono. `production`
- Puoi includere fino a 5 filtri di tag, ciascuno con un massimo di 5 valori.

L'esempio seguente seleziona tutti i gruppi di log contrassegnati con`team=team1 OR team=team2`.

```
SOURCE logGroupTags([{"key":"team", "values":["team1", "team2"]}])
| fields @message, @timestamp
```

L'esempio successivo seleziona i gruppi di log in cui il `service` tag inizia con `payment` e il `environment` tag no. `production`

```
SOURCE logGroupTags([{"key":"service", "values":["payment*"]}, {"key":"environment",
 "values":["!production"]}])
| fields @message, @timestamp
```

L'esempio seguente combina il filtro dei tag con un filtro per il prefisso dei nomi.

```
SOURCE logGroups(namePrefix: ['/aws/lambda']) logGroupTags([{"key":"environment",
 "values":["production"]}])
| fields @message, @timestamp
```

pattern

Utilizza `pattern` per raggruppa automaticamente in cluster i dati di log in pattern.

Un pattern è una struttura di testo condivisa che ricorre tra i campi del registro. È possibile utilizzarlo `pattern` per evidenziare le tendenze emergenti, monitorare gli errori noti e identificare le righe di registro frequenti o ad alto costo. CloudWatch Logs Insights offre anche un'esperienza da console che puoi utilizzare per trovare e analizzare ulteriormente i modelli nei tuoi eventi di registro. Per ulteriori informazioni, consulta Analisi del modello.

Poiché il `pattern` comando identifica automaticamente gli schemi comuni, è possibile utilizzarlo come punto di partenza per cercare e analizzare i log. Puoi anche combinare `pattern` con i comandi `filter`, `parse` o `sort` e identificare i pattern nelle query più precise.

Input di comandi nei pattern

Il comando `pattern` prevede uno dei seguenti input: il campo `@message`, un campo estratto creato utilizzando il comando `parse` o una stringa manipolata utilizzando una o più funzioni String.

Se CloudWatch Logs non è in grado di dedurre il tipo di dati rappresentato da un token dinamico, lo visualizza come <Token- *number* > e *number* indica dove nel pattern appare questo token rispetto agli altri token dinamici.

Esempi comuni di token dinamici includono codici di errore, indirizzi IP, timestamp e ID di richiesta.

Output di comandi nei pattern

Il comando `pattern` produce il seguente output:

- `@pattern`: un pattern è una struttura di testo condivisa che ricorre tra i campi dei log eventi. I campi che variano all'interno di uno schema, come l'ID della richiesta o il timestamp, sono rappresentati da token. Se CloudWatch i registri sono in grado di determinare il tipo di dati rappresentato da un token dinamico, il token viene visualizzato come. <*string*-*number*> *string*È una descrizione del tipo di dati rappresentato dal token. *number*Mostra dove nel modello appare questo token, rispetto agli altri token dinamici.

  CloudWatch Logs assegna la parte stringa del nome in base all'analisi del contenuto degli eventi di registro che la contengono.

  Se CloudWatch Logs non è in grado di dedurre il tipo di dati rappresentato da un token dinamico, lo visualizza come <Token- *number* > e *number* indica dove nel modello appare questo token rispetto agli altri token dinamici.

  Ad esempio, `[INFO] Request time: <Time-1> ms` è un potenziale output per il messaggio di log `[INFO] Request time: 327 ms`.
- `@ratio`: il rapporto tra i log eventi da un periodo di tempo selezionato e i gruppi di log specificati che corrispondono a un pattern identificato. Ad esempio, se metà dei log eventi nei gruppi di log e nel periodo di tempo selezionati corrispondono al pattern, `@ratio` restituisce `0.50`
- `@sampleCount`: il numero di log eventi da un periodo di tempo selezionato e i gruppi di log specificati che corrispondono a un pattern identificato.
- `@severityLabel`: la gravità o il livello del log, che indica il tipo di informazioni contenute in un log. Ad esempio, `Error`, `Warning`, `Info` o `Debug`.

Esempi

Il comando seguente identifica i log con strutture simili in gruppi di log specificati nell'intervallo di tempo selezionato, raggruppandoli per pattern e numero

```
pattern @message
```

Il comando `pattern` può essere usato in combinazione con il comando `filter`

```
filter @message like /ERROR/
| pattern @message
```

Il comando `pattern` può essere utilizzato con i comandi `parse` e `sort`

```
filter @message like /ERROR/
| parse @message 'Failed to do: *' as cause
| pattern cause
| sort @sampleCount asc
```

diff

Confronta gli eventi di registro trovati nel periodo di tempo richiesto con gli eventi di registro di un periodo di tempo precedente di uguale durata. In questo modo, puoi cercare le tendenze e scoprire se eventi di registro specifici sono nuovi.

Aggiungi un modificatore al `diff` comando per specificare il periodo di tempo con cui vuoi confrontare:

- `diff`confronta gli eventi di registro nell'intervallo di tempo attualmente selezionato con gli eventi di registro dell'intervallo di tempo immediatamente precedente.
- `diff previousDay`confronta gli eventi di registro nell'intervallo di tempo attualmente selezionato con gli eventi di registro della stessa ora del giorno precedente.
- `diff previousWeek`confronta gli eventi di registro nell'intervallo di tempo attualmente selezionato con gli eventi di registro della stessa ora della settimana precedente.
- `diff previousMonth`confronta gli eventi di registro nell'intervallo di tempo attualmente selezionato con gli eventi di registro dello stesso periodo del mese precedente.

Per ulteriori informazioni, consulta Confronta (diff) con gli intervalli di tempo precedenti.

parse

Consente `parse` di estrarre dati da un campo di registro e creare campi estratti da elaborare nella query. Il `parse` comando supporta quattro modalità: espressioni glob, espressioni regolari, logfmt e CSV.

Se `fieldName` viene omesso, `@message` viene utilizzato per impostazione predefinita. È possibile eseguire l'analisi da qualsiasi campo denominato specificando il nome del campo come primo argomento.

Se un evento di registro non corrisponde al modello specificato, lo vedrai comunque nei risultati, ma senza i campi estratti.

modalità Glob

Usa i caratteri jolly (*) come segnaposto per i valori che desideri estrarre e assegnali a campi denominati con. `as`

Sintassi

```
parse fieldName "pattern" as alias1, alias2
```

Il numero di caratteri * jolly deve essere uguale al numero di alias.

Esempi

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method, @user
```

```
parse @logStream "*/*/*/*" as env, service, instance, shard
| stats count(*) by env, service
```

Analisi concatenata

Estrai un campo, quindi analizza ulteriormente il campo estratto.

```
parse @message "url=*" as url
| parse url "/api/*/users/*" as apiVersion, userId
| display apiVersion, userId
```

Modalità Regex

Usa un'espressione regolare con gruppi di acquisizione denominati per estrarre i campi. Per informazioni sulla sintassi delle espressioni regolari, vedereSintassi delle espressioni regolari (regex) supportate.

Sintassi

```
parse fieldName /regex/
```

Utilizzate gruppi di acquisizione denominati `(?<name>...)` per definire i campi estratti.

Esempi

Usa gruppi di acquisizione denominati per estrarre i campi

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Usa un gruppo di acquisizione denominato per estrarre l'ENI da un log di flusso VPC

```
parse @message /(?<NetworkInterface>eni-.*?) /
| display NetworkInterface, @message
```

Multi-match mode

Utilizza la modalità multi-match per estrarre tutte le corrispondenze di un'espressione regolare da un campo, producendo più righe per evento di registro. Aggiungi la parola chiave `multi` dopo il modello regex.

Sintassi

```
parse fieldName /regex/ multi
```

Esempi

Estrai tutti gli indirizzi IP da una riga di registro (multi-match)

```
parse @message /(\d+\.\d+\.\d+\.\d+)/ as ip_addr multi
```

```
| stats count(*) by ip_addr
```

modalità Logfmt

Si usa `parse logfmt` per analizzare le righe di registro in formato logfmt in coppie chiave-valore. Logfmt è un formato di registrazione strutturato in cui ogni riga contiene coppie separate da spazi. `key=value`

Sintassi

```
parse fieldName logfmt as alias
```

Il risultato è una mappa a cui si accede con notazione a punti (ad esempio,). `lf.level` `lf.msg`

Esempi

```
parse @message logfmt as lf
| filter lf.level = "error"
| display lf.msg, lf.duration
```

```
parse @message logfmt as lf
| stats count(*) by lf.host
```

modalità CSV

Utilizzato `parse csv` per analizzare le righe di CSV-formatted registro in campi strutturati. Ogni valore separato da virgole viene assegnato all'alias corrispondente.

Sintassi

```
parse fieldName csv as alias1, alias2, alias3
```

Esempi

```
parse @message csv as timestamp, level, message
| filter level = "ERROR"
| display timestamp, message
```

```
parse @message csv as host, method, path, status, duration
```

```
| stats avg(duration) by method
```

Estrazione di campi JSON

Da utilizzare `json field=`*`fieldName`* per l'estrazione JSON concatenata esplicita da un campo oggetto analizzato in precedenza. Ciò consente di estrarre chiavi annidate da un campo strutturato senza dover analizzare nuovamente il messaggio non elaborato.

Sintassi

```
json field=fieldName "key.subkey" as alias
```

Esempi

```
parse @message /(?<payload>\{.*\})/ as payload
| json field=payload "user.name" as username
| display username
```

```
json field=requestContext "identity.sourceIp" as caller_ip
| stats count(*) by caller_ip
```

campi pertinenti

Utilizzato `relevantfields` per identificare automaticamente i campi dei dati di registro più pertinenti per una determinata condizione. Il comando confronta i log di base con il sottoinsieme che corrisponde alla condizione specificata e restituisce i campi classificati in base al punteggio di pertinenza.

Sintassi

```
relevantfields [field1, field2, ...] where condition
```

L'elenco dei campi è facoltativo. Se omesso, tutti i campi vengono analizzati.

Il comando restituisce i seguenti campi di output:

- `@fieldName`— Nome del campo
- `@relevanceScore`— Un punteggio che indica la pertinenza su una scala da 0 a 1

- `@topRelevanceContributors`— Per i campi categoriali, mostra le voci con le variazioni di frequenza più elevate.
- `@conditionalMedian`— Per i campi numerici, la mediana dei valori nei log che soddisfano la condizione
- `@baselineMedian`— Per i campi numerici, la mediana dei valori nei log che non soddisfano la condizione

Esempi

Analizza tutti i campi per individuare risposte API lente

```
relevantfields where Time > 400
```

Concentra l'analisi su campi specifici

```
relevantfields Controller, Path, Service where Time > 400
```

Identifica i fattori che determinano i timeout Lambda

```
relevantfields where Duration > 5000
```

espansione

Si usa `expand` per prendere un campo contenente un array JSON e creare un evento di registro separato per ogni elemento dell'array. Tutti gli altri campi dell'evento di registro originale vengono duplicati in ogni nuovo evento.

Sintassi

```
expand fieldName
```

Esempio

Se un evento di registro contiene `items = ["apple","banana","cherry"]` e`host = "web-01"`, `expand items` produce tre eventi di registro: `{items: "apple", host: "web-01"}``{items: "banana", host: "web-01"}`, e`{items: "cherry", host: "web-01"}`.

```
expand items
| stats count(*) by items, host
```

sort

Utilizza `sort` per visualizzare i log eventi in ordine crescente (`asc`) o decrescente (`desc`) di un campo specificato. Puoi utilizzare questa opzione con il comando `limit` per creare query "primi N" o "ultimi N".

L'algoritmo di ordinamento è una versione aggiornata dell'ordinamento naturale. Se si ordina in ordine crescente, viene utilizzata la seguente logica.

- Tutti i valori non numerici vengono prima di tutti i valori numerici. I valori numerici sono valori che includono solo numeri, non una combinazione di numeri e altri caratteri.
- Per i valori non numerici, l'algoritmo raggruppa caratteri numerici consecutivi e caratteri alfabetici consecutivi in blocchi separati per il confronto. Ordina le porzioni non numeriche in base ai valori Unicode e le porzioni numeriche in base alla lunghezza prima e poi in base al valore numerico.

Per ulteriori informazioni sull'ordine Unicode, vedere Elenco dei caratteri Unicode.

Ad esempio, quanto segue è il risultato di un ordinamento in ordine crescente.

```
!: >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> sorted by unicode order
#
*%04
0# >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Alphanumeric starting with numbers
5A
111A   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  Starts with more digits than 5A, so it sorted
 to be later than 5A
2345_
@ >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> 2345 is compared with @ in the unicode order,
@_
A >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Values starting with letters
A9876fghj
a12345hfh
0 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Number values
01
1
2
3
```

Se si esegue l'ordinamento in ordine decrescente, i risultati dell'ordinamento sono invertiti.

Ad esempio, la seguente query per i log di flusso di Amazon VPC individua i primi 15 trasferimenti di pacchetti tra gli host.

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

stats

Utilizza `stats` per creare visualizzazioni dei dati di log come grafici a barre, grafici a linee e grafici ad area in pila. Ciò consente di identificare in modo più efficiente i modelli nei dati di registro. CloudWatch Logs Insights genera visualizzazioni per le query che utilizzano la `stats` funzione e una o più funzioni di aggregazione.

Ad esempio, la seguente query in un gruppo di log Route 53 restituisce visualizzazioni che mostrano la distribuzione dei record di Route 53 all'ora, per tipo di query.

```
stats count(*) by queryType, bin(1h)
```

La seguente query raccoglie i valori di stato distinti per ogni servizio.

```
stats values(status) as statuses by service
```

Tutte queste query possono produrre grafici a barre. Se la query utilizza la funzione `bin()` per raggruppare i dati per un campo nel tempo, è anche possibile visualizzare grafici a linee e grafici ad area in pila.

La funzione `bin` supporta le seguenti abbreviazioni e unità di tempo. Per tutte le unità e le abbreviazioni che includono più di un carattere, è supportata l'aggiunta di `s` per il plurale. Quindi entrambi `hr` e `hrs` lavorano per specificare gli orari.

- `millisecond` `ms` `msec`
- `second` `s` `sec`
- `minute` `m` `min`
- `hour` `h` `hr`

- `day d`
- `week w`
- `month mo mon`
- `quarter q qtr`
- `year y yr`

Argomenti

- Visualizzazione dei dati di serie temporali
- Visualizzazione dei dati di log raggruppati per campi
- Utilizzo di più comandi stats in un'unica query
- Funzioni da utilizzare con le statistiche

Visualizzazione dei dati di serie temporali

Le visualizzazioni delle serie temporali funzionano per le query con le seguenti caratteristiche:

- La query contiene una o più funzioni di aggregazione. Per ulteriori informazioni, consulta Aggregation Functions in the Stats Command.
- La query utilizza la funzione `bin()` per raggruppare i dati di un campo.

Queste query possono produrre grafici a linee, grafici ad area in pila, grafici a barre e grafici a torta.

Esempi

Per un tutorial completo, consulta the section called "Esercitazione: eseguire una query che produce una visualizzazione serie temporale".

Di seguito sono riportati altri esempi di query che funzionano per la visualizzazione delle serie temporali.

La seguente query genera una visualizzazione dei valori medi del campo `myfield1`, con un punto dati creato ogni cinque minuti. Ogni punto dati è l'aggregazione delle medie dei valori `myfield1` dei log dei cinque minuti precedenti.

```
stats avg(myfield1) by bin(5m)
```

La seguente query genera una visualizzazione di tre valori basati su campi diversi, con un punto dati creato ogni cinque minuti. La visualizzazione viene generata perché la query contiene funzioni di aggregazione e usa `bin()` come campo di raggruppamento.

```
stats avg(myfield1), min(myfield2), max(myfield3) by bin(5m)
```

Restrizioni del grafico a linee e del grafico ad area in pila

Le query che aggregano le informazioni sulle voci di log, ma non utilizzano la funzione `bin()`, possono generare grafici a barre. Tuttavia, le query non possono generare grafici a linee o grafici ad area in pila. Per ulteriori informazioni su questi tipi di query, consulta the section called "Visualizzazione dei dati di log raggruppati per campi".

Visualizzazione dei dati di log raggruppati per campi

È possibile produrre grafici a barre per le query che utilizzano la funzione `stats` e una o più funzioni di aggregazione. Per ulteriori informazioni, consulta Aggregation Functions in the Stats Command.

Per visualizzare la visualizzazione, eseguire la query. Quindi scegliere la scheda Visualization (Visualizzazione) selezionare la freccia accanto a Linea (Linea), e scegliere Bar (barra). Le visualizzazioni sono limitate a un massimo di 100 barre nel grafico a barre.

Esempi

Per un tutorial completo, consulta the section called "Esercitazione: eseguire una query che produce una visualizzazione raggruppata per campi di log". I paragrafi seguenti includono ulteriori query di esempio per la visualizzazione in base ai campi.

La seguente query di log di flusso VPC trova il numero medio di byte trasferiti per sessione per ogni indirizzo di destinazione.

```
stats avg(bytes) by dstAddr
```

È inoltre possibile produrre un grafico che include più di una barra per ogni valore risultante. Ad esempio, la query del log di flusso VPC seguente trova il numero medio e massimo di byte trasferiti per sessione per ogni indirizzo di destinazione.

```
stats avg(bytes), max(bytes) by dstAddr
```

La query seguente trova il numero di registro di query Amazon Route 53 per ogni tipo di query.

```
stats count(*) by queryType
```

Utilizzo di più comandi stats in un'unica query

È possibile utilizzare più `stats` comandi in una singola query. Ciò consente di eseguire aggregazioni aggiuntive sull'output della prima aggregazione. Il numero massimo di `stats` comandi consentiti in una query dipende dalla classe di log del gruppo di log.

Esempio: interrogazione con due comandi **`stats`**

Ad esempio, la seguente query trova innanzitutto il volume di traffico totale in contenitori da 5 minuti, quindi calcola il volume di traffico più alto, più basso e medio tra questi contenitori da 5 minuti.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length)/1024/1024 as logs_mb BY bin(5m)
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb
```

Esempio: combinazione di più comandi stats con altre funzioni come **`filter`**, **`fields`** e **`bin`**

È possibile combinare più `stats` comandi con altri comandi come `filter` e `fields` in una singola query. Ad esempio, la seguente query trova il numero di indirizzi IP distinti nelle sessioni e trova il numero di sessioni per piattaforma client, filtra tali indirizzi IP e infine trova la media delle richieste di sessione per piattaforma client.

```
STATS countDistinct(client_ip) AS session_ips,
      count(*) AS requests BY session_id, client_platform
| FILTER session_ips > 1
| STATS count(*) AS multiple_ip_sessions,
        sum(requests) / count(*) AS avg_session_requests BY client_platform
```

È possibile utilizzare le funzioni `bin` e `dateceil` nelle query con più comandi `stats`. Ad esempio, la seguente query combina prima i messaggi in blocchi da 5 minuti, quindi aggrega i blocchi da 5 minuti in blocchi da 10 minuti e calcola i volumi di traffico più alti, più bassi e medi all'interno di ogni blocco di 10 minuti.

```
FIELDS strlen(@message) AS message_length
```

```
| STATS sum(message_length) / 1024 / 1024 AS logs_mb BY BIN(5m) as @t
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb BY dateceil(@t, 10m)
```

Note e limitazioni

Una query può avere un massimo di 10 `stats` comandi. Questa quota non può essere modificata.

Se si utilizza un `limit` comando `sort` or, questo deve apparire dopo l'ultimo `stats` comando. Se è precedente all'ultimo `stats` comando, la query non è valida.

Quando una query include più `stats` comandi, i risultati parziali della query non iniziano a essere visualizzati fino al completamento della prima `stats` aggregazione.

Nei `stats` comandi successivi di una singola query, è possibile fare riferimento solo ai campi definiti nel comando precedente`stats`. Ad esempio, la seguente query non è valida perché il campo `@message` non sarà disponibile dopo la prima aggregazione `stats`.

```
FIELDS @message
| STATS SUM(Fault) by Operation
# You can only reference `SUM(Fault)` or Operation at this point
| STATS MAX(strlen(@message)) AS MaxMessageSize # Invalid reference to @message
```

Tutti i campi a cui si fa riferimento dopo un `stats` comando devono essere definiti in quel `stats` comando.

```
STATS sum(x) as sum_x by y, z
| STATS max(sum_x) as max_x by z
# You can only reference `max(sum_x)`, max_x or z at this point
```

Important

La funzione `bin` utilizza sempre implicitamente il campo `@timestamp`. Ciò significa che non è possibile utilizzarlo `bin` in un `stats` comando successivo senza utilizzare il `stats` comando precedente per propagare il `timestamp` campo. Ad esempio, la query seguente non è valida.

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes BY @logStream
```

```
 | STATS avg(ingested_bytes) BY bin(5m) # Invalid reference to @timestamp field
```

Definite invece il `@timestamp` campo nel `stats` comando precedente e quindi potete utilizzarlo con `dateceil` un `stats` comando successivo, come nell'esempio seguente.

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes, max(@timestamp) as @t BY
 @logStream
 | STATS avg(ingested_bytes) BY dateceil(@t, 5m)
```

Funzioni da utilizzare con le statistiche

CloudWatch Logs Insights supporta sia le funzioni di aggregazione delle statistiche che le funzioni non di aggregazione delle statistiche.

Utilizza funzioni di aggregazione delle statistiche nel comando `stats` e come argomenti per altre funzioni.

Funzione	Tipo di risultato	Description
`avg(fieldName: NumericLogField)`	numero	La media dei valori nel campo specificato.

`count()`
 	
`count(fieldName: LogField)`
	numero	Conta i log eventi. `count()` (o `count(*)`) conta tutti gli eventi restituiti dalla query, mentre `count(fieldName)` conta tutti i registri che includono il nome di campo specificato.

`countDistinct(fieldName: LogField)`	numero	Restituisce il numero di valori univoci per il campo. Se il campo dispone di alta cardinalità (contiene molti valori univoci), il valore restituito da `countDistinct` è solo un'approssimazione.
`max(fieldName: LogField)`	LogFieldValue	Il numero massimo di valori per questo campo di log nei log di query.

Funzione	Tipo di risultato	Description
`min(fieldName: LogField)`	LogFieldValue	Il numero minimo di valori per questo campo di log nei log di query.
`pct(fieldName: LogFieldValue, percent: number)`	LogFieldValue	Un percentile indica lo stato relativo di un valore in un set di dati. Ad esempio, `pct(@duration, 95)` restituisce il valore `@duration` a cui il 95% dei valori di `@duration` sono inferiori a questo valore e il 5% sono superiori a questo valore.
`stddev(fieldName: NumericLogField)`	numero	La deviazione standard dei valori nel campo specificato.
`sum(fieldName: NumericLogField)`	numero	La somma dei valori nel campo specificato.
`values(fieldName: LogField)`	array	Raccoglie i valori distinti del campo specifica to per ogni gruppo. Conosciuto anche come`COLLECT_VALUES` .
`variance(fieldName: NumericLogField)`	numero	Varianza della popolazione dei valori nel campo specificato. Ad esempio: `stats variance(responseT ime) as rt_var by service`
`topk(k: number, fieldName: LogField)`	array	Restituisce i k valori più frequenti del campo specificato. I valori validi per k vanno da 1 a 10000. Questa funzione non può essere combinata con by altre funzioni di aggregazi one. Ad esempio: `stats topk(3, status)`

Funzioni di non aggregazione delle statistiche

Utilizza funzioni di non aggregazione nel comando `stats` e come argomenti per altre funzioni.

Funzione	Tipo di risultato	Description
`earliest(fieldName: LogField)`	LogField	Restituisce il valore di `fieldName` dal log eventi con il timestamp meno recente nei log oggetto della query.
`latest(fieldName: LogField)`	LogField	Restituisce il valore di `fieldName` dal log eventi con il timestamp più recente nei log oggetto della query.
`sortsFirst(fieldName: LogField)`	LogField	Restituisce il valore di `fieldName` che occupa la prima posizione dell'ordine nei log oggetto della query.
`sortsLast(fieldName: LogField)`	LogField	Restituisce il valore di `fieldName` che occupa l'ultima posizione dell'ordine nei log oggetto della query.

limit

Utilizza `limit` per specificare il numero di log eventi che la query deve restituire. Se si omette`limit`, per impostazione predefinita la query restituisce fino a 10.000 eventi di registro. È possibile specificare un `limit` valore fino a 100.000.

L'esempio seguente restituisce solo i 25 log eventi più recenti

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

limita qualsiasi

Si usa `limit any` *N* per interrompere la query non appena vengono trovati gli eventi di registro *N* corrispondenti, senza scansionare i dati rimanenti. In questo modo è possibile ridurre in modo significativo la quantità di dati scansionati e velocizzare le interrogazioni quando è necessario solo un campione di eventi corrispondenti anziché uno specifico set ordinato.

Normalmente`limit`, la query analizza tutti i dati nell'intervallo di tempo e quindi restituisce i risultati migliori. *N* Con`limit any`, la query viene completata in anticipo non appena vengono trovati risultati sufficienti, il che significa che gli eventi restituiti possono provenire da qualsiasi parte dell'intervallo di tempo.

Ad esempio, la seguente query restituisce 1 evento di registro e interrompe la scansione non appena trova una corrispondenza:

```
fields @message | limit any 1
```

> **Note**
>
> Poiché `limit any` interrompe anticipatamente la scansione, non è garantito che i risultati siano in un ordine particolare. Se hai bisogno di risultati ordinati, usa `limit` without `any` insieme `asort`.

dedup

Utilizza `dedup` per rimuovere i risultati duplicati in base a valori specifici nei campi indicati. Puoi utilizzare `dedup` con uno o più campi. Se specifichi un campo con `dedup`, viene restituito un solo log eventi per ogni valore univoco di tale campo. Se specifichi più campi, viene restituito un log eventi per ogni combinazione univoca di valori per tali campi.

I duplicati vengono eliminati in base al criterio di ordinamento, conservando solo il primo risultato dell'ordinamento. Ti consigliamo di ordinare i risultati prima di sottoporli al comando `dedup`. Se i risultati non vengono ordinati prima di essere sottoposti a `dedup`, viene utilizzato l'ordinamento decrescente predefinito tramite `@timestamp`.

I valori nulli non sono considerati duplicati per la valutazione. I log eventi con valori nulli per uno qualsiasi dei campi specificati vengono conservati. Per eliminare i campi con valori nulli, utilizza **`filter`** tramite la funzione `isPresent(field)`.

L'unico comando di query che puoi utilizzare in una query dopo il comando `dedup` è `limit`.

Quando si utilizza `dedup` in una query, la console visualizza un messaggio come Mostra i record X di Y, dove X è il numero di risultati deduplicati e Y è il numero totale di record corrispondenti prima della deduplicazione. Ciò indica che i record duplicati sono stati rimossi e non significa che mancano dati.

Esempio: visualizza solo il log eventi più recente per ogni valore univoco del campo denominato **`server`**

L'esempio seguente mostra i campi `timestamp`, `server`, `severity` e `message` solo per l'evento più recente per ogni valore univoco di `server`.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server
```

Per altri esempi di query di CloudWatch Logs Insights, consulta. Query generali

unmask

Utilizza `unmask` per visualizzare tutto il contenuto di un log eventi nel quale alcuni contenuti sono mascherati a causa di una policy di protezione dei dati. Per eseguire questo comando, è necessario disporre dell'autorizzazione `logs:Unmask`.

Per ulteriori informazioni sulla protezione dei dati nei gruppi di log, consulta la sezione Incremento della protezione dei dati di log sensibili con il mascheramento.

unnest

Si usa `unnest` per appiattire un elenco preso come input per produrre più record con un singolo record per ogni elemento dell'elenco. In base al numero di elementi contenuti in un campo, questo comando elimina il record corrente e genera nuovi record. Ogni record include il`unnested_field`, che rappresenta un elemento. Tutti gli altri campi provengono dal record originale.

L'input per `unnest` è`LIST`, che proviene dalla `jsonParse` funzione. Per ulteriori informazioni, vedere Tipi di struttura. Qualsiasi altro tipo, ad esempio `String` e `MAPnumbers`, viene considerato come un elenco contenente un elemento`unnest`.

Struttura di comando

L'esempio seguente descrive il formato di questo comando.

```
unnest field into unnested_field
```

Query di esempio

L'esempio seguente analizza una stringa di oggetto JSON ed espande un elenco di eventi di campo.

```
fields jsonParse(@message) as json_message
| unnest json_message.events into event
| display event.name
```

L'evento di registro per questa query di esempio potrebbe essere una stringa JSON come segue:

```
{
   "events": [
        {
            "name": "exception"
        },
        {
            "name": "user action"
        }
   ]
}
```

In questo caso, la query di esempio produce due record nel risultato della query, uno con `event.name` as `exception` e l'altro con `event.name` un'azione utente

Query di esempio

L'esempio seguente appiattisce un elenco e quindi filtra gli elementi.

```
fields jsonParse(@message) as js
| unnest js.accounts into account
| filter account.type = "internal"
```

Query di esempio

L'esempio seguente appiattisce un elenco per l'aggregazione.

```
fields jsonParse(trimmedData) as accounts
| unnest accounts into account
| stats sum(account.droppedSpans) as n by account.accountId
| sort n desc
| limit 10
```

cercare

`lookup`Da utilizzare per arricchire i risultati della query con i dati di riferimento di una tabella di ricerca. Puoi compilare una tabella di ricerca con dati CSV caricati su Amazon CloudWatch Logs o

con i risultati di una query di registro completata o annullata. Quando viene eseguita una query, il `lookup` comando confronta un campo nei tuoi eventi di registro con un campo nella tabella di ricerca e aggiunge i campi di output specificati ai risultati.

Utilizza le tabelle di ricerca per scenari di arricchimento dei dati, ad esempio la mappatura degli ID utente ai dettagli utente, dei codici prodotto alle informazioni sui prodotti o dei codici di errore alle descrizioni degli errori.

Creazione e gestione di tabelle di ricerca

Prima di poter utilizzare il `lookup` comando in una query, è necessario creare una tabella di ricerca. Puoi creare e gestire tabelle di ricerca dalla CloudWatch console o utilizzando l'API Amazon CloudWatch Logs.

Per creare una tabella di ricerca (console)

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Impostazioni, quindi scegli la scheda Registri.
3. Scorri fino alle tabelle di ricerca e scegli Gestisci.
4. Scegli Crea tabella di ricerca.
5. Inserisci un nome per la tabella di ricerca. Il nome può contenere solo caratteri alfanumerici e caratteri di sottolineatura.
6. (Opzionale) Immettere una descrizione.
7. Carica un file CSV. Il file deve includere una riga di intestazione con i nomi delle colonne, utilizzare la UTF-8 codifica e non superare i 10 MB.
8. (Facoltativo) Specificate una AWS KMS chiave per crittografare i dati della tabella.
9. Scegli Crea.

Dopo aver creato una tabella di ricerca, è possibile visualizzarla nell'editor di query di CloudWatch Logs Insights. Scegli la scheda Tabelle di ricerca per sfogliare le tabelle disponibili e i relativi campi.

Per aggiornare una tabella di ricerca, seleziona la tabella e scegli Azioni, Aggiorna. Carica un nuovo file CSV per sostituire tutto il contenuto esistente. Per eliminare una tabella di ricerca, scegli Azioni, Elimina.

Invece di caricare dati CSV, puoi anche creare o aggiornare una tabella di ricerca dai risultati di una query di registro completata o annullata che esegui nella console di Log Analytics o con

l'API Amazon Logs. CloudWatch Specificate l'ID della query con il `queryId` parametro dell'API or. `CreateLookupTable` `UpdateLookupTable` Una query annullata popola la tabella con i risultati parziali disponibili quando l'interrogazione è stata interrotta. È necessario specificare il contenuto CSV o un ID di query, ma non entrambi. L'aggiornamento di una tabella è un'operazione di sostituzione completa: tutto il contenuto esistente viene sostituito indipendentemente dalla fonte. Per aggiornare automaticamente una tabella di ricerca con i risultati delle query in base a una pianificazione, utilizzate una query pianificata con una destinazione nella tabella di ricerca. Per ulteriori informazioni, consulta Configurazione delle destinazioni delle tabelle di ricerca per le interrogazioni pianificate.

> **Note**
>
> È possibile creare fino a 100 tabelle di ricerca per account per. Regione AWS I file CSV possono pesare fino a 10 MB. Puoi anche gestire le tabelle di ricerca utilizzando l'API Amazon CloudWatch Logs. Per ulteriori informazioni, CreateLookupTableconsulta Amazon CloudWatch Logs API Reference.

> **Note**
>
> Se la tabella di ricerca è crittografata con una chiave KMS, il chiamante deve disporre dell'`kms:Decrypt`autorizzazione sulla chiave (la chiave KMS utilizzata per crittografare la tabella di ricerca) per utilizzare l'`StartQuery`API con una query che faccia riferimento a tale tabella di ricerca. Per ulteriori informazioni, consulta Crittografa le tabelle di ricerca nei CloudWatch registri utilizzando AWS Key Management Service.

Sintassi delle query per la ricerca

Struttura di comando

Di seguito viene illustrato il formato di questo comando.

```
lookup table match-fields output-mode output-field[,...]
```

Il comando utilizza i seguenti argomenti:

- *`table`*— Il nome della tabella di ricerca da utilizzare.

- *match-fields*— Specificare uno o più campi per confrontare gli eventi di registro con la tabella di ricerca. È possibile utilizzare uno dei seguenti moduli:
  - *lookup-field* `as` *log-field* `[,...]`— Da utilizzare `as` quando il nome della colonna della tabella di ricerca è diverso dal nome del campo degli eventi di registro. Ad esempio, fa `ip_address as srcAddr` corrispondere la `ip_address` colonna nella tabella di ricerca con il `srcAddr` campo negli eventi di registro.
  - *lookup-field* `[,...]`— Quando il nome del campo degli eventi di registro è lo stesso del nome della colonna della tabella di ricerca, è possibile omettere `as` e specificare direttamente il nome del campo. Ad esempio, `department, role` abbina entrambe le colonne ai campi degli eventi di registro con gli stessi nomi.

  Quando vengono specificati più campi di confronto, una riga nella tabella di ricerca deve corrispondere a tutti i campi per produrre un risultato (logica AND).
- *output-mode*— Speciifica in che modo i campi di output vengono aggiunti ai risultati. Utilizzare una delle seguenti operazioni:
  - `OUTPUT`— Aggiunge i campi di output ai risultati. Se nell'evento di registro esiste già un campo con lo stesso nome, viene sovrascritto con il valore della tabella di ricerca. Se non viene trovata alcuna corrispondenza, il campo viene impostato su null.
  - `OUTPUTNEW`— Aggiunge i campi di output ai risultati solo se il campo non esiste già nell'evento di registro. Se il campo ha già un valore, viene mantenuto il valore originale. Se non viene trovata alcuna corrispondenza, il campo rimane invariato.
- *output-field*— Uno o più campi della tabella di ricerca da aggiungere ai risultati.

Esempio: arricchire gli eventi del registro con i dettagli dell'utente

Supponiamo di avere un gruppo di log con eventi che contengono un `id` campo e una tabella di ricerca denominata `user_data` con colonne`id`,`name`, `email` e. `department` La seguente query arricchisce ogni evento di registro con il nome, l'e-mail e il reparto dell'utente presenti nella tabella di ricerca.

```
fields action, status, name, email, department
| lookup user_data id OUTPUT name, email, department
```

Esempio: utilizzare la ricerca con l'aggregazione

È possibile utilizzare i campi di output di ricerca con funzioni di aggregazione. La seguente query arricchisce gli eventi di registro con i dettagli dell'utente e quindi conta gli eventi raggruppati per indirizzo e-mail.

```
fields user_id, action, username, email, department
| lookup user_data user_id OUTPUT username, email, department
| stats count(*) by email
```

Esempio: utilizzare la ricerca con filtro

È possibile filtrare i risultati in base ai campi restituiti dalla ricerca. La seguente query arricchisce gli eventi di registro e quindi filtra per mostrare solo gli eventi di un reparto specifico.

```
fields user_id, action
| lookup user_data user_id OUTPUT username, email, department
| filter department = "Engineering"
```

Esempio: utilizzare OUTPUTNEW per arricchire senza sovrascrivere

Se i tuoi eventi di registro contengono già un `hostname` campo ma a volte è vuoto, usalo `OUTPUTNEW` per inserire i valori mancanti senza sovrascrivere quelli esistenti.

```
fields srcAddr, hostname
| lookup known_hosts ip_address as srcAddr OUTPUTNEW hostname, region
```

Esempio: utilizza la ricerca con più campi di confronto

È possibile eseguire la corrispondenza su più di un campo. La seguente query confronta entrambi `srcAddr` e `dstPort` confronta la tabella di ricerca per identificare i servizi di rete noti.

```
fields @timestamp, srcAddr, dstAddr, dstPort
| lookup network_services ip_address as srcAddr, port as dstPort OUTPUT service_name,
 owner
| filter ispresent(service_name)
```

Esempio: utilizzare la ricerca con i nomi di campo corrispondenti

Quando i nomi dei campi degli eventi di registro corrispondono esattamente ai nomi delle colonne della tabella di ricerca, puoi omettere la parola chiave. `as` La seguente query confronta entrambi `department` i `role` campi direttamente con la tabella di ricerca.

```
fields @timestamp, department, role
| lookup employees department, role OUTPUT office, manager
```

join

Combina gli eventi di registro di un gruppo di log di origine con gli eventi di un altro gruppo di log o il risultato di una query in base a un campo corrispondente.

Utilizzate il `join` comando per correlare gli eventi di registro correlati tra diverse fonti, ad esempio i gruppi di log, utilizzando chiavi comuni tra loro, come la corrispondenza degli identificatori di richiesta o degli ID delle transazioni.

Sintassi

```
join type=<join_type> left=<left_alias> right=<right_alias>
    where <left_alias>.<field>=<right_alias>.<field>
    (SOURCE <right_log_group>)
```

Parameters

- `<right_log_group>`— La fonte di dati secondaria a cui unirsi.
- `<left_alias>`e `<right_alias>` — Alias per distinguere i campi dalle fonti di dati sinistra (principale) e destra (secondaria).
- `where <field>`— specifica il campo utilizzato come chiave di unione. Il campo deve esistere in entrambe le fonti di dati.
- `type=<join_type>`(opzionale): specifica il tipo di join. I valori validi sono:
  - `inner`(impostazione predefinita) — Restituisce solo i record corrispondenti
  - `left`— Restituisce tutti i record dall'origine dati principale e i record corrispondenti dall'origine dati secondaria

Esempi

Example Esempio 1: correlazione delle richieste API Gateway con i log di esecuzione Lambda

Questo esempio mostra come unire i log di accesso di API Gateway con i log delle funzioni Lambda per correlare le richieste in entrata con la loro elaborazione in backend. Ciò è utile per la risoluzione

dei problemi dei flussi di richieste end-to-end e per identificare quali chiamate Lambda corrispondono a richieste API specifiche.

```
filter status >= 500
| join type=inner left=api right=lambda
    where api.requestId=lambda.requestId
    (SOURCE '/aws/lambda/my-function')
| fields api.requestId, api.status, api.latency, lambda.duration, lambda.memoryUsed
| sort api.latency desc
```

Questa interrogazione:

1. Interroga i log di accesso e i filtri dell'API Gateway per individuare eventuali errori del server (stato >= 500)
2. Si unisce ai log delle funzioni Lambda utilizzando `requestId` il campo che appare in entrambe le sorgenti di log
3. Utilizza gli alias (`api`and`lambda`) per distinguere i campi da ciascuna fonte
4. Restituisce informazioni combinate che mostrano la latenza dell'API insieme alla durata dell'esecuzione Lambda e all'utilizzo della memoria
5. Ordina i risultati in base alla latenza dell'API per identificare le richieste più lente

Example Esempio 2: monitora le transazioni distribuite tra microservizi

Quando si eseguono il debug di problemi in un'architettura di microservizi, spesso è necessario tracciare una transazione su più servizi. Questo esempio mostra come unire i log di due servizi diversi utilizzando un ID di transazione comune.

```
filter eventType = "ORDER_CREATED"
| join type=left left=order right=payment
    where order.transactionId=payment.transactionId
    (SOURCE '/aws/lambda/payment-service')
| filter payment.eventType = "PAYMENT_PROCESSED" or !ispresent(payment.eventType)
| fields order.transactionId, order.orderId, order.customerId,
    payment.paymentStatus, payment.amount
| filter payment.paymentStatus != "SUCCESS" or !ispresent(payment.paymentStatus)
```

Questa interrogazione:

1. Inizia con gli eventi di creazione degli ordini dal servizio ordini

2. Utilizza `left join` a per includere tutti gli ordini, anche quelli senza record di pagamento corrispondenti
3. Partecipa agli eventi di elaborazione dei pagamenti utilizzando il campo condiviso `transactionId`
4. Filtra i risultati finali per mostrare solo gli ordini con pagamenti non riusciti o record di pagamento mancanti

L'unione a sinistra è importante in questo caso perché consente di visualizzare gli ordini che sono stati creati ma che non hanno mai avuto un evento di pagamento corrispondente, il che potrebbe indicare un errore di sistema.

Comportamento

- La fonte di dati principale (lato sinistro) viene elaborata per prima.
- L'origine dati secondaria viene valutata e abbinata utilizzando la chiave di join specificata.
- La corrispondenza viene eseguita utilizzando il confronto delle uguaglianze nel campo di unione.
- Per i left join, i record non corrispondenti dell'origine dati primaria vengono conservati con valori nulli per i campi secondari.

Note e limitazioni

- Sono supportate solo le condizioni di uguaglianza (=).
- È supportato un solo comando join per query.
- Le chiavi di join devono esistere in entrambe le fonti di dati ed essere di tipi compatibili.
- Le query che utilizzano join possono scansionare più dati e comportare costi più elevati.
- Il numero di valori chiave univoci nell'origine dati secondaria è limitato a 50.000 per garantire le prestazioni delle query.
- Le sottoquery sul lato destro di join non sono supportate.

Comandi correlati

- campi

- filtro
- analizzare
- statistiche
- ordinare
- limite

subqueries

Una sottoquery è una query annidata di Logs Insights che può essere utilizzata come input per un'altra query. Le sottoquery possono essere utilizzate per derivare set di risultati intermedi che vengono poi utilizzati dai comandi successivi.

Sintassi

Sottoquery nel filtro

```
filter <field> in (
    <subquery>
)
```

Parameters

- `<subquery>`— Una query Logs Insights valida che restituisce un set di risultati. La sottoquery deve produrre campi a cui fa riferimento la query esterna.

Esempi

Example Esempio 1: trova le richieste che hanno riscontrato errori nei servizi a valle

Questo esempio mostra come utilizzare una sottoquery per identificare le richieste nel servizio principale che hanno provocato errori in un servizio a valle. Ciò è utile per la risoluzione dei problemi a catena nei sistemi distribuiti.

```
filter requestId in (
    SOURCE '/aws/lambda/database-service'
```

```
    | filter errorType = "DatabaseConnectionTimeout"
    | fields requestId
)
| fields @timestamp, requestId, endpoint, userId, responseTime
| sort @timestamp desc
```

Questa interrogazione:

1. La sottoquery trova tutti i `requestId` valori del servizio di database che ha subito dei timeout di connessione
2. La query esterna filtra i log del servizio principale per mostrare solo le richieste che corrispondono a quegli ID di richiesta soggetti a errori
3. I risultati mostrano il contesto completo delle richieste non riuscite a valle, inclusi gli endpoint e gli utenti interessati

Questo modello consente di comprendere l'impatto a monte dei guasti a valle.

Example Esempio 2: Identifica le richieste di indagini mirate che spesso non vanno a buon fine

Questo esempio dimostra l'utilizzo di una sottoquery con aggregazione per trovare le richieste che falliscono ripetutamente, il che spesso indica problemi sistematici anziché errori transitori.

```
filter requestId in (
    SOURCE '/aws/lambda/payment-processor'
    | filter status = "FAILED"
    | stats count(*) as failureCount by requestId
    | filter failureCount > 3
    | fields requestId
)
| fields @timestamp, requestId, customerId, amount, failureReason
| sort @timestamp asc
```

Questa interrogazione:

1. La sottoquery aggrega i tentativi di pagamento non riusciti e identifica gli ID delle richieste che hanno avuto esito negativo più di 3 volte
2. La query esterna recupera tutti gli eventi di registro per quegli ID di richiesta problematici
3. I risultati vengono ordinati cronologicamente per mostrare la progressione dei tentativi di nuovo tentativo

Questo aiuta a distinguere tra errori transitori (singola occorrenza) e problemi persistenti (errori multipli) che richiedono un'indagine più approfondita.

Comportamento

- Le sottoquery vengono eseguite indipendentemente dalla query esterna.
- I risultati vengono materializzati prima di essere utilizzati dalla query esterna.
- Solo i campi selezionati in modo esplicito nella sottoquery sono disponibili per la query esterna.

Note e limitazioni

- Le sottoquery devono restituire i campi a cui fa riferimento la query esterna.
- Le sottoquery annidate non sono supportate.
- Le sottoquery possono aumentare i tempi e i costi di esecuzione delle query.
- Le sottoquery correlate non sono supportate.
- L'esecuzione della query interna è limitata a 30 secondi.

Comandi correlati

- campi
- filtro
- analizzare
- statistiche
- ordinare
- limite

regressione automatica

Usa il `autoregress` comando per creare copie ritardate (riga precedente) dei valori di un campo. Questo è l'equivalente di Logs Insights lag/lead . CloudWatch

Sintassi

```
| autoregress field [AS alias] [p=start[-end]]
```

Il comando utilizza i seguenti argomenti:

- *field*— Il campo per cui creare i valori di ritardo.
- AS *alias*(Facoltativo) — Un alias per il campo di output.
- p=*N*(Facoltativo) — Profondità di ritardo singola (impostazione predefinita 1). Da utilizzare p=*N*-*M* per una serie di ritardi.

Esempio

La seguente query crea quattro campi di ritardo (`bytes_p1`through`bytes_p4`).

```
fields @timestamp, bytes
| autoregress bytes p=1-4
```

logcompare

Usa il `logcompare` comando per confrontare la finestra temporale corrente con una finestra di base spostata indietro di una durata specificata.

Sintassi

```
| logcompare timeshift duration
```

Il comando utilizza i seguenti argomenti:

- `timeshift` *duration*— La durata del passaggio all'indietro per la finestra della linea di base (ad esempio,`7d`).

Esempio

La seguente query confronta i log del giorno precedente con quelli dello stesso periodo della settimana precedente.

```
SOURCE lg start=-1d end=now
| logcompare timeshift 7d
```

riempimento

Usa il `filldown` comando per portare avanti l'ultimo valore non nullo per colmare gli spazi vuoti. È possibile specificare i nomi dei campi o utilizzare caratteri jolly. Se non si specificano campi, tutti i campi vengono compilati.

Sintassi

```
| filldown [field1 [field2 ...]]
```

Il comando utilizza i seguenti argomenti:

- *`field`*(Facoltativo) — Uno o più nomi di campo o pattern di caratteri jolly. Se omesso, tutti i campi vengono compilati.

Esempio

La seguente query riporta valori non nulli nel `host` campo e in tutti i campi corrispondenti. `cpu*`

```
fields @timestamp, host, cpu_user, cpu_system
| filldown host cpu*
```

riempimento mancante

Usa il `fillmissing` comando per inserire le righe successive per i periodi di tempo vuoti. `stats ... by bin()` Facoltativamente, puoi riempire i campi con un valore costante.

Sintassi

```
| fillmissing [with value for field [, value for field ...]]
```

Il comando utilizza i seguenti argomenti:

- `with` *`value`* `for` *`field`*(Facoltativo) — Il valore costante da assegnare al campo specificato nelle righe inserite.

Esempio

La seguente query riempie i contenitori vuoti da 1 minuto con 0 per il campo. `avg_latency`

```
fields @timestamp, latency
| stats avg(latency) as avg_latency by bin(1m)
| fillmissing with 0 for avg_latency
```

cidrlookup

Utilizzate il `cidrlookup` comando per arricchire gli eventi confrontando un campo IP con gli intervalli CIDR in una tabella di ricerca.

Sintassi

```
| cidrlookup table ipField [AS cidrColumn]
    OUTPUT|OUTPUTNEW lookupField [AS eventField] [, ...]
```

Il comando utilizza i seguenti argomenti:

- *`table`*— Il nome della tabella di ricerca contenente gli intervalli CIDR.
- *`ipField`*— Il campo dell'evento contenente l'indirizzo IP da abbinare.
- `AS` *`cidrColumn`*(Facoltativo): la colonna nella tabella di ricerca che contiene gli intervalli CIDR.
- `OUTPUT|OUTPUTNEW`— `OUTPUT` Da utilizzare per sovrascrivere i campi esistenti o `OUTPUTNEW` per riempire solo i campi vuoti.
- *`lookupField`*— Uno o più campi della tabella di ricerca da aggiungere all'evento.

Esempio

La seguente query arricchisce gli eventi con regione, proprietario e datacenter da una tabella di ricerca di rete.

```
fields @timestamp, srcAddr
| cidrlookup net_table srcAddr OUTPUT region, owner, datacenter
```

anomalo

Utilizzate il `outlier` comando per rilevare valori anomali statistici nei campi numerici in base all'intervallo interquartile (IQR). È possibile rimuovere o bloccare le righe anomale.

Sintassi

```
| outlier [action=remove|transform] [param=n] [uselower=true|false] [mark=true|
false] field [field ...]
```

Il comando utilizza i seguenti argomenti:

- `action`(Facoltativo): `remove` (elimina le righe anomale) o `transform` (si blocca al limite). Default: `transform`.
- `param`(Facoltativo) — Il moltiplicatore IQR. Default: `2.5`.
- `uselower`(Facoltativo): indica se rilevare i valori anomali inferiori. Default: `false`.
- `mark`(Facoltativo) — Se aggiungere un campo marker booleano. Default: `false`.
- *`field`*— Uno o più campi numerici per verificare la presenza di valori anomali.

Esempio

La seguente query rimuove le righe in cui cpu o memoria sono valori anomali, utilizzando un moltiplicatore IQR 3.0 con rilevamento e marcatura del limite inferiore.

```
fields @timestamp, cpu, memory
| outlier action=remove param=3.0 uselower=true mark=true cpu memory
```

accum

Usa il `accum` comando per calcolare la somma cumulativa corrente di un campo numerico.

Sintassi

```
| accum field [AS out]
```

Il comando utilizza i seguenti argomenti:

- *`field`*— Il campo numerico da accumulare.
- `AS` *`out`*(Facoltativo) — Un alias per il campo di output.

Esempio

La seguente query calcola un totale progressivo di byte.

```
fields @timestamp, bytes
| accum bytes AS totalBytes
```

appendici

Usa il `appendcols` comando per aggiungere le colonne di una sottoquery ai risultati correnti mediante la corrispondenza posizionale delle righe.

Sintassi

```
| appendcols [override=true|false] [max=n] ( subquery )
```

Il comando utilizza i seguenti argomenti:

- `override`(Facoltativo) — Indica se sovrascrivere i campi esistenti. Default: `false`.
- `max`(Facoltativo) — Numero massimo di righe da elaborare (1—100000, impostazione predefinita 10000).
- *`subquery`*— Una query completa di CloudWatch Logs Insights racchiusa tra parentesi.

Esempio

La seguente query aggiunge la latenza media per servizio da un altro gruppo di log.

```
fields svc
| stats count(*) as cnt by svc
| appendcols override=true ( SOURCE lg | stats avg(latency) as avgLat by svc )
```

sessionizzare

Usa il `sessionize` comando per raggruppare gli eventi in sessioni in base ai campi di identità e al divario di inattività. Il comando emette un campo identificativo di sessione.

Sintassi

```
| sessionize field[, field ...] [maxspan duration] [AS out]
```

Il comando utilizza i seguenti argomenti:

- *`field`*— Uno o più campi di identità in base ai quali raggruppare.

- `maxspan` *`duration`*(Facoltativo) — Intervallo massimo di inattività prima di iniziare una nuova sessione. Default: `30m`.
- `AS` *`out`*(Facoltativo) — Il nome del campo di output. Default: `session_id`.

Esempio

La seguente query raggruppa gli eventi in sessioni per utente e dispositivo con un intervallo di inattività di 1 ora.

```
fields @timestamp, userId, deviceId
| sessionize userId, deviceId maxspan 1h as my_session
```

Conta Frequenti

Usa il `countFrequent` comando (disponibile anche come`count_frequent`) per calcolare un conteggio approssimativo di ogni combinazione univoca campo-valore, ordinata in ordine decrescente. Il comando emette un campo. `_approxcount`

Sintassi

```
| countFrequent field[, field ...]
```

Il comando utilizza i seguenti argomenti:

- *`field`*— Uno o più campi per cui contare le frequenze.

Esempio

La seguente query conta la frequenza delle combinazioni di metodi e stati.

```
fields method, status
| countFrequent method, status
```

dove

Utilizzate il `where` comando come alias per il `filter` comando. Accetta sintassi e comportamento identici.

Sintassi

```
| where condition
```

Il comando utilizza i seguenti argomenti:

- *condition*— Un'espressione booleana identica a quella accettata dal `filter` comando.

Esempio

La seguente query filtra gli eventi di registro contenenti «errore».

```
fields @timestamp, @message
| where @message like /error/
```

stimare

Utilizzate il `estimate` comando per restituire i byte stimati che la query scansionerebbe nei gruppi di log e nell'intervallo di tempo selezionati, senza eseguire la query. Utilizzando i byte stimati restituiti da questo comando, è possibile rifinire la selezione del gruppo di log, l'intervallo di tempo e i filtri prima di eseguire la query. Il `estimate` comando non comporta costi di interrogazione di CloudWatch Logs Insights.

Il `estimate` comando deve essere l'ultimo comando della query.

> **Note**
>
> Il valore `estimate` restituito è approssimativo e può differire dal volume effettivo di dati analizzati quando si esegue la query.

Sintassi

```
| estimate
```

Esempio

La seguente query restituisce una stima del volume di dati, in byte, che la query scansionerebbe senza eseguirla.

```
fields @timestamp, @message
```

```
| filter @message like /error/
| estimate
```

Funzioni booleane, di confronto, numeriche, datetime e altre

CloudWatch Logs Insights supporta molte altre operazioni e funzioni nelle query, come spiegato nelle sezioni seguenti.

Argomenti

- Operatori aritmetici
- Operatori booleani
- Operatori di confronto
- Operatori numerici
- Tipi di struttura
- Funzioni DateTime
- Funzioni generali
- Funzioni JSON
- Funzioni della stringa di indirizzi IP
- Funzioni stringa

Operatori aritmetici

Le operazioni aritmetiche accettano tipi di dati numerici come argomenti e restituiscono risultati numerici. Puoi utilizzare operazioni aritmetiche nei comandi `filter` e `fields` e come argomenti per altre funzioni.

Operation	Description
`a + b`	Addizione
`a - b`	Sottrazione
`a * b`	Moltiplicazione
`a / b`	Divisione
`a ^ b`	Elevamento a potenza (`2 ^ 3` restituisce 8)

Operation	Description
`a % b`	Resto o modulo (`10 % 3` restituisce 1)

Operatori booleani

Utilizza gli operatori booleani **`and`**, **`or`** e **`not`**.

> **Note**
>
> Utilizza gli operatori booleani solo nelle funzioni che restituiscono un valore TRUE o FALSE.

Operatori di confronto

Le operazioni di confronto accettano tutti i tipi di dati come argomenti e restituiscono un risultato booleano. Utilizza operazioni di confronto nel comando `filter` e come argomenti per altre funzioni.

Operatore	Description
=	Uguale
!=	Non uguale
<	Minore di
>	Maggiore di
<=	Minore o uguale a
>=	Maggiore o uguale a

Operatori numerici

Le operazioni numeriche accettano tipi di dati numerici come argomenti e restituiscono risultati numerici. Utilizza operazioni aritmetiche nei comandi `filter` e `fields`, oltre che come argomenti per altre funzioni.

Operation	Tipo di risultato	Description
`abs(a: number)`	numero	Valore assoluto
`ceil(a: number)`	numero	Arrotonda per eccesso (l'intero più piccolo che è maggiore del valore di a)
`floor(a: number)`	numero	Arrotonda per difetto (l'intero più grande che è minore del valore di a)
`greatest(a: number, ...numbers: number[])`	numero	Restituisce il valore più grande
`least(a: number, ...numbers: number[])`	numero	Restituisce il valore di piccolo
`log(a: number)`	numero	Log naturale
`round(a: number [, d: number])`	numero	Arrotonda il valore di. a Con un argomento, arrotonda al numero intero più vicino. Con due argomenti, viene arrotondato al d numero decimale.
`sqrt(a: number)`	numero	Radice quadrata
`haversine(lat1: number, lon1: number, lat2: number, lon2: number)`	numero	Calcola la distanza del grande cerchio in chilometri tra due punti geografici specificati dalla latitudine e dalla longitudi ne in gradi.

Operation	Tipo di risultato	Description
`toNumber(fieldName: string)`	numero	Converte il valore di un campo stringa in un valore numerico.
`toInt(fieldName: string)`	numero	Converte il valore di un campo in un numero intero (32 bit).
`toLong(fieldName: string)`	numero	Converte il valore di un campo in un numero intero lungo (64 bit).
`toDouble(fieldName: string)`	numero	Converte il valore di un campo in un numero a virgola mobile a precisione doppia.

Tipi di struttura

Una mappa o un elenco è un tipo di struttura in CloudWatch Logs Insights che consente di accedere e utilizzare gli attributi per le query.

Esempio: per ottenere una mappa o un elenco

`jsonParse`Utilizzalo per analizzare un campo che è una stringa json in una mappa o in un elenco.

```
fields jsonParse(@message) as json_message
```

Esempio: per accedere agli attributi

Utilizza l'operatore dot access (map.attribute) per accedere agli elementi in una mappa. Se un attributo in una mappa contiene caratteri speciali, usa i backtick per racchiudere il nome dell'attributo (map.attributes). `special.char`).

```
fields jsonParse(@message) as json_message
| stats count() by json_message.status_code
```

Usa l'operatore di accesso tra parentesi (list [index]) per recuperare un elemento in una posizione specifica all'interno dell'elenco.

```
fields jsonParse(@message) as json_message
| filter json_message.users[1].action = "PutData"
```

Raccogli i caratteri speciali nei backtick (``) quando nel nome della chiave sono presenti caratteri speciali.

```
fields jsonParse(@message) as json_message
| filter json_message.`user.id` = "123"
```

Esempio: risultati vuoti

Le mappe e gli elenchi vengono considerati nulli per le funzioni di tipo stringa, numerico e datetime.

```
fields jsonParse(@message) as json_message
| display toupper(json_message)
```

Il risultato del confronto della mappa e dell'elenco con qualsiasi altro campo risulta. `false`

> **Note**
>
> L'utilizzo di map and list in `deduppattern`,`sort`, e `stats` non è supportato.

Funzioni DateTime

Funzioni DateTime

Utilizza le funzioni datetime nei comandi `fields` e `filter` e come argomenti per altre funzioni. Utilizza queste funzioni per creare bucket temporali per le query con funzioni di aggregazione. Utilizza periodi di tempo composti da un numero e da uno dei seguenti:

- `ms`per millisecondi
- `s`per secondi
- `m`per minuti
- `h`per ore

Ad esempio, `10m` è 10 minuti e `1h` è un'ora.

> **Note**
>
> Usa l'unità di tempo più appropriata per la tua funzione datetime. CloudWatch Logs limita la richiesta in base all'unità di tempo scelta. Ad esempio, limita 60 come valore massimo per qualsiasi richiesta che utilizza. `s` Quindi, se si specifica`bin(300s)`, CloudWatch Logs lo implementa effettivamente come 60 secondi, poiché 60 è il numero di secondi in un minuto, quindi CloudWatch Logs non utilizzerà un numero superiore a 60 con. `s` Per creare un bucket da 5 minuti, usa invece. `bin(5m)`
> Il limite di `ms` è 1000, i cappucci di `s` e `m` sono 60 e il limite di h è 24.

La tabella seguente contiene un elenco delle diverse funzioni datetime che è possibile utilizzare nei comandi di query. La tabella elenca il tipo di risultato di ciascuna funzione e contiene una descrizione di ciascuna funzione.

> **Tip**
>
> Quando crei un comando di query, puoi utilizzare il selettore temporale per selezionare un periodo di tempo per il quale desideri eseguire query. Ad esempio, puoi impostare un periodo di tempo con intervalli di 5 e 30 minuti, intervalli di 1, 3 e 12 ore oppure un intervallo di tempo personalizzato. Puoi anche impostare periodi di tempo tra date specifiche.

Funzione	Tipo di risultato	Description

`bin(period: Period)`	Time stamp	Arrotonda il valore di `@timestamp` al periodo di tempo specificato per poi troncarlo . Ad esempio, `bin(5m)` arrotonda il valore `@timestamp` ai 5 minuti più vicini.
		
		È possibile utilizzarlo per raggruppare più voci di log in un'interrogazione. L'esempio seguente restituisce il numero di eccezioni all'ora:
		
		`filter @message like /Exception/`
		`  | stats count(*) as exceptionCount`
		`by bin(1h)`

Funzione	Tipo di risultato	Description
		`| sort exceptionCount desc` La funzione `bin` supporta le seguenti abbreviazioni e unità di tempo. Per tutte le unità e le abbreviazioni che includono più di un carattere, è supportata l'aggiunta di `s` per il plurale. Quindi entrambi `hr` e `hrs` lavorano per specificare gli orari. • `millisecond ms msec` • `second s sec` • `minute m min` • `hour h hr` • `day d` • `week w` • `month mo mon` • `quarter q qtr` • `year y yr`
`datefloor(timestamp: Timestamp, period: Period)`	Time stamp	Tronca il time stamp al periodo specificato. Ad esempio, `datefloor(@timestamp, 1h)` tronca tutti i valori di `@timestamp` alla mezzora.
`dateceil(timestamp : Timestamp, period: Period)`	Time stamp	Arrotonda il time stamp al periodo specificato e quindi tronca. Ad esempio, `dateceil(@timestamp, 1h)` tronca tutti i valori di `@timestamp` all'inizio di ogni ora.
`fromMillis(fieldName: number)`	Time stamp	Interpreta il campo di input come il numero di millisecondi dall'epoca Unix e lo converte in un timestamp.

Funzione	Tipo di risultato	Description
`toMillis(fieldName: Timestamp)`	numero	Converte il timestamp trovato nel campo denominato in un numero che rappresenta i millisecondi dall'epoca di Unix. Ad esempio, `toMillis(@timestamp)` converte il timestamp `2022-01-14T13:18:0 31.000-08:00` in `1642195111000` .

`now()`	numero	Restituisce l'ora in cui è stata avviata l'elabora zione della query, in secondi epocali. Questa funzione non accetta argomenti.
		
		È possibile utilizzarla per filtrare i risultati della query in base all'ora corrente.
		
		Ad esempio, la seguente query restituisce tutti gli errori 4xx delle ultime due ore:
		
		`parse @message "Status Code: *;" as statusCode\n | filter statusCode >= 400 and statusCode <= 499 \n | filter toMillis(@timestamp) >= (now() * 1000 - 7200000)`
		
		L'esempio seguente restituisce tutte le voci di registro delle ultime cinque ore che contengono la parola `error` o `failure`
		
		`fields @timestamp, @message | filter @message like /(?i)(error|failure)/ | filter toMillis(@timestamp) >= (now() * 1000 - 18000000)`

Funzione	Tipo di risultato	Description
`parseDate(fieldName: string, format: string [, timezone: string])`	numero	Analizza una stringa di data in millisecondi di epoca utilizzando un pattern Java. DateTimeFormatter L'`timezone`argomento opzionale specifica il fuso orario da utilizzare per l'analisi. Ad esempio: `fields parseDate(field, "yyyy-MM-dd", "UTC") as epoch`
`formatDate(timestamp: LogField, format: string [, timezone: string])`	stringa	Formatta un timestamp utilizzando una stringa di formato in stile strftime. Disponibile anche come alias. `strftime` L'`timezone`argomento facoltativo specifica il fuso orario per la formattazione. Ad esempio: `fields formatDate(@timestamp, "%Y-%m-%d", "UTC") as date`

> **Note**
>
> Attualmente, CloudWatch Logs Insights non supporta il filtraggio dei log con timestamp leggibili dall'uomo.

Funzioni generali

Funzioni generali

Utilizza le funzioni generali nei comandi `fields` e `filter` e come argomenti per altre funzioni.

Funzione	Tipo di risultato	Description
`ispresent(fieldName: LogField)`	Booleano	Restituisce `true` se il campo esiste

Funzione	Tipo di risultato	Description
`coalesce(fieldName: LogField, ...fieldNames: LogField[])`	LogField	Restituisce il primo valore non nullo dall'elenco.
`case(cond1, val1, cond2, val2, ..., [default])`	LogField	Valuta le condizioni in ordine e restituisce il valore per la prima condizione vera. Se nessuna condizione è vera e viene fornita un'impost azione predefinita, restituis ce l'impostazione predefinita. Supporta fino a 10 filiali.
`if(condition: Boolean, trueValue: LogField, falseValue: LogField)`	LogField	Valuta una condizione e restituisce `trueValue` se la condizione è vera o `falseValue` meno.
`isNumeric(fieldName: LogField)`	Booleano	Restituisce `true` se il valore del campo può essere analizzato come numero. Ad esempio: `filter isNumeric(@duration)`
`messageSize(fieldName: LogField)`	numero	Restituisce la lunghezza in byte di un campo di stringa. Ad esempio: `fields messageSize(@messa ge) as size`
`queryStartTime()`	numero	Restituisce l'ora di inizio della finestra di interrogazione in millisecondi di epoca.

Funzione	Tipo di risultato	Description
`queryEndTime()`	numero	Restituisce l'ora di fine della finestra di interrogazione in millisecondi di epoca.
`queryTimeRange()`	numero	Restituisce la durata della finestra di interrogazione in millisecondi.

Funzioni JSON

Funzioni JSON

Usa le funzioni JSON nei `filter` comandi `fields` and e come argomenti per altre funzioni.

Funzione	Tipo di risultato	Description
`jsonParse(fieldName: string)`	Mappa | Elenco | Vuoto	Restituisce una mappa o un elenco quando l'input è una rappresentazione in formato stringa di un oggetto JSON o di un array JSON. Restituisce un valore vuoto, se l'input non è una delle rappresentazioni.
`jsonStringify(fieldName: Map | List)`	Stringa	Restituisce una stringa JSON da una mappa o da un elenco di dati.
`jsonArraySize(fieldName: LogField)`	numero	Restituisce il conteggio degli elementi di un campo di stringa di matrice JSON.

Funzione	Tipo di risultato	Description
		Ad esempio: `fields jsonArraySize(Operation) as arrayLen`
`jsonArrayContains(fieldName: LogField, value: string)`	Booleano	Restituisce `true` (1) se l'array JSON nel campo contiene il valore specificato. Restituisce `false` (0) per JSON non valido, non array o array vuoto. Utilizza il confronto tra maiuscole e minuscole. Ad esempio: `filter jsonArrayContains(@roles, "admin")`

Funzioni della stringa di indirizzi IP

Funzioni della stringa di indirizzi IP

Utilizza le funzioni stringa per gli indirizzi IP nei comandi `filter` e `fields`, oltre che come argomenti per altre funzioni.

Funzione	Tipo di risultato	Description
`isValidIp(fieldName: string)`	booleano	Restituisce `true` se il campo è un indirizzo IPv4 o IPv6 valido.
`isValidIpV4(fieldName: string)`	booleano	Restituisce `true` se il campo è un indirizzo IPv4 valido.
`isValidIpV6(fieldName: string)`	booleano	Restituisce `true` se il campo è un indirizzo IPv6 valido.

Funzione	Tipo di risultato	Description
`isIpInSubnet(fieldName: string, subnet: string)`	booleano	Restituisce `true` se il campo è un indirizzo IPv4 o IPv6 valido all'interno della sottorete v4 o v6 specificata. Quando si specifica la sottorete, utilizzare la notazione CIDR, come `192.0.2.0/24` o `2001:db8::/32` , dove `192.0.2.0` o `2001:db8::` è l'inizio del blocco CIDR.
`isIpv4InSubnet(fieldName: string, subnet: string)`	booleano	Restituisce `true` se il campo è un indirizzo IPv4 valido all'interno della sottorete v4 specificata. Quando si specifica la sottorete, utilizzare la notazione CIDR, come `192.0.2.0/24` , dove `192.0.2.0` è l'inizio del blocco CIDR.
`isIpv6InSubnet(fieldName: string, subnet: string)`	booleano	Restituisce `true` se il campo è un indirizzo IPv6 valido all'interno della sottorete v6 specificata. Quando si specifica la sottorete, utilizzare la notazione CIDR, come `2001:db8::/32` , dove `2001:db8::` è l'inizio del blocco CIDR.
`ipv4ToNumber(fieldName: string)`	numero	Converte una stringa di indirizzo IPv4 nella relativa rappresentazione numerica.
`isPrivateIP(fieldName: string)`	booleano	Restituisce `true` se l'indirizzo IP si trova in un intervallo privato (RFC 1918).
`isPublicIP(fieldName: string)`	booleano	Restituisce `true` se l'indirizzo IP è instradabile pubblicamente.
`isReservedIP(fieldName: string)`	booleano	Restituisce `true` se l'indirizzo IP si trova in un intervallo riservato.

Funzioni stringa

Funzioni stringa

Utilizza le funzioni stringa nei comandi `fields` e `filter`, oltre che come argomenti per altre funzioni.

Funzione	Tipo di risultato	Description
`isempty(fieldName: string)`	Numero	Restituisce 1 se il campo manca o è una stringa vuota.
`isblank(fieldName: string)`	Numero	Restituisce 1 se il campo manca, è una stringa vuota o contiene solo spazi vuoti.
`concat(str: string, ...strings: string[])`	stringa	Concatena le stringhe.

`ltrim(str: string)`
 	
`ltrim(str: string, trimChars: string)`
	stringa	Se la funzione non dispone di un secondo argomento, rimuove gli spazi vuoti dalla sinistra della stringa. Se la funzione dispone di un secondo argomento stringa, non rimuove gli spazi vuoti. Invece, rimuove i caratteri in `trimChars` dalla sinistra di `str`. Ad esempio, `ltrim("xyZxyfooxyZ","xyZ")` restituisce `"fooxyZ"`.

`rtrim(str: string)`
 	
`rtrim(str: string, trimChars: string)`
	stringa	Se la funzione dispone di un secondo argomento stringa, rimuove gli spazi vuoti dalla destra della stringa. Se la funzione dispone di un secondo argomento stringa,

Funzione	Tipo di risultato	Description
		non rimuove gli spazi vuoti. Invece, rimuove i caratteri di `trimChars` dalla destra di `str`. Ad esempio, `rtrim("xyZfooxyxyZ","xyZ")` restituisce `"xyZfoo"`.

`trim(str: string)`
 	
`trim(str: string, trimChars: string)`
	stringa	Se la funzione dispone di un secondo argomento, rimuove gli spazi vuoti da entrambe le estremità della stringa. Se la funzione dispone di un secondo argomento stringa, non rimuove gli spazi vuoti. Invece, rimuove i caratteri di `trimChars` da entrambi i lati di `str`. Ad esempio, `trim("xyZxyfooxyxyZ","xyZ")` restituisce `"foo"`.

`strlen(str: string)`	numero	Restituisce la lunghezza della stringa in punti di codice Unicode.
`toupper(str: string)`	stringa	Converte la stringa in maiuscolo.
`tolower(str: string)`	stringa	Converte la stringa in minuscolo.

Funzione	Tipo di risultato	Description

`substr(str: string, startIndex: number)`
 	
`substr(str: string, startIndex: number, length: number)`
	stringa	Restituisce una sottostringa partendo dall'indice specificato dall'argomento numero fino alla fine della stringa. Se la funzione dispone di un secondo argomento numero, contiene la lunghezza della sottostringa da recuperare. Ad esempio, `substr("xyZfooxyZ",3, 3)` restituisce `"foo"`.

`replace(fieldName: string, searchValue: string, replaceValue: string)`	stringa	Sostituisce tutte le istanze di `searchValue` in `fieldName: string` con `replaceValue` . Ad esempio, la funzione `replace(logGroup,"smoke_test","Smoke")` cerca log eventi in cui il campo `logGroup` contiene il valore della stringa `smoke_test` e sostituisce il valore con la stringa `Smoke`.
`regexReplace(fieldName: string, pattern: string, replacement: string)`	stringa	Sostituisce tutte le sottostringhe che corrispondono all'espressione regolare con. `pattern` `replacement` Utilizza la sintassi regex RE2.

Funzione	Tipo di risultato	Description

`strcontains(str: string, searchValue: string)`
 	
`strcontains(str: string, searchValue: string, caseInsensitive: boolean)`
	numero	Restituisce 1 se `str` contiene `searchValue` e 0 in caso contrario. Se il terzo parametro è impostato su`true`, la corrispondenza non fa distinzione tra maiuscole e minuscole.

`startsWith(str: string, searchValue: string)`	numero	Restituisce 1 se `str` inizia con `searchValue` e 0 in caso contrario.
`endsWith(str: string, searchValue: string)`	numero	Restituisce 1 se `str` termina con `searchValue` e 0 in caso contrario.
`urlencode(str: string)`	stringa	URL-encodes la stringa.
`urldecode(str: string)`	stringa	URL-decodes la corda.
`base64encode(str: string)`	stringa	Base64-encodes la corda.
`base64decode(str: string)`	stringa	Base64-decodes la corda.
`split(str: string, delimiter: string)`	array	Divide una stringa in base al delimitatore specificato e restituisce una matrice di sottostringhe.
`hexToAscii(value: string)`	stringa	Converte una stringa esadecimale in testo ASCII. Ad esempio: `fields hexToAscii("48656c6c6f") as text`

Funzione	Tipo di risultato	Description
`hexToDec(value: string)`	numero	Converte una stringa esadecimale in un numero decimale. Ad esempio: `fields hexToDec("0xff") as dec`
`decToHex(value: number)`	stringa	Converte un numero intero decimale in una stringa esadecimale minuscola con prefisso. `0x` I `-0x` numeri negativi producono un prefisso. Non-integers vengono troncati. Ad esempio: `fields decToHex(255) as hex`

Funzioni di hashing

È possibile utilizzare le funzioni di hashing nei comandi e. `fields` `filter`

Funzione	Tipo di risultato	Description
`md5(fieldName: string)`	stringa	Calcola l'hash MD5 del valore della stringa.
`sha256(fieldName: string)`	stringa	Calcola l' SHA-256 hash del valore della stringa.

Time-series funzioni

Usa le funzioni delle serie temporali con il `stats` comando per analizzare le metriche nelle finestre temporali e calcolare i tassi di variazione.

Funzione	Tipo di risultato	Description
`rate(fieldName: NumericLogField, interval: Period)`	numero	Calcola il tasso di variazione per intervallo per un campo numerico.
`countOverTime(fieldName: LogField)`	numero	Conta gli eventi di registro per intervallo di tempo. Usa with `by bin(interval)` per impostare la finestra.
`sumOverTime(fieldName: NumericLogField)`	numero	Somma i valori dei campi per intervallo di tempo. Usa con `by bin(interval)` per impostare la finestra.
`histogram(fieldName: NumericLogField, buckets: number)`	map	Bucketizza i valori dei campi numerici nel numero specificato di intervalli di larghezza uguale e restituisce la distribuzione.

offset

Viene utilizzato `offset` alla fine di una `stats ... by bin()` clausola per spostare i contenitori delle serie temporali entro una durata specificata. Ciò consente confronti con variazioni temporali, ad esempio il confronto delle metriche correnti con lo stesso periodo dell'ora o del giorno precedente.

Sintassi

```
stats <aggregation> by bin(<period>) offset <duration>
```

Esempi

```
stats count(*) by bin(5m) offset 1h
```

```
stats avg(latency) by bin(1m) offset 1d
```

Campi che contengono caratteri speciali

Se un campo contiene caratteri non alfanumerici diversi dal @ simbolo o dal punto (.), è necessario circondare il campo con caratteri di contrassegno (). ` Ad esempio, il campo di log `foo-bar` deve essere racchiuso tra caratteri apice inverso (`` `foo-bar` ``) perché contiene un carattere non alfanumerico, ossia il trattino (-).

Per i campi di sistema che iniziano con @ e contengono caratteri speciali nel nome rimanente, posizionate il @ simbolo fuori dai segni di spunta e racchiudete solo la parte relativa al nome del campo. Ad esempio, i campi dei tag delle risorse utilizzano il modello. `@aws.tag.<tagKey>` Se la chiave tag contiene caratteri speciali come i due punti, è necessario utilizzare la seguente sintassi:

```
@`aws.tag.aws:cloudformation:stack-name`
```

L'esempio seguente mostra una query di filtro che utilizza questa sintassi:

```
filter @`aws.tag.aws:cloudformation:stack-name` = "my-stack"
```

Utilizzo di alias e commenti nelle query

Crea query che contengono alias. Utilizza gli alias per rinominare i campi di log o per estrarre valori nei campi. Usa la parola chiave `as` per assegnare un alias a un campo di log o un risultato. Puoi utilizzare più alias in una query. Puoi utilizzare gli alias nei seguenti comandi:

- `fields`
- `parse`
- `sort`
- `stats`

Negli esempi seguenti viene illustrato come creare query che contengono alias.

Esempio

La query contiene un alias nel comando `fields`.

```
fields @timestamp, @message, accountId as ID
| sort @timestamp desc
| limit 20
```

La query restituisce i valori per i campi `@timestamp`, `@message` e `accountId`. I risultati sono ordinati in ordine decrescente e sono limitati a 20. I valori per `accountId` sono elencati sotto l'alias `ID`.

Esempio

La query contiene alias nei comandi `sort` e `stats`.

```
stats count(*) by duration as time
| sort time desc
```

La query conta il numero di volte che il campo `duration`è presente nel gruppo di log e ordina i risultati in ordine decrescente. I valori per `duration` sono elencati sotto l'alias `time`.

Utilizzo di commenti

CloudWatch Logs Insights supporta i commenti nelle query. Utilizza il carattere cancelletto (#) per impostare i commenti. Puoi utilizzare i commenti per ignorare le righe nelle query o nelle query di documenti.

Esempio: query

Quando viene emessa la seguente query, la seconda riga viene ignorata.

```
fields @timestamp, @message, accountId
# | filter accountId not like "7983124201998"
| sort @timestamp desc
| limit 20
```

Inizia a usare Logs Insights QL: tutorial sulle query

Le sezioni seguenti includono esempi di tutorial sulle query per aiutarti a iniziare a usare Logs Insights QL.

Argomenti

- Tutorial: esecuzione e modifica di una query di esempio
- Tutorial: esecuzione di una query con una funzione di aggregazione
- Esercitazione: eseguire una query che produce una visualizzazione raggruppata per campi di log
- Esercitazione: eseguire una query che produce una visualizzazione serie temporale

Tutorial: esecuzione e modifica di una query di esempio

Il seguente tutorial ti aiuta a iniziare a usare Logs Insights. CloudWatch Esegui una query di esempio in Logs Insights QL, quindi scopri come modificarla ed eseguirla nuovamente.

Per eseguire una query, è necessario che i log siano già archiviati in Logs. CloudWatch Se stai già utilizzando CloudWatch Logs e hai configurato gruppi di log e flussi di log, sei pronto per iniziare. Potresti anche avere già dei log se utilizzi servizi come AWS CloudTrail Amazon Route 53 o Amazon VPC e hai configurato i log di tali servizi in modo che vadano a Logs. CloudWatch Per ulteriori informazioni sull'invio di log a Logs, consulta. CloudWatch Guida introduttiva ai CloudWatch registri

Le query in CloudWatch Logs Insights restituiscono un insieme di campi degli eventi di registro o il risultato di un'aggregazione matematica o di un'altra operazione eseguita sugli eventi di registro. Questo tutorial illustra una query che restituisce un elenco di eventi di log.

Esecuzione di una query di esempio

Per eseguire una query di esempio di Logs CloudWatch Insights

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.

   Nella pagina Logs Insights, l'editor di query contiene una query predefinita in Logs Insights QL che restituisce i 20 eventi di registro più recenti.
3. Nell'elenco a discesa Select log group(s) (Seleziona uno o più gruppi di log), scegli uno o più gruppi di log su cui eseguire query.

   Se si tratta di un account di monitoraggio in modalità CloudWatch osservabile tra più account, è possibile selezionare gruppi di log negli account di origine e nell'account di monitoraggio. Una singola query può interrogare i log di diversi account contemporaneamente.

   È possibile filtrare i gruppi di log in base al nome del gruppo di log, all'ID account o all'etichetta dell'account.

Quando si seleziona un gruppo di log nella classe di log Standard, CloudWatch Logs Insights rileva automaticamente i campi di dati nel gruppo. Per visualizzare i campi individuati, seleziona il menu Fields (Campi) in alto a destra nella pagina.

> **Note**
>
> I campi rilevati sono supportati solo per i gruppi di log nella classe di log Standard. Per ulteriori informazioni sulle classi di log, vedereClassi di registro.

4. (Facoltativo) Utilizza il selettore temporale per selezionare un periodo di tempo per il quale desideri eseguire query.

   Puoi scegliere tra intervalli di 5 e 30 minuti, intervalli di 1, 3 e 12 ore oppure un intervallo di tempo personalizzato.

5. Scegli Run (Esegui) per visualizzare i risultati.

   Per questo tutorial, i risultati includono i 20 log eventi aggiunti più di recente.

   CloudWatch Logs visualizza un grafico a barre degli eventi di registro nel gruppo di log nel tempo. Questo grafico a barre mostra la distribuzione di eventi nel gruppo di log che corrisponde alla query e all'intervallo di tempo, non solo gli eventi visualizzati nella tabella.

6. Per visualizzare tutti i campi di un log eventi restituito, scegli l'icona triangolare del menu a discesa a sinistra dell'evento numerato.

Modifica della query di esempio

In questo tutorial, viene modificata la query di esempio per mostrare gli ultimi 50 eventi di log.

Se non hai già eseguito il tutorial precedente, fallo ora. Questo tutorial inizia da dove il tutorial precedente è terminato.

> **Note**
>
> Alcune query di esempio fornite con CloudWatch Logs Insights utilizzano i `tail` comandi `head` or invece di. `limit` Questi comandi sono obsoleti e sostituiti con `limit`. Utilizza `limit` invece di `head` o `tail` in tutte le query di tua creazione.

Per modificare la query di esempio di CloudWatch Logs Insights

1. Nell'editor di query, imposta il valore 20 su 50, e quindi scegli Run (Esegui).

   Vengono visualizzati i risultati della nuova query. Ipotizzando che il gruppo di log contenga dati sufficienti nell'intervallo di tempo predefinito, sono ora elencati 50 eventi di log.
2. (Facoltativo) Puoi salvare le query create. Per salvare questa query, scegli Save (Salva). Per ulteriori informazioni, consulta Salva ed esegui nuovamente le query di Logs Insights CloudWatch.

Aggiunta di un comando di filtro alla query di esempio

Questo tutorial mostra come apportare una modifica alla query nell'editor di query. In questo tutorial, i risultati della query precedente vengono filtrati in base a un campo negli eventi di log recuperati.

Se non hai già eseguito i tutorial precedenti, fallo in questo momento. Questo tutorial inizia da dove il tutorial precedente è terminato.

Per aggiungere un comando di filtro alla query precedente

1. Stabilisci quale campo filtrare. Per visualizzare i campi più comuni rilevati da CloudWatch Logs negli eventi di registro contenuti nei gruppi di log selezionati negli ultimi 15 minuti e la percentuale di tali eventi di registro in cui viene visualizzato ogni campo, seleziona Campi sul lato destro della pagina.

   Per visualizzare i campi contenuti in un determinato log eventi, scegli l'icona a sinistra di tale riga.

   Il campo awsRegion può essere visualizzato nel log eventi, a seconda degli eventi presenti nei log. Nel resto di questo tutorial, awsRegion viene utilizzato come campo di filtro, ma puoi utilizzarne uno diverso se tale campo non è disponibile.
2. Nella casella Editor di query, posiziona il cursore dopo 50 e premi Invio.
3. Nella nuova riga, digita innanzitutto | (il carattere barra verticale) e uno spazio. I comandi in una query di CloudWatch Logs Insights devono essere separati dal carattere pipe.
4. Specificare **`filter awsRegion="us-east-1"`**.
5. Scegli Esegui.

La query viene eseguita nuovamente e mostra ora i 50 risultati più recenti che soddisfano il nuovo filtro.

Se il filtro è stato eseguito su un campo diverso e hai ricevuto un risultato di errore, potrebbe essere necessario applicare un carattere di escape al nome campo. Se il nome campo include caratteri non alfanumerici, è necessario inserire caratteri apice inverso (`) prima e dopo il nome del campo (ad esempio, **`` `error-code`="102" ``**).

È necessario utilizzare i caratteri di apice inverso per i nomi di campo che contengono caratteri non alfanumerici, ma non per i valori. I valori sono sempre contenuti tra virgolette (").

Logs Insights QL include potenti funzionalità di interrogazione, tra cui diversi comandi e supporto per espressioni regolari, operazioni matematiche e statistiche. Per ulteriori informazioni, consulta CloudWatch Sintassi delle interrogazioni in linguaggio Logs Insights.

Tutorial: esecuzione di una query con una funzione di aggregazione

Puoi utilizzare funzioni di aggregazione con il comando `stats` e come argomenti per altre funzioni. In questo tutorial, viene eseguito un comando di query che conta il numero di log eventi contenenti un campo specificato. Il comando di query restituisce un conteggio totale raggruppato in base al valore o ai valori del campo specificato. Per ulteriori informazioni sulle funzioni di aggregazione, consulta Operazioni e funzioni supportate nella Amazon CloudWatch Logs User Guide.

Esecuzione di una query con una funzione di aggregazione

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. Conferma che la scheda Logs Insights QL sia selezionata.
4. Nell'elenco a discesa Select log group(s) (Seleziona uno o più gruppi di log), scegli uno o più gruppi di log su cui eseguire query.

   Se si tratta di un account di monitoraggio in CloudWatch modalità osservabile su più account, è possibile selezionare gruppi di log negli account di origine e nell'account di monitoraggio. Una singola query può interrogare i log di diversi account contemporaneamente.

   È possibile filtrare i gruppi di log in base al nome del gruppo di log, all'ID account o all'etichetta dell'account.

Quando si seleziona un gruppo di log, CloudWatch Logs Insights rileva automaticamente i campi di dati nel gruppo di log se si tratta di un gruppo di log di classe Standard. Per visualizzare i campi individuati, seleziona il menu Fields (Campi) in alto a destra nella pagina.

5. Elimina la query predefinita nell'editor di query e immetti il seguente comando:

```
stats count(*) by fieldName
```

6. Sostituisci *fieldName* con un campo rilevato dal menu Campi.

   Il menu Campi si trova in alto a destra della pagina e mostra tutti i campi rilevati da CloudWatch Logs Insights nel gruppo di log.

7. Scegli Run (Esegui) per visualizzare i risultati della query.

   I risultati della query mostrano il numero di record nel gruppo di log che corrispondono al comando di query e il conteggio totale raggruppato in base al valore o ai valori del campo specificato.

Esercitazione: eseguire una query che produce una visualizzazione raggruppata per campi di log

Quando si esegue una query che utilizza la funzione `stats` per raggruppare i risultati restituiti in base ai valori di uno o più campi nelle voci del log, è possibile visualizzare i risultati come grafico a barre, grafico a torta, grafico a linee o grafico ad area in pila. In questo modo è possibile visualizzare in modo più efficiente le tendenze nei log.

Per eseguire una query per la visualizzazione

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. Nell'elenco a discesa Select log group(s) (Seleziona uno o più gruppi di registro), scegli uno o più gruppi di registro su cui eseguire query.

   Se si tratta di un account di monitoraggio in CloudWatch modalità osservabile su più account, è possibile selezionare gruppi di log negli account di origine e nell'account di monitoraggio. Una singola query può interrogare i log di diversi account contemporaneamente.

   È possibile filtrare i gruppi di log in base al nome del gruppo di log, all'ID account o all'etichetta dell'account.

4. Nell'editor di query, eliminare i contenuti correnti, immettere la funzione seguente `stats` quanto segue e scegliere Run query (Esegui query).

```
stats count(*) by @logStream
    | limit 100
```

   I risultati mostrano il numero di eventi di log nel gruppo di log per ogni flusso di log. I risultati sono limitati a solo 100 righe.

5. Seleziona la scheda Visualization (Visualizzazione).
6. Seleziona la freccia accanto a Line (Linee), quindi scegli Bar (Barre).

   Viene visualizzato il grafico a barre, che mostra una barra per ogni flusso di log nel gruppo di log.

Esercitazione: eseguire una query che produce una visualizzazione serie temporale

Quando esegui una query che utilizza la funzione `bin()` per raggruppare i risultati restituiti da un periodo di tempo, puoi visualizzare i risultati come un grafico a linee, grafico ad area in pila, grafico a torta o grafico a barre. Ciò consente di visualizzare in modo più efficace le tendenze nei log eventi nel tempo.

Per eseguire una query per la visualizzazione

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. Conferma che la scheda Logs Insights QL sia selezionata.
4. Nell'elenco a discesa Select log group(s) (Seleziona uno o più gruppi di log), scegli uno o più gruppi di log su cui eseguire query.

   Se si tratta di un account di monitoraggio in CloudWatch modalità osservabile su più account, è possibile selezionare gruppi di log negli account di origine e nell'account di monitoraggio. Una singola query può interrogare i log di diversi account contemporaneamente.

   È possibile filtrare i gruppi di log in base al nome del gruppo di log, all'ID account o all'etichetta dell'account.

5. Nell'editor di query, eliminare i contenuti correnti, immettere la funzione seguente `stats` quanto segue e scegliere Run query (Esegui query).

```
stats count(*) by bin(30s)
```

I risultati mostrano il numero di eventi di registro nel gruppo di log ricevuti da CloudWatch Logs per ogni periodo di 30 secondi.

6. Seleziona la scheda Visualization (Visualizzazione).

   I risultati sono mostrati come un grafico a linee. Per passare a un grafico a barre, grafico a torta o grafico ad area in pila, scegli la freccia accanto a Line (Linea) in alto a sinistra del grafico.

Query di esempio

Questa sezione contiene un elenco di comandi di query generali e utili che è possibile eseguire nella console. CloudWatch Per informazioni su come eseguire un comando di query, consulta Tutorial: Esegui e modifica una query di esempio nella Amazon CloudWatch Logs User Guide.

Per ulteriori informazioni sulla sintassi delle query, consulta. CloudWatch Sintassi delle interrogazioni in linguaggio Logs Insights

Argomenti

- Query generali
- Query per i registri di Lambda
- Query per i flussi di log Amazon VPC
- Query per i registri di Route 53
- Interrogazioni per i log CloudTrail
- Domande per Gateway Amazon API
- Query per il gateway NAT
- Query per i registri del server Apache
- Interrogazioni per Amazon EventBridge
- Interrogazioni per i tag delle risorse
- Esempi del comando parse

Query generali

Trova i 25 log eventi aggiunti più di recente.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

Ottieni un elenco del numero di eccezioni all'ora.

```
filter @message like /Exception/
    | stats count(*) as exceptionCount by bin(1h)
    | sort exceptionCount desc
```

Ottieni un elenco di log eventi che non sono eccezioni.

```
fields @message | filter @message not like /Exception/
```

Ottieni il log eventi più recente per ogni valore univoco del campo **`server`**.

```
fields @timestamp, server, severity, message
| sort @timestamp asc
| dedup server
```

Ottieni il log eventi più recente per ogni valore univoco del campo **`server`** per ogni tipo di **`severity`**.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server, severity
```

Query per i registri di Lambda

Determina la quantità di memoria per la quale è stato effettuato un provisioning eccessivo.

```
filter @type = "REPORT"
    | stats max(@memorySize / 1000 / 1000) as provisonedMemoryMB,
        min(@maxMemoryUsed / 1000 / 1000) as smallestMemoryRequestMB,
        avg(@maxMemoryUsed / 1000 / 1000) as avgMemoryUsedMB,
        max(@maxMemoryUsed / 1000 / 1000) as maxMemoryUsedMB,
        provisonedMemoryMB - maxMemoryUsedMB as overProvisionedMB
```

Crea un report sulla latenza.

```
filter @type = "REPORT" |
```

```
    stats avg(@duration), max(@duration), min(@duration) by bin(5m)
```

Cerca le invocazioni lente delle funzioni ed elimina le richieste duplicate che possono derivare da nuovi tentativi o dal codice lato client. In questa query, **`@duration`** è espresso in millisecondi.

```
fields @timestamp, @requestId, @message, @logStream
| filter @type = "REPORT" and @duration > 1000
| sort @timestamp desc
| dedup @requestId
| limit 20
```

Query per i flussi di log Amazon VPC

Trova i primi 15 trasferimenti di pacchetti tra gli host:

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

Trova i primi 15 trasferimenti di byte per gli host su una determinata sottorete.

```
filter isIpv4InSubnet(srcAddr, "192.0.2.0/24")
    | stats sum(bytes) as bytesTransferred by dstAddr
    | sort bytesTransferred desc
    | limit 15
```

Trova gli indirizzi IP che utilizzano UDP come protocollo di trasferimento dei dati.

```
filter protocol=17 | stats count(*) by srcAddr
```

Trova gli indirizzi IP in cui i record di flusso sono stati ignorati durante la finestra di acquisizione.

```
filter logStatus="SKIPDATA"
    | stats count(*) by bin(1h) as t
    | sort t
```

Trova un singolo record per ogni connessione, per risolvere i problemi di connettività di rete.

```
fields @timestamp, srcAddr, dstAddr, srcPort, dstPort, protocol, bytes
| filter logStream = 'vpc-flow-logs' and interfaceId = 'eni-0123456789abcdef0'
| sort @timestamp desc
| dedup srcAddr, dstAddr, srcPort, dstPort, protocol
| limit 20
```

Query per i registri di Route 53

Trova la distribuzione di record all'ora in base al tipo di query.

```
stats count(*) by queryType, bin(1h)
```

Trova i 10 resolver DNS con il più elevato numero di richieste.

```
stats count(*) as numRequests by resolverIp
    | sort numRequests desc
    | limit 10
```

Trova il numero di record in base a dominio e sottodominio in cui il server non è riuscito a completare la richiesta DNS.

```
filter responseCode="SERVFAIL" | stats count(*) by queryName
```

Interrogazioni per i log CloudTrail

Trova il numero di voci di registro per ogni servizio, tipo di evento e AWS regione.

```
stats count(*) by eventSource, eventName, awsRegion
```

Trova gli host Amazon EC2 che sono stati avviati o interrotti in una determinata AWS regione.

```
filter (eventName="StartInstances" or eventName="StopInstances") and awsRegion="us-
east-2"
```

Trova le AWS regioni, i nomi utente e gli ARN degli utenti IAM appena creati.

```
filter eventName="CreateUser"
```

```
    | fields awsRegion, requestParameters.userName, responseElements.user.arn
```

Trova il numero di record in cui si è verificata un'eccezione durante il richiamo dell'API **`UpdateTrail`**.

```
filter eventName="UpdateTrail" and ispresent(errorCode)
    | stats count(*) by errorCode, errorMessage
```

Trova le voci di log in cui è stato utilizzato TLS 1.0 o 1.1

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by userIdentity.accountId, recipientAccountId,
 eventSource, eventName, awsRegion, tlsDetails.tlsVersion, tlsDetails.cipherSuite,
 userAgent
| sort eventSource, eventName, awsRegion, tlsDetails.tlsVersion
```

Trova il numero di chiamate per servizio che ha utilizzato le versioni TLS 1.0 o 1.1

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by eventSource
| sort numOutdatedTlsCalls desc
```

Domande per Gateway Amazon API

Trova gli ultimi 10 errori 4XX

```
fields @timestamp, status, ip, path, httpMethod
| filter status>=400 and status<=499
| sort @timestamp desc
| limit 10
```

Identifica le 10 Gateway Amazon API richieste che richiedono più tempo nel tuo gruppo di log di accesso Gateway Amazon API

```
fields @timestamp, status, ip, path, httpMethod, responseLatency
```

```
| sort responseLatency desc
| limit 10
```

Restituisci l'elenco dei percorsi API più comuni nel gruppo di log di accesso Gateway Amazon API

```
stats count(*) as requestCount by path
| sort requestCount desc
| limit 10
```

Crea un report sulla latenza di integrazione per il gruppo di log di accesso Gateway Amazon API

```
filter status=200
| stats avg(integrationLatency), max(integrationLatency),
min(integrationLatency) by bin(1m)
```

Query per il gateway NAT

Se noti costi superiori al normale nella tua AWS fattura, puoi utilizzare CloudWatch Logs Insights per trovare i principali contributori. Per ulteriori informazioni sui seguenti comandi di query, vedi Come posso trovare i principali contributori al traffico attraverso il gateway NAT nel mio VPC? nella pagina di supporto AWS premium.

> **Note**
>
> Nei seguenti comandi di query, sostituisci "x.x.x.x" con l'IP privato del gateway NAT e "y.y" con i primi due ottetti dell'intervallo CIDR VPC.

Trova le istanze che inviano più traffico attraverso il gateway NAT.

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Determina il traffico da e verso le istanze nei gateway NAT.

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.') or (srcAddr like 'xxx.xx.xx.xx'
 and dstAddr like 'y.y.')
```

```
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Determina le destinazioni Internet con cui le istanze del VPC comunicano più spesso per caricamenti e download.

Per i caricamenti

```
filter (srcAddr like 'x.x.x.x' and dstAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Per i download

```
filter (dstAddr like 'x.x.x.x' and srcAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Query per i registri del server Apache

È possibile utilizzare CloudWatch Logs Insights per interrogare i log del server Apache. Per ulteriori informazioni sulle seguenti domande, consulta Semplificazione dei log del server Apache con CloudWatch Logs Insights nel blog Cloud Operations & Migrations. AWS

Trova i campi più pertinenti in modo da poter rivedere i log di accesso e verificare la presenza di traffico nel percorso /admin dell'applicazione.

```
fields @timestamp, remoteIP, request, status, filename| sort @timestamp desc
| filter filename="/var/www/html/admin"
| limit 20
```

Trova il numero di richieste GET univoche che hanno effettuato l'accesso alla pagina principale con il codice di stato "200" (operazione riuscita).

```
fields @timestamp, remoteIP, method, status
| filter status="200" and referrer= http://34.250.27.141/ and method= "GET"
| stats countDistinct(remoteIP) as UniqueVisits
```

```
| limit 10
```

Trova il numero di volte in cui il servizio Apache è stato riavviato.

```
fields @timestamp, function, process, message
| filter message like "resuming normal operations"
| sort @timestamp desc
| limit 20
```

Interrogazioni per Amazon EventBridge

Ottieni il numero di EventBridge eventi raggruppati per tipo di dettaglio dell'evento

```
fields @timestamp, @message
| stats count(*) as numberOfEvents by `detail-type`
| sort numberOfEvents desc
```

Interrogazioni per i tag delle risorse

Filtra i log in base al valore di un tag di risorsa.

```
filter @aws.tag.env = "production"
```

Filtra in base a una chiave di tag che contiene caratteri speciali.

```
filter @`aws.tag.aws:cloudformation:stack-name` = "my-stack"
```

Ottieni un conteggio degli eventi di registro raggruppati in base al valore di un tag.

```
stats count(*) by @aws.tag.team
```

Esempi del comando parse

Utilizza un'espressione glob per estrarre i campi **`@user`**, **`@method`** e **`@latency`** dal campo di log **`@message`** e restituire la latenza media per ogni combinazione univoca di **`@method`** e **`@user`**.

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
```

```
    @user
```

Utilizza un'espressione regolare per estrarre i campi temporanei **@user2**, **@method2** e **@latency2** dal campo di log **@message** e restituire la latenza media per ogni combinazione univoca di **@method2** e **@user2**.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Estrae i campi **loggingTime**, **loggingType** e **loggingMessage**, filtra per log eventi che contengono le stringhe **ERROR** o **INFO** e quindi mostra solo i campi **loggingMessage** e **loggingType** per gli eventi che contengono una stringa **ERROR**.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

Confronta (diff) con gli intervalli di tempo precedenti

Puoi utilizzare CloudWatch Logs Insights con Logs Insights QL per confrontare le modifiche degli eventi di registro nel tempo. È possibile confrontare gli eventi di registro acquisiti durante un intervallo di tempo recente con i registri del periodo immediatamente precedente. In alternativa, è possibile effettuare il confronto con periodi di tempo precedenti simili. Questo può aiutarti a scoprire se un errore nei registri è stato introdotto di recente o se si è già verificato e può aiutarti a individuare altre tendenze.

Le query di confronto restituiscono solo modelli nei risultati, non eventi di log non elaborati. I modelli restituiti ti aiuteranno a visualizzare rapidamente le tendenze e le modifiche negli eventi di registro nel tempo. Dopo aver eseguito una query di confronto e aver ottenuto i risultati del pattern, puoi visualizzare esempi di eventi di log non elaborati per i pattern che ti interessano. Per ulteriori informazioni sui modelli di log, consultaAnalisi del modello.

Quando si esegue una query di confronto, la query viene analizzata in base a due diversi periodi di tempo: il periodo di interrogazione originale selezionato e il periodo di confronto. Il periodo di confronto ha sempre la stessa durata del periodo di interrogazione originale. Gli intervalli di tempo predefiniti per i confronti sono i seguenti.

- Periodo precedente: viene confrontato con il periodo di tempo immediatamente precedente al periodo di tempo della query.
- Giorno precedente: viene confrontato con il periodo di tempo del giorno precedente al periodo di tempo della query.
- Settimana precedente: viene confrontato con il periodo di tempo della settimana precedente al periodo di tempo della query.
- Mese precedente: viene confrontato con il periodo di tempo di un mese precedente al periodo di tempo della richiesta.

> **Note**
>
> Le query che utilizzano i confronti comportano costi simili all'esecuzione di una singola query di CloudWatch Logs Insights nell'intervallo di tempo combinato. Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

Per eseguire una query di confronto

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs, quindi Logs Insights.

   Nella casella di interrogazione viene visualizzata un'interrogazione predefinita.
3. Conferma che la scheda Logs Insights QL sia selezionata.
4. Mantieni la query predefinita o inserisci una query diversa.
5. Nel menu a discesa Seleziona gruppi di log, scegli uno o più gruppi di log da interrogare.
6. (Facoltativo) Utilizza il selettore temporale per selezionare un periodo di tempo per il quale desideri eseguire query. L'interrogazione predefinita riguarda l'ora precedente di dati di registro.
7. Nel selettore dell'intervallo di tempo, scegli Confronta. Quindi scegli il periodo di tempo precedente con cui vuoi confrontare i log originali e scegli Applica.
8. Scegli Esegui query.

   Per fare in modo che la query recuperi i dati del periodo di confronto, il `diff` comando viene aggiunto alla query.
9. Scegli la scheda Patterns per vedere i risultati.

La tabella mostra le seguenti informazioni:

- Ogni pattern, con parti variabili del pattern sostituite dal simbolo del token dinamico<*string*-*number*>. *string*È una descrizione del tipo di dati che il token rappresenta. *number*Mostra dove nel modello appare questo token, rispetto agli altri token dinamici. Per ulteriori informazioni, consulta Analisi del modello.
- Il conteggio degli eventi è il numero di eventi di registro con quel pattern nel periodo di tempo originale, più attuale.
- Il conteggio degli eventi di differenza è la differenza tra il numero di eventi di registro corrispondenti nel periodo di tempo corrente e il periodo di confronto. Un valore diverso positivo indica che vi sono più eventi di questo tipo nel periodo di tempo corrente.
- La descrizione della differenza riassume brevemente la variazione di tale schema tra il periodo di tempo corrente e il periodo di confronto.
- Il tipo di severità è la gravità probabile degli eventi di registro con questo schema, in base a parole trovate negli eventi di registro come`FATAL`, `ERROR` e. `WARN`

10. Per esaminare ulteriormente uno dei modelli nell'elenco, scegliete l'icona nella colonna Ispeziona per uno dei modelli.

    Viene visualizzato il riquadro Pattern Inspect che mostra quanto segue:

    - Il modello. Seleziona un token all'interno del pattern per analizzare i valori di quel token.
    - Un istogramma che mostra il numero di occorrenze del pattern nell'intervallo di tempo richiesto. Questo può aiutarvi a identificare tendenze interessanti, ad esempio un aumento improvviso della ricorrenza di un pattern.
    - La scheda Registra esempi mostra alcuni degli eventi di registro che corrispondono al modello selezionato.
    - La scheda Valori dei token mostra i valori del token dinamico selezionato, se ne è stato selezionato uno.

    > **Note**
    >
    > Per ogni token vengono acquisiti un massimo di 10 valori di token. Il conteggio dei token potrebbe non essere preciso. CloudWatch Logs utilizza un contatore probabilistico per generare il conteggio dei token, non il valore assoluto.

- La scheda Schemi correlati mostra altri motivi che si sono verificati frequentemente più o meno nello stesso periodo del modello che si sta esaminando. Ad esempio, se lo schema di un `ERROR` messaggio era in genere accompagnato da un altro evento di registro contrassegnato `INFO` con dettagli aggiuntivi, tale modello viene visualizzato qui.

Visualizzazione dei dati di log nei grafici

È possibile utilizzare visualizzazioni come grafici a barre, grafici a linee e grafici ad area in pila per identificare in modo più efficiente i modelli nei dati di registro. CloudWatch Logs Insights genera visualizzazioni per le query che utilizzano la `stats` funzione e una o più funzioni di aggregazione. Per ulteriori informazioni, consulta stats.

OpenSearch Piped Processing Language (PPL)

Questa sezione contiene un'introduzione di base all'interrogazione dei CloudWatch log utilizzando PPL. OpenSearch Con PPL, è possibile recuperare, interrogare e analizzare i dati utilizzando comandi concatenati, semplificando la comprensione e la composizione di interrogazioni complesse. La sua sintassi si basa sulle pipe Unix e consente il concatenamento di comandi per trasformare ed elaborare i dati. Con PPL, puoi filtrare e aggregare i dati e utilizzare un ricco set di funzioni matematiche, di stringhe, di data, condizionali e di altro tipo per l'analisi.

L'inclusione `SOURCE` in una query PPL è un modo utile per specificare i gruppi di log, gli indici di campo e le fonti di dati da includere in una query quando si utilizza l' AWS CLI API o per creare una query. Il `SOURCE` comando è supportato solo nell'API AWS CLI and, non nella console. CloudWatch Quando si utilizza la CloudWatch console per avviare una query, si utilizza l'interfaccia della console per specificare i gruppi di log e il nome e il tipo di origine dati.

`aws:fieldIndex`Da utilizzare per restituire solo dati indicizzati, forzando una query a scansionare solo i gruppi di log indicizzati su un campo specificato nella query. I gruppi di log pertinenti vengono selezionati automaticamente, in base ai campi specificati nel comando. `filterIndex` Ciò riduce il volume scansionato ignorando i gruppi di log che non contengono eventi di registro contenenti il campo specificato nella query e analizzando solo i gruppi di log che corrispondono al valore specificato nella query per questo indice di campo. `aws:fieldIndex`Utilizzatelo per specificare il nome del campo, insieme al nome e al valore del campo nel comando di origine, per interrogare solo i dati indicizzati contenenti il campo e il valore specificati. Per ulteriori informazioni, consulta Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione

È possibile utilizzare OpenSearch PPL per le interrogazioni dei gruppi di log nella Standard Log Class.

> **Note**
>
> Per informazioni su tutti i comandi di interrogazione OpenSearch PPL supportati nei CloudWatch registri e informazioni dettagliate sulla sintassi e le restrizioni, vedere Comandi PPL supportati nella Service Developer Guide. OpenSearch
> Per informazioni su altri linguaggi di interrogazione che è possibile utilizzare, vedere CloudWatch Logs Insights, Service SQL e OpenSearch CloudWatch Metrics Insights

Comando o funzione	Query di esempio	Description
`fields`	`fields field1, field2`	Visualizza un insieme di campi che necessitano di proiezione.
`join`	`LEFT JOIN left=l, right=r on l.id = r.id` `` `join_right_lg` `` `| fields l.field_1, r.field_2`	Unisce due set di dati.
`where`	`where field1="success" | where field2 != "i-023fe0a90929d8822" | fields field3, field4, field5,field6 | head 1000`	Filtra i dati in base alle condizioni specificate.
`aws:field Index`	`source = [` `` `aws:fieldIndex` `` `="region",` `` `region` `` `= "us-west-2"] | where status = 200 | head 10`	Restituisce solo dati indicizzati, forzando una query a scansionare solo i gruppi di log indicizzati su un

Comando o funzione	Query di esempio	Description
		campo specifica to nella query.
`stats`	`stats count(), count(field1), min(field 1), max(field1), avg(field1) by field2 | head 1000`	Esegue aggregazioni e calcoli
`parse`	`parse field1 ".*/(?<field2>[^/]+$)" | where field2 = "requestId" | fields field1, field2 | head 1000`	Estrae un pattern di espressione regolare (regex) da una stringa e visualizza il pattern estratto. Il pattern estratto può essere ulteriormente utilizzato per creare nuovi campi o filtrare i dati.
`sort`	`stats count(), count(field1), min(field 1) as field1Alias, max(`field1`), avg(`field1`) by field2 | sort -field1Al ias | head 1000`	Ordina i risultati visualizzati in base al nome di un campo. Usa sort - FieldName per ordinare in ordine decrescen te.

Comando o funzione	Query di esempio	Description
`eval`	`eval field2 = field1 * 2 | fields field1, field2 | head 20`	Modifica o elabora il valore di un campo e lo memorizza in un campo diverso. Ciò è utile per modificar e matematic amente una colonna, applicare funzioni di stringa a una colonna o applicare funzioni di data a una colonna.
`rename`	`rename field2 as field1 | fields field1;`	Rinomina uno o più campi nel risultato della ricerca.
`head`	`fields `@message` | head 20`	Limita i risultati dell'interrogazion e visualizzati alle prime N righe.
`top`	`top 2 field1 by field2`	Trova i valori più frequenti per un campo.

Comando o funzione	Query di esempio	Description
dedup	`dedup field1 | fields field1, field2, field3`	Rimuove le voci duplicate in base ai campi specificati.
rare	`rare field1 by field2`	Trova i valori meno frequenti di tutti i campi nell'elenco dei campi.
subquery	`where field_1 IN [ search source= `subquery_lg` | fields field_2 ] | fields id, field_1`	Esegue interrogazioni complesse e annidate all'interno delle istruzioni PPL.
trendline	`trendline sma(2, field1) as field1Alias`	Calcola le medie mobili dei campi.

Comando o funzione	Query di esempio	Description
eventStats	`eventstats sum(field1) by field2`	Arricchisce i dati degli eventi con statistic he riassunti ve calcolate. Analizza i campi specifici all'inter no degli eventi, calcola varie misure statistic he e quindi aggiunge questi risultati a ciascun evento originale come nuovi campi.
expand	`eval tags_array_string = json_extr act(`@message`, '$.tags')\| eval tags_array = json_array(json_ex tract(tags_string, '$[0]'), json_extr act(tags_string, '$[1]'))| expand tags_array as color_tags`	Suddivide un campo contenent e più valori in righe separate, creando una nuova riga per ogni valore nel campo specifica to.
fillnull	`fields `@timestamp`, error_code, status_code | fillnull using status_code = "UNKNOWN", error_code = "UNKNOWN"`	Riempie i campi nulli con il valore fornito. Può essere utilizzat o in uno o più campi.

Comando o funzione	Query di esempio	Description
`flatten`	`eval metadata_struct = json_object('size', json_extract(metadata_string, '$.size'), 'color', json_extract(metadata_string, '$.color')) | flatten metadata_struct as (meta_size, meta_color)`	Appiattisce un campo. Il campo deve essere di questo tipo: `struct<?,?>` o. `array<struct<?,?>>`
`cidrmatch`	`where cidrmatch(ip, '2003:db8::/32') | fields ip`	Verifica se l'indirizzo IP specificato rientra nell'intervallo CIDR specificato.
`fieldsummary`	`where field1 != 200 | fieldsummary includefields= field1 nulls=true`	Calcola le statistiche di base per ogni campo (conteggio, conteggio distinto, min, max, avg, stddev e mean).
`grok`	`grok email '.+@%{HOSTNAME:host}' | fields email, host`	Analizza un campo di testo con uno schema grok e aggiunge i risultati al risultato della ricerca.

Comando o funzione	Query di esempio	Description
Funzioni stringa	`eval field1Len = LENGTH(field1) | fields field1Len`	Built-in funzioni in PPL in grado di manipolare e trasformare stringhe e dati di testo all'interno di query PPL. Ad esempio, convertire maiuscole e minuscole, combinare stringhe, estrarre parti e pulire il testo.
Date-Time funzioni	`eval newDate = ADDDATE(DATE('2020-08-26'), 1) | fields newDate`	Built-in funzioni per la gestione e la trasformazione dei dati di data e ora nelle query PPL. Ad esempio, date_add, date_format, datediff, date-sub, timestamp add, timestamp diff, current_timezone, utc_timestamp e current_date.

Comando o funzione	Query di esempio	Description
Funzioni di condizione	`eval field2 = isnull(field1) | fields field2, field1, field3`	Built-in funzioni che verifican o le condizion i specifiche dei campi e valutano le espressioni in modo condizion ale. Ad esempio, se field1 è nullo, restituisci field2.
Funzioni matematiche	`eval field2 = ACOS(field1) | fields field1`	Built-in funzioni per eseguire calcoli e trasformazioni matematiche nelle interroga zioni PPL. Ad esempio, abs (valore assoluto) , round (arrotond a i numeri), sqrt (radice quadrata) , pow (calcolo della potenza) e ceil (arrotonda al numero intero più vicino).
CryptoGraphic funzioni	`eval crypto = MD5(field)| head 1000`	Per calcolare l'hash di un determinato campo

Comando o funzione	Query di esempio	Description
Funzioni JSON	`eval valid_json = json('[1,2,3,{"f1":1,"f2":[5,6]},4]') | fields valid_json`	Built-in funzioni per la gestione di JSON tra cui array, estrazione e convalida. Ad esempio, json_object, json_array, to_json_string, json_array_length, json_extract, json_keys e json_valid.

Ambito della query

L'inclusione di SOURCE in una query è un modo utile per specificare i gruppi di log da includere in una query quando si utilizza l'API AWS CLI or per creare una query. Il comando SOURCE è supportato solo nell'API AWS CLI and, non nella CloudWatch console. Quando si utilizza la CloudWatch console per avviare una query, si utilizza l'interfaccia della console per specificare i gruppi di log e il nome e il tipo di origine dati.

Il comando source di PPL ora supporta diversi modi per specificarli:

1. Gruppo di log
2. Indici di campo - Nuovi
3. Fonte e tipo di dati: nuovo

Gruppo di log

La selezione della fonte del gruppo di log può essere utilizzata quando i clienti sanno esattamente quali gruppi di log devono essere cercati

```
source = [lg:`/aws/lambda/my-function`] | where status = 200 | head 10
```

Indici di campo

La selezione dell'origine basata sull'indice dei campi riduce la quantità di dati interrogati limitando i risultati ai soli dati indicizzati quando i filtri riguardano i campi che sono stati indicizzati. I gruppi di log pertinenti vengono selezionati automaticamente, in base ai campi specificati nel comando. `filterIndex` Per ulteriori informazioni sugli indici di campo e su come crearli, consulta Creare indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione.

`aws:fieldIndex`Da utilizzare per restituire solo dati indicizzati, forzando una query a scansionare solo i gruppi di log indicizzati su un campo specificato nella query. Per questi gruppi di log indicizzati in questo campo, ottimizza ulteriormente la query ignorando i gruppi di log che non contengono eventi di registro contenenti il campo specificato nella query per il campo indicizzato. Riduce ulteriormente il volume scansionato tentando di analizzare solo gli eventi di registro di questi gruppi di log che corrispondono al valore specificato nella query per questo indice di campo. Per ulteriori informazioni sugli indici di campo e su come crearli, consulta Creare indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione.

In PPL, `aws:fieldIndex` viene utilizzato per specificare quali coppie chiave-valore devono essere trattate come indici. La sintassi è la seguente

```
source = [`aws:fieldIndex`="region", `region` = "us-west-2"] | where status = 200 |
 head 10
```

dove:

1. `` `aws:fieldIndex`="region" ``identifica la regione come campo Index.
   a. Nota: invece di = i clienti possono utilizzare IN per specificare più indici (esempio seguente)
2. `` `region`="us-west-2" ``identifica la condizione del filtro da applicare
   a. Nota: invece di = i clienti possono utilizzare IN per specificare più valori (esempio seguente)

I clienti possono specificare più FieldIndex come segue

```
source = [`aws:fieldIndex` IN ("status", "region"), `status` = 200, `region` IN ("us-
west-2", "us-east-1")] | head 10
```

Fonte e tipo di dati

La selezione della fonte di dati e della fonte basata sul tipo può essere utilizzata quando i clienti sanno quali fonti di dati esatte devono essere interrogate. Questa query viene eseguita su uno o più gruppi di log che contengono l'origine e il tipo di dati specificati.

```
source = [ds:`data_source.type`] | where status = 200 | head 10
```

PPL supportato per le interrogazioni sulle fonti di dati

Per supportare il caso d'uso per l'interrogazione di sorgenti di dati in PPL, è possibile utilizzare la clausola Dynamic Source Selector. Utilizzando questa sintassi, è possibile interrogare le fonti di dati specificandole nel comando di ricerca. È possibile specificare fino a 10 fonti di dati.

Sintassi

```
source=[ds:`DataSource1.Type1`, ds:`DataSource2.Type2`, ...ds:`DataSourcen.Typen`]
```

Query di esempio

```
search source=[ds:`DataSource1.Type1`, ds:`DataSource2.Type2`] | fields field1, field2
```

Esempio per la combinazione

I clienti possono specificare tutti gli operatori di selezione della sorgente in qualsiasi ordine e i risultati sarebbero l'intersezione di tutte le condizioni applicate.

Ad esempio,/aws/lambda/my-function-1 potrebbe contenere più fonti e tipi di dati, inclusa un'ampia varietà di indici. Quando viene eseguita la seguente query, i risultati restituiti avranno solo eventi di origine e tipo DataSource1.Type1 e corrispondenti ai criteri di «status» = 200.

```
search source=[
    ds:`DataSource1.Type1`,
    lg:`/aws/lambda/my-function-1`,
    `aws:fieldIndex` IN ("status"), `status` = 200
]
```

Restrizioni

Le seguenti restrizioni si applicano quando si utilizza OpenSearch PPL per eseguire query in Logs Insights. CloudWatch

- Non è possibile utilizzare i comandi join o subquery con le query relative alle fonti di dati.

OpenSearch Structured Query Language (SQL)

Questa sezione contiene un'introduzione di base all'interrogazione dei CloudWatch log utilizzando OpenSearch SQL. Fornisce un'opzione familiare se sei abituato a lavorare con database relazionali. OpenSearch SQL offre un sottoinsieme di funzionalità SQL, il che lo rende una buona scelta per eseguire query ad hoc e attività di analisi dei dati. Con OpenSearch SQL, è possibile utilizzare comandi come SELECT, FROM, WHERE, GROUP BY, HAVING e vari altri comandi e funzioni SQL. È possibile eseguire JOIN tra gruppi di log, correlare i dati tra gruppi di log utilizzando sottoquery e utilizzare il ricco set di funzioni SQL JSON, matematiche, di stringa, condizionali e di altro tipo per eseguire analisi approfondite sui dati di log e di sicurezza.

Consente `filterIndex` di restituire solo dati indicizzati, forzando una query a scansionare solo i gruppi di log indicizzati su un campo specificato nella query. Riduci il volume scansionato ignorando i gruppi di log che non contengono eventi di registro contenenti il campo specificato nella query e analizzando solo i gruppi di log che corrispondono al valore specificato nella query per questo indice di campo. `filterIndex`Utilizzatelo per specificare il nome del campo, insieme al nome e al valore del campo, per interrogare solo i dati indicizzati contenenti il campo e il valore specificati.

È possibile utilizzare OpenSearch SQL per le interrogazioni dei gruppi di log nella classe di log standard. SQL supporta anche l'esecuzione di query utilizzando il nome dell'origine dati e il tipo di origine dati.

> **Note**
>
> Nella tabella seguente sono elencati i comandi e le funzioni SQL supportati nei CloudWatch registri. Per informazioni su tutti i comandi OpenSearch SQL, inclusa la sintassi, vedere Comandi SQL supportati nella OpenSearch Service Developer Guide.
> Per informazioni sugli altri linguaggi di query che è possibile utilizzare, vedere CloudWatch Logs Insights, OpenSearch Service PPL e. CloudWatch Metrics Insights

Comandi SQL supportati

> **Note**
>
> Nella colonna della query di esempio, *<logGroup>* sostituiscila secondo necessità a seconda della fonte di dati su cui stai interrogando.

Comando o funzione	Query di esempio	Description
SELECT	`SELECT \`@message\`, Operation FROM \`LogGroupA\``	Visualizza i valori proiettati.
FROM	`SELECT \`@message\`, Operation FROM \`LogGroupA\``	Built-in clausola che specifica le tabelle o le viste di origine da cui recuperare i dati, supportando vari tipi di join e sottoquery.
WHERE	`SELECT * FROM \`LogGroupA\` WHERE Operation = 'x'`	Filtra gli eventi di registro in base ai criteri di campo forniti.
FilterIndex	`SELECT * FROM \`filterIndex('region' = 'us-east-1')\` WHERE status = 200 LIMIT 10;`	Restituisce solo dati indicizza ti, forzando una query a scansionare solo i gruppi di log indicizzati in un campo specifica to nella query.

Comando o funzione	Query di esempio	Description
GROUP BY	`SELECT `@logStream`, COUNT(*) as log_count FROM `LogGroupA` GROUP BY `@logStream``	I gruppi registrano gli eventi in base alla categoria e trovano la media in base alle statistiche.
HAVING	`SELECT `@logStream`, COUNT(*) as log_count FROM `LogGroupA` GROUP BY `@logStream` HAVING log_count > 100`	Filtra i risultati in base alle condizioni di raggruppamento.
ORDER BY	`SELECT * FROM `LogGroupA` ORDER BY `@timestamp` DESC`	Ordina i risultati in base ai campi della clausola ORDER BY. È possibile ordinare in ordine decrescente o crescente.
JOIN	`SELECT A.`@message`, B.`@timestamp`FROM `LogGroupA` as A INNER JOIN `LogGroupB` as B ON A.`requestId` = B.`requestId``	Unisce i risultati di due tabelle in base a campi comuni. È necessario specificare Inner JOIN o Left Outer Join

Comando o funzione	Query di esempio	Description
LIMIT	`Select * from `LogGroupA` limit 10`	Limita i risultati della query visualizzati alle prime N righe.
Funzioni stringa	`SELECT upper(Operation) , lower(Operation), Operation FROM `LogGroupA``	Built-in funzioni in SQL in grado di manipolare e trasformare stringhe e dati di testo all'interno di query SQL. Ad esempio, la conversione di maiuscole e minuscole, la combinazione di stringhe, l'estrazione di parti e la pulizia del testo.
Funzioni di data	`SELECT current_date() as today, date_add(current_date(), 30) as thirty_days_later, last_day(current_date()) as month_end FROM `LogGroupA``	Built-in funzioni per la gestione e la trasformazione dei dati di data e ora nelle query SQL. Ad esempio, date_add, date_format, datediff e current_date.

Comando o funzione	Query di esempio	Description
Funzioni condizionali	`SELECT Operation, IF(Error > 0, 'High', 'Low') as error_category FROM `LogGroup A`;`	Built-in funzioni che eseguono azioni in base a condizion i specifiche o che valutano le espressioni in modo condizion ale. Ad esempio, CASE e IF.
Funzioni di aggregazione	`SELECT AVG(bytes) as bytesWritten FROM `LogGroupA``	Built-in funzioni che eseguono calcoli su più righe per produrre un unico valore riepilogato. Ad esempio, SUM, COUNT, AVG, MAX e MIN.

Comando o funzione	Query di esempio	Description
Funzioni JSON	`SELECT get_json_object(json_column, '$.name') as name FROM `LogGroupA``	Built-in funzioni per l'analisi, l'estrazione, la modifica e l'interrogazione JSON-form atted dei dati all'interno delle query SQL (ad esempio, from_json , to_json, get_json_object, json_tuple) che consentono la manipolazione delle strutture JSON nei set di dati.

Comando o funzione	Query di esempio	Description
Funzioni di array	`SELECT scores, size(scores) as length, array_contains(scores, 90) as has_90 FROM `LogGroupA`;`	Built-in funzioni per lavorare con colonne di tipo array nelle query SQL, che consentono operazioni come l'accesso, la modifica e l'analisi dei dati dell'array (ad esempio, size, explode, array_contains).
Funzioni finestra	`SELECT field1, field2, RANK() OVER (ORDER BY field2 DESC) as field2Rank FROM `LogGroupA`;`	Built-in funzioni che eseguono calcoli su un insieme specifico di righe relative alla riga (finestra) corrente, abilitando operazioni come la classificazione, l'esecuzione dei totali e le medie mobili. Ad esempio, ROW_NUMBER, RANK, LAG e LEAD

Comando o funzione	Query di esempio	Description
Funzioni di conversione	`SELECT CAST('123' AS INT) as converted_number, CAST(123 AS STRING) as converted_string FROM `LogGroupA``	Built-in funzioni per convertire i dati da un tipo all'altro all'interno delle query SQL, che consentono trasformazioni dei tipi di dati e conversioni di formato. Ad esempio, CAST, TO_DATE, TO_TIMESTAMP e BINARY.
Funzioni di predicato	`SELECT scores, size(scores) as length, array_contains(scores, 90) as has_90 FROM `LogGroupA`;`	Built-in funzioni che valutano le condizioni e restituiscono valori booleani () true/false in base a criteri o modelli specifici . Ad esempio, IN, LIKE, BETWEEN, IS NULL ed EXISTS.
Seleziona più gruppi di log	`SELECT lg1.field1, lg1.field2 from `logGroups(logGroupIdentifier: ['LogGroup1', 'LogGroup2'])` as lg1 where lg1.field3= "Success"`	Consente di specificare più gruppi di log in un'istruzione SELECT

Comando o funzione	Query di esempio	Description
Seleziona più fonti di dati	`SELECT ds1.field1, ds1.field2 from `dataSource(['DataSource1', 'DataSource2'])` as ds1 where ds1.field3= "Success"`	Consente di specificare più fonti di dati in un'istruzione SELECT

SQL supportato per interrogazioni con gruppi di log multipli

Per supportare il caso d'uso per l'interrogazione di più gruppi di log in SQL, è possibile utilizzare il comando. `logGroups` Utilizzando questa sintassi, è possibile interrogare più gruppi di log specificandoli nel comando FROM.

Sintassi:

```
`logGroups(
    logGroupIdentifier: ['LogGroup1','LogGroup2', ...'LogGroupn']
)`
```

In questa sintassi, è possibile specificare fino a 50 gruppi di log nel parametro. `logGroupIdentifier` Per fare riferimento ai gruppi di log in un account di monitoraggio, utilizzate gli ARN anziché `LogGroup` i nomi.

Query di esempio:

```
SELECT LG1.Column1, LG1.Column2 from `logGroups(
    logGroupIdentifier: ['LogGroup1', 'LogGroup2']
)` as LG1 WHERE LG1.Column1 = 'ABC'
```

La seguente sintassi che coinvolge più gruppi di log dopo l'FR0Mistruzione NON è supportata durante l' CloudWatch interrogazione dei log.

```
SELECT Column1, Column2 FROM 'LogGroup1', 'LogGroup2', ...'LogGroupn'
WHERE Column1 = 'ABC'
```

SQL supportato per le interrogazioni sulle fonti di dati

Per supportare il caso d'uso per l'interrogazione di sorgenti di dati in SQL, è possibile utilizzare il comando DataSource. Utilizzando questa sintassi, è possibile interrogare le fonti di dati specificandole nel comando. `FROM` È possibile specificare fino a 10 fonti di dati.

Sintassi

```
`dataSource(
    ['DataSource1', 'DataSource2', ...'DataSourcen']
)`
```

Query di esempio

```
SELECT DS1.Column1, DS1.Column2 from `dataSource(
    ['DataSource1', 'DataSource2']
)` as DS1 WHERE DS1.Column1 = 'ABC'
```

Ambito della query

Nell'API AWS CLI and, è possibile specificare i log da interrogare utilizzando il gruppo di log, l'origine e il tipo di dati e gli indici dei campi.

Gruppo di log

La selezione della fonte del gruppo di log può essere utilizzata quando i clienti sanno esattamente quali gruppi di log devono essere cercati

```
SELECT * FROM `logGroups(logGroupIdentifier: ['/aws/lambda/my-function'])`;
```

Fonte e tipo di dati

I clienti possono interrogare i propri registri utilizzando il nome e il tipo di origine dati.

La selezione delle fonti basata sull'origine dei dati e sul tipo può essere utilizzata quando i clienti sanno quali fonti di dati esatte devono essere interrogate. Questa query viene eseguita su uno o più gruppi di log che contengono l'origine e il tipo di dati specificati.

Per supportare il caso d'uso per l'interrogazione di fonti di dati in SQL, è possibile utilizzare il comando DataSource. Utilizzando questa sintassi, è possibile interrogare le fonti di dati specificandole nel comando FROM. È possibile specificare fino a 10 fonti di dati.

Sintassi:

```
`dataSource(
    ['DataSource1.Type1', 'DataSource2.Type2', ...'DataSourcen.Typen']
)`
```

Query di esempio:

```
SELECT DS1.Column1, DS1.Column2 from `dataSource(
    ['DataSource1.Type1', 'DataSource2.Type2']
)` as DS1 WHERE DS1.Column1 = 'ABC'
```

Per ulteriori informazioni sulle interrogazioni in base alle fonti di dati, vedereUsa le sfaccettature per raggruppare ed esplorare i log.

Esempio per la combinazione

I clienti possono specificare tutti gli operatori di selezione della fonte all'interno dei backtick in qualsiasi ordine e i risultati si baseranno sull'intersezione di tutte le condizioni applicate.

Ad esempio,/aws/lambda/my-function-1 potrebbe contenere più fonti e tipi di dati tra cui un'ampia varietà di indici. Quando viene eseguita la seguente query, i risultati restituiti avranno solo eventi di origine e tipo DataSource1.Type1 e corrispondenti ai criteri di 'status' = 200.

```
SELECT * FROM `
   logGroups(logGroupIdentifier: ['/aws/lambda/my-function'])
   filterIndex('status' = 200)
   dataSource(['DataSource1.Type1'])
`;
```

Indici di campo

La selezione dell' Index-based origine dei campi identifica automaticamente i gruppi di log pertinenti quando i filtri sono indirizzati ai campi indicizzati, riducendo il volume di scansione e il tempo di esecuzione delle query.

`filterIndex`Da utilizzare per restituire solo dati indicizzati, forzando una query a scansionare solo i gruppi di log indicizzati su un campo specificato nella query. Per questi gruppi di log indicizzati in questo campo, ottimizza ulteriormente la query ignorando i gruppi di log che non contengono eventi

di registro contenenti il campo specificato nella query per il campo indicizzato. Riduce ulteriormente il volume scansionato tentando di analizzare solo gli eventi di registro di questi gruppi di log che corrispondono al valore specificato nella query per questo indice di campo. Per ulteriori informazioni sugli indici di campo e su come crearli, consulta Creare indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione.

In SQL, filterIndex viene utilizzato per specificare quali coppie chiave-valore devono essere trattate come indici. La sintassi è la seguente

```
SELECT * FROM `filterIndex('region' = 'us-east-1')`;
```

dove:

1. filterIndex (...) specifica, tratta i valori chiave al loro interno come indici di campo. Ogni coppia chiave-valore è separata da una virgola (esempio seguente)
2. 'region' = 'us-east-1' specifica la condizione effettiva da applicare
   a. Nota: invece di = i clienti possono utilizzare IN per specificare più valori (esempio seguente)

L'utilizzo di FilterIndex multiplo combinerebbe le condizioni utilizzando «AND». Nell'esempio, verrebbero interrogati i log che corrispondono a status=200 e region in us-east-1 o us-west-2.

```
SELECT * FROM `filterIndex('status' = 200, 'region' IN ['us-east-1', 'us-west-2'])`;
```

Restrizioni

Le seguenti restrizioni si applicano quando si utilizza SQL per eseguire query in Logs Insights. OpenSearch CloudWatch

- È possibile includere un solo JOIN in un'istruzione SELECT.
- Non è possibile utilizzare JOIN o sottoquery con query sull'origine dati.
- È supportato solo un livello di sottoquery annidate.
- Le query di istruzioni multiple separate da punto e virgola (;) non sono supportate.
- Le query contenenti nomi di campo identici ma che differiscono solo per maiuscole e minuscole (ad esempio field1 e FIELD1) non sono supportate.

  Ad esempio, la seguente query non è supportata:

```
Select AWSAccountId, AwsAccountId from LogGroup
```

Tuttavia, la seguente query è supportata perché il nome del campo (`@logStream`) è identico in entrambi i gruppi di log:

```
Select a.`@logStream`, b.`@logStream` from Table A INNER Join Table B on a.id = b.id
```

- Le funzioni e le espressioni devono operare sui nomi dei campi e far parte di un'istruzione SELECT con un gruppo di log specificato nella clausola FROM.

  Ad esempio, questa query non è supportata:

  ```
  SELECT cos(10) FROM LogGroup
  ```

  Questa interrogazione è supportata:

  ```
  SELECT cos(field1) FROM LogGroup
  ```

- Quando usi i comandi SQL o PPL, racchiudi alcuni campi tra i backtick per interrogarli correttamente. I backtick sono necessari per i campi con caratteri speciali (non alfabetici e non numerici). Ad esempio, racchiude e inserisce i segni di spunta rovesciati`@message`. `Operation.Export` `Test::Field` Non è necessario racchiudere campi con nomi puramente alfabetici nei backtick.

  Query di esempio con campi semplici:

  ```
  SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
  LIMIT 1000;
  ```

  Query simile con backtick aggiunti:

  ```
  SELECT `@SessionToken`, `@Operation`, `@StartTime`  FROM `LogGroup-A` LIMIT 1000;
  ```

Usa il linguaggio naturale per generare e aggiornare CloudWatch le query di Logs Insights

CloudWatch Logs supporta una funzionalità di interrogazione in linguaggio naturale per aiutarti a generare e aggiornare le query per CloudWatch Logs Insights, OpenSearch Service PPL, Service SQL e. OpenSearch CloudWatch Metrics Insights

Con questa funzionalità, puoi porre domande o descrivere i dati di CloudWatch Logs che stai cercando in un inglese semplice. La funzionalità di linguaggio naturale genera una query basata su una richiesta inserita da te e fornisce una spiegazione riga per riga del funzionamento della query. Puoi anche aggiornare la tua query per analizzare ulteriormente i dati.

A seconda dell'ambiente, puoi inserire istruzioni come «Quali sono i primi 100 indirizzi IP di origine per byte trasferiti?» e «Trova le 10 richieste di funzioni Lambda più lente».

> **Note**
>
> La funzionalità di interrogazione in linguaggio naturale è un servizio regionale. Per alcune Regioni, la funzionalità effettua chiamate tra Regioni verso le Regioni degli Stati Uniti d'America per elaborare i prompt di query. Per ulteriori informazioni, consulta Amazon CloudWatch estende il supporto regionale per il riepilogo dei risultati delle query in linguaggio naturale e la generazione di query.

Per generare una query di CloudWatch Logs Insights con questa funzionalità, apri l'editor di query di CloudWatch Logs Insights, seleziona il gruppo di log su cui desideri interrogare e scegli Genera query.

> **Important**
>
> Per utilizzare la funzionalità di interrogazione in linguaggio naturale, devi aver effettuato l'accesso con le politiche CloudWatchLogsFullAccessCloudWatchLogsReadOnlyAccess, AdministratorAccess, o ReadOnlyAccessIAM o disporre dell'`cloudwatch:GenerateQuery`autorizzazione.

Query di esempio

Gli esempi in questa sezione descrivono come generare e aggiornare le query utilizzando la funzionalità di linguaggio naturale.

> **Note**
>
> Per ulteriori informazioni sull'editor di query e sulla sintassi di CloudWatch Logs Insights, vedere Sintassi delle query di CloudWatch Logs Insights.

Esempi: generazione di una query in linguaggio naturale

Per generare un'interrogazione utilizzando il linguaggio naturale, inserisci un prompt e scegli Genera nuova query. Questo esempio mostra le interrogazioni che eseguono una ricerca di base.

Prompt

Di seguito è riportato un esempio di prompt che indirizza la capacità di cercare le 10 chiamate di funzione Lambda più lente.

```
Find the 10 slowest requests
```

Query

Di seguito è riportata la query che utilizza il linguaggio di query CloudWatch Logs Insights generata dalla funzionalità del linguaggio naturale in base al prompt. Nota come il prompt appare in un commento prima della query. Dopo l'esecuzione della query, puoi leggere una spiegazione del funzionamento della query.

```
# Find the 10 slowest requests
fields @timestamp, @message, @duration
| sort @duration desc
| limit 10
# This query retrieves the timestamp, message and duration fields from the logs and
 sorts them in descending order by duration to find the 10 slowest requests.
```

> **Note**
>
> Per disattivare la visualizzazione del prompt e la spiegazione del funzionamento della query, utilizza l'icona a forma di ingranaggio nell'editor.

Prompt

Per generare una query OpenSearch SQL, seleziona la scheda OpenSearch SQL, quindi apri la casella di richiesta del generatore di query per inserire il prompt del linguaggio naturale. Di seguito è riportato un esempio di prompt che utilizza la funzionalità del linguaggio naturale per generare una OpenSearch query SQL.

```
Give me the number of errors and exceptions per hour
```

Query

Di seguito è riportata la query SQL generata da quel prompt che è possibile utilizzare per trovare il numero di errori ed eccezioni aggregati all'ora:

```
SELECT DATE_FORMAT(`@timestamp`, 'yyyy-MM-dd HH') AS hour,
       COUNT(*) AS error_count
FROM `/aws/lambda/CloudWatchOdysseyQueryGen`
WHERE `@message` LIKE '%error%'
  OR `@message` LIKE '%exception%'
GROUP BY DATE_FORMAT(`@timestamp`, 'yyyy-MM-dd HH')
ORDER BY hour
```

Prompt

Per generare una query OpenSearch PPL, seleziona la scheda OpenSearch PPL, quindi apri la casella di richiesta del generatore di query per inserire il prompt del linguaggio naturale. Di seguito è riportato un esempio di prompt che utilizza la funzionalità del linguaggio naturale per generare una query PPL. OpenSearch

```
Give me all unique exception messages
```

Query

Di seguito è riportata la query PPL generata da quel prompt che è possibile utilizzare per trovare i messaggi di eccezione univoci nei registri:

```
dedup @message
| fields @message
```

Esempio: aggiornamento di una query in linguaggio naturale

È possibile aggiornare una query modificando il prompt iniziale e scegliendo Aggiorna query.

Richiesta aggiornata

L'esempio seguente mostra una versione aggiornata del prompt precedente. Invece di un prompt che cerca le 10 chiamate di funzione Lambda più lente, questo prompt ora indirizza la capacità di cercare le 20 chiamate di funzione Lambda più lente e include un'altra colonna per eventi di registro aggiuntivi.

```
Show top 20 slowest requests instead and display requestId as a column
```

Query aggiornata

Di seguito è riportato un esempio di query aggiornata che utilizza il linguaggio di query Logs Insights. CloudWatch Nota come il prompt aggiornato appare in un commento prima della query aggiornata. Dopo l'esecuzione della query, puoi leggere una spiegazione di come la query originale è stata aggiornata.

```
# Show top 20 slowest requests instead and display requestId as a column
fields @timestamp, @message, @requestId, @duration
| sort @duration desc
| limit 20
# This query modifies the original query by replacing the @message field with the
 @requestId field and changing the limit from 10 to 20 to return the top 20 log events
 by duration instead of the top 10.
```

Rifiuto esplicito all'utilizzo dei dati volto al miglioramento del servizio

I dati prompt in linguaggio naturale forniti per addestrare il modello di intelligenza artificiale e generare query pertinenti vengono utilizzati esclusivamente per fornire e gestire il servizio. Questi dati possono

essere utilizzati per migliorare la qualità di CloudWatch Logs Insights. La tua fiducia, la tua privacy e la sicurezza dei tuoi contenuti sono le nostre maggiori priorità. Per ulteriori informazioni, consulta i Termini del servizio AWS e la Policy sull'IA responsabile di AWS.

Puoi scegliere di non utilizzare i tuoi contenuti per sviluppare o migliorare la qualità delle query in linguaggio naturale creando una policy di rifiuto dei servizi di intelligenza artificiale. Per disattivare la raccolta dei dati per tutte le funzionalità di CloudWatch Logs AI, inclusa la funzionalità di generazione di query, è necessario creare una politica di opt-out per Logs. CloudWatch Per ulteriori informazioni, consulta le Policy di rifiuto dei servizi di IA nella Guida per l'utente di AWS Organizations .

Registri supportati e campi rilevati

CloudWatch Logs Insights supporta diversi tipi di log. Per ogni log inviato a un gruppo di log di classe Standard in Amazon CloudWatch Logs, CloudWatch Logs Insights genera automaticamente cinque campi di sistema:

- `@message` contiene il log eventi non analizzato, non elaborato. Questo è l'equivalente del `message` campo in. InputLogevent
- `@timestamp` contiene il timestamp dell'evento nel campo `timestamp` del log eventi. È l'equivalente del `timestamp` campo in InputLogevent.
- `@ingestionTime`contiene l'ora in cui CloudWatch Logs ha ricevuto l'evento di registro.
- `@logStream` contiene il nome del flusso di log a cui il log eventi è stato aggiunto. I flussi di log raggruppano i registri in base allo stesso processo che li ha generati.
- `@log` è un identificatore di gruppo di log sotto forma di *`account-id`*`:`*`log-group-name`*. Nelle query di più gruppi di log, può essere utile per identificare a quale gruppo di log appartiene un particolare evento.
- `@entity`contiene JSON appiattito relativo alle entità per la funzionalità di telemetria relativa a Explore.

  Ad esempio, questo JSON può rappresentare un'entità.

```
{
  "Entity": {
    "KeyAttributes": {
      "Type": "Service",
      "Name": "PetClinic"
    },
```

```
    "Attributes": {
      "PlatformType": "AWS::EC2",
      "EC2.InstanceId": "i-1234567890123"
    }
  }
}
```

Per questa entità, i campi di sistema estratti sarebbero i seguenti:

```
@entity.KeyAttributes.Type = Service
@entity.KeyAttributes.Name = PetClinic
@entity.Attributes.PlatformType = AWS::EC2
@entity.Attributes.EC2.InstanceId = i-1234567890123
```

> **Note**
>
> L'individuazione dei campi è supportata solo per i gruppi di log nella classe di log Standard. Per ulteriori informazioni sulle classi di log, vedereClassi di registro.

CloudWatch Logs Insights inserisce il simbolo @ all'inizio dei campi generati.

Per molti tipi di log, CloudWatch Logs rileva inoltre automaticamente i campi di log contenuti nei log. Questi campi di rilevamento automatici sono mostrati nella tabella seguente.

Per altri tipi di log con campi che CloudWatch Logs Insights non rileva automaticamente, puoi utilizzare il `parse` comando per estrarre e creare campi estratti da utilizzare in quella query. Per ulteriori informazioni, consulta CloudWatch Sintassi delle interrogazioni in linguaggio Logs Insights.

Se il nome di un campo di registro rilevato inizia con il `@` carattere, CloudWatch Logs Insights lo visualizza con un altro `@` aggiunto all'inizio. Ad esempio, se un nome di un campo di log è `@example.com`, questo nome di campo viene visualizzato come `@@example.com`.

> **Note**
>
> Ad eccezione di`@message`, o `@timestamp@log`, è possibile creare indici di campo per i campi scoperti. Per ulteriori informazioni sugli indici di campo, vedere. Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione

Tipo di log	Campi di log rilevati
Log di flusso Amazon VPC	`@timestamp` , `@logStream` , `@message`, `accountId` , `endTime`, `interfaceId` , `logStatus` , `startTime` , `version`, `action`, `bytes`, `dstAddr`, `dstPort`, `packets`, `protocol`, `srcAddr`, `srcPort`
Log di Route 53	`@timestamp` , `@logStream` , `@message`, `edgeLocation` , `ednsClientSubnet` , `hostZoneId` , `protocol`, `queryName` , `queryTimestamp` , `queryType` , `resolverIp` , `responseCode` , `version`
Log di Lambda	`@timestamp` , `@logStream` , `@message`, `@requestId` , `@duration`, `@billedDuration` , `@type`, `@maxMemoryUsed` , `@memorySize` Se una riga di registro Lambda contiene un ID di X-Ray traccia, include anche i seguenti campi: `@xrayTraceId` e. `@xraySegmentId` CloudWatch Logs Insights rileva automaticamente i campi di registro nei log Lambda, ma solo per il primo frammento JSON incorporato in ogni evento di registro. Se un log eventi Lambda contiene più frammenti JSON, puoi analizzare ed estrarre i campi dei log utilizzando il comando **`parse`**. Per ulteriori informazioni, consulta Campi nei registri JSON.
CloudTrail registri Log in formato JSON	Per ulteriori informazioni, consulta Campi nei registri JSON.
Altri tipi di log	`@timestamp` , `@ingestionTime` , `@logStream` , `@message`, `@log`.

Campi nei registri JSON

Con CloudWatch Logs Insights, si utilizza la notazione a punti per rappresentare i campi JSON. Questa sezione contiene un esempio di evento e di frammento di codice JSON che mostra come accedere ai campi JSON utilizzando la notazione con il punto.

Esempio: evento JSON

```
{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123",
                    "currentState": {
                        "code": 0,
                        "name": "pending"
                    },
                    "previousState": {
                        "code": 80,
                        "name": "stopped"
                    }
                }
            ]
        }
    }
}
```

L'evento JSON di esempio contiene un oggetto denominato `userIdentity`. `userIdentity` contiene un campo denominato `type`. Per rappresentare il valore di `type` usando la notazione con il punto, utilizza `userIdentity.type`.

L'evento JSON di esempio contiene matrici che si livellano in elenchi di nomi e valori di campi annidati. Per rappresentare il valore di `instanceId` per il primo elemento di `requestParameters.instancesSet`, utilizza `requestParameters.instancesSet.items.0.instanceId`. Il numero `0` posizionato prima del campo `instanceID` si riferisce alla posizione dei valori per il campo `items`. L'esempio seguente contiene un frammento di codice che mostra come è possibile accedere ai campi JSON annidati in un log eventi JSON.

Esempio: query

```
fields @timestamp, @message
| filter requestParameters.instancesSet.items.0.instanceId="i-abcde123"
| sort @timestamp desc
```

Il frammento di codice mostra una query che utilizza la notazione con il punto con il comando `filter` per accedere al valore del campo JSON annidato `instanceId`. La query filtra i messaggi in cui il valore di `instanceId` è uguale a `"i-abcde123"` e restituisce tutti i log eventi che contengono il valore specificato.

Notazione a punti e campi a stringa JSON-encoded

La notazione a punti attraversa solo i campi archiviati come oggetti JSON strutturalmente annidati al momento dell'inserimento. Se il valore di un campo è una JSON-encoded stringa (una stringa il cui contenuto è in formato JSON valido), la notazione a punti lo considera come un valore foglia opaco e non accede ai sottocampi al suo interno.

Ciò si verifica in genere con:

- Registri trasformati dalle pipeline OCSF (ad esempio, vengono digitati come `json_t` nello `api.request.data` schema OCSF)
- Qualsiasi origine di log in cui i payload a struttura variabile vengono serializzati come stringhe prima dell'ingestione

Ad esempio, se `api.request.data` contiene `{"startTime":1234,"endTime":5678}` come valore di stringa, non `api.request.data.startTime` restituisce alcun risultato perché

CloudWatch Logs Insights tratta l'intera stringa come un valore a foglia singola anziché come un oggetto nidificato.

Per accedere ai sottocampi all'interno di una JSON-encoded stringa, usa `jsonParse` per convertire la stringa in una mappa attraversabile.

Esempio: eseguire una query utilizzando JSONParse per un campo stringa JSON-encoded

```
fields jsonParse(api.request.data).startTime as startTime
```

Questa query analizza la JSON-encoded stringa in `api.request.data` una mappa e quindi accede al sottocampo con la `startTime` notazione a punti. `jsonParse` Per ulteriori informazioni sujsonParse, vedere Tipi di struttura.

> **Note**
>
> CloudWatch Logs Insights può estrarre un massimo di 200 campi di eventi di registro da un registro JSON. Per i campi aggiuntivi che non vengono estratti, è possibile utilizzare il comando `parse` per estrarre questi campi dal log eventi non analizzato e non elaborato nel campo del messaggio. Per ulteriori informazioni sul `parse` comando, consulta la sintassi delle query nella Amazon CloudWatch User Guide.

Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione

Puoi creare indici di campo dei campi nei tuoi eventi di registro per ricerche efficienti basate sull'uguaglianza. Quando poi si utilizza un indice di campo in una query di CloudWatch Logs Insights, la query tenta di ignorare l'elaborazione degli eventi di registro che notoriamente non includono il campo indicizzato. Ciò riduce il volume di scansione delle query che utilizzano gli indici di campo, rendendo possibile la restituzione dei risultati più rapidamente. Questo può aiutarti a cercare rapidamente petabyte di log totali in migliaia di gruppi di log e ad affinare più rapidamente i log pertinenti. I campi validi da indicizzare sono i campi per i quali spesso è necessario eseguire delle query. I campi con un'elevata cardinalità di valori sono anche ottimi candidati per gli indici di campo, perché una query che utilizza questi indici di campo verrà completata più rapidamente perché limita gli eventi di registro che vengono abbinati al valore di destinazione.

Ad esempio, supponiamo di aver creato un indice di campo per. `requestId` Quindi, qualsiasi query di CloudWatch Logs Insights su quel gruppo di log che includa `requestId = value` o `requestId IN [value, value, ...]` tenterà di elaborare solo gli eventi di registro che sono noti per contenere quel campo indicizzato e il valore interrogato e per i quali CloudWatch Logs ha rilevato un valore per quel campo in passato.

È inoltre possibile utilizzare gli indici dei campi per creare query efficienti su un numero maggiore di gruppi di log. Quando si utilizza il `filterIndex` comando nella query anziché il `filter` comando, la query verrà eseguita su gruppi di log selezionati sugli eventi di registro con indici di campo. Queste query possono scansionare fino a 10.000 gruppi di log, scelti specificando fino a cinque prefissi dei nomi dei gruppi di log. Se si tratta di un account di monitoraggio in modalità osservabile CloudWatch tra più account, è possibile scegliere tutti gli account di origine o specificare singoli account di origine per selezionare i gruppi di log.

I campi indicizzati fanno distinzione tra maiuscole e minuscole. Ad esempio, un indice di campo non `RequestId` corrisponderà a un evento di registro contenente. `requestId`

Gli indici dei campi sono supportati solo per i formati di log strutturati di JSON e i log di servizio.

CloudWatch Logs fornisce indici di campo predefiniti per tutti i gruppi di log nella classe di log Standard. Gli indici di campo predefiniti sono automaticamente disponibili per i seguenti campi:

- `@logStream`
- `@aws.region`
- `@aws.account`
- `@source.log`
- `@data_source_name`
- `@data_source_type`
- `@data_format`
- `traceId`
- `severityText`
- `attributes.session.id`

CloudWatch Logs fornisce indici di campo predefiniti anche per determinate combinazioni di nomi e tipi di origini dati. Gli indici di campo predefiniti sono automaticamente disponibili per le seguenti combinazioni di nomi e tipi di origini dati:

Nome e tipo di origine dati	Indici di campo predefiniti
amazon_vpc.flow	action logStatus region flowDirection type
amazon_route53.resolver_query	query_type transport rcode
aws_waf.access	action httpRequest.country
aws_cloudtrail.data aws_cloudtrail.management	eventSource eventName awsRegion userAgent errorCode eventType managementEvent readOnly eventCategory requestId

Gli indici di campo predefiniti si aggiungono a tutti gli indici di campo personalizzati definiti all'interno della politica. Gli indici di campo predefiniti non vengono conteggiati ai fini della quota dell'indice di campo.

CloudWatch Registra gli indici solo gli eventi di registro acquisiti dopo la creazione di una politica di indicizzazione. Non indicizza gli eventi di registro acquisiti prima della creazione della policy. Dopo aver creato un indice di campo, ogni evento di registro corrispondente rimane indicizzato per 30 giorni dal momento di inserimento dell'evento di registro.

> **Note**
>
> Se si crea una politica di indicizzazione dei campi in un account di monitoraggio, tale politica non viene utilizzata per i gruppi di log negli account di origine collegati. Una politica di indicizzazione dei campi si applica solo all'account in cui è stata creata.

Gli altri argomenti di questa sezione spiegano i campi indicizzati automaticamente e come creare indici di campo. Per informazioni sul riferimento agli indici di campo nelle query, consulta e. FilterIndex filter

Argomenti

- Campi indicizzati automaticamente
- Sintassi e quote dell'indice di campo
- Crea una politica di indicizzazione dei campi a livello di account
- Crea una politica di indicizzazione dei campi a livello di gruppo di log
- Registra le opzioni di selezione dei gruppi durante la creazione di un'interrogazione
- Effetti dell'eliminazione di una politica di indicizzazione dei campi

Campi indicizzati automaticamente

CloudWatch Logs indicizza automaticamente i campi in base al loro utilizzo recente nei filtri di uguaglianza di CloudWatch Logs Insights che utilizzano gli operatori or. `=` `IN` Questi indici si aggiungono agli indici di campo predefiniti e agli indici di campo configurati nelle politiche. Non è necessario configurare una politica per i campi indicizzati automaticamente.

CloudWatch Logs aggiorna il set di campi indicizzati automaticamente in base alle attività di query recenti. Quando CloudWatch Logs rimuove un campo dal set, interrompe l'indicizzazione dei nuovi

eventi inseriti per quel campo. CloudWatch Logs gestisce automaticamente i campi indicizzati e li conserva per 30 giorni. Per mantenere un campo indicizzato in modo permanente, aggiungilo a una politica di indicizzazione dei campi a livello di account o gruppo di log.

⚠ Utilizza il filtro anziché filterIndex per i campi indicizzati automaticamente

Si consiglia di utilizzare `filter` al posto dei campi `filterIndex` indicizzati automaticamente. CloudWatch I log possono aggiornare o rimuovere questi campi in base a modelli di query. CloudWatch I registri li conservano solo per 30 giorni. Il `filterIndex` comando restituisce solo dati indicizzati. Poiché CloudWatch Logs aggiorna il set in base all'attività di query, una `filterIndex` query non cerca gli eventi inseriti prima che CloudWatch Logs selezionasse il campo o dopo che Logs avesse interrotto l'indicizzazione. CloudWatch Se lo utilizzi`filterIndex`, aggiungi il campo a una politica di indice dei campi per mantenerlo indicizzato indipendentemente dall'attività di query. Per ulteriori informazioni, consulta FilterIndex rispetto al filtro.

La scheda Indici di campo per un gruppo di log elenca i campi indicizzati automaticamente.

È inoltre possibile utilizzare l'`DescribeFieldIndexes`operazione per elencare i campi indicizzati automaticamente. AUTOIncludi nel parametro di `indexCategories` richiesta. La risposta è impostata `indexCategory` su AUTO per i campi indicizzati automaticamente. Le richieste omesse non `indexCategories` restituiscono campi indicizzati automaticamente.

Per aggiungere un campo indicizzato automaticamente a una politica di indicizzazione a livello di gruppo di log

1. Apri la console all'indirizzo. CloudWatch https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione a sinistra, scegli Logs (Log), Log groups (Gruppi di log).
3. Scegli il nome del gruppo di log.
4. Scegli la scheda Indici dei campi, quindi scegli Gestisci indici di campo.
5. Nella pagina Gestisci gli indici dei campi a livello di gruppo di log, esamina i campi della politica. Aggiungi o rimuovi i campi secondo necessità, quindi scegli Salva.

Sintassi e quote dell'indice di campo

Gli indici di campo vengono creati creando criteri per gli indici di campo. È possibile creare politiche di indicizzazione a livello di account che si applicano all'intero account e anche creare politiche che si applicano a un solo gruppo di log. Per quanto riguarda le politiche di indicizzazione a livello di account, puoi averne una che si applichi a tutti i gruppi di log dell'account. È inoltre possibile creare politiche di indicizzazione a livello di account che si applicano a un sottoinsieme di gruppi di log dell'account, selezionati in base ai prefissi dei nomi dei rispettivi gruppi di log. Se hai più politiche a livello di account nello stesso account, i prefissi dei nomi dei gruppi di log per queste politiche non possono sovrapporsi. Allo stesso modo, puoi creare politiche di indicizzazione a livello di account che si applicano a una combinazione specifica di nome e tipo di origine dati. È possibile creare una sola politica dell'account per combinazione di nome e tipo di origine dati.

Le politiche dell'indice dei campi a livello di gruppo di log hanno la precedenza sulle politiche dell'indice dei campi a livello di account: si applicano all'intero gruppo di log (ad esempio, le politiche a livello di account senza criteri di selezione o con criteri di selezione basati sul prefisso del nome del gruppo di log). Account-level le politiche che corrispondono a livello di evento di registro (ad esempio, per una determinata combinazione di nome e tipo di origine dati) verranno applicate in aggiunta alle politiche che corrispondono al gruppo di log nel suo insieme. Se si creano criteri di indicizzazione a livello di gruppo di log, tale gruppo di log non utilizza politiche a livello di account che corrispondono a livello di gruppo di log.

Le corrispondenze degli eventi di registro con i nomi degli indici dei campi fanno distinzione tra maiuscole e minuscole. Ad esempio, un indice di campo di non `RequestId` corrisponderà a un evento di registro contenente. `requestId`

È possibile avere fino a 40 criteri di indicizzazione a livello di account, di questi criteri 20 possono utilizzare criteri di selezione del prefisso del nome del gruppo di log e 20 possono utilizzare criteri di selezione basati sull'origine dei dati. Se hai più politiche di indicizzazione a livello di account filtrate in base ai prefissi dei nomi dei gruppi di log, nessuna di esse può utilizzare prefissi di nomi di gruppi di log uguali o sovrapposti. Ad esempio, se hai una politica filtrata in base ai gruppi di log che iniziano con`my-log`, non puoi avere un'altra politica di indice dei campi filtrata su o. `my-logpprod` `my-logging` Analogamente, se hai più policy di indicizzazione a livello di account filtrate in base a combinazioni di nomi e tipi di origini dati, non esistono due di esse che possano utilizzare lo stesso nome e tipo di origine dati. Ad esempio, se hai una politica filtrata in base al nome dell'origine dati `amazon_vpc` e al tipo di origine dati, non `flow` puoi creare un'altra politica con questa combinazione.

Se si dispone di una politica di indicizzazione a livello di account che non ha prefissi di nome e si applica a tutti i gruppi di log, non è possibile creare altre politiche di indicizzazione a livello di account con filtri per i prefissi dei nomi dei gruppi di log; è possibile creare politiche di indice a livello di account che utilizzano filtri per il nome e il tipo di origine dati.

Ogni politica di indicizzazione prevede le seguenti quote e restrizioni:

- Nella policy possono essere inclusi fino a 20 campi.
- Ogni nome di campo può includere fino a 100 caratteri.
- Per creare un indice di un campo personalizzato nei gruppi di log che inizi con`@`, devi specificare il campo con un extra `@` all'inizio del nome del campo. Ad esempio, se gli eventi del registro includono un campo denominato`@userId`, è necessario specificare `@@userId` di creare un indice per questo campo.

Per le politiche di indicizzazione a livello di account con criteri di selezione basati sul nome dell'origine dati e sul tipo, si applica un'ulteriore restrizione: tutti i campi devono essere tipi di dati primitivi, le primitive annidate sono supportate solo per le strutture.

Campi generati e campi riservati

CloudWatch Logs Insights genera automaticamente campi di sistema in ogni evento di registro. Questi campi generati hanno il prefisso `@` Per ulteriori informazioni sui campi generati, vedere. Registri supportati e campi rilevati

Di questi campi generati, i seguenti sono supportati per l'uso come indici di campo:

- `@logStream`
- `@ingestionTime`
- `@requestId`
- `@type`
- `@initDuration`
- `@duration`
- `@billedDuration`
- `@memorySize`
- `@maxMemoryUsed`
- `@xrayTraceId`

- `@xraySegmentId`

Per indicizzare questi campi generati, non è necessario aggiungerne un altro `@` quando li si specifica, come è necessario fare per i campi personalizzati che iniziano con. `@` Ad esempio, per creare un indice di campo per`@logStream`, è sufficiente specificare `@logStream` come indice di campo.

CloudWatch Logs fornisce indici di campo predefiniti per tutti i gruppi di log nella classe di log Standard. Gli indici di campo predefiniti sono automaticamente disponibili per i seguenti campi:

- `@logStream`
- `@aws.region`
- `@aws.account`
- `@source.log`
- `@data_source_name`
- `@data_source_type`
- `@data_format`
- `traceId`
- `severityText`
- `attributes.session.id`

CloudWatch Logs fornisce indici di campo predefiniti anche per determinate combinazioni di nomi e tipi di origini dati. Gli indici di campo predefiniti sono automaticamente disponibili per le seguenti combinazioni di nomi e tipi di origini dati:

Nome e tipo di origine dati	Indici di campo predefiniti

`amazon_vpc.flow`	`action`
	`logStatus`
	`region`
	`flowDirection`
	`type`

Nome e tipo di origine dati	Indici di campo predefiniti

`amazon_route53.resolver_query`	`query_type`
	`transport`
	`rcode`

`aws_waf.access`	`action` `httpRequest.country`

`aws_cloudtrail.data`
`aws_cloudtrail.management`
	`eventSource`
	`eventName`
	`awsRegion`
	`userAgent`
	`errorCode`
	`eventType`
	`managementEvent`
	`readOnly`
	`eventCategory`
	`requestId`

Gli indici di campo predefiniti si aggiungono a tutti gli indici di campo personalizzati definiti all'interno della politica. Gli indici di campo predefiniti non vengono conteggiati ai fini della quota dell'indice di campo.

Campi secondari e campi matrice nei log JSON

È possibile indicizzare i campi che sono campi secondari o campi array annidati nei log JSON.

Ad esempio, puoi creare un indice del campo `accessKeyId` figlio all'interno del `userIdentity` campo all'interno di questo registro:

```
{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "11112222",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [{
                "instanceId": "i-abcde123",
                "currentState": {
                    "code": 0,
                    "name": "pending"
                },
                "previousState": {
                    "code": 80,
                    "name": "stopped"
                }
            }]
        }
    }
}
```

Per creare questo campo, vi fate riferimento utilizzando la notazione a punti (`userIdentity.accessKeyId`) sia durante la creazione dell'indice del campo che quando lo specificate in una query. La query potrebbe avere il seguente aspetto:

```
fields @timestamp, @message
| filterIndex userIdentity.accessKeyId = "11112222"
```

Nell'evento di esempio precedente, il `instanceId` campo si trova in una matrice all'interno di `requestParameters.instancesSet.items` Per rappresentare

questo campo sia durante la creazione dell'indice del campo che durante `requestParameters.instancesSet.items.0.instanceId` l'interrogazione, fate riferimento ad esso poiché 0 si riferisce alla posizione di quel campo nell'array.

Quindi una query per questo campo potrebbe essere la seguente:

```
fields @timestamp, @message
| filterIndex requestParameters.instancesSet.items.0.instanceId="i-abcde123"
```

Crea una politica di indicizzazione dei campi a livello di account

Utilizza i passaggi descritti in questa sezione per creare una politica di indice dei campi che si applichi a tutti i gruppi di log dell'account o a più gruppi di log i cui nomi di gruppi di log iniziano con la stessa stringa.

Per creare una politica di indicizzazione dei campi a livello di account

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione a sinistra, scegli Impostazioni, quindi scegli la scheda Registri.
3. Nella sezione Politiche relative all'indice a livello di account, scegli Gestisci.
4. Scegli Crea politica sull'indice.
5. Per Nome della politica, inserisci un nome per la nuova politica.
6. Per Seleziona l'ambito della politica, esegui una delle seguenti operazioni:
   - Scegliete Tutti i gruppi di log standard per applicare la politica di indicizzazione a tutti i gruppi di log di Standard Class presenti nell'account.
   - Scegli Registra i gruppi per corrispondenza del prefisso per applicare la politica a un sottoinsieme di gruppi di log i cui nomi iniziano con la stessa stringa. Quindi, inserisci il prefisso per questi gruppi di log in Inserisci un nome di prefisso.

     Dopo aver inserito il prefisso, puoi scegliere Anteprima dei gruppi di log corrispondenti al prefisso per confermare che il prefisso corrisponde ai gruppi di log che ti aspettavi.

     Scegli Registra dati per origine dati per applicare la politica a una combinazione specifica di nome e tipo di origine dati. Puoi quindi selezionare l'origine dei dati e il tipo di dati dal menu a discesa.

Dopo aver selezionato il nome e il tipo di origine dati, puoi selezionare Ottieni campi per compilare la sezione Configura gli indici e i facet dei campi con informazioni pertinenti come i campi disponibili, i gruppi di log inclusi e gli indici di campo predefiniti e personalizzati.

7. Per la configurazione personalizzata dei campi indice, scegli Aggiungi percorso campo per inserire il primo campo da indicizzare.

   Quindi inserisci la stringa da utilizzare come valore del nome del campo o seleziona un campo dal menu a discesa. Deve corrispondere esattamente al maiuscolo/minuscolo visualizzato nel registro degli eventi. Ad esempio, se gli eventi di registro includonorequestId, devi inserirli requestId qui. RequestIdrequestID, e non request Id corrisponderebbe.

   Se si desidera indicizzare un campo di registro personalizzato che inizia con il @ carattere, è necessario includere un @ carattere aggiuntivo quando si immette la stringa dell'indice. Ad esempio, se disponi di un campo di registro personalizzato@emailname, inseriscilo **@@emailname** nella casella Aggiungi percorso del campo.

   Puoi anche creare indici per i @logStream campi @ingestionTime e generati automaticamente da CloudWatch Logs. In tal caso, non è necessario aggiungerne altri @ quando li si specifica.
8. (Facoltativo) Oltre a specificare il percorso del campo, puoi selezionare Imposta come sfaccettatura per creare il campo come sfaccettatura.
9. Ripeti il passaggio precedente per aggiungere fino a 20 indici di campo.
10. Al termine, scegliere Create (Crea).

Crea una politica di indicizzazione dei campi a livello di gruppo di log

Utilizza i passaggi descritti in questa sezione per creare una politica di indice dei campi che si applichi a un singolo gruppo di log.

Per creare una politica di indicizzazione dei campi a livello di gruppo di log

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione a sinistra, scegli Logs (Log), Log groups (Gruppi di log).
3. Scegli il nome del gruppo di log.
4. Scegli la scheda Indici di campo.
5. Scegli Gestisci gli indici dei campi per questo gruppo di log

6. Per Gestisci gli indici dei campi a livello di gruppo di log, scegli Aggiungi percorso di campo per inserire il primo campo da indicizzare.

   Quindi inserisci la stringa da utilizzare come valore del nome del campo. Deve corrispondere esattamente al maiuscolo/minuscolo visualizzato nel registro degli eventi. Ad esempio, se gli eventi di registro includono`requestId`, devi inserirli `requestId` qui. `RequestIdrequestID`, e non `request Id` corrisponderebbe.

   Se si desidera indicizzare un campo di registro personalizzato che inizia con il `@` carattere, è necessario includere un `@` carattere aggiuntivo quando si immette la stringa dell'indice. Ad esempio, se disponi di un campo di registro personalizzato`@emailname`, inseriscilo **`@@emailname`** nella casella Aggiungi percorso del campo.

   Puoi anche creare indici per i `@logStream` campi `@ingestionTime` e generati automaticamente da CloudWatch Logs. In tal caso, non è necessario aggiungerne altri `@` quando li si specifica.
7. (Facoltativo) Oltre a specificare il percorso del campo, puoi selezionare Imposta come sfaccettatura per creare il campo come sfaccettatura.
8. Ripeti il passaggio precedente per aggiungere fino a 20 indici di campo.
9. Al termine, scegliere Save (Salva).

Registra le opzioni di selezione dei gruppi durante la creazione di un'interrogazione

Questa sezione spiega i vari modi in cui è possibile selezionare i gruppi di log da includere in una query.

Per selezionare i gruppi di log per un'interrogazione nella console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs, quindi Logs Insights.
3. Seleziona la lingua di interrogazione che desideri utilizzare per questa interrogazione. Puoi scegliere tra: Logs Insights QL, OpenSearchPPL o OpenSearch SQL.
4. Esistono tre modi per selezionare i gruppi di log per la query:
   - Utilizzate la casella Nome del gruppo di log. Questo è il metodo di selezione predefinito. Con questo metodo è possibile inserire fino a 50 nomi di gruppi di log. Se si tratta di un account

di monitoraggio in CloudWatch modalità osservabile tra più account, è possibile selezionare gruppi di log negli account di origine e nell'account di monitoraggio. Una singola query può interrogare i log di diversi account contemporaneamente.

- Utilizza la sezione Criteri del gruppo di log. In questa sezione, è possibile scegliere i gruppi di log in base al prefisso dei nomi dei gruppi di log. È possibile includere fino a cinque prefissi in una query. Verranno selezionati i gruppi di log che hanno questi prefissi nei nomi. In alternativa, l'opzione Tutti i gruppi di log seleziona tutti i gruppi di log dall'account.
- Se si tratta di un account di monitoraggio in modalità CloudWatch osservabile su più account, puoi selezionare Tutti gli account nel menu a discesa degli account per selezionare i gruppi di log tra tutti gli account collegati. In alternativa, puoi selezionare singolarmente quali account includere in questa query.

Se le tue scelte corrispondono a più di 10.000 gruppi di log, visualizzerai un errore che ti chiederà di restringere la selezione.

5. La classe di log predefinita per una query è Standard. È possibile utilizzare la classe Log per modificarla in Accesso infrequente.

Usando la AWS CLI

Per effettuare questo tipo di selezione quando si avvia una query dalla riga di comando, è possibile utilizzare il `source` comando contenuto nella query. Per maggiori informazioni ed esempi, consulta SOURCE.

Effetti dell'eliminazione di una politica di indicizzazione dei campi

Se si elimina una politica di indicizzazione dei campi che è in vigore da un certo periodo, si verifica quanto segue:

- Fino a 30 giorni dopo l'eliminazione della policy, le query possono ancora trarre vantaggio dagli eventi di registro indicizzati.
- Se si elimina una politica di indicizzazione a livello di gruppo di log ed esiste già una politica a livello di account applicabile a quel gruppo di log, la politica a livello di account verrà infine applicata a quel gruppo di log.

Usa le sfaccettature per raggruppare ed esplorare i log

I facet sono utili per analizzare i log in quanto consentono di filtrare e raggruppare i dati in modo interattivo senza eseguire query. Un facet è un campo nei log (ad esempio `ServiceName` or`StatusCode`) che consente il filtraggio, l'aggregazione e l'analisi tra gruppi di log. È possibile visualizzare l'elenco dei campi sfaccettati nella console CloudWatch Logs Insights, insieme al conteggio degli eventi di registro per ogni valore di facet in base all'intervallo di tempo selezionato. Quando selezioni sfaccettature e valori diversi, i valori e i conteggi delle sfaccettature vengono aggiornati in tempo reale, consentendoti di esplorare in modo interattivo i log.

Ogni sfaccettatura mostra i valori e i conteggi disponibili estratti automaticamente dai campi dei log in base all'intervallo di tempo e all'ambito della query selezionati e conservati per 30 giorni. I conteggi delle sfaccettature mostrati sono approssimativi. Puoi utilizzare i facet predefiniti, come il nome dell'origine dati o il tipo di origine dati, per esplorare i log o creare facet personalizzati su qualsiasi campo dei log. Data Source name è un servizio o un'applicazione AWS che genera i log (ad esempio, Route 53, Amazon VPC CloudTrail o) e il tipo Data Source è il tipo specifico di log generato da quel servizio. I facet predefiniti vengono creati CloudWatch e includono`@aws.region`,`@data_source_name`, `@data_source_type` e. `@data_format` Per ulteriori informazioni, consulta Gestione dei registri. I facet sono disponibili solo per i log che vengono inseriti nell'account. Se hai impostato l'osservabilità tra account, l'account di monitoraggio non può visualizzare i facet in base ai log degli account di origine.

Se hai abilitato i tag di risorsa per la telemetria, i campi dei tag di risorsa (`@aws.tag.*`) vengono aggiunti automaticamente come facet predefiniti. Questi facet consentono di filtrare e raggruppare gli eventi di registro in base ai tag delle risorse che li hanno prodotti.

Per creare facet aggiuntivi, seleziona i campi nei log pertinenti alla risoluzione dei problemi e configurali utilizzando le politiche dell'indice. Per i facet personalizzati, consigliamo di crearli su campi a bassa cardinalità (campi con meno di 100 valori univoci al giorno come Status e). ApplicationName I facet con più di 100 valori univoci al giorno sono classificati come ad alta cardinalità e i relativi valori non vengono visualizzati. Seleziona uno o più facet e scegli di eseguire query nei log.

Per iniziare a usare i facet in Logs Insights: CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Logs, quindi Logs Insights.

3. (Facoltativo) Utilizzate il selettore dell'intervallo di tempo per selezionare un periodo di tempo da analizzare. Per l'intervallo di tempo selezionato, le sfaccettature e i valori disponibili vengono visualizzati nel pannello.
4. Seleziona le sfaccettature per esplorare i dati e visualizzare gli aggiornamenti in tempo reale delle distribuzioni del valore tra le sfaccettature.

   Le sfaccettature con più di 100 valori univoci non vengono visualizzate. Per interrogare valori specifici, utilizza invece i filtri nella query.

Per eseguire una query basata su facet

1. Seleziona uno o più valori tra le sfaccettature.
2. Il conteggio degli eventi verrà aggiornato in base alle sfaccettature e ai valori selezionati.
3. Quando i valori delle sfaccettature vengono selezionati, l'ambito della query viene aggiornato in modo da riflettere la selezione.
4. Dopo aver selezionato i valori dei facet, scegliete Esegui per eseguire la query.
5. Il numero massimo di valori univoci supportati per sfaccettatura è 100. Ad esempio, se sono presenti più di 100 valori per una sfaccettatura, tutti i conteggi vengono visualizzati come «-», a indicare che i valori sono sconosciuti.

Per salvare un'interrogazione basata su facet

1. Crea la tua query utilizzando uno o più valori di sfaccettatura.
2. Il resto dei passaggi è lo stesso del salvataggio di una query di Logs Insights. Vedi Salvare le interrogazioni di CloudWatch Logs Insights.
3. Le interrogazioni salvate sono disponibili nella sezione Interrogazioni salvate. Quando recuperi un'interrogazione salvata, questa includerà automaticamente i facet e i valori utilizzati per l'interrogazione, semplificando l'analisi dei log.

Per creare un facet a livello di account

1. Per creare facet, devi prima creare il campo come indice e configurarlo come facet. Nel riquadro di navigazione, scegli Impostazioni, Registri, Politiche di indicizzazione a livello di account. In alternativa, puoi selezionare Gestisci sfaccettature nel pannello delle sfaccettature.

2. Scegliete Crea nuova politica di indicizzazione. Per i dettagli sulla creazione di politiche di indicizzazione, consultaCrea una politica di indicizzazione dei campi a livello di account.
3. Per creare un facet, seleziona Imposta come facet per il campo selezionato nella pagina di creazione della politica dell'indice.

Gestione dei facet tramite API

La gestione delle sfaccettature può essere eseguita utilizzando la policy Field Index. Consulta le `field index`API per i dettagli.

API Field Index

No.	Nome	Description
1	PutIndexPolicy	Crea o aggiorna una politica di indicizzazione dei campi per il gruppo di log specifico
2	PutAccountPolicy	Crea una politica di protezione dei dati a livello di account, una politica di filtro di abbonamen to, una politica di indicizzazione dei campi, una politica di trasformazione o una politica di estrazione delle metriche che si applica a tutti i gruppi di log o a un sottoinsieme di gruppi di log nell'account
3	DeleteIndexPolicy	Elimina una politica di indice dei campi a livello di gruppo di log applicata a un singolo gruppo di log
4	DeleteAccountPolicy	Elimina una politica relativa all'account Logs CloudWatch

Visualizzazione dei log circostanti in Logs Insights CloudWatch

È possibile visualizzare gli eventi di registro che si sono verificati prima e dopo uno specifico record di registro restituito dalla query di CloudWatch Logs Insights. I log circostanti consentono di comprendere il contesto relativo a errori, avvisi o altri eventi significativi.

Come funzionano i registri circostanti

Quando si esegue una query di CloudWatch Logs Insights, ogni record di registro nei risultati include l'opzione Surrounding logs. Quando si sceglie questa opzione, CloudWatch Logs Insights visualizza eventi di registro aggiuntivi dello stesso flusso di log che si sono verificati immediatamente prima e dopo il record selezionato. Questi eventi vengono visualizzati anche se non corrispondono ai filtri di query originali.

È possibile configurare il numero di righe di registro circostanti da visualizzare. Scegli tra 5, 10, 20, 50 o 100 righe di registro sopra e sotto il record selezionato.

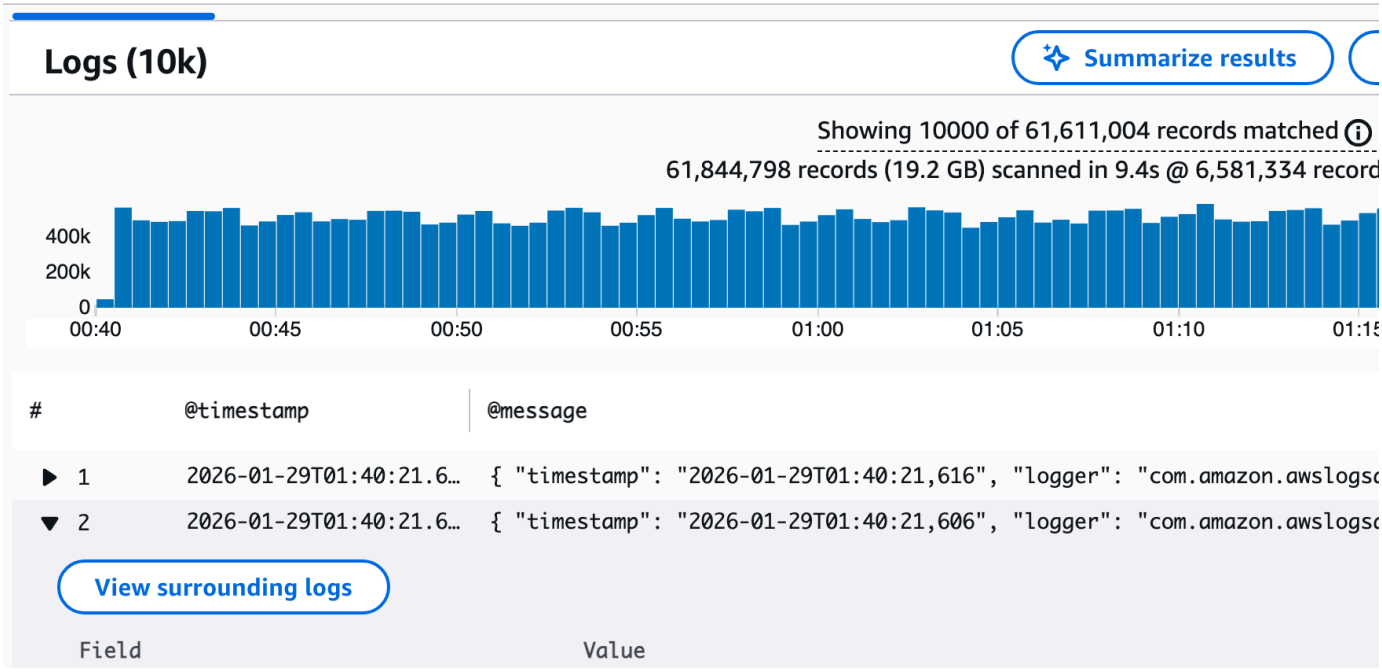
I log circostanti sono utili nei seguenti scenari:

- Eseguire il debug degli errori mediante la visualizzazione degli eventi che hanno causato un errore
- Analisi degli avvisi o dei timeout nei sistemi distribuiti
- Comprensione della sequenza degli eventi nell'applicazione
- Analisi dei problemi in ambienti di log ad alto volume

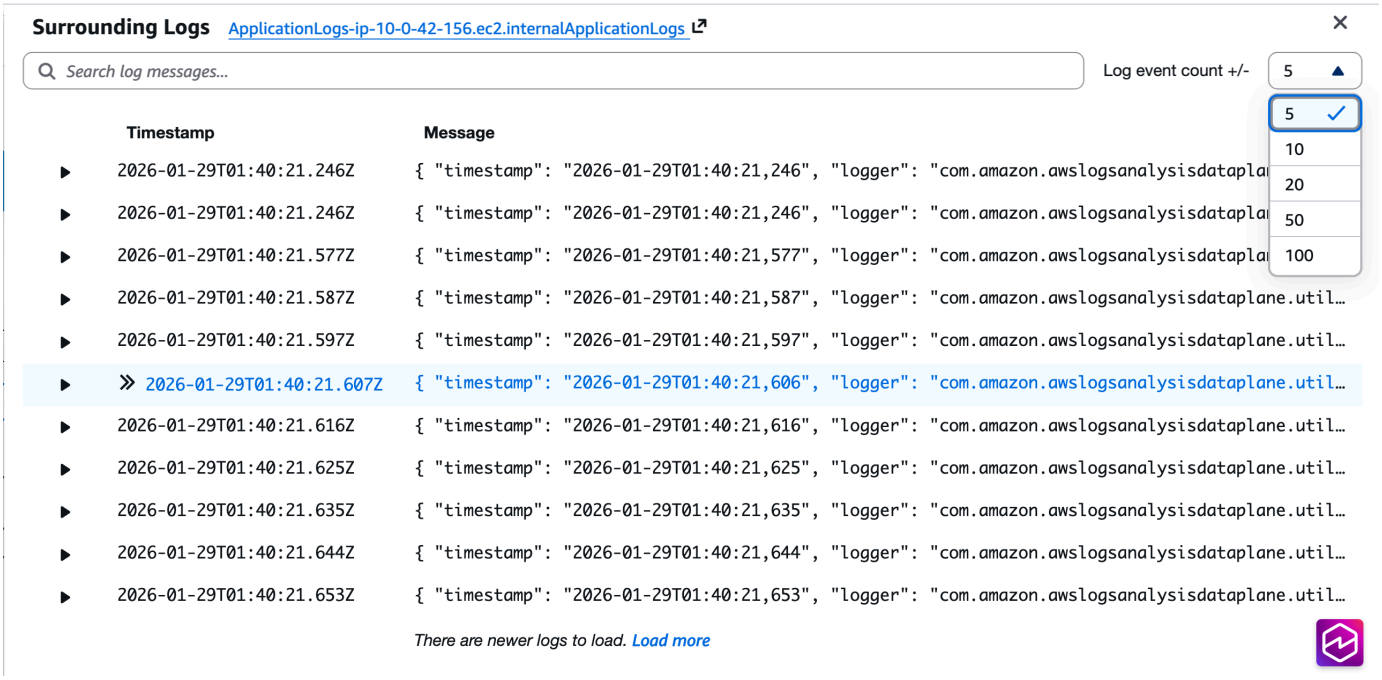
Visualizza i registri circostanti

Per visualizzare i registri circostanti

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs, quindi Logs Insights.
3. Eseguire la query.
4. Nei risultati della query, individua il record di registro che desideri esaminare.
5. Scegliete Registri circostanti.



6. Nel menu a discesa Numero di eventi, scegli il numero di eventi di registro da visualizzare sopra e sotto il record selezionato. Puoi scegliere 5, 10, 20, 50 o 100.

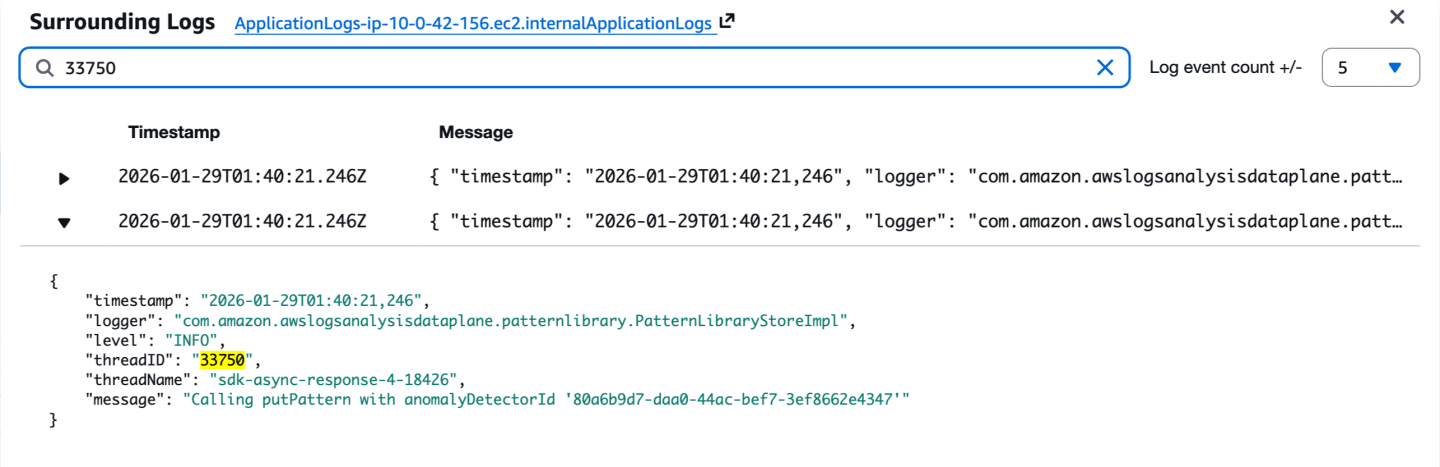


Cerca nei log circostanti

Dopo aver aperto il pannello dei log adiacenti, puoi cercare parole chiave specifiche per individuare il contesto pertinente.

Per effettuare una ricerca nei registri circostanti

1. Nel pannello dei log circostanti, inserisci il termine di ricerca nella casella di ricerca.
2. Controlla le righe di registro corrispondenti evidenziate.
3. Usa i controlli di navigazione per spostarti tra le partite.



Analisi del modello

CloudWatch Logs Insights utilizza algoritmi di apprendimento automatico per trovare modelli quando esegui una query sui log. Un pattern è una struttura di testo condivisa che ricorre tra i campi di registro. Quando visualizzi i risultati di una query, puoi scegliere la scheda Patterns per vedere i modelli trovati da CloudWatch Logs in base a un campione dei tuoi risultati. In alternativa, è possibile aggiungere il `pattern` comando alla query per analizzare i modelli nell'intero set di eventi di registro corrispondenti.

I pattern sono utili per analizzare set di log di grandi dimensioni perché spesso un gran numero di eventi di log può essere compresso in pochi pattern.

Considerate il seguente esempio di tre eventi di registro.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

Nell'esempio precedente, tutti e tre gli eventi di registro seguono uno schema:

```
<Time-1> [INFO] Calling DynamoDB to store for resource id <ID-2>
```

I campi all'interno di un pattern sono chiamati token. I campi che variano all'interno di uno schema, come l'ID della richiesta o il timestamp, sono token dinamici. Ogni token dinamico è rappresentato da. <*string*-*number*> *string*È una descrizione del tipo di dati rappresentato dal token. *number*Mostra dove nel modello appare questo token, rispetto agli altri token dinamici.

Esempi comuni di token dinamici includono codici di errore, timestamp e ID di richiesta. Un valore token rappresenta un valore particolare di un token dinamico. Ad esempio, se un token dinamico rappresenta un codice di errore HTTP, il valore del token potrebbe essere`501`.

Il rilevamento dei pattern viene utilizzato anche nel rilevatore di anomalie CloudWatch Logs e nelle funzionalità di confronto. Per ulteriori informazioni, consulta Rilevamento delle anomalie nei registri e Confronta (diff) con gli intervalli di tempo precedenti.

Guida introduttiva all'analisi dei pattern

Il rilevamento dei pattern viene eseguito automaticamente in qualsiasi query di CloudWatch Logs Insights. Le query che non includono il `pattern` comando ottengono sia gli eventi che i modelli di registro nei risultati.

Se si include il `pattern` comando nella query, l'analisi dei pattern viene eseguita sull'intero set di eventi di registro corrispondenti. Ciò consente di ottenere risultati di pattern più accurati, ma gli eventi di log non elaborati non vengono restituiti quando si utilizza il `pattern` comando. Quando una query non include`pattern`, i risultati del modello si basano sui primi 1000 eventi di registro restituiti o sul valore limite utilizzato nella query. Se si include `pattern` nella query, i risultati visualizzati nella scheda Patterns derivano da tutti gli eventi di registro corrispondenti alla query.

Per iniziare con l'analisi dei pattern in CloudWatch Logs Insights

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs, quindi Logs Insights.

   L'editor di query della pagina Logs Insights contiene una query predefinita che restituisce gli ultimi 20 log eventi.
3. Rimuovete la `| limit 20` riga nella casella di interrogazione, in modo che la query abbia il seguente aspetto:

   ```
   fields @timestamp, @message, @logStream, @log
   ```

```
| sort @timestamp desc
```

4. Nel menu a discesa Seleziona gruppi di log, scegli uno o più gruppi di log da interrogare.
5. (Facoltativo) Utilizza il selettore temporale per selezionare un periodo di tempo per il quale desideri eseguire query.

   Puoi scegliere tra intervalli di 5 e 30 minuti, intervalli di 1 ora, 3 ore e 12 ore o un intervallo di tempo personalizzato.
6. Scegli Esegui interrogazione per avviare l'interrogazione.

   Al termine dell'esecuzione della query, la scheda Registri visualizza una tabella degli eventi di registro restituiti dalla query. Sopra la tabella c'è un messaggio che indica quanti record corrispondono alla query, simile a Mostra 10.000 record su 71.101 corrispondenti.
7. Scegli la scheda Patterns.
8. La tabella ora mostra i modelli trovati nella query. Poiché la query non includeva il `pattern` comando, questa scheda mostra solo i modelli rilevati tra i 10.000 eventi di registro mostrati nella tabella della scheda Registri.

   Per ogni pattern, vengono visualizzate le seguenti informazioni:

   - Il modello, con ogni token dinamico visualizzato come<*string*-*number*>. *string*È una descrizione del tipo di dati rappresentato dal token. *number*Mostra dove nel modello appare questo token, rispetto agli altri token dinamici.
   - Il conteggio degli eventi, che è il numero di volte in cui il pattern è apparso negli eventi del registro interrogati. Scegli l'intestazione della colonna Conteggio degli eventi per ordinare i modelli in base alla frequenza.
   - Il rapporto degli eventi, che è la percentuale degli eventi di registro interrogati che contengono questo modello.
   - Il tipo di severità, che sarà uno dei seguenti:
     - ERRORE se il pattern contiene la parola Error.
     - AVVISA se il pattern contiene la parola Warn ma non contiene Error.
     - INFO se il pattern non contiene né Warn né Error.

   Scegli l'intestazione della colonna Informazioni sulla gravità per ordinare i modelli in base alla gravità.

9. Ora cambia la query. Sostituite la `| sort @timestamp desc` riga dell'interrogazione con `| pattern @message`, in modo che l'interrogazione completa sia la seguente:

```
fields @timestamp, @message, @logStream, @log
| pattern @message
```

10. Scegli Esegui query.

    Al termine dell'interrogazione, non ci sono risultati nella scheda Registri. Tuttavia, è probabile che la scheda Patterns contenga un numero maggiore di pattern elencati, a seconda del numero totale di eventi di registro che sono stati interrogati.

11. Indipendentemente dal fatto che l'interrogazione sia stata inclusa `pattern` o meno, è possibile esaminare ulteriormente i pattern restituiti dalla query. A tale scopo, scegliete l'icona nella colonna Ispeziona per uno dei motivi.

    Viene visualizzato il riquadro Pattern Inspect che mostra quanto segue:

    - Il modello. Seleziona un token all'interno del pattern per analizzare i valori di quel token.
    - Un istogramma che mostra il numero di occorrenze del pattern nell'intervallo di tempo richiesto. Questo può aiutarvi a identificare tendenze interessanti, ad esempio un aumento improvviso della ricorrenza di un pattern.
    - La scheda Registra esempi mostra alcuni degli eventi di registro che corrispondono al modello selezionato.
    - La scheda Valori dei token mostra i valori del token dinamico selezionato, se ne è stato selezionato uno.

      > **Note**
      >
      > Per ogni token vengono acquisiti un massimo di 10 valori di token. Il conteggio dei token potrebbe non essere preciso. CloudWatch Logs utilizza un contatore probabilistico per generare il conteggio dei token, non il valore assoluto.

    - La scheda Schemi correlati mostra altri motivi che si sono verificati frequentemente più o meno nello stesso periodo del modello che si sta esaminando. Ad esempio, se lo schema di un `ERROR` messaggio era in genere accompagnato da un altro evento di registro contrassegnato `INFO` con dettagli aggiuntivi, tale modello viene visualizzato qui.

Dettagli sul comando pattern

Questa sezione contiene ulteriori dettagli sul `pattern` comando e sui relativi utilizzi.

- Nel tutorial precedente, abbiamo rimosso il `sort` comando quando l'abbiamo aggiunto `pattern` perché una query non è valida se include un `pattern` comando dopo un `sort` comando. È valido avere un `pattern` precedente `asort`.

  Per ulteriori dettagli sulla `pattern` sintassi, vederepattern.
- Quando si utilizza `pattern` in una query, `@message` deve essere uno dei campi selezionati nel `pattern` comando.
- È possibile includere il `filter` comando prima di un `pattern` comando per fare in modo che solo l'insieme filtrato di eventi di registro venga utilizzato come input per l'analisi dei modelli.
- Per visualizzare i risultati del pattern per un campo particolare, ad esempio un campo derivato dal `parse` comando, usa`pattern @fieldname`.
- Le query con output non di registro, ad esempio le query con il `stats` comando, non restituiscono risultati di pattern.

Salva ed esegui nuovamente le query di Logs Insights CloudWatch

Dopo aver creato una query, è possibile salvarla in modo da poterla eseguire di nuovo in un secondo momento. Le interrogazioni vengono salvate in una struttura di cartelle, in modo da poterle organizzare. Puoi salvare fino a 1000 query per regione per account.

Le interrogazioni vengono salvate a un Region-specific livello, non a un livello specifico dell'utente. Se si crea e si salva una query, gli altri utenti con accesso ai CloudWatch registri nella stessa area possono visualizzare tutte le interrogazioni salvate e le relative strutture di cartelle nella regione.

Per salvare una query, è necessario accedere a un ruolo che dispone dell'autorizzazione `logs:PutQueryDefinition`. Per visualizzare un elenco di query salvate, è necessario accedere a un ruolo che dispone dell'autorizzazione `logs:DescribeQueryDefinitions`.

> **Note**
>
> È possibile creare e salvare interrogazioni con parametri, ovvero modelli riutilizzabili con segnaposto denominati. Invece di salvare più varianti della stessa interrogazione con valori diversi, create un modello e fornite valori di parametro diversi quando lo eseguite. Questa

funzionalità è attualmente supportata solo per le query che utilizzano il linguaggio di query Logs Insights. Per ulteriori informazioni, vedere Utilizzo delle interrogazioni salvate con parametri.

Console

Per salvare un'interrogazione

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. Nell'editor di query, crea una query.
4. Selezionare Salva.
5. Immettere un nome per la query.
6. (Facoltativo) Scegliere una cartella in cui si desidera salvare la query. Selezionare Create new (Crea nuova) per creare una cartella. Se si crea una nuova cartella, è possibile utilizzare i caratteri barra (/) nel nome della cartella per definire una struttura di cartelle. Ad esempio, la denominazione di una nuova cartella **`folder-level-1/folder-level-2`** crea una cartella di livello superiore denominata **`folder-level-1`**, con un'altra cartella chiamata **`folder-level-2`** all'interno di tale cartella. La query viene salvata in **`folder-level-2`**.
7. (Facoltativo) Modificare i gruppi di log o il testo della query.
8. (Facoltativo) Per utilizzare i parametri nella query, segui questi passaggi aggiuntivi:
   a. Aggiungi parametri alla tua query. Sostituisci i valori statici con segnaposto utilizzando la `{{parameter}}` sintassi (doppie parentesi prima e dopo il nome del parametro).

      Esempio: interrogazione originale con valori statici:

      ```
      fields @timestamp, @message
      | filter level = "Error"
      | filter applicationName = "OrderService"
      ```

      Interrogazione aggiornata con parametri:

      ```
      fields @timestamp, @message
      | filter level = {{logLevel}}
      | filter applicationName = {{applicationName}}
      ```

b. Definire i parametri utilizzati nella query. Per ogni parametro segnaposto, specificate:

- Nome: deve corrispondere esattamente al nome del segnaposto (ad esempio,). `logLevel` `applicationName`
- Valore predefinito (opzionale): il valore da utilizzare se non viene fornito alcun valore di parametro.
- Descrizione (opzionale): spiega lo scopo del parametro.

c. Le interrogazioni con parametri possono essere eseguite utilizzando il nome della query con un $ prefisso e passando i nomi dei parametri come coppie chiave-valore. Per ulteriori informazioni, vedere Eseguire un'interrogazione salvata.

9. Selezionare Salva.

AWS CLI

Per salvare un'interrogazione, usa`put-query-definition`:

```
aws logs put-query-definition \
  --name "ErrorsByLevel" \
  --query-string "fields @timestamp, @message | filter level = \"ERROR\"" \
  --log-group-names "/aws/lambda/my-function" \
  --region us-east-1
```

(Facoltativo) Per salvare un'interrogazione con parametri, aggiungete l'`--parameters`opzione e utilizzate i `{{parameterName}}` segnaposto nella stringa di query:

```
aws logs put-query-definition \
  --name "ErrorsByLevel" \
  --query-string "fields @timestamp, @message | filter level = {{logLevel}} | filter
 applicationName = {{applicationName}}" \
  --parameters '[{"name":"logLevel","defaultValue":"ERROR","description":"Log level
 to filter"},
{"name":"applicationName","defaultValue":"OrderService","description":"Application
 name to filter"}]' \
  --log-group-names "/aws/lambda/my-function" \
  --region us-east-1
```

Per salvare un'interrogazione in una cartella, aggiungete il percorso della cartella come prefisso al nome della query:

```
aws logs put-query-definition \
  --name "my-folder/ErrorsByLevel" \
  --query-string "fields @timestamp, @message | filter level = {{logLevel}}" \
  --parameters '[{"name":"logLevel","defaultValue":"ERROR","description":"Log level
 to filter"}]' \
  --log-group-names "/aws/lambda/my-function" \
  --region us-east-1
```

API

Per salvare un'interrogazione, chiama PutQueryDefinition:

```
{
  "name": "ErrorsByLevel",
  "queryString": "fields @timestamp, @message | filter level = \"ERROR\"",
  "logGroupNames": ["/aws/lambda/my-function"]
}
```

(Facoltativo) Per salvare un'interrogazione con parametri, includete il `parameters` campo e utilizzate i `{{parameterName}}` segnaposto nella stringa di query:

```
{
  "name": "ErrorsByLevel",
  "queryString": "fields @timestamp, @message | filter level = {{logLevel}} | filter
 applicationName = {{applicationName}}",
  "logGroupNames": ["/aws/lambda/my-function"],
  "parameters": [
    {
      "name": "logLevel",
      "defaultValue": "ERROR",
      "description": "Log level to filter"
    },
    {
      "name": "applicationName",
      "defaultValue": "OrderService",
      "description": "Application name to filter"
    }
  ]
}
```

> **Tip**
>
> Puoi creare una cartella per le query salvate con`PutQueryDefinition`. Per creare una cartella per le query salvate, utilizza una barra (/) per anteporre al nome della query desiderata il nome della cartella: <*folder-name*>/<*query-name*>. Per ulteriori informazioni su questa azione, vedere. PutQueryDefinition

Console

Per eseguire una query salvata

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. A destra, scegliere Query.
4. Seleziona la tua interrogazione dall'elenco Interrogazioni salvate. Il testo della query viene visualizzato nell'editor delle query.
5. (Facoltativo) Per utilizzare un'interrogazione con parametri:
   a. Scegliete l'icona + accanto al nome dell'interrogazione nel pannello laterale Interrogazioni salvate.
   b. L'interrogazione con parametri viene visualizzata nell'editor delle interrogazioni. Ad esempio, se si sceglie l'icona + accanto a`ErrorsByLevel`, l'editor di query viene popolato con: `$ErrorsByLevel(level=, applicationName=)`
   c. Fornire i valori per i parametri (level, ApplicationName) ed eseguire la query. Ad esempio, `$ErrorsByLevel(level= "ERROR", applicationName= "OrderService")`
6. Scegli Esegui.

AWS CLI

Per eseguire un'interrogazione salvata con parametri

Utilizzare `start-query` con la `$QueryName()` sintassi:

```
aws logs start-query \
```

```
  --log-group-names "/aws/lambda/my-function" \
  --start-time 1707566400 --end-time 1707570000 \
  --query-string '$ErrorsByLevel(level= "ERROR", applicationName= "OrderService")' \
  --region us-east-1
```

API

Per eseguire un'interrogazione salvata con parametri

Chiama StartQuerycon la `$QueryName( )` sintassi presente nel `queryString` campo:

```
{
  "logGroupNames": ["/aws/lambda/my-function"],
  "startTime": 1707566400,
  "endTime": 1707570000,
  "queryString": "$ErrorsByLevel(level=\"ERROR\", applicationName= \"OrderService
\")"
}
```

Per salvare una nuova versione di una query salvata

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. A destra, scegliere Query.
4. Selezionare la query dall'elenco Saved queries (Query salvate). Viene visualizzata nell'editor di query.
5. Modificare la query. Se è necessario eseguire la query per controllare il proprio lavoro, scegliere Run query (Esegui query).
6. Quando si è pronti per salvare la nuova versione, scegliere Actions (Operazioni), Save as (Salva con nome).
7. Immettere un nome per la query.
8. (Facoltativo) Scegliere una cartella in cui si desidera salvare la query. Selezionare Create new (Crea nuova) per creare una cartella. Se si crea una nuova cartella, è possibile utilizzare i caratteri barra (/) nel nome della cartella per definire una struttura di cartelle. Ad esempio, la denominazione di una nuova cartella **`folder-level-1/folder-level-2`** crea una cartella di livello superiore denominata **`folder-level-1`**, con un'altra cartella chiamata **`folder-level-2`** all'interno di tale cartella. La query viene salvata in **`folder-level-2`**.

9. (Facoltativo) Modificare i gruppi di log o il testo della query.
10. Selezionare Salva.

Per eliminare una query, è necessario accedere a un ruolo che dispone dell'autorizzazione `logs:DeleteQueryDefinition`.

Per modificare o eliminare una query salvata

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. A destra, scegliere Query.
4. Selezionare la query dall'elenco Saved queries (Query salvate). Viene visualizzata nell'editor di query.
5. Scegliere Actions (Operazioni), Edit (Modifica) oppure Actions (Operazioni), Delete (Elimina).

Utilizzo di interrogazioni salvate con parametri

Le interrogazioni salvate con parametri sono modelli di query riutilizzabili con segnaposto denominati. Invece di conservare più copie di interrogazioni quasi identiche, è possibile salvare un modello e fornire valori di parametro diversi durante l'esecuzione della query. I parametri sono supportati solo nel linguaggio di interrogazione CloudWatch Logs Insights.

Come funziona

Quando si salva un'interrogazione, i segnaposto identificano i valori che è possibile fornire al momento dell'esecuzione della query. I segnaposti utilizzano la sintassi. `{{parameterName}}`
Di seguito è riportato un esempio di interrogazione salvata denominata `ErrorsByLevel` con due parametri e. `logLevel` `applicationName`

```
fields @timestamp, @message
| filter level = {{logLevel}}
| filter applicationName = {{applicationName}}
```

Per eseguire un'interrogazione salvata, è possibile richiamarla utilizzando il nome dell'interrogazione preceduto da $ e passando i valori dei parametri. Il motore di query CloudWatch Logs Insights sostituisce ogni segnaposto. Se un parametro contiene valori predefiniti, tali valori vengono utilizzati se non vengono forniti altri valori.

```
# Run query by using query name and passing parameter values explicitly
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService")

# Run query without specifying parameter values - default values are used in this case.
$ErrorsByLevel()
```

I nomi delle interrogazioni salvate contenenti spazi o caratteri speciali devono essere racchiusi tra segni di spunta inversa:

```
$`Errors By Level`(logLevel = "WARN")
```

Esempi di interrogazioni salvate con parametri

Aggiungere un limite di risultati come parametro

Nome della query: `ErrorsByLevel` con parametri `logLevel` (default:`"ERROR"`), `applicationName` (default:`"OrderService"`) e `maxResults` (default:`50`)

```
fields @timestamp, @message, @logStream
| filter level = {{logLevel}}
| filter applicationName = {{applicationName}}
| sort @timestamp desc
| limit {{maxResults}}
```

```
# Run the query using the query name and passing parameter values
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService", maxResults = 100)
```

Utilizzo di più interrogazioni salvate con parametri

L'esempio seguente utilizza `ErrorsByLevel` una seconda query salvata `RecentN` che è definita come `sort @timestamp desc | limit {{count}}` (con parametro`count`, impostazione predefinita`20`). Il motore di query CloudWatch Logs Insights espande ogni query prima di eseguirla.

```
# Using multiple queries with parameters in sequence
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService")
| $RecentN(count = 10)

# Each of the queries is expanded, resulting in the following query when it is run.
fields @timestamp, @message
```

```
| filter level = "WARN"
| filter applicationName = "OrderService"
| sort @timestamp desc
| limit 10
```

Quote e gestione degli errori

> **Note**
>
> Ogni interrogazione salvata può avere un massimo di 20 parametri.

La stringa di query espansa non può superare i 10.000 caratteri. I nomi dei parametri devono iniziare con una lettera o un carattere di sottolineatura. Un'interrogazione salvata non può fare riferimento a un'altra interrogazione salvata (le chiamate annidate non sono supportate).

Errori comuni

Errore	Causa
I parametri sono supportati solo per il linguaggio di interrogazione CWLI	I parametri sono supportati solo nel linguaggio di query CloudWatch Logs Insights.
Parametri obbligatori non trovati in QueryString	Il nome di un parametro in `--parameters` non ha una corrispondenza `{{placeholder}}` nella stringa di query.
Il numero di parametri supera il massimo di 20	Le interrogazioni salvate attualmente supportano solo 20 parametri.
Nome di parametro duplicato	La definizione della query contiene parametri duplicati. `parameters`

> **Note**
>
> Per creare o aggiornare un'interrogazione salvata con parametri, è necessaria l'`logs:PutQueryDefinition`autorizzazione. Per eseguirne una, è necessario `logs:StartQuery` e`logs:DescribeQueryDefinitions`.

Aggiunta di query a pannello di controllo o esportazione dei risultati della query

Dopo aver eseguito una query, puoi aggiungere la query a una CloudWatch dashboard o copiare i risultati negli Appunti.

Le query aggiunte ai pannelli di controllo vengono eseguite ogni volta che carichi il pannello di controllo e ogni volta che il pannello di controllo viene aggiornato. Queste query vengono conteggiate ai fini del limite di 100 query simultanee di Logs Insights CloudWatch .

Per aggiungere i risultati delle query a un pannello di controllo

1. Apri la console all'indirizzo. CloudWatch https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. Scegli uno o più gruppi di log ed esegui una query.
4. Scegli Add to dashboard (Aggiungi a pannello di controllo).
5. Seleziona il pannello di controllo, oppure scegli Create new (Crea nuovo) per creare un pannello di controllo per i risultati delle query.
6. Seleziona il tipo di widget da utilizzare per i risultati della query.
7. Inserisci un nome per il widget.
8. Scegli Add to dashboard (Aggiungi a pannello di controllo).

Per copiare i risultati della query negli appunti o scaricare i risultati della query

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. Scegli uno o più gruppi di log ed esegui una query.
4. Scegli Export results (Esporta risultati), quindi scegli l'opzione desiderata.

Visualizzazione di query in esecuzione o cronologia delle query

Puoi visualizzare le query attualmente in corso, nonché la cronologia delle query recenti.

Le query attualmente in esecuzione includono quelle aggiunte a un pannello di controllo. Hai un limite di 100 query simultanee di CloudWatch Logs Insights per account, incluse le query aggiunte

ai dashboard. Inoltre, è possibile eseguire 15 query simultanee per Service PPL o Service SQL. OpenSearch OpenSearch

Per visualizzare la cronologia delle query recenti

1. Apri la console all' CloudWatch indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Logs (Registri), quindi Logs Insights.
3. Scegli Cronologia, se stai usando il nuovo design per la console CloudWatch Logs. Se utilizzi il vecchio design, scegli Actions (Operazioni), View query history for this account (Visualizza cronologia query per questo account).

   Viene visualizzato un elenco delle query recenti. Puoi eseguire di nuovo qualsiasi query selezionandola e scegliendo Run (Esegui).

   In Stato, viene visualizzato il CloudWatch messaggio Registri in corso per tutte le interrogazioni attualmente in esecuzione.

Crittografa i risultati delle interrogazioni con AWS Key Management Service

Per impostazione predefinita, CloudWatch Logs crittografa i risultati archiviati delle query di CloudWatch Logs Insights utilizzando il metodo di crittografia predefinito Logs lato server. CloudWatch Puoi invece scegliere di utilizzare una chiave per crittografare questi risultati AWS KMS . Se associ una AWS KMS chiave ai risultati della crittografia, CloudWatch Logs utilizza quella chiave per crittografare i risultati archiviati di tutte le query nell'account.

Se successivamente si dissocia una chiave dai risultati della query, CloudWatch Logs torna al metodo di crittografia predefinito per le query successive. Tuttavia, le query eseguite mentre la chiave era associata sono ancora crittografate con quella chiave. CloudWatch I log possono comunque restituire tali risultati dopo la dissociazione della chiave KMS, poiché CloudWatch i registri possono continuare a fare riferimento alla chiave. Tuttavia, se la chiave viene successivamente disabilitata, CloudWatch Logs non è in grado di leggere i risultati della query che sono stati crittografati con quella chiave.

> **Important**
>
> CloudWatch Logs supporta solo chiavi KMS simmetriche. Non utilizzare una chiave asimmetrica per crittografare i risultati della query. Per ulteriori informazioni, consulta la sezione relativa all'uso di chiavi simmetriche e asimmetriche.

Limits

- Per eseguire la procedura seguente, devi avere le seguenti autorizzazioni: `kms:CreateKey`, `kms:GetKeyPolicy` e `kms:PutKeyPolicy`.
- Dopo aver associato o dissociato una chiave dai risultati della query, possono essere necessari fino a cinque minuti per rendere effettiva l'operazione.
- Se CloudWatch revochi l'accesso dei log a una chiave associata o elimini una chiave KMS associata, i dati crittografati in Logs non possono più essere recuperati. CloudWatch
- Non puoi utilizzare la CloudWatch console per associare una chiave, devi utilizzare l'API o Logs. AWS CLI CloudWatch

Passaggio 1: creare un AWS KMS key

Per creare una chiave KMS, utilizza il seguente comando create-key:

```
aws kms create-key
```

L'output contiene l'ID chiave e l'Amazon Resource Name (ARN) della chiave. Di seguito è riportato un output di esempio:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
```

```
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Fase 2: Impostazione delle autorizzazioni sulla chiave KMS

Per impostazione predefinita, tutte le chiavi KMS sono private. Solo il proprietario della risorsa può utilizzarla per crittografare e decrittare i dati. Tuttavia, il proprietario della risorsa può concedere ad altri utenti e risorse le autorizzazioni per accedere alla chiave. Con questo passaggio, si CloudWatch concede al servizio Logs l'autorizzazione principale a utilizzare la chiave. L'entità del servizio deve trovarsi nella stessa AWS regione in cui è memorizzata la chiave.

Come procedura ottimale, si consiglia di limitare l'uso della chiave solo agli AWS account specificati.

In primo luogo, salva la policy predefinita per la chiave KMS come `policy.json` utilizzando il seguente comando get-key-policy:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Aprire il file `policy.json` in un editor di testo e aggiungere la sezione in grassetto da una delle seguenti istruzioni. Separare l'istruzione esistente dalla nuova istruzione con una virgola. Queste istruzioni utilizzano `Condition` sezioni per migliorare la sicurezza della AWS KMS chiave. Per ulteriori informazioni, consulta AWS KMS chiavi e contesto di crittografia.

La `Condition` sezione di questo esempio limita l'uso della AWS KMS chiave ai risultati della query CloudWatch Logs Insights nell'account specificato.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
```

```
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                "aws:SourceArn": "arn:aws:logs:us-east-1:111122223333:query-
result:*"
                },
                "StringEquals": {
                "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Infine, aggiungi la policy aggiornata utilizzando il comando put-key-policy seguente:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Fase 3: associazione di una chiave KMS ai risultati della query

Associazione della chiave KMS ai risultati della query nell'account

Utilizza il comando disassociate-kms-key come riportato di seguito:

```
aws logs associate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-result:*" --kms-key-id "key-arn"
```

Fase 4: Dissociazione di una chiave dai risultati della query nell'account

Per dissociare la chiave KMS associata ai risultati della query, utilizza il seguente comando disassociate-kms-key:

```
aws logs disassociate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-result:*"
```

Genera un riepilogo in linguaggio naturale dai risultati delle query di CloudWatch Logs Insights

L'analisi dei dati di registro è fondamentale per comprendere il comportamento delle applicazioni, ma l'interpretazione di grandi volumi di voci di registro può richiedere molto tempo. CloudWatch Logs Insights offre ora una funzionalità di riepilogo in linguaggio naturale che trasforma i risultati di query complesse in riepiloghi chiari e concisi. Questa funzionalità consente di identificare rapidamente i problemi e ottenere informazioni utili dai dati di registro.

Come funziona

CloudWatch Logs Insights può generare un riepilogo leggibile dai risultati delle tue query utilizzando Amazon Bedrock. La funzionalità supporta tutti i linguaggi di interrogazione di CloudWatch Logs Insights e fornisce informazioni chiare e fruibili a partire dai dati di log.

Disponibilità regionale ed elaborazione dei dati

> **Important**
>
> Quando si utilizza questa funzionalità, i risultati della query potrebbero essere elaborati in un modo diverso Regione AWS. Ad esempio, se si esegue una query negli Stati Uniti orientali (Virginia settentrionale), il riepilogo potrebbe avvenire negli Stati Uniti occidentali (Oregon).

La tabella seguente elenca l'elaborazione possibile Regione AWS per le diverse aree geografiche in cui è disponibile la funzionalità dei risultati delle query:

Geografia dei log supportata CloudWatch	Regione di elaborazione possibile
Stati Uniti (US)	Stati Uniti orientali (Virginia settentrionale) US East (Ohio) Region US West (Oregon) Region
Europa	Regione Europa (Francoforte) Regione Europa (Irlanda) Regione Europa (Parigi) Regione Europa (Stoccolma) Regione Europa (Londra)
Asia Pacifico	Stati Uniti orientali (Virginia settentrionale) US East (Ohio) Region US West (Oregon) Region
Sud America	Stati Uniti orientali (Virginia settentrionale) US East (Ohio) Region US West (Oregon) Region

Nozioni di base

Per generare un riepilogo in linguaggio naturale

1. Esegui la tua query CloudWatch Logs Insights.
2. Al termine della query, seleziona Riepiloga risultati.

Permissions

È necessario disporre di una delle seguenti opzioni:

- Autorizzazione `CloudWatchLogsFullAccess`
- Autorizzazione `CloudWatchLogsReadOnlyAccess`
- Policy IAM personalizzata che include le `cloudwatch:GenerateQueryResultsSummary` `logs:FilterLogEvents` azioni `logs:DescribeQueries` e `logs:GetQueryResults`

Privacy dei dati

I risultati delle query vengono elaborati in modo sicuro e non vengono utilizzati per addestrare o migliorare CloudWatch Logs Insights o Amazon Bedrock. Se scegli di fornire un feedback sul riepilogo dei risultati della query utilizzando i pulsanti di feedback, il feedback indica il tuo livello di soddisfazione rispetto alla funzionalità fornita in CloudWatch Logs Insights.

Automatizzazione dell'analisi dei log con interrogazioni pianificate

Le interrogazioni pianificate consentono di automatizzare l'esecuzione delle query di registro in base a una pianificazione regolare. Invece di eseguire manualmente le query per analizzare i dati di log, puoi configurare le query pianificate in modo che vengano eseguite automaticamente e forniscano risultati a destinazioni come bucket Amazon S3, bus di eventi Amazon o tabelle di ricerca. EventBridge Questa automazione è ideale per generare report regolari, monitorare le tendenze, mantenere aggiornate le tabelle di ricerca con i dati di registro più recenti o attivare processi a valle in base ai risultati dell'analisi dei log.

Le interrogazioni pianificate supportano tutti e tre i linguaggi di interrogazione disponibili per le query di log:

- Linguaggio di interrogazione Logs Insights (Logs Insights QL)
- OpenSearch Service Piped Processing Language (PPL)
- OpenSearch Service Structured Query Language (SQL)

Indice

- Comprensione dei concetti relativi alle interrogazioni pianificate
- Riferimento alle espressioni di pianificazione
- Best practice
- Guida introduttiva alle query pianificate
- Configurazione delle destinazioni S3 per le interrogazioni pianificate
- Configurazione delle destinazioni delle tabelle di ricerca per le interrogazioni pianificate
- Risoluzione dei problemi relativi alle interrogazioni pianificate

Comprensione dei concetti relativi alle interrogazioni pianificate

Prima di creare interrogazioni pianificate, è necessario comprendere questi concetti chiave che influiscono sulla modalità di esecuzione delle query e sul luogo in cui vengono forniti i risultati.

Separazione dei ruoli IAM

Le query pianificate richiedono due ruoli IAM separati: uno per l'esecuzione delle query e l'altro per fornire risultati a destinazioni come bucket Amazon S3, bus di EventBridge eventi Amazon o tabelle di ricerca. Comprendere il motivo per cui esiste questa separazione ti aiuta a configurare correttamente le autorizzazioni e a utilizzare i vantaggi operativi e di sicurezza che offre.

L'architettura a due ruoli divide le responsabilità tra accesso e distribuzione dei dati. Il ruolo di esecuzione delle query accede ai dati di registro ed esegue le query, mentre il ruolo di destinazione di consegna scrive i risultati nella destinazione prescelta. Questa separazione segue il principio del privilegio minimo: ogni ruolo dispone solo delle autorizzazioni necessarie per la sua funzione specifica.

Ruolo di esecuzione delle interrogazioni

Consente a CloudWatch Logs di eseguire query CloudWatch Logs Insights per tuo conto. Questo ruolo richiede le autorizzazioni per accedere ai gruppi di log ed eseguire le query, ma non ha bisogno dell'accesso alle risorse di destinazione. Autorizzazioni richieste:

- `logs:StartQuery`
- `logs:StopQuery`
- `logs:GetQueryResults`
- `logs:DescribeLogGroups`
- `logs:Unmaskse` è richiesto l'annullamento della mascheratura dei dati

Per i gruppi di KMS-encrypted log: `kms:Decrypt` e `kms:DescribeKey` le autorizzazioni per la chiave KMS utilizzata per crittografare i gruppi di log. Anche queste autorizzazioni devono essere aggiunte.

Requisito della relazione di fiducia: il ruolo di esecuzione della query deve includere una politica di fiducia che consenta al servizio CloudWatch Logs (`logs.amazonaws.com`) di assumere il ruolo. Senza questa relazione di fiducia, le query pianificate falliranno con errori di autorizzazione.

Esempio di politica di fiducia per il ruolo di esecuzione della query:

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

```
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Esempio di politica di autorizzazione per il ruolo di esecuzione della query:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:GetQueryResults",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        }
    ]
}
```

Ruolo di consegna della destinazione

Consente a CloudWatch Logs di fornire i risultati delle query alla destinazione prescelta. Questo ruolo richiede solo le autorizzazioni per il servizio di destinazione specifico, in base al principio del privilegio minimo. Le autorizzazioni richieste variano in base al tipo di destinazione.

Requisito della relazione di fiducia: il ruolo di consegna della destinazione deve includere anche una politica di fiducia che consenta al servizio CloudWatch Logs (`logs.amazonaws.com`) di assumere il ruolo.

Esempio di politica di autorizzazione per il ruolo di consegna a destinazione S3:

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::your-scheduled-query-results-bucket/*"
        }
    ]
}
```

Esempio di politica delle autorizzazioni per un ruolo di consegna della destinazione nella tabella di ricerca:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLookupTable",
                "logs:UpdateLookupTable",
                "logs:GetQueryResults"
            ],
            "Resource": "*"
        }
    ]
}
```

Questa separazione offre vantaggi pratici per le tue operazioni. Dal punto di vista della sicurezza, se è necessario modificare il luogo di consegna dei risultati, è sufficiente modificare il ruolo di destinazione senza modificare le autorizzazioni di esecuzione delle query. Per la conformità e il controllo, puoi tracciare chiaramente quale ruolo accede ai dati di registro sensibili e quale ruolo scrive su sistemi esterni. In questo modo è più facile dimostrare che l'infrastruttura di analisi dei log segue le best practice di sicurezza.

Cross-region e utilizzo su più account

Una query pianificata viene creata in una regione specifica e viene eseguita in quella regione. Tuttavia, puoi interrogare i gruppi di log e fornire risultati in tutte le regioni e gli account. È necessario

configurare uno o più AWS account come account di monitoraggio e collegarli a più account di origine. Un account di monitoraggio è un AWS account centrale in grado di visualizzare e interagire con i dati di osservabilità generati dagli account di origine. Un account di origine è un AWS account individuale che genera dati di osservabilità per le risorse che vi risiedono. Gli account di origine condividono i dati di osservabilità con l'account di monitoraggio. Quindi puoi impostare query pianificate dall'account di monitoraggio utilizzando i gruppi di log di tutti gli account collegati.

Interrogazione di gruppi di log interregionali

L'interrogazione pianificata può accedere ai gruppi di log in qualsiasi regione. Specifica i gruppi di log utilizzando il loro formato ARN completo:`arn:aws:logs:region:account-id:log-group:log-group-name`. I requisiti `logs:StartQuery` e le `logs:GetQueryResults` autorizzazioni del ruolo di esecuzione delle query per i gruppi di log in tutte le regioni di destinazione.

Important

Quando si interrogano i gruppi di log o si forniscono risultati in più regioni, i dati di log superano i confini regionali. Considera i seguenti aspetti:

- Requisiti di residenza dei dati: assicurati che il trasferimento dei dati tra regioni sia conforme alle politiche di governance dei dati e ai requisiti normativi della tua organizzazione
- Costi di trasferimento dei dati: il trasferimento Cross-region dei dati comporta costi aggiuntivi
- Latenza di rete: le query che accedono a gruppi di log in regioni distanti possono presentare una latenza maggiore

Per prestazioni ottimali ed efficienza in termini di costi, crea query pianificate nella stessa regione dei gruppi di log principali.

Approccio alternativo: utilizza la centralizzazione dei CloudWatch log per replicare i dati di log provenienti da più account e regioni in un account di monitoraggio centrale. Ciò consente di creare query pianificate in un'unica regione che accedano a tutti i log centralizzati, evitando query interregionali e semplificando la gestione delle autorizzazioni IAM.

Pianifica le espressioni e la gestione dei fusi orari

La pianificazione definita determina quando viene eseguita la query e la frequenza di esecuzione. La scelta dell'espressione di pianificazione corretta influisce sul momento in cui si ricevono i risultati e sulla quantità di dati interrogati. La comprensione dei tipi di espressione consente di scegliere tra semplicità e precisione.

Le espressioni Cron forniscono un controllo preciso sulla tempistica, consentendo di specificare orari, giorni della settimana o giorni del mese esatti. Usa le espressioni cron quando hai bisogno che le query vengano eseguite in orari lavorativi specifici o in linea con le pianificazioni operative. Nella console puoi anche pianificare le query utilizzando semplici opzioni di calendario.

Espressioni Cron

Esegui le interrogazioni in momenti specifici. Formato:`cron(minute hour day-of-month month day-of-week year)`. Esempi:

- `cron(0 9 * * ? *)`- Ogni giorno alle 9:00 UTC
- `cron(0 18 ? * MON-FRI *)`- Giorni feriali alle 18:00 UTC
- `cron(0 0 1 * ? *)`- Il primo giorno di ogni mese a mezzanotte UTC
- `cron(0 12 ? * SUN *)`- Ogni domenica a mezzogiorno UTC
- `cron(30 8 1 1 ? *)`- 1° gennaio alle 8:30 UTC

Tutte le interrogazioni pianificate vengono eseguite in UTC, indipendentemente dal fuso orario locale o da dove si trovano le risorse. AWS Ciò è particolarmente importante quando si pianificano le query per l'orario di lavoro o per analisi sensibili al fattore tempo. Ad esempio, se la tua azienda opera nell'ora orientale degli Stati Uniti e desideri un rapporto giornaliero alle 9:00 ET, devi tenere conto della differenza UTC (14:00 UTC durante l'ora legale, 13:00 UTC altrimenti). Pianifica le espressioni di pianificazione tenendo conto dell'UTC per garantire che le query vengano eseguite negli orari previsti.

Scelta di un linguaggio di interrogazione

Le interrogazioni pianificate supportano tre diversi linguaggi di interrogazione e la tua scelta influisce sia sul modo in cui scrivi le query sia sulla facilità con cui il tuo team può gestirle. La lingua giusta dipende dai requisiti di analisi e dalle competenze esistenti del team.

Se stai principalmente filtrando e aggregando i dati di registro, CloudWatch Logs Insights Query Language offre la sintassi più semplice. Per trasformazioni di dati complesse in cui è necessario

rimodellare o arricchire i dati in più passaggi, l'approccio basato sulla pipeline di PPL semplifica l'applicazione della logica. Quando è necessario eseguire join o aggregazioni complesse simili alle operazioni di database, SQL fornisce una sintassi familiare che i team esperti di database possono adottare rapidamente.

CloudWatch Logs Insights Query Language (CWLI)

Purpose-built per l'analisi dei log con sintassi intuitiva. Ideale per:

- Text-based analisi e filtraggio dei log
- Time-series aggregazioni e statistiche
- Team che non conoscono l'analisi dei log

OpenSearch Service Piped Processing Language (PPL)

Pipeline-based linguaggio di interrogazione con potenti funzionalità di trasformazione dei dati. Ideale per:

- Trasformazioni e arricchimento di dati complessi
- Multi-step flussi di lavoro per l'elaborazione dei dati
- Team che hanno familiarità con l'elaborazione basata su pipeline

OpenSearch Service Structured Query Language (SQL)

Sintassi SQL standard per interrogazioni familiari in stile database. Ideale per:

- Unioni e aggregazioni complesse
- Business intelligence e reportistica
- Team con una forte esperienza in SQL

Selezione della destinazione e casi d'uso

Il luogo in cui invii i risultati delle query determina cosa puoi fare con essi. Questa scelta modella l'intero flusso di lavoro a valle, che si tratti di creare analisi a lungo termine, attivare risposte automatiche o entrambe le cose. Comprendere i punti di forza di ogni tipo di destinazione ti aiuta a progettare l'architettura giusta per il tuo caso d'uso.

Le destinazioni Amazon S3 sono ottimizzate per lo storage e l'elaborazione in batch. Quando devi conservare i risultati delle query per mesi o anni, analizzare le tendenze nel tempo o inserire dati in piattaforme di analisi, Amazon S3 offre uno storage conveniente con conservazione illimitata.

EventBridge le destinazioni sono ottimizzate per l'automazione in tempo reale. Quando i risultati delle query devono innescare azioni immediate, come l'invio di avvisi, l'avvio di flussi di lavoro o l'aggiornamento dei sistemi, i risultati si ottengono sotto forma EventBridge di eventi a cui le applicazioni possono rispondere istantaneamente. Per impostazione predefinita, tutti gli eventi di completamento delle query vengono inviati automaticamente come eventi al bus di eventi predefinito, consentendo l'integrazione con i sistemi di elaborazione a valle, le funzioni Lambda o altre architetture basate sugli eventi. I risultati vengono pubblicati nelle destinazioni solo quando la query viene eseguita correttamente. Le destinazioni delle tabelle di ricerca sono ottimizzate per mantenere aggiornati i dati di riferimento. Una destinazione della tabella di ricerca popola o aggiorna automaticamente la tabella di ricerca specificata con i risultati della query a ogni esecuzione pianificata, in modo che altre query possano fare riferimento ai dati più recenti con il comando. `lookup`

Destinazioni di Amazon S3

Archivia i risultati delle query come file JSON per la conservazione a lungo termine e l'elaborazione in batch. Le destinazioni Amazon S3 funzionano meglio per i seguenti scenari:

- Analisi storica e archiviazione dei dati
- Integrazione con data lake e piattaforme di analisi
- Requisiti di conformità e audit
- Cost-effective archiviazione di set di risultati di grandi dimensioni

EventBridge destinazioni

Invia i risultati delle query come eventi per l'elaborazione e l'automazione in tempo reale. Usa l'`queryId`in the event per recuperare i risultati della query, che rimangono disponibili per 30 giorni dopo l'esecuzione della query. EventBridgele destinazioni funzionano meglio per i seguenti scenari:

- Attivazione di risposte automatiche ai risultati delle query
- Integrazione con flussi di lavoro serverless e funzioni Lambda
- Real-time sistemi di allerta e notifica
- Event-driven architetture e microservizi

Destinazioni delle tabelle di ricerca

Crea o aggiorna automaticamente una tabella di ricerca con i risultati delle query su ogni esecuzione pianificata. Ogni aggiornamento sostituisce completamente il contenuto della tabella. Le destinazioni delle tabelle di ricerca funzionano meglio per i seguenti scenari:

- Mantenere aggiornati i dati di riferimento per il `lookup` comando nelle query di registro
- Gestione degli elenchi consentiti, delle denylist o degli inventari delle entità derivati dai dati di registro
- Arricchimento delle query con riepiloghi delle attività recenti, ad esempio elenchi di utenti o risorse attivi

Formato e struttura dei risultati delle query

Le query pianificate forniscono risultati in formato JSON, ma ogni tipo di destinazione riceve un payload diverso. Per una destinazione per una tabella di ricerca, i risultati della query diventano il contenuto della tabella di ricerca e ogni esecuzione sostituisce tale contenuto. Per ulteriori informazioni, consulta Configurazione delle destinazioni delle tabelle di ricerca per le interrogazioni pianificate.

Le destinazioni Amazon S3 ricevono le righe dei risultati della query. Ogni oggetto contiene un array JSON con una voce per ogni riga del set di risultati e ogni voce associa i nomi dei campi di output della query ai rispettivi valori. L'oggetto non contiene metadati o statistiche di interrogazione e omette il `@ptr` campo anche se la query lo richiede, poiché tale campo è utilizzabile solo nella console.

EventBridge le destinazioni ricevono i metadati delle query, incluse le statistiche delle query, ma nessuna riga dei risultati. Per recuperare le righe di una query completata, chiama GetQueryResults con il valore di `queryId` from the event.

L'esempio seguente mostra l'evento in cui CloudWatch Logs viene pubblicato EventBridge quando viene completata una query pianificata.

```
{
    "version": "0",
    "id": "be72061b-eca2-e068-a7e1-83e01d6fe807",
    "detail-type": "Scheduled Query Completed",
    "source": "aws.logs",
    "account": "123456789012",
    "time": "2025-11-18T11:31:48Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:logs:us-east-1:123456789012:scheduled-query:477b4380-b098-474e-9c5e-
e10a8cc2e6e7"
    ],
    "detail": {
```

```
        "queryId": "2038fd57-ab4f-4018-bb2f-61d363f4a004",
        "queryString": "fields @timestamp, @message, @logStream\n| filter @message
 like /ERROR/\n| sort @timestamp desc\n| limit 10000",
        "logGroupIdentifiers": [
            "/aws/lambda/my-function"
        ],
        "status": "Complete",
        "startTime": 1763465460,
        "statistics": {
            "recordsMatched": 1842,
            "recordsScanned": 48325,
            "estimatedRecordsSkipped": 0,
            "bytesScanned": 12081250,
            "estimatedBytesSkipped": 0,
            "logGroupsScanned": 1,
            "resultCount": 1842
        }
    }
}
```


Questa query non viene aggregata e ha restituito un numero di righe inferiore a 10.000, quindi ognuno `limit` dei 1.842 eventi di log corrispondenti è diventato una riga di output ed è uguale. `recordsMatched` `resultCount` Una query che aggrega, o una il cui set di risultati è troncato da, produce un valore inferiore a. `limit` `resultCount` `recordsMatched` Per ulteriori informazioni, consulta Informazioni su RecordsScanned, RecordsMatched e ResultCount.

Gli elementi chiave includono:

- `statistics`- Contatori che descrivono la quantità di dati di registro letti dalla query e la dimensione del set di risultati. Per una descrizione di ogni campo, vedere la tabella seguente.
- `startTime`- Quando è iniziata l'esecuzione della query (timestamp Unix)
- `queryString`- L'interrogazione effettiva che è stata eseguita
- `queryId`- ID della query utilizzando il quale è possibile recuperare i risultati
- `logGroupIdentifiers`- Elenco dei gruppi di log che sono stati interrogati
- `status`- Stato di esecuzione della query (Completata, Non riuscita, ecc.)

La tabella seguente descrive ogni campo dell'`statistics`oggetto. Per le definizioni API di questi campi, consulta QueryStatistics.

Campo	Description
`recordsScanned`	Il numero totale di eventi di registro scansionati durante la query.
`recordsMatched`	Il numero di eventi di registro che corrispondono alla stringa di query. Questo valore conta gli eventi di registro, non le righe di output. Per il numero di righe nel set di risultati, usa`resultCount` .
`resultCount`	Il numero di righe nel set di risultati dell'interrogazione. Questo valore conta solo le righe sopravvissute a tutte le operazioni della query, quindi potrebbe essere inferiore a`recordsMatched` . Copre tutte le pagine dei risultati `GetQueryResults` restituiti. Per ulteriori informazioni, consulta Informazioni su RecordsScanned, RecordsMatched e ResultCount.
`estimatedRecordsSkipped`	Una stima del numero di eventi di registro che sono stati ignorati durante l'elaborazione di questa query, poiché la query conteneva un campo indicizzato. Ignorare queste voci riduce i costi delle interrogazioni e ne migliora i tempi di esecuzione. Per ulteriori informazioni, consulta Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione.
`bytesScanned`	Il numero totale di byte negli eventi di log scansionati durante la query.
`estimatedBytesSkipped`	Una stima del numero di byte negli eventi di log che sono stati ignorati durante l'elaborazione di questa query, poiché la query conteneva un campo indicizzato.
`logGroupsScanned`	Il numero di gruppi di log scansionati da questa query.

Informazioni su RecordsScanned, RecordsMatched e ResultCount

Tre delle statistiche delle query contano cose diverse e confrontarle direttamente può essere fuorviante. Ognuna misura una diversa fase di elaborazione delle query:

- `recordsScanned`- Il numero di eventi di registro che la query ha letto dai tuoi gruppi di log. Questo è l'input per l'interrogazione.

- `recordsMatched`- Il numero di eventi di registro che corrispondono alla stringa di query. Questo valore conta gli eventi di registro.
- `resultCount`- Il numero di righe nel set di risultati prodotto dalla query. Questo valore conta le righe di output.

Ogni fase restringe i dati. Un comando come quello `stats` combina molti eventi di registro in un'unica riga di output, quindi `resultCount` può essere molto più piccolo di`recordsMatched`. Un valore grande `recordsMatched` con un valore piccolo non `resultCount` significa che nel set di risultati manchino delle righe.

Esempio: aggregazione

La seguente query conta i messaggi di errore in ogni flusso di log di un gruppo di log in un periodo di un'ora.

```
filter @message like /ERROR/
| stats count(*) as errorCount by @logStream
```

Se la query legge 1.500.000 eventi di registro, 24.318 dei quali contengono ERROR e gli eventi di registro corrispondenti provengono da 12 flussi di log, la query viene completata con le seguenti statistiche.

```
"statistics": {
    "recordsMatched": 24318,
    "recordsScanned": 1500000,
    "estimatedRecordsSkipped": 0,
    "bytesScanned": 450000000,
    "estimatedBytesSkipped": 0,
    "logGroupsScanned": 1,
    "resultCount": 12
}
```

`stats`produce una riga per ogni flusso di log, quindi `resultCount` è 12 mentre è 24.318. `recordsMatched` La somma dei 12 valori di è `errorCount` 24.318.

Esempio: filtro Post-aggregation

La seguente query mantiene solo i flussi di log che hanno prodotto più di 1.000 errori.

```
filter @message like /ERROR/
```

```
| stats count(*) as errorCount by @logStream
| filter errorCount > 1000
```

L'ultimo `filter` comando viene eseguito dopo il raggruppamento, quindi rimuove le righe dal set di risultati anziché registrare gli eventi dalla scansione. Se 3 dei 12 flussi di log hanno un valore `errorCount` superiore a 1.000, allora `resultCount` è 3 anziché 12. `recordsScanned`e `bytesScanned` non modificate, perché la query legge gli stessi dati di registro in entrambi i casi.

Esempio: limite

Un `limit` comando riduce `resultCount` senza alcuna aggregazione. La seguente query restituisce i 100 messaggi di errore più recenti.

```
filter @message like /ERROR/
| sort @timestamp desc
| limit 100
```

Se la query analizza gli stessi 1.500.000 eventi di registro e corrisponde agli stessi 24.318, allora `resultCount` è 100, perché limita il set di risultati `limit` a 100 righe. Nella console, questa relazione viene visualizzata come Visualizzazione della corrispondenza di 100 record su 24.318.

Ogni statistica risponde a una domanda diversa.

Quante righe ha restituito questa query?

> Utilizza `resultCount`. Un valore pari a 0 indica che la query non ha prodotto righe. Non utilizzarlo `recordsMatched` per questo scopo, poiché conta gli eventi di registro anziché le righe.

Quanti dati di registro ha letto questa query?

> Uso di `recordsScanned` e `bytesScanned`. Il volume di scansione determina il costo e il tempo di esecuzione di una query. Per ridurlo, riduci l'intervallo di tempo, interroga un numero inferiore di gruppi di log o crea indici di campo. Per ulteriori informazioni, consulta Crea indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione.

Quanti eventi di registro corrispondono a questa query?

> Utilizza `recordsMatched`.

> **Note**
>
> CloudWatch Logs omette una statistica dall'`statistics`oggetto quando non è disponibile alcun valore, anziché riportare la statistica come 0.
> Se il consumatore dell'evento non trova `resultCount` un evento, considera il valore come sconosciuto anziché come 0. Scrivi ai consumatori degli eventi in modo che tollerino le statistiche assenti e ignorino le statistiche che non riconoscono.

Riferimento alle espressioni di pianificazione

Usa queste tabelle di riferimento per creare espressioni di pianificazione per le tue interrogazioni pianificate. Tutti gli orari sono in UTC.

Sintassi delle espressioni Cron

Formato: `cron(minute hour day-of-month month day-of-week year)`

Caso d'uso	Espressione cron	Description	Usa quando
Pianificazioni giornaliere	`cron(0 9 * * ? *)`	Ogni giorno alle 9:00 UTC	Rapporti giornalieri
	`cron(0 */6 * * ? *)`	Ogni 6 ore (00:00, 06:00, 12:00, 18:00 UTC)	Monitoraggio frequente
	`cron(30 2 * * ? *)`	Ogni giorno alle 2:30 UTC	Off-peak analisi
Orario lavorativo	`cron(0 9-17 ? * MON-FRI *)`	Ogni ora dalle 9:00 alle 17:00, dal lunedì al venerdì UTC	Monitoraggio aziendale
	`cron(0 18 ? * MON-FRI *)`	Giorni feriali alle 18:00 UTC	Fine della giornata lavorativa

Caso d'uso	Espressione cron	Description	Usa quando
	`cron(0 8,12,17 ? * MON-FRI *)`	8:00, mezzogiorno e 17:00 nei giorni feriali UTC	Orari lavorativi principali
Pianificazioni settimanali	`cron(0 12 ? * SUN *)`	Ogni domenica a mezzogiorno UTC	Riepiloghi settimanali
	`cron(0 9 ? * MON *)`	Ogni lunedì alle 9:00 UTC	Rapporti sull'inizio della settimana
	`cron(0 23 ? * FRI *)`	Ogni venerdì alle 23:00 UTC	Pulizia del fine settimana
Pianificazioni mensili	`cron(0 0 1 * ? *)`	Il primo giorno di ogni mese a mezzanotte UTC	Rapporti mensili
	`cron(0 9 L * ? *)`	L'ultimo giorno di ogni mese alle 9:00 UTC	Elaborazione di fine mese
	`cron(0 10 1 1,4,7,10 ? *)`	Primo giorno di ogni trimestre alle 10:00 UTC	Analisi trimestrale
Alta frequenza	`cron(*/15 * * * ? *)`	Ogni 15 minuti	Real-time monitoraggio
	`cron(0,30 * * * ? *)`	Ogni 30 minuti (alle :00 e alle :30)	Controlli frequenti
	`cron(0 */2 * * ? *)`	Ogni 2 ore	Intervalli regolari

Caso d'uso	Espressione cron	Description	Usa quando
Casi speciali	`cron(30 8 1 1 ? *)`	1 gennaio alle 8:30 UTC	Rapporti annuali
	`cron(0 6 * * SAT,SUN *)`	Fine settimana alle 6:00 UTC	Elaborazione nel weekend
	`cron(0 0 ? * MON#1 *)`	Primo lunedì di ogni mese a mezzanotte UTC	Pianificazione mensile

Riferimento al campo di espressione Cron

Campo	Valori	Caratteri jolly	Esempi
Minuto (1°)	0-59	`* , - /`	`0`(massimo 1 ora), `*/15` (ogni 15 minuti), `0,30` (due volte all'ora)
Ora (seconda)	0-23	`* , - /`	9(9:00), `*/2` (ogni 2 ore), `9-17` (orario lavorativo)
Day-of-month (3°)	1-31, L, W	`* , - / ?`	1(1° giorno), `L` (ultimo giorno), `?` (se si utilizza il giorno della settimana)
Mese (4°)	1-12 o JAN-DEC	`* , - /`	1(gennaio)`JAN`, `1,4,7,10` (trimestrale)
Day-of-week (5°)	1-7 o SUN-SAT	`* , - / ? # L`	`MON-FRI`(giorni feriali)`SUN`, `MON#1` (1° lunedì)
Anno (6°)	1970-2199	`* , - /`	`*`(ogni anno), `2024` (anno specifico), `2024-2026` (intervallo)

Caratteri jolly ed espressioni speciali

`*` (asterisco)

Corrisponde a tutti i valori nel campo. Esempio: `*` nel campo delle ore si intende ogni ora.

`?`(punto interrogativo)

Nessun valore specifico. Da utilizzare nel giorno del mese o nel giorno della settimana quando viene specificato l'altro. Esempio: utilizzare il giorno del mese quando si specifica il giorno `?` della settimana. `MON-FRI`

`-` (trattino)

Intervallo di valori. Esempio: `MON-FRI` (dal lunedì al venerdì), `9-17` (dalle 9:00 alle 17:00).

`,`(virgola)

Valori specifici multipli. Esempio: `MON,WED,FRI` (lunedì, mercoledì, venerdì), `8,12,17` (8:00, mezzogiorno, 17:00).

`/`(barra)

Valori o incrementi dei passaggi. Esempio: `0/15` in minuti significa ogni 15 minuti a partire dal minuto 0 (0, 15, 30, 45). `*/2`in ore significa ogni 2 ore.

`L`(ultimo)

Ultimo giorno del mese o ultima ricorrenza del giorno feriale. Esempio: `L` in giorno del mese si intende l'ultimo giorno del mese. `FRIL`significa ultimo venerdì del mese.

`W`(giorno della settimana)

Giorno feriale più vicino. Esempio: `15W` indica il giorno della settimana più vicino al 15 del mese.

`#`(ennesima ricorrenza)

Ennesima ricorrenza di un giorno della settimana in un mese. Esempio: `MON#1` indica il primo lunedì del mese, `FRI#2` significa il secondo venerdì del mese.

Schemi e best practice comuni

- Per applicazioni aziendali: utilizzo `MON-FRI` e orario lavorativo (ad esempio`9-17`) per evitare di eseguire domande durante i fine settimana o fuori orario.

- Per il monitoraggio ad alta frequenza: utilizza incrementi come `*/15` (ogni 15 minuti) ma fai attenzione ai limiti di concorrenza delle query.
- Per l'efficienza delle risorse: pianifica le query che richiedono molte risorse durante le ore non di punta utilizzando le ore del mattino presto, come l'UTC. `2-6`
- Per i report mensili: utilizzali per l'ultimo giorno del mese o `L` per date specifiche, ad esempio il primo giorno, `1` per garantire una tempistica costante.

Best practice

Segui queste best practice per garantire operazioni di interrogazione pianificate affidabili ed efficienti:

Ottimizzazione delle query

- Testa le query manualmente prima della pianificazione per verificare prestazioni e risultati
- Utilizza gli indici di filtro nelle prime fasi della query per ridurre l'elaborazione dei dati
- Limita gli intervalli di tempo per evitare timeout con gruppi di log ad alto volume
- Considera la complessità delle query e i limiti di tempo di esecuzione

Pianificazione della pianificazione

- Evita le esecuzioni sovrapposte assicurando che le query vengano completate prima della successiva esecuzione pianificata
- Prendi in considerazione i ritardi nell'inserimento dei log quando imposti gli intervalli di tempo
- Usa le espressioni cron per orari specifici
- Distribuisci le pianificazioni per assicurarti di non raggiungere il limite di concorrenza delle query

Monitoraggio e manutenzione

- Monitora regolarmente la cronologia delle esecuzioni per identificare guasti o problemi di prestazioni
- Rivedi e aggiorna periodicamente i ruoli IAM per mantenere la sicurezza
- Verifica l'accessibilità della destinazione prima dell'implementazione in produzione

Autorizzazione

- Tutte le API per le interrogazioni pianificate vengono autorizzate sulla risorsa di query pianificata e non sulle risorse utilizzate nell'input, ad esempio i gruppi di log. Configura le policy IAM di conseguenza
- Gestisci l'autorizzazione dei gruppi di log utilizzando il ruolo di esecuzione passato nelle API

Guida introduttiva alle query pianificate

Quando crei una query pianificata, configurerai diversi componenti chiave che definiscono come viene eseguita la query e dove vengono forniti i risultati. La comprensione di questi componenti ti aiuterà a configurare un'efficace analisi automatizzata dei log.

Ogni interrogazione pianificata è costituita dai seguenti componenti chiave:

Configurazione dell'interrogazione

La stringa di query di CloudWatch Logs Insights, i gruppi di log di destinazione e il linguaggio di interrogazione da utilizzare per l'analisi.

Pianifica espressione

Un'espressione cron o un calendario di frequenza che definisce quando viene eseguita la query. È possibile specificare le impostazioni del fuso orario per garantire che le query vengano eseguite all'ora locale corretta. La console mostra una descrizione leggibile della pianificazione, ad esempio «Esegui le query ogni martedì alle 15:10 per un intervallo di tempo di 5 minuti, con effetto immediato, in UTC, fino a tempo indeterminato».

Intervallo temporale

Il periodo di lookback per ogni esecuzione della query, definito da una differenza tra l'ora di inizio e l'ora di esecuzione. Questo determina la quantità di dati storici che ogni esecuzione di query analizzerà.

Anteprima della pianificazione di esecuzione

La console mostra le successive tre esecuzioni di query pianificate con date e ore esatte (ad esempio, 2025/10 /28 15:10, UTC; 2025/11 /04 15:10, UTC; 2025/11 /11 15:10, UTC), aiutandoti a verificare che la pianificazione sia configurata correttamente.

Destinazioni

Dove vengono consegnati i risultati della query dopo l'esecuzione corretta. Le destinazioni supportate includono i bucket Amazon S3 e, per impostazione predefinita, i metadati dei risultati vengono inviati al bus di eventi predefinito.

Ruolo di esecuzione

Un ruolo IAM che CloudWatch Logs assume per eseguire la query e fornire risultati alle destinazioni specificate.

Prima di creare query pianificate, assicurati di avere configurato le autorizzazioni e le risorse necessarie.

Creazione di un'interrogazione pianificata

Crea una query pianificata che esegua automaticamente le query di CloudWatch Logs Insights e fornisca risultati alle destinazioni scelte.

Prerequisiti

Prima di creare un'interrogazione pianificata, assicurati di avere quanto segue:

- Gruppi di log: uno o più gruppi di log contenenti i dati da analizzare
- Ruolo IAM di esecuzione: un ruolo IAM con le seguenti autorizzazioni:
  - `logs:StartQuery`- Autorizzazione ad avviare le query di CloudWatch Logs Insights
  - `logs:GetQueryResults`- Autorizzazione a recuperare i risultati delle query
  - `logs:DescribeLogGroups`- Autorizzazione ad accedere alle informazioni sul gruppo di log. Questa opzione è richiesta solo per i gruppi di log basati su prefissi per l'individuazione dei gruppi di log
- Autorizzazioni di destinazione: autorizzazioni IAM aggiuntive per la destinazione scelta:
  - Per le destinazioni Amazon S3: `s3:PutObject`
- Per AWS CLI e utilizzo delle API: AWS credenziali configurate con autorizzazioni per chiamare le API Logs CloudWatch

Per esempi dettagliati di policy IAM, consulta. Gestione delle identità e degli accessi per Amazon CloudWatch Logs Da notare inoltre che puoi avere solo 1000 query pianificate per account.

Console

Per creare un'interrogazione pianificata (console)

1. Aprire la console CloudWatch Logs all'indirizzo https://us-east-1.console.aws.amazon.com/cloudwatch/home? region=us-east-1 #logs. V2:logs-insights
2. Nel pannello di navigazione, scegli Logs Insights.
3. Scegli Crea interrogazione pianificata.
4. Nella sezione Definizione dell'interrogazione:

a. Per il linguaggio di interrogazione, scegli il linguaggio di interrogazione da utilizzare dall'elenco.

b. Per Query string, inserisci la tua query CloudWatch Logs Insights nella casella.

c. Per i gruppi di log, seleziona dall'elenco i gruppi di log da interrogare.

5. Nella sezione Configurazione della pianificazione:

a. Per l'espressione Schedule, configura quando viene eseguita la query. Scegli tra le opzioni predefinite o inserisci un'espressione cron personalizzata.

b. Per Efficace al momento della creazione, specifica quando la pianificazione diventa attiva. Scegli di iniziare immediatamente o a una data e ora specifiche utilizzando il YYYY/MM/DD formato.

c. Per Intervallo di tempo, specifica il periodo di lookback per ogni esecuzione della query. Immettete la durata in minuti che definisce quanto manca al tempo di esecuzione dell'interrogazione.

d. Per Continua all'infinito, specifica quando termina la pianificazione. Scegli di eseguirlo a tempo indeterminato o fino a una data e ora specifiche utilizzando YYYY/MM/DD il formato.

6. La console mostra le successive tre esecuzioni di query pianificate in base alla configurazione, mostrando le date e le ore esatte in UTC in cui verrà eseguita la query.

7. Nella sezione Invia i risultati della query a S3 - opzionale (se si utilizza la destinazione S3):

a. Per il bucket S3, seleziona Questo account se il bucket di destinazione si trova nello stesso AWS account oppure seleziona Altro account se il bucket si trova in un account diverso e fornisci l'ID AWS account dell'account proprietario del bucket come input.

b. Per l'URI Amazon S3, inserisci il bucket e il prefisso Amazon S3 in cui verranno archiviati i risultati (ad esempio,). `s3://my-bucket/query-results/` Se hai selezionato Questo account, puoi scegliere Sfoglia Amazon S3 per navigare e selezionare una posizione Amazon S3 esistente.

c. (Facoltativo) Per la chiave KMS ARN, inserisci l'ARN di una AWS KMS chiave gestita dal cliente per crittografare i risultati della query utilizzando. SSE-KMS La chiave deve trovarsi nella stessa AWS regione del bucket Amazon S3 di destinazione.

8. Nella sezione dedicata al ruolo IAM per la pubblicazione dei risultati delle query su Amazon S3, scegli una delle seguenti opzioni:

a. Scegli Auto-create un nuovo ruolo con autorizzazioni predefinite per configurare automaticamente un ruolo IAM con le autorizzazioni necessarie a CloudWatch Logs per fornire i risultati delle query ad Amazon S3.

b. Scegli Usa un ruolo esistente per selezionare un ruolo IAM esistente con le policy richieste per i CloudWatch log per fornire i risultati delle query ad Amazon S3. Utilizza il campo di ricerca per trovare e selezionare il ruolo IAM appropriato dall'elenco.

> **Note**
>
> Se il role manager è abilitato nel tuo account CloudWatch, assegna il ruolo per te e le opzioni di ruolo qui descritte (ad esempio il pulsante Crea nuovo ruolo) vengono sostituite da un'opzione Personalizza. Per utilizzare un ruolo diverso, scegli Personalizza. Per ulteriori informazioni sulla creazione del ruolo IAM, consulta la creazione del ruolo IAM nella IAM User Guide.

9. Nella sezione Ruolo IAM per l'esecuzione pianificata delle query, scegli una delle seguenti opzioni:

a. Scegli Auto-create un nuovo ruolo con autorizzazioni predefinite per configurare automaticamente un ruolo IAM con le autorizzazioni richieste ai CloudWatch log per eseguire le query pianificate.

b. Scegli Usa un ruolo esistente per selezionare un ruolo IAM esistente con le politiche richieste affinché i CloudWatch log eseguano le query pianificate. Usa il campo di ricerca per trovare e selezionare il ruolo IAM appropriato dall'elenco.

> **Note**
>
> Se il role manager è abilitato nel tuo account CloudWatch, assegna il ruolo per te e le opzioni di ruolo qui descritte (ad esempio il pulsante Crea nuovo ruolo) vengono sostituite da un'opzione Personalizza. Per utilizzare un ruolo diverso, scegli Personalizza. Per ulteriori informazioni sulla creazione del ruolo IAM, consulta la creazione del ruolo IAM nella IAM User Guide.

10. Scegli Crea pianificazione per creare l'interrogazione pianificata.

AWS CLI

Per creare un'interrogazione pianificata (AWS CLI)

- Usa il `create-scheduled-query` comando per creare una nuova interrogazione pianificata:

```
aws logs create-scheduled-query \
    --name "ErrorAnalysisQuery" \
    --query-language "CWLI" \
    --query-string "fields @timestamp, @message | filter @message like /ERROR/ |
 stats count() by bin(5m)" \
    --schedule-expression "cron(8 * * * ? *)" \
    --execution-role-arn "arn:aws:iam::123456789012:role/
CloudWatchLogsScheduledQueryRole" \
    --log-group-identifiers "/aws/lambda/my-function" "/aws/apigateway/my-api" \
    --state "ENABLED"
```

API

Per creare una query pianificata (API)

- Usa l'`CreateScheduledQuery`azione per creare una nuova query pianificata. L'esempio seguente crea una query pianificata che viene eseguita ogni ora:

```
{
    "name": "ErrorAnalysisQuery",
    "queryLanguage": "CWLI",
    "queryString": "fields @timestamp, @message | filter @message like /ERROR/ |
 stats count() by bin(5m)",
    "scheduleExpression": "cron(8 * * * ? *)",
    "executionRoleArn": "arn:aws:iam::123456789012:role/
CloudWatchLogsScheduledQueryRole",
    "logGroupIdentifiers": ["/aws/lambda/my-function", "/aws/apigateway/my-
api"],
    "state": "ENABLED"
}
```

Dopo aver creato la query pianificata, è possibile visualizzarla e gestirla dalla pagina Query pianificate e utilizzando l' ListScheduledQueries API, che mostra tutte le query pianificate con i

relativi nomi, date di creazione, stato dell'ultima esecuzione, ora dell'ultima attivazione e frequenza di ripetizione.

Visualizzazione e gestione delle interrogazioni pianificate

Per ogni interrogazione sono disponibili le seguenti informazioni:

Nome

Il nome univoco assegnato all'interrogazione pianificata. Seleziona il nome per visualizzare la cronologia dettagliata della configurazione e dell'esecuzione.

Data di creazione

La data di creazione dell'interrogazione pianificata, visualizzata in YYYY-MM-DD formato.

Stato dell'ultima esecuzione

Lo stato di esecuzione dell'ultima query eseguita. I valori possibili includono:

- Completa: l'interrogazione è stata eseguita correttamente e i risultati sono stati consegnati a tutte le destinazioni configurate.
- Fallito: l'esecuzione della query o la consegna dei risultati non sono riuscite. Controlla la cronologia delle esecuzioni per i dettagli degli errori.
- Interrogazione non valida: la query non è valida e presenta problemi sintattici
- Timeout: la query è scaduta. Una query scade automaticamente dopo 60 minuti

Ultima ora di attivazione

La data e l'ora dell'ultima esecuzione dell'interrogazione, visualizzate nel YYYY-MM-DD HH:MM:SS formato. Mostra Mai se l'interrogazione non è ancora stata eseguita.

Ripetendo ogni

La frequenza di pianificazione dell'interrogazione. Mostra Personalizzato per le query che utilizzano espressioni cron o descrizioni di frequenza specifiche per pianificazioni più semplici.

La pagina Query pianificate fornisce una panoramica di tutte le query pianificate, mostrandone lo stato attuale e la cronologia di esecuzione in modo da poter visualizzare, monitorare e gestire tutte le query pianificate da una posizione centralizzata. Utilizza queste informazioni per monitorare le prestazioni delle query, identificare i problemi e gestire i flussi di lavoro automatici di analisi dei log.

Console

Per visualizzare le interrogazioni pianificate (console)

1. Aprire la console CloudWatch Logs all'indirizzo https://us-east-1.console.aws.amazon.com/cloudwatch/home? region=us-east-1 #logs. V2:logs-insights
2. Nella console CloudWatch Logs, scegli Interrogazione pianificata, Visualizza interrogazioni pianificate.

AWS CLI

Per elencare le interrogazioni pianificate (AWS CLI)

- Usa il `list-scheduled-queries` comando per elencare tutte le interrogazioni pianificate:

```
aws logs list-scheduled-queries --max-results 10
```

API

Per elencare le interrogazioni pianificate (API)

- Usa l'`ListScheduledQueries`azione per recuperare tutte le query pianificate:

```
{
    "maxResults": 10
}
```

L'intestazione della pagina Query pianificate mostra il numero totale di query pianificate nel tuo account, aiutandoti a monitorare l'utilizzo e a gestire efficacemente i flussi di lavoro di analisi automatizzata dei log.

Visualizzazione della cronologia delle esecuzioni pianificate delle query

Utilizza la cronologia delle esecuzioni per monitorare le prestazioni delle query pianificate e risolvere eventuali problemi relativi all'esecuzione delle query o alla consegna dei risultati.

La cronologia delle esecuzioni mostra lo stato di ogni esecuzione di query, comprese le esecuzioni riuscite, gli errori e i risultati dell'elaborazione di destinazione. È possibile utilizzare queste

informazioni per identificare modelli, diagnosticare problemi e verificare che le query vengano eseguite come previsto.

Console

Per visualizzare la cronologia delle esecuzioni (console)

1. Nella console CloudWatch Logs, scegli Interrogazione pianificata, Visualizza interrogazioni pianificate.
2. Seleziona l'interrogazione pianificata che desideri esaminare.
3. Scegliere la scheda Cronologia delle esecuzioni.

AWS CLI

Per visualizzare la cronologia delle esecuzioni (AWS CLI)

1. Usa il `get-scheduled-query-history` comando per recuperare la cronologia delle esecuzioni di una query pianificata:

```
aws logs get-scheduled-query-history \
    --identifier "DailyErrorMonitoring" \
    --start-time 1743379200 \
    --end-time 1743465600 \
    --max-results 10
```

2. Per filtrare in base allo stato di esecuzione, aggiungi il `--execution-statuses` parametro:

```
aws logs get-scheduled-query-history \
    --identifier "DailyErrorMonitoring" \
    --start-time 1743379200 \
    --end-time 1743465600 \
    --max-results 1 \
    --execution-statuses "SUCCEEDED"
```

API

Per visualizzare la cronologia delle esecuzioni (API)

- Usa l'`GetScheduledQueryHistory`azione per recuperare la cronologia delle esecuzioni:

```
{
    "identifier": "DailyErrorMonitoring",
    "startTime": 1743379200,
    "endTime": 1743465600,
    "maxResults": 10,
    "executionStatuses": ["SUCCEEDED", "FAILED"]
}
```

La cronologia delle esecuzioni mostra:

- Stato di esecuzione: In esecuzione, Completata, Non riuscita, Timeout o InvalidQuery
- Ora di attivazione: quando la query è stata eseguita
- Destinazioni: stato di elaborazione per ogni destinazione configurata EventBridge, inclusi S3 e tabelle di ricerca
- Messaggi di errore: dettagli su eventuali errori nell'esecuzione delle query o nell'elaborazione delle destinazioni

Aggiornamento di una query pianificata

Modifica la configurazione delle query pianificate per modificare la stringa di query, la pianificazione, le destinazioni o il ruolo di esecuzione in base all'evoluzione dei requisiti.

È possibile aggiornare qualsiasi aspetto di un'interrogazione pianificata, tra cui la stringa di query, l'espressione di pianificazione, le destinazioni e il ruolo di esecuzione. Le modifiche hanno effetto immediato per le esecuzioni future.

Console

Per aggiornare una query pianificata (console)

1. Nella console CloudWatch Logs, scegli Interrogazione pianificata, Visualizza interrogazioni pianificate.
2. Seleziona l'interrogazione pianificata che desideri aggiornare.
3. Scegli Modifica.
4. Modificare la configurazione in base alle esigenze.
5. Scegli Save changes (Salva modifiche).

AWS CLI

Per aggiornare un'interrogazione pianificata (AWS CLI)

- Usa il `update-scheduled-query` comando per modificare un'interrogazione pianificata esistente:

```
aws logs update-scheduled-query \
    --identifier "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f" \
    --description "Monitor for ERROR level logs daily" \
    --query-language "LogsQL" \
    --query-string "fields @timestamp, @message | filter @message like /ERROR/"
 \
    --log-group-identifiers "/aws/lambda/my-function-1" "/aws/lambda/my-
function-2"
```

API

Per aggiornare un'interrogazione pianificata (API)

1. Usa l'`UpdateScheduledQuery`azione per modificare la configurazione delle query pianificate:

```
{
    "identifier": "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f",
    "queryString": "fields @timestamp, @message | filter @message like /WARNING|
ERROR/ | stats count() by bin(5m)",
    "scheduleExpression": "cron(0 */2 * * ? *)",
    "state": "ENABLED"
}
```

2. Per aggiornare più parametri di configurazione contemporaneamente:

```
{
    "identifier": "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f",
    "queryString": "fields @timestamp, @message, @level | filter @level =
 'ERROR'",
    "scheduleExpression": "cron(0 8,12,16 * * ? *)",
```

```
    "executionRoleArn": "arn:aws:iam::111122223333:role/
UpdatedScheduledQueryRole",
    "logGroupIdentifiers": ["/aws/lambda/my-function", "/aws/lambda/another-
function"],
    "destinationConfiguration": {
        "s3Configuration": {
            "destinationIdentifier": "s3://111122223333-sqn-results-bucket/
processed-results",
            "roleArn": "arn:aws:iam::111122223333:role/Admin"
        }
    }
}
```

Configurazione delle destinazioni S3 per le interrogazioni pianificate

Configura Amazon S3 come destinazione per archiviare i risultati delle query pianificate come file JSON per la conservazione e l'analisi a lungo termine.

Quando si utilizza Amazon S3 come destinazione, i risultati delle query vengono archiviati come file JSON nel bucket e nel prefisso specificati. Questa opzione è ideale per l'archiviazione dei risultati, l'esecuzione di analisi in batch o l'integrazione con altri AWS servizi che elaborano i dati S3.

Puoi fornire i risultati delle query a un bucket Amazon S3 nello stesso AWS account della query pianificata o a un bucket in un account diverso. AWS È inoltre possibile crittografare i risultati delle query utilizzando una chiave gestita dal cliente (). AWS KMS SSE-KMS

Fornire risultati a un bucket Amazon S3 nello stesso account

Quando il bucket Amazon S3 di destinazione si trova nello stesso AWS account della query pianificata, puoi sfogliare e selezionare il bucket direttamente dalla console.

Per configurare una destinazione Amazon S3 (console) con lo stesso account

1. Nella sezione Invia i risultati della query a S3, per il bucket S3, seleziona Questo account.
2. Per l'URI Amazon S3, inserisci il bucket e il prefisso Amazon S3 in cui verranno archiviati i risultati (ad esempio`s3://my-bucket/query-results/`) oppure scegli Browse Amazon S3 per navigare e selezionare una posizione Amazon S3 esistente.
3. (Facoltativo) Per crittografare i risultati con una AWS KMS chiave gestita dal cliente, inserisci l'ARN della chiave nel campo ARN della AWS KMS chiave KMS. La chiave deve trovarsi nella

stessa AWS regione del bucket Amazon S3 di destinazione. Se non si specifica una AWS KMS chiave, si applicano le impostazioni di crittografia predefinite del bucket.

4. Nella sezione Ruolo IAM per la pubblicazione dei risultati delle query su Amazon S3, scegli Auto-create un nuovo ruolo con autorizzazioni predefinite per configurare automaticamente le autorizzazioni richieste oppure scegli Usa un ruolo esistente per selezionare un ruolo IAM esistente con le politiche richieste.

Il ruolo IAM di destinazione per la consegna richiede le seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Fornitura dei risultati a un bucket Amazon S3 in un altro account

Puoi inviare i risultati delle query pianificate a un bucket Amazon S3 in un account diverso. AWS Quando si utilizza un bucket con più account, è necessario fornire l'URI Amazon S3 e l'ID dell'account proprietario del bucket.

Per configurare una destinazione Amazon S3 con più account (console)

1. Nella sezione Invia i risultati della query a S3, per il bucket S3, seleziona Un altro account e fornisci come input l'ID dell'account proprietario del bucket.
2. Per l'URI Amazon S3, inserisci l'URI Amazon S3 completo del bucket di destinazione e il prefisso nell'altro account (ad esempio,). `s3://cross-account-bucket/query-results/`

3. (Facoltativo) Per crittografare i risultati con una AWS KMS chiave gestita dal cliente, inserisci l'ARN della AWS KMS chiave nel campo KMS key ARN. La chiave deve trovarsi nella stessa AWS regione del bucket Amazon S3 di destinazione.
4. Nella sezione Ruolo IAM per la pubblicazione dei risultati delle query su Amazon S3, scegli Auto-create un nuovo ruolo con autorizzazioni predefinite per configurare automaticamente le autorizzazioni richieste oppure scegli Usa un ruolo esistente per selezionare un ruolo IAM esistente con le politiche richieste.

Cross-account la consegna richiede autorizzazioni su entrambi i lati. Il ruolo IAM di destinazione per la consegna nell'account di origine richiede le seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::cross-account-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

La policy del bucket Amazon S3 nell'account di destinazione deve concedere al ruolo IAM dell'account di origine l'autorizzazione a scrivere oggetti:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowScheduledQueryRolePutObject",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/my-s3-delivery-role"
            },
            "Action": "s3:PutObject",
```

```
            "Resource": "arn:aws:s3:::cross-account-bucket/prefix/*"
        }
    ]
}
```

Crittografia dei risultati con un cliente gestito AWS KMS Chiave

Facoltativamente, puoi specificare una AWS KMS chiave gestita dal cliente per crittografare i risultati delle query consegnati ad Amazon S3 utilizzando. SSE-KMS La AWS KMS chiave può trovarsi nello stesso account della query pianificata o in un account diverso.

Quando si specifica una AWS KMS chiave, l'interrogazione pianificata utilizza tale chiave per crittografare i risultati tramite SSE-KMS. Quando non si specifica una AWS KMS chiave, si applicano le impostazioni di crittografia predefinite del bucket. Se il bucket è configurato con la SSE-KMS crittografia predefinita utilizzando una chiave gestita dal cliente, il ruolo IAM di destinazione deve ancora disporre `kms:GenerateDataKey` dell'autorizzazione per tale chiave.

Il ruolo IAM di destinazione per la consegna richiede `kms:GenerateDataKey` l'autorizzazione sulla AWS KMS chiave. L'esempio seguente mostra le autorizzazioni richieste per una destinazione Amazon S3 con una chiave gestita AWS KMS dal cliente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "kms:GenerateDataKey",
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "s3.us-east-1.amazonaws.com"
                },
```

```
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::my-bucket*"
                }
            }
        }
    ]
}
```

Quando la AWS KMS chiave si trova in un account diverso dal ruolo IAM di destinazione per la consegna, la policy AWS KMS chiave dell'account proprietario della chiave deve concedere esplicitamente l'accesso al ruolo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowScheduledQueryRoleToEncrypt",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/my-s3-delivery-role"
            },
            "Action": "kms:GenerateDataKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "s3.us-east-1.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::my-bucket*"
                }
            }
        }
    ]
}
```

> **Note**
>
> Quando la AWS KMS chiave e il ruolo IAM di consegna di destinazione si trovano nello stesso account, la sola politica di identità IAM è sufficiente se la policy AWS KMS chiave include l'istruzione principale predefinita «Abilita politiche IAM». Una concessione esplicita della policy AWS KMS chiave è richiesta solo se la policy chiave non viene delegata a IAM.

Il ruolo IAM per la pubblicazione dei risultati delle query su Amazon S3 deve essere configurato separatamente dal ruolo IAM per l'esecuzione pianificata delle query. Questa separazione consente un controllo granulare degli accessi, in cui il ruolo di esecuzione può eseguire le query mentre il ruolo Amazon S3 gestisce specificamente la consegna dei risultati. Entrambi i ruoli devono includere una policy di fiducia che consenta al servizio CloudWatch Logs (`logs.amazonaws.com`) di assumere il ruolo.

Configurazione delle destinazioni delle tabelle di ricerca per le interrogazioni pianificate

Configura una tabella di ricerca come destinazione per aggiornare automaticamente una tabella di ricerca con i risultati delle query pianificate. Ad ogni esecuzione pianificata, CloudWatch Logs compila la tabella di ricerca specificata con i risultati della query, mantenendo la tabella aggiornata con i dati di registro più recenti. È quindi possibile fare riferimento alla tabella con il `lookup` comando in altre query di registro. Per ulteriori informazioni sulle tabelle di ricerca, vedere. cercare

Se la tabella di ricerca non esiste, CloudWatch Logs la crea alla prima esecuzione riuscita. Ogni esecuzione successiva è un'operazione di sostituzione completa: tutto il contenuto della tabella esistente viene sostituito con i nuovi risultati della query.

La configurazione della tabella di ricerca include le seguenti impostazioni:

- Nome della tabella: il nome della tabella di ricerca da creare o aggiornare. Il nome può contenere solo caratteri alfanumerici e caratteri di sottolineatura.
- Ruolo di consegna: l'ARN del ruolo IAM che CloudWatch concede a Logs le autorizzazioni per creare o aggiornare la tabella di ricerca con i risultati delle query.
- Descrizione: (Facoltativo) Una descrizione della tabella di ricerca.
- AWS KMS key — (Facoltativo) L'ARN della AWS KMS chiave da utilizzare per crittografare i dati della tabella di ricerca. Se non si specifica una chiave, i dati vengono crittografati con una AWS chiave di proprietà.
- Tag: Key-value coppie (facoltative) da associare alla tabella di ricerca. I tag vengono applicati solo quando la tabella viene creata inizialmente.

Il ruolo IAM di destinazione per la consegna richiede le seguenti autorizzazioni e deve includere una policy di fiducia che consenta al servizio CloudWatch Logs (`logs.amazonaws.com`) di assumere il ruolo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLookupTable",
                "logs:UpdateLookupTable",
                "logs:GetQueryResults"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> I risultati di destinazione delle tabelle di ricerca sono soggetti alle stesse quote delle tabelle di ricerca create direttamente, ad esempio il numero massimo di tabelle di ricerca per account per. Regione AWS Se un'esecuzione pianificata non riesce ad aggiornare la tabella, il contenuto della tabella esistente rimane invariato e l'errore viene visualizzato nella cronologia delle esecuzioni delle query pianificate.

Risoluzione dei problemi relativi alle interrogazioni pianificate

Utilizza questi argomenti per la risoluzione dei problemi più comuni relativi alle interrogazioni pianificate.

L'esecuzione della query non riesce a causa di errori di autorizzazione

Risolvi gli errori di autorizzazione che impediscono l'esecuzione o la consegna dei risultati alle destinazioni da parte delle query pianificate.

Gli errori di autorizzazione si verificano quando il ruolo di esecuzione non dispone delle autorizzazioni necessarie per leggere dai gruppi di log o scrivere nelle risorse di destinazione.

Per risolvere gli errori di autorizzazione

1. Verifica che il ruolo di esecuzione `logs:StartQuery` disponga delle autorizzazioni e `logs:DescribeLogGroups` delle autorizzazioni per i gruppi di log di destinazione. `logs:GetQueryResults`
2. Assicurati che il ruolo di esecuzione disponga delle autorizzazioni di scrittura per le risorse di destinazione (ad esempio `s3:PutObject` per i bucket S3).
3. Verifica che la politica di fiducia consenta a CloudWatch Logs di assumere il ruolo di esecuzione. Il ruolo deve considerare attendibile il servizio logs principal (`logs.amazonaws.com`) nella sua politica di fiducia.

Le cause più comuni includono autorizzazioni IAM mancanti, ARN di risorse errati nella policy o problemi di configurazione della policy di trust.

Per evitare errori di autorizzazione, utilizza il principio del privilegio minimo nella creazione dei ruoli di esecuzione e verifica le autorizzazioni prima di distribuire le query pianificate alla produzione.

Scade il timeout delle query

Risolvi gli errori di timeout che si verificano quando le query pianificate superano il limite massimo di esecuzione.

I timeout delle query si verificano quando la query impiega più di 60 minuti per elaborare l'intervallo di dati specificato, spesso a causa di set di dati di grandi dimensioni o di una logica di interrogazione complessa.

Per risolvere gli errori di timeout

1. Riduci l'intervallo di tempo diminuendo l'offset dell'ora di inizio per elaborare meno dati per esecuzione.
2. Ottimizza la query aggiungendo filtri nelle prime fasi della query per ridurre la quantità di dati elaborati. Utilizza gli indici dei filtri per ridurre le dimensioni della scansione dei dati.
3. Prendi in considerazione la possibilità di suddividere le query complesse in domande più semplici e mirate.

Le cause più comuni includono l'interrogazione di ampi intervalli di tempo, l'elaborazione di gruppi di log ad alto volume o l'utilizzo di aggregazioni complesse senza un filtro adeguato.

Per evitare i timeout, testa le query manualmente in CloudWatch Logs Insights con il volume di dati previsto e ottimizza le prestazioni prima della pianificazione.

L'elaborazione della destinazione non riesce

Risolve gli errori che si verificano quando i risultati delle query pianificate non possono essere consegnati alle destinazioni configurate.

Gli errori di elaborazione delle destinazioni si verificano quando il bucket o il bus di EventBridge eventi Amazon S3 di destinazione è inaccessibile o configurato in modo errato.

Per risolvere gli errori in cui i risultati delle query non vengono pubblicati a destinazione

1. Verifica che il bucket Amazon S3 specificato esista e sia accessibile.
2. Controlla la configurazione di destinazione per gli URI corretti.
3. Assicurati che il ruolo di esecuzione disponga delle autorizzazioni necessarie per scrivere nella destinazione.

Le cause più comuni includono risorse di destinazione eliminate o rinominate, URI di destinazione errati o problemi di connettività di rete.

Per evitare errori di destinazione, convalidate regolarmente le configurazioni di destinazione e monitorate la disponibilità delle risorse di destinazione.

Errori di interrogazione non validi

Risolvi gli errori di sintassi e logica nelle stringhe di query pianificate che ne impediscono l'esecuzione corretta.

Gli errori di interrogazione non validi si verificano quando la stringa di query contiene errori di sintassi, fa riferimento a campi inesistenti o utilizza funzionalità del linguaggio di query non supportate.

Per risolvere errori di interrogazione non validi

1. Testa la tua query manualmente in CloudWatch Logs Insights per verificare la sintassi e la logica.
2. Verifica che tutti i campi di log a cui si fa riferimento siano presenti nei gruppi di log di destinazione.

3. Verifica che le funzionalità del linguaggio di interrogazione che stai utilizzando siano supportate per le query pianificate.

Le cause più comuni includono errori di battitura nei nomi dei campi, sintassi errata delle query o utilizzo di funzionalità di interrogazione non supportate nell'ambiente di esecuzione pianificata.

Per evitare errori di interrogazione non validi, testate sempre le interrogazioni in modo interattivo prima di programmarle e utilizzate le funzionalità di individuazione dei campi per verificare i nomi dei campi.

Errori di concorrenza nelle query

Di seguito sono indicati alcuni punti importanti da tenere a mente quando si riscontrano errori di concorrenza, poiché le query pianificate utilizzano la stessa quota delle query di Cloudwatch Logs Insights. Si consiglia di suddividere le pianificazioni per evitare di raggiungere il limite di concorrenza.

- Quota: puoi eseguire fino a 100 query CloudWatch Logs Insights simultanee per account. AWS
- Dashboard: anche le query aggiunte alle CloudWatch dashboard vengono conteggiate ai fini di questo limite di concorrenza, poiché vengono eseguite quando la dashboard viene caricata o aggiornata.
- OpenSearch Servizio PPL/SQL: puoi eseguire fino a 15 query OpenSearch PPL o SQL simultanee per account. OpenSearch AWS
- Cross-account interrogazioni: la quota di concorrenza si applica alle interrogazioni su un solo account e su più account. Quando si utilizza l'osservabilità CloudWatch tra più account, anche le interrogazioni avviate in un account di monitoraggio rispetto a un account di origine collegato vengono conteggiate ai fini del limite di concorrenza dell'account di monitoraggio.
- Gruppi di log di accesso non frequenti: per i gruppi di log della classe di log degli accessi poco frequenti, il numero massimo di query Logs Insights simultanee è limitato a cinque.

Rilevamento delle anomalie nei registri

È possibile rilevare le anomalie nei dati di registro in due modi: creando un rilevatore di anomalie nei log per il monitoraggio continuo o utilizzando il comando nelle query di Logs Insights per l'`anomaly detection`analisi su richiesta. CloudWatch

Un rilevatore di anomalie nel registro analizza gli eventi di registro inseriti in un gruppo di log e trova automaticamente le anomalie nei dati di registro. Il rilevamento delle anomalie utilizza l'apprendimento automatico e il riconoscimento dei pattern per stabilire le linee di base dei contenuti tipici dei log. Per l'analisi su richiesta, puoi utilizzare il `anomaly detection` comando nelle query di CloudWatch Logs Insights per identificare modelli insoliti nei dati delle serie temporali. Per ulteriori informazioni sul rilevamento delle anomalie basato su query, vedere. Utilizzo del rilevamento delle anomalie in Logs Insights CloudWatch

Dopo aver creato un rilevatore di anomalie per un gruppo di log, questo si allena utilizzando gli eventi di registro delle ultime due settimane nel gruppo di log per l'addestramento. Il periodo di formazione può durare fino a 15 minuti. Una volta completato l'addestramento, inizia ad analizzare i log in entrata per identificare le anomalie e le anomalie vengono visualizzate nella console CloudWatch Logs per essere esaminate dall'utente.

CloudWatch Logs Pattern Recognition estrae i pattern di log identificando i contenuti statici e dinamici nei log. I pattern sono utili per analizzare set di log di grandi dimensioni perché un gran numero di eventi di log può spesso essere compresso in pochi pattern.

Ad esempio, vedete il seguente esempio di tre eventi di registro.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for ResourceID: 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for ResourceID: 324892398123-1234R
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for ResourceID: 3ff231242342-12345
```

Nell'esempio precedente, tutti e tre gli eventi di registro seguono uno schema:

```
<Date-1> <Time-2> [INFO] Calling DynamoDB to store for resource id <ResourceID-3>
```

I campi all'interno di un pattern sono chiamati token. I campi che variano all'interno di uno schema, come l'ID della richiesta o il timestamp, vengono definiti token dinamici. Ogni valore diverso trovato per un token dinamico viene chiamato valore token.

Se CloudWatch Logs è in grado di dedurre il tipo di dati rappresentato da un token dinamico, il token viene visualizzato come. <*string*-*number*> *string*È una descrizione del tipo di dati rappresentato dal token. *number*Mostra dove nel modello appare questo token, rispetto agli altri token dinamici.

CloudWatch Logs assegna la parte stringa del nome in base all'analisi del contenuto degli eventi di registro che la contengono.

Se CloudWatch Logs non è in grado di dedurre il tipo di dati rappresentato da un token dinamico, visualizza il token come <Token- *number* > e *number* indica dove nel modello appare questo token rispetto agli altri token dinamici.

Esempi comuni di token dinamici includono codici di errore, indirizzi IP, timestamp e richieste. IDs

Il rilevamento delle anomalie nei log utilizza questi modelli per trovare anomalie. Dopo il periodo di addestramento del modello di rilevatore di anomalie, i log vengono valutati in base alle tendenze note. Il rilevatore di anomalie segnala le fluttuazioni significative come anomalie.

Questo capitolo descrive come abilitare il rilevamento delle anomalie, visualizzare le anomalie, creare allarmi per i rilevatori di anomalie di registro e le metriche pubblicate dai rilevatori di anomalie di registro. Descrive inoltre come crittografare il rilevatore di anomalie e i relativi risultati con. AWS Key Management Service

La creazione di rilevatori di anomalie di registro non comporta costi.

Gravità e priorità delle anomalie e dei modelli

A ogni anomalia rilevata da un rilevatore di anomalie di registro viene assegnata una priorità. A ogni pattern trovato viene assegnata una gravità.

- La priorità viene calcolata automaticamente e si basa sia sul livello di gravità del modello che sulla quantità di deviazione dai valori previsti. Ad esempio, se il valore di un determinato token aumenta improvvisamente del 500%, tale anomalia potrebbe essere designata come `HIGH` prioritaria anche se la sua gravità lo è. `NONE`
- La severità si basa solo sulle parole chiave presenti nei modelli come `FATALERROR`, e. `WARN` Se non viene trovata nessuna di queste parole chiave, la gravità di un pattern viene contrassegnata come`NONE`.

Tempo di visibilità dell'anomalia

Quando si crea un rilevatore di anomalie, si specifica il periodo massimo di visibilità delle anomalie. Questo è il numero di giorni in cui l'anomalia viene visualizzata nella console e viene restituita dall'operazione API. ListAnomalies Trascorso questo periodo di tempo relativo a un'anomalia, se continua a verificarsi, viene automaticamente accettata come comportamento normale e il modello di rilevatore di anomalie smette di contrassegnarla come anomalia.

Se non modifichi il tempo di visibilità quando crei un rilevatore di anomalie, per impostazione predefinita vengono utilizzati 21 giorni.

Soppressione di un'anomalia

Dopo aver rilevato un'anomalia, è possibile scegliere di sopprimerla temporaneamente o permanentemente. La soppressione di un'anomalia fa sì che il rilevatore di anomalie smetta di contrassegnare tale evento come anomalia per il periodo di tempo specificato. Quando si sopprime un'anomalia, è possibile scegliere di sopprimere solo quell'anomalia specifica o di sopprimere tutte le anomalie relative al modello in cui è stata rilevata l'anomalia.

È ancora possibile visualizzare le anomalie soppresse nella console. Puoi anche scegliere di smettere di sopprimerle.

Domande frequenti

AWS Utilizza i miei dati per addestrare algoritmi di apprendimento automatico da AWS utilizzare o per altri clienti?

No. Il modello di rilevamento delle anomalie creato dal corso di formazione si basa sugli eventi di registro in un gruppo di log e viene utilizzato solo all'interno di quel gruppo di log e di quell' AWS account.

Quali tipi di eventi di registro funzionano bene con il rilevamento delle anomalie?

Il rilevamento delle anomalie nei log è ideale per: registri delle applicazioni e altri tipi di log in cui la maggior parte delle voci di registro rientra negli schemi tipici. I gruppi di log con eventi che contengono parole chiave a livello di registro o di gravità come INFO, ERROR e DEBUG sono particolarmente adatti per il rilevamento delle anomalie nei log.

Il rilevamento delle anomalie nei log non è adatto per: Registra eventi con strutture JSON estremamente lunghe, come Logs. CloudTrail L'analisi dei pattern analizza solo fino ai primi 1500 caratteri di una riga di registro, quindi tutti i caratteri oltre tale limite vengono ignorati.

Anche i log di controllo o di accesso, come i log di flusso VPC, avranno meno successo con il rilevamento delle anomalie. Il rilevamento delle anomalie serve a individuare i problemi delle applicazioni, quindi potrebbe non essere adatto per le anomalie di rete o di accesso.

Per aiutarvi a determinare se un rilevatore di anomalie è adatto a un determinato gruppo di log, utilizzate l'analisi del pattern CloudWatch Logs per trovare il numero di pattern negli eventi di registro del gruppo. Se il numero di pattern non è superiore a circa 300, il rilevamento delle anomalie potrebbe funzionare bene. Per ulteriori informazioni sull'analisi dei pattern, vedereAnalisi del modello.

Cosa viene contrassegnato come anomalia?

Le seguenti occorrenze possono far sì che un evento di registro venga contrassegnato come anomalia:

- Un evento di registro con uno schema mai visto prima nel gruppo di log.
- Una variazione significativa rispetto a un modello noto.
- Un nuovo valore per un token dinamico che ha un insieme discreto di valori usuali.
- Una grande variazione nel numero di occorrenze di un valore per un token dinamico.

Sebbene tutti gli elementi precedenti possano essere contrassegnati come anomalie, non tutti significano che l'applicazione stia funzionando male. Ad esempio, higher-than-usual alcuni valori di `200` successo potrebbero essere contrassegnati come anomalia. In casi come questo, potresti prendere in considerazione la possibilità di eliminare queste anomalie che non indicano problemi.

Cosa succede con i dati sensibili che vengono mascherati?

Qualsiasi parte degli eventi di registro mascherata come dati sensibili non viene analizzata per individuare eventuali anomalie. Per ulteriori informazioni sul mascheramento dei dati sensibili, consulta Aiutare a proteggere i dati di registro sensibili con il mascheramento.

Utilizzo del rilevamento delle anomalie in Logs Insights CloudWatch

Oltre a creare rilevatori di anomalie nei log per il monitoraggio continuo, puoi anche utilizzare il `anomaly` comando nelle query di CloudWatch Logs Insights per identificare modelli insoliti nei

dati di registro su richiesta. Questo comando estende le `pattern` funzionalità esistenti e utilizza l'apprendimento automatico per rilevare cinque tipi di anomalie, tra cui modifiche della frequenza dei pattern, nuovi pattern e variazioni dei token.

Il `anomaly` comando è particolarmente utile per:

- Analisi ad hoc dei dati di registro storici per identificare modelli insoliti
- Analisi di periodi di tempo specifici per individuare comportamenti anomali
- Monitoraggio di applicazioni come le funzioni Lambda per problemi di esecuzione

Per ulteriori informazioni sull'utilizzo del `anomaly` comando nelle query, consulta. anomalia

Questo rilevamento di anomalie basato su query integra i rilevatori di anomalie continui descritti nelle sezioni seguenti, offrendo funzionalità di monitoraggio in tempo reale e analisi su richiesta.

Abilita il rilevamento delle anomalie su un gruppo di log

Utilizza i passaggi seguenti per utilizzare la CloudWatch console per creare un rilevatore di anomalie di registro che scansiona un gruppo di log alla ricerca di anomalie.

È inoltre possibile creare rilevatori di anomalie a livello di codice. Per ulteriori informazioni, consulta CreateLogAnomalyDetector.

Per creare un rilevatore di anomalie di registro

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Scegli Logs, Log Anomalies.
3. Scegli Crea rilevatore di anomalie.
4. Seleziona il gruppo di log per cui creare questo rilevatore di anomalie.
5. Immettere un nome per il rilevatore in Nome del rilevatore di anomalie.
6. (Facoltativo) Modifica la frequenza di valutazione dal valore predefinito di 5 minuti. Imposta questo valore in base alla frequenza con cui il gruppo di log riceve nuovi registri. Ad esempio, se il gruppo di log riceve nuovi eventi di registro in batch ogni 10 minuti, potrebbe essere appropriato impostare la frequenza di valutazione su 15 minuti.
7. (Facoltativo) Per configurare il rilevatore di anomalie in modo che cerchi le anomalie solo negli eventi di registro che contengono determinate parole o stringhe, scegli Filter pattern.

Quindi, inserisci uno schema nel pattern di filtro di rilevamento delle anomalie. Per ulteriori informazioni sulla sintassi del pattern,. Sintassi del modello di filtro per filtri di parametri, filtri di sottoscrizione, filtri di log eventi e Live Tail

(Facoltativo) Per testare il modello di filtro, inserisci alcuni messaggi di registro in Log event messages, quindi scegli Test Pattern.

8. (Facoltativo) Per modificare il periodo di visibilità delle anomalie rispetto a quello predefinito o per associare una AWS KMS chiave a questo rilevatore di anomalie, scegli Configurazione avanzata.

   a. Per modificare il periodo di visibilità delle anomalie rispetto a quello predefinito, inserisci un nuovo valore in Periodo massimo di visibilità delle anomalie (giorni).

   b. Per associare una AWS KMS chiave a questo rilevatore di anomalie, inserisci l'ARN nella chiave KMS ARN. Se si assegna una chiave, le informazioni sull'anomalia rilevate da questo rilevatore vengono crittografate quando sono inattive con la chiave. Gli utenti devono disporre delle autorizzazioni per questa chiave e per consentire al rilevatore di anomalie di recuperare le informazioni sulle anomalie rilevate.

      È inoltre necessario assicurarsi che il responsabile del servizio CloudWatch Logs sia autorizzato a utilizzare la chiave. Per ulteriori informazioni, consulta Crittografa un rilevatore di anomalie e i relativi risultati con AWS KMS.

9. Scegli Abilita il rilevamento delle anomalie.

   Il rilevatore di anomalie viene creato e inizia ad addestrare il suo modello, in base agli eventi di registro che il gruppo di log sta acquisendo. Dopo circa 15 minuti, il rilevamento delle anomalie è attivo e inizia a rilevare ed evidenziare le anomalie.

Visualizza le anomalie rilevate

Dopo aver creato uno o più rilevatori di anomalie nei registri, puoi utilizzare la CloudWatch console per visualizzare le anomalie rilevate.

È possibile visualizzare le anomalie a livello di codice. Per ulteriori informazioni, consulta ListAnomalies.

Per visualizzare le anomalie rilevate da tutti i rilevatori di anomalie di registro

1. Apri la console all'indirizzo. CloudWatch https://console.aws.amazon.com/cloudwatch/

2. Scegli Logs, Log Anomalies.

   Viene visualizzata la tabella delle anomalie dei registri. Il numero in alto accanto a Log anomalies mostra quante anomalie di log sono elencate nella tabella. Ogni riga della tabella mostra le seguenti informazioni:

   - La colonna Anomalia mostra un breve riepilogo dell'anomalia. Questi riepiloghi sono generati da Logs. CloudWatch
   - La priorità dell'anomalia. La priorità viene calcolata automaticamente in base alla quantità di modifiche negli eventi di registro, alle parole chiave, ad esempio, `Exception` che si verificano in un evento di registro e altro ancora.
   - Il pattern di log su cui si basa l'anomalia. Per ulteriori informazioni sui modelli, vedereRilevamento delle anomalie nei registri.
   - L'andamento del registro delle anomalie visualizza un istogramma che rappresenta il volume di log che corrispondono al modello.
   - L'ora dell'ultimo rilevamento mostra l'ora più recente in cui è stata rilevata questa anomalia.
   - L'ora del primo rilevamento indica la prima volta in cui è stata rilevata l'anomalia.
   - Il rilevatore di anomalie visualizza il nome del gruppo di log contenente gli eventi di registro relativi a questa anomalia. È possibile scegliere questo nome per visualizzare la pagina dei dettagli del gruppo di log.

3. Per esaminare ulteriormente un'anomalia, scegli il pulsante di opzione nella relativa riga.

   Viene visualizzato il riquadro Pattern inspect che mostra quanto segue:

   - Il pattern su cui si basa questa anomalia. Seleziona un token all'interno del pattern per analizzare i valori di quel token.
   - Un istogramma che mostra il numero di ricorrenze dell'anomalia nell'intervallo di tempo richiesto.
   - La scheda Registra esempi mostra alcuni degli eventi di registro che fanno parte dell'anomalia.
   - La scheda Valori dei token mostra i valori del token dinamico selezionato, se ne è stato selezionato uno.

> **Note**
>
> Per ogni token vengono acquisiti un massimo di 10 valori di token. Il conteggio dei token potrebbe non essere preciso. CloudWatch Logs utilizza un contatore probabilistico per generare il conteggio dei token, non il valore assoluto.

4. Per eliminare un'anomalia, scegli il pulsante di opzione nella relativa riga, quindi procedi come segue:

   a. Scegliete Azioni, Sopprimi l'anomalia.

   b. Quindi specifica per quanto tempo desideri che l'anomalia venga soppressa.

   c. Per sopprimere tutte le anomalie relative a questo pattern, selezionate Sopprimi pattern.

   d. Scegliete Sopprimi l'anomalia.

Per visualizzare le anomalie rilevate in un singolo gruppo di log

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Scegli Log, Gruppi di log.
3. Scegli il nome di un gruppo di log, quindi scegli la scheda Rilevamento delle anomalie.

   Viene visualizzata la tabella di rilevamento delle anomalie. Il numero in alto accanto a Log anomalies mostra quante anomalie di log sono elencate nella tabella. Ogni riga della tabella mostra le seguenti informazioni:

   - La colonna Anomalia mostra un breve riepilogo dell'anomalia. Questi riepiloghi sono generati da Logs. CloudWatch
   - La priorità dell'anomalia. La priorità viene calcolata automaticamente in base alla quantità di modifiche negli eventi di registro, alle parole chiave, ad esempio, `Exception` che si verificano in un evento di registro e altro ancora.
   - Il pattern di log su cui si basa l'anomalia. Per ulteriori informazioni sui modelli, vedereRilevamento delle anomalie nei registri.
   - L'andamento del registro delle anomalie visualizza un istogramma che rappresenta il volume di log che corrispondono al modello.
   - L'ora dell'ultimo rilevamento mostra l'ora più recente in cui è stata rilevata questa anomalia.
   - L'ora del primo rilevamento indica la prima volta in cui è stata rilevata l'anomalia.

4. Per esaminare ulteriormente un'anomalia, scegli il pulsante di opzione nella relativa riga.

   Viene visualizzato il riquadro Pattern inspect che mostra quanto segue:

   - Il pattern su cui si basa questa anomalia. Seleziona un token all'interno del pattern per analizzare i valori di quel token.
   - Un istogramma che mostra il numero di ricorrenze dell'anomalia nell'intervallo di tempo richiesto.
   - La scheda Registra esempi mostra alcuni degli eventi di registro che fanno parte dell'anomalia.
   - La scheda Valori dei token mostra i valori del token dinamico selezionato, se ne è stato selezionato uno.

   > **Note**
   >
   > Per ogni token vengono acquisiti un massimo di 10 valori di token. Il conteggio dei token potrebbe non essere preciso. CloudWatch Logs utilizza un contatore probabilistico per generare il conteggio dei token, non il valore assoluto.

5. Per eliminare un'anomalia, scegli il pulsante di opzione nella relativa riga, quindi procedi come segue:

   a. Scegliete Azioni, Sopprimi l'anomalia.
   b. Quindi specifica per quanto tempo desideri che l'anomalia venga soppressa.
   c. Per sopprimere tutte le anomalie relative a questo pattern, selezionate Sopprimi pattern.
   d. Scegliete Sopprimi l'anomalia.

Crea allarmi sui rilevatori di anomalie di registro

È possibile creare un allarme per un rilevatore di anomalie di registro in un gruppo di log. È possibile specificare che l'allarme entri in `ALARM` stato quando viene rilevato un determinato numero di anomalie nel gruppo di log durante un determinato periodo di tempo. È inoltre possibile utilizzare filtri in modo che nell'allarme vengano conteggiate solo le anomalie con priorità specificate.

Per creare un allarme per un rilevatore di anomalie nel registro

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione, scegli Registri, Registra anomalie.

Viene visualizzata la tabella dei rilevatori di anomalie di registro.

3. Scegli il pulsante radio per il rilevatore di anomalie per cui desideri impostare l'allarme e scegli Crea allarme.

   Viene visualizzata la procedura guidata per la creazione degli CloudWatch allarmi. Il LogAnomalyDetectorcampo mostra il nome del rilevatore di anomalie che hai scelto. Viene visualizzato il campo del nome della metrica. AnomalyCount

4. (Facoltativo) Per filtrare questo allarme in base alla priorità delle anomalie, effettuate una delle seguenti operazioni:

   - Per fare in modo che l'allarme conti solo le anomalie ad alta priorità, inserisci per. **`HIGH`** LogAnomalyPriority
   - Per fare in modo che l'allarme conti solo le anomalie ad alta e media priorità, inserisci per. **`MEDIUM`** LogAnomalyPriority

   Per ulteriori informazioni sui livelli di priorità, vedere. Gravità e priorità delle anomalie e dei modelli

5. Scegli di utilizzare una soglia di rilevamento delle anomalie statica o metrica per l'allarme. Questa selezione determina come viene impostata la soglia di allarme. Una soglia statica significa che la soglia di allarme è un numero statico e costante scelto dall'utente. Una soglia di rilevamento delle anomalie significa che CloudWatch determina un intervallo di valori usuali e l'allarme si attiva se il conteggio effettivo supera la soglia di questa banda. Non è necessario scegliere il rilevamento delle anomalie per un allarme di rilevamento delle anomalie nel registro. Per ulteriori informazioni sul rilevamento delle anomalie metriche, consulta Utilizzo del rilevamento delle anomalie. CloudWatch

6. Per Whenever ***`your-metric-name`*** is.. , scegli Maggiore, Maggiore/Uguale, Minore/Uguale o Inferiore. Per di . . ., specifica un numero per il valore di soglia. L'allarme entra in **`ALARM`** stato se il rilevatore di anomalie rileva più di questo numero di allarmi durante un periodo specificato da Period.

7. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

   Per creare un allarme M di N, specifica un numero del primo valore inferiore a quello del secondo valore. Per ulteriori informazioni, vedere Valutazione di un allarme.

8. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, vedi Configurazione del modo in cui gli CloudWatch allarmi trattano i dati mancanti.
9. Scegli Next (Successivo).
10. Per Notifica, scegli Aggiungi notifica, quindi specifica un argomento di Amazon SNS per notificare quando l'allarme passa allo stato `ALARM`OK, o. `INSUFFICIENT_DATA`

    a. (Facoltativo) Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

    > **Note**
    >
    > Ti consigliamo di impostare l'allarme in modo da intervenire quando entra in stato Dati insufficienti oltre a quando entra in stato Allarme. Questo perché molti problemi relativi alla funzione Lambda che si connette all'origine dati possono causare il passaggio dell'allarme a Dati insufficienti.

    b. (Facoltativo) Per fare in modo che l'allarme non invii notifiche Amazon SNS, scegli Rimuovi.
11. (Facoltativo) Se desideri che l'allarme esegua azioni per Amazon EC2 Auto Scaling, Amazon EC2 AWS Systems Manager, ticket oppure scegli il pulsante appropriato e specifica lo stato e l'azione dell'allarme.

    > **Note**
    >
    > L'allarme può eseguire le operazioni Systems Manager solo quando entra nello stato `ALARM`. Per informazioni sulle azioni di Systems Manager, vedere Configurazione CloudWatch per creare OpsItems e Creazione di incidenti.
12. Scegli Next (Successivo).
13. In Add a description (Aggiungere una descrizione), immetti un nome e una descrizione per l'allarme e scegli Next (Successivo). Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.

> **Tip**
>
> Il nome dell'allarme deve contenere solo caratteri UTF-8. Non può contenere caratteri di controllo ASCII.

14. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni dell'allarme sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Metriche pubblicate dai rilevatori di anomalie di registro

CloudWatch Logs pubblica la metrica in metrics. AnomalyCount CloudWatch Questa metrica viene pubblicata nel namespace. `AWS/Logs`

La AnomalyCountmetrica viene pubblicata con le seguenti dimensioni:

- LogAnomalyDetector— Il nome del rilevatore di anomalie
- LogAnomalyPriority— Il livello di priorità dell'anomalia

Crittografa un rilevatore di anomalie e i relativi risultati con AWS KMS

I dati del rilevatore di anomalie sono sempre crittografati nei registri. CloudWatch Per impostazione predefinita, CloudWatch Logs utilizza la crittografia lato server per i dati inattivi. In alternativa, è possibile utilizzare questa crittografia AWS Key Management Service . In tal caso, la crittografia viene eseguita utilizzando una AWS KMS chiave. L'utilizzo della crittografia AWS KMS è abilitato a livello di rilevatore di anomalie, associando una chiave KMS a un rilevatore di anomalie.

> **Important**
>
> CloudWatch Logs supporta solo chiavi KMS simmetriche. Non utilizzare una chiave asimmetrica per crittografare i dati nei gruppi di log. Per ulteriori informazioni, consulta la sezione relativa all'uso di chiavi simmetriche e asimmetriche.

Limits

- Per eseguire la procedura seguente, devi avere le seguenti autorizzazioni: `kms:CreateKey`, `kms:GetKeyPolicy` e `kms:PutKeyPolicy`.
- Dopo aver associato o dissociato una chiave da un rilevatore di anomalie, possono essere necessari fino a cinque minuti prima che l'operazione abbia effetto.
- Se si revoca l'accesso CloudWatch dei log a una chiave associata o si elimina una chiave KMS associata, i dati crittografati in CloudWatch Logs non possono più essere recuperati.

Fase 1: Creare una chiave AWS KMS

Per creare una chiave KMS, utilizza il seguente comando create-key:

```
aws kms create-key
```

L'output contiene l'ID chiave e l'Amazon Resource Name (ARN) della chiave. Di seguito è riportato un output di esempio:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "key-default-1",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/key-default-1",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Fase 2: Impostazione delle autorizzazioni sulla chiave KMS

Per impostazione predefinita, tutte AWS KMS le chiavi sono private. Solo il proprietario della risorsa può utilizzarla per crittografare e decrittare i dati. Tuttavia, il proprietario della risorsa può concedere ad altri utenti e risorse le autorizzazioni per accedere alla chiave KMS. Con questo passaggio, si CloudWatch concede al servizio Logs l'autorizzazione principale a utilizzare la chiave. L'entità del servizio deve trovarsi nella stessa AWS regione in cui è archiviata la chiave KMS.

Come procedura ottimale, ti consigliamo di limitare l'uso della chiave KMS solo agli AWS account o ai rilevatori di anomalie specificati.

Innanzitutto, salva la politica predefinita per la tua chiave KMS `policy.json` utilizzando il seguente comando: get-key-policy

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Aprire il file `policy.json` in un editor di testo e aggiungere la sezione in grassetto da una delle seguenti istruzioni. Separare l'istruzione esistente dalla nuova istruzione con una virgola. Queste istruzioni utilizzano `Condition` sezioni per migliorare la sicurezza della AWS KMS chiave. Per ulteriori informazioni, consulta AWS KMS chiavi e contesto di crittografia.

La `Condition` sezione di questo esempio limita l'uso della AWS KMS chiave all'account specificato, ma può essere utilizzata per qualsiasi rilevatore di anomalie.

JSON

```
{
"Version":"2012-10-17",
"Id": "key-default-1",
"Statement": [
    {
        "Sid": "EnableIAMUserPermissions",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:root"
        },
        "Action": "kms:*",
        "Resource": "*"
    },
```

```
        {
            "Sid": "AllowCloudWatchLogsEncryption",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsDescribeKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsReEncryption",
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws-crypto-ec:aws:logs:arn":
 "arn:aws:logs:us-east-1:123456789012:anomaly-detector:*"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsDescribeKeyForReEncryption",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

Infine, aggiungi la politica aggiornata utilizzando il seguente comando: put-key-policy

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Fase 3: Associare una chiave KMS a un rilevatore di anomalie

Puoi associare una chiave KMS a un rilevatore di anomalie quando la crei nella console o utilizzando l'opzione o. AWS CLI APIs

Fase 4: Dissociare la chiave da un rilevatore di anomalie

Dopo che una chiave è stata associata a un rilevatore di anomalie, non è possibile aggiornarla. L'unico modo per rimuovere la chiave è eliminare il rilevatore di anomalie e quindi ricrearlo.

Risolvi i problemi con CloudWatch Logs Live Tail

CloudWatch Logs Live Tail ti aiuta a risolvere rapidamente gli incidenti visualizzando un elenco in streaming dei nuovi eventi di registro man mano che vengono inseriti. Puoi visualizzare, filtrare ed evidenziare i log importati quasi in tempo reale, in modo da poter rilevare e risolvere rapidamente i problemi. Puoi filtrare i log in base ai termini specificati e anche evidenziare quelli che contengono termini specifici per in modo da poter trovare rapidamente ciò che stai cercando.

Le sessioni Live Tail comportano costi al minuto in base al tempo di utilizzo della sessione. Per ulteriori informazioni sui prezzi, consulta la scheda Logs in Amazon CloudWatch Pricing.

Live Tail è supportato solo per i gruppi di log nella classe di log Standard. Per ulteriori informazioni sulle classi di log, consultateClassi di registro.

Le seguenti sezioni spiegano come usare Live Tail nella console e in AWS CLI. Puoi anche avviare una sessione di Live Tail in modo programmatico. Per ulteriori informazioni, consulta StartLiveTail. Per esempi SDK, consulta Avvio di una sessione Live Tail utilizzando un AWS SDK.

Puoi anche usare Live Tail in. AWS Toolkit for Visual Studio Code Per avviare una sessione Live Tail dalla palette di comandi VS Code, consulta la sezione Amazon CloudWatch Logs Live Tail della Guida per l' AWS Toolkit for Visual Studio Code utente.

La funzione Live Tail è disponibile in tutte le AWS regioni commerciali. Non è disponibile nelle regioni della Cina o nelle regioni AWS GovCloud (Stati Uniti).

> **Note**
>
> L'`StartLiveTail`API indirizza le richieste utilizzando l'iniezione del prefisso dell'host SDK. Le versioni SDK rilasciate prima del 1° aprile 2026 indirizzano `astreaming-logs.`*`Region`*`.amazonaws.com`, che non supporta gli endpoint VPC. Le versioni SDK rilasciate a partire dal 1° aprile 2026 indirizzano `astream-logs.`*`Region`*`.amazonaws.com`, che supporta gli endpoint VPC. Per configurare un endpoint VPC per questa API, consulta Creazione di un endpoint VPC per i registri. CloudWatch

Avvia una sessione Live Tail utilizzando il AWS CLI

Il `start-live-tail` AWS CLI comando avvia una sessione di streaming Live Tail per uno o più gruppi di log in un terminale. Una sessione Live Tail può durare fino a tre ore. Se più di 500 eventi

di registro al secondo corrispondono al filtro, gli eventi di registro visualizzati sono un esempio degli eventi di registro totali, per fornire un'esperienza di tailing in tempo reale. Per ulteriori informazioni sul `start-live-tail` comando, consulta start-live-tail

È possibile utilizzarlo in due modalità: `start-live-tail`

- solo stampa: questa è la modalità predefinita
- interattivo

solo per la stampa

In `print-only` modalità, gli eventi di registro vengono trasmessi in streaming sul terminale. Ogni secondo vengono aggiunti nuovi eventi in fondo, creando un'esperienza di tailing quasi in tempo reale simile a quella di Linux`tail -f`.

Per avviare una sessione di Live Tail in modalità di sola stampa, inserisci il seguente comando.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs
```

Quando utilizzate la modalità di sola stampa, potete anche collegarla ad altri comandi Linux per aumentarne le capacità analitiche. L'esempio seguente filtra gli eventi di registro con la `error` parola chiave e stampa la seconda e la quarta colonna di questi eventi per facilitare l'estrazione di informazioni particolari.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs --mode print-only | grep "error" | awk '{print
 $2, $4}'
```

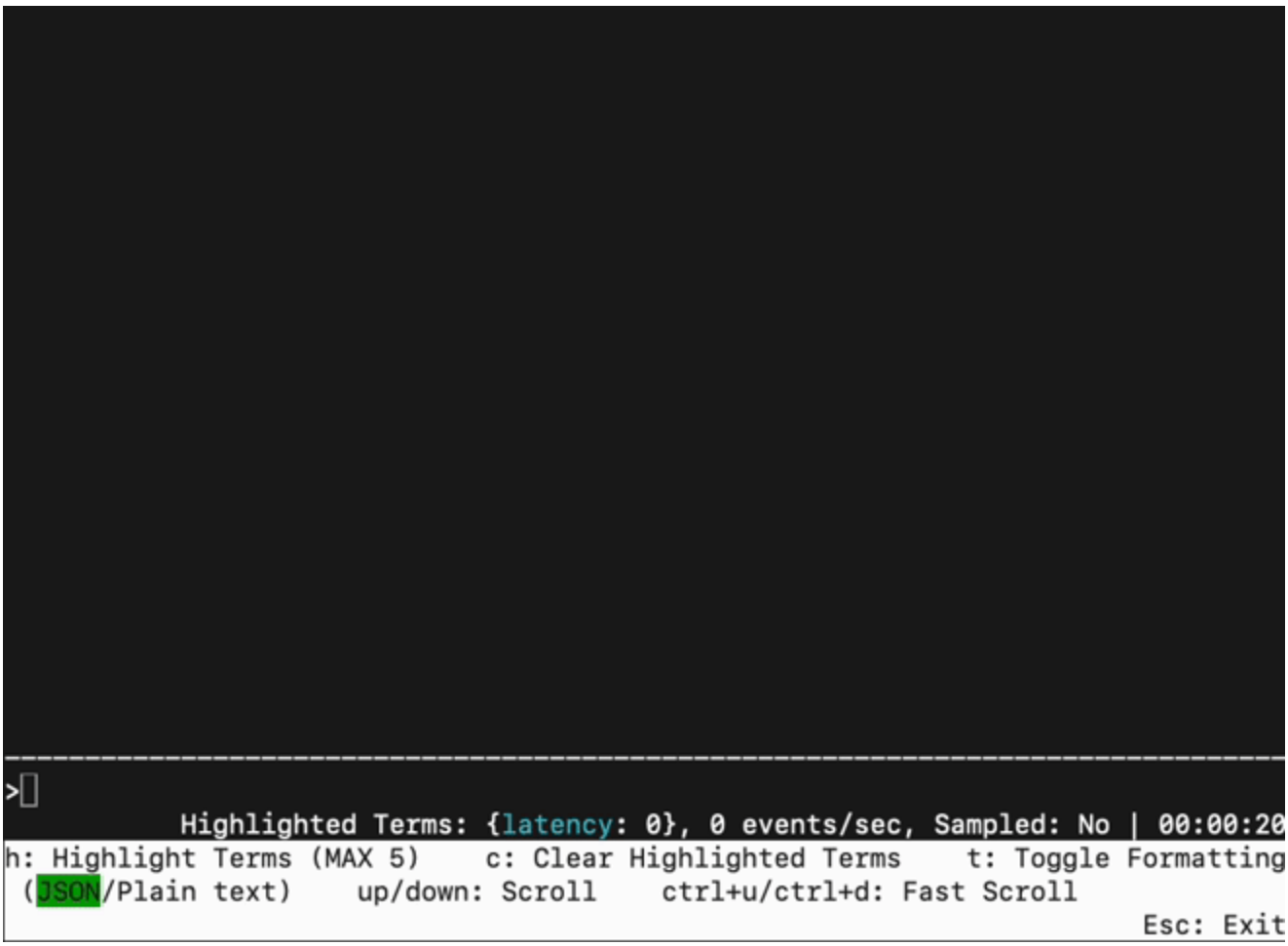
interattivo

In `interactive` modalità, è possibile evidenziare i termini e alternare il formato degli eventi del registro di output tra JSON e testo normale. La modalità interattiva mostra anche informazioni sulla sessione di Live Tail, come la durata della sessione, se la sessione viene campionata, i termini attualmente evidenziati e il conteggio delle volte in cui sono stati incontrati.

Per avviare una sessione Live Tail in modalità interattiva, immettete il seguente comando.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs --mode interactive
```

La sessione Live Tail ha inizio. Il video seguente mostra parte di una sessione di esempio.



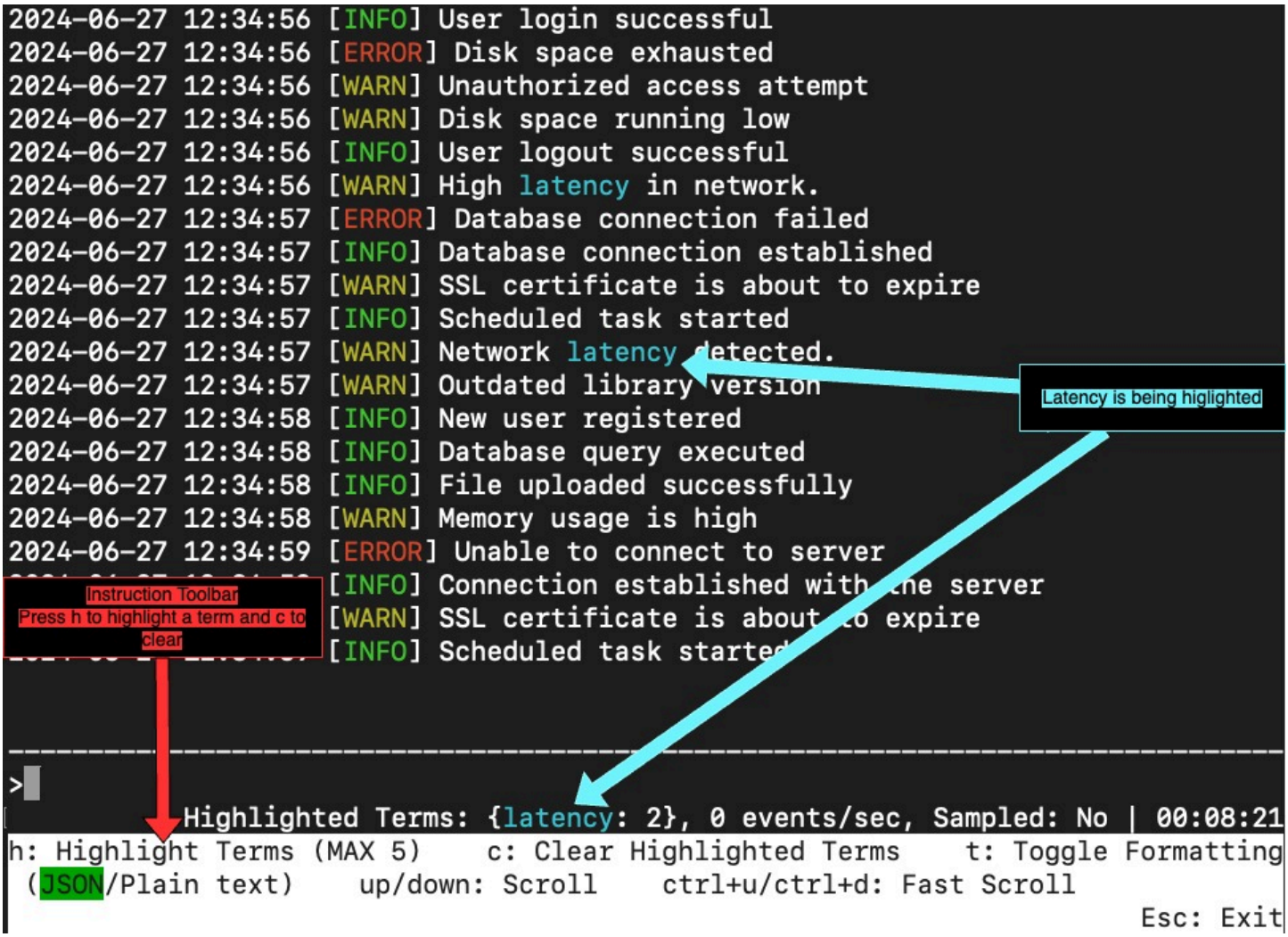
Per evidenziare un termine nei log di streaming, premi h e inserisci il termine. Di seguito viene mostrata la schermata dopo l'evidenziazione del termine`latency`.

Per cancellare un termine evidenziato, premi c e digita il numero che rappresenta il termine che desideri interrompere l'evidenziazione.

Puoi premere t per alternare il formato di visualizzazione degli eventi in arrivo tra JSON e testo normale. Questa funzionalità di commutazione è la soluzione ottimale e viene utilizzata solo se il formato degli eventi di registro è compatibile.

È possibile utilizzare i tasti freccia su e freccia giù per scorrere e utilizzare `CTRL+u` e `CTRL+d` per scorrere più velocemente.

L'immagine seguente mostra l'evidenziazione del `latency` termine durante una sessione di Live Tail.



Avvia una sessione Live Tail nella console

Utilizzi la CloudWatch console per avviare una sessione Live Tail. La procedura seguente spiega come avviare una sessione Live Tail utilizzando l'opzione Live Tail nel riquadro di navigazione a sinistra. Puoi anche avviare sessioni Live Tail dalla pagina Log Groups o dalla pagina CloudWatch Logs Insights.

Se utilizzi policy di protezione dei dati per mascherare i dati sensibili in un gruppo di log che stai visualizzando con Live Tail, i dati sensibili appaiono sempre mascherati nella sessione Live Tail. Per ulteriori informazioni sul mascheramento dei dati sensibili nei gruppi di log, consulta Incremento della protezione dei dati di log sensibili con il mascheramento.

> **Important**
>
> Se il team addetto alla sicurezza della rete non consente l'uso di socket Web, al momento non è possibile accedere alla parte Live Tail della CloudWatch console. Puoi usare Live Tail con le AWS CLI o API. Per ulteriori informazioni, consultare Avvia una sessione Live Tail utilizzando il AWS CLI e StartLiveTail.

Per avviare una sessione Live Tail

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Log, Live Tail.
3. In Seleziona gruppi di log, seleziona i gruppi di log di cui desideri visualizzare gli eventi, nella sessione Live Tail. Puoi selezionare fino a 10 gruppi di log.
4. (Facoltativo) Se hai selezionato un solo gruppo di log, puoi filtrare ulteriormente la tua sessione Live Tail selezionando uno o più flussi di log da cui visualizzare i log eventi. A tale scopo, in Seleziona flussi di log, seleziona i nomi dei flussi di log dall'elenco a discesa. In alternativa, puoi utilizzare la seconda casella in Seleziona flussi di log per inserire il prefisso del nome del flusso di log, quindi vengono selezionati tutti i flussi di log con nomi che corrispondono al prefisso.
5. (Facoltativo) Per visualizzare solo i log eventi che contengono determinate parole o altre stringhe, inserisci la parola o la stringa in `Add filter patterns`.

   Ad esempio, per visualizzare solo i log eventi che includono la parola `Warning`, inserisci **`Warning`**. I filtri fanno distinzione tra maiuscole e minuscole. Puoi includere più termini e operatori di modelli in questo campo:

   - **`error 404`** mostra solo i log eventi che includono sia `error` che `404`
   - **`?Error ?error`** mostra i log eventi che includono `Error` o `error`
   - **`-INFO`** mostra tutti i log eventi che non includono `INFO`
   - **`{ $.eventType = "UpdateTrail" }`** mostra tutti i log eventi JSON in cui il valore del campo del tipo di evento è `UpdateTrail`

   Puoi anche usare l'espressione regolare (regex) per filtrare:

   - **`%ERROR%`** utilizza la regex per visualizzare tutti i log eventi costituiti dalla parola chiave ERROR

- **`{ $.names = %Steve% }`** utilizza la regex per visualizzare i log eventi JSON in cui Steve si trova nella proprietà `"name"`
- **`[ w1 = %abc%, w2 ]`** utilizza la regex per visualizzare i log eventi delimitati da spazi in cui la prima parola è abc

Per ulteriori informazioni sulla sintassi dei modelli di filtro, consulta la sezione Sintassi di filtri e modelli.

6. (Facoltativo) Per evidenziare alcuni dei log eventi visualizzati, inserisci un termine da cercare ed evidenziare in Live Tail. Inserisci i termini da evidenziare uno alla volta. Se aggiungi più termini da evidenziare, viene assegnato un colore diverso per indicare ogni termine. Un indicatore di evidenziazione viene visualizzato a sinistra di qualsiasi log eventi che contiene il termine specificato e viene visualizzato anche sotto il termine stesso quando si espande il log eventi nella finestra principale per visualizzarlo nella sua interezza.

   Puoi utilizzare il filtro insieme all'evidenziazione per risolvere rapidamente i problemi. Ad esempio, potresti filtrare gli eventi per visualizzare solo quelli che contengono `Error` e quindi evidenziare anche quelli che contengono `404`.

7. Per avviare la sessione, scegli Applica filtri

   I log eventi corrispondenti vengono visualizzati nella finestra. Inoltre, vengono visualizzate le seguenti informazioni:

   - Il timer mostra per quanto tempo la sessione Live Tail è stata attiva.
   - events/secmostra quanti eventi di registro acquisiti al secondo corrispondono ai filtri impostati.
   - Per evitare che la sessione scorra troppo velocemente perché molti eventi corrispondono ai filtri, i CloudWatch registri potrebbero visualizzare solo alcuni eventi corrispondenti. In tal caso, la percentuale di eventi corrispondenti visualizzati sullo schermo viene mostrata in % visualizzata.

8. Per mettere in pausa il flusso degli eventi e analizzare ciò che è attualmente visualizzato, scegli un punto qualsiasi nella finestra degli eventi.

9. Durante la sessione, puoi utilizzare le seguenti opzioni per visualizzare maggiori dettagli su ogni log eventi.

   - Per visualizzare l'intero testo di un log eventi nella finestra principale, scegli la freccia accanto al relativo log eventi.

- Per visualizzare l'intero testo di un log eventi in una finestra laterale, scegli la lente di ingrandimento + accanto al relativo log eventi. Il flusso di eventi viene sospeso e viene visualizzata la finestra laterale.

  La visualizzazione del testo di un log eventi nella finestra laterale può essere utile per confrontarne il testo con altri eventi nella finestra principale.

10. Per arrestare la sessione Live Tail, scegli Arresta.
11. Per riavviare la sessione, utilizza facoltativamente il pannello Filtro per modificare i criteri di filtro e scegli Applica filtri. Quindi selezionare Start (Avvia).

Cross-account Centralizzazione dei log interregionali

La centralizzazione dei dati di Amazon CloudWatch Logs consente di AWS Organizations raccogliere i dati di log da più account membri in un unico repository di dati utilizzando regole di centralizzazione tra account e regioni. È l'utente a definire le regole che replicano automaticamente i dati di log da più account e Regioni AWS in un account centralizzato all'interno dell'organizzazione. Questa funzionalità semplifica il consolidamento dei log per migliorare il monitoraggio, l'analisi e la conformità centralizzati nell'intera infrastruttura. AWS

CloudWatch La centralizzazione dei dati dei log offre flessibilità di configurazione per soddisfare i requisiti operativi e di sicurezza, come la possibilità di configurare un'area di backup durante l'impostazione delle regole all'interno dell'account di destinazione per garantire una maggiore resilienza. Inoltre, hai il pieno controllo sul comportamento di crittografia dei gruppi di log copiati dagli account di origine per gestire i dati originariamente crittografati con chiavi KMS gestite dal cliente.

> **Note**
>
> La funzionalità di centralizzazione dei CloudWatch log elabora solo i nuovi dati di registro che arrivano negli account di origine dopo aver creato la regola di centralizzazione. I dati di registro cronologici (registri esistenti prima della creazione della regola) non sono centralizzati.

Concetti di centralizzazione dei dati

Prima di iniziare a utilizzare la centralizzazione dei dati di CloudWatch Logs, acquisisci familiarità con i seguenti concetti:

Regola di centralizzazione

: Una configurazione che definisce il modo in cui i dati di registro degli account e delle aree di origine vengono replicati su un account e un'area di destinazione. Le regole specificano i criteri di origine e le impostazioni di destinazione.

Account di origine

: L' AWS account da cui provengono i dati di registro. Gli eventi di registro degli account di origine vengono replicati nell'account di destinazione in base alle regole di centralizzazione definite dall'utente.

Account di destinazione

L' AWS account di destinazione in cui sono archiviati i dati di registro replicati. Questo account funge da posizione centralizzata per l'analisi e il monitoraggio dei log.

Regione di backup

Un'area secondaria opzionale all'interno dell'account di destinazione in cui i dati di registro possono essere replicati per una maggiore resilienza e scopi di disaster recovery.

Propagazione di tag

Una funzionalità opt-in che propaga i tag delle risorse dai gruppi di log di origine ai corrispondenti gruppi di log di destinazione. La propagazione dei tag utilizza un ruolo IAM gestito dal cliente nell'account di destinazione per aggiungere, aggiornare e rimuovere i tag nei gruppi di log di destinazione. È possibile configurare la propagazione dei tag aggiungendo un `TagPropagationConfiguration` blocco alla configurazione di destinazione della regola di centralizzazione.

Crittografia nei log CloudWatch

I dati dei gruppi di log sono sempre crittografati nei CloudWatch log. Per impostazione predefinita, CloudWatch Logs utilizza la crittografia lato server con Advanced Encryption Standard Galois/Counter Mode (AES-GCM) a 256 bit per crittografare i dati di registro inattivi. In alternativa, è possibile utilizzare AWS Key Management Service per questa crittografia. Per ulteriori informazioni, consulta la documentazione sulla crittografia dei CloudWatch log.

- Come funziona la crittografia durante la centralizzazione: la centralizzazione dei CloudWatch log copia attivamente i dati di registro al momento dell'inserimento dagli account di origine agli account di destinazione. Durante questo processo, i dati rimangono crittografati durante il transito utilizzando una AWS chiave di servizio di proprietà. I dati inattivi nei gruppi di log di origine e di destinazione vengono crittografati utilizzando il metodo di crittografia scelto (chiavi KMS gestite dal cliente o AWS di proprietà). Se utilizzi una chiave KMS gestita dal cliente nei gruppi di log di destinazione, aggiungi il tag `LogsManaged = true` alla chiave kms per consentire al servizio di centralizzazione di accedervi.
- Quando sono richieste le autorizzazioni KMS:
  - Se utilizzi chiavi KMS gestite dal cliente nei tuoi account di origine, CloudWatch Logs richiede le autorizzazioni KMS nei seguenti scenari di esempio:
    - Gestione del throughput: quando vengono raggiunti i limiti di throughput della centralizzazione, i dati di registro vengono temporaneamente archiviati crittografati con la chiave KMS gestita dal cliente fino a quando la larghezza di banda non diventa disponibile.

- Protezione e redazione dei dati: quando i gruppi di log di origine hanno le policy di protezione dei dati abilitate, CloudWatch Logs richiede le autorizzazioni di decrittografia per accedere ai dati di registro non elaborati per centralizzarli.

> **Important**
>
> Le regole di centralizzazione sono gestite dall'account di gestione dell' AWS organizzazione o dall'amministratore delegato. Per escludere i gruppi di KMS-encrypted log gestiti dai clienti dalla centralizzazione, configura le impostazioni della regola su «Non centralizzare i gruppi di log crittografati con AWS chiave KMS».

Configurazione della centralizzazione dei log

Per configurare CloudWatch Logs Centralization, è necessario configurare le regole di centralizzazione che definiscono il modo in cui i dati di log fluiscono dai gruppi di log negli account di origine ai gruppi di log nell'account di destinazione.

Una volta abilitata la regola di centralizzazione e replicati gli eventi di registro sull'account di destinazione, puoi creare filtri per metriche, sottoscrizioni e account su gruppi di log centralizzati con funzionalità di filtro avanzate. Questi filtri possono indirizzare gli eventi di registro provenienti da account e aree di origine specifici e possono emettere informazioni sull'account di origine e sulla regione come dimensioni metriche. Per ulteriori informazioni, consulta Creazione di parametri da log eventi mediante filtri.

Prerequisiti

- AWS Organizations devono essere configurati e gli account di origine e di destinazione devono appartenere entrambi all'organizzazione.
- È necessario abilitare l'accesso affidabile per l'account di gestione e l'account di destinazione CloudWatch, in modo da fornire l'accesso ai dati di registro.

> **Note**
>
> Si consiglia di abilitare l'accesso affidabile tramite la console, che crea automaticamente il ruolo collegato al servizio (SLR) richiesto. Se l'accesso affidabile è abilitato tramite altri metodi, il ruolo collegato al servizio dovrà essere creato separatamente.

Prerequisiti per la propagazione dei tag

Per abilitare la propagazione dei tag, è necessario completare la seguente configurazione aggiuntiva:

Crea un ruolo IAM gestito dal cliente

È necessario creare un ruolo IAM gestito dal cliente nell'account di destinazione con la seguente configurazione:

Policy di attendibilità

La politica di fiducia del ruolo deve consentire al ruolo collegato al servizio di centralizzazione di assumerla. Il seguente esempio di politica di fiducia concede al ruolo `AWSServiceRoleForObservabilityAdmin_LogsCentralization` collegato al servizio l'autorizzazione ad assumere il ruolo di destinazione. Sostituisci `<destination-account-id>` con l'ID del tuo account di destinazione e `<organization-id>` con il tuo ID. AWS Organizations

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::<destination-account-id>:role/aws-
service-role/logs-centralization.observabilityadmin.amazonaws.com/
AWSServiceRoleForObservabilityAdmin_LogsCentralization"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "sts:ExternalId": "<organization-id>"
      }
    }
  }]
}
```

Policy delle autorizzazioni

La politica delle autorizzazioni del ruolo deve concedere operazioni sui tag sui gruppi di log di destinazione. L'esempio seguente concede le autorizzazioni minime richieste. Sostituisci `<destination-account-id>` con il tuo valore. È possibile estendere l'ambito `Resource` a modelli ARN di gruppi di log specifici.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "logs:ListTagsForResource",
      "logs:TagResource",
      "logs:UntagResource"
    ],
    "Resource": "arn:aws:logs:*:<destination-account-id>:log-group:*"
  }]
}
```

Concedi iam: PassRole autorizzazione

È necessario disporre dell'`iam:PassRole`autorizzazione per il ruolo di destinazione, accessibile al servizio di centralizzazione tramite la chiave di `iam:PassedToService` condizione. L'esempio seguente concede l'autorizzazione a passare il ruolo. Sostituisci `<destination-account-id>` e `<your-tag-role-name>` con i tuoi valori.

> **Note**
>
> Questa dichiarazione si riferisce alla tua politica di identità (la chiamata principale `CreateCentralizationRuleForOrganization` o`UpdateCentralizationRuleForOrganization`), non al ruolo di destinazione stesso.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::<destination-account-id>:role/<your-tag-role-name>",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "logs-centralization.observabilityadmin.amazonaws.com"
      }
    }
  }]
}
```

Personalizzazione dei nomi dei gruppi di log di destinazione

Quando si crea una regola di centralizzazione, è possibile personalizzare la struttura dei nomi dei gruppi di log di destinazione utilizzando gli attributi. Questi attributi vengono automaticamente sostituiti con valori effettivi quando vengono creati i gruppi di log, consentendo di organizzare i log gerarchicamente nell'account di destinazione. Per impostazione predefinita, viene utilizzato solo l'`${source.logGroup}`attributo, che unisce tutti i gruppi di log con lo stesso nome nell'account di destinazione. Se una variabile non può essere risolta, eredita il valore dalla sua variabile padre nella gerarchia.

Attributi disponibili

È possibile utilizzare i seguenti attributi nel modello di nome del gruppo di log di destinazione:

Attributi del nome del gruppo di log di destinazione

Attributo	Description
`${source.accountId}`	L'ID AWS dell'account da cui ha avuto origine il registro.
`${source.region}`	Il Regione AWS luogo di origine del registro.
`${source.logGroup}`	Il nome del gruppo di log originale dell'account di origine.
`${source.org.id}`	L' AWS Organizations ID dell'account di origine.
`${source.org.ouId}`	L'ID dell'unità organizzativa dell'account di origine
`${source.org.rootId}`	L'ID principale dell'organizzazione
`${source.org.path}`	Il percorso organizzativo completo dall'account alla radice

Esempi

Conserva la struttura originale del gruppo di log

Modello: `/centralized/${source.accountId}${source.logGroup}`

Risultato: `/centralized/123456789012/aws/lambda/my-function`

Organizza per account e area geografica

Modello: `/centralized/${source.accountId}/${source.region}`

Risultato: `/centralized/123456789012/us-east-1`

Organizza per struttura organizzativa

Modello: `/logs/${source.org.id}/${source.org.ouId}/${source.accountId}`

Risultato: `/logs/o-abc123/ou-xyz-12345678/123456789012`

Struttura piatta semplice

Modello: `/centralized-logs`

Risultato: `/centralized-logs`

Best practice

- Includi l'ID dell'account di origine per identificare facilmente da quale account log provengono.
- Includi la regione di origine se esegui la centralizzazione da più regioni.
- Strutturare i nomi dei gruppi di log di destinazione in modo che contengano meno di 512 caratteri. CloudWatch Logs impone una lunghezza massima dei nomi dei gruppi di log di 512 caratteri.
- Se si abilita la propagazione dei tag, il pattern deve contenere tutti e tre gli attributi: `${source.logGroup}${source.accountId}`, e. `${source.region}` Ciò garantisce che ogni gruppo di log di origine sia mappato su un gruppo di log di destinazione univoco.

Creazione di una regola di centralizzazione

Utilizza la procedura seguente per creare una regola di centralizzazione che replica i dati di registro dagli account di origine all'account di destinazione.

Per creare una regola di centralizzazione

1. Accedi alla CloudWatch console nell'account Management o Delegated Administrator dell'organizzazione.
2. Seleziona Impostazioni.
3. Vai alla scheda Organizzazione.

4. Scegli Configura regola.
5. Specifica i dettagli della fonte impostando i seguenti campi, quindi scegli Avanti:
   a. Nome della regola di centralizzazione: inserisci un nome univoco per la regola di centralizzazione.
   b. Account di origine: definisci i criteri di selezione della fonte per scegliere gli account da cui centralizzare i dati di telemetria. I criteri di selezione possono includere:
      - Un elenco degli account membri dell'organizzazione
      - Un elenco delle unità organizzative dell'organizzazione
      - L'intera organizzazione

      È possibile fornire i criteri di selezione in due modalità:
      - Builder: un'esperienza basata sui clic per generare i criteri di selezione della fonte
      - Editor: una casella di testo in formato libero per fornire i criteri di selezione della fonte

      Sintassi supportata per i criteri di selezione della fonte:
      - Chiavi supportate: OrganizationId | OrganizationUnitId | AccountId | *
      - Operatori supportati: = | IN | OR
   c. Regioni di origine: seleziona un elenco di regioni per cercare i dati di telemetria da centralizzare.
6. Specifica i dettagli della destinazione impostando i seguenti campi, quindi scegli Avanti:
   a. Account di destinazione: seleziona un account nell'organizzazione che funge da destinazione centrale per i dati di telemetria.
   b. Regione di destinazione: seleziona una regione principale che memorizza una copia dei dati di telemetria centralizzati.
   c. Regione di backup: facoltativamente, seleziona una regione che memorizza una seconda copia dei dati di telemetria centralizzati.
7. Specifica i dati di telemetria impostando i seguenti campi, quindi scegli Avanti:
   a. Gruppi di log: scegli una delle seguenti opzioni:
      - Tutti i gruppi di log: centralizza i log di tutti i gruppi di log negli account di origine.

- Filtra il gruppo di log: centralizza i log da un sottoinsieme di gruppi di log negli account di origine, in base ai criteri di selezione. È possibile fornire i criteri di selezione in due modalità:
  - Builder: un'esperienza basata sulla scelta per generare i criteri di selezione
  - Editor: una casella di testo in formato libero per fornire i criteri di selezione

  Sono disponibili due criteri di selezione che è possibile utilizzare per filtrare i log:

  - Criteri di selezione dei gruppi di log: i criteri di selezione che specificano quali gruppi di log di origine centralizzare.
    - Chiavi supportate: | * LogGroupName
    - Operatori supportati: = |! = | IN | NON IN | E | O | MI PIACE | NON MI PIACE
  - Criteri di selezione delle fonti di dati: i criteri di selezione che specificano quali fonti di dati centralizzare.
    - Chiavi supportate: | DataSourceName DataSourceType
    - Operatori supportati: = |! = | IN | NON IN | E | O | MI PIACE | NON MI PIACE

  Quando vengono specificati sia i criteri di selezione dei gruppi di log che i criteri di selezione dell'origine dati, un evento di log deve corrispondere a entrambi i criteri per essere centralizzato.

b. Gruppo di log crittografato KMS

> **Important**
>
> CloudWatch le regole di centralizzazione non riusciranno a fornire i log dall'account di origine ai gruppi di log di destinazione se la chiave KMS fornita nella regola di centralizzazione non consente a CloudWatch Logs di utilizzarla. Se utilizzi una chiave KMS gestita dal cliente nei gruppi di log di destinazione, aggiungi il tag LogsManaged = true alla chiave kms. Per ulteriori informazioni, consulta Fase 2: Impostazione delle autorizzazioni sulla chiave KMS.

Scegli una delle seguenti opzioni:

- Centralizza i gruppi di log di origine crittografati con chiavi KMS gestite dal cliente utilizzando una chiave KMS gestita dal cliente specifica per la destinazione: centralizza gli eventi di registro dai gruppi di log di origine crittografati con chiavi KMS gestite dal

cliente nei gruppi di log di destinazione crittografati con una chiave KMS gestita dal cliente nell'account di destinazione.

Quando si seleziona questa impostazione, è necessario impostare anche quanto segue:

- Chiave di crittografia di destinazione ARN: ARN della chiave KMS gestita dal cliente nell'account di destinazione e nella regione di destinazione principale, da associare ai gruppi di log di destinazione appena creati.
- Chiave di crittografia della destinazione di backup ARN (se è selezionata la regione di backup): ARN della chiave KMS gestita dal cliente nell'account di destinazione e nell'area di destinazione del backup, da associare ai gruppi di log di destinazione appena creati.
- Salta la centralizzazione ai gruppi di log di destinazione non crittografati (opzionale): se esiste già un gruppo di log senza una chiave KMS gestita dal cliente, non è possibile aggiornarne la crittografia. CloudWatch Scegli questa opzione per saltare la centralizzazione degli eventi di registro dai gruppi di log di origine crittografati con chiavi KMS gestite dal cliente nei gruppi di log di destinazione che non sono associati a una chiave KMS gestita dal cliente.

- Centralizza i gruppi di log crittografati con chiavi KMS gestite dal cliente nell'account di destinazione con chiave KMS AWS proprietaria: centralizza gli eventi di registro dai gruppi di log di origine crittografati con chiavi KMS gestite dal cliente in gruppi di log di destinazione di nuova creazione crittografati utilizzando una chiave KMS proprietaria. AWS
- Non centralizzare i gruppi di log crittografati con chiavi KMS gestite dal cliente: salta la centralizzazione degli eventi di registro dai gruppi di log di origine crittografati con chiavi KMS gestite dal cliente.

8. Rivedi la regola di centralizzazione, opzionalmente apporta le modifiche dell'ultimo minuto e scegli Crea politica di centralizzazione.

Modifica di una regola di centralizzazione

Utilizzare la procedura seguente per modificare una regola di centralizzazione esistente.

Per modificare una regola di centralizzazione

1. Accedi alla CloudWatch console nell'account Management o Delegated Administrator dell'organizzazione.
2. Seleziona Impostazioni.
3. Vai alla scheda Organizzazione.
4. Scegli Gestisci regole.
5. Seleziona la regola da aggiornare e scegli Modifica.
6. Aggiorna la configurazione della regola secondo necessità, scegliendo Avanti per procedere con ogni passaggio.
7. Nel passaggio 4, Rivedi e configura, scegli Aggiorna la politica di centralizzazione.

Visualizzazione di una regola di centralizzazione

Utilizzare la procedura seguente per visualizzare i dettagli di una regola di centralizzazione esistente.

Per visualizzare una regola di centralizzazione

1. Accedi alla CloudWatch console nell'account Management o Delegated Administrator dell'organizzazione.
2. Seleziona Impostazioni.
3. Vai alla scheda Organizzazione.
4. Scegli Gestisci regole.
5. Visualizza un elenco di tutte le regole di centralizzazione esistenti e scegli un nome di regola specifico per visualizzarne i dettagli.

Eliminazione di una regola di centralizzazione

Utilizzare la procedura seguente per eliminare una regola di centralizzazione esistente.

Per eliminare una regola di centralizzazione

1. Accedi alla CloudWatch console nell'account Management o Delegated Administrator dell'organizzazione.
2. Seleziona Impostazioni.

3. Vai alla scheda Organizzazione.
4. Scegli Gestisci regole.
5. Seleziona la regola da eliminare e scegli Elimina.
6. Per confermare l'eliminazione, scegliere Delete (Elimina).

Monitoraggio e risoluzione dei problemi relativi alle regole di centralizzazione

Puoi monitorare lo stato e le prestazioni delle regole di centralizzazione utilizzando le CloudWatch metriche, la console CloudWatch Logs e i log. AWS CloudTrail Ciò consente di garantire che i dati di registro vengano replicati correttamente e di identificare eventuali problemi relativi alla configurazione di centralizzazione.

CloudWatch I log forniscono:

1. Integrità delle regole per regola di centralizzazione
   a. Seleziona Impostazioni.
   b. Vai alla scheda Organizzazione.
   c. Scegli Gestisci regole.
2. Registra le chiamate API con AWS CloudTrail
3. CloudWatch pubblica anche le metriche per la centralizzazione, inclusi gli eventi di registro replicati, gli errori e la limitazione. Per ulteriori informazioni su queste metriche e le relative dimensioni, consulta. Metriche e dimensioni di centralizzazione

Stato di integrità delle regole di centralizzazione

Ogni regola di centralizzazione ha uno stato di integrità che indica se funziona correttamente. Puoi controllare lo stato delle regole tramite la console o a livello di codice utilizzando l'API.

Gli stati di integrità delle regole includono:

- `HEALTHY`: la regola funziona normalmente e replica i dati di registro come configurato
- `UNHEALTHY`: la regola ha riscontrato problemi ed è possibile che i dati non vengano replicati correttamente

- `PROVISIONING`: La centralizzazione per l'organizzazione è in fase di configurazione.

Quando una regola è contrassegnata come NON SANA, il `FailureReason` campo fornisce dettagli sul problema specifico che deve essere risolto.

Stato di integrità della propagazione dei tag

Lo stato di integrità della propagazione dei tag (`TagPropagationStatus`) è indipendente da quello complessivo della distribuzione `RuleHealth` dei log. Se configuri erroneamente il ruolo di destinazione, lo stato generale della regola non ne risente: solo la propagazione dei tag viene visualizzata come non integra.

- `Healthy`: il tentativo di propagazione dei tag più recente è riuscito.
- `Unhealthy`: L'ultimo tentativo di propagazione dei tag non è riuscito. Controlla il `TagPropagationFailureReason` campo per i dettagli. Per le procedure di risoluzione dei problemi, consulta la sezione sulla risoluzione dei problemi riportata di seguito.

Monitoraggio delle chiamate API di centralizzazione con AWS CloudTrail

AWS CloudTrail registra le chiamate API effettuate al servizio di centralizzazione, consentendoti di tenere traccia delle modifiche alla configurazione e risolvere i problemi relativi agli account che sono membri del tuo. AWS Organizations

CloudTrail Gli eventi chiave per la centralizzazione includono:

- `CreateCentralizationRuleForOrganization`: Quando viene creata una nuova regola di centralizzazione
- `UpdateCentralizationRuleForOrganization`: Quando una regola esistente viene modificata
- `DeleteCentralizationRuleForOrganization`: Quando una regola viene eliminata
- `GetCentralizationRuleForOrganization`: Quando vengono recuperati i dettagli della regola
- `ListCentralizationRulesForOrganization`: Quando vengono elencate le regole

È possibile utilizzare CloudTrail i log per verificare le modifiche alla configurazione della centralizzazione e correlarle con problemi di prestazioni o errori di replica.

Consigli sul monitoraggio

Per garantire che la centralizzazione funzioni correttamente, consigliamo di impostare CloudWatch allarmi sulle principali metriche di centralizzazione che forniamo a Metrics. CloudWatch Questo monitoraggio proattivo ti aiuta a rilevare tempestivamente i problemi e a mantenere una centralizzazione affidabile dei log in tutta l'organizzazione.

Le metriche chiave da monitorare includono:

- `IncomingCopiedBytes`: monitora il volume dei dati di registro che vengono replicati correttamente sul tuo account di destinazione. Un calo improvviso o l'assenza di questa metrica può indicare problemi di centralizzazione.
- `CentralizationError`: imposta allarmi per eventuali errori nel processo di centralizzazione per identificare e risolvere rapidamente i problemi.
- `CentralizationThrottled`: monitora gli eventi di limitazione che potrebbero influire sulle prestazioni di replica dei log.

Per un elenco completo delle metriche di centralizzazione disponibili e delle relative dimensioni, consulta. Metriche e dimensioni di centralizzazione

Se i log non vengono centralizzati come previsto, esamina i seguenti scenari comuni che possono impedire la centralizzazione dei log.

Dati di registro storici

La funzionalità di centralizzazione dei CloudWatch log elabora solo i nuovi dati di registro che arrivano negli account di origine dopo aver creato la regola di centralizzazione. I dati di registro cronologici (registri esistenti prima della creazione della regola) non sono centralizzati.

Autorizzazioni chiave KMS

Le regole di centralizzazione non riusciranno a fornire i log dall'account di origine ai gruppi di log di destinazione se la chiave KMS fornita nella regola di centralizzazione non consente CloudWatch a Logs di utilizzarla. Assicurati che la policy chiave KMS conceda le autorizzazioni necessarie a Logs. CloudWatch Per ulteriori informazioni, consulta Fase 2: Impostazione delle autorizzazioni sulla chiave KMS.

Configurazione delle chiavi KMS gestita dal cliente

Se hai selezionato Non centralizzare i gruppi di log crittografati con la chiave Customer Managed KMS durante la creazione della regola, gli eventi di log provenienti dai gruppi di log di origine crittografati con la chiave Customer Managed KMS verranno ignorati e non centralizzati.

Mancata corrispondenza della crittografia della destinazione

Se il gruppo di log di destinazione esiste già con una configurazione di crittografia KMS diversa da quella specificata dalla regola di centralizzazione e la risoluzione dei conflitti è impostata su SKIP, i record verranno eliminati e verrà generato un `DestinationEncryptionMismatch` errore. Ad esempio, ciò si verifica quando la destinazione ha una crittografia predefinita ma la regola specifica una chiave KMS gestita dal cliente.

Accesso affidabile non abilitato

L'accesso attendibile deve essere abilitato AWS Organizations affinché CloudWatch l'account di gestione e l'account di destinazione forniscano l'accesso ai dati di registro.

Criteri di selezione della fonte

Verifica che i criteri di selezione dell'origine della regola di centralizzazione siano configurati correttamente:

- Account e aree: assicurati che gli account e le aree di origine da cui provengono i log siano inclusi nella regola. I gruppi di log provenienti da account o aree non specificati nella regola non saranno centralizzati.
- Filtri dei gruppi di log: se sono stati configurati i filtri dei gruppi di log, verranno centralizzati solo i gruppi di log che corrispondono ai criteri specificati. Verifica che i criteri di selezione dei gruppi di log includano i gruppi di log che prevedi di centralizzare.
- Appartenenza all'organizzazione: sia l'account di origine che quello di destinazione devono appartenere alla stessa AWS Organizations organizzazione. Gli account esterni all'organizzazione non possono partecipare alla centralizzazione.

È stato raggiunto il limite di quota del gruppo di log

Se l'account di destinazione ha raggiunto il limite di quota per i gruppi di log, non è possibile creare nuovi gruppi di log per la centralizzazione. Verifica che l'account di destinazione disponga di una quota sufficiente per ospitare gruppi di log centralizzati di tutti gli account di origine. Puoi richiedere un aumento della quota, se necessario.

È stato superato il limite di lunghezza del nome del log stream

I nomi dei log stream hanno limiti di lunghezza massima. Quando la centralizzazione replica i flussi di log sull'account di destinazione, viene aggiunto un suffisso al nome del flusso di log. Se il nome del flusso di log risultante supera la lunghezza massima consentita, i record verranno eliminati e verrà emesso un `InvalidLogStream` errore nell'account del cliente.

Stato di integrità della regola

Controlla lo stato di integrità della regola di centralizzazione nella console o utilizzando l'`GetCentralizationRuleForOrganization`API. Se la regola è contrassegnata come NON SANA, esamina il `FailureReason` campo per dettagli specifici sul problema.

Stato di integrità della propagazione dei tag

Se `TagPropagationStatus` viene visualizzato`Unhealthy`, seleziona il `TagPropagationFailureReason` campo:

- `RoleNotAssumable`: il servizio non può assumere il ruolo di destinazione. Verifica che la politica di fiducia del ruolo consenta al ruolo collegato al servizio di centralizzazione di assumerlo e che `sts:ExternalId` corrisponda all'ID della tua organizzazione.
- `RoleLacksPermissions`: il ruolo è stato assunto ma la chiamata all'API del tag è stata negata. Verifica che le autorizzazioni del ruolo siano concesse `logs:ListTagsForResource` e `logs:UntagResource` relative ai gruppi di log di destinazione. `logs:TagResource`

Per diagnosticare i problemi di centralizzazione, esamina lo stato di integrità delle regole di centralizzazione nella console, controlla le CloudWatch metriche per eventuali errori e limitazioni ed esamina i log relativi alle chiamate API non riuscite. AWS CloudTrail Per ulteriori informazioni sulle metriche di centralizzazione, vedere. Metriche e dimensioni di centralizzazione

Utilizzo di gruppi di log e flussi di log

Un flusso di log è una sequenza di log eventi che condividono la stessa origine. Ogni fonte separata di log in CloudWatch Logs costituisce un flusso di log separato.

Un gruppo di log è un gruppo di flussi di log che condividono le stesse impostazioni di conservazione, monitoraggio e controllo degli accessi. Puoi definire i gruppi di log e specificare quali flussi inserire in ciascun gruppo. Non vi è alcun limite al numero di flussi di log che possono appartenere a un gruppo di log.

Per le organizzazioni che devono consolidare i dati di registro di più account e regioni, è possibile utilizzare CloudWatch Logs Centralization per replicare automaticamente i gruppi di log su un account centrale. Per ulteriori informazioni, consulta Cross-account Centralizzazione dei log interregionali.

È possibile utilizzare le procedure in questa sezione per lavorare con gruppi di log e flussi di log.

Crea un gruppo di log in CloudWatch Logs

Quando installi l'agente CloudWatch Logs su un'istanza Amazon EC2 utilizzando i passaggi nelle sezioni precedenti della CloudWatch Amazon Logs User Guide, il gruppo di log viene creato come parte di tale processo. Puoi anche creare un gruppo di log direttamente nella console. CloudWatch

Creazione di un gruppo di log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Gestione dei registri.
3. Selezionare Actions (Operazioni) e scegliere Create log group (Crea gruppo di log).
4. Immettere un nome per il gruppo di log, quindi selezionare Create log group (Crea gruppo di log).

> **Tip**
>
> È possibile preferire i gruppi di flussi di log, nonché i dashboard e gli allarmi, dal Preferiti e recenti nel riquadro di navigazione. Nella colonna Visitati di recente, passare il mouse sul gruppo di log che si desidera impostare come preferito e scegliere il simbolo della stella accanto a esso.

Invio di log a un gruppo di log

CloudWatch Logs riceve automaticamente gli eventi di registro da diversi AWS servizi. È inoltre possibile inviare altri eventi di registro a CloudWatch Logs utilizzando uno dei seguenti metodi:

- CloudWatch agente: l' CloudWatch agente unificato può inviare sia le metriche che i log ai registri. CloudWatch Per informazioni sull'installazione e l'utilizzo dell' CloudWatch agente, consulta Collecting metrics and logs from Amazon EC2 Instances and On-Premises Server with the CloudWatch Agent nella Amazon User Guide. CloudWatch
- AWS CLIput-log-events—Carica batch di eventi di registro su Logs. CloudWatch
- A livello di codice: l'PutLogEventsAPI consente di caricare in modo programmatico batch di eventi di registro nei registri. CloudWatch

Visualizza i dati di registro inviati ai registri CloudWatch

È possibile visualizzare e scorrere i dati di registro in stream-by-stream base a quelli inviati a CloudWatch Logs dall'agente CloudWatch Logs. Puoi specificare l'intervallo di tempo dei dati di log da visualizzare.

Visualizzazione dati di log

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione, scegli Gestione dei registri.
3. In Log Groups (Gruppi di log), seleziona il gruppo per visualizzare i flussi.
4. Nell'elenco dei gruppi di log, scegliere il nome del gruppo di log che si desidera visualizzare.
5. Nell'elenco dei flussi di log, scegliere il nome del flusso di log che si desidera visualizzare.
6. Per modificare la modalità di visualizzazione dei dati di log, procedi in uno dei seguenti modi:
   - Per espandere un singolo log eventi, scegliere la freccia accanto all'evento di log.
   - Per espandere tutti gli eventi di log e visualizzarli come testo normale, sopra l'elenco di eventi di log, seleziona Text (Testo).
   - Per filtrare gli eventi di log, digitare il filtro di ricerca desiderato nel campo di ricerca. Per ulteriori informazioni, consulta Creazione di parametri da log eventi mediante filtri.
   - Per visualizzare i dati di log per un intervallo di data e ora specificato, scegliere la freccia accanto alla data e all'ora, accanto al filtro di ricerca. Per specificare un intervallo di data e

ora, scegliere Absolute (Assoluto). Per scegliere un numero predefinito di minuti, ore, giorni o settimane, selezionare Relative (Relativo). Inoltre, è possibile passare da UTC a fuso orario locale.

Ricerca di dati di log utilizzando i modelli dei filtri

Puoi cercare i tuoi dati di log utilizzando la Sintassi del modello di filtro per filtri di parametri, filtri di sottoscrizione, filtri di log eventi e Live Tail. È possibile cercare tutti i flussi di log all'interno di un gruppo di log oppure utilizzando il AWS CLI è possibile cercare anche flussi di log specifici. Quando ciascuna ricerca è in esecuzione, restituisce fino alla prima pagina di dati disponibili e un token per recuperare la pagina successiva di dati o per continuare la ricerca. Se non ottieni alcun risultato, puoi continuare la ricerca.

Puoi impostare l'intervallo di tempo per cui desideri eseguire query per limitare l'ambito della ricerca. Puoi iniziare con un intervallo di dimensioni maggiori per individuare i punti in cui le linee di log interessati cadono e ridurre quindi l'intervallo di tempo per diminuire la visualizzazione dei log nell'intervallo di tempo interessato.

Inoltre puoi muoverti direttamente dai tuoi parametri estratti dai log ai log corrispondenti.

Se hai effettuato l'accesso a un account configurato come account di monitoraggio in modalità osservabile CloudWatch tra più account, puoi cercare e filtrare gli eventi di registro dagli account di origine collegati a questo account di monitoraggio. Per maggiori informazioni, consulta la sezione Osservabilità su più account di CloudWatch .

Ricerca di voci di log utilizzando la console

Puoi cercare voci di log che soddisfino un criterio specificato utilizzando la console.

Ricerca di voci di log utilizzando la console

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione, scegli Gestione dei registri.
3. In Log Groups (Gruppi di log), seleziona il nome del gruppo di log contenente il flusso di log da cercare.
4. In Log Streams (Flussi di log), seleziona il nome del flusso di log da cercare.
5. In Eventi di log, immettere la sintassi del filtro da utilizzare.

Cercare tutte le voci di log per un intervallo di tempo utilizzando la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Gestione dei registri.
3. In Log Groups (Gruppi di log), seleziona il nome del gruppo di log contenente il flusso di log da cercare.
4. Scegliere Cerca gruppo di log.
5. In Eventi di log, selezionare l'intervallo di data e ora e immettere la sintassi del filtro.

Cerca nelle voci del registro utilizzando il AWS CLI

È possibile cercare le voci di registro che soddisfano un criterio specificato utilizzando AWS CLI.

Per cercare le voci di registro utilizzando il AWS CLI

Al prompt dei comandi, esegui il comando seguente filter-log-events. Utilizza `--filter-pattern` per limitare i risultati per modello del filtro specificato e `--log-stream-names` per limitare i risultati a determinati flussi di log.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Per cercare le voci di registro in un determinato intervallo di tempo utilizzando il AWS CLI

Al prompt dei comandi, esegui il filter-log-eventscomando seguente:

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--start-time 1482197400000] [--end-
time 1482217558365] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Cambiare da parametri a log

Puoi accedere a specifiche voci di log da altre parti della console.

Per accedere dai widget del pannello di controllo ai log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Seleziona un pannello di controllo.

4. Nel widget, scegli l'icona View logs (Visualizza log) e quindi scegli View logs in this time range (Visualizza log in questo intervallo di tempo). Se è presente più di un filtro parametri, selezionane uno dall'elenco. Se sono presenti più filtri di parametri che possiamo mostrare nell'elenco, scegli Più filtri di parametri e seleziona o cerca un filtro di parametro.

Accesso da parametri a log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nel campo di ricerca nella scheda All metrics (Tutti i parametri), digitare il nome del parametro e premi Invio.
4. Seleziona uno o più parametri dai risultati della tua ricerca.
5. Scegli Actions (Azioni), View logs (Visualizza log). Se è presente più di un filtro parametri, selezionane uno dall'elenco. Se sono presenti più filtri di parametri che possiamo mostrare nell'elenco, scegli Più filtri di parametri e seleziona o cerca un filtro di parametro.

Risoluzione dei problemi

Search takes too long to complete (La ricerca richiede troppo tempo per essere completata)

Se si dispone di una notevole quantità di dati di log, la ricerca potrebbe richiedere molto tempo per essere completata. Per velocizzare la ricerca, è possibile eseguire le operazioni descritte di seguito:

- Se utilizzi il AWS CLI, puoi limitare la ricerca solo ai flussi di log che ti interessano. Ad esempio, se il tuo gruppo di log ha 1000 flussi di log, ma desideri visualizzare solo tre flussi di log che ritieni pertinenti, puoi utilizzare il AWS CLI per limitare la ricerca solo ai tre flussi di log all'interno del gruppo di log.
- Utilizza un intervallo di tempo più breve e più granulare, che riduce la quantità di dati da ricercare e velocizza la query.

Modifica la conservazione dei dati di registro in Logs CloudWatch

Per impostazione predefinita, i dati di registro vengono archiviati nei CloudWatch registri a tempo indeterminato. Tuttavia, puoi configurare per quanto tempo archiviare i dati di log in un gruppo di log. I dati che superano l'impostazione di conservazione corrente verranno eliminati. Puoi modificare la conservazione dei log per ciascun gruppo di log in qualsiasi momento.

> **Note**
>
> CloudWatch Logs non elimina immediatamente gli eventi di registro quando raggiungono l'impostazione di conservazione. In genere sono necessarie fino a 72 ore prima che gli eventi di log vengano eliminati, ma in rare situazioni potrebbe essere necessario più tempo.
> Ciò significa che se modifichi un gruppo di log per avere un'impostazione di conservazione più lunga quando contiene eventi di log che hanno superato la data di scadenza, ma non sono stati effettivamente eliminati, l'eliminazione di tali eventi di log richiederà fino a 72 ore dopo il raggiungimento della nuova data di conservazione. Per assicurarsi che i dati di log vengano eliminati in modo definitivo, mantieni un gruppo di log con l'impostazione di conservazione inferiore fino a quando non sono trascorse 72 ore dopo la fine del periodo di conservazione precedente oppure è stata confermata l'eliminazione degli eventi di log precedenti.
> Quando i log eventi raggiungono l'impostazione di conservazione, vengono contrassegnati per l'eliminazione. Una volta contrassegnati per l'eliminazione, non vengono più considerati nei costi di archiviazione, anche se vengono effettivamente eliminati solo in un secondo momento. Questi log eventi contrassegnati per l'eliminazione non sono inclusi quando si utilizza un'API per recuperare il valore `storedBytes` per vedere quanti byte sta archiviando un gruppo di log.

Modifica dell'impostazione di conservazione di log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra, scegli Log, Gruppi di log.
3. Individua il gruppo di log da aggiornare.
4. Nella colonna Conservazione per quel gruppo di log, scegli l'impostazione di conservazione corrente, ad esempio Never Expire.
5. Nell'impostazione Conservazione, per gli eventi Expire after, scegliete un valore di conservazione dei log, quindi scegliete Salva.

Protezione dei gruppi di log dall'eliminazione

Facoltativamente, è possibile abilitare la protezione dall'eliminazione per impedire l'eliminazione accidentale di importanti gruppi di log. Per informazioni dettagliate sulla protezione da eliminazione, vedereProtezione dei gruppi di log dall'eliminazione.

Protezione dei gruppi di log dall'eliminazione

Attivazione della protezione dall'eliminazione

È possibile abilitare la protezione dall'eliminazione quando si crea un nuovo gruppo di log o su gruppi di log esistenti. Durante la creazione del gruppo di log, seleziona «Protezione da eliminazione abilitata» o passando il parametro`--deletion-protection-enabled`. Per impostazione predefinita, la protezione da eliminazione non è abilitata.

Per abilitare o disabilitare la protezione da eliminazione su un gruppo di log esistente (console)

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Gestione dei registri.
3. Seleziona il gruppo di log che desideri proteggere.
4. Scegli Azioni, Modifica protezione da eliminazione.
5. Nella finestra di dialogo, rivedi e invia le modifiche.

Se utilizzate il AWS CLI, per abilitare la protezione da eliminazione su un gruppo di log esistente:

```
aws logs put-log-group-deletion-protection \
--log-group-identifier "/my-application/logs" \
--deletion-protection-enabled
```

Per rimuovere la protezione da eliminazione su un gruppo di log esistente:

```
aws logs put-log-group-deletion-protection \
--log-group-identifier "/my-application/logs" \
--no-deletion-protection-enabled
```

Gestione degli errori

Se si tenta di eliminare un gruppo di log con la protezione da eliminazione abilitata, si riceve un messaggio `ValidationException` con il messaggio: «Impossibile eliminare il gruppo di log con la protezione da eliminazione abilitata. Disattiva prima la protezione da eliminazione».

Contrassegna i gruppi di log in Amazon CloudWatch Logs

Puoi assegnare i tuoi metadati ai gruppi di log che crei in Amazon CloudWatch Logs sotto forma di tag. Un tag è una coppia chiave-valore che definisci per un gruppo di log. L'utilizzo dei tag è un modo semplice ma efficace per gestire AWS le risorse e organizzare i dati, compresi i dati di fatturazione.

> **Note**
>
> È possibile utilizzare i tag per controllare l'accesso alle risorse di CloudWatch Logs, inclusi i gruppi di log e le destinazioni. L'accesso ai flussi di log è controllato a livello di gruppo di log per via della relazione gerarchica tra i gruppi di log e i flussi di log. Per ulteriori informazioni sull'utilizzo dei tag per controllare l'accesso alle risorse, consulta Controllo dell'accesso alle risorse di Amazon Web Services utilizzando i tag.

Indice

- Nozioni di base sui tag
- Monitoraggio dei costi mediante l'assegnazione di tag
- Limitazioni applicate ai tag
- Taggare i gruppi di log utilizzando il AWS CLI
- Etichettatura dei gruppi di log utilizzando l' CloudWatch API Logs

Nozioni di base sui tag

È possibile utilizzare AWS CloudFormation l'API AWS CLI, o CloudWatch Logs, per completare le seguenti attività:

- aggiunta di tag a un gruppo di log al momento della creazione;
- aggiunta di tag a un gruppo di log esistente;
- elencazione di tag di un gruppo di log;
- eliminazione di tag da un gruppo di log.

Puoi utilizzare i tag per categorizzare i gruppi di log. Ad esempio, puoi categorizzarli in base a scopo, proprietario o ambiente. Poiché definisci una chiave e un valore per ogni tag, puoi creare un set di

categorie personalizzate per soddisfare esigenze specifiche. Ad esempio, puoi definire un set di tag che consente di monitorare i gruppi di log per proprietario e applicazione associata. Di seguito sono riportati vari esempi di tag:

- Progetto: nome del progetto
- Proprietario: nome
- Scopo: test di carico
- Applicazione: nome dell'applicazione
- Ambiente: produzione

Monitoraggio dei costi mediante l'assegnazione di tag

Puoi utilizzare i tag per classificare e tenere traccia AWS dei costi. Quando applichi tag alle tue AWS risorse, inclusi i gruppi di log, il report sull'allocazione AWS dei costi include l'utilizzo e i costi aggregati per tag. Puoi applicare i tag che rappresentano categorie di business (come centri di costo, nomi di applicazioni o proprietari) per organizzare i costi tra più servizi. Per ulteriori informazioni, consulta Utilizzo dei tag per l'allocazione dei costi ai fini dei report di fatturazione personalizzati nella AWS Billing User Guide (Guida per l'utente di Amazon API Gateway).

Limitazioni applicate ai tag

Ai tag si applicano le limitazioni seguenti.

Limitazioni di base

- Il numero massimo di tag per ogni gruppo di log è 50.
- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Non puoi cambiare o modificare i tag di un gruppo di log eliminato.

Limitazioni applicate alle chiavi di tag

- Ogni chiave di tag deve essere univoca. Se aggiungi un tag con una chiave già in uso, il nuovo tag sovrascrive la coppia chiave-valore esistente.
- Non puoi iniziare una chiave di tag con `aws:` perché questo prefisso è riservato all'uso di. AWS AWS crea tag che iniziano con questo prefisso per tuo conto, ma non puoi modificarli o eliminarli.
- Le chiavi di tag devono avere una lunghezza compresa tra 1 e 128 caratteri Unicode.

- Le chiavi di tag devono contenere i seguenti caratteri: lettere Unicode, cifre, spazio e i seguenti caratteri speciali: `_ . / = + - @`.

Limitazioni applicate ai valori dei tag

- I valori dei tag devono avere una lunghezza compresa tra 0 e 255 caratteri Unicode.
- I valori dei tag possono essere vuoti. In caso contrario, devono contenere i seguenti caratteri: lettere Unicode, cifre, spazio e i seguenti caratteri speciali: `_ . / = + - @`.

Taggare i gruppi di log utilizzando il AWS CLI

Puoi aggiungere, elencare e rimuovere tag tramite AWS CLI. Per alcuni esempi, consultare la seguente documentazione:

create-log-group

Crea un gruppo di log. Puoi opzionalmente aggiungere tag quando al momento della creazione del gruppo di log.

tag-resource

Assegna uno o più tag (coppie chiave-valore) alla risorsa Logs specificata. CloudWatch

list-tags-for-resource

Visualizza i tag associati a una risorsa Logs. CloudWatch

untag-resource

Rimuove uno o più tag dalla risorsa CloudWatch Logs specificata.

Etichettatura dei gruppi di log utilizzando l' CloudWatch API Logs

È possibile aggiungere, elencare e rimuovere tag utilizzando l'API CloudWatch Logs. Per alcuni esempi, consultare la seguente documentazione:

CreateLogGroup

Crea un gruppo di log. Puoi opzionalmente aggiungere tag quando al momento della creazione del gruppo di log.

TagResource

Assegna uno o più tag (coppie chiave-valore) alla risorsa Logs specificata. CloudWatch

ListTagsForResource

Visualizza i tag associati a una risorsa Logs. CloudWatch

UntagResource

Rimuove uno o più tag dalla risorsa CloudWatch Logs specificata.

Crittografa i dati di registro in CloudWatch Logs utilizzando AWS Key Management Service

I dati dei gruppi di log sono sempre crittografati in CloudWatch Logs. Per impostazione predefinita, CloudWatch Logs utilizza la crittografia lato server con la Galois/Counter modalità AES-GCM (Advanced Encryption Standard Mode) a 256 bit per crittografare i dati di registro inattivi. In alternativa, è possibile utilizzare AWS Key Management Service per questa crittografia. In tal caso, la crittografia viene eseguita utilizzando una chiave. AWS KMS L'utilizzo della crittografia AWS KMS è abilitato a livello di gruppo di log, associando una chiave KMS a un gruppo di log, al momento della creazione del gruppo di log o dopo la sua esistenza.

Important

CloudWatch I registri ora supportano il contesto di crittografia, utilizzando `kms:EncryptionContext:aws:logs:arn` come chiave e l'ARN del gruppo di log come valore per quella chiave. Se disponi di gruppi di log già crittografati con una chiave KMS e desideri limitare l'utilizzo della chiave a un singolo account e gruppo di log, è necessario assegnare una nuova chiave KMS che include una condizione nella policy IAM. Per ulteriori informazioni, consulta AWS KMS chiavi e contesto di crittografia.

Important

CloudWatch Ora supporta Logs, `kms:ViaService` che consente ai log di effettuare AWS KMS chiamate per conto dell'utente. Dovresti aggiungerlo ai tuoi ruoli che chiamano CloudWatch Logs nella tua Key Policy o in IAM. Per ulteriori informazioni, vedere kms:. ViaService

Dopo aver associato una chiave KMS a un gruppo di log, tutti i nuovi dati importati per il gruppo di log saranno crittografati tramite questa chiave. Questi dati vengono archiviati in formato crittografato per tutto il periodo di conservazione. CloudWatch I registri decrittografano questi dati ogni volta che vengono richiesti. CloudWatch I log devono disporre delle autorizzazioni per la chiave KMS ogni volta che vengono richiesti dati crittografati.

Se successivamente si dissocia una chiave KMS da un gruppo di log, CloudWatch Logs crittografa i dati appena acquisiti utilizzando il metodo di crittografia predefinito Logs. CloudWatch Tutti i dati precedentemente importati che sono stati crittografati con la chiave KMS rimangono crittografati con la chiave KMS. CloudWatch I log possono ancora restituire quei dati dopo che la chiave KMS è stata dissociata, perché CloudWatch i log possono continuare a fare riferimento alla chiave. Tuttavia, se la chiave viene successivamente disabilitata, CloudWatch Logs non è in grado di leggere i log che sono stati crittografati con quella chiave.

> **Important**
>
> CloudWatch Logs supporta solo chiavi KMS simmetriche. Non utilizzare una chiave asimmetrica per crittografare i dati nei gruppi di log. Per ulteriori informazioni, consulta la sezione relativa all'uso di chiavi simmetriche e asimmetriche.

Limits

- Per eseguire la procedura seguente, devi avere le seguenti autorizzazioni: `kms:CreateKey`, `kms:GetKeyPolicy` e `kms:PutKeyPolicy`.
- Dopo aver associato o dissociato una chiave da un gruppo di log, possono essere necessari fino a cinque minuti per rendere effettiva l'operazione.
- Se CloudWatch revochi l'accesso dei log a una chiave associata o elimini una chiave KMS associata, i dati crittografati in Logs non possono più essere recuperati. CloudWatch
- Non puoi associare una chiave KMS a un gruppo di log esistente utilizzando la console. CloudWatch

Passaggio 1: creare una chiave AWS KMS

Per creare una chiave KMS, utilizza il seguente comando create-key:

```
aws kms create-key
```

L'output contiene l'ID chiave e l'Amazon Resource Name (ARN) della chiave. Di seguito è riportato un output di esempio:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Fase 2: Impostazione delle autorizzazioni sulla chiave KMS

Per impostazione predefinita, tutte AWS KMS le chiavi sono private. Solo il proprietario della risorsa può utilizzarla per crittografare e decrittare i dati. Tuttavia, il proprietario della risorsa può concedere ad altri utenti e risorse le autorizzazioni per accedere alla chiave KMS. Con questo passaggio, concedi al responsabile del servizio CloudWatch Logs e al ruolo chiamante il permesso di utilizzare la chiave. L'entità del servizio deve trovarsi nella stessa AWS regione in cui è archiviata la chiave KMS.

Come procedura ottimale, consigliamo di limitare l'uso della chiave KMS solo agli AWS account o ai gruppi di log specificati.

Innanzitutto, salva la politica predefinita per la tua chiave KMS `policy.json` utilizzando il seguente comando: get-key-policy

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Aprire il file `policy.json` in un editor di testo e aggiungere la sezione in grassetto da una delle seguenti istruzioni. Separare l'istruzione esistente dalla nuova istruzione con una virgola. Queste istruzioni utilizzano `Condition` le sezioni per migliorare la sicurezza della AWS KMS chiave. Per ulteriori informazioni, consulta AWS KMS chiavi e contesto di crittografia.

La sezione `Condition` in questo esempio limita la chiave a un singolo ARN del gruppo di log.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:111122223333:log-group:log-group-name"
                }
            }
        }
```

```
    ]
}
```

La sezione `Condition` di questo esempio limita l'utilizzo di chiave AWS KMS all'account specificato, ma può essere utilizzato per qualsiasi gruppo di log.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:123456789012:*"
                }
            }
        }
```

```
    ]
}
```

Passaggio 3: imposta le autorizzazioni sul principale IAM chiamante

Aggiungi le autorizzazioni al principale IAM (utente o ruolo) che CloudWatch chiamerà Logs. APIs Le autorizzazioni richieste dipendono dalle operazioni che il principale deve eseguire. Puoi aggiungere queste autorizzazioni nella policy AWS KMS chiave o tramite IAM sul ruolo stesso. CloudWatch Registra gli usi `kms:ViaService` per effettuare chiamate per AWS KMS conto del cliente. Per ulteriori informazioni, vedere kms:. ViaService

Autorizzazioni per associare una chiave KMS a un gruppo di log

Il principale IAM che chiama `CreateLogGroup` con uno `kmsKeyId` o più chiamate deve avere `AssociateKmsKey` l'`kms:DescribeKey`autorizzazione sulla chiave KMS specificata. Se il chiamante non dispone di questa autorizzazione, la chiamata API avrà esito negativo con un. `AccessDeniedException`

L'esempio seguente di dichiarazione politica chiave concede le autorizzazioni minime necessarie per associare una chiave KMS a un gruppo di log:

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::account_id:role/role_name"
  },
  "Action": [
    "kms:Describe*"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "logs.region.amazonaws.com"
      ]
    }
  }
}
```

Autorizzazioni per la lettura e la scrittura di dati di registro crittografati

Se il principale IAM deve anche leggere o scrivere dati di registro crittografati (ad esempio, chiamando, `PutLogEvents` `GetLogEventsFilterLogEvents`, o `StartQuery` su un gruppo di log crittografato con una chiave KMS gestita dal cliente), il principale necessita di autorizzazioni aggiuntive AWS KMS . L'esempio seguente di dichiarazione politica chiave concede l'intero set di autorizzazioni necessarie sia per l'associazione di una chiave che per la lettura o la scrittura di dati di registro crittografati:

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::account_id:role/role_name"
  },
  "Action": [
    "kms:Encrypt",
    "kms:ReEncrypt*",
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:Describe*"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "logs.region.amazonaws.com"
      ]
    }
  }
}
```

Come procedura consigliata, estendi la policy solo ai ruoli che interagiranno con AWS KMS i gruppi di log crittografati.

In alternativa, per concedere l'intero set di autorizzazioni sia per l'associazione di una chiave che per la lettura o la scrittura di dati di registro crittografati tramite IAM, puoi aggiungere la seguente policy al ruolo del chiamante. Questa può essere aggiunta a una politica di ruolo esistente o associata a un ruolo come politica separata aggiuntiva. Se si utilizza questo metodo, come best practice è consigliabile applicare la policy solo alle AWS KMS chiavi che verranno utilizzate per la crittografia dei log. Per ulteriori informazioni, consulta Modificare le politiche IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:ReEncrypt*",
                "kms:Decrypt",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "logs.us-east-1.amazonaws.com"
                    ]
                }
            },
            "Resource": "arn:aws:kms:us-east-1:444455556666:key/key_id"
        }
    ]
}
```

Infine, aggiungi la politica aggiornata utilizzando il seguente put-key-policycomando:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Fase 4: Associare una chiave KMS a un gruppo di log

Puoi associare una chiave KMS a un gruppo di log al momento della sua creazione o successivamente.

Per scoprire se a un gruppo di log è già associata una chiave KMS, usa il seguente comando: describe-log-groups

```
aws logs describe-log-groups --log-group-name-prefix "log-group-name-prefix"
```

Se l'output include un campo `kmsKeyId`, il gruppo di log è associato alla chiave visualizzata per il valore di tale campo.

Associazione di una chiave KMS a un gruppo di log al momento della creazione

Utilizza il comando create-log-group come segue:

```
aws logs create-log-group --log-group-name my-log-group --kms-key-id "key-arn"
```

Associazione di una chiave KMS a un gruppo di log esistente

Utilizza il comando associate-kms-key come segue:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id "key-arn"
```

Passo 5: Dissocia la chiave da un gruppo di log

Per dissociare la chiave KMS associata a un gruppo di log, usa il seguente comando: disassociate-kms-key

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

AWS KMS chiavi e contesto di crittografia

Per migliorare la sicurezza delle AWS Key Management Service chiavi e dei gruppi di log crittografati, CloudWatch Logs ora inserisce il gruppo di log ARNs come parte del contesto di crittografia utilizzato per crittografare i dati di registro. Il contesto di crittografia è un insieme di coppie chiave-valore che vengono utilizzate come dati autenticati aggiuntivi. Il contesto di crittografia consente di utilizzare le condizioni delle policy IAM per limitare l'accesso alla AWS KMS chiave per AWS account e gruppo di log. Per ulteriori informazioni, consulta Contesto di crittografia ed Elementi delle policy IAM JSON: condizione.

Si consiglia di utilizzare chiavi KMS diverse per ciascuno dei gruppi di log crittografati.

Se disponi di un gruppo di log crittografato in precedenza e desideri modificare il gruppo di log in modo da utilizzare una nuova chiave KMS dal cliente che funziona solo per tale gruppo di log, completa la seguente procedura.

Conversione di un gruppo di log crittografato in modo da utilizzare una chiave KMS con una policy che lo limita a tale gruppo di log

1. Immettere il seguente comando per trovare l'ARN del chiave corrente del gruppo di log:

   ```
   aws logs describe-log-groups
   ```

   L'output include la seguente riga. Prendere nota dell'ARN. È necessario utilizzarlo nel passaggio 7.

   ```
   ...
   "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
   cdef-0123-456789abcdef"
   ...
   ```

2. Immettere il seguente comando per creare una nuova KMS:

   ```
   aws kms create-key
   ```

3. Immettere il comando seguente per salvare la policy della nuova chiave in un file `policy.json`:

   ```
   aws kms get-key-policy --key-id new-key-id --policy-name default --output text > ./
   policy.json
   ```

4. Utilizzare un editor di testo per aprire `policy.json` e aggiungere un'espressione `Condition` alla policy:

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Id": "key-default-1",
       "Statement": [
           {
               "Sid": "Enable IAM User Permissions",
               "Effect": "Allow",
               "Principal": {
                   "AWS": "arn:aws:iam::111122223333:root"
               },
               "Action": "kms:*",
               "Resource": "*"
   ```

```
        },
        {
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:111122223333:log-group:LOG-GROUP-NAME"
                }
            }
        }
    ]
}
```

5. Immettere il seguente comando per aggiungere la policy aggiornata alla nuova chiave KMS:

```
aws kms put-key-policy --key-id new-key-ARN --policy-name default --policy file://
policy.json
```

6. Immettere il comando seguente per associare la policy al gruppo di log:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id new-key-ARN
```

   CloudWatch I log ora crittografano tutti i nuovi dati utilizzando la nuova chiave.

7. Quindi, revocare tutte le autorizzazioni tranne `Decrypt` dalla vecchia chiave. Innanzitutto, immettere il seguente comando per recuperare la vecchia policy:

```
aws kms get-key-policy --key-id old-key-ARN --policy-name default --output text
 > ./policy.json
```

8. Utilizzare un editor di testo per aprire `policy.json` e rimuovere tutti i valori dall'elenco `Action`, ad eccezione di `kms:Decrypt`

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

9. Immettere il seguente comando per aggiungere la policy aggiornata alla vecchia chiave:

```
aws kms put-key-policy --key-id old-key-ARN --policy-name default --policy file://
policy.json
```

Incremento della protezione dei dati di log sensibili con il mascheramento

Puoi contribuire a proteggere i dati sensibili che vengono acquisiti da CloudWatch Logs utilizzando le politiche di protezione dei dati dei gruppi di log. Queste policy consentono di verificare e mascherare i dati sensibili che appaiono nei log eventi importati dai gruppi di log dell'account.

Quando crei una politica di protezione dei dati, per impostazione predefinita, i dati sensibili che corrispondono agli identificatori di dati che hai selezionato vengono mascherati in tutti i punti di uscita, inclusi CloudWatch Logs Insights, filtri metrici e filtri di abbonamento. Solo gli utenti che dispongono dell'autorizzazione IAM `logs:Unmask` possono visualizzare i dati non mascherati.

Puoi creare una policy di protezione dei dati per tutti i gruppi di log del tuo account e puoi anche crearne una per i singoli gruppi di log. Quando crei una policy per l'intero account, questa si applica sia ai gruppi di log esistenti che a quelli creati successivamente.

Se crei una policy di protezione dei dati per l'intero account e una per un singolo gruppo di log, entrambe le policy si applicano a tale gruppo di log. Tutti gli identificatori di dati gestiti specificati in entrambe le policy vengono verificati e mascherati in tale gruppo di log.

> **Note**
>
> Il mascheramento dei dati sensibili è supportato per i gruppi di log nelle classi di registro Standard e Infrequent Access. Per ulteriori informazioni sulle classi di log, vedere. Classi di registro

Ogni gruppo di log può avere solo una policy di protezione dei dati a livello di gruppo di log, ma tale policy può specificare molti identificatori di dati gestiti da verificare e mascherare. Il limite per una policy di protezione dei dati è di 30.720 caratteri.

> **Important**
>
> I dati sensibili vengono rilevati e mascherati quando vengono importati nel gruppo di log. Quando si imposta una policy di protezione dei dati, i log eventi importati nel gruppo di log prima di quel momento non vengono mascherati.

CloudWatch Logs supporta molti identificatori di dati gestiti, che offrono tipi di dati preconfigurati che è possibile selezionare per proteggere i dati finanziari, le informazioni sanitarie personali (PHI) e le informazioni di identificazione personale (PII). CloudWatch La protezione dei dati di Logs consente di sfruttare modelli di pattern matching e machine learning per rilevare dati sensibili. Per alcuni tipi di identificatori di dati gestiti, il rilevamento dipende anche dalla ricerca di determinate parole chiave in prossimità dei dati sensibili. Puoi anche utilizzare identificatori di dati personalizzati per creare identificatori di dati personalizzati in base al tuo caso d'uso specifico.

Viene emessa una metrica relativa al CloudWatch momento in cui vengono rilevati dati sensibili che corrispondono agli identificatori di dati selezionati. Questa è la LogEventsWithFindingsmetrica e viene emessa nello spazio dei nomi AWS/Logs. Puoi utilizzare questa metrica per creare CloudWatch allarmi e visualizzarla in grafici e dashboard. Le metriche emesse dalla protezione dei dati sono metriche distribuite gratuite. Per ulteriori informazioni sulle metriche a cui invia Logs, consulta CloudWatch . CloudWatch Monitoraggio con CloudWatch metriche

Ogni identificatore di dati gestito è progettato per rilevare un tipo specifico di dati sensibili, come numeri di carte di credito, chiavi di accesso AWS segrete o numeri di passaporto per un determinato paese o area geografica. Quando crei una policy di protezione dei dati, puoi configurarla in modo che utilizzi questi identificatori per analizzare i log importati dal gruppo di log ed esegua operazioni specifiche quando tali dati vengono rilevati.

CloudWatch La protezione dei dati dei registri è in grado di rilevare le seguenti categorie di dati sensibili utilizzando identificatori di dati gestiti:

- Credenziali, come chiavi private o AWS chiavi di accesso segrete
- Informazioni finanziarie, ad esempio i numeri di carte di credito
- Informazioni personali di identificazione (PII), ad esempio patenti di guida o codici fiscali
- Dati sanitari protetti (PHI), ad esempio numeri di identificazione medica e assistenza sanitaria
- Identificatori di dispositivo, ad esempio indirizzi IP o indirizzi MAC

Per informazioni dettagliate sui tipi di dati che puoi proteggere, consulta la sezione Tipi di dati che è possibile proteggere.

Indice

- Informazioni sulle policy di protezione dei dati
  - Cosa sono le policy di protezione dei dati?
  - Come è strutturata una policy di protezione dei dati?
    - Proprietà JSON per la policy di protezione dei dati
    - Proprietà JSON per una dichiarazione di policy
    - Proprietà JSON per un'operazione della dichiarazione di policy
- Autorizzazioni IAM necessarie per la creazione o l'uso di una policy di protezione dei dati
  - Autorizzazioni necessarie per le policy di protezione dei dati a livello di account
  - Autorizzazioni necessarie per le policy di protezione dei dati per un singolo gruppo di log

- Policy di protezione dei dati di esempio
- Creazione di una policy di protezione dei dati a livello di account
  - Console
  - AWS CLI
    - Sintassi della politica di protezione dei dati per le AWS CLI nostre operazioni API
- Creazione di una policy di protezione dei dati per un singolo gruppo di log
  - Console
  - AWS CLI
    - Sintassi della politica di protezione dei dati per le AWS CLI nostre operazioni API
- Visualizzazione di dati senza mascheramento
- Report sui risultati della verifica
  - Politica chiave richiesta per inviare i risultati dell'audit a un bucket protetto da AWS KMS
- Tipi di dati che è possibile proteggere
  - CloudWatch Registra gli identificatori di dati gestiti per tipi di dati sensibili
    - Credenziali
      - Identificatore di dati ARNs per i tipi di dati relativi alle credenziali
    - Identificatori di dispositivo
      - Identificatore di dati ARNs per i tipi di dati del dispositivo
    - Informazioni finanziarie
      - Identificatore ARNs di dati per tipi di dati finanziari
    - Dati sanitari protetti (PHI)
      - Identificatore di dati ARNs per i tipi di dati relativi alle informazioni sanitarie protette (PHI)
    - Informazioni personali di identificazione (PII)
      - Parole chiave per i numeri identificativi delle patenti di guida
      - Parole chiave per i numeri di carta d'identità
      - Parole chiave per i numeri di passaporto
      - Parole chiave per i numeri di identificazione dei contribuenti e i numeri di riferimento
      - Identificatore di dati ARNs per informazioni di identificazione personale (PII)
    - Identificatori di dati personalizzati

- Cosa sono gli identificatori di dati personalizzati?
- Vincoli degli identificatori di dati personalizzati
- Utilizzo di identificatori di dati personalizzati nella console
- Utilizzo degli identificatori di dati personalizzati nella policy di protezione dei dati

Informazioni sulle policy di protezione dei dati

Argomenti

- Cosa sono le policy di protezione dei dati?
- Come è strutturata una policy di protezione dei dati?

Cosa sono le policy di protezione dei dati?

CloudWatch Logs utilizza le politiche di protezione dei dati per selezionare i dati sensibili da scansionare e le azioni da intraprendere per proteggere tali dati. Per selezionare i dati sensibili di interesse, si utilizzano identificatori di dati. CloudWatch Registra la protezione dei dati, quindi rileva i dati sensibili utilizzando l'apprendimento automatico e il pattern matching. Per agire sugli identificatori di dati trovati, è possibile definire operazioni di audit (verifica) e de-identify (deidentifica). Queste operazioni consentono di registrare i dati sensibili trovati (o non trovati) e di mascherare i dati sensibili quando vengono visualizzati i log eventi.

Come è strutturata una policy di protezione dei dati?

Come illustrato nella figura riportata di seguito, un documento relativo alla policy di protezione dei dati include questi elementi:

- Informazioni opzionali sulla policy nella parte superiore del documento
- Una dichiarazione che definisce le azioni di audit e di deidentifica

È possibile definire una sola politica di protezione dei dati per gruppo di CloudWatch log Logs. La policy di protezione dei dati può includere una o più dichiarazioni di rifiuto o deidentificazione, ma solo una dichiarazione di verifica.

Proprietà JSON per la policy di protezione dei dati

Una policy di protezione dei dati richiede le seguenti informazioni di base ai fini dell'identificazione:

- Name (Nome): nome della policy.
- Description (Descrizione): (facoltativo) la descrizione della policy.
- Version (Versione): la versione del linguaggio della policy. La versione corrente è 2021-06-01.
- Statement (Dichiarazione): l'elenco di dichiarazioni che specificano le operazioni della policy di protezione dei dati.

```
{
  "Name": "CloudWatchLogs-PersonalInformation-Protection",
  "Description": "Protect basic types of sensitive data",
  "Version": "2021-06-01",
  "Statement": [
        ...
  ]
}
```

Proprietà JSON per una dichiarazione di policy

Una dichiarazione di policy definisce il contesto di rilevamento per l'operazione di protezione dei dati.

- Sid: (facoltativo) l'identificatore della dichiarazione.
- DataIdentifier— I dati sensibili per i quali CloudWatch Logs deve eseguire la scansione. Ad esempio, nome, indirizzo o numero di telefono.
- Funzionamento: le azioni successive, Audit o De-identify. CloudWatch Logs esegue queste azioni quando rileva dati sensibili.

```
{
  ...
  "Statement": [
    {
      "Sid": "audit-policy",
      "DataIdentifier": [
        "arn:aws:dataprotection::aws:data-identifier/Address"
      ],
      "Operation": {
        "Audit": {
          "FindingsDestination": {}
        }
      }
```

```
    },
```

Proprietà JSON per un'operazione della dichiarazione di policy

Una dichiarazione di policy definisce una delle seguenti operazioni di protezione dei dati.

- Verifica: emette metriche e report sui risultati senza interrompere la registrazione. Le stringhe che corrispondono incrementano la LogEventsWithFindingsmetrica che CloudWatch Logs pubblica nello spazio dei nomi AWS/Logs in. CloudWatch Puoi utilizzare queste metriche per creare allarmi.

  Per un esempio di un report sui risultati della verifica, consulta Report sui risultati della verifica.

  Per ulteriori informazioni sulle metriche a cui invia Logs, consulta. CloudWatch CloudWatch Monitoraggio con CloudWatch metriche
- Deidentifica: maschera i dati sensibili senza interrompere la registrazione.

Autorizzazioni IAM necessarie per la creazione o l'uso di una policy di protezione dei dati

Per poter utilizzare le policy di protezione dei dati per i gruppi di log, è necessario disporre di determinate autorizzazioni, come mostrato nelle tabelle seguenti. Le autorizzazioni sono diverse per le policy di protezione dei dati a livello di account e per quelle che si applicano a un singolo gruppo di log.

Autorizzazioni necessarie per le policy di protezione dei dati a livello di account

> **Note**
>
> Se si esegue una di queste operazioni all'interno di una funzione Lambda, il ruolo di esecuzione Lambda e il limite delle autorizzazioni devono includere anche le seguenti autorizzazioni.

Operation	Autorizzazione IAM necessaria	Risorsa
Crea una policy di protezione dei dati senza destinazioni di verifica	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
Crea una politica di protezione dei dati con CloudWatch Logs come destinazione di controllo	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`logs:PutResourcePolicy`	*
	`logs:DescribeResourcePolicies`	*
	`logs:DescribeLogGroups`	*
Crea una politica di protezione dei dati con Firehose come destinazione di controllo	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`firehose:TagDeliveryStream`	`arn:aws:logs:::deliverystre`

Operation	Autorizzazione IAM necessaria	Risorsa
		am/ *YOUR_DELIVERY_STREAM*
Crea una policy di protezione dei dati con Amazon S3 come destinazione di verifica	logs:PutAccountPolicy	*
	logs:PutDataProtectionPolicy	*
	logs:CreateLogDelivery	*
	s3:GetBucketPolicy	arn:aws:s3::: *YOUR_BUCKET*
	s3:PutBucketPolicy	arn:aws:s3::: *YOUR_BUCKET*
Smaschera i log eventi mascherati in un gruppo di log specificato	logs:Unmask	arn:aws:logs:::log-group:*
Visualizza una policy di protezione dei dati esistente	logs:GetDataProtectionPolicy	*
Elimina una policy di protezione dei dati	logs:DeleteAccountPolicy	*
	logs:DeleteDataProtectionPolicy	*

Se i log di controllo della protezione dei dati sono già stati inviati a una destinazione, le altre policy che inviano i log alla stessa destinazione richiedono solo le autorizzazioni `logs:PutDataProtectionPolicy` e `logs:CreateLogDelivery`.

Autorizzazioni necessarie per le policy di protezione dei dati per un singolo gruppo di log

> **Note**
>
> Se si esegue una di queste operazioni all'interno di una funzione Lambda, il ruolo di esecuzione Lambda e il limite delle autorizzazioni devono includere anche le seguenti autorizzazioni.

Operation	Autorizzazione IAM necessaria	Risorsa
Crea una policy di protezione dei dati senza destinazioni di verifica	`logs:PutDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Crea una politica di protezione dei dati con CloudWatch Logs come destinazione di controllo	`logs:PutDataProtectionPolicy`
	`logs:CreateLogDelivery`
	`logs:PutResourcePolicy`
	`logs:DescribeResourcePolicies`
	`logs:DescribeLogGroups`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		`*`
		`*`
		`*`

Crea una politica di protezione dei dati con Firehose come destinazione di controllo	`logs:PutDataProtectionPolicy` `logs:CreateLogDelivery`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*` `*`

Operation	Autorizzazione IAM necessaria	Risorsa
	`firehose:TagDeliveryStream`	`arn:aws:logs:::deliverystream/` *YOUR_DELIVERY_STREAM*

Creazione di una policy di protezione dei dati con Amazon S3 come destinazione di controllo	`logs:PutDataProtectionPolicy`
	
	`logs:CreateLogDelivery`
	
	`s3:GetBucketPolicy`
	
	`s3:PutBucketPolicy`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		
		`*`
		
		`arn:aws:s3:::` *YOUR_BUCKET*
		
		`arn:aws:s3:::` *YOUR_BUCKET*

Smascheramento di eventi di log mascherati	`logs:Unmask`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
Visualizzazione di una policy di protezione dei dati esistente	`logs:GetDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
Elimina una policy di protezione dei dati	`logs:DeleteDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Se i log di controllo della protezione dei dati sono già stati inviati a una destinazione, le altre policy che inviano i log alla stessa destinazione richiedono solo le autorizzazioni `logs:PutDataProtectionPolicy` e `logs:CreateLogDelivery`.

Policy di protezione dei dati di esempio

La seguente policy di esempio consente a un utente di creare, visualizzare ed eliminare policy di protezione dei dati in grado di inviare i risultati del controllo a tutti e tre i tipi di destinazioni di controllo. Non consente all'utente di visualizzare dati non mascherati.

JSON

```
    {
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogDeliveryConfiguration",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:DescribeLogGroups",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowDataProtectionAndBucketConfiguration",
            "Effect": "Allow",
            "Action": [
                "logs:GetDataProtectionPolicy",
                "logs:DeleteDataProtectionPolicy",
                "logs:PutDataProtectionPolicy",
                "s3:PutBucketPolicy",
                "firehose:TagDeliveryStream",
                "s3:GetBucketPolicy"
            ],
            "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/delivery-
stream-name",
            "arn:aws:s3:::amzn-s3-demo-destination-bucket",
            "arn:aws:logs:us-east-1:111122223333:log-group:log-group-name:*"
            ]
        }
    ]
```

```
}
```

Creazione di una policy di protezione dei dati a livello di account

Puoi utilizzare la console o i AWS CLI comandi CloudWatch Logs per creare una politica di protezione dei dati per mascherare i dati sensibili per tutti i gruppi di log del tuo account. Questa operazione ha effetto sia sui gruppi di log correnti che su quelli creati successivamente.

> **Important**
>
> I dati sensibili vengono rilevati e mascherati quando vengono importati nel gruppo di log. Quando si imposta una policy di protezione dei dati, i log eventi importati nel gruppo di log prima di quel momento non vengono mascherati.

Argomenti

- Console
- AWS CLI

Console

Utilizzo della console per creare una policy di protezione dei dati a livello di account

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Impostazioni. Si trova quasi in fondo all'elenco.
3. Scegliere la scheda Log.
4. Scegli Configura.
5. Per gli identificatori di dati gestiti, seleziona i tipi di dati che desideri controllare e mascherare per tutti i tuoi gruppi di log. Per individuare gli identificatori che ti interessano, digita il testo nella casella di selezione.

   Ti consigliamo di selezionare solo gli identificatori di dati pertinenti per i tuoi dati di log e la tua attività. La scelta di numerosi tipi di dati può portare a falsi positivi.

   Per informazioni dettagliate sui tipi di dati che puoi proteggere, consulta la sezione Tipi di dati che è possibile proteggere.

6. (Facoltativo) Se desideri controllare e mascherare altri tipi di dati utilizzando identificatori di dati personalizzati, scegli Aggiungi identificatore di dati personalizzato. Quindi inserisci un nome per il tipo di dati e l'espressione regolare da utilizzare per cercare quel tipo di dati nel registro degli eventi. Per ulteriori informazioni, consulta Identificatori di dati personalizzati.

   Una singola politica di protezione dei dati può includere fino a 10 identificatori di dati personalizzati. Ogni espressione regolare che definisce un identificatore di dati personalizzato deve contenere al massimo 200 caratteri.

7. (Facoltativo) Scegli uno o più servizi a cui inviare i risultati della verifica. Anche se scegli di non inviare i risultati della verifica ad alcun servizio, i tipi di dati sensibili selezionati verranno comunque mascherati.

8. Scegli Activate data protection (Attiva la protezione dei dati).

AWS CLI

Da utilizzare per AWS CLI creare una politica di protezione dei dati

1. Utilizza un editor di testo per creare un file di policy denominato `DataProtectionPolicy.json`. Per informazioni sulla sintassi delle policy, consulta la sezione seguente.

2. Immetti il comando seguente:

```
aws logs put-account-policy \
--policy-name TEST_POLICY --policy-type "DATA_PROTECTION_POLICY" \
--policy-document file://policy.json \
--scope "ALL" \
--region us-west-2
```

Sintassi della politica di protezione dei dati per le AWS CLI nostre operazioni API

Quando crei una policy di protezione dei dati JSON da utilizzare in un AWS CLI comando o in un'operazione API, la policy deve includere due blocchi JSON:

- Il primo blocco deve includere sia una matrice `DataIdentifer` sia una proprietà `Operation` con un'operazione `Audit`. La matrice `DataIdentifer` elenca i tipi di dati sensibili che desideri mascherare. Per ulteriori informazioni sulle opzioni disponibili, consulta Tipi di dati che è possibile proteggere.

La proprietà `Operation` con l'operazione `Audit` è necessaria per individuare i termini relativi ai dati sensibili. L'operazione `Audit` deve contenere un oggetto `FindingsDestination`. È possibile utilizzare tale oggetto `FindingsDestination` per elencare una o più destinazioni a cui inviare il report sui risultati della verifica. Se specifichi destinazioni come gruppi di log, flussi Amazon Data Firehose e bucket S3, devono già esistere. Per un esempio di report sui risultati della verifica, consulta Report sui risultati della verifica.

- Il secondo blocco deve includere sia una matrice `DataIdentifer` sia una proprietà `Operation` con un'operazione `Deidentify`. La matrice `DataIdentifer` deve corrispondere esattamente alla matrice `DataIdentifer` nel primo blocco della policy.

  La proprietà `Operation` con l'operazione `Deidentify` è ciò che maschera effettivamente i dati e deve contenere l'oggetto `"MaskConfig": {}`. L'oggetto `"MaskConfig": {}` deve essere vuoto.

Di seguito è riportato un esempio di politica di protezione dei dati che utilizza solo identificatori di dati gestiti. Questa politica maschera gli indirizzi e-mail e le patenti di guida degli Stati Uniti d'America.

Per informazioni sulle politiche che specificano identificatori di dati personalizzati, consulta. Utilizzo degli identificatori di dati personalizzati nella policy di protezione dei dati

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
```

```
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Creazione di una policy di protezione dei dati per un singolo gruppo di log

È possibile utilizzare la console o i AWS CLI comandi CloudWatch Logs per creare una policy di protezione dei dati per mascherare i dati sensibili.

È possibile assegnare una policy di protezione dei dati a ciascun gruppo di log. Ogni policy di protezione dei dati può verificare diversi tipi di informazioni. Ogni policy di protezione dei dati può includere un'istruzione di verifica.

Argomenti

- Console
- AWS CLI

Console

Utilizzo della console per creare una policy di protezione dei dati

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel pannello di navigazione a sinistra, scegli Log, Gruppi di log.
3. Scegli il nome del gruppo di log.
4. Scegli Actions (Operazioni), quindi scegli Create data protection policy (Crea policy di protezione dei dati).
5. Per gli identificatori di dati gestiti, seleziona i tipi di dati che desideri controllare e mascherare in questo gruppo di log. Per individuare gli identificatori che ti interessano, digita il testo nella casella di selezione.

   Ti consigliamo di selezionare solo gli identificatori di dati pertinenti per i tuoi dati di log e la tua attività. La scelta di numerosi tipi di dati può portare a falsi positivi.

   Per informazioni dettagliate sui tipi di dati che è possibile proteggere utilizzando identificatori di dati gestiti, consulta. Tipi di dati che è possibile proteggere
6. (Facoltativo) Se desideri controllare e mascherare altri tipi di dati utilizzando identificatori di dati personalizzati, scegli Aggiungi identificatore di dati personalizzato. Quindi inserisci un nome per il tipo di dati e l'espressione regolare da utilizzare per cercare quel tipo di dati nel registro degli eventi. Per ulteriori informazioni, consulta Identificatori di dati personalizzati.

   Una singola politica di protezione dei dati può includere fino a 10 identificatori di dati personalizzati. Ogni espressione regolare che definisce un identificatore di dati personalizzato deve contenere al massimo 200 caratteri.
7. (Facoltativo) Scegli uno o più servizi a cui inviare i risultati della verifica. Anche se scegli di non inviare i risultati della verifica ad alcun servizio, i tipi di dati sensibili selezionati verranno comunque mascherati.
8. Scegli Activate data protection (Attiva la protezione dei dati).

AWS CLI

Da utilizzare per AWS CLI creare una politica di protezione dei dati

1. Utilizza un editor di testo per creare un file di policy denominato `DataProtectionPolicy.json`. Per informazioni sulla sintassi delle policy, consulta la sezione seguente.
2. Immetti il comando seguente:

```
aws logs put-data-protection-policy --log-group-identifier "my-log-group" --policy-
document file:///Path/DataProtectionPolicy.json --region us-west-2
```

Sintassi della politica di protezione dei dati per le AWS CLI nostre operazioni API

Quando crei una policy di protezione dei dati JSON da utilizzare in un AWS CLI comando o in un'operazione API, la policy deve includere due blocchi JSON:

- Il primo blocco deve includere sia una matrice `DataIdentifer` sia una proprietà `Operation` con un'operazione `Audit`. La matrice `DataIdentifer` elenca i tipi di dati sensibili che desideri mascherare. Per ulteriori informazioni sulle opzioni disponibili, consulta Tipi di dati che è possibile proteggere.

  La proprietà `Operation` con l'operazione `Audit` è necessaria per individuare i termini relativi ai dati sensibili. L'operazione `Audit` deve contenere un oggetto `FindingsDestination`. È possibile utilizzare tale oggetto `FindingsDestination` per elencare una o più destinazioni a cui inviare il report sui risultati della verifica. Se specifichi destinazioni come gruppi di log, flussi Amazon Data Firehose e bucket S3, devono già esistere. Per un esempio di report sui risultati della verifica, consulta Report sui risultati della verifica.
- Il secondo blocco deve includere sia una matrice `DataIdentifer` sia una proprietà `Operation` con un'operazione `Deidentify`. La matrice `DataIdentifer` deve corrispondere esattamente alla matrice `DataIdentifer` nel primo blocco della policy.

  La proprietà `Operation` con l'operazione `Deidentify` è ciò che maschera effettivamente i dati e deve contenere l'oggetto `"MaskConfig": {}`. L'oggetto `"MaskConfig": {}` deve essere vuoto.

Quello che segue è un esempio di policy di protezione dei dati che maschera gli indirizzi e-mail e le patenti di guida degli Stati Uniti.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
```

```
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Visualizzazione di dati senza mascheramento

Per visualizzare i dati senza mascheramento, un utente deve disporre dell'autorizzazione `logs:Unmask`. Gli utenti che dispongono di questa autorizzazione possono visualizzare i dati senza mascheramento nei seguenti modi:

- Quando visualizzi gli eventi in un flusso di log, scegli Display (Visualizza), Unmask (Rimuovi mascheramento).

- Utilizza una query CloudWatch Logs Insights che include il comando unmask (@message). La seguente query di esempio mostra i 20 log eventi più recenti nel flusso senza mascheramento:

```
fields @timestamp, @message, unmask(@message)
| sort @timestamp desc
| limit 20
```

  Per ulteriori informazioni sui comandi CloudWatch Logs Insights, vedere. CloudWatch Sintassi delle interrogazioni in linguaggio Logs Insights
- Utilizzare un' FilterLogEventsoperazione GetLogEventsor con il `unmask` parametro.

La CloudWatchLogsFullAccesspolitica include l'`logs:Unmask`autorizzazione. Per concedere `logs:Unmask` a un utente che non lo ha CloudWatchLogsFullAccess, puoi allegare una policy IAM personalizzata a quell'utente. Per ulteriori informazioni, consulta la sezione Adding permissions to a user (console) (Aggiunta di autorizzazioni a un utente [console]).

Report sui risultati della verifica

Se configuri le politiche di controllo della protezione dei dati di CloudWatch Logs per scrivere report di controllo su CloudWatch Logs, Amazon S3 o Firehose, questi report sui risultati sono simili all'esempio seguente. CloudWatch Logs scrive un rapporto sui risultati per ogni evento di registro che contiene dati sensibili.

```
{
    "auditTimestamp": "2023-01-23T21:11:20Z",
    "resourceArn": "arn:aws:logs:us-west-2:111122223333:log-group:/aws/lambda/
MyLogGroup:*",
    "dataIdentifiers": [
        {
            "name": "EmailAddress",
            "count": 2,
            "detections": [
                {
                    "start": 13,
                    "end": 26
                },
{
                    "start": 30,
                    "end": 43
                }
```

```
            ]
        }
    ]
}
```

I campi del report sono i seguenti:

- Il campo `resourceArn` visualizza il gruppo di log in cui sono stati trovati i dati sensibili.
- L'oggetto `dataIdentifiers` visualizza le informazioni sui risultati di un tipo di dati sensibili in corso di verifica.
- Il campo `name` identifica il tipo di dati sensibili riportati in questa sezione.
- Il campo `count` visualizza il numero di volte in cui questo tipo di dati sensibili viene visualizzato nel log eventi.
- I campi `start` e `end` mostrano dove appare ogni occorrenza dei dati sensibili nel log eventi in base al numero di caratteri.

L'esempio precedente mostra un report sulla ricerca di due indirizzi e-mail in un log eventi. Il primo indirizzo e-mail inizia al 13° carattere del log eventi e termina al 26° carattere. Il secondo indirizzo e-mail va dal 30° al 43° carattere. Anche se questo log eventi ha due indirizzi e-mail, il valore della metrica `LogEventsWithFindings` viene incrementato solo di uno, poiché tale metrica conta il numero di log eventi che contengono dati sensibili, non il numero di occorrenze dei dati sensibili.

Politica chiave richiesta per inviare i risultati dell'audit a un bucket protetto da AWS KMS

Puoi proteggere i dati in un bucket Amazon S3 abilitando la crittografia lato server con Amazon S3 Managed Keys (SSE-S3) o la crittografia lato server con chiavi KMS (SSE-KMS). Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server nella Guida per l'utente di Amazon S3.

Se si sceglie di inviare gli esiti dell'audit a un bucket protetto con SSE-S3, non è richiesta alcuna configurazione aggiuntiva. Amazon S3 gestisce la chiave di crittografia.

Se gli esiti dell'audit vengono inviati a un bucket protetto con SSE-KMS è necessario aggiornare la policy della chiave per la chiave KMS in modo che l'account di distribuzione dei log possa scrivere nel bucket S3. Per ulteriori informazioni sulla politica chiave richiesta per l'uso con SSE-KMS, consulta Amazon S3 la Amazon CloudWatch Logs User Guide.

Tipi di dati che è possibile proteggere

Questa sezione contiene informazioni sui tipi di dati che è possibile proteggere in una politica di protezione dei dati di CloudWatch Logs. CloudWatch Gli identificatori di dati gestiti da Logs offrono tipi di dati preconfigurati per proteggere i dati finanziari, le informazioni sanitarie personali (PHI) e le informazioni di identificazione personale (PII). Puoi anche utilizzare identificatori di dati personalizzati per creare identificatori di dati personalizzati in base al tuo caso d'uso specifico.

Indice

- CloudWatch Registra gli identificatori di dati gestiti per tipi di dati sensibili
  - Credenziali
    - Identificatore di dati ARNs per i tipi di dati relativi alle credenziali
  - Identificatori di dispositivo
    - Identificatore di dati ARNs per i tipi di dati del dispositivo
  - Informazioni finanziarie
    - Identificatore ARNs di dati per tipi di dati finanziari
  - Dati sanitari protetti (PHI)
    - Identificatore di dati ARNs per i tipi di dati relativi alle informazioni sanitarie protette (PHI)
  - Informazioni personali di identificazione (PII)
    - Parole chiave per i numeri identificativi delle patenti di guida
    - Parole chiave per i numeri di carta d'identità
    - Parole chiave per i numeri di passaporto
    - Parole chiave per i numeri di identificazione dei contribuenti e i numeri di riferimento
    - Identificatore di dati ARNs per informazioni di identificazione personale (PII)
- Identificatori di dati personalizzati
  - Cosa sono gli identificatori di dati personalizzati?
  - Vincoli degli identificatori di dati personalizzati
  - Utilizzo di identificatori di dati personalizzati nella console
  - Utilizzo degli identificatori di dati personalizzati nella policy di protezione dei dati

CloudWatch Registra gli identificatori di dati gestiti per tipi di dati sensibili

Questa sezione contiene informazioni sui tipi di dati che è possibile proteggere utilizzando identificatori di dati gestiti e sui paesi e le aree geografiche pertinenti per ciascuno di questi tipi di dati.

Per alcuni tipi di dati sensibili, CloudWatch Logs data protection analizza le parole chiave in prossimità dei dati e trova una corrispondenza solo se trova quella parola chiave. Se una parola chiave deve trovarsi in prossimità di un particolare tipo di dati, in genere deve trovarsi entro 30 caratteri (inclusi) dai dati.

Se una parola chiave contiene uno spazio, la protezione dei dati di CloudWatch Logs corrisponde automaticamente alle varianti delle parole chiave che non contengono lo spazio o che contengono un carattere di sottolineatura (_) o un trattino (-) anziché lo spazio. In alcuni casi, CloudWatch Logs amplia o abbrevia anche una parola chiave per rispondere alle varianti più comuni della parola chiave.

Nelle tabelle seguenti sono elencati i tipi di credenziali, dispositivi, informazioni finanziarie, mediche e informazioni sanitarie protette (PHI) che CloudWatch Logs è in grado di rilevare utilizzando identificatori di dati gestiti. Questi dati si aggiungono a determinati tipi di dati che potrebbero anche essere considerati informazioni personali di identificazione (PII).

Identificatori supportati indipendenti dalla lingua e dall'area geografica

Identificatore	Categoria
`Address`	Personale
`AwsSecretKey`	Credenziali
`CreditCardExpiration`	Servizi finanziari
`CreditCardNumber`	Servizi finanziari
`CreditCardSecurityCode`	Servizi finanziari
`EmailAddress`	Personale
`IpAddress`	Personale
`LatLong`	Personale

Identificatore	Categoria
`Name`	Personale
`OpenSshPrivateKey`	Credenziali
`PgpPrivateKey`	Credenziali
`PkcsPrivateKey`	Credenziali
`PuttyPrivateKey`	Credenziali
`VehicleIdentificationNumber`	Personale

Gli identificatori di dati dipendenti dall'area geografica prevedono il nome dell'identificatore seguito da un trattino e dai codici a due lettere (ISO 3166-1 alpha-2). Ad esempio, `DriversLicense-US`.

Identificatori supportati che devono includere un codice di Paese o area geografica a due lettere

Identificatore	Categoria	Paesi e lingue
BankAccountNumber	Servizi finanziari	DE, ES, FR, GB, IT, US
CepCode	Personale	BR
Cnpj	Personale	BR
CpfCode	Personale	BR
DriversLicense	Personale	AT, AU, BE, BG, CA, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IT, LT, LU, LV, MT, NL, PL, PT, RO, SE, SI, SK, US
DrugEnforcementAgencyNumber	Integrità	US
ElectoralRollNumber	Personale	GB

Identificatore	Categoria	Paesi e lingue
HealthInsuranceCardNumber	Integrità	UE
HealthInsuranceClaimNumber	Integrità	US
HealthInsuranceNumber	Integrità	FR
HealthcareProcedureCode	Integrità	US
IndividualTaxIdentification Number	Personale	US
InseeCode	Personale	FR
MedicareBeneficiaryNumber	Integrità	US
NationalDrugCode	Integrità	US
NationalIdentificationNumber	Personale	DE, ES, IT
NationalInsuranceNumber	Personale	GB
NationalProviderId	Integrità	US
NhsNumber	Integrità	GB
NieNumber	Personale	ES
NifNumber	Personale	ES
PassportNumber	Personale	CA, DE, ES, FR, GB, IT, US
PermanentResidenceNumber	Personale	CA
PersonalHealthNumber	Integrità	CA
PhoneNumber	Personale	BR, DE, ES, FR, GB, IT, US
PostalCode	Personale	CA
RgNumber	Personale	BR

Identificatore	Categoria	Paesi e lingue
SocialInsuranceNumber	Personale	CA
Ssn	Personale	ES, US
TaxId	Personale	DE, ES, FR, GB
ZipCode	Personale	US

Credenziali

CloudWatch I registri di protezione dei dati possono trovare i seguenti tipi di credenziali.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni
AWS chiave di accesso segreta	`AwsSecretKey`	`aws_secret_access_key` , `credentials` , `secret access key`, `secret key`, `set-awscredential`	Tutti
Chiave privata OpenSSH	`OpenSSHPrivateKey`	Nessuno	Tutti
Chiave privata PGP	`PgpPrivateKey`	Nessuno	Tutti
Chiave privata Pkcs	`PkcsPrivateKey`	Nessuno	Tutti
Chiave privata PuTTY	`PuttyPrivateKey`	Nessuno	Tutti

Identificatore di dati ARNs per i tipi di dati relativi alle credenziali

Di seguito sono elencati gli Amazon Resource Names (ARNs) per gli identificatori di dati che puoi aggiungere alle tue politiche di protezione dei dati.

Identificatore di dati di credenziali ARNs
`arn:aws:dataprotection::aws:data-identifier/AwsSecretKey`
`arn:aws:dataprotection::aws:data-identifier/OpenSshPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PgpPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PkcsPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PuttyPrivateKey`

Identificatori di dispositivo

CloudWatch La protezione dei dati dei registri può trovare i seguenti tipi di identificatori di dispositivo.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e regioni
IP address (Indirizzo IP)	`IpAddress`	Nessuno	Tutti

Identificatore di dati ARNs per i tipi di dati del dispositivo

Di seguito sono elencati gli Amazon Resource Names (ARNs) per gli identificatori di dati che puoi aggiungere alle tue politiche di protezione dei dati.

ARN di identificatore dei dati del dispositivo
`arn:aws:dataprotection::aws:data-identifier/IpAddress`

Informazioni finanziarie

CloudWatch Logs Data Protection consente di trovare i seguenti tipi di informazioni finanziarie.

Se imposti una politica di protezione dei dati, CloudWatch Logs cerca gli identificatori di dati specificati indipendentemente dalla geolocalizzazione in cui si trova il gruppo di log. Le informazioni nella colonna Paesi e aree geografiche di questa tabella indicano se è necessario aggiungere i codici

Paese a due lettere all'identificatore di dati per individuare le parole chiave appropriate per tali Paesi e aree geografiche.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero del conto bancario	`BankAccountNumber`	Sì. A Paesi diversi si applicano parole chiave diverse. Per maggiori dettagli, consulta la tabella Parole chiave per i numeri di conto bancario più avanti in questa sezione.	Francia, Germani Italia, Spagna, Regno Unito, Stati Uniti d'Americ	Include numeri di conto bancario internazionali (IBANs) composti da un massimo di 34 caratteri alfanumerici, inclusi elementi come i codici dei paesi.
Data di scadenza della carta di credito	`CreditCardExpiration`	`exp d`, `exp m`, `exp y`, `expiration` , `expiry`	Tutti	
Numero di carta di credito	`CreditCardNumber`	`account number`, `american express`, `amex`, `bank card`, `card`, `card number`, `card`	Tutti	Il rilevamento richiede

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
		num, cc #, ccn, check card, credit, credit card#, dankort, debit, debit card, diners club, discover, electron, japanese card bureau, jcb, mastercard , mc, pan, payment account number, payment card number, pcn, union pay, visa		che i dati siano una sequenza di 13-19 cifre che rispetti la formula Luhn check e utilizzi un prefisso numerico di carta standard per uno dei seguenti tipi di carte di credito: American Express, Dankort,

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
				Diner's Club, Discover, Electron, Japanese Card Bureau (JCB), Mastercard e Visa. UnionPay
Codice di verifica della carta di credito	`CreditCardSecurityCode`	`card id`, `card identification code`, `card identification number` , `card security code`, `card validation code` , `card validation number` , `card verification data` , `card verification value`, `cvc`, `cvc2`, `cvv`, `cvv2`, `elo verification code`	Tutti	

Parole chiave per i numeri di conto bancario

Utilizza le seguenti parole chiave per i numeri di conto bancario. Sono inclusi i numeri di conto bancario internazionali (IBANs) composti da un massimo di 34 caratteri alfanumerici, inclusi elementi come i codici dei paesi.

Paese	Parole chiave
Francia	account code, account number, accountno# , accountnumber# , bban, code bancaire, compte bancaire, customer account id, customer account number, customer bank account id, iban, numéro de compte
Germania	account code, account number, accountno# , accountnumber# , bankleitzahl , bban, customer account id, customer account number, customer bank account id, geheimzahl , iban, kartennummer , kontonummer , kreditkartennummer , sepa
Italia	account code, account number, accountno# , accountnumber# , bban, codice bancario, conto bancario, customer account id, customer account number, customer bank account id, iban, numero di conto
Spagna	account code, account number, accountno# , accountnumber# , bban, código cuenta, código cuenta bancaria, cuenta cliente id, customer account ID, customer account number, customer bank account id, iban, número cuenta bancaria cliente, número cuenta cliente
Regno Unito	account code, account number, accountno# , accountnumber# , bban, customer account ID, customer account number, customer bank account id, iban, sepa
Stati Uniti	bank account, bank acct, checking account, checking acct, deposit account, deposit acct, savings account, savings acct, chequing account, chequing acct

CloudWatch I registri non riportano le occorrenze delle seguenti sequenze, che gli emittenti di carte di credito hanno riservato ai test pubblici.

```
122000000000003, 2222405343248877, 2222990905257051, 2223007648726984,
 2223577120017656,
30569309025904, 34343434343434, 3528000700000000, 3530111333300000, 3566002020360505,
 36148900647913,
```

```
36700102000000, 371449635398431, 378282246310005, 378734493671000, 38520000023237,
 4012888888881881,
4111111111111111, 4222222222222, 4444333322221111, 4462030000000000, 4484070000000000,
 4911830000000,
4917300800000000, 4917610000000000, 4917610000000000003, 5019717010103742,
 5105105105105100,
5111010030175156, 5185540810000019, 5200828282828210, 5204230080000017,
 5204740009900014, 5420923878724339,
5454545454545454, 5455330760000018, 5506900490000436, 5506900490000444,
 5506900510000234, 5506920809243667,
5506922400634930, 5506927427317625, 5553042241984105, 5555553753048194,
 5555555555554444, 5610591081018250,
6011000990139424, 6011000400000000, 6011111111111117, 630490017740292441,
 630495060000000000,
6331101999990016, 6759649826438453, 6799990100000000019, and 76009244561.
```

Identificatore ARNs di dati per tipi di dati finanziari

Di seguito sono elencati gli Amazon Resource Names (ARNs) per gli identificatori di dati che puoi aggiungere alle tue politiche di protezione dei dati.

Identificatore di dati finanziari ARNs
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-DE
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-ES
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-FR
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-GB
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-IT
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-US
arn:aws:dataprotection::aws:data-identifier/CreditCardExpiration
arn:aws:dataprotection::aws:data-identifier/CreditCardNumber
arn:aws:dataprotection::aws:data-identifier/CreditCardSecurityC ode

Dati sanitari protetti (PHI)

CloudWatch I registri di protezione dei dati possono trovare i seguenti tipi di informazioni sanitarie protette (PHI).

Se imposti una politica di protezione dei dati, CloudWatch Logs cerca gli identificatori di dati specificati indipendentemente dalla geolocalizzazione in cui si trova il gruppo di log. Le informazioni nella colonna Paesi e aree geografiche di questa tabella indicano se è necessario aggiungere i codici Paese a due lettere all'identificatore di dati per individuare le parole chiave appropriate per tali Paesi e aree geografiche.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e regioni
Numero di registrazione Drug Enforcement Agency (DEA)	`DrugEnforcementAgencyNumber`	`dea number`, `dea registration`	Stati Uniti
Numero EHIC (Health Insurance Card)	`HealthInsuranceCardNumber`	`assicurazione sanitaria numero`, `carta assicurazione numero`, `carte d'assurance maladie `, `carte européenne d'assurance maladie `, `ceam`, `ehic`, `ehic#`, `finlandehicnumber#  `, `gesundheitskarte `, `hälsokort `, `health card`, `health card number`, `health insurance card`, `health insurance number`, `insurance card number`, `krankenversicherungskarte `, `krankenversicherungsnummer  `, `medical account number`, `numero conto medico`,	Unione Europea

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e regioni
		numéro d'assurance maladie , numéro de carte d'assurance , numéro de compte medical, número de cuenta médica, número de seguro de salud, número de tarjeta de seguro, sairaanhoitokortin , sairausvakuutuskortti , sairausvakuutusnumero , sjukförsäkringnummer, sjukförsäkringskort , suomi ehic-numero , tarjeta de salud, terveyskortti , tessera sanitaria assicurazione numero , versicherungsnummer	
Health Insurance Claim Number (HICN)	HealthInsuranceClaimNumber	health insurance claim number, hic no, hic no., hic number, hic#, hicn, hicn#, hicno#	Stati Uniti
Numero di identificazione medica e assistenza sanitaria	HealthInsuranceNumber	carte d'assuré social, carte vitale, insurance card	Francia

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e regioni
Codice HCPCS (Healthcare Common Procedure Coding System)	`HealthcareProcedureCode`	`current procedural terminology` , `hcpcs`, `healthcare common procedure coding system`	Stati Uniti
Numero MBN (Medicare Beneficiary Number)	`MedicareBeneficiaryNumber`	`mbi`, `medicare beneficiary`	Stati Uniti
National Drug Code (NDC)	`NationalDrugCode`	`national drug code`, `ndc`	Stati Uniti
National Provider Identifier (NPI)	`NationalProviderId`	`hipaa`, `n.p.i.`, `national provider`, `npi`	Stati Uniti
Numero National Health Service (NHS)	`NhsNumber`	`national health service`, `NHS`	Gran Bretagna
Personal Health Number	`PersonalHealthNumber`	`canada healthcare number`, `msp number`, `care number`, `phn`, `soins de santé`	Canada

Identificatore di dati ARNs per i tipi di dati relativi alle informazioni sanitarie protette (PHI)

Di seguito sono elencati gli identificatori di dati Amazon Resource Names (ARNs) che possono essere utilizzati nelle politiche di protezione dei dati relativi alle informazioni sanitarie protette (PHI).

Identificatore di dati PHI ARNs
`arn:aws:dataprotection::aws:data-identifier/DrugEnforcementAgencyNumber-US`

Identificatore di dati PHI ARNs
arn:aws:dataprotection::aws:data-identifier/HealthcareProcedureCode-US
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceCardNumber-EU
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceClaimNumber-US
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceNumber-FR
arn:aws:dataprotection::aws:data-identifier/MedicareBeneficiaryNumber-US
arn:aws:dataprotection::aws:data-identifier/NationalDrugCode-US
arn:aws:dataprotection::aws:data-identifier/NationalInsuranceNumber-GB
arn:aws:dataprotection::aws:data-identifier/NationalProviderId-US
arn:aws:dataprotection::aws:data-identifier/NhsNumber-GB
arn:aws:dataprotection::aws:data-identifier/PersonalHealthNumber-CA

Informazioni personali di identificazione (PII)

CloudWatch La protezione dei dati dei registri può trovare i seguenti tipi di informazioni di identificazione personale (PII).

Se imposti una politica di protezione dei dati, CloudWatch Logs cerca gli identificatori di dati specificati indipendentemente dalla geolocalizzazione in cui si trova il gruppo di log. Le informazioni nella colonna Paesi e aree geografiche di questa tabella indicano se è necessario aggiungere i codici Paese a due lettere all'identificatore di dati per individuare le parole chiave appropriate per tali Paesi e aree geografiche.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Data di nascita	`DateOfBirth`	`dob`, `date of birth`, `birthdate` , `birth date`, `birthday`, `b-day`, `bday`	Qualsias	Il supporto include la maggior parte dei formati di data, ad esempio tutte le cifre e le combinazioni di cifre e nomi dei mesi. I componenti della data possono essere separati da spazi, barre (/) o

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
				trattini (-).
Código de Endereçamento Postal (CEP)	`CepCode`	`cep`, `código de endereçamento postal`, `codigo de endereçamento postal`	Brasile	
Cadastro Nacional da Pessoa Jurídica (CNPJ)	`Cnpj`	`cadastro nacional da pessoa jurídica`, `cadastro nacional da pessoa juridica`, `cnpj`	Brasile	
Cadastro de Pessoas Físicas (CPF)	`CpfCode`	`Cadastro de pessoas fisicas`, `cadastro de pessoas físicas`, `cadastro de pessoa física`, `cadastro de pessoa fisica`, `cpf`	Brasile	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero identificativo della patente di guida	`DriversLicense`	Sì. A Paesi diversi si applicano parole chiave diverse. Per i dettagli, consulta la tabella Numeri identificativi della patente di guida più avanti in questa sezione.	Molti Paesi. Per i dettagli, consulta la tabella Numeri identificativi della patente di guida.	
Numero di lista elettorale	`ElectoralRollNumber`	`electoral #`, `electoral number`, `electoral roll #`, `electoral roll no.`, `electoral roll number`, `electoral rollno`	Regno Unito	
Identificazione del singolo contribuente	`IndividualTaxIdentificationNumber`	Sì. A Paesi diversi si applicano parole chiave diverse. Per i dettagli, consulta la tabella Numeri di identificazione del singolo contribuente più avanti in questa sezione.	Brasile, Francia, Germani Regno Unito, Spagna	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Institut national de la statistique et des études économiques (INSEE)	`InseeCode`	Sì. A Paesi diversi si applicano parole chiave diverse. Per i dettagli, consulta la tabella Parole chiave per i numeri di identificazione nazionale più avanti in questa sezione.	Francia	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero di identificazione nazionale	`NationalIdentificationNumber`	Sì. Per i dettagli, consulta la tabella Parole chiave per i numeri di identificazione nazionale più avanti in questa sezione.	Germania, Italia, Spagna	Sono inclusi gli identificatori DNI (Documento Nacional de Identidad, Spagna), il codice fiscale (Italia) e i numeri di carta d'identità nazionale (Germania).

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero NINO (National Insurance Number)	`NationalInsuranceNumber`	`insurance no.`, `insurance number`, `insurance#` , `national insurance number`, `nationalinsurance#` , `nationalinsurancenumber` , `nin`, `nino`	Regno Unito	–
Número de identidad de extranjero (NIE)	`NieNumber`	Sì. A Paesi diversi si applicano parole chiave diverse. Per i dettagli, consulta la tabella Numeri di identificazione del singolo contribuente più avanti in questa sezione.	Spagna	
Número de Identificación Fiscal (NIF)	`NifNumber`	Sì. A Paesi diversi si applicano parole chiave diverse. Per i dettagli, consulta la tabella Numeri di identificazione del singolo contribuente più avanti in questa sezione.	Spagna	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero di passaporto	`PassportNumber`	Sì. A Paesi diversi si applicano parole chiave diverse. Per maggiori dettagli, consulta la tabella Parole chiave per i numeri di passaporto più avanti in questa sezione.	Canada, Francia, Germani Italia, Regno Unito, Spagna, Stati Uniti	
Numero di residenza permanente (Green Card)	`PermanentResidenceNumber`	`carte résident permanent` , `numéro carte résident permanent` , `numéro résident permanent` , `permanent resident card`, `permanent resident card number`, `permanent resident no`, `permanent resident no.`, `permanent resident number`, `pr no`, `pr no.`, `pr non`, `pr number`, `résident permanent no.`, `résident permanent non`	Canada	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero di telefono	PhoneNumber	Brasile: le parole chiave comprendono anche: `cel`, `celular`, `fone`, `móvel`, `número residencial` , `numero residencial` , `telefone` Altri Paesi: `cell`, `contact`, `fax`, `fax number`, `mobile`, `phone`, `phone number`, `tel`, `telephone` , `telephone number`	Brasile, Canada, Francia, Germani Italia, Regno Unito, Spagna, Stati Uniti	Sono inclusi i numeri di fax e i numeri verdi negli Stati Uniti. Se una parola chiave si trova in prossimità dei dati, il numero non deve includere il prefisso internazionale. Se una parola chiave non

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
				è in prossimità dei dati, il numero deve includere un prefisso internazionale.
Codice postale	`PostalCode`	Nessuno	Canada	
Registro Geral (RG)	`RgNumber`	Sì. A Paesi diversi si applicano parole chiave diverse. Per i dettagli, consulta la tabella Numeri di identificazione del singolo contribuente più avanti in questa sezione.	Brasile	
Social Insurance Number (SIN)	`SocialInsuranceNumber`	`canadian id`, `numéro d'assurance sociale`, `social insurance number`, `sin`	Canada	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Social Security Number (SSN)	`Ssn`	Spagna: `número de la seguridad social`, `social security no.`, `social security no.` `número de la seguridad social`, `social security number`, `socialsecurityno#` , `ssn`, `ssn#` Stati Uniti: `social security`, `ss#`, `ssn`	Spagna, Stati Uniti	
Numero identificativo del contribuente o codice fiscale	`TaxId`	Sì. A Paesi diversi si applicano parole chiave diverse. Per i dettagli, consulta la tabella Numeri di identificazione del singolo contribuente più avanti in questa sezione. .	Francia, Germania, Regno Unito, Spagna	Sono inclusi: TIN (Francia); Steueridentifikationsnummer (Germania); CIF (Spagna) e TRN, UTR (Regno Unito).

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Codice postale	`ZipCode`	`zip code`, `zip+4`	Stati Uniti	Codice postale degli Stati Uniti.
Indirizzo postale	`Address`	Nessuno	Australia, Canada, Francia, Germania, Italia, Regno Unito, Spagna, Stati Uniti	Sebbene non sia richiesta una parola chiave, il rilevamento richiede che l'indirizzo includa il nome di una città o di un luogo e un CAP o codice postale.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Indirizzo e-mail	`EmailAddress`	Nessuno	Qualsias	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Coordinate del sistema di posizionamento globale (GPS)	`LatLong`	`coordinate` , `coordinates` , `lat long`, `latitude longitude` , `location`, `position`	Qualsias	CloudWatch I log possono rilevare le coordinate GPS se le coordinate di latitudine e longitudine sono memorizzate come coppia e sono in formato DD (Decimal Degrees), ad esempio 41.948614 , -87,655311. Il supporto

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
				non include le coordinate nel formato DDM (Gradi Decimali Minuti), ad esempio 41°56.9168'N 87°39.3187'W, o nel formato Gradi, Minuti, Secondi (DMS), ad esempio 41°56'55.0104"N 87°39'19.1196"W.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Nome completo	`Name`	Nessuno	Qualsias	CloudWatch I log possono rilevare solo i nomi completi. Il supporto è limitato ai set di caratteri latini.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero di matricola del veicolo (VIN)	`VehicleIdentificationNumber`	`Fahrgestellnummer`, `niv`, `numarul de identificare`, `numarul seriei de sasiu`, `serie sasiu`, `numer VIN`, `Número de Identificação do Veículo`, `Número de Identificación de Automóviles`, `numéro d'identification du véhicule`, `vehicle identification number`, `vin`, `VIN numeris`	Qualsias	CloudWatch I log sono in grado di rilevare VINs che consistono in una sequenza di 17 caratteri e sono conformi agli standard ISO 3779 e 3780. Questi standard sono stati progettati per l'uso a livello mondiale.

Parole chiave per i numeri identificativi delle patenti di guida

Per rilevare vari tipi di numeri identificativi della patente di guida, CloudWatch Logs richiede che una parola chiave si trovi in prossimità dei numeri. La tabella seguente elenca le parole chiave che CloudWatch Logs riconosce per paesi e aree geografiche specifici.

Paese o regione	Parole chiave
Australia	dl# dl:, dl :, dlno# driver licence, driver license, driver permit, drivers lic., drivers licence, driver's licence, drivers license, driver's license, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Austria	führerschein, fuhrerschein, führerschein republik österreich, fuhrerschein republik osterreich
Belgio	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer, numéro permis conduire, permis de conduire, rijbewijs, rijbewijsnummer
Bulgaria	превозно средство, свидетелство за управление на моторно, свидетелство за управление на мпс, сумпс, шофьорска книжка
Canada	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit,

Paese o regione	Parole chiave
	drivers permit number, driving licence, driving license, driving permit, permis de conduire
Croazia	vozačka dozvola
Cipro	άδεια οδήγησης
Repubblica Ceca	číslo licence, číslo licence řidiče, číslo řidičskéh o průkazu, ovladače lic., povolení k jízdě, povolení řidiče, řidiči povolení, řidičský prúkaz, řidičský průkaz
Danimarca	kørekort, kørekortnummer
Estonia	juhi litsentsi number, juhiloa number, juhiluba, juhiluba number
Finlandia	ajokortin numero, ajokortti, förare lic., körkort, körkort nummer, kuljettaja lic., permis de conduire
Francia	permis de conduire
Germania	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer
Grecia	δεια οδήγησης, adeia odigisis
Ungheria	illesztőprogramok lic, jogosítvány, jogsi, licencszám, vezető engedély, vezetői engedély
Irlanda	ceadúnas tiomána
Italia	patente di guida, patente di guida numero, patente guida, patente guida numero

Paese o regione	Parole chiave
Lettonia	autovadītāja apliecība, licences numurs, vadītāja apliecība, vadītāja apliecības numurs, vadītāja atļauja, vadītāja licences numurs, vadītāji lic.
Lituania	vairuotojo pažymėjimas
Lussemburgo	fahrerlaubnis, führerschäin
Malta	liċenzja tas-sewqan
Paesi Bassi	permis de conduire, rijbewijs, rijbewijsnummer
Polonia	numer licencyjny, prawo jazdy, zezwolenie na prowadzenie
Portogallo	carta de condução, carteira de habilitação, carteira de motorist, carteira habilitação, carteira motorist, licença condução, licença de condução, número de licença, número licença, permissão condução, permissão de condução
Romania	numărul permisului de conducere, permis de conducere
Slovacchia	číslo licencie, číslo vodičského preukazu, ovládače lic., povolenia vodičov, povolenie jazdu, povolenie na jazdu, povolenie vodiča, vodičský preukaz
Slovenia	vozniško dovoljenje

Paese o regione	Parole chiave
Spagna	carnet conducer, el carnet de conducer, licencia conducer, licencia de manejo, número carnet conducer, número de carnet de conducer, número de permiso conducer, número de permiso de conducer, número licencia conducer, número permiso conducer, permiso conducción, permiso conducer, permiso de conducción
Svezia	ajokortin numero, dlno# ajokortti, drivere lic., förare lic., körkort, körkort nummer, körkortsn ummer, kuljettajat lic.
Regno Unito	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Stati Uniti	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit

Parole chiave per i numeri di carta d'identità

Per rilevare vari tipi di numeri di identificazione nazionali, CloudWatch Logs richiede che una parola chiave si trovi in prossimità dei numeri. Sono inclusi gli identificatori DNI (Documento Nacional de

Identidad, Spagna), i codici INSEE (Institut national de la statistique et des études économiques), i numeri delle carte d'identità tedesche e i numeri RG (Registro Geral, Brasile).

La tabella seguente elenca le parole chiave che CloudWatch Logs riconosce per paesi e regioni specifici.

Paese o regione	Parole chiave
Brasile	registro geral, rg
Francia	assurance sociale, carte nationale d'identit é, cni, code sécurité sociale, French social security number, fssn#, insee, insurance number, national id number, nationalid#, numéro d'assurance, sécurité sociale, sécurité sociale non., sécurité sociale numéro, social, social security, social security number, socialsecuritynumber, ss#, ssn, ssn#
Germania	ausweisnummer, id number, identification number, identity number, insurance number, personal id, personalausweis
Italia	codice fiscal, dati anagrafici, ehic, health card, health insurance card, p. iva, partita i.v.a., personal data, tax code, tessera sanitaria
Spagna	dni, dni#, dninúmero#, documento nacional de identidad, identidad único, identidadúnico#, insurance number, national identification number, national identity, nationalid#, nationali dno#, número nacional identidad, personal identification number, personal identity no, unique identity number, uniqueid#

Parole chiave per i numeri di passaporto

Per rilevare vari tipi di numeri di passaporto, CloudWatch Logs richiede che una parola chiave si trovi in prossimità dei numeri. La tabella seguente elenca le parole chiave che CloudWatch Logs riconosce per paesi e aree geografiche specifici.

Paese o regione	Parole chiave
Canada	passeport, passeport#, passport, passport#, passportno, passportno#
Francia	numéro de passeport, passeport, passeport #, passeport #, passeportn °, passeport n °, passeportNon, passeport non
Germania	ausstellungsdatum, ausstellungsort, geburtsdatum, passport, passports, reisepass, reisepass–nr, reisepassnummer
Italia	italian passport number, numéro passeport, numéro passeport italien, passaporto, passaporto italiana, passaporto numero, passport number, repubblica italiana passaporto
Spagna	españa pasaporte, libreta pasaporte, número pasaporte, pasaporte, passport, passport book, passport no, passport number, spain passport
Regno Unito	passeport #, passeport n °, passeportNon, passeport non, passeportn °, passport #, passport no, passport number, passport#, passportid
Stati Uniti	passport, travel document

Parole chiave per i numeri di identificazione dei contribuenti e i numeri di riferimento

Per rilevare vari tipi di codice identificativo e di riferimento dei contribuenti, CloudWatch Logs richiede che una parola chiave si trovi in prossimità dei numeri. La tabella seguente elenca le parole chiave che CloudWatch Logs riconosce per paesi e aree geografiche specifici.

Paese o regione	Parole chiave
Brasile	cadastro de pessoa física, cadastro de pessoa fisica, cadastro de pessoas físicas, cadastro de pessoas fisicas, cadastro nacional da pessoa jurídica, cadastro nacional da pessoa juridica, cnpj, cpf
Francia	numéro d'identification fiscale, tax id, tax identification number, tax number, tin, tin#
Germania	identifikationsnummer, steuer id, steueride ntifikationsnummer, steuernummer, tax id, tax identification number, tax number
Spagna	cif, cif número, cifnúmero#, nie, nif, número de contribuyente, número de identidad de extranjero, número de identificación fiscal, número de impuesto corporativo, personal tax number, tax id, tax identification number, tax number, tin, tin#
Regno Unito	paye, tax id, tax id no., tax id number, tax identification, tax identification#, tax no., tax number, tax reference, tax#, taxid#, temporary reference number, tin, trn, unique tax reference, unique taxpayer reference, utr
Stati Uniti	Individual Taxpayer Identification Numbers (ITIN o i.t.i.n.)

Identificatore di dati ARNs per informazioni di identificazione personale (PII)

La tabella seguente elenca gli Amazon Resource Names (ARNs) per gli identificatori di dati personali (PII) che puoi aggiungere alle tue politiche di protezione dei dati.

Identificatore di dati PII ARNs
`arn:aws:dataprotection::aws:data-identifier/Address`
`arn:aws:dataprotection::aws:data-identifier/CepCode-BR`
`arn:aws:dataprotection::aws:data-identifier/Cnpj-BR`
`arn:aws:dataprotection::aws:data-identifier/CpfCode-BR`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-AT`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-AU`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-BE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-BG`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CA`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CY`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CZ`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-DE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-DK`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-EE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-ES`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-FI`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-FR`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-GB`

Identificatore di dati PII ARNs
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LV
arn:aws:dataprotection::aws:data-identifier/DriversLicense-MT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-NL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-RO
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-US
arn:aws:dataprotection::aws:data-identifier/ElectoralRollNumber-GB
arn:aws:dataprotection::aws:data-identifier/EmailAddress

Identificatore di dati PII ARNs
arn:aws:dataprotection::aws:data-identifier/IndividualTaxIdentificationNumber-US
arn:aws:dataprotection::aws:data-identifier/InseeCode-FR
arn:aws:dataprotection::aws:data-identifier/LatLong
arn:aws:dataprotection::aws:data-identifier/Name
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-DE
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-ES
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-IT
arn:aws:dataprotection::aws:data-identifier/NieNumber-ES
arn:aws:dataprotection::aws:data-identifier/NifNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-CA
arn:aws:dataprotection::aws:data-identifier/PassportNumber-DE
arn:aws:dataprotection::aws:data-identifier/PassportNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-FR
arn:aws:dataprotection::aws:data-identifier/PassportNumber-GB
arn:aws:dataprotection::aws:data-identifier/PassportNumber-IT
arn:aws:dataprotection::aws:data-identifier/PassportNumber-US
arn:aws:dataprotection::aws:data-identifier/PermanentResidenceNumber-CA

Identificatore di dati PII ARNs
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-BR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-DE
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-ES
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-FR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-GB
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-IT
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-US
arn:aws:dataprotection::aws:data-identifier/PostalCode-CA
arn:aws:dataprotection::aws:data-identifier/RgNumber-BR
arn:aws:dataprotection::aws:data-identifier/SocialInsuranceNumber-CA
arn:aws:dataprotection::aws:data-identifier/Ssn-ES
arn:aws:dataprotection::aws:data-identifier/Ssn-US
arn:aws:dataprotection::aws:data-identifier/TaxId-DE
arn:aws:dataprotection::aws:data-identifier/TaxId-ES
arn:aws:dataprotection::aws:data-identifier/TaxId-FR
arn:aws:dataprotection::aws:data-identifier/TaxId-GB
arn:aws:dataprotection::aws:data-identifier/VehicleIdentificationNumber
arn:aws:dataprotection::aws:data-identifier/ZipCode-US

Identificatori di dati personalizzati

Argomenti

- Cosa sono gli identificatori di dati personalizzati?
- Vincoli degli identificatori di dati personalizzati
- Utilizzo di identificatori di dati personalizzati nella console
- Utilizzo degli identificatori di dati personalizzati nella policy di protezione dei dati

Cosa sono gli identificatori di dati personalizzati?

Gli identificatori di dati personalizzati (CDIs) consentono di definire espressioni regolari personalizzate che possono essere utilizzate nella politica di protezione dei dati. Utilizzando gli identificatori di dati personalizzati, puoi indirizzare i casi d'uso delle informazioni di identificazione personale (PII) specifici dell'azienda che gli identificatori di dati gestiti non sono in grado di fornire. Ad esempio, è possibile utilizzare un identificatore di dati personalizzato per cercare dipendenti specifici dell'azienda. IDs Gli identificatori di dati personalizzati possono essere utilizzati insieme agli identificatori di dati gestiti.

Vincoli degli identificatori di dati personalizzati

CloudWatch Gli identificatori di dati personalizzati dei registri presentano le seguenti limitazioni:

- Ciascuna policy di protezione dei dati attualmente supporta un massimo di 10 identificatori di dati personalizzati.
- I nomi degli identificatori di dati personalizzati hanno una lunghezza massima di 128 caratteri. Sono supportati i seguenti caratteri:
  - Alfanumerici: (a-zA-Z0-9)
  - Simboli: ('_' | '-')
- RegEx ha una lunghezza massima di 200 caratteri. Sono supportati i seguenti caratteri:
  - Alfanumerici: (a-zA-Z0-9)
  - Simboli: ('_' | '#' | '=' | '@' |'/' | ';' | ',' | '-' | ' ')
  - Caratteri riservati RegEx: ('^' | '$' | '?' | '[' | ']' | '{' | '}' | '|' | '\\' | '*' | '+' | '.')
- Gli identificatori di dati personalizzati non possono condividere lo stesso nome di un identificatore di dati gestito.

- Gli identificatori di dati personalizzati possono essere specificati all'interno di una politica di protezione dei dati a livello di account o nelle politiche di protezione dei dati a livello di gruppo di log. Analogamente agli identificatori di dati gestiti, gli identificatori di dati personalizzati definiti all'interno di una politica a livello di account funzionano in combinazione con gli identificatori di dati personalizzati definiti in una politica a livello di gruppo di log.

Utilizzo di identificatori di dati personalizzati nella console

Quando si utilizza la CloudWatch console per creare o modificare una politica di protezione dei dati, per specificare un identificatore di dati personalizzato è sufficiente inserire un nome e un'espressione regolare per l'identificatore di dati. Ad esempio, è possibile immettere **`Employee_ID`** come nome e **`EmployeeID-\d{9}`** come espressione regolare. Questa espressione regolare rileverà e maschererà gli eventi di registro seguiti da nove numeri`EmployeeID-`. Ad esempio, `EmployeeID-123456789`

Utilizzo degli identificatori di dati personalizzati nella policy di protezione dei dati

Se si utilizza l' AWS API AWS CLI or per specificare un identificatore di dati personalizzato, è necessario includere il nome dell'identificatore di dati e l'espressione regolare nella politica JSON utilizzata per definire la politica di protezione dei dati. La seguente politica di protezione dei dati rileva e maschera gli eventi di registro che coinvolgono dipendenti specifici dell'azienda. IDs

1. Creazione di un blocco `Configuration` all'interno della policy di protezione dei dati.
2. Inserisci un `Name` per l'identificatore di dati personalizzato. Ad esempio, **`EmployeeId`**.
3. Inserisci un `Regex` per l'identificatore di dati personalizzato. Ad esempio, **`EmployeeID-\d{9}`**. Questa espressione regolare corrisponderà agli eventi di registro `EmployeeID-` che contengono nove cifre successive. `EmployeeID-` Ad esempio, `EmployeeID-123456789`
4. Riferimento al seguente identificatore di dati personalizzato in una dichiarazione di policy.

```
{
    "Name": "example_data_protection_policy",
    "Description": "Example data protection policy with custom data identifiers",
    "Version": "2021-06-01",
    "Configuration": {
      "CustomDataIdentifier": [
        {"Name": "EmployeeId", "Regex": "EmployeeId-\\d{9}"}
      ]
    },
    "Statement": [
```

```
        {
            "Sid": "audit-policy",
            "DataIdentifier": [
                "EmployeeId"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
            "EmployeeId"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {
                    }
                }
            }
        }
    ]
}
```

5. (Facoltativo) Continua ad aggiungere altri identificatori di dati personalizzati al blocco `Configuration`, se necessario. Le policy di protezione dei dati attualmente supportano un massimo di 10 identificatori di dati personalizzati.

Trasforma i log durante l'ingestione

Con la trasformazione e l'arricchimento dei log, puoi normalizzare tutti i log in un formato coerente e ricco di contesto al momento dell'inserimento in Logs. CloudWatch Puoi aggiungere una struttura ai tuoi log utilizzando modelli preconfigurati per AWS servizi comuni come AWS WAF Amazon Route 53 o creare trasformatori personalizzati con parser nativi come Grok. Puoi anche rinominare gli attributi esistenti e aggiungere metadati aggiuntivi ai log, come l'ID dell'account e la regione.

La trasformazione dei log aiuta a semplificare e abbreviare le query di registro tra le applicazioni e aiuta a semplificare la creazione di avvisi sui log. Questa funzionalità fornisce la trasformazione per i tipi di log più comuni con modelli di out-of-the-box trasformazione per le principali fonti di AWS log come i log di flusso VPC, Route 53 e. Amazon RDS for PostgreSQL Puoi utilizzare modelli di trasformazione preconfigurati o creare trasformatori personalizzati per soddisfare le tue esigenze.

La trasformazione dei registri consente di gestire i log emessi da varie fonti che variano notevolmente nel formato e nei nomi degli attributi.

Dopo aver creato un trasformatore, gli eventi di registro importati vengono convertiti e archiviati in un formato standard. Puoi sfruttare questi log trasformati per accelerare la tua esperienza di analisi con le seguenti funzionalità:

- Indici di campo
- CloudWatch Campi rilevati da Logs Insights
- Flessibilità nell'invio di allarmi utilizzando filtri metrici
- Inoltro tramite filtri di abbonamento
- Creazione di dati metrici dagli eventi di registro con Contributor Insights, dove puoi scegliere di fare in modo che la regola Contributor Insights valuti gli eventi di registro prima o dopo la loro trasformazione.

Le trasformazioni avvengono solo durante l'inserimento dei log. Non è possibile trasformare gli eventi di registro che sono già stati inseriti. Le trasformazioni non sono reversibili. Sia i log originali che quelli trasformati vengono archiviati in CloudWatch Log con la stessa politica di conservazione. La funzionalità di trasformazione e arricchimento dei log è inclusa nel prezzo di inserimento della classe di log Standard esistente. I costi di archiviazione dei log si baseranno sulla dimensione del registro dopo la trasformazione, che potrebbe superare il volume di log originale.

> **Important**
>
> Dopo la trasformazione degli eventi di registro, è necessario utilizzare le query di CloudWatch Logs Insights per visualizzare le versioni trasformate dei log. Le FilterLogEventsazioni GetLogEventsand restituiscono solo le versioni originali degli eventi di registro, prima che venissero trasformati.

> **Important**
>
> Nonostante un singolo evento di registro PutLogEvents consenta fino a 1 MB, la trasformazione dei log può gestire solo eventi di registro di dimensioni inferiori a 512 kb. Qualsiasi evento di registro superiore a 512 kb fallirà nella trasformazione e genererà un errore. La dimensione totale di PutLogEvents può comunque superare i 512 kb.

Oltre a trasformarli in diversi formati, puoi anche arricchire i tuoi log con un contesto aggiuntivo, come l'ID dell'account, la regione e la parola chiave. Questi vengono estratti dal nome del gruppo di log e dalle parole chiave statiche.

La trasformazione dei log ti aiuta con i log emessi da varie fonti che variano notevolmente nel formato e nei nomi degli attributi.

La trasformazione e l'arricchimento dei log sono supportati solo per i gruppi di log nella classe di log Standard.

Puoi creare trasformatori per singoli gruppi di log e puoi anche creare trasformatori a livello di account che si applicano a tutti o a molti gruppi di log del tuo account. Se un gruppo di log ha un trasformatore a livello di gruppo di log, tale trasformatore sostituisce qualsiasi trasformatore a livello di account che altrimenti si applicherebbe a quel gruppo di log.

Argomenti

- Crea e gestisci trasformatori di log
- Processori configurabili di tipo parser
- Processori integrati per i AWS log venduti
- Processori di mutazione delle stringhe
- Processori JSON con mutazione

- Processori di conversione di tipi di dati
- Metriche ed errori di trasformazione

Crea e gestisci trasformatori di log

Un trasformatore di log include uno o più processori che si trovano insieme in una pipeline logica. Ogni processore viene applicato a un evento di registro, uno dopo l'altro nell'ordine in cui sono elencati nella configurazione del trasformatore.

Alcuni processori sono del tipo parser. Ogni trasformatore deve avere almeno un parser e il primo processore di un trasformatore deve essere un parser.

Alcuni parser sono parser integrati configurati per un determinato tipo di log fornito. AWS

Altri tipi di processori sono i mutatori di stringa, i mutatori JSON e i processori di dati.

Puoi creare trasformatori per singoli gruppi di log e puoi anche creare trasformatori a livello di account che si applicano a tutti o a molti gruppi di log del tuo account. Se un gruppo di log ha un trasformatore a livello di gruppo di log, tale trasformatore sostituisce qualsiasi trasformatore a livello di account che altrimenti si applicherebbe a quel gruppo di log. Nel tuo account puoi avere fino a 20 trasformatori a livello di account in una regione.

È necessario seguire queste linee guida quando si crea un trasformatore:

- Se includete un parser preconfigurato per un tipo di log AWS forniti, deve essere il primo processore elencato nel trasformatore. È possibile includere solo un processore di questo tipo in un trasformatore.
- È possibile includere un solo `grok` processore in un trasformatore.
- È necessario disporre di almeno un processore di tipo parser in un trasformatore. È possibile includere fino a cinque processori di tipo parser. Questo limite di cinque include sia i parser integrati che i parser configurabili.
- È possibile avere fino a 20 processori in un trasformatore.
- È possibile includere un solo processore AddKeys in un trasformatore.
- È possibile includere un solo processore CopyValue in un trasformatore.
- Ogni trasformatore può estrarre fino a 200 campi da un evento di registro.
- Ogni evento di registro DEVE essere inferiore a 512 KB. La dimensione totale degli eventi di registro può comunque superare i 512 KB.

Argomenti

- Crea una politica di trasformazione a livello di account
- Modifica o elimina una politica di trasformazione a livello di account
- Crea un trasformatore di log-group-level log partendo da zero
- Crea un log-group-level trasformatore copiandone uno esistente
- Modifica un trasformatore log-group-level
- Eliminare un trasformatore log-group-level

Crea una politica di trasformazione a livello di account

Utilizza i passaggi di questa sezione per creare una politica di trasformazione che si applichi a tutti i gruppi di log dell'account o a più gruppi di log i cui nomi di gruppi di log iniziano con la stessa stringa (prefisso). È possibile avere fino a 20 politiche di trasformazione a livello di account in una regione.

Non è possibile creare due politiche di trasformazione nella stessa regione che utilizzano lo stesso prefisso o che hanno un prefisso contenuto all'interno di un altro. Ad esempio, se create una politica di trasformazione per il prefisso di stringa`/aws/lambda`, non potete crearne un'altra con il prefisso. `/aws` Ma potresti avere un trasformatore per e un altro per `/aws/lambda` `/aws/waf`

Per creare una politica di trasformazione a livello di account

1. Apri la console all' CloudWatch indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione a sinistra, scegli Impostazioni, quindi scegli la scheda Registri.
3. Nella sezione Transformer policy for account, scegli Crea policy Transformer.
4. Per il nome della politica Transformer, inserisci un nome per la tua nuova politica.
5. Per Select log groups, effettuate una delle seguenti operazioni:
   - Scegliete Tutti i gruppi di log standard per applicare la politica del trasformatore a tutti i gruppi di log Standard Class presenti nell'account.
   - scegli Registra i gruppi per corrispondenza del prefisso per applicare la politica a un sottoinsieme di gruppi di log i cui nomi iniziano tutti con la stessa stringa. Quindi, inserisci il prefisso per questi gruppi di log in Criteri di selezione.
6. Nell'area Seleziona parser, usa Parsers per selezionare un parser da includere nel tuo trasformatore.

Se si tratta di un parser preconfigurato per un tipo di registro AWS fornito, non è necessario specificare alcuna configurazione.

Se si tratta di un parser diverso, è necessario specificarne la configurazione. Per ulteriori informazioni, consulta le informazioni relative al processore in[Processori configurabili di tipo parser](#).

7. Per aggiungere un altro processore, scegli Seleziona processore. Quindi seleziona il processore che desideri nella casella Processore e inserisci i parametri di configurazione.

   Ricordate che i processori operano sugli eventi di registro nell'ordine in cui li aggiungete al trasformatore.

8. (Facoltativo) Per aggiungere processori aggiuntivi, scegliete + Processore e ripetete il passaggio precedente.

9. (Facoltativo) In qualsiasi momento, potete testare il trasformatore che avete creato finora sulla base di un evento di registro di esempio. A tale scopo, effettuate una delle seguenti operazioni nella sezione di anteprima di Transformer:

   - Seleziona fino a cinque gruppi di log in Seleziona gruppi di log, quindi scegli Carica gli ultimi eventi di registro. Quindi scegli Test transformer.
   - Copia gli eventi di registro direttamente in Sample log events, quindi scegli Test transformer.

   Viene quindi visualizzata la versione trasformata del registro.

10. Quando hai finito di aggiungere processori e sei soddisfatto dei test sui log di esempio, scegli Salva.

11. Al termine, scegliere Create (Crea).

Modifica o elimina una politica di trasformazione a livello di account

Utilizza i passaggi di questa sezione per modificare o eliminare una politica di trasformazione a livello di account.

Per modificare o eliminare una politica di trasformazione a livello di account

1. Apri la console all' CloudWatch indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione a sinistra, scegli Impostazioni, quindi scegli la scheda Registri.
3. Nella sezione Politica dell'account Transformer, scegli Gestisci.

4. Seleziona il pulsante accanto alla politica del trasformatore che desideri gestire, quindi scegli Modifica o Elimina.

   Se stai modificando la politica, consulta i passaggi da 5 a 11 Processori configurabili di tipo parser per vedere le opzioni a tua disposizione.

Crea un trasformatore di log-group-level log partendo da zero

Usa questi passaggi per creare un log-group-level trasformatore da zero.

Per utilizzare la console per creare un trasformatore di log per un gruppo di log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra, scegli Log, Gruppi di log.
3. Scegli il gruppo di log per cui vuoi creare il trasformatore.
4. Scegli la scheda Transformer. Potrebbe essere necessario scorrere l'elenco delle schede verso destra per visualizzarlo.
5. Scegli Crea trasformatore.
6. Nella casella Scegli un parser, seleziona un parser da includere nel tuo trasformatore.

   Se si tratta di un parser preconfigurato per un tipo di AWS registro fornito, non è necessario specificare alcuna configurazione.

   Se si tratta di un parser diverso, è necessario specificarne la configurazione. Per ulteriori informazioni, consulta le informazioni relative al processore inProcessori configurabili di tipo parser.
7. Per aggiungere un altro processore, scegli + Aggiungi processore. Quindi seleziona il processore che desideri nella casella Scegli processori e inserisci i parametri di configurazione.

   Ricordate che i processori operano sugli eventi di registro nell'ordine in cui li aggiungete al trasformatore.
8. (Facoltativo) In qualsiasi momento, potete testare il trasformatore che avete creato finora sulla base di un evento di registro di esempio. Per far ciò, completa le seguenti operazioni:

   - Nella sezione Anteprima della trasformazione, scegliete Carica registro di esempio per caricare un evento di registro di esempio dal gruppo di log a cui è destinato questo trasformatore, oppure incollate un evento di registro nella casella di testo.

Scegliete Test transformer. Viene visualizzata la versione trasformata del registro

9. Quando hai finito di aggiungere processori e sei soddisfatto dei test sui log di esempio, scegli Salva.

Per utilizzarlo AWS CLI per creare un trasformatore di log partendo da zero

- Utilizza il comando `aws logs put-transformer`. Quando si utilizza `parseJSON` come primo processore, è necessario analizzare l'intero evento di registro utilizzandolo `@message` come campo sorgente. Dopo l'analisi JSON iniziale, è possibile manipolare campi specifici nei processori successivi. Di seguito è riportato un esempio che crea un trasformatore che include i processori and: `parseJSON` `addKeys`

```
 aws logs put-transformer \
  --transformer-config '[{"parseJSON":{"source":"@message"}},{"addKeys":
{"entries":[{"key":"metadata.transformed_in","value":"CloudWatchLogs"},
{"key":"feature","value":"Transformation"}]}},{"trimString":{"withKeys":
["status"]}}]' \
  --log-group-identifier my-log-group-name
```

Crea un log-group-level trasformatore copiandone uno esistente

È possibile utilizzare la console per copiare la configurazione JSON di un trasformatore esistente. È quindi possibile utilizzare quel codice per creare un trasformatore identico utilizzando o modificare prima la configurazione. AWS CLI

Per creare un trasformatore di log copiandone uno esistente

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione a sinistra, scegli Log, Gruppi di log.
3. Scegli il gruppo di log contenente il trasformatore che desideri copiare.
4. Scegli la scheda Trasformazioni. Potrebbe essere necessario scorrere l'elenco delle schede verso destra per visualizzarlo.
5. Scegli Manage transformer.
6. Scegli Copy transformer. Questo copia il trasformatore JSON negli appunti.

7. Crea un file e incollalo nella configurazione del trasformatore. In questo esempio chiameremo il file `CopiedTransformer.json`
8. Usa il AWS CLI per creare un nuovo trasformatore con quella configurazione.

```
aws logs put-transformer --log-group-identifier my-log-group-name \
--transformer-config file://CopiedTransformer.json
```

Modifica un trasformatore log-group-level

Usa questi passaggi per modificare un trasformatore di log esistente.

Per modificare un trasformatore di log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra, scegli Log, Gruppi di log.
3. Scegli il gruppo di log contenente il trasformatore che desideri modificare.
4. Scegli la scheda Trasformazioni. Potrebbe essere necessario scorrere l'elenco delle schede verso destra per visualizzarlo.
5. Scegli Manage transformer.
6. Nelle sezioni Parser e Processori, apporta le modifiche.
7. Per aggiungere un altro processore, scegli + Aggiungi processore. Quindi seleziona il processore che desideri nella casella Processore e inserisci i parametri di configurazione.

   Ricordate che i processori operano sugli eventi di registro nell'ordine in cui li aggiungete al trasformatore.
8. (Facoltativo) In qualsiasi momento, potete testare il trasformatore che avete creato finora sulla base di un evento di registro di esempio. Per far ciò, completa le seguenti operazioni:

   - Nella sezione Transformation Preview, scegliete Load Sample Log per caricare un evento di log di esempio dal gruppo di log a cui è destinato questo trasformatore, oppure incollate un evento di log nella casella di testo.

     Scegliete Test Transformation. Viene visualizzata la versione trasformata del registro
9. Quando hai finito di aggiungere processori e sei soddisfatto dei test sui log di esempio, scegli Salva.

Eliminare un trasformatore log-group-level

Usa questi passaggi per eliminare un trasformatore di registro.

Per eliminare un trasformatore di log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra, scegli Log, Gruppi di log.
3. Scegli il gruppo di log contenente il trasformatore che desideri modificare.
4. Scegli la scheda Trasformazioni. Potrebbe essere necessario scorrere l'elenco delle schede verso destra per visualizzarlo.
5. Scegli Elimina.
6. Nella casella di conferma, scegli Elimina politica.

Processori configurabili di tipo parser

Questa sezione contiene informazioni sui processori di analisi dei dati configurabili che è possibile utilizzare in un trasformatore di eventi di registro.

Indice

- parseJSON
- grok
  - Esempi di Grok
    - Esempio 1: Usa grok per estrarre un campo da log non strutturati
    - Esempio 2: usa grok in combinazione con parseJSON per estrarre campi da un evento di registro JSON
    - Esempio 3: pattern Grok con annotazione punteggiata in FIELD_NAME
  - Schemi grok supportati
    - Esempi di formati di log comuni
      - Esempio di log di Apache
      - Esempio di log NGINX
      - Esempio di log del protocollo Syslog (RFC 5424)
- csv
- parseKeyValue

parseJSON

Il processore ParseJSON analizza gli eventi di registro JSON e inserisce le coppie chiave-valore JSON estratte nella destinazione. Se non si specifica una destinazione, il processore posiziona la coppia chiave-valore sotto il nodo radice. Quando si utilizza `parseJSON` come primo processore, è necessario analizzare l'intero evento di registro utilizzandolo `@message` come campo di origine. Dopo l'analisi JSON iniziale, è possibile manipolare campi specifici nei processori successivi.

Il `@message` contenuto originale non viene modificato, le nuove chiavi vengono aggiunte al messaggio.

Campo	Description	Obbligatorio?	Predefinita	Limits
origine	Percorso del campo nel registro dell'evento che verrà analizzato. Usa la notazione a punti per accedere ai campi secondari. Ad esempio, `store.book`	No	`@message`	Lunghezza massima: 128. Profondità massima della chiave annidata: 3
destinazione	Il campo di destinazione del file JSON analizzato	No	`Parent JSON node`	Lunghezza massima: 128. Profondità massima della chiave annidata: 3

Esempio

Supponiamo che un evento di registro importato abbia il seguente aspetto:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

Quindi se abbiamo questo processore ParseJson:

```
[
   {
        "parseJSON": {
            "destination": "new_key"
        }
   }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
    "new_key": {
        "outer_key": {
            "inner_key": "inner_value"
        }
    }
}
```

grok

Usa il processore grok per analizzare e strutturare i dati non strutturati utilizzando il pattern matching. Questo processore può anche estrarre campi dai messaggi di registro.

Campo	Description	Obbligatorio?	Predefinita	Limits	Note
origine	Percorso del campo su cui applicare la corrispondenza Grok	No	`@messag`	Lunghezza massima: 128. Profondità massima della chiave annidata: 3	
match	Il pattern grok da confrontare con l'evento log	Sì		Lunghezza massima: 512 Schemi di gruppo massimi: 20	Vedi tutti i pattern Grok supportati

Campo	Description	Obbligatorio?	Predefinita	Limits	Note
				Alcuni tipi di pattern grok hanno limiti di utilizzo individuali. Qualsiasi combinazione dei seguenti modelli può essere utilizzata fino a cinque volte: {URI, URIPARAM, URIPATHPARAM, SPACE, DATA, GREEDYDATA, GREEDYDATA_MULTILINE} I pattern Grok non supportano le conversioni di tipo. Per i modelli di formato di log comuni (APACHE_ACCESS_LOG, NGINX_ACCESS_LOG, SYSLOG5424), è supportato l'inclusione solo dei modelli DATA, GREEDYDATA o GREEDYDAT	

Campo	Description	Obbligatorio?	Predefinita	Limits	Note
				A_MULTILINE dopo il modello di log comune.	

Struttura di un pattern Grok

Questa è la struttura del pattern grok supportata:

```
%{PATTERN_NAME:FIELD_NAME}
```

- PATTERN_NAME: si riferisce a un'espressione regolare predefinita per la corrispondenza di un tipo specifico di dati. Sono supportati solo i modelli grok predefiniti. La creazione di modelli personalizzati non è consentita.
- FIELD_NAME: assegna un nome al valore estratto. `FIELD_NAME`è facoltativo, ma se non si specifica questo valore, i dati estratti verranno eliminati dall'evento di registro trasformato. Se `FIELD_NAME` utilizza la notazione punteggiata (ad esempio, «parent.child»), viene considerata un percorso JSON.
- Conversione dei tipi: le conversioni di tipo esplicite non sono supportate. Usa il TypeConverter processore per convertire il tipo di dati di qualsiasi valore estratto da grok.

Per creare espressioni di corrispondenza più complesse, puoi combinare diversi modelli grok. È possibile combinare fino a 20 pattern grok per abbinare un evento log. Ad esempio, questa combinazione di pattern `%{NUMBER:timestamp} [%{NUMBER:db} %{IP:client_ip}: %{NUMBER:client_port}] %{GREEDYDATA:data}` può essere utilizzata per estrarre campi da una voce di slow log di Redis come questa:

```
1629860738.123456 [0 127.0.0.1:6379] "SET" "key1" "value1"
```

Esempi di Grok

Esempio 1: Usa grok per estrarre un campo da log non strutturati

Registro di esempio:

```
293750 server-01.internal-network.local OK "[Thread-000] token generated"
```

Trasformatore utilizzato:

```
[
     {
         "grok": {
             "match": "%{NUMBER:version} %{HOSTNAME:hostname} %{NOTSPACE:status}
 %{QUOTEDSTRING:logMsg}"
         }
    }
]
```

Output:

```
{
  "version": "293750",
  "hostname": "server-01.internal-network.local",
  "status": "OK",
  "logMsg": "[Thread-000] token generated"
}
```

Registro di esempio:

```
23/Nov/2024:10:25:15 -0900 172.16.0.1 200
```

Trasformatore utilizzato:

```
[
    {
        "grok": {
            "match": "%{HTTPDATE:timestamp} %{IPORHOST:clientip}
 %{NUMBER:response_status}"
        }
    }
]
```

Output:

```
{
```

```
  "timestamp": "23/Nov/2024:10:25:15 -0900",
  "clientip": "172.16.0.1",
  "response_status": "200"
}
```

Esempio 2: usa grok in combinazione con parseJSON per estrarre campi da un evento di registro JSON

Registro di esempio:

```
{
    "timestamp": "2024-11-23T16:03:12Z",
    "level": "ERROR",
    "logMsg": "GET /page.html HTTP/1.1"
}
```

Trasformatore utilizzato:

```
[
     {
        "parseJSON": {}
    },
    {
         "grok": {
            "source": "logMsg",
             "match": "%{WORD:http_method} %{NOTSPACE:request} HTTP/
%{NUMBER:http_version}"
         }
    }
]
```

Output:

```
{
  "timestamp": "2024-11-23T16:03:12Z",
  "level": "ERROR",
  "logMsg": "GET /page.html HTTP/1.1",
  "http_method": "GET",
  "request": "/page.html",
  "http_version": "1.1"
}
```

Esempio 3: pattern Grok con annotazione punteggiata in FIELD_NAME

Registro di esempio:

```
192.168.1.1 GET /index.html?param=value 200 1234
```

Trasformatore utilizzato:

```
[
    {
        "grok": {
            "match": "%{IP:client.ip} %{WORD:method} %{URIPATHPARAM:request.uri}
 %{NUMBER:response.status} %{NUMBER:response.bytes}"
        }
    }
]
```

Output:

```
{
  "client": {
    "ip": "192.168.1.1"
  },
  "method": "GET",
  "request": {
    "uri": "/index.html?param=value"
  },
  "response": {
    "status": "200",
    "bytes": "1234"
  }
}
```

Schemi grok supportati

Le tabelle seguenti elencano i modelli supportati dal `grok` processore.

Schemi di gruppo generali

Modello Grok	Description	Limite massimo del pattern	Esempio
USERNAME o USER	Corrisponde a uno o più caratteri che possono includere lettere minuscole (a-z), lettere maiuscole (A-Z), cifre (0-9), punti (.), caratteri di sottolineatura (_) o trattini (-)	20	Input: `user123.name-TEST` Modello: `%{USERNAME:name}` Uscita: `{"name": "user123.name-TEST"}`
INT	Corrisponde a un segno più o meno opzionale seguito da una o più cifre.	20	Ingresso: `-456` Modello: `%{INT:num}` Uscita: `{"num": "-456"}`
BASE10NUM	Corrisponde a un numero intero o a virgola mobile con segno e virgola decimale opzionali	20	Ingresso: `-0.67` Modello: `%{BASE10NUM:num}` Uscita: `{"num": "-0.67"}`
BASE16NUM	Corrisponde ai numeri decimali ed esadecimali con un segno opzionale (+ o -) e un prefisso 0x opzionale	20	Ingresso: `+0xA1B2` Modello: `%{BASE16NUM:num}` Uscita: `{"num": "+0xA1B2"}`
PUNTO	Corrisponde a numeri interi positivi senza zeri iniziali, costituiti da una o più cifre (1-9 seguito da 0-9)	20	Ingresso: `123` Modello: `%{POSINT:num}` Uscita: `{"num": "123"}`
NONNEGINT	Corrisponde a qualsiasi numero intero (composto	20	Ingresso: `007`

Modello Grok	Description	Limite massimo del pattern	Esempio
	da una o più cifre da 0 a 9) compresi zero e numeri con zeri iniziali.		Modello: `%{NONNEGINT:num}` Uscita: `{"num": "007"}`
PAROLA	Corrisponde a parole intere composte da uno o più caratteri di parola (\ w), incluse lettere, cifre e caratteri di sottolineatura	20	Ingresso: `user_123` Modello: `%{WORD:user}` Uscita: `{"user": "user_123"}`
NON SPAZIO	Corrisponde a uno o più caratteri diversi dagli spazi bianchi.	5	Ingresso: `hello_world123` Modello: `%{NOTSPACE:msg}` Uscita: `{"msg": "hello_world123"}`
SPACE	Corrisponde a zero o più caratteri di spazio bianco.	5	Ingresso: `" "` Modello: `%{SPACE:extra}` Uscita: `{"extra": " "}`
DATA	Corrisponde a qualsiasi carattere (eccetto la nuova riga) zero o più volte, non è avido.	5	Ingresso: `abc def ghi` Modello: `%{DATA:x} %{DATA:y}` Uscita: `{"x": "abc", "y": "def ghi"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
DATI AVIDI	Corrisponde a qualsiasi carattere (eccetto la nuova riga) zero o più volte, greedy.	5	Ingresso: `abc def ghi` Modello: `%{GREEDYDATA:x} %{GREEDYDATA:y}` Uscita: `{"x": "abc def", "y": "ghi"}`

GREEDYDATA_MULTILINE	Corrisponde a qualsiasi carattere (inclusa la nuova riga) zero o più volte, greedy.	1	Input:
			
			`abc`
			
			`def`
			
			`ghi`
			
			Modello: `%{GREEDYDATA_MULTILINE:data}`
			
			Uscita: `{"data": "abc\ndef\nghi"}`

STRINGA TRA VIRGOLETTE	Corrisponde alle stringhe tra virgolette (virgolette singole o doppie) ai caratteri con escape.	20	Ingresso: `"Hello, world!"` Modello: `%{QUOTEDSTRING:msg}` Uscita: `{"msg": "Hello, world!"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
UUID	Corrisponde a un formato UUID standard: 8 caratteri esadecimali, seguiti da tre gruppi di 4 caratteri esadecimali e termina con 12 caratteri esadecimali, tutti separati da trattini.	20	Ingresso: `550e8400-e29b-41d4-a716-446655440000` Modello: `%{UUID:id}` Uscita: `{"id": "550e8400-e29b-41d4-a716-446655440000"}`
URNA	Corrisponde alla sintassi URN (Uniform Resource Name).	20	Ingresso: `urn:isbn:0451450523` Modello: `%{URN:urn}` Uscita: `{"urn": "urn:isbn:0451450523"}`

AWS modelli grok

Pattern	Description	Limite massimo del modello	Esempio
ARN	Corrisponde AWS ad Amazon Resource Names (ARNs), acquisendo la partizione (`aws`,, o`aws-us-gov`)`aws-cn`, il servizio, la regione, l'ID account e fino a 5 identific atori gerarchici di risorse separati da barre. Non ARNs	5	Ingresso: `arn:aws:iam:us-east-1:123456789012:user/johndoe` Modello: `%{ARN:arn}` Uscita: `{"arn": "arn:aws:iam:us-east-1:1234`

Pattern	Description	Limite massimo del modello	Esempio
	corrisponderà alle informazioni mancanti tra i due punti.		`56789012:user/john doe"}`

Schemi di gruppo di rete

Modello Grok	Description	Limite massimo del pattern	Esempio
CISCOMAC	Corrisponde a un indirizzo MAC in formato esadecimale 4-4-4.	20	Ingresso: `0123.4567.89AB` Modello: `%{CISCOMAC:MacAddress}` Uscita: `{"MacAddress": "0123.4567.89AB"}`
WINDOWS MAC	Corrisponde a un indirizzo MAC in formato esadecimale con trattini	20	Ingresso: `01-23-45-67-89-AB` Modello: `%{WINDOWSMAC:MacAddress}` Uscita: `{"MacAddress": "01-23-45-67-89-AB"}`
MAC COMUNE	Corrisponde a un indirizzo MAC in formato esadecimale con due punti.	20	Ingresso: `01:23:45:67:89:AB` Modello: `%{COMMONMAC:MacAddress}` Uscita: `{"MacAddress": "01:23:45:67:89:AB"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
MAC	Corrisponde a uno dei pattern grok CISCOMAC, WINDOWSMAC o COMMONMAC	20	Ingresso: `01:23:45:67:89:AB` Modello: `%{MAC:m1}` Uscita: `{"m1":"01:23:45:67:89:AB"}`
IPV6	Corrisponde IPv6 agli indirizzi, inclusi i moduli compressi e gli indirizzi IPv4 mappati IPv6 .	5	Ingresso: `2001:db8:3333:4444:5555:6666:7777:8888` Modello: `%{IPV6:ip}` Uscita: `{"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}`
IPV4	Corrisponde a un IPv4 indirizzo.	20	Ingresso: `192.168.0.1` Modello: `%{IPV4:ip}` Uscita: `{"ip": "192.168.0.1"}`
IP	IPv6 Corrisponde agli indirizzi supportati da% {IPv6} o IPv4 agli indirizzi supportati da% {IPv4}	5	Ingresso: `192.168.0.1` Modello: `%{IP:ip}` Uscita: `{"ip": "192.168.0.1"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
HOSTNAME o HOST	Corrisponde ai nomi di dominio, inclusi i sottodomini	5	Ingresso: `server-01.internal-network.local` Modello: `%{HOST:host}` Uscita: `{"host": "server-01.internal-network.local"}`
IP O HOST	Corrisponde a un nome host o a un indirizzo IP	5	Ingresso: `2001:db8:3333:4444:5555:6666:7777:8888` Modello: `%{IPORHOST:ip}` Uscita: `{"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}`
HOSTPORT	Corrisponde a un indirizzo IP o a un nome host come supportato dal pattern% {IPORHOST} seguito da due punti e da un numero di porta, che nell'output registra la porta come «PORT».	5	Ingresso: `192.168.1.1:8080` Modello: `%{HOSTPORT:ip}` Uscita: `{"ip":"192.168.1.1:8080","PORT":"8080"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
URIHOST	Corrisponde a un indirizzo IP o a un nome host, come supportato dal pattern% {IPORHOST}, seguito facoltativamente da due punti e da un numero di porta, identificando la porta come «porta», se presente.	5	Ingresso: `example.com:443 10.0.0.1` Modello: `%{URIHOST:host} %{URIHOST:ip}` Uscita: `{"host":"example.com:443","port":"443","ip":"10.0.0.1"}`

Schemi di gruppi di tracciati

Modello Grok	Description	Limite massimo del pattern	Esempio
UNIXPATH	Corrisponde ai percorsi degli URL, inclusi potenzialmente i parametri di query.	20	Ingresso: `/search?q=regex` Modello: `%{UNIXPATH:path}` Uscita: `{"path":"/search?q=regex"}`
PERCORSO VINCENTE	Corrisponde ai percorsi dei file di Windows.	5	Ingresso: `C:\Users\John\Documents\file.txt` Modello: `%{WINPATH:path}` Uscita: `{"path": "C:\\Users\\John\\`

Modello Grok	Description	Limite massimo del pattern	Esempio
			`Documents\\file.txt"}`
PATH	Corrisponde ai percorsi dei file URL o Windows	5	Ingresso: `/search?q=regex` Modello: `%{PATH:path}` Uscita: `{"path":"/search?q=regex"}`
TTY	Corrisponde ai percorsi dei dispositivi Unix per terminali e pseudo-terminali.	20	Ingresso: `/dev/tty1` Modello: `%{TTY:path}` Uscita: `{"path":"/dev/tty1"}`
URIPROTO	Corrisponde alle lettere, seguite facoltativamente da un carattere più (+) e lettere aggiuntive o caratteri più (+)	20	Ingresso: `web+transformer` Modello: `%{URIPROTO:protocol}` Uscita: `{"protocol":"web+transformer"}`
URIPATA	Corrisponde al componente del percorso di un URI	20	Ingresso: `/category/sub-category/product_name` Modello: `%{URIPATH:path}` Uscita: `{"path":"/category/sub-category/product_name"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
URIPARAM	Corrisponde ai parametri della query URL	5	Ingresso: `?param1=value1&param2=value2` Modello: `%{URIPARAM:url}` Uscita: `{"url":"?param1=value1&param2=value2"}`
PARAMETRO URIPATH	Corrisponde a un percorso URI seguito facoltativamente da parametri di query	5	Ingresso: `/category/sub-category/product?id=12345&color=red` Modello: `%{URIPATHPARAM:path}` Uscita: `{"path":"/category/sub-category/product?id=12345&color=red"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
URI	Corrisponde a un URI completo	5	Ingresso: `https://user:password@example.com/path/to/resource?param1=value1&param2=value2` Modello: `%{URI:uri}` Uscita: `{"path":"https://user:password@example.com/path/to/resource?param1=value1&param2=value2"}`

Schemi grok di data e ora

Modello Grok	Description	Limite massimo del pattern	Esempio
MESE	Corrisponde ai nomi completi o abbreviati dei mesi inglesi a parole intere	20	Ingresso: `Jan` Modello: `%{MONTH:month}` Uscita: `{"month": "Jan"}` Ingresso: `January` Modello: `%{MONTH:month}`

Modello Grok	Description	Limite massimo del pattern	Esempio
			Uscita: `{"month": "January"}`
NUMERO DI MESI	Corrisponde ai numeri dei mesi da 1 a 12, con lo zero iniziale opzionale per i mesi a una cifra.	20	Ingresso: `5` Modello: `%{MONTHNUM:month}` Uscita: `{"month": "5"}` Ingresso: `05` Modello: `%{MONTHNUM:month}` Uscita: `{"month": "05"}`
MONTHNUM	Corrisponde ai numeri dei mesi a due cifre compresi tra 01 e 12.	20	Ingresso: `05` Modello: `%{MONTHNUM2:month}` Uscita: `{"month": "05"}`
MESE/ GIORNO	Corrisponde al giorno del mese compreso tra 1 e 31, con zero iniziale opzionale.	20	Ingresso: `31` Modello: `%{MONTHDAY:monthDay}` Uscita: `{"monthDay":"31"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
ANNO	Corrisponde all'anno in due o quattro cifre	20	Ingresso: `2024` Modello: `%{YEAR:year}` Uscita: `{"year":"2024"}` Ingresso: `24` Modello: `%{YEAR:year}` Uscita: `{"year":"24"}`
GIORNO	Corrisponde ai nomi dei giorni completi o abbreviati.	20	Ingresso: `Tuesday` Modello: `%{DAY:day}` Uscita: `{"day":"Tuesday"}`
ORA	Corrisponde all'ora nel formato a 24 ore con uno zero iniziale opzionale (0) 0-23.	20	Ingresso: `22` Modello: `%{HOUR:hour}` Uscita: `{"hour":"22"}`
MINUTO	Corrisponde ai minuti (00-59).	20	Ingresso: `59` Modello: `%{MINUTE:min}` Uscita: `{"min":"59"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
SECOND	Corrisponde a un numero che rappresenta i secondi (0) 0-60, seguito facoltativamente da un punto decimale o due punti e da una o più cifre per i minuti frazionari	20	Ingresso: `3` Modello: `%{SECOND:second}` Uscita: `{"second":"3"}` Ingresso: `30.5` Modello: `%{SECOND:minSec}` Uscita: `{"minSec":"30.5"}` Ingresso: `30:5` Modello: `%{SECOND:minSec}` Uscita: `{"minSec":"30:5"}`
TIME	Corrisponde a un formato orario con ore, minuti e secondi nel formato (H) H:mm :(s) s. I secondi includono il secondo intercalare (0) 0-60.	20	Ingresso: `09:45:32` Modello: `%{TIME:time}` Uscita: `{"time":"09:45:32"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
DATA_US	Corrisponde a una data nel formato (M) M/ (d) d/ (yy) yy o (M) M- (d) d- (yy) yy.	20	Ingresso: `11/23/2024` Modello: `%{DATE_US:date}` Uscita: `{"date":"11/23/2024"}` Ingresso: `1-01-24` Modello: `%{DATE_US:date}` Uscita: `{"date":"1-01-24"}`
DATE_EU	Corrisponde alla data nel formato (d) d/ (M) M/ (yy) yy, (d) d- (M) M- (yy) yy o (d) d. (M) M. (yy) yy.	20	Ingresso: `23/11/2024` Modello: `%{DATE_EU:date}` Uscita: `{"date":"23/11/2024"}` Ingresso: `1.01.24` Modello: `%{DATE_EU:date}` Uscita: `{"date":"1.01.24"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
ISO8601_FUSO ORARIO	Corrisponde all'offset UTC 'Z' o all'offset del fuso orario con due punti opzionali nel formato [+-] (H) H (:) mm.	20	Ingresso: `+05:30` Modello: `%{ISO8601_TIMEZONE:tz}` Uscita: `{"tz":"+05:30"}` Ingresso: `-530` Modello: `%{ISO8601_TIMEZONE:tz}` Uscita: `{"tz":"-530"}` Ingresso: `Z` Modello: `%{ISO8601_TIMEZONE:tz}` Uscita: `{"tz":"Z"}`
ISO8601_SECONDO	Corrisponde a un numero che rappresenta i secondi (0) 0-60, seguito facoltativamente da un punto decimale o due punti e da una o più cifre per frazioni di secondo	20	Ingresso: `60` Modello: `%{ISO8601_SECOND:second}` Uscita: `{"second":"60"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
TIMESTAMP_ ISO8601	Corrisponde al formato ISO8601 datetime (yy) yy- (M) M- (d) dT (H) H:mm :((s) s) (Z| [+-] (H) H:mm) con secondi e fuso orario opzionali.	20	Ingresso: `2023-05-15T14:30:00+05:30` Modello: `%{TIMESTAMP_ISO8601:timestamp}` Uscita: `{"timestamp":"2023-05-15T14:30:00+05:30"}` Ingresso: `23-5-1T1:25+5:30` Modello: `%{TIMESTAMP_ISO8601:timestamp}` Uscita: `{"timestamp":"23-5-1T1:25+5:30"}` Ingresso: `23-5-1T1:25Z` Modello: `%{TIMESTAMP_ISO8601:timestamp}` Uscita: `{"timestamp":"23-5-1T1:25Z"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
DATE	Corrisponde a una data nel formato USA utilizzando% {DATE_US} o nel formato UE utilizzando% {DATE_EU}	20	Ingresso: `11/29/2024` Modello: `%{DATE:date}` Uscita: `{"date":"11/29/2024"}` Ingresso: `29.11.2024` Modello: `%{DATE:date}` Uscita: `{"date":"29.11.2024"}`
TIMBRO DATARIO	Corrisponde a% {DATE} seguito dallo schema% {TIME}, separato da uno spazio o da un trattino.	20	Ingresso: `29-11-2024 14:30:00` Modello: `%{DATESTAMP:dateTime}` Uscita: `{"dateTime":"29-11-2024 14:30:00"}`
TZ	Corrisponde alle abbreviazioni dei fusi orari comuni (PST, PDT, MST, MDT, CST CDT, EST, EDT, UTC).	20	Ingresso: `PDT` Modello: `%{TZ:tz}` Uscita: `{"tz":"PDT"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
DATESTAMP_ RFC822	Corrisponde a data e ora nel formato: Day MonthName (D) D (YY) YY (H) H:mm :(s) s Timezone	20	Ingresso: `Monday Jan 5 23 1:30:00 CDT` Modello: `%{DATESTAMP_RFC822 :dateTime}` Uscita: `{"dateTime":"Monday Jan 5 23 1:30:00 CDT"}` Ingresso: `Mon January 15 2023 14:30:00 PST` Modello: `%{DATESTAMP_RFC822 :dateTime}` Uscita: `{"dateTime":"Mon January 15 2023 14:30:00 PST"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
DATESTAMP_RFC2822	Corrisponde al formato RFC2822 data-ora: Day, (d) d (yy) yy MonthName (H) H:mm :(s) s Z| [+-] (H) H:mm	20	Ingresso: `Mon, 15 May 2023 14:30:00 +0530` Modello: `%{DATESTAMP_RFC2822:dateTime}` Uscita: `{"dateTime":"Mon, 15 May 2023 14:30:00 +0530"}` Ingresso: `Monday, 15 Jan 23 14:30:00 Z` Modello: `%{DATESTAMP_RFC2822:dateTime}` Uscita: `{"dateTime":"Monday, 15 Jan 23 14:30:00 Z"}`
DATESTAMP_OTHER	Corrisponde a data e ora nel formato: Giorno MonthName (d) d (H) H:mm :(s) s Fuso orario (yy) yy	20	Ingresso: `Mon May 15 14:30:00 PST 2023` Modello: `%{DATESTAMP_OTHER:dateTime}` Uscita: `{"dateTime":"Mon May 15 14:30:00 PST 2023"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
DATESTAMP_EVENTLOG	Corrisponde al formato datetime compatto senza separatori: (yy) yYmm (d) d (H) Hmm (s) s	20	Ingresso: `20230515143000` Modello: `%{DATESTAMP_EVENTLOG:dateTime}` Uscita: `{"dateTime":"20230515143000"}`

Schemi di gruppo logaritmici

Modello Grok	Description	Limite massimo del pattern	Esempio
LIVELLO DI REGISTRO	Corrisponde ai livelli di log standard in diverse lettere maiuscole e abbreviazioni, tra cui:`Alert/ALERT` ,`Trace/TRACE` ,`Debug/DEBUG` ,`Notice/NOTICE` ,`Info/INFO` ,,`Warn/Warning/WARN/WARNING` ,`Err/Error/ERR/ERROR` , `Crit/Critical/CRIT/CRITICAL` `Fatal/FATAL` `Severe/SEVERE` `Emerg/Eme`	20	Ingresso: `INFO` Modello: `%{LOGLEVEL:logLevel}` Uscita: `{"logLevel":"INFO"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
	`rgency/EMERG/ EMERGENCY`		
HTTP: DATA	Corrisponde al formato di data e ora spesso utilizzato nei file di registro. Formato: (d) d/ MonthName/(yy) yy :(H) H:mm :(s) s Fuso orario: corrisponde ai nomi completi o abbreviati dei mesi in inglese (esempio MonthName: «Jan» o «January») Fuso orario: corrisponde al modello% {INT} grok	20	Ingresso: `23/Nov/20 24:14:30:00 +0640` Modello: `%{HTTPDAT E:date}` Uscita: `{"date":"23/ Nov/2024:14:30:00 +0640"}`
TIMESTAMP SYSLOG	Corrisponde al formato della data con MonthName (d) d (H) H:mm :(s) s: corrispon de ai nomi completi o abbreviati dei mesi in inglese (esempio MonthName: «Jan» o «January»)	20	Ingresso: `Nov 29 14:30:00` Modello: `%{SYSLOGT IMESTAMP: dateTime}` Uscita: `{"dateTim e":"Nov 29 14:30:00"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
PROG	Corrisponde al nome di un programma composto da una stringa di lettere, cifre, punti, trattini bassi, barra, segno di percentuale e trattini.	20	Ingresso: `user.profile/settings-page` Modello: `%{PROG:program}` Uscita: `{"program":"user.profile/settings-page"}`
SYSLOGPROG	Corrisponde al pattern PROG grok seguito facoltativamente da un ID di processo tra parentesi quadre.	20	Ingresso: `user.profile/settings-page[1234]` Modello: `%{SYSLOGPROG:programWithId}` Uscita: `{"programWithId":"user.profile/settings-page[1234]","program":"user.profile/settings-page","pid":"1234"}`

Modello Grok	Description	Limite massimo del pattern	Esempio
SYSLOGHOST	Corrisponde a uno schema% {HOST} o% {IP}	5	Ingresso: `2001:db8:3333:4444:5555:6666:7777:8888` Modello: `%{SYSLOGHOST:ip}` Uscita: `{"ip":"2001:db8:3333:4444:5555:6666:7777:8888"}`
IMPIANTO SYSLOG	Corrisponde alla priorità syslog in formato decimale. Il valore deve essere racchiuso tra parentesi angolari (**<>**).	20	Ingresso: `<13.6>` Modello: `%{SYSLOGFACILITY:syslog}` Uscita: `{"syslog":"<13.6>","facility":"13","priority":"6"}`

Schemi di log grok comuni

È possibile utilizzare modelli grok personalizzati predefiniti per abbinare i formati di registro Apache, NGINX e Syslog Protocol (RFC 5424). Quando si utilizzano questi modelli specifici, devono essere i primi modelli nella configurazione corrispondente e nessun altro modello può precederli. Inoltre, puoi seguirli solo con esattamente un DATO. Pattern GREEDYDATA o GREEDYDATA_MULTILINE.

Pattern grok	Description	Limite massimo del modello
APACHE_ACCESS_LOG	Corrisponde ai log di accesso di Apache	1
NGINX_ACCESS_LOG	Corrisponde ai log di accesso NGINX	1
SYSLOG5424	Corrispon de ai log del protocollo Syslog (RFC 5424)	1

Di seguito sono riportati esempi validi e non validi per l'utilizzo di questi modelli di formato di registro comuni.

```
"%{NGINX_ACCESS_LOG} %{DATA}" // Valid
"%{SYSLOG5424}%{DATA:logMsg}" // Valid
"%{APACHE_ACCESS_LOG} %{GREEDYDATA:logMsg}" // Valid
"%{APACHE_ACCESS_LOG} %{SYSLOG5424}" // Invalid (multiple common log patterns used)
"%{NGINX_ACCESS_LOG} %{NUMBER:num}" // Invalid (Only GREEDYDATA and DATA patterns are
 supported with common log patterns)
"%{GREEDYDATA:logMsg} %{SYSLOG5424}" // Invalid (GREEDYDATA and DATA patterns are
 supported only after common log patterns)
```

Esempi di formati di log comuni

Esempio di log di Apache

Registro di esempio:

```
127.0.0.1 - - [03/Aug/2023:12:34:56 +0000] "GET /page.html HTTP/1.1" 200 1234
```

Trasformatore:

```
[
```

```
     {
        "grok": {
            "match": "%{APACHE_ACCESS_LOG}"
        }
    }
]
```

Output:

```
{
    "request": "/page.html",
    "http_method": "GET",
    "status_code": 200,
    "http_version": "1.1",
    "response_size": 1234,
    "remote_host": "127.0.0.1",
    "timestamp": "2023-08-03T12:34:56Z"
}
```

Esempio di log NGINX

Registro di esempio:

```
192.168.1.100 - Foo [03/Aug/2023:12:34:56 +0000] "GET /account/login.html HTTP/1.1"
 200 42 "https://www.amazon.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36"
```

Trasformatore:

```
[
     {
        "grok": {
            "match": "%{NGINX_ACCESS_LOG}"
        }
    }
]
```

Output:

```
{
    "request": "/account/login.html",
    "referrer": "https://www.amazon.com/",
```

```
    "agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/92.0.4515.131 Safari/537.36",
    "http_method": "GET",
    "status_code": 200,
    "auth_user": "Foo",
    "http_version": "1.1",
    "response_size": 42,
    "remote_host": "192.168.1.100",
    "timestamp": "2023-08-03T12:34:56Z"
}
```

Esempio di log del protocollo Syslog (RFC 5424)

Registro di esempio:

```
<165>1 2003-10-11T22:14:15.003Z mymachine.example.com evntslog - ID47
 [exampleSDID@32473 iut="3" eventSource= "Application" eventID="1011"]
[examplePriority@32473 class="high"]
```

Trasformatore:

```
[
     {
        "grok": {
            "match": "%{SYSLOG5424}"
        }
    }
]
```

Output:

```
{
  "pri": 165,
  "version": 1,
  "timestamp": "2003-10-11T22:14:15.003Z",
  "hostname": "mymachine.example.com",
  "app": "evntslog",
  "msg_id": "ID47",
  "structured_data": "exampleSDID@32473 iut=\"3\" eventSource= \"Application\" eventID=
\"1011\"",
  "message": "[examplePriority@32473 class=\"high\"]"
}
```

CSV

Il processore csv analizza i valori separati da virgole (CSV) dagli eventi di registro in colonne.

Campo	Description	Obbligatorio?	Predefinita	Limits
origine	Percorso del campo nell'evento di registro che verrà analizzato	No	`@message`	Lunghezza massima: 128. Profondità massima della chiave annidata: 3
delimitatore	Il carattere utilizzato per separare ogni colonna nell'evento di registro dei valori separati da virgole originale	No	`,`	Lunghezza massima: 1 a meno che il valore non sia o `\t` `\s`
Cita carattere	Carattere utilizzato come qualificatore di testo per una singola colonna di dati	No	`"`	Lunghezza massima: 1
columns	Elenco di nomi da utilizzare per le colonne dell'evento di registro trasformato.	No	`[column_1, column_2`	Numero massimo di colonne CSV: 100 Lunghezza massima: 128. Profondità massima della chiave annidata: 3
destinazione	Il campo principale in cui inserire le coppie chiave-valore trasformate	No	`Root node`	Lunghezza massima: 128. Profondità massima della chiave annidata: 3

L'impostazione `delimiter` su `\t` separerà ogni colonna su un carattere di tabulazione e `\t` separerà ogni colonna su un singolo carattere di spazio.

Esempio

Supponiamo che parte di un evento di registro importato abbia il seguente aspetto:

```
'Akua Mansa':28:'New York: USA'
```

Supponiamo di utilizzare solo il processore csv:

```
[
     {
        "csv": {
            "delimiter": ":",
            "quoteCharacter": "'"
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "column_1": "Akua Mansa",
  "column_2": "28",
  "column_3": "New York: USA"
}
```

Esempio 2

Supponiamo che un evento di registro importato abbia il seguente aspetto:

```
{
    "timestamp": "2024-11-23T16:03:12Z",
    "type": "user_data",
    "logMsg": "'Akua Mansa':28:'New York: USA'"
}
```

Supponiamo di analizzare l'evento come JSON, loro analizzano un campo JSON con il processore csv, specificando i nomi delle colonne e la destinazione:

```
[
    {
        "parseJSON": {}
    },
    {
        "csv": {
            "source": "logMsg",
            "delimiter": ":",
            "quoteCharacter": "'",
            "columns":["name","age","location"],
            "destination": "msg"
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
    "timestamp": "2024-11-23T16:03:12Z",
    "logMsg": "'Akua Mansa':28:'New York: USA'",
    "type": "user_data",
    "msg": {
        "name": "Akua Mansa",
        "age": "28",
        "location": "New York: USA"
    }
}
```

parseKeyValue

Utilizzate il parseKeyValueprocessore per analizzare un campo specificato in coppie chiave-valore. È possibile personalizzare il processore per analizzare le informazioni sui campi con le seguenti opzioni.

Campo	Description	Obbligatorio?	Predefinita	Limits
origine	Percorso del campo nel registro dell'evento che verrà analizzato	No	`@message`	Lunghezza massima: 128.

Campo	Description	Obbligatorio?	Predefinita	Limits
				Profondità massima della chiave annidata: 3
destinazione	Il campo di destinazione in cui inserire le coppie chiave-valore estratte	No		Lunghezza massima: 128.
FieldDelimiter	La stringa delimitatrice di campo utilizzata tra coppie chiave-valore negli eventi del registro originale	No	&	Lunghezza massima: 128.
keyValueDelimiter	La stringa delimitatrice da utilizzare tra la chiave e il valore in ogni coppia nell'evento di registro trasformato	No	=	Lunghezza massima: 128.
nonMatchValue	Un valore da inserire nel campo del valore del risultato, quando una coppia chiave-valore non viene divisa correttamente.	No		Lunghezza massima: 128.
keyPrefix	Se vuoi aggiungere un prefisso a tutte le chiavi trasformate, specificalo qui.	No		Lunghezza massima: 128.
overwriteIfExists	Se sovrascrivere il valore se la chiave di destinazione esiste già	No	`false`	

Esempio

Prendiamo il seguente esempio di evento di registro:

```
key1:value1!key2:value2!key3:value3!key4
```

Supponiamo di utilizzare la seguente configurazione del processore:

```
[
    {
        "parseKeyValue": {
            "destination": "new_key",
            "fieldDelimiter": "!",
            "keyValueDelimiter": ":",
            "nonMatchValue": "defaultValue",
            "keyPrefix": "parsed_"
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "new_key": {
    "parsed_key1": "value1",
    "parsed_key2": "value2",
    "parsed_key3": "value3",
    "parsed_key4": "defaultValue"
  }
}
```

Processori integrati per i AWS log venduti

Questa sezione contiene informazioni sui processori integrati che è possibile utilizzare con AWS i servizi che vendono log.

Indice

- ParseWAF
- parsePostGres
- parseCloudFront
- parseRoute53
- ParseVPC
- parseToOCSF

ParseWAF

Usa questo processore per analizzare i log AWS WAF venduti, prende il contenuto `httpRequest.headers` e crea chiavi JSON da ogni nome di intestazione, con il valore corrispondente. Fa lo stesso anche per. `labels` Queste trasformazioni possono rendere molto più semplice l'interrogazione dei AWS WAF log. Per ulteriori informazioni sul formato dei AWS WAF log, consulta Esempi di log per il traffico ACL web.

Questo processore accetta solo `@message` come input.

> **Important**
>
> Se si utilizza questo processore, deve essere il primo processore del trasformatore.

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
  "timestamp": 1576280412771,
  "formatVersion": 1,
  "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
  "terminatingRuleId": "STMTest_SQLi_XSS",
  "terminatingRuleType": "REGULAR",
  "action": "BLOCK",
  "terminatingRuleMatchDetails": [
    {
      "conditionType": "SQL_INJECTION",
      "sensitivityLevel": "HIGH",
      "location": "HEADER",
      "matchedData": ["10", "AND", "1"]
    }
  ],
  "httpSourceName": "-",
  "httpSourceId": "-",
  "ruleGroupList": [],
  "rateBasedRuleList": [],
  "nonTerminatingMatchingRules": [],
  "httpRequest": {
    "clientIp": "1.1.1.1",
```

```
    "country": "AU",
    "headers": [
      { "name": "Host", "value": "localhost:1989" },
      { "name": "User-Agent", "value": "curl/7.61.1" },
      { "name": "Accept", "value": "*/*" },
      { "name": "x-stm-test", "value": "10 AND 1=1" }
    ],
    "uri": "/myUri",
    "args": "",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "rid"
  },
  "labels": [{ "name": "value" }]
}
```

La configurazione del processore è la seguente:

```
[
    {
        "parseWAF": {}
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "httpRequest": {
    "headers": {
      "Host": "localhost:1989",
      "User-Agent": "curl/7.61.1",
      "Accept": "*/*",
      "x-stm-test": "10 AND 1=1"
    },
    "clientIp": "1.1.1.1",
    "country": "AU",
    "uri": "/myUri",
    "args": "",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "rid"
  },
  "labels": { "name": "value" },
```

```
  "timestamp": 1576280412771,
  "formatVersion": 1,
  "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
  "terminatingRuleId": "STMTest_SQLi_XSS",
  "terminatingRuleType": "REGULAR",
  "action": "BLOCK",
  "terminatingRuleMatchDetails": [
    {
      "conditionType": "SQL_INJECTION",
      "sensitivityLevel": "HIGH",
      "location": "HEADER",
      "matchedData": ["10", "AND", "1"]
    }
  ],
  "httpSourceName": "-",
  "httpSourceId": "-",
  "ruleGroupList": [],
  "rateBasedRuleList": [],
  "nonTerminatingMatchingRules": []
}
```


parsePostGres

Usa questo processore per analizzare i log Amazon RDS for PostgreSQL forniti, estrarre campi e convertirli in formato JSON. Per ulteriori informazioni sul formato di registro RDS for PostgreSQL, consulta File di log del database RDS for PostgreSQL.

Questo processore accetta solo come input. `@message`

> **Important**
>
> Se si utilizza questo processore, deve essere il primo processore del trasformatore.

Esempio

Prendiamo il seguente esempio di evento di registro:

```
2019-03-10 03:54:59 UTC:10.0.0.123(52834):postgres@logtestdb:[20175]:ERROR: column
 "wrong_column_name" does not exist at character 8
```

La configurazione del processore è la seguente:

```
[
    {
        "parsePostgres": {}
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "logTime": "2019-03-10 03:54:59 UTC",
  "srcIp": "10.0.0.123(52834)",
  "userName": "postgres",
  "dbName": "logtestdb",
  "processId": "20175",
  "logLevel": "ERROR"
}
```

parseCloudFront

Usa questo processore per analizzare i log Amazon CloudFront forniti, estrarre campi e convertirli in formato JSON. I valori dei campi codificati vengono decodificati. I valori interi e doppi vengono trattati come tali. Per ulteriori informazioni sul formato dei Amazon CloudFront log, consulta Configurare e utilizzare i log standard (log di accesso).

Questo processore accetta solo `@message` come input.

> **Important**
>
> Se si utilizza questo processore, deve essere il primo processore del trasformatore.

Esempio

Prendiamo il seguente esempio di evento di registro:

```
2019-12-04  21:02:31   LAX1   392    192.0.2.24    GET
 d111111abcdef8.cloudfront.net  /index.html    200    -  Mozilla/5.0%20(Windows
%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
```

```
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36  -  -  Hit
    SOX4xwn4XV6Q4rgb7XiVGOHms_BGlTAC4KyHmureZmBNrjGdRLiNIQ==
 d111111abcdef8.cloudfront.net  https  23 0.001  -  TLSv1.2    ECDHE-RSA-AES128-GCM-
SHA256    Hit    HTTP/2.0   -  -  11040  0.001  Hit    text/html  78 -  -
```

La configurazione del processore è la seguente:

```
[
    {
        "parseCloudfront": {}
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "date": "2019-12-04",
  "time": "21:02:31",
  "x-edge-location": "LAX1",
  "sc-bytes": 392,
  "c-ip": "192.0.2.24",
  "cs-method": "GET",
  "cs(Host)": "d111111abcdef8.cloudfront.net",
  "cs-uri-stem": "/index.html",
  "sc-status": 200,
  "cs(Referer)": "-",
  "cs(User-Agent)": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36",
  "cs-uri-query": "-",
  "cs(Cookie)": "-",
  "x-edge-result-type": "Hit",
  "x-edge-request-id": "SOX4xwn4XV6Q4rgb7XiVGOHms_BGlTAC4KyHmureZmBNrjGdRLiNIQ==",
  "x-host-header": "d111111abcdef8.cloudfront.net",
  "cs-protocol": "https",
  "cs-bytes": 23,
  "time-taken": 0.001,
  "x-forwarded-for": "-",
  "ssl-protocol": "TLSv1.2",
  "ssl-cipher": "ECDHE-RSA-AES128-GCM-SHA256",
  "x-edge-response-result-type": "Hit",
  "cs-protocol-version": "HTTP/2.0",
  "fle-status": "-",
  "fle-encrypted-fields": "-",
```

```
  "c-port": 11040,
  "time-to-first-byte": 0.001,
  "x-edge-detailed-result-type": "Hit",
  "sc-content-type": "text/html",
  "sc-content-len": 78,
  "sc-range-start": "-",
  "sc-range-end": "-"
}
```

parseRoute53

Usa questo processore per analizzare i log Amazon Route 53 Public Data Plane forniti, estrarre campi e convertirli in formato JSON. I valori dei campi codificati vengono decodificati. Questo processore non supporta Amazon Route 53 Resolver i log.

Questo processore accetta solo `@message` come input.

> **Important**
>
> Se si utilizza questo processore, deve essere il primo processore del trasformatore.

Esempio

Prendiamo il seguente esempio di evento di registro:

```
1.0 2017-12-13T08:15:50.235Z Z123412341234 example.com AAAA NOERROR TCP IAD12 192.0.2.0
 198.51.100.0/24
```

La configurazione del processore è la seguente:

```
[
    {
        "parseRoute53": {}
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
```

```
  "version": 1.0,
  "queryTimestamp": "2017-12-13T08:15:50.235Z",
  "hostZoneId": "Z123412341234",
  "queryName": "example.com",
  "queryType": "AAAA",
  "responseCode": "NOERROR",
  "protocol": "TCP",
  "edgeLocation": "IAD12",
  "resolverIp": "192.0.2.0",
  "ednsClientSubnet": "198.51.100.0/24"
}
```

ParseVPC

Usa questo processore per analizzare i log venduti da Amazon VPC, estrarre campi e convertirli in formato JSON. I valori dei campi codificati vengono decodificati.

Questo processore accetta solo `@message` come input.

> **Important**
>
> Se si utilizza questo processore, deve essere il primo processore del trasformatore.

Esempio

Prendiamo il seguente esempio di evento di registro:

```
2 123456789010 eni-abc123de 192.0.2.0 192.0.2.24 20641 22 6 20 4249 1418530010
 1418530070 ACCEPT OK
```

La configurazione del processore è la seguente:

```
[
    {
        "parseVPC": {}
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "version": 2,
  "accountId": "123456789010",
  "interfaceId": "eni-abc123de",
  "srcAddr": "192.0.2.0",
  "dstAddr": "192.0.2.24",
  "srcPort": 20641,
  "dstPort": 22,
  "protocol": 6,
  "packets": 20,
  "bytes": 4249,
  "start": 1418530010,
  "end": 1418530070,
  "action": "ACCEPT",
  "logStatus": "OK"
}
```

parseToOCSF

Il `parseToOCSF` processore converte i log in eventi Open Cybersecurity Schema Framework (OCSF). OCSF è uno standard aperto che fornisce uno schema comune per i dati di sicurezza, che consente una migliore interoperabilità e analisi tra diversi strumenti e piattaforme di sicurezza.

Questo processore è particolarmente utile per i flussi di lavoro di analisi della sicurezza in cui è necessario standardizzare i formati di registro di vari AWS servizi in uno schema coerente per l'analisi a valle.

Parametri

`eventSource`(richiesto)

Speciifica il AWS servizio o il processo che produce gli eventi di registro da convertire. I valori validi sono:

- `CloudTrail`- registri CloudTrail
- `Route53Resolver`- Registri Route 53 Resolver
- `VPCFlow`- Registri di flusso di Amazon VPC
- `EKSAudit`- Registri di controllo di Amazon EKS
- `AWSWAF`- registri AWS WAF

`ocsfVersion`(richiesto)

Speciifica quale versione dello schema OCSF utilizzare per gli eventi di registro trasformati. Versioni attualmente supportate: `V1.1, V1.5`

`mappingVersion` (facoltativo)

Specifica la versione di mappatura delle trasformazioni OCSF. Controlla quale logica di trasformazione viene applicata durante la conversione dei log in formato OCSF. Se non specificato, utilizza l'ultima versione disponibile al momento della creazione della politica. Le politiche esistenti non vengono aggiornate automaticamente quando vengono rilasciate nuove versioni di mappatura. Ultima versione attuale:`v1.5.0`.

Nota: Non supportato quando lo `ocsfVersion` è`V1.1`.

`source` (facoltativo)

Il percorso del campo nel registro dell'evento che desideri analizzare. Se omesso, viene analizzato l'intero messaggio di registro.

Esempio

L'esempio seguente mostra come utilizzare per `parseTo0CSF` convertire i log di flusso VPC in formato OCSF:

```
{
  "parseToOCSF": {
    "eventSource": "VPCFlow",
    "ocsfVersion": "V1.1"
  }
}
```

L'esempio seguente mostra come specificare una particolare versione di mappatura per un comportamento di trasformazione coerente:

```
{
  "parseToOCSF": {
    "eventSource": "CloudTrail",
    "ocsfVersion": "V1.5",
    "mappingVersion": "v1.5.0"
  }
}
```

Processori di mutazione delle stringhe

Questa sezione contiene informazioni sui processori di mutazione delle stringhe che è possibile utilizzare con un trasformatore di eventi di registro.

Indice

- lowerCaseString
- upperCaseString
- SplitString
- Stringa sostitutiva
- TrimString

lowerCaseString

Il `lowerCaseString` processore converte una stringa nella sua versione minuscola.

Campo	Description	Obbligatorio?	Predefinita	Limits
Con chiavi	Un elenco di chiavi da convertire in lettere minuscole	Sì		Numero massimo di voci: 10

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "INNER_VALUE"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `lowerCaseString` con`parseJSON`:

```
[
```

```
    {
        "parseJSON": {}
    },
    {
        "lowerCaseString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key": "inner_value"
  }
}
```

upperCaseString

Il `upperCaseString` processore converte una stringa nella sua versione maiuscola.

Campo	Description	Obbligatorio?	Predefinita	Limits
Con chiavi	Un elenco di chiavi da convertire in lettere maiuscole	Sì		Numero massimo di voci: 10

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `upperCaseString` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "upperCaseString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key": "INNER_VALUE"
  }
}
```

SplitString

Il `splitString` processore è un tipo di processore di mutazione di stringhe che divide un campo in un array utilizzando un carattere delimitante.

Campo	Description	Obbligatorio?	Predefinita	Limits
voci	Matrice di voci. Ogni elemento dell'array deve contenere `source` `delimiter` campi.	Sì		Numero massimo di iscrizioni: 10
origine	La chiave del valore del campo da dividere	Sì		Lunghezza massima: 128.
delimitatore	La stringa delimitatrice su cui dividere il valore del campo	Sì		Lunghezza massima: 128.

Esempio 1

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `splitString` con`parseJSON`:

```
[
     {
        "parseJSON": {}
    },
    {
         "splitString": {
            "entries": [
                {
                    "source": "outer_key.inner_key",
                    "delimiter": "_"
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key": [
      "inner",
      "value"
    ]
  }
}
```

Esempio 2

Il delimitatore su cui dividere la stringa può contenere più caratteri.

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "item1, item2, item3"
    }
}
```

La configurazione del trasformatore è la seguente:

```
[
     {
        "parseJSON": {}
    },
    {
         "splitString": {
            "entries": [
                {
                    "source": "outer_key.inner_key",
                    "delimiter": ", "
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key": [
      "item1",
      "item2",
      "item3"
    ]
  }
}
```

Stringa sostitutiva

Il `substituteString` processore è un tipo di processore di mutazione di stringhe che abbina il valore di una chiave a un'espressione regolare e sostituisce tutte le corrispondenze con una stringa sostitutiva.

Campo	Description	Obbligatorio?	Predefinita	Limits
voci	Matrice di voci. Ogni elemento dell'array deve contenere `source` `to` campi e. `from`	Sì		Numero massimo di iscrizioni: 10
origine	La chiave del campo da modificare	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3
from	La stringa di espressioni regolari da sostituire. I caratteri regex speciali come [e] devono essere evitati usando\\ quando si usano le virgolette doppie e con\ quando si usano le virgolette singole o quando sono configurati da. Console di gestione AWS Per ulteriori informazioni, vedere Class Pattern sul sito Web di Oracle. È possibile inserire un pattern `(...)` per creare un gruppo di acquisizione numerato e creare gruppi di acquisizione `(?P<group_name>...)` denominati a cui è possibile fare riferimento nel campo. `to`	Sì		Lunghezza massima: 128.
in	È possibile utilizzare la stringa da sostituire per ogni corrispondenza di Backreference ai gruppi di `from` acquisizione. Usa il modulo $n per gruppi numerati come	Sì		Lunghezza massima: 128.

Campo	Description	Obbligatorio?	Predefinita	Limits
	$1 e ${group_name} usalo per gruppi denominati come $.{my_group} >			Numero massimo di riferimenti retrospettivi: 10 Numero massimo di riferimenti retrospettivi duplicati: 2

Esempio 1

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key1": "[]",
        "inner_key2": "123-345-567",
        "inner_key3": "A cat takes a catnap."
    }
}
```

La configurazione del trasformatore è questa, utilizzata substituteString conparseJSON:

```
[
    {
        "parseJSON": {}
    },
    {
        "substituteString": {
            "entries": [
                {
                    "source": "outer_key.inner_key1",
                    "from": "\\[\\]",
                    "to": "value1"
                },
                {
                    "source": "outer_key.inner_key2",
                    "from": "[0-9]{3}-[0-9]{3}-[0-9]{3}",
                    "to": "xxx-xxx-xxx"
```

```
                },
                {
                    "source": "outer_key.inner_key3",
                    "from": "cat",
                    "to": "dog"
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key1": "value1",
    "inner_key2": "xxx-xxx-xxx",
    "inner_key3": "A dog takes a dognap."
  }
}
```

Esempio 2

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key1": "Tom, Dick, and Harry",
        "inner_key2": "arn:aws:sts::123456789012:assumed-role/MyImportantRole/
MySession"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `substituteString` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "substituteString": {
            "entries": [
```

```
                {
                    "source": "outer_key.inner_key1",
                    "from": "(\w+), (\w+), and (\w+)",
                    "to": "$1 and $3"
                },
                {
                    "source": "outer_key.inner_key2",
                    "from": "^arn:aws:sts::(?P<account_id>\\d{12}):assumed-role/(?
P<role_name>[\\w+=,.@-]+)/(?P<role_session_name>[\\w+=,.@-]+)$",
                    "to": "${account_id}:${role_name}:${role_session_name}"
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key1": "Tom and Harry",
    "inner_key2": "123456789012:MyImportantRole:MySession"
  }
}
```

TrimString

Il `trimString` processore rimuove gli spazi bianchi dall'inizio e dalla fine di una chiave.

Campo	Description	Obbligatorio?	Predefinita	Limits
Con chiavi	Un elenco di tasti da tagliare	Sì		Numero massimo di iscrizioni: 10

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
```

```
    "outer_key": {
        "inner_key": "   inner_value  "
    }
}
```

La configurazione del trasformatore è questa, utilizzata `trimString` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "trimString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key": "inner_value"
  }
}
```

Processori JSON con mutazione

Questa sezione contiene informazioni sui processori di mutazione JSON che è possibile utilizzare con un trasformatore di eventi di registro.

Indice

- AddKeys
- DeleteKeys
- moveKeys
- RenameKeys
- CopyValue
- listToMap

AddKeys

Utilizzate il `addKeys` processore per aggiungere nuove coppie chiave-valore all'evento di registro.

Campo	Description	Obbligatorio?	Predefinita	Limits
voci	Matrice di voci. Ogni elemento dell'array può contenere `key` `overwriteIfExists` campi e. `value`	Sì		Numero massimo di iscrizioni: 5
Chiave	La chiave della nuova voce da aggiungere	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3
value	Il valore della nuova voce da aggiungere	Sì		Lunghezza massima: 256
overwriteIfExists	Se lo imposti su`true`, il valore esistente viene sovrascritto se `key` già presente nell'evento. Il valore predefinito è `false`.	No	false	Nessun limite

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `addKeys` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "addKeys": {
            "entries": [
                {
                    "key": "outer_key.new_key",
                    "value": "new_value"
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key": "inner_value",
    "new_key": "new_value"
  }
}
```

DeleteKeys

Usa il `deleteKeys` processore per eliminare i campi da un evento di registro. Questi campi possono includere coppie chiave-valore.

Campo	Description	Obbligatorio?	Predefinita	Limits
Con chiavi	L'elenco delle chiavi da eliminare.	Sì	Nessun limite	Numero massimo di iscrizioni: 5

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `deleteKeys` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "deleteKeys": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {}
}
```

moveKeys

Usa il `moveKeys` processore per spostare una chiave da un campo all'altro.

Campo	Description	Obbligatorio?	Predefinita	Limits
voci	Matrice di voci. Ogni elemento dell'array può contenere `source` `overwriteIfExists` campi e. `target`	Sì		Numero massimo di iscrizioni: 5
origine	La chiave per muoversi	Sì		Lunghezza massima: 128.

Campo	Description	Obbligatorio?	Predefinita	Limits
				Profondità massima della chiave annidata: 3
target	La chiave verso cui passare	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3
overwriteIfExists	Se lo impostate su `true`, il valore esistente viene sovrascritto se `key` già presente nell'evento. Il valore predefinito è `false`.	No	false	Nessun limite

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key1": {
        "inner_key1": "inner_value1"
    },
    "outer_key2": {
        "inner_key2": "inner_value2"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `moveKeys` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "moveKeys": {
```

```
            "entries": [
                {
                    "source": "outer_key1.inner_key1",
                    "target": "outer_key2"
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key1": {},
  "outer_key2": {
    "inner_key2": "inner_value2",
    "inner_key1": "inner_value1"
  }
}
```

RenameKeys

Usa il `renameKeys` processore per rinominare le chiavi in un evento di registro.

Campo	Description	Obbligatorio?	Predefinita	Limits
voci	Matrice di voci. Ogni elemento dell'array può contenere `key` `overwriteIfExists` campi e. `target`	Sì	Nessun limite	Numero massimo di iscrizioni: 5
Chiave	La chiave per rinominare	Sì	Nessun limite	Lunghezza massima: 128.
target	Il nuovo nome della chiave	Sì	Nessun limite	Lunghezza massima: 128.

Campo	Description	Obbligatorio?	Predefinita	Limits
				Profondità massima della chiave annidata: 3
overwriteIfExists	Se lo impostate su `true`, il valore esistente viene sovrascritto se `key` già presente nell'evento. Il valore predefinito è `false`.	No	false	Nessun limite

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `renameKeys` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "renameKeys": {
            "entries": [
                {
                    "key": "outer_key",
                    "target": "new_key"
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "new_key": {
    "inner_key": "inner_value"
  }
}
```

CopyValue

Usa il `copyValue` processore per copiare i valori all'interno di un evento di registro. È inoltre possibile utilizzare questo processore per aggiungere metadati agli eventi di registro, copiando i valori delle seguenti chiavi di metadati negli eventi di registro:`@logGroupName`,,,`@logGroupStream`. `@accountId` `@regionName` Questo è illustrato nell'esempio seguente.

Campo	Description	Obbligatorio?	Predefinita	Limits
voci	Matrice di voci. Ogni elemento dell'array può contenere `source` `overwriteIfExists` campi e. `target`	Sì		Numero massimo di iscrizioni: 5
origine	La chiave da copiare	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3
target	La chiave in cui copiare il valore	Sì	Nessun limite	Lunghezza massima: 128. Profondità massima della chiave annidata: 3
overwriteIfExists	Se lo impostate su`true`, il valore esistente viene sovrascritto se `key`	No	false	Nessun limite

Campo	Description	Obbligatorio?	Predefinita	Limits
	già presente nell'evento. Il valore predefinito è `false`.			

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, utilizzata `copyValue` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "copyValue": {
            "entries": [
                {
                    "key": "outer_key.new_key",
                    "target": "new_key"
                },
                {
                    "source": "@logGroupName",
                    "target": "log_group_name"
                },
                {
                    "source": "@logGroupStream",
                    "target": "log_group_stream"
                },
                {
                    "source": "@accountId",
                    "target": "account_id"
                },
                {
```

```
                    "source": "@regionName",
                    "target": "region_name"
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key": "inner_value"
  },
  "new_key": "inner_value",
  "log_group_name": "myLogGroupName",
  "log_group_stream": "myLogStreamName",
  "account_id": "012345678912",
  "region_name": "us-east-1"
}
```

listToMap

Il `listToMap` processore prende un elenco di oggetti che contengono campi chiave e li converte in una mappa di chiavi di destinazione.

Campo	Description	Obbligatorio?	Predefinita	Limits
origine	La chiave contiene un elenco di oggetti che verranno convertiti in una mappa ProcessingEvent	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3
Chiave	La chiave dei campi da estrarre come chiavi nella mappa generata	Sì		Lunghezza massima: 128.
ValueKey	Se viene specificato, i valori specificati in questo parametro	No		Lunghezza massima: 128.

Campo	Description	Obbligatorio?	Predefinita	Limits
	verranno estratti dagli `source` oggetti e inseriti nei valori della mappa generata. Altrimenti, gli oggetti originali nell'elenco dei sorgenti verranno inseriti nei valori della mappa generata.			
target	La chiave del campo che conterrà la mappa generata	No	Nodo radice	Lunghezza massima: 128. Profondità massima della chiave annidata: 3
flatten	Un valore booleano per indicare se l'elenco verrà appiattito in singoli elementi o se i valori nella mappa generata saranno elenchi. Per impostazione predefinita, i valori per le chiavi corrispondenti saranno rappresentati in un array. `flattentrue`Impostare su per convertire l'array in un singolo valore basato sul valore di`flattenedElement` .	No	false	
Elemento appiattito	Se impostate su `flattentrue`, utilizzate `flattenedElement` per specificare quale elemento, `first` o`last`, conservare.	Obbligatorio quando `flatten` è impostato su `true`		Il valore può essere solo `first` o `last`

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ]
}
```

Trasformatore per il caso d'uso 1: `flatten` è `false`

```
[
    {
        "parseJSON": {}
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": false
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": [
        "val-a"
    ],
    "b": [
        "val-b1",
        "val-b2"
    ],
    "c": [
        "val-c"
    ]
}
```

Trasformatore per il caso d'uso 2: `flatten` è `true` ed `flattenedElement` è `first`

```
[
    {
        "parseJSON": {}
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": true,
```

```
            "flattenedElement": "first"
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": "val-a",
    "b": "val-b1",
    "c": "val-c"
}
```

Trasformatore per il caso d'uso 3: `flatten` è `true` ed `flattenedElement` è `last`

```
[
    {
        "parseJSON": {}
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": true,
```

```
            "flattenedElement": "last"
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": "val-a",
    "b": "val-b2",
    "c": "val-c"
}
```

Processori di conversione di tipi di dati

Questa sezione contiene informazioni sui processori di conversione dei tipi di dati che è possibile utilizzare con un trasformatore di eventi di registro.

Indice

- TypeConverter
- DateTimeConverter

TypeConverter

Utilizzate il `typeConverter` processore per convertire un tipo di valore associato alla chiave specificata nel tipo specificato. È un processore di casting che modifica i tipi dei campi specificati. I valori possono essere convertiti in uno dei seguenti tipi di dati:`integer`,, `double` e. `string` `boolean`

Campo	Description	Obbligatorio?	Predefinita	Limits
voci	Matrice di voci. Ogni elemento dell'array deve contenere `key` `type` campi.	Sì		Numero massimo di iscrizioni: 10
Chiave	La chiave con il valore da convertire in un tipo diverso	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3
tipo	Il tipo in cui eseguire la conversione. I valori validi sono `integer`double, `string` eboolean.	Sì		

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "name": "value",
    "status": "200"
}
```

La configurazione del trasformatore è questa, utilizzata `typeConverter` con`parseJSON`:

```
[
```

```
    {
        "parseJSON": {}
    },
    {
        "typeConverter": {
            "entries": [
                {
                    "key": "status",
                    "type": "integer"
                }
            ]
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
    "name": "value",
    "status": 200
}
```

DateTimeConverter

Utilizzate il `datetimeConverter` processore per convertire una stringa datetime in un formato specificato dall'utente.

Campo	Description	Obbligatorio?	Predefinita	Limits
origine	La chiave a cui applicare la conversione della data.	Sì		Numero massimo di iscrizioni: 10
Match Patterns	Un elenco di modelli da abbinare al campo `source`	Sì		Numero massimo di iscrizioni: 5
target	Il campo JSON in cui memorizzare il risultato.	Sì		Lunghezza massima: 128.

Campo	Description	Obbligatorio?	Predefinita	Limits
				Profondità massima della chiave annidata: 3
Formato di destinazione	Il formato datetime da utilizzare per i dati convertiti nel campo di destinazione.	No	`yyyy-MM-dd'T'HH:mm:ss.SSS'Z`	Lunghezza massima: 64
Fuso orario di origine	Il fuso orario del campo sorgente. Per un elenco dei valori possibili, consulta Java Supported Zone Ids and Offsets.	No	UTC	Lunghezza minima:1
Fuso orario di destinazione	Il fuso orario del campo di destinazione. Per un elenco dei valori possibili, consulta Java Supported Zone Ids and Offsets.	No	UTC	Lunghezza minima:1
locale	La localizzazione del campo sorgente. Per un elenco di valori possibili, vedete Metodo Locale getAvailableLocales () in Java con esempi.	Sì		Lunghezza minima:1

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{"german_datetime": "Samstag 05. Dezember 1998 11:00:00"}
```

La configurazione del trasformatore è questa, utilizzata `dateTimeConverter` con`parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "dateTimeConverter": {
            "source": "german_datetime",
            "target": "target_1",
            "locale": "de",
            "matchPatterns": ["EEEE dd. MMMM yyyy HH:mm:ss"],
            "sourceTimezone": "Europe/Berlin",
            "targetTimezone": "America/New_York",
            "targetFormat": "yyyy-MM-dd'T'HH:mm:ss z"
        }
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
    "german_datetime": "Samstag 05. Dezember 1998 11:00:00",
    "target_1": "1998-12-05T17:00:00 MEZ"
}
```

Metriche ed errori di trasformazione

CloudWatch Logs pubblica le metriche di trasformazione su. CloudWatch Queste metriche includono`TransformedLogEvents`, e. `TransformedBytes` `TransformationErrors` Per ulteriori informazioni, consulta Metriche e dimensioni del trasformatore di log.

Ogni volta che CloudWatch Logs tenta e fallisce di trasformare un evento di registro, aggiunge un campo di `@transformationError` sistema a quell'evento di registro. Quando esegui una query di CloudWatch Logs Insights, vedrai questo campo in tutti gli eventi di registro che hanno avuto un errore di trasformazione. È possibile eseguire una query per questo campo con una query in `filter ispresent(@transformationError)` modo da trovare tutti gli eventi di trasformazione non riusciti.

Accedi ai log con S3 Tables Integration

L'integrazione di S3 Tables con CloudWatch consente di accedere ai dati di log inseriti CloudWatch utilizzando motori di analisi come Amazon Athena, Amazon Redshift e strumenti di terze parti che supportano la connessione agli store Apache. Iceberg-compatible Questa integrazione ti consente di eseguire un'analisi completa dei log utilizzando strumenti di tua preferenza e di correlare i dati nei CloudWatch log con quelli non dati. CloudWatch Questa funzionalità è disponibile senza costi aggiuntivi.

Comprendere l'integrazione delle tabelle S3

Amazon S3 Tables Integration è una soluzione completamente gestita che rende disponibili i log in CloudWatch Logs come tabelle Amazon S3 gestite. Con questa integrazione, oltre alle funzionalità di Logs, ottieni una maggiore flessibilità nel modo in cui analizzi i log. CloudWatch

L'integrazione funziona creando un bucket di tabelle Amazon S3 gestito (`aws-cloudwatch`) e associando fonti di log specifiche a Amazon S3 Tables in base al nome e al tipo di origine dati (che possono essere gestiti dalla scheda Gestione dei log **>** Sorgenti dati in Logs Console). CloudWatch Una volta associati, CloudWatch i dati di Logs diventano accessibili tramite Amazon S3 Tables utilizzando il formato Apache Iceberg. Questo formato offre a vari motori di analisi un modo standardizzato per interrogare i dati in modo efficiente.

Componenti principali

Tabelle Apache Iceberg

: Il formato di tabella sottostante utilizzato da S3 Tables che fornisce l'archiviazione strutturata dei dati e consente la compatibilità con più motori di analisi.

Associazione delle fonti di dati

: Il processo di collegamento di fonti di CloudWatch log specifiche all'integrazione di S3 Tables in base a criteri di origine e tipo di dati.

Flusso di dati verso le tabelle S3

Comprendere come fluiscono i dati tra CloudWatch Logs e S3 Tables ti aiuta a pianificare l'integrazione e gestire i dati di registro in modo efficace.

Quando crei un'associazione, CloudWatch Logs invia automaticamente nuovi eventi di registro corrispondenti al nome e al tipo dell'origine dati associata a un bucket di tabelle CloudWatch-managed S3. Questi eventi vengono visualizzati nello spazio dei nomi dei log sotto la tabella corrispondente per quell'origine dati. I processi di integrazione registrano solo gli eventi ricevuti dopo la creazione dell'associazione. I dati di registro esistenti non vengono riempiti. Quando crei un'integrazione con S3 Tables, se lasci selezionata la casella di controllo Abilita tutte le fonti e i tipi di log per essere disponibili nella tabella S3, tutte le fonti di dati nel tuo account vengono automaticamente associate e inviate a S3 Tables per impostazione predefinita, incluse eventuali fonti di dati aggiunte in futuro. Per fornire solo fonti di dati specifiche a S3 Tables, deseleziona questa casella di controllo durante la creazione dell'integrazione, quindi associa singolarmente le fonti di dati che desideri includere.

La conservazione dei dati nel bucket di tabelle S3 corrisponde alla politica di conservazione impostata per il gruppo di log. Ad esempio, se imposti un gruppo di log sulla conservazione di 1 giorno, CloudWatch Logs rimuove i dati sia dai CloudWatch registri che dalla tabella S3 dopo un giorno. Quando elimini un gruppo di log o un flusso di log, CloudWatch Logs rimuove anche i dati dal bucket di tabella S3.

Quando utilizzare S3 Tables Integration

Prendi in considerazione l'utilizzo dell'integrazione di S3 Tables per correlare i dati di log con altri CloudWatch dati esterni o non, o se preferisci utilizzare altri strumenti di analisi come Amazon Athena per eseguire analisi CloudWatch sui dati di Logs. Usa questa integrazione quando hai bisogno di funzionalità che vanno oltre a quelle disponibili in Logs. CloudWatch Questa integrazione è particolarmente utile quando:

- È necessario eseguire SQL-like query complesse su grandi volumi di dati di registro
- Desideri integrare l'analisi dei log con i flussi di lavoro e gli strumenti di analisi esistenti
- Sono necessarie funzionalità complete di analisi dei log che si estendono su più fonti di dati

Non sono previsti costi aggiuntivi di archiviazione o manutenzione delle tabelle per le tabelle S3 create tramite questa integrazione, oltre ai prezzi di CloudWatch ingestione e archiviazione esistenti.

Prerequisiti

Prima di implementare l'integrazione, assicurati di disporre di quanto segue:

- Dati di CloudWatch log esistenti
- Autorizzazioni IAM appropriate per l'accesso interservizio tra CloudWatch log e tabelle S3, come descritto nella sezione seguente

autorizzazioni IAM

Per integrare CloudWatch Logs con S3 Tables, devi configurare le autorizzazioni IAM per due entità separate: l'utente o il ruolo che configura l'integrazione e il ruolo di servizio che CloudWatch Logs assume per scrivere dati su S3 Tables.

Per il ruolo o l'utente che crea l'integrazione

L'utente o il ruolo che configura l'integrazione richiede le seguenti autorizzazioni:

- `observabilityadmin:CreateS3TableIntegration`per creare l'integrazione e `logs:AssociateSourceToS3TableIntegration` aggiungere fonti
- `s3tables:CreateTableBucket`e `s3tables:PutTableBucketPolicy` per configurare il bucket da tavolo S3 `s3tables:PutTableBucketEncryption`

Per il ruolo di servizio

Allega la seguente policy IAM al ruolo del servizio IAM utilizzato da CloudWatch Logs per scrivere i dati nel bucket di tabella. Questa politica concede il permesso di scrivere nelle tabelle. Sostituisci *`aws-region`* e *`log-group-name`* con la tua AWS regione, l'ID dell'account e il nome del gruppo di registro. *`123456789012`*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:integrateWithS3Table"
            ],
            "Resource": ["arn:aws:logs:aws-region:123456789012:log-group:log-group-
name"],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "123456789012"
```

```
                }
            }
        }
    ]
}
```

Allega la seguente policy di fiducia al ruolo del servizio IAM che CloudWatch Logs assumerà per scrivere i dati di registro su S3 Tables. Questo ruolo viene creato o selezionato durante la configurazione dell'integrazione. Le condizioni limitano il ruolo in modo che CloudWatch Logs possa assumerlo solo per l'account e il gruppo di log specificati. Sostituisci *aws-region* e *log-group-name* con la tua AWS regione, l'ID dell'account e il nome del gruppo di registro. *123456789012*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:aws-region:123456789012:log-
group:log-group-name"]
                }
            }
        }
    ]
}
```

Politica delle chiavi KMS (per dati crittografati)

Se utilizzi una chiave gestita dal cliente per crittografare i dati di registro, devi concedere al responsabile del CloudWatch servizio e al responsabile del servizio di manutenzione di S3 Tables l'accesso alla chiave. Aggiungi le seguenti istruzioni alla tua politica sulle chiavi KMS. Sostituisci i valori segnaposto con il tuo Account AWS ID, la regione, l'ID della chiave KMS e l'ARN della tabella o del bucket di tabella S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableSystemTablesKeyUsage",
            "Effect": "Allow",
            "Principal": {
                "Service": "systemtables.cloudwatch.amazonaws.com"
            },
            "Action": [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:aws-region:123456789012:key/key-id",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        },
        {
            "Sid": "EnableKeyUsage",
            "Effect": "Allow",
            "Principal": {
                "Service": "maintenance.s3tables.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:aws-region:123456789012:key/key-id",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "<table-or-table-bucket-arn>/*"
                }
            }
        }
    ]
}
```

Nozioni di base

Per iniziare con S3 Tables Integration, devi configurare l'integrazione tra Logs e S3 Tables. CloudWatch Questo processo prevede la configurazione delle associazioni delle fonti di dati e l'impostazione delle autorizzazioni IAM appropriate.

Per creare un'integrazione con S3 Tables

1. Apri la console CloudWatch Logs all'indirizzo https://console.aws.amazon.com/cloudwatch/».
2. Scegli Impostazioni, Globale, Crea integrazione con tabelle S3.
3. Personalizza il modo in cui i log verranno crittografati in S3 Tables e il ruolo che CloudWatch Logs utilizzerà per scrivere i log in S3 Tables.
4. Se desideri che tutte le fonti di dati vengano associate automaticamente all'integrazione, lascia selezionata la casella di controllo Abilita tutte le fonti e i tipi di log nella tabella S3 (è selezionata per impostazione predefinita). Se desideri associare solo fonti di dati specifiche, deseleziona questa casella di controllo.
5. Scegli Crea integrazione con tabelle S3.

> **Note**
>
> Se hai selezionato Abilita la disponibilità di tutte le sorgenti e i tipi di log nella tabella S3 durante la creazione, tutte le fonti di dati vengono associate automaticamente, incluse tutte le fonti di dati aggiunte in futuro. Puoi fermarti qui. I seguenti passaggi sono necessari solo se hai deselezionato la casella di controllo e desideri associare fonti di dati specifiche.

Per associare le fonti a una S3 Table Integration

1. Apri la console CloudWatch Logs all'indirizzo https://console.aws.amazon.com/cloudwatch/».
2. Scegli Impostazioni, Globale, Gestisci l'integrazione delle tabelle S3.
3. Scegli Associa origine dati.
4. Seleziona il nome dell'origine dati e il tipo di origine dati per cui desideri abilitare l'integrazione.
5. Scegli Associa origine dati.

Per associare le fonti a un'integrazione di tabelle S3 dalla pagina di gestione dei log

1. Apri la console CloudWatch Logs all'indirizzo https://console.aws.amazon.com/cloudwatch/».
2. Scegli Log Management nel pannello di navigazione.
3. Seleziona la scheda Fonti dati.
4. Scegli il nome dell'origine dati e il tipo di origine dati che desideri integrare.
5. Scegli le azioni relative all'origine dei dati.
6. Seleziona Associa a S3 Tables Integration.
7. Esamina le fonti di dati, quindi scegli Associa origine dati.

Prima di poter utilizzare i dati devi eseguire i seguenti 3 passaggi:

1. Integrazione delle tabelle Amazon Amazon S3 AWS con i servizi di analisi - Utilizzo della console Amazon S3
2. Configura i permessi di Lake Formation
3. Connect con gli strumenti di analisi

Integrazione delle tabelle Amazon Amazon S3 con AWS servizi di analisi - Utilizzo della console Amazon S3 (Link)

Per abilitare l'integrazione di S3 Tables utilizzando la console S3

1. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Nel pannello di navigazione a sinistra, scegli Bucket di tabelle.
3. Scegli l'opzione Abilita integrazione in alto.
4. La prima volta che integri i table bucket in qualsiasi regione, Amazon Amazon S3 crea un nuovo ruolo di servizio IAM per tuo conto. Questo ruolo consente a Lake Formation di accedere a tutti i table bucket del tuo account e di federare l'accesso ai tuoi tavoli in AWS Glue Data Catalog.

Configura i permessi di Lake Formation

Sebbene CloudWatch Logs disponga dell'autorizzazione per scrivere sulla tabella (configurata nei passaggi precedenti), gli utenti e i ruoli di analisi non dispongono automaticamente

dell'autorizzazione a leggere i dati. È necessario concedere esplicitamente l'accesso utilizzando AWS Lake Formation. Devi farlo per ogni principale IAM a cui desideri fornire l'accesso alla tabella.

Per concedere l'accesso alle query a utenti o ruoli

È necessario concedere le autorizzazioni SELECT e DESCRIBE ai principali IAM (utenti o ruoli) che eseguiranno le query in Athena o Redshift.

1. Apri la console AWS Lake Formation.
2. Nel riquadro di navigazione, sotto Autorizzazioni, scegli Autorizzazioni Data lake.
3. Scegliere Concedi.
4. Principi: seleziona gli utenti o i ruoli IAM che richiedono l'accesso (ad esempio, i tuoi analisti di dati o il ruolo di amministratore che stai utilizzando attualmente).
5. LF-Tags o risorse del catalogo: seleziona le risorse di Named Data Catalog.
6. Database e tabelle:
   - Seleziona il bucket S3 Table creato dall' CloudWatch integrazione ()`aws-cloudwatch`.
   - Seleziona la tabella specifica associata alla tua fonte di dati (opzionale).
7. Autorizzazioni per la tabella: seleziona Seleziona e descrivi.
8. Scegliere Concedi.

> **Note**
>
> Se riscontri errori «Accesso negato» durante l'interrogazione dei log in Athena, assicurati che l'utente che esegue la query disponga sia delle autorizzazioni IAM appropriate per Athena sia delle autorizzazioni Lake Formation definite sopra.

Scopri di più sui permessi di Lake Formation su https://docs.aws.amazon.com/lake-formation/latest/dg/granting-catalog-permissions.html.

Connect con gli strumenti di analisi

Una volta concesse le autorizzazioni, puoi configurare il tuo servizio di analisi preferito per interrogare le tabelle S3. Le tabelle S3 utilizzano il formato Apache Iceberg, supportato nativamente da Amazon Athena, Amazon Redshift e Amazon EMR.

Per interrogare i dati di log in Amazon Athena

Amazon Athena interagisce con S3 Tables tramite il catalogo Amazon S3 Tables.

Per configurare Athena per interrogare i dati di registro

1. Apri la console Amazon Athena all'indirizzo. https://console.aws.amazon.com/athena/
2. Nell'editor di query, seleziona il catalogo Amazon S3 Tables dal menu a discesa delle origini dati.
3. Se non vedi il catalogo, assicurati di aver completato i passaggi di autorizzazione di Lake Formation sopra riportati per il tuo ruolo utente specifico.
4. Una volta selezionato il catalogo, le tabelle di registro verranno visualizzate nell'elenco del database. È ora possibile eseguire query SQL standard sui dati di registro.

Query di esempio: `SELECT * FROM "amazon_vpc__flow" LIMIT 100;`

Scopri di più sulla connessione dei servizi di Analytics con S3 Tables all'indirizzo https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-tables-integrating-aws.html.

Creazione di parametri da log eventi mediante filtri

Puoi cercare e filtrare i dati di log che entrano in CloudWatch Logs creando uno o più filtri metrici. I filtri metrici definiscono i termini e i modelli da cercare nei dati di log quando vengono inviati ai CloudWatch Logs. CloudWatch Logs utilizza questi filtri metrici per trasformare i dati di registro in CloudWatch metriche numeriche su cui è possibile rappresentare graficamente o impostare un allarme.

Quando si crea un parametro da un filtro di log, è anche possibile scegliere di assegnare dimensioni e un'unità al parametro. Se si sceglie un'unità, assicurarsi di specificare quella corretta quando si crea il filtro. In seguito, la modifica dell'unità per il filtro non avrà alcun effetto.

Se hai configurato AWS Organizations e stai lavorando con gli account dei membri, puoi utilizzare la centralizzazione dei log per raccogliere i dati di log dagli account di origine in un account di monitoraggio centrale.

Quando si lavora con gruppi di log centralizzati, è possibile utilizzare queste dimensioni dei campi di sistema durante la creazione di filtri metrici.

- `@aws.account`- Questa dimensione rappresenta l'ID dell' AWS account da cui ha avuto origine l'evento di registro.
- `@aws.region`- Questa dimensione rappresenta la AWS regione in cui è stato generato l'evento di registro.

Queste dimensioni aiutano a identificare l'origine dei dati di registro, consentendo un filtraggio e un'analisi più granulari delle metriche derivate dai log centralizzati. Per ulteriori informazioni, consulta Cross-account Centralizzazione dei log interregionali.

Se un gruppo di log con un abbonamento utilizza la trasformazione dei log, il pattern di filtro viene applicato alle versioni trasformate degli eventi di log. Per ulteriori informazioni, consulta Trasforma i log durante l'ingestione.

> **Note**
>
> I filtri metrici sono supportati solo per i gruppi di log nella classe di log Standard. Per ulteriori informazioni sulle classi di log, vedereClassi di registro.

Puoi utilizzare qualsiasi tipo di CloudWatch statistica, incluse le statistiche percentili, quando visualizzi queste metriche o imposti gli allarmi.

> **Note**
>
> Le statistiche dei percentili sono supportate per un parametro solo se nessuno dei valori del parametro è negativo. Se configuri il filtro parametri in modo che possa indicare i numeri negativi, le statistiche dei percentili non saranno disponibili per i parametri quando includono numeri negativi come valore. Per ulteriori informazioni, consulta Percentili.

Filtri non filtrano retroattivamente i dati. I filtri pubblicano solo i punti di dati dei parametri per eventi che accadono dopo la creazione del filtro. Quando si verifica un modello di filtro, l'anteprima dei risultati del filtro mostra fino alle prime 50 righe di log corrispondenti a fini di convalida. Se il timestamp sui risultati filtrati è precedente all'ora di creazione della metrica, non viene visualizzato alcun registro.

> **Note**
>
> I filtri metrici garantiscono l'invio almeno una volta degli eventi di registro alla metrica specificata, mentre occasionalmente possono verificarsi consegne duplicate.

Indice

- Concetti
- Sintassi del modello di filtro per i filtri di parametri
- Creazione di filtri di parametri
- Elencazione di filtri di parametri
- Eliminazione di un filtro di parametri

Concetti

Ogni filtro parametri è formato dai seguenti elementi chiave:

Valore predefinito

Il valore riportato per il filtro di parametri durante un periodo di tempo in cui i log vengono inseriti ma non ne vengono individuati di corrispondenti. Impostandolo su 0, ti assicuri che i dati siano riportati durante ogni periodo, evitando parametri "instabili" con periodi di assenza di dati. Se nessun log viene inserito nell'arco di tempo di un minuto, allora nessun valore viene segnalato.

Se si assegnano dimensioni a un parametro creato da un filtro di parametri, non è possibile assegnare un valore predefinito per tale parametro.

dimensioni

Le dimensioni sono le coppie chiave-valore che definiscono ulteriormente un parametro. È possibile assegnare dimensioni al parametro creato da un filtro di parametri. Poiché le dimensioni fanno parte dell'identificatore univoco di una metrica, ogni volta che viene estratta una name/value coppia univoca dai log, si crea una nuova variante di tale metrica.

Modello di filtro

Una descrizione simbolica di come CloudWatch i log devono interpretare i dati in ogni evento di registro. Ad esempio, una voce di log potrebbe contenere timestamp, indirizzi IP, stringhe e così via. Utilizza il modello per specificare che cosa ricercare nei file di log.

nome parametro

Il nome della CloudWatch metrica in cui devono essere pubblicate le informazioni di registro monitorate. Ad esempio, puoi pubblicare in base a una metrica denominata. ErrorCount

Spazio dei nomi del parametro:

Lo spazio dei nomi di destinazione della nuova metrica. CloudWatch

Valore del parametro:

Il valore numerico da pubblicare nel parametro ogni volta che viene trovato un log corrispondente. Ad esempio, se stai contando le occorrenze di un determinato termine come "Error", il valore sarà "1" per ogni ricorrenza. Se stai contando i byte trasferiti, puoi incrementare con il numero effettivo di byte presenti nel log eventi.

Sintassi del modello di filtro per i filtri di parametri

> **Note**
>
> In che modo i filtri metrici differiscono dalle query di CloudWatch Logs Insights
> I filtri metrici differiscono dalle query di CloudWatch Logs Insights in quanto un valore numerico specificato viene aggiunto a un filtro metrico ogni volta che viene trovato un log corrispondente. Per ulteriori informazioni, consulta Configurazione di valori di parametri per un filtro di parametri.
> Per informazioni su come interrogare i tuoi gruppi di log con il linguaggio di query Amazon CloudWatch Logs Insights, consulta. CloudWatch Sintassi delle interrogazioni in linguaggio Logs Insights
> Esempi di modelli di filtro generici
> Per ulteriori informazioni sulla sintassi del modello di filtro generico applicabile ai filtri di parametri, nonché ai filtri di sottoscrizione e ai filtri di log eventi, consulta Sintassi del modello di filtro per filtri di parametri, filtri di sottoscrizione e filtri di log eventi, che include i seguenti esempi:
>
> - Sintassi delle espressioni regolari (regex) supportate
> - Corrispondenza dei termini nei log eventi non strutturati
> - Corrispondenza dei termini negli log eventi JSON
> - Corrispondenza dei termini in log eventi delimitati da spazi

I filtri metrici consentono di cercare e filtrare i dati di log che entrano in CloudWatch Logs, estrarre osservazioni metriche dai dati di log filtrati e trasformare i punti dati in una CloudWatch metrica Logs. Definisci i termini e i modelli da cercare nei dati di log quando vengono inviati a Logs. CloudWatch I filtri di parametro vengono assegnati ai gruppi di log e tutti i filtri assegnati a un gruppo di log vengono applicati ai relativi flussi di log.

Quando un modello di filtro trova un termine corrispondente, aumenta il conteggio del parametro di un valore numerico specificato. Ad esempio, puoi creare un filtro di parametri per ricercare e contare l'occorrenza della parola ERROR nei tuoi log eventi.

È possibile assegnare unità e dimensioni ai parametri. Ad esempio, se crei un filtro di parametri che conta l'occorrenza della parola ERROR nei tuoi log eventi, puoi specificare una dimensione chiamata

`ErrorCode` per mostrare il numero totale di log eventi che contengono la parola ERROR e filtrare i dati per codici di errore segnalati.

> **Tip**
>
> Quando assegni un'unità di misura a un parametro, assicurati di specificare quella corretta. Se cambi l'unità in un secondo momento, la modifica potrebbe non avere effetto. Per l'elenco completo delle unità CloudWatch supportate, consulta MetricDatum Amazon CloudWatch API Reference.

Argomenti

- Configurazione di valori di parametri per un filtro di parametri
- Pubblicazione di dimensioni con parametri da valori in JSON o log eventi delimitati da spazi
- Utilizzo di valori nei log eventi per incrementare il valore di un parametro

Configurazione di valori di parametri per un filtro di parametri

Quando crei un filtro di parametri, definisci il modello di filtro e specifichi il valore del parametro e il valore predefinito. Puoi impostare i valori dei parametri su numeri, identificatori di nome o identificatori numerici. Se non specifichi un valore predefinito, CloudWatch non riporterà i dati quando il filtro metrico non trova una corrispondenza. Consigliamo di specificare un valore predefinito, anche se il valore è 0. L'impostazione di un valore predefinito aiuta a CloudWatch riportare i dati in modo più accurato e CloudWatch impedisce l'aggregazione di metriche irregolari. CloudWatch aggrega e riporta i valori delle metriche ogni minuto.

Quando il filtro di parametri trova una corrispondenza nei log eventi, aumenta il conteggio del parametro in base al valore dello stesso. Se il filtro metrico non trova una corrispondenza, CloudWatch riporta il valore predefinito della metrica. Ad esempio, il gruppo di log pubblica due record ogni minuto, il valore del parametro è 1 e quello predefinito è 0. Se il filtro di parametri rileva corrispondenze in entrambi i record di log nel primo minuto, il valore del parametro per quel minuto è pari a 2. Se il filtro di parametri non trova corrispondenze in entrambi i record durante il secondo minuto, il valore predefinito per quel minuto è 0. Se assegni dimensioni ai parametri generati dai filtri di parametri, non puoi specificare i valori predefiniti per tali parametri.

Puoi anche impostare un filtro di parametri per incrementare un parametro con un valore estratto da un log eventi, anziché un valore statico. Per ulteriori informazioni, consulta Utilizzo di valori nei log eventi per incrementare il valore di un parametro.

Pubblicazione di dimensioni con parametri da valori in JSON o log eventi delimitati da spazi

Puoi utilizzare la CloudWatch console o la AWS CLI per creare filtri metrici che pubblicano dimensioni con metriche generate da JSON e da eventi di log delimitati da spazi. Le dimensioni sono coppie di name/value valori e sono disponibili solo per JSON e modelli di filtro delimitati da spazi. Puoi creare filtri di parametri JSON e delimitati da spazi con un massimo di tre dimensioni. Per ulteriori informazioni sulle dimensioni e su come assegnare dimensioni ai parametri, consulta le seguenti sezioni:

- Dimensioni nella guida per l'utente di Amazon CloudWatch
- Esempio: estrai i campi da un log di Apache e assegna le dimensioni nella Amazon CloudWatch Logs User Guide

> **Important**
>
> Le dimensioni contengono valori che raccolgono addebiti identici ai parametri personalizzati. Per evitare addebiti imprevisti, non specificare come dimensioni campi ad alta cardinalità, ad esempio `IPAddress` o `requestID`.
> Se estrai parametri dai log eventi, ti verranno fatturati come parametri personalizzati. Per evitare di riscuotere addebiti accidentali elevati, Amazon potrebbe disattivare il filtro metrico se genera 1000 name/value coppie diverse per dimensioni specificate in un determinato periodo di tempo.
> Puoi creare allarmi di fatturazione per ricevere una notifica sugli addebiti stimati. Per ulteriori informazioni, consulta Creazione di un allarme di fatturazione per il monitoraggio dei costi di AWS stimati.

Pubblicazione di dimensioni con parametri dal log eventi JSON

Gli esempi seguenti contengono frammenti di codice che descrivono come specificare le dimensioni in un filtro di parametri JSON.

Example: JSON log event

```
{
  "eventType": "UpdateTrail",
  "sourceIPAddress": "111.111.111.111",
  "arrayKey": [
        "value",
        "another value"
  ],
  "objectList": [
       {"name": "a",
         "id": 1
       },
       {"name": "b",
         "id": 2
       }
  ]

}
```

> **Note**
>
> Se testi il filtro di parametri di esempio con il log eventi JSON di esempio, devi inserire il log JSON di esempio su una singola riga.

Example: Metric filter

Il filtro di parametri incrementa il parametro ogni volta che un log eventi JSON contiene le proprietà `eventType` e `"sourceIPAddress"`.

```
{ $.eventType = "*" && $.sourceIPAddress != 123.123.* }
```

Quando crei un filtro di parametri JSON, puoi specificare una qualsiasi proprietà del filtro di parametri come dimensione. Ad esempio, per impostare `eventType` come dimensione, segui questa procedura:

```
"eventType" : $.eventType
```

La metrica di esempio contiene una dimensione denominata `"eventType"` e il valore della dimensione nel log eventi di esempio è `"UpdateTrail"`.

Pubblicazione di dimensioni con parametri da log eventi delimitati da spazi

Gli esempi seguenti contengono frammenti di codice che descrivono come specificare le dimensioni in un filtro di parametri delimitato da spazi.

Example: Space-delimited log event

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

Example: Metric filter

```
[ip, server, username, timestamp, request, status_code, bytes > 1000]
```

Il filtro di parametri incrementa il parametro quando un log eventi delimitato da spazi include uno dei campi specificati nel filtro. Ad esempio, il filtro di parametri trova i seguenti campi e valori nell'esempio del log eventi delimitato da spazi.

```
{
   "$bytes": "1534",
   "$status_code": "404",

   "$request": "GET /index.html HTTP/1.0",
   "$timestamp": "10/Oct/2000:13:25:15 -0700",
   "$username": "frank",
   "$server": "Prod",
   "$ip": "127.0.0.1"
}
```

Quando crei un filtro di parametri delimitato da spazi, puoi specificare uno dei campi nel filtro di parametri come dimensione. Ad esempio, per impostare `server` come dimensione, segui questa procedura:

```
"server" : $server
```

Il filtro di parametri di esempio ha una dimensione denominata `server` e il valore della dimensione nel log eventi di esempio è `"Prod"`.

Example: Match terms with AND (&&) and OR (||)

Puoi utilizzare gli operatori logici AND ("&&") e OR ("||") per creare filtri di parametri delimitati da spazi che contengono condizioni. Il seguente modello di filtro restituisce log eventi in cui la prima parola è ERROR o WARNING.

```
[w1=ERROR || w1=%WARN%, w2]
```

Utilizzo di valori nei log eventi per incrementare il valore di un parametro

Puoi creare filtri di parametri che pubblicano i valori numerici trovati nei log eventi. La procedura in questa sezione utilizza il seguente filtro di parametri di esempio per mostrare come è possibile pubblicare un valore numerico in un log eventi JSON in un parametro.

```
{ $.latency = * } metricValue: $.latency
```

Creazione di un filtro di parametri che pubblica un valore in un log eventi

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Log, quindi Gruppi di log.
3. Seleziona o crea un gruppo di log.

   Per informazioni su come creare un gruppo di log, consulta Create a log group in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.
4. Scegli Actions (Operazioni) e quindi Create metric filter (Crea filtro parametri).
5. In Modello di filtro, digita **`{ $.latency = * }`**, quindi scegli Next (Successivo).

6. In Metric Name (Nome parametro), digita myMetric.
7. Per Valore parametro, immettere **`$.latency`**.
8. (Facoltativo) In Valore predefinito immetti 0, quindi scegli Next (Successivo).

   Consigliamo di specificare un valore predefinito, anche se il valore è 0. L'impostazione di un valore predefinito aiuta a CloudWatch riportare i dati in modo più accurato e CloudWatch impedisce l'aggregazione di metriche irregolari. CloudWatch aggrega e riporta i valori delle metriche ogni minuto.
9. Scegli Crea filtro parametri.

Il filtro di parametri di esempio corrisponde al termine `"latency"` nel log eventi JSON di esempio e pubblica un valore numerico pari a 50 per il parametro myMetric.

```
{
"latency": 50,
"requestType": "GET"
}
```

Creazione di filtri di parametri

La procedura e gli esempi seguenti mostrano come creare filtri di parametri.

Esempi

- Creazione di un filtro di parametri per un gruppo di log
- Esempio: conteggio di log eventi
- Esempio: conteggio delle occorrenze di un termine
- Esempio: conteggio di codici HTTP 404
- Esempio: conteggio di codici HTTP 4xx
- Esempio: estrazione di campi da un log di Apache e assegnare dimensioni

Creazione di un filtro di parametri per un gruppo di log

Per creare un filtro di parametri per un gruppo di log, procedi nel seguente modo. Il parametro non sarà visibile finché non saranno disponibili alcuni punti dati.

Per creare un filtro metrico utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Log, quindi Gruppi di log.
3. Scegli il nome del gruppo di log.
4. Scegli `Actions` e quindi Crea filtro parametri.
5. Per Modello di filtro inserisci il modello di filtro. Per ulteriori informazioni, consulta Sintassi del modello di filtro per filtri di parametri, filtri di sottoscrizione, filtri di log eventi e Live Tail.
6. (Facoltativo) Se si utilizzano gruppi di log centralizzati, in Criteri di selezione dei filtri, è possibile specificare i filtri in base all'account di origine (`@aws.account`), alla regione di origine (`@aws.region`) o a entrambe le condizioni.
7. (Facoltativo) Per testare il modello di filtro, in Test Pattern (Modello di test), inserisci uno o più log eventi per testare il modello. Ogni log eventi deve essere formattato su una riga. Le interruzioni di riga vengono utilizzate per separare i log eventi nel riquadro Messaggi del log eventi.
8. Scegli Successivo e poi inserisci un nome per il filtro.
9. In Dettagli delle metriche, per lo spazio dei nomi delle metriche, inserisci un nome per lo spazio dei CloudWatch nomi in cui verrà pubblicata la metrica. Se questo spazio dei nomi non esiste già, assicurati che sia selezionato Crea nuovo.
10. Per Metric name (Nome parametro) inserisci un nome per il nuovo parametro.
11. In Metric value (Valore del parametro), se il filtro di parametri conta le occorrenze delle parole chiave nel filtro, inserisci 1. In questo modo il parametro viene incrementato di 1 per ogni log eventi che include una delle parole chiave.

    In alternativa, inserisci un token come **`$size`**. In questo modo il parametro viene incrementato in base al valore del numero nel campo `size` per ogni log eventi che contiene un campo `size`.
12. (Facoltativo) Per Unit (Unità), seleziona un'unità da assegnare al parametro. Se non specifichi un'unità, l'unità viene impostata come `None`.
13. (Facoltativo) Inserisci i nomi e i token per fino a tre dimensioni per il parametro. Se assegni dimensioni ai parametri creati dai filtri di parametri, non puoi assegnare i valori predefiniti per tali parametri.

> **Note**
>
> Le dimensioni sono supportate solo nei filtri JSON o di parametri delimitati da spazi.

14. Scegli Crea filtro parametri. Puoi trovare il filtro dei parametri che hai creato nel riquadro di navigazione. Scegli Log e quindi il gruppo di log. Scegli il nome del gruppo di log per cui hai creato il filtro dei parametri, quindi seleziona il tab Filtri dei parametri.

Esempio: conteggio di log eventi

Il tipo più semplice di monitoraggio di eventi di log è contare il numero di eventi di log che si verificano. Potresti voler eseguire questa operazione per mantenere un conteggio di tutti gli eventi, per creare un monitor in stile "battito cardiaco" o anche solo per provare la creazione di filtri di parametro.

Nel seguente esempio CLI, un filtro metrico chiamato MyAppAccessCount viene applicato al gruppo di log per MyApp/access.log creare la metrica EventCount nel namespace. CloudWatch MyNamespace Il filtro è configurato corrispondere a qualsiasi log eventi e per incrementare il parametro con "1".

Per creare un filtro metrico utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Scegliere il nome di un gruppo di log.
4. Scegli `Actions`, Crea filtro parametri.
5. Lasciare vuoti i campi Modello di filtro e Seleziona i dati di log per il test .
6. Scegliere Avanti, quindi per Nome filtro, digitare **`EventCount`**.
7. In Metric Details (Dettagli parametro), per Metric Namespace (Spazio dei nomi del parametro), inserisci **`MyNameSpace`**.
8. Per Metric Name (Nome parametro) digita **`MyAppEventCount`**.
9. Verificare che il Valore parametro sia 1. Questo specifica che il conteggio viene incrementato di 1 per ogni log eventi.
10. In Valore predefinito immettere 0, quindi scegliere Avanti. La specificazione di un valore predefinito garantisce che i dati siano riportati anche durante i periodi in cui non si verificano eventi di log, impedendo parametri instabili in cui i dati talvolta non esistono.
11. Scegli Crea filtro parametri.

Per creare un filtro metrico utilizzando AWS CLI

Al prompt dei comandi, esegui il comando seguente:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name EventCount \
  --filter-pattern " " \
  --metric-transformations \
  metricName=MyAppEventCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Puoi testare questa nuova policy pubblicando qualsiasi dati di evento. Dovresti vedere i punti dati pubblicati nella metrica MyAppAccessEventCount.

Per pubblicare i dati degli eventi utilizzando AWS CLI

Al prompt dei comandi, esegui il comando seguente:

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="Test event 1" \
    timestamp=1394793518000,message="Test event 2" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Esempio: conteggio delle occorrenze di un termine

Gli eventi di log includono spesso messaggi importanti che desideri contare, forse per quel che concerne il successo o il fallimento delle operazioni. Ad esempio, se una determinata operazione non riesce, potrebbe verificarsi un errore che viene registrato in un file di log. Potresti voler monitorare queste voci per comprendere l'andamento dei tuoi errori.

Nell'esempio sottostante, una filtro parametri viene creato per monitorare il termine Error. La policy è stata creata e aggiunta al gruppo di log. MyApp/message.log CloudWatch Logs pubblica un punto dati della metrica CloudWatch personalizzata ErrorCount nello spazio dei MyApp/message.log nomi con il valore «1" per ogni evento contenente Error. Se nessun evento contiene la parola Error, il valore pubblicato è 0. Quando rappresentate graficamente questi dati nella CloudWatch console, assicuratevi di utilizzare la statistica della somma.

Dopo aver creato un filtro metrico, puoi visualizzare la metrica nella console. CloudWatch Quando si seleziona il parametro da visualizzare, selezionare lo spazio dei nomi del parametro che corrisponde al nome del gruppo di log. Per ulteriori informazioni, consulta Visualizzazione dei parametri disponibili.

Per creare un filtro metrico utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Scegli il nome del gruppo di log.
4. Scegliere Operazioni, Crea filtro parametri.
5. In Modello di filtro, immettere **`Error`**.

   > **Note**
   >
   > Tutte le voci in Filtra Pattern (Modello File) fanno distinzione tra lettere maiuscole e minuscole.

6. (Facoltativo) Per testare il modello di filtro, in Test Pattern (Modello di test), inserisci uno o più log eventi da utilizzare per testare il modello. Ogni log eventi deve essere all'interno di una riga, in quanto le interruzioni di riga vengono utilizzate per separare i log eventi nella casella Log event messages (Messaggi di registro eventi).
7. Scegliere Avanti, quindi nella pagina Assegna parametro, per Nome filtro, digitare **`MyAppErrorCount`**.
8. In Metric Details (Dettagli parametro), per Metric Namespace (Spazio dei nomi del parametro), inserisci MyNameSpace.
9. Per Metric Name (Nome parametro) digita ErrorCount.
10. Verificare che il Valore parametro sia 1. Questo specifica che il conteggio viene incrementato di 1 per ogni log eventi contente "Error".
11. Per Valore predefinito digitare 0, quindi scegliere Avanti.
12. Scegli Crea filtro parametri.

Per creare un filtro metrico utilizzando AWS CLI

Al prompt dei comandi, esegui il comando seguente:

```
aws logs put-metric-filter \
  --log-group-name MyApp/message.log \
  --filter-name MyAppErrorCount \
  --filter-pattern 'Error' \
  --metric-transformations \
```

```
      metricName=ErrorCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Puoi verificare questa nuova policy pubblicando eventi che contengano la parola "Error" nel messaggio.

Per pubblicare eventi utilizzando il AWS CLI

Al prompt dei comandi, esegui il comando seguente. Nota che i modelli fanno distinzione tra lettere maiuscole e minuscole.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="This message contains an Error" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Esempio: conteggio di codici HTTP 404

Usando CloudWatch Logs, puoi monitorare quante volte i tuoi server Apache restituiscono una risposta HTTP 404, che è il codice di risposta per la pagina non trovata. Potresti voler monitorare questo per comprendere quanto spesso i visitatori del tuo sito non sono in grado di individuare le risorse ricercate. Supponiamo che i tuoi record di log siano strutturati per includere le seguenti informazioni per ogni log eventi (visita del sito):

- Indirizzo IP del richiedente
- Identità RFC 1413
- Username
- Time stamp
- Richiedi metodo con risorsa di richiesta e protocollo
- Codice di risposta HTTP per richiesta
- Byte trasferiti nella richiesta

Un esempio del genere potrebbe apparire come segue:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 404 2326
```

Potresti specificare una regola che cerca di confrontare gli eventi di tale struttura per errori HTTP 404, come mostrato nell'esempio seguente:

Per creare un filtro metrico utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Scegli `Actions`, Crea filtro parametri.
4. In Modello di filtro, digitare **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`**.
5. (Facoltativo) Per testare il modello di filtro, in Test Pattern (Modello di test), inserisci uno o più log eventi da utilizzare per testare il modello. Ogni log eventi deve essere all'interno di una riga, in quanto le interruzioni di riga vengono utilizzate per separare i log eventi nella casella Log event messages (Messaggi di registro eventi).
6. Scegliere Avanti, quindi per Nome filtro, digitare HTTP404Errors.
7. In Dettagli parametro, per Spazio dei nomi del parametro, immettere **`MyNameSpace`**.
8. Per Nome parametro, immettere **`ApacheNotFoundErrorCount`**.
9. Verificare che il Valore parametro sia 1. Questo specifica che il conteggio viene incrementato di 1 per ogni evento di errore 404.
10. In Valore predefinito immettere 0, quindi scegliere Avanti.
11. Scegli Crea filtro parametri.

Per creare un filtro metrico utilizzando AWS CLI

Al prompt dei comandi, esegui il comando seguente:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP404Errors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=404, size]' \
  --metric-transformations \
      metricName=ApacheNotFoundErrorCount,metricNamespace=MyNamespace,metricValue=1
```

In questo esempio, sono stati utilizzati caratteri letterali, ad esempio le parentesi quadre sinistra e destra, le virgolette e la stringa di caratteri 404. Il modello deve corrispondere con l'intero messaggio del log eventi, in modo da essere considerato per il monitoraggio.

Puoi verificare la creazione del filtro parametri utilizzando il comando describe-metric-filters. L'output visualizzato dovrebbe essere di questo tipo:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log

{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

Ora puoi pubblicare alcuni eventi manualmente:

```
aws logs put-log-events \
--log-group-name MyApp/access.log --log-stream-name hostname \
--log-events \
timestamp=1394793518000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 404 2326" \
timestamp=1394793528000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb2.gif HTTP/1.0\" 200 2326"
```

Subito dopo aver inserito questi eventi di registro di esempio, puoi recuperare la metrica denominata nella CloudWatch console come. ApacheNotFoundErrorCount

Esempio: conteggio di codici HTTP 4xx

Come nell'esempio precedente, potresti voler monitorare i tuoi log di accesso al servizio Web e monitorare i livelli del codice di risposta HTTP. Ad esempio, potresti voler monitorare tutti gli errori di livello HTTP 400. Tuttavia, è possibile tu non voglia specificare un nuovo filtro parametri per ogni codice restituito.

L'esempio seguente spiega come creare un parametro che includa tutte le risposte del codice di livello HTTP 400 da un log di accesso utilizzando il formato di log di accesso Apache dall'esempio Esempio: conteggio di codici HTTP 404.

Per creare un filtro metrico utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Scegli il nome del gruppo di log per il server Apache.
4. Scegli `Actions`, Crea filtro parametri.
5. In Filter pattern (Modello di filtro), inserisci **`[ip, id, user, timestamp, request, status_code=4*, size]`**.
6. (Facoltativo) Per testare il modello di filtro, in Modello di test, inserisci uno o più log eventi da utilizzare per testare il modello. Ogni log eventi deve essere all'interno di una riga, in quanto le interruzioni di riga vengono utilizzate per separare i log eventi nella casella Log event messages (Messaggi di registro eventi).
7. Scegli Next (Avanti), quindi per Filter name (Nome filtro), digita **`HTTP4xxErrors`**.
8. In Metric details (Dettagli parametro), per Metric namespace (Spazio dei nomi del parametro), inserisci **`MyNameSpace`**.
9. Per Metric name (Nome parametro), inserisci HTTP4xxErrors.
10. In Metric value (Valore parametro), inserisci 1. Questo specifica che il conteggio viene incrementato di 1 per ogni log contente un errore 4xx.
11. In Default value (Valore predefinito) inserisci 0, quindi scegli Next (Avanti).
12. Scegli Crea filtro parametri.

Per creare un filtro metrico utilizzando AWS CLI

Al prompt dei comandi, esegui il comando seguente:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP4xxErrors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=4*, size]' \
  --metric-transformations \
  metricName=HTTP4xxErrors,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Puoi utilizzare i seguenti dati in chiamate put-event per verificare questa regola. Se non rimuovi la regola di monitoraggio nell'esempio precedente, genererai due parametri differenti.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Esempio: estrazione di campi da un log di Apache e assegnare dimensioni

A volte, invece di contare, è utile utilizzare i valori all'interno dei singoli eventi di log per i valori di parametro. Questo esempio illustra il modo in cui puoi creare una regola di estrazione per creare un parametro che misura i byte trasferiti da un server Web Apache.

In questo esempio viene inoltre illustrato come assegnare le dimensioni al parametro che si sta creando.

Per creare un filtro metrico utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Scegli il nome del gruppo di log per il server Apache.
4. Scegli `Actions`, Crea filtro parametri.
5. In Filter pattern (Modello di filtro), inserisci **`[ip, id, user, timestamp, request, status_code, size]`**.
6. (Facoltativo) Per testare il modello di filtro, in Modello di test, inserisci uno o più log eventi da utilizzare per testare il modello. Ogni log eventi deve essere all'interno di una riga, in quanto le interruzioni di riga vengono utilizzate per separare i log eventi nella casella Log event messages (Messaggi di registro eventi).
7. Scegli Next (Avanti), quindi per Filter name (Nome filtro), digita **`size`**.
8. In Metric details (Dettagli parametro), per Metric namespace (Spazio dei nomi del parametro), inserisci **`MyNameSpace`**. Poiché si tratta di un nuovo spazio dei nomi, assicurarsi che sia selezionato Create new (Creazione di nuovi).
9. Per Metric name (Nome parametro), inserisci **`BytesTransferred`**
10. In Metric value (Valore parametro), inserisci **`$size`**.

11. Per Unit (Unità) seleziona Bytes (Byte).
12. In Dimension Name (Nome dimensione) digita **IP**.
13. Per Dimension Value (Valore dimensione) digita **$ip**, quindi scegli Next (Avanti).
14. Scegli Crea filtro parametri.

Per creare questo filtro metrico utilizzando AWS CLI

Al prompt dei comandi, esegui il comando seguente.

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
 --filter-name BytesTransferred \
 --filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
 --metric-transformations  \
 metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size'
```

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
--filter-name BytesTransferred \
--filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
--metric-transformations  \
metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size',unit=Bytes,dimensio
$ip}}'
```

> **Note**
>
> In questo comando, utilizzare questo formato per specificare più dimensioni.
>
> ```
> aws logs put-metric-filter \
> --log-group-name my-log-group-name \
> --filter-name my-filter-name \
> --filter-pattern 'my-filter-pattern' \
> --metric-transformations  \
> metricName=my-metric-name,metricNamespace=my-metric-namespace,metricValue=my-
> token,unit=unit,dimensions='{dimension1=$dim,dimension2=$dim2,dim3=$dim3}'
> ```

Puoi utilizzare i seguenti dati in chiamate put-log-event per verificare questa regola. Questo genererà due parametri differenti, se non rimuovi la regola di monitoraggio nell'esempio precedente.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Elencazione di filtri di parametri

Puoi creare un elenco di tutti i filtri di parametri in un gruppo di log.

Per elencare i filtri metrici utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Nel riquadro contenuti, nell'elenco dei gruppi di log, nella colonna Filtri di parametri, seleziona il numero di filtri.

   Nella schermata Gruppi di log > Filtri per sono elencati tutti i filtri di parametri associati al gruppo di log.

Per elencare i filtri metrici utilizzando il AWS CLI

Al prompt dei comandi, esegui il comando seguente:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

Di seguito è riportato un output di esempio:

```
{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
```

```
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

Eliminazione di un filtro di parametri

Una policy è identificata dal relativo nome e dal gruppo di log a cui appartiene.

Per eliminare un filtro metrico utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Nel riquadro contenuti, nella colonna Metric Filters (Filtro di parametri), seleziona il numero di filtri di parametri per il gruppo di log.
4. Nella schermata Metric Filters (Filtri di parametri) seleziona la casella corrispondente al nome del filtro che si desidera eliminare. Scegli Elimina.
5. Quando viene richiesta la conferma, seleziona Elimina.

Per eliminare un filtro metrico utilizzando il AWS CLI

Al prompt dei comandi, esegui il comando seguente:

```
aws logs delete-metric-filter --log-group-name MyApp/access.log \
 --filter-name MyFilterName
```

Real-time trattamento dei dati di registro con abbonamenti

Puoi utilizzare gli abbonamenti per accedere a un feed in tempo reale di eventi di log di CloudWatch Logs e farlo distribuire ad altri servizi come uno stream Amazon Kinesis, uno stream Amazon Data Firehose o AWS Lambda per l'elaborazione, l'analisi o il caricamento personalizzati su altri sistemi. Quando i log eventi vengono inviati al servizio ricevente, vengono codificati in formato base64 e compressi nel formato gzip.

Puoi anche utilizzare la centralizzazione CloudWatch dei log per replicare i dati di log da più account e regioni in una posizione centrale. Per ulteriori informazioni, consulta Cross-account Centralizzazione dei log interregionali.

Per iniziare a sottoscrivere gli eventi di registro, crea la risorsa ricevente, ad esempio uno stream di Amazon Kinesis Data Streams, in cui verranno recapitati gli eventi. Un filtro di abbonamento definisce il modello di filtro da utilizzare per filtrare gli eventi di registro che vengono inviati alla AWS risorsa, nonché le informazioni su dove inviare gli eventi di registro corrispondenti. Gli eventi di registro vengono inviati alla risorsa ricevente subito dopo essere stati inseriti, in genere meno di tre minuti.

> **Note**
>
> Se un gruppo di log con un abbonamento utilizza la trasformazione dei log, il pattern di filtro viene confrontato con le versioni trasformate degli eventi di log. Per ulteriori informazioni, consulta Trasforma i log durante l'ingestione.

È possibile creare sottoscrizioni a livello di account e a livello di gruppo di log. Ogni account può avere un filtro di abbonamento a livello di account per regione. A ogni gruppo di log possono essere associati fino a cinque filtri di abbonamento.

> **Note**
>
> Se il servizio di destinazione restituisce un errore ripetibile come un'eccezione di limitazione o un'eccezione del servizio riattivabile (ad esempio HTTP 5xx), CloudWatch Logs continua a ritentare la consegna per un massimo di 24 ore. CloudWatch Logs non tenta di effettuare nuovamente la consegna se l'errore è un errore non ripetibile, ad esempio o. AccessDeniedException ResourceNotFoundException In questi casi, il filtro delle

sottoscrizioni è disabilitato per un massimo di 10 minuti, quindi CloudWatch Logs riprova a inviare i log alla destinazione. Durante questo periodo di disattivazione, i log vengono saltati.

CloudWatch Logs produce anche CloudWatch metriche sull'inoltro degli eventi di registro alle sottoscrizioni. Per ulteriori informazioni, consulta Monitoraggio con CloudWatch metriche.

Puoi anche utilizzare un abbonamento CloudWatch Logs per trasmettere i dati di log quasi in tempo reale a un cluster Amazon Service. OpenSearch Per ulteriori informazioni, consulta Streaming CloudWatch Logs data to Amazon OpenSearch Service.

Le sottoscrizioni sono supportate solo per i gruppi di log nella classe di log Standard. Per ulteriori informazioni sulle classi di log, vedereClassi di registro.

> **Note**
>
> I filtri di abbonamento possono registrare in batch gli eventi per ottimizzare la trasmissione e ridurre la quantità di chiamate effettuate verso la destinazione. Il batching non è garantito ma viene utilizzato quando possibile.

Per l'elaborazione in batch e l'analisi dei dati di registro in base a una pianificazione, prendi in considerazione l'utilizzoAutomatizzazione dell'analisi dei log con interrogazioni pianificate. Le query pianificate eseguono automaticamente le query di CloudWatch Logs Insights e forniscono risultati a destinazioni come bucket Amazon S3 o Amazon Event Bus. EventBridge

> **Note**
>
> I filtri di abbonamento assicurano la consegna degli eventi almeno una volta, mentre occasionalmente possono verificarsi eventi duplicati.

Indice

- Concetti
- Filtri di sottoscrizione a livello di gruppo di log
- Account-level filtri di abbonamento
- Cross-account Abbonamenti interregionali
- Prevenzione del "confused deputy"

- Prevenzione della ricorsione dei log

Concetti

Ogni filtro sottoscrizione è formato dai seguenti elementi chiave.

Modello di filtro

Una descrizione simbolica di come CloudWatch i log devono interpretare i dati in ogni evento di registro, insieme a espressioni di filtro che limitano ciò che viene recapitato alla risorsa di destinazione. AWS Per ulteriori informazioni sulla sintassi del modello di filtro, consulta Sintassi del modello di filtro per filtri di parametri, filtri di sottoscrizione, filtri di log eventi e Live Tail.

Arn di destinazione

L'Amazon Resource Name (ARN) dello stream di Amazon Kinesis Data Streams, del flusso Firehose o della funzione Lambda che desideri utilizzare come destinazione del feed di abbonamento.

Arn del ruolo

Un ruolo IAM che concede a CloudWatch Logs le autorizzazioni necessarie per inserire i dati nella destinazione scelta. Questo ruolo non è necessario per le destinazioni Lambda perché CloudWatch Logs può ottenere le autorizzazioni necessarie dalle impostazioni di controllo degli accessi sulla funzione Lambda stessa.

> **Note**
>
> Se il role manager è abilitato nel tuo account, CloudWatch assegna il ruolo per te e le opzioni di ruolo qui descritte (ad esempio il pulsante Crea nuovo ruolo) vengono sostituite da un'opzione Personalizza. Per utilizzare un ruolo diverso, scegli Personalizza. Per ulteriori informazioni sulla creazione del ruolo IAM, consulta la creazione del ruolo IAM nella IAM User Guide.

distribuzione

Il metodo utilizzato per distribuire i dati di log alla destinazione, quando la destinazione è un flusso di dati Amazon Kinesis. Per impostazione predefinita, i dati di log vengono raggruppati dal flusso di log. Per una distribuzione più uniforme, puoi raggruppare i dati di log casualmente.

Per le sottoscrizioni a livello di gruppo di log, è incluso anche il seguente elemento chiave:

nome gruppo di log

Il gruppo di log a cui associare il filtro sottoscrizione. Tutti i log eventi caricati a questo gruppo di log sono soggetti al filtro di sottoscrizione e quelli che corrispondono al filtro vengono consegnati al servizio di destinazione che riceve i log eventi corrispondenti.

Per gli abbonamenti a livello di account, è incluso anche il seguente elemento chiave:

criteri di selezione

I criteri utilizzati per selezionare a quali gruppi di log è applicato il filtro di sottoscrizione a livello di account. Se non lo specifichi, il filtro di sottoscrizione a livello di account viene applicato a tutti i gruppi di log dell'account. Questo campo viene utilizzato per evitare loop di log infiniti. Per ulteriori informazioni sul problema dei loop di log infiniti, vederePrevenzione della ricorsione dei log.

I criteri di selezione hanno un limite di dimensione di 25 KB.

Per i gruppi di log centralizzati, sono inclusi anche i seguenti elementi chiave. Questi elementi possono essere utilizzati come criteri di selezione dei campi per aiutare a identificare l'origine dei dati di registro, consentendo un filtraggio e un'analisi più granulari delle metriche derivate dai log centralizzati.

@aws .account

Questo campo identifica l'ID dell' AWS account da cui ha avuto origine l'evento di registro.

@aws .region

Questo campo identifica la AWS regione in cui è stato generato l'evento di registro.

@source .log

Questo campo identifica il gruppo di log di origine da cui ha avuto origine l'evento di log.

Filtri di sottoscrizione a livello di gruppo di log

Puoi utilizzare un filtro di abbonamento con Amazon Kinesis Data Streams AWS Lambda, Amazon Data Firehose o Amazon Service. OpenSearch I log inviati a un servizio tramite un filtro di abbonamento sono codificati in base64 e compressi con il formato gzip. Se utilizzi registri centralizzati

con il tuo AWS Organizations, puoi scegliere di emettere i campi `@aws.account@aws.region`, e di `@source.log` sistema per identificare quali dati provengono da quali account, regioni e gruppi di log di origine della tua organizzazione. Questa sezione fornisce esempi da seguire per creare un filtro di sottoscrizione CloudWatch Logs che invii i dati di log a Firehose, Lambda, Amazon Kinesis Data Streams e Service. OpenSearch

> **Note**
>
> Se desideri effettuare una ricerca nei dati di log, consulta Filter and pattern syntax.

Esempi

- Esempio 1: filtri di abbonamento con Amazon Kinesis Data Streams
- Esempio 2: filtri di abbonamento con AWS Lambda
- Esempio 3: filtri di abbonamento con Amazon Data Firehose
- Esempio 4: filtri di abbonamento con Amazon Service OpenSearch

Esempio 1: filtri di abbonamento con Amazon Kinesis Data Streams

L'esempio seguente associa un filtro di sottoscrizione a un gruppo di log contenente eventi AWS CloudTrail . Il filtro di abbonamento invia tutte le attività registrate effettuate dalle AWS credenziali «Root» a uno stream in Amazon Kinesis Data Streams chiamato "». RootAccess Per ulteriori informazioni su come inviare AWS CloudTrail eventi ai CloudWatch log, consulta Sending Events to Logs nella CloudTrail Guida per CloudWatch l'utente. AWS CloudTrail

> **Note**
>
> Prima di creare il flusso , calcola il volume dei dati di log che verrà generato. Assicurati di creare un flusso che includa partizioni sufficienti per gestire questo volume. Se il flusso non dispone di shard sufficienti, il flusso di log verrà limitato. Per ulteriori informazioni sui limiti dei volumi di flussi, consulta Quote e limiti.
> Vengono fatti nuovi tentativi per i risultati limitati per un massimo di 24 ore. Dopo 24 ore, i risultati non riusciti saranno eliminati.
> Per ridurre il rischio di limitazione, procedi nel seguente modo:
>
> - Specifica `distribution` quando `random` crei il filtro di abbonamento con PutSubscriptionFilter o https://awscli.amazonaws.com/v2/documentation/api/2.4.18/

reference/logs/put-subscription-filter.html put-subscription-filter. Per impostazione predefinita, la distribuzione del filtro di flusso è basata su log stream e ciò può causare limitazioni.

- Monitora il tuo stream utilizzando CloudWatch le metriche. Questo ti aiuta a identificare eventuali limitazioni e a regolare la configurazione di conseguenza. Ad esempio, la `DeliveryThrottling` metrica può essere utilizzata per tenere traccia del numero di eventi di registro per CloudWatch i quali Logs è stato limitato durante l'inoltro dei dati alla destinazione dell'abbonamento. Per ulteriori informazioni sul monitoraggio, consulta Monitoraggio con CloudWatch metriche.
- Usa la modalità di capacità su richiesta per il tuo streaming in Amazon Kinesis Data Streams. On-demandla modalità adatta istantaneamente i carichi di lavoro man mano che aumentano o diminuiscono. Per ulteriori informazioni sulla modalità di capacità su richiesta, vedere modalità. On-demand
- Limita il modello di filtro CloudWatch dell'abbonamento in modo che corrisponda alla capacità del tuo stream in Amazon Kinesis Data Streams. Se invii una quantità eccessiva di dati al flusso, potrebbe essere necessario ridurre le dimensioni del filtro o modificarne i criteri.

Per creare un filtro di abbonamento per Amazon Kinesis Data Streams

1. Crea un flusso di destinazione utilizzando il comando seguente:

```
$ C:\>  aws kinesis create-stream --stream-name "RootAccess" --shard-count 1
```

2. Attendi finché il flusso non diventa attivo. Questa operazione può richiedere uno o due minuti. Puoi utilizzare il seguente comando https://docs.aws.amazon.com/cli/latest/reference/kinesis/describe-stream.html describe-stream di Amazon Kinesis Data Streams per verificare la proprietà. StreamDescription.StreamStatus Inoltre, annota il StreamDescription.StreamARN valore, poiché ti servirà in una fase successiva:

```
aws kinesis describe-stream --stream-name "RootAccess"
```

   Di seguito è riportato un output di esempio:

```
{
    "StreamDescription": {
```

```
        "StreamStatus": "ACTIVE",
        "StreamName": "RootAccess",
        "StreamARN": "arn:aws:kinesis:us-east-1:123456789012:stream/RootAccess",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "340282366920938463463374607431768211455",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "49551135218688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Crea il ruolo IAM che concederà a CloudWatch Logs il permesso di inserire dati nel tuo stream. In primo luogo, sarà necessario creare una policy di attendibilità in un file (ad esempio, `~/TrustPolicyForCWL-Kinesis.json`). Utilizza un editor di testo per creare questa policy. Non utilizzare la console IAM per crearla.

   Questa policy include una chiave di contesto della condizione globale `aws:SourceArn` per prevenire il problema di sicurezza noto come "confused deputy". Per ulteriori informazioni, consulta Prevenzione del "confused deputy".

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "ArnLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. Utilizza il comando create-role per creare un ruolo IAM, specificando il file della policy di attendibilità. Annota il Role.Arn valore restituito, poiché ti servirà anche per un passaggio successivo:

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

Di seguito è riportato un esempio di output.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account. In primo luogo, creerai una policy delle autorizzazioni in un file (ad esempio, `~/PermissionsForCWL-Kinesis.json`). Utilizza un editor di testo per creare questa policy. Non utilizzare la console IAM per crearla.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/RootAccess"
    }
```

```
  ]
}
```

6. Associa la policy delle autorizzazioni al ruolo utilizzando il seguente comando put-role-policy:

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. Dopo che lo stream è in stato attivo e hai creato il ruolo IAM, puoi creare il filtro di sottoscrizione CloudWatch Logs. Il filtro sottoscrizioni avvia immediatamente il flusso di dati di log in tempo reale dal gruppo di log selezionato nel flusso :

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RootAccess" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:kinesis:region:123456789012:stream/RootAccess" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

8. Dopo aver impostato il filtro delle sottoscrizioni, CloudWatch Logs inoltra al tuo stream tutti gli eventi di log in entrata che corrispondono al pattern di filtro. Puoi verificare che ciò avvenga utilizzando un iteratore shard di Amazon Kinesis Data Streams e utilizzando il comando Amazon Kinesis Data Streams get-records per recuperare alcuni record di Amazon Kinesis Data Streams:

```
aws kinesis get-shard-iterator --stream-name RootAccess --shard-id
 shardId-000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
```

```
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Tieni presente che potresti dover effettuare questa chiamata alcune volte prima che Amazon Kinesis Data Streams inizi a restituire i dati.

In genere si visualizzerà una risposta con una matrice di record. L'attributo Data in un record Amazon Kinesis Data Streams è codificato in base64 e compresso con il formato gzip. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando i seguenti comandi Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

I dati con codifica base64 e decompressi sono in formato JSON con la seguente struttura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
```

```
}
```

Gli elementi chiave nella struttura di dati precedente sono i seguenti:

owner

L'ID dell' AWS account dei dati di registro di origine.

logGroup

Nome del gruppo di log dei dati di log originari.

logStream

Nome del flusso di log dei dati di log originari.

subscriptionFilters

Elenco dei nomi di filtro sottoscrizione che corrispondono con i dati di log originari.

messageType

I messaggi di dati utilizzeranno il tipo "DATA_MESSAGE". A volte CloudWatch i log possono emettere record di Amazon Kinesis Data Streams con un tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

logEvents

I dati di log effettivi, rappresentati come una varietà di record di eventi di log. La proprietà "id" è un identificatore univoco per ogni log eventi.

Esempio 2: filtri di abbonamento con AWS Lambda

In questo esempio, creerai un filtro di abbonamento CloudWatch Logs che invia i dati di registro alla tua AWS Lambda funzione.

> **Note**
>
> Prima di creare la funzione Lambda, calcola il volume dei dati di log che verrà generato. Assicurati di creare una funzione che possa gestire questo volume. Se la funzione non dispone di volume sufficiente, il flusso di log verrà limitato. Per ulteriori informazioni sui limiti di Lambda, consulta Limiti di AWS Lambda.

Creazione di un filtro di sottoscrizione per Lambda

1. Crea la AWS Lambda funzione.

   Assicurati di aver configurato il ruolo di esecuzione di Lambda. Per ulteriori informazioni, consulta Fase 2.2: creazione di un ruolo IAM (ruolo di esecuzione) nella Guida per gli sviluppatori AWS Lambda .

2. Apri un editor di testo e crea un file denominato `helloWorld.js` con il seguente contenuto:

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Comprimi il file hello World.js e salvalo con il nome`helloWorld.zip`.

4. Utilizza il comando seguente, in cui il ruolo è il ruolo di esecuzione Lambda configurato nella prima fase:

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs12.x
```

5. Grant CloudWatch Logs consente di eseguire la tua funzione. Utilizza il comando seguente, sostituendo l'account di segnaposto con il tuo account e il gruppo di log di segnaposto con il gruppo di log da elaborare:

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
```

```
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789123:log-group:TestLambda:*" \
    --source-account "123456789012"
```

6. Crea un filtro di sottoscrizione utilizzando il comando seguente, sostituendo l'account di segnaposto con il tuo account e il gruppo di log di segnaposto con il gruppo di log da elaborare:

```
aws logs put-subscription-filter \
    --log-group-name myLogGroup \
    --filter-name demo \
    --filter-pattern "" \
    --destination-arn arn:aws:lambda:region:123456789123:function:helloworld
```

7. Verifica tramite un log eventi di esempio (facoltativo). Al prompt dei comandi, esegui il comando seguente, il quale inserirà un semplice messaggio di log nel flusso sottoscritto.

   Per vedere l'output della tua funzione Lambda, vai alla funzione Lambda dove vedrai l'output in/aws/lambda/helloworld:

```
aws logs put-log-events --log-group-name myLogGroup --log-stream-name stream1 --
log-events "[{\"timestamp\":<CURRENT TIMESTAMP MILLIS> , \"message\": \"Simple
 Lambda Test\"}]"
```

   Dovresti visualizzare una risposta con una matrice di Lambda. L'attributo Data (Dati) nel log Lambda usa la codifica base64 e la compressione nel formato gzip. Il payload effettivo che Lambda riceve è nel formato seguente, `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando i seguenti comandi Unix:

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

   I dati con codifica base64 e decompressi sono in formato JSON con la seguente struttura:

```
{
    "owner": "123456789012",
    "logGroup": "CloudTrail",
    "logStream": "123456789012_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
```

```
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
}
```

Gli elementi chiave nella struttura di dati precedente sono i seguenti:

owner

L'ID dell' AWS account dei dati di registro di origine.

logGroup

Nome del gruppo di log dei dati di log originari.

logStream

Nome del flusso di log dei dati di log originari.

subscriptionFilters

Elenco dei nomi di filtro sottoscrizione che corrispondono con i dati di log originari.

messageType

I messaggi di dati utilizzeranno il tipo "DATA_MESSAGE". A volte CloudWatch i log possono emettere record Lambda di tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

logEvents

I dati di log effettivi, rappresentati come una varietà di record di eventi di log. La proprietà "id" è un identificatore univoco per ogni log eventi.

Esempio 3: filtri di abbonamento con Amazon Data Firehose

In questo esempio, creerai un abbonamento CloudWatch Logs che invia tutti gli eventi di log in entrata che corrispondono ai filtri definiti al tuo flusso di consegna Amazon Data Firehose. I dati inviati da CloudWatch Logs ad Amazon Data Firehose sono già compressi con gzip di livello 6, quindi non è necessario utilizzare la compressione nel flusso di distribuzione Firehose. Puoi quindi utilizzare la funzione di decompressione di Firehose per decomprimere automaticamente i log. Per ulteriori informazioni, consulta CloudWatch Inviare i log a Firehose.

> **Note**
>
> Prima di creare lo stream Firehose, calcola il volume di dati di registro che verranno generati. Assicuratevi di creare uno stream Firehose in grado di gestire questo volume. Se il flusso non è in grado di gestire il volume, il flusso di log verrà limitato. Per ulteriori informazioni sui limiti di volume dello stream Firehose, consulta Amazon Data Firehose Data Limits.

Per creare un filtro di abbonamento per Firehose

1. Crea un bucket Amazon Simple Storage Service (Amazon S3). Ti consigliamo di utilizzare un bucket creato appositamente per CloudWatch Logs. Tuttavia, se intendi utilizzare un bucket esistente, puoi passare alla fase 2.

   Esegui il comando seguente, sostituendo il segnaposto Regione con la Regione che desideri utilizzare:

   ```
   aws s3api create-bucket --bucket amzn-s3-demo-bucket2 --create-bucket-configuration
    LocationConstraint=region
   ```

Di seguito è riportato un output di esempio:

```
{
    "Location": "/amzn-s3-demo-bucket2"
}
```

2. Crea il ruolo IAM che concede ad Amazon Data Firehose l'autorizzazione a inserire dati nel tuo bucket Amazon S3.

   Per ulteriori informazioni, consulta Controlling Access with Amazon Data Firehose nella Amazon Data Firehose Developer Guide.

   In primo luogo, utilizza un editor di testo per creare una policy di attendibilità in un file `~/TrustPolicyForFirehose.json`, come segue:

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "firehose.amazonaws.com" },
    "Action": "sts:AssumeRole"
    }
}
```

3. Utilizza il comando create-role per creare un ruolo IAM, specificando il file della policy di attendibilità. Annota il Role.Arn valore restituito, poiché ti servirà in una fase successiva:

```
aws iam create-role \
 --role-name FirehosetoS3Role \
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
```

```
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Crea una politica di autorizzazioni per definire quali azioni può eseguire Firehose sul tuo account. In primo luogo, utilizza un editor di testo per creare una policy di autorizzazione in un file *~/*`PermissionsForFirehose.json`:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::amzn-s3-demo-bucket2",
          "arn:aws:s3:::amzn-s3-demo-bucket2/*" ]
    }
  ]
}
```

5. Associa la policy di autorizzazioni al ruolo utilizzando il comando put-role-policy seguente:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Crea un flusso di consegna Firehose di destinazione come segue, sostituendo i valori segnaposto per ROLearn e BucketARN con il ruolo e il bucket ARN che hai creato:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::amzn-s3-demo-bucket2"}'
```

Tieni presente che Firehose utilizza automaticamente un prefisso in formato orario UTC per gli oggetti Amazon S3 consegnati. YYYY/MM/DD/HH Puoi specificare un ulteriore prefisso da aggiungere davanti al prefisso del formato data e ora. Se il prefisso termina con una barra (/), viene visualizzato come cartella nel bucket Amazon S3.

7. Attendi finché il flusso non si attiva (questa operazione potrebbe richiedere alcuni minuti). Puoi usare il comando Firehose describe-delivery-stream per controllare la proprietà. DeliveryStreamDescription.DeliveryStreamStatus Inoltre, annota il DeliveryStreamDescription.DeliveryStreamARN valore, poiché ti servirà in un passaggio successivo:

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket2",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. Crea il ruolo IAM che CloudWatch concede a Logs il permesso di inserire dati nel tuo flusso di distribuzione Firehose. In primo luogo, utilizza un editor di testo per creare una policy di attendibilità in un file `~/TrustPolicyForCWL.json`:

   Questa policy include una chiave di contesto della condizione globale `aws:SourceArn` per prevenire il problema di sicurezza noto come "confused deputy". Per ulteriori informazioni, consulta Prevenzione del "confused deputy".

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole",
       "Condition": {
            "ArnLike": {
                "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
            }
        }
     }
   }
   ```

9. Utilizza il comando create-role per creare un ruolo IAM, specificando il file della policy di attendibilità. Nota il Role.Arn valore restituito, poiché ti servirà in un passaggio successivo:

   ```
   aws iam create-role \
   --role-name CWLtoKinesisFirehoseRole \
   --assume-role-policy-document file://~/TrustPolicyForCWL.json

   {
       "Role": {
           "AssumeRolePolicyDocument": {
               "Statement": {
                   "Action": "sts:AssumeRole",
                   "Effect": "Allow",
                   "Principal": {
                       "Service": "logs.amazonaws.com"
                   },
                   "Condition": {
                        "StringLike": {
                            "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                        }
                    }
   ```

```
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account. In primo luogo, utilizza un editor di testo per creare un file di policy di autorizzazioni (ad esempio, `~/PermissionsForCWL.json`):

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. Associa la policy di autorizzazioni al ruolo utilizzando il seguente comando put-role-policy:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Dopo che il flusso di distribuzione di Amazon Data Firehose è attivo e hai creato il ruolo IAM, puoi creare il filtro di sottoscrizione CloudWatch Logs. Il filtro di abbonamento avvia immediatamente il flusso di dati di log in tempo reale dal gruppo di log scelto al flusso di distribuzione di Amazon Data Firehose:

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "Destination" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/my-
delivery-stream" \
```

```
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
```

13. Dopo aver impostato il filtro di abbonamento, CloudWatch Logs inoltrerà tutti gli eventi di log in entrata che corrispondono al modello di filtro al flusso di consegna di Amazon Data Firehose. I tuoi dati inizieranno a comparire in Amazon S3 in base all'intervallo di tempo del buffer impostato nel flusso di consegna di Amazon Data Firehose. Quando è trascorso tempo sufficiente, puoi verificare i dati controllando il Bucket Amazon S3.

```
aws s3api list-objects --bucket 'amzn-s3-demo-bucket2' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2015-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-35-40-7cc92023-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'amzn-s3-demo-bucket2' --key 'firehose/2015/10/29/00/
my-delivery-stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz
```

```
{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2015 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

I dati nell'oggetto di Amazon S3 vengono compressi nel formato gzip. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando i seguenti comandi Unix:

```
zcat testfile.gz
```

Esempio 4: filtri di abbonamento con Amazon Service OpenSearch

In questo esempio, creerai un abbonamento CloudWatch Logs che invia gli eventi di log in entrata che corrispondono ai filtri definiti al tuo dominio OpenSearch Service.

Per creare un filtro di abbonamento per Service OpenSearch

1. Crea un dominio OpenSearch di servizio. Per ulteriori informazioni, vedere Creazione di domini OpenSearch di servizio
2. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
3. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
4. Selezionare il nome del gruppo di registro.
5. Scegli Azioni, Filtri di abbonamento, Crea filtro OpenSearch di abbonamento Amazon Service.
6. Scegli se eseguire lo streaming in un cluster di questo account o di un altro account.
   - Se hai scelto Questo account, seleziona il dominio che hai creato nel passaggio 1.
   - Se hai scelto Un altro account, inserisci l'ARN e l'endpoint di quel dominio.
7. Se hai scelto un altro account, fornire il dominio ARN e l'endpoint.
8. Per il cluster Amazon OpenSearch Service, scegli il nome del cluster in cui verranno consegnati i dati del gruppo di log
9. Scegli un formato di registro.

10. Per lo schema di filtro degli abbonamenti, inserisci i termini o lo schema da trovare nel registro degli eventi. Questo ti assicura di inviare solo i dati che ti interessano al tuo cluster OpenSearch di servizi. Per ulteriori informazioni, consulta Sintassi del modello di filtro per i filtri di parametri.
11. (Opzionale) In Select log data to test (Seleziona i dati di registro per il test), seleziona un flusso di registro, quindi scegli Test pattern (Modello di test) per verificare che il filtro di ricerca restituisca i risultati previsti.
12. Quindi scegli Start streaming (Avvia streaming).

Account-level filtri di abbonamento

Important

Esiste il rischio di provocare un ciclo ricorsivo infinito con i filtri di abbonamento che, se non risolti, può comportare un notevole aumento della fatturazione di ingresso. Per mitigare questo rischio, ti consigliamo di utilizzare i criteri di selezione nei filtri di abbonamento a livello di account per escludere i gruppi di log che inseriscono i dati di registro dalle risorse che fanno parte del flusso di lavoro di distribuzione degli abbonamenti. Per ulteriori informazioni su questo problema e per determinare quali gruppi di log escludere, consulta. Prevenzione della ricorsione dei log

È possibile impostare una politica di sottoscrizione a livello di account che includa un sottoinsieme di gruppi di log nell'account. La politica di sottoscrizione dell'account può funzionare con Amazon Kinesis Data Streams o Amazon Data AWS Lambda Firehose. I log inviati a un servizio tramite una policy di abbonamento a livello di account sono codificati in base64 e compressi con il formato gzip. Questa sezione fornisce esempi da seguire per creare un abbonamento a livello di account per Amazon Kinesis Data Streams, Lambda e Firehose.

Note

Per visualizzare un elenco di tutte le politiche di filtro delle sottoscrizioni nel tuo account, utilizza il `describe-account-policies` comando con il valore di for. `SUBSCRIPTION_FILTER_POLICY` `--policy-type` Per ulteriori informazioni, consulta https://docs.aws.amazon.com/cli/latest/reference/logs/describe-account-policies.html describe-account-policies¶.

Esempi

- Esempio 1: filtri di abbonamento con Amazon Kinesis Data Streams
- Esempio 2: filtri di abbonamento con AWS Lambda
- Esempio 3: filtri di abbonamento con Amazon Data Firehose

Esempio 1: filtri di abbonamento con Amazon Kinesis Data Streams

Prima di creare un flusso di dati di Amazon Kinesis Data Streams da utilizzare con una policy di abbonamento a livello di account, calcola il volume di dati di registro che verranno generati. Assicurati di creare un flusso che includa partizioni sufficienti per gestire questo volume. Se uno stream non contiene abbastanza frammenti, viene limitato. Per ulteriori informazioni sui limiti del volume dello stream, consulta Quote e limiti nella documentazione di Amazon Kinesis Data Streams.

Warning

Poiché gli eventi di log di più gruppi di log vengono inoltrati alla destinazione, esiste il rischio di limitazione. Vengono fatti nuovi tentativi per i risultati limitati per un massimo di 24 ore. Dopo 24 ore, i risultati non riusciti saranno eliminati.
Per ridurre il rischio di limitazione, procedi nel seguente modo:

- Monitora il tuo flusso di Amazon Kinesis Data Streams con delle metriche. CloudWatch Questo ti aiuta a identificare le limitazioni e a regolare la configurazione di conseguenza. Ad esempio, la `DeliveryThrottling` metrica tiene traccia del numero di eventi di registro per i quali CloudWatch Logs è stato limitato durante l'inoltro dei dati alla destinazione dell'abbonamento. Per ulteriori informazioni, consulta Monitoraggio con CloudWatch metriche.
- Usa la modalità di capacità su richiesta per il tuo streaming in Amazon Kinesis Data Streams. On-demandla modalità adatta istantaneamente i carichi di lavoro man mano che aumentano o diminuiscono. Per ulteriori informazioni, vedi modalità. On-demand
- Limita il modello di filtro dell'abbonamento CloudWatch Logs in modo che corrisponda alla capacità del tuo stream in Amazon Kinesis Data Streams. Se invii una quantità eccessiva di dati al flusso, potrebbe essere necessario ridurre le dimensioni del filtro o modificarne i criteri.

L'esempio seguente utilizza una policy di sottoscrizione a livello di account per inoltrare tutti gli eventi di log a uno stream in Amazon Kinesis Data Streams. Lo schema di filtro abbina qualsiasi evento di registro al testo `Test` e li inoltra allo stream in Amazon Kinesis Data Streams.

Per creare una policy di abbonamento a livello di account per Amazon Kinesis Data Streams

1. Crea un flusso di destinazione utilizzando il comando seguente:

```
$ C:\>  aws kinesis create-stream –stream-name "TestStream" –shard-count 1
```

2. Attendi qualche minuto affinché lo stream diventi attivo. Puoi verificare se lo stream è attivo utilizzando il comando describe-stream per controllare la proprietà. StreamDescription.StreamStatus

```
aws kinesis describe-stream --stream-name "TestStream"
```

Di seguito è riportato un output di esempio:

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "TestStream",
        "StreamARN": "arn:aws:kinesis:region:123456789012:stream/TestStream",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "EXAMPLE846346337460743176821145 5",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "EXAMPLE688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Crea il ruolo IAM che concederà a CloudWatch Logs il permesso di inserire dati nel tuo stream. In primo luogo, sarà necessario creare una policy di attendibilità in un file (ad esempio, `~/TrustPolicyForCWL-Kinesis.json`). Utilizza un editor di testo per creare questa policy.

   Questa policy include una chiave di contesto della condizione globale `aws:SourceArn` per prevenire il problema di sicurezza noto come "confused deputy". Per ulteriori informazioni, consulta Prevenzione del "confused deputy".

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole",
       "Condition": {
           "ArnLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
        }
      }
   }
   ```

4. Utilizza il comando create-role per creare un ruolo IAM, specificando il file della policy di attendibilità. Annota il Role.Arn valore restituito, poiché ti servirà anche per un passaggio successivo:

   ```
   aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
    file://~/TrustPolicyForCWL-Kinesis.json
   ```

   Di seguito è riportato un esempio di output.

   ```
   {
       "Role": {
           "AssumeRolePolicyDocument": {
               "Statement": {
                   "Action": "sts:AssumeRole",
                   "Effect": "Allow",
                   "Principal": {
                       "Service": "logs.amazonaws.com"
                   },
                   "Condition": {
                       "StringLike": {
                           "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                       }
   ```

```
                }
            }
        },
        "RoleId": "EXAMPLE450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account. In primo luogo, creerai una policy delle autorizzazioni in un file (ad esempio, `~/PermissionsForCWL-Kinesis.json`). Utilizza un editor di testo per creare questa policy. Non utilizzare la console IAM per crearlo.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/TestStream"
    }
  ]
}
```

6. Associa la policy delle autorizzazioni al ruolo utilizzando il seguente comando put-role-policy:

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. Dopo che lo stream è in stato Attivo e hai creato il ruolo IAM, puoi creare la politica di filtro CloudWatch delle sottoscrizioni Logs. La policy avvia immediatamente il flusso di dati di log in tempo reale verso il tuo stream. In questo esempio, tutti gli eventi di registro che contengono la stringa ERROR vengono trasmessi in streaming, ad eccezione di quelli nei gruppi di log denominati `LogGroupToExclude1` e`LogGroupToExclude2`.

```
aws logs put-account-policy \
    --policy-name "ExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
```

```
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisRole", "DestinationArn":"arn:aws:kinesis:region:123456789012:stream/
TestStream", "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

8. Dopo aver impostato il filtro delle sottoscrizioni, CloudWatch Logs inoltra allo stream tutti gli eventi di log in entrata che corrispondono al modello di filtro e ai criteri di selezione.

   Il `selection-criteria` campo è facoltativo, ma è importante per escludere i gruppi di log che possono causare una ricorsione infinita di log da un filtro di sottoscrizione. Per ulteriori informazioni su questo problema e sulla determinazione dei gruppi di log da escludere, vedere. Prevenzione della ricorsione dei log Attualmente, NOT IN è l'unico operatore supportato per`selection-criteria`.

   Puoi verificare il flusso degli eventi di log utilizzando un iteratore di shard di Amazon Kinesis Data Streams e utilizzando il `get-records` comando Amazon Kinesis Data Streams per recuperare alcuni record di Amazon Kinesis Data Streams:

   ```
   aws kinesis get-shard-iterator --stream-name TestStream --shard-id
    shardId-000000000000 --shard-iterator-type TRIM_HORIZON
   ```

   ```
   {
       "ShardIterator":
       "AAAAAAAAAAFGU/
   kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
   +wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
   +p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
   +g6rMo7UKWeI4+IWiK2OSh0uP"
   }
   ```

   ```
   aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
   kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
   +wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
   +p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
   +g6rMo7UKWeI4+IWiK2OSh0uP"
   ```

   Potrebbe essere necessario utilizzare questo comando alcune volte prima che Amazon Kinesis Data Streams inizi a restituire i dati.

In genere si visualizzerà una risposta con una matrice di record. L'attributo Data in un record Amazon Kinesis Data Streams è codificato in base64 e compresso con il formato gzip. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando i seguenti comandi Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

I dati con codifica base64 e decompressi sono in formato JSON con la seguente struttura:

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicy"
    ],
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

Gli elementi chiave della struttura dei dati sono i seguenti:

messageType

I messaggi di dati utilizzeranno il tipo "DATA_MESSAGE". A volte CloudWatch i log possono emettere record Amazon Kinesis Data Streams di tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

owner

L'ID dell' AWS account dei dati di registro di origine.

logGroup

Nome del gruppo di log dei dati di log originari.

logStream

Nome del flusso di log dei dati di log originari.

subscriptionFilters

Elenco dei nomi di filtro sottoscrizione che corrispondono con i dati di log originari.

logEvents

I dati di log effettivi, rappresentati come una varietà di record di eventi di log. La proprietà "id" è un identificatore univoco per ogni log eventi.

Livello di politica

Il livello al quale la politica è stata applicata. «ACCOUNT_LEVEL_POLICY» è la politica di filtro degli abbonamenti a livello `policyLevel` di account.

Esempio 2: filtri di abbonamento con AWS Lambda

In questo esempio, creerai una politica di filtro delle sottoscrizioni a livello di account CloudWatch Logs che invia i dati di registro alla tua AWS Lambda funzione.

> **Warning**
>
> Prima di creare la funzione Lambda, calcola il volume dei dati di log che verrà generato. Assicurati di creare una funzione che possa gestire questo volume. Se la funzione non è in grado di gestire il volume, il flusso di log verrà limitato. Poiché gli eventi di registro di tutti i gruppi di log o di un sottoinsieme dei gruppi di log dell'account vengono inoltrati alla

destinazione, esiste il rischio di limitazione. Per ulteriori informazioni sui limiti di Lambda, consulta Limiti di AWS Lambda.

Per creare una politica di filtro delle sottoscrizioni a livello di account per Lambda

1. Crea la funzione. AWS Lambda

   Assicurati di aver configurato il ruolo di esecuzione di Lambda. Per ulteriori informazioni, consulta Fase 2.2: creazione di un ruolo IAM (ruolo di esecuzione) nella Guida per gli sviluppatori AWS Lambda .

2. Apri un editor di testo e crea un file denominato `helloWorld.js` con il seguente contenuto:

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Comprimi il file hello World.js e salvalo con il nome`helloWorld.zip`.

4. Utilizza il comando seguente, in cui il ruolo è il ruolo di esecuzione Lambda configurato nella prima fase:

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs18.x
```

5. Grant CloudWatch Logs consente di eseguire la funzione. Usa il seguente comando, sostituendo l'account segnaposto con il tuo account.

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789012:log-group:*" \
    --source-account "123456789012"
```

6. Crea una politica di filtro dell'abbonamento a livello di account utilizzando il seguente comando, sostituendo l'account segnaposto con il tuo account. In questo esempio, tutti gli eventi di registro che contengono la stringa ERROR vengono trasmessi in streaming, ad eccezione di quelli nei gruppi di log denominati e. `LogGroupToExclude1` `LogGroupToExclude2`

   ```
   aws logs put-account-policy \
       --policy-name "ExamplePolicyLambda" \
       --policy-type "SUBSCRIPTION_FILTER_POLICY" \
       --policy-document
    '{"DestinationArn":"arn:aws:lambda:region:123456789012:function:helloWorld",
    "FilterPattern": "Test", "Distribution": "Random"}' \
       --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
    "LogGroupToExclude2"]' \
       --scope "ALL"
   ```

   Dopo aver impostato il filtro delle sottoscrizioni, CloudWatch Logs inoltra allo stream tutti gli eventi di log in entrata che corrispondono al modello di filtro e ai criteri di selezione.

   Il `selection-criteria` campo è facoltativo, ma è importante per escludere i gruppi di log che possono causare una ricorsione infinita di log da un filtro di sottoscrizione. Per ulteriori informazioni su questo problema e sulla determinazione dei gruppi di log da escludere, vedere. Prevenzione della ricorsione dei log Attualmente, NOT IN è l'unico operatore supportato per`selection-criteria`.

7. Verifica tramite un log eventi di esempio (facoltativo). Al prompt dei comandi, esegui il comando seguente, il quale inserirà un semplice messaggio di log nel flusso sottoscritto.

   Per vedere l'output della tua funzione Lambda, vai alla funzione Lambda dove vedrai l'output in/aws/lambda/helloworld:

```
aws logs put-log-events --log-group-name Example1 --log-stream-name logStream1 --
log-events "[{\"timestamp\":CURRENT TIMESTAMP MILLIS , \"message\": \"Simple Lambda
 Test\"}]"
```

Dovresti visualizzare una risposta con una matrice di Lambda. L'attributo Data (Dati) nel log Lambda usa la codifica base64 e la compressione nel formato gzip. Il payload effettivo che Lambda riceve è nel formato seguente, `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando i seguenti comandi Unix:

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

I dati con codifica base64 e decompressi sono in formato JSON con la seguente struttura:

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicyLambda"
    ],
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
```

```
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

> **Note**
>
> Il filtro di abbonamento a livello di account non verrà applicato al gruppo di log della funzione Lambda di destinazione. Questo serve a evitare una ricorsione infinita dei log che può portare a un aumento della fatturazione di ingresso. Per ulteriori informazioni su questo problema, vedere. Prevenzione della ricorsione dei log

Gli elementi chiave della struttura dei dati sono i seguenti:

messageType

I messaggi di dati utilizzeranno il tipo "DATA_MESSAGE". A volte CloudWatch i log possono emettere record Amazon Kinesis Data Streams di tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

owner

L'ID dell' AWS account dei dati di registro di origine.

logGroup

Nome del gruppo di log dei dati di log originari.

logStream

Nome del flusso di log dei dati di log originari.

subscriptionFilters

Elenco dei nomi di filtro sottoscrizione che corrispondono con i dati di log originari.

logEvents

I dati di log effettivi, rappresentati come una varietà di record di eventi di log. La proprietà "id" è un identificatore univoco per ogni log eventi.

Livello di politica

Il livello al quale la politica è stata applicata. «ACCOUNT_LEVEL_POLICY» è la politica di filtro degli abbonamenti a livello `policyLevel` di account.

Esempio 3: filtri di abbonamento con Amazon Data Firehose

In questo esempio, creerai una policy di filtro delle sottoscrizioni a livello di account CloudWatch Logs che invia gli eventi di log in entrata che corrispondono ai filtri definiti al tuo flusso di distribuzione di Amazon Data Firehose. I dati inviati da CloudWatch Logs ad Amazon Data Firehose sono già compressi con gzip di livello 6, quindi non è necessario utilizzare la compressione nel flusso di distribuzione Firehose. Puoi quindi utilizzare la funzione di decompressione di Firehose per decomprimere automaticamente i log. Per ulteriori informazioni, consulta Writing to Kinesis Data Firehose Using Logs. CloudWatch

> **Warning**
>
> Prima di creare lo stream Firehose, calcolate il volume di dati di registro che verranno generati. Assicuratevi di creare uno stream Firehose in grado di gestire questo volume. Se il flusso non è in grado di gestire il volume, il flusso di log verrà limitato. Per ulteriori informazioni sui limiti di volume dello stream Firehose, consulta Amazon Data Firehose Data Limits.

Per creare un filtro di abbonamento per Firehose

1. Crea un bucket Amazon Simple Storage Service (Amazon S3). Ti consigliamo di utilizzare un bucket creato appositamente per CloudWatch Logs. Tuttavia, se intendi utilizzare un bucket esistente, puoi passare alla fase 2.

   Esegui il comando seguente, sostituendo il segnaposto Regione con la Regione che desideri utilizzare:

   ```
   aws s3api create-bucket --bucket amzn-s3-demo-bucket2 --create-bucket-configuration
    LocationConstraint=region
   ```

   Di seguito è riportato un output di esempio:

```
{
    "Location": "/amzn-s3-demo-bucket2"
}
```

2. Crea il ruolo IAM che concede ad Amazon Data Firehose l'autorizzazione a inserire dati nel tuo bucket Amazon S3.

   Per ulteriori informazioni, consulta Controlling Access with Amazon Data Firehose nella Amazon Data Firehose Developer Guide.

   In primo luogo, utilizza un editor di testo per creare una policy di attendibilità in un file `~/TrustPolicyForFirehose.json`, come segue:

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "firehose.amazonaws.com" },
    "Action": "sts:AssumeRole"
    }
}
```

3. Utilizza il comando create-role per creare un ruolo IAM, specificando il file della policy di attendibilità. Prendi nota del Role.Arn valore restituito, poiché ti servirà in una fase successiva:

```
aws iam create-role \
 --role-name FirehosetoS3Role \
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "EXAMPLE50GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
```

```
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Crea una politica di autorizzazioni per definire quali azioni può eseguire Firehose sul tuo account. In primo luogo, utilizza un editor di testo per creare una policy di autorizzazione in un file ~/PermissionsForFirehose.json:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::amzn-s3-demo-bucket2",
          "arn:aws:s3:::amzn-s3-demo-bucket2/*" ]
    }
  ]
}
```

5. Associa la policy di autorizzazioni al ruolo utilizzando il comando put-role-policy seguente:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Crea un flusso di consegna Firehose di destinazione come segue, sostituendo i valori segnaposto per ROLearn e BucketARN con il ruolo e il bucket ARN che hai creato:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::amzn-s3-demo-bucket2"}'
```

NFirehose utilizza automaticamente un prefisso in formato orario UTC per gli oggetti Amazon S3 consegnati. YYYY/MM/DD/HH Puoi specificare un ulteriore prefisso da aggiungere davanti al prefisso del formato data e ora. Se il prefisso termina con una barra (/), viene visualizzato come cartella nel bucket Amazon S3.

7. Attendi qualche minuto affinché lo stream diventi attivo. È possibile utilizzare il comando Firehose describe-delivery-stream per verificare la proprietà. DeliveryStreamDescription.DeliveryStreamStatus Inoltre, annota il DeliveryStreamDescription.DeliveryStreamARN valore, poiché ti servirà in un passaggio successivo:

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket2",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. Crea il ruolo IAM che CloudWatch concede a Logs il permesso di inserire dati nel tuo flusso di consegna Firehose. In primo luogo, utilizza un editor di testo per creare una policy di attendibilità in un file `~/TrustPolicyForCWL.json`:

   Questa policy include una chiave di contesto della condizione globale `aws:SourceArn` per prevenire il problema di sicurezza noto come "confused deputy". Per ulteriori informazioni, consulta Prevenzione del "confused deputy".

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole",
       "Condition": {
            "ArnLike": {
                "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
            }
        }
     }
   }
   ```

9. Utilizza il comando create-role per creare un ruolo IAM, specificando il file della policy di attendibilità. Prendi nota del Role.Arn valore restituito, poiché ti servirà in un passaggio successivo:

   ```
   aws iam create-role \
   --role-name CWLtoKinesisFirehoseRole \
   --assume-role-policy-document file://~/TrustPolicyForCWL.json

   {
       "Role": {
           "AssumeRolePolicyDocument": {
               "Statement": {
                   "Action": "sts:AssumeRole",
                   "Effect": "Allow",
                   "Principal": {
                       "Service": "logs.amazonaws.com"
                   },
                   "Condition": {
                        "StringLike": {
                            "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                        }
   ```

```
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account. In primo luogo, utilizza un editor di testo per creare un file di policy di autorizzazioni (ad esempio, `~/PermissionsForCWL.json`):

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. Associa la policy di autorizzazioni al ruolo utilizzando il seguente comando put-role-policy:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Dopo che il flusso di distribuzione di Amazon Data Firehose è attivo e hai creato il ruolo IAM, puoi creare la politica di filtro delle sottoscrizioni a livello di account di CloudWatch Logs. La policy avvia immediatamente il flusso di dati di log in tempo reale dal gruppo di log scelto al flusso di distribuzione di Amazon Data Firehose:

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyFirehose" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisFirehoseRole", "DestinationArn":"arn:aws:firehose:us-
```

```
east-1:123456789012:deliverystream/delivery-stream-name", "FilterPattern": "Test",
 "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

13. Dopo aver impostato il filtro dell'abbonamento, CloudWatch Logs inoltra gli eventi di log in entrata che corrispondono al modello di filtro al flusso di consegna di Amazon Data Firehose.

    Il `selection-criteria` campo è facoltativo, ma è importante per escludere i gruppi di log che possono causare una ricorsione infinita di log da un filtro di abbonamento. Per ulteriori informazioni su questo problema e sulla determinazione dei gruppi di log da escludere, vedere. Prevenzione della ricorsione dei log Attualmente, NOT IN è l'unico operatore supportato per`selection-criteria`.

    I tuoi dati inizieranno a comparire in Amazon S3 in base all'intervallo di tempo del buffer impostato nel flusso di consegna di Amazon Data Firehose. Quando è trascorso tempo sufficiente, puoi verificare i dati controllando il Bucket Amazon S3.

```
aws s3api list-objects --bucket 'amzn-s3-demo-bucket2' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2023-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2023/10/29/00/my-delivery-stream-2023-10-29-00-35-40-
EXAMPLE-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
```

```
                "ID": "EXAMPLE6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'amzn-s3-demo-bucket2' --key 'firehose/2023/10/29/00/
my-delivery-stream-2023-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2023 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

I dati nell'oggetto di Amazon S3 vengono compressi nel formato gzip. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando i seguenti comandi Unix:

```
zcat testfile.gz
```

Cross-account Abbonamenti interregionali

Puoi collaborare con il proprietario di un AWS account diverso e ricevere i relativi eventi di registro sulle tue AWS risorse, ad esempio uno stream Amazon Kinesis o Amazon Data Firehose (operazione nota come condivisione dei dati tra account). Ad esempio, questi dati degli eventi di registro possono essere letti da uno stream centralizzato di Amazon Kinesis Data Streams o Firehose per eseguire elaborazioni e analisi personalizzate. L'elaborazione personalizzata è particolarmente utile quando collabori e analizzi dati tra più account.

Ad esempio, il gruppo di sicurezza di informazioni di un'azienda, potrebbe voler analizzare i dati per il rilevamento delle intrusioni in tempo reale o i comportamenti anomali, per poter condurre un'ispezione degli account in tutte le divisioni dell'azienda raccogliendo i log di produzione federata per l'elaborazione centralizzata. Un flusso in tempo reale di dati sugli eventi tra questi account può

essere assemblato e distribuito ai gruppi di sicurezza delle informazioni, che possono utilizzare Amazon Kinesis Data Streams per allegare i dati ai loro sistemi di analisi di sicurezza esistenti.

> **Note**
>
> Il gruppo di log e la destinazione devono trovarsi nella stessa regione. AWS Tuttavia, la risorsa AWS a cui punta la destinazione può trovarsi in una Regione diversa. Negli esempi delle sezioni seguenti, tutte le Region-specific risorse vengono create negli Stati Uniti orientali (Virginia settentrionale).

Se hai configurato AWS Organizations e stai utilizzando gli account dei membri, puoi utilizzare la centralizzazione dei log per raccogliere i dati di registro dagli account di origine in un account di monitoraggio centrale.

Quando lavori con gruppi di log centralizzati, puoi utilizzare queste dimensioni dei campi di sistema durante la creazione di filtri di abbonamento:

- `@aws.account`- Questa dimensione rappresenta l'ID dell' AWS account da cui ha avuto origine l'evento di registro.
- `@aws.region`- Questa dimensione rappresenta la AWS regione in cui è stato generato l'evento di registro.
- `@source.log`- Questa dimensione rappresenta il gruppo di log di origine da cui ha avuto origine l'evento di registro.

Queste dimensioni aiutano a identificare l'origine dei dati di registro, consentendo un filtraggio e un'analisi più granulari delle metriche derivate dai log centralizzati.

Argomenti

- Cross-account Condivisione dei dati di registro tra regioni utilizzando Amazon Kinesis Data Streams
- Cross-account Condivisione dei dati di registro tra regioni tramite Firehose
- Cross-account Abbonamenti a livello di account interregionali che utilizzano Amazon Kinesis Data Streams
- Cross-account Abbonamenti a livello di account interregionale che utilizzano Firehose

Cross-account Condivisione dei dati di registro tra regioni utilizzando Amazon Kinesis Data Streams

Durante la creazione di una sottoscrizione tra più account, è possibile specificare un singolo account o un'organizzazione come mittente. Nel caso in cui si specifichi un'organizzazione, la procedura illustrata di seguito consente a tutti gli account dell'organizzazione di inviare log all'account del destinatario.

Per condividere dati di log tra diversi account, è necessario stabilire un mittente e un ricevitore di dati di log:

- Mittente dei dati di registro: riceve le informazioni sulla destinazione dal destinatario e comunica a CloudWatch Logs che è pronto per inviare gli eventi di registro alla destinazione specificata. Nelle procedure descritte nel resto di questa sezione, il mittente dei dati di registro viene mostrato con un numero di conto fittizio pari AWS a 1111.

  Nel caso in cui più account all'interno di un'organizzazione inviano log a un account del destinatario, è possibile creare una policy che conceda a tutti gli account dell'organizzazione l'autorizzazione per eseguire tale operazione. Devi comunque impostare filtri di sottoscrizione separati per ciascun account del mittente.
- Destinatario dei dati di log: imposta una destinazione che incapsula un flusso di Amazon Kinesis Data Streams e comunica a Logs che il destinatario desidera ricevere i dati di CloudWatch log. Il destinatario quindi condivide le informazioni su questa destinazione con il mittente. Nelle procedure descritte nel resto di questa sezione, il destinatario dei dati di log viene mostrato con un numero di conto fittizio AWS pari a 9999.

Per iniziare a ricevere eventi di registro da utenti con più account, il destinatario dei dati di registro crea innanzitutto una destinazione Logs. CloudWatch Ogni destinazione è formata dai seguenti elementi chiave:

Nome della destinazione

Il nome della destinazione che intendi creare.

ARN di destinazione

L'Amazon Resource Name (ARN) della AWS risorsa che desideri utilizzare come destinazione del feed di abbonamento.

ARN del ruolo

Un ruolo AWS Identity and Access Management (IAM) che CloudWatch concede a Logs le autorizzazioni necessarie per inserire i dati nel flusso scelto.

Policy di accesso

Un documento della policy IAM (in formato JSON, scritto utilizzando la grammatica delle policy IAM) che controlla l'insieme degli utenti ai quali è concesso scrivere nella tua destinazione.

> **Note**
>
> Il gruppo di log e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la AWS risorsa a cui punta la destinazione può trovarsi in una regione diversa. Negli esempi delle sezioni seguenti, tutte le Region-specific risorse vengono create negli Stati Uniti orientali (Virginia settentrionale).

Argomenti

- Configurazione di una nuova sottoscrizione tra più account
- Aggiornamento di una sottoscrizione tra più account esistente

Configurazione di una nuova sottoscrizione tra più account

Segui la procedura riportata in queste sezioni per configurare una nuova sottoscrizione del log tra più account.

Argomenti

- Passaggio 1: creazione di una destinazione
- Fase 2: creazione di un ruolo IAM (solo se si utilizza un'organizzazione)
- Fase 3: autorizzazioni Add/validate IAM per la destinazione tra più account
- Passaggio 4: creazione di un filtro di sottoscrizione
- Convalida del flusso dei log eventi
- Modifica dell'appartenenza alla destinazione in fase di runtime

Passaggio 1: creazione di una destinazione

> **Important**
>
> Tutte le fasi di questa procedura devono essere eseguite nell'account del destinatario dei dati di log.

Per questo esempio, l'account del destinatario dei dati di registro ha un ID AWS account di 9999, mentre l'ID dell'account del mittente AWS dei dati di registro è 1111.

Questo esempio crea una destinazione utilizzando un flusso di Amazon Kinesis Data Streams chiamato RecipientStream e un ruolo che consente a CloudWatch Logs di scrivere dati su di esso.

Quando viene creata la destinazione, CloudWatch Logs invia un messaggio di prova alla destinazione per conto dell'account del destinatario. Quando il filtro di sottoscrizione è attivo in un secondo momento, CloudWatch Logs invia gli eventi di registro alla destinazione per conto dell'account di origine.

Creazione di una destinazione

1. Nell'account del destinatario, crea uno stream di destinazione in Amazon Kinesis Data Streams. Al prompt dei comandi, digita:

   ```
   aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
   ```

2. Attendi finché il flusso non diventa attivo. Puoi utilizzare il comando aws kinesis describe-stream per verificare la proprietà. StreamDescription.StreamStatus Inoltre, prendi nota del StreamDescription.StreamARN valore perché lo passerai a Logs in un secondo momento: CloudWatch

   ```
   aws kinesis describe-stream --stream-name "RecipientStream"
   {
     "StreamDescription": {
       "StreamStatus": "ACTIVE",
       "StreamName": "RecipientStream",
       "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
       "Shards": [
         {
           "ShardId": "shardId-000000000000",
           "HashKeyRange": {
   ```

```
            "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
            "StartingHashKey": "0"
          },
          "SequenceNumberRange": {
            "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702l8EXAMPLE"
          }
        }
      ]
    }
}
```

Potrebbero essere necessari uno o due minuti perché il flusso sia in stato attivo.

3. Crea il ruolo IAM che CloudWatch concede a Logs il permesso di inserire dati nel tuo stream. Innanzitutto, devi creare una politica di fiducia in un file TrustPolicyForCWL.json ~/. Utilizza un editor di testo per creare questo file di policy, non utilizzare la console IAM.

   Questa policy include una chiave di contesto della condizione globale `aws:SourceArn` che specifica il `sourceAccountId` per prevenire il problema di sicurezza noto come "confused deputy". Se non conosci ancora l'ID dell'account di origine nella prima chiamata, consigliamo di inserire l'ARN di destinazione nel campo ARN di origine. Nelle chiamate successive, è necessario impostare l'ARN di origine come l'ARN di origine effettivo raccolto dalla prima chiamata. Per ulteriori informazioni, consulta Prevenzione del "confused deputy".

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. Utilizza il comando aws iam create-role per creare il ruolo IAM, specificando il file della policy di attendibilità. Prendi nota del Role.Arn valore restituito perché verrà passato anche a CloudWatch Logs in un secondo momento:

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Crea una politica di autorizzazioni per definire quali azioni può eseguire CloudWatch Logs sul tuo account. Innanzitutto, utilizza un editor di testo per creare una politica di autorizzazioni in un file ~/: PermissionsForCWL.json

```
{
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Associa la policy delle autorizzazioni al ruolo utilizzando il comando aws iam put-role-policy:

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Dopo che lo stream è attivo e hai creato il ruolo IAM, puoi creare la destinazione CloudWatch Logs.

   a. In questa fase alla tua destinazione non si associa una policy d'accesso predefinita e costituisce solo la prima delle due fasi necessarie per completare la creazione della destinazione. Prendi nota di DestinationArn ciò che viene restituito nel payload:

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. Dopo aver completato la fase 7, nell'account del destinatario dei dati di log associa alla destinazione una policy d'accesso predefinita. Questa policy deve specificare i log: PutSubscriptionFilter action e concede all'account mittente l'autorizzazione ad accedere alla destinazione.

      La politica concede l'autorizzazione all' AWS account che invia i log. Puoi specificare solo questo account nella policy oppure, se l'account del mittente è membro di un'organizzazione, la policy può specificare l'ID dell'organizzazione. In questo modo, puoi

creare una sola policy per consentire a più account di un'organizzazione di inviare log a questo account di destinazione.

Utilizza un editor di testo per creare un file denominato `~/AccessPolicy.json` con una delle seguenti istruzioni di policy.

Questa prima policy di esempio consente a tutti gli account dell'organizzazione che hanno un ID di `o-1234567890` di inviare log all'account del destinatario.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-1234567890"
                    ]
                }
            }
        }
    ]
}
```

Nell'esempio seguente, solo l'account del mittente dei dati di log (111111111111) può inviare log all'account del destinatario dei dati di log.

JSON

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

```
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111111111111"
            },
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination"
        }
    ]
}
```

c. Collega la policy creata nel passaggio precedente alla destinazione.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Questa politica di accesso consente agli utenti dell' AWS Account con ID 1111 di chiamare la destinazione con ARN arn:aws:logs PutSubscriptionFilter ::44449999:Destination:TestDestination. *region* Qualsiasi tentativo di chiamata da parte di un altro utente verso questa destinazione verrà rifiutato. PutSubscriptionFilter

Per convalidare i privilegi di un utente su una policy d'accesso predefinita, consulta Utilizzo dello strumento di validazione delle policy nella guida per l'utente IAM.

Al termine, se utilizzi le autorizzazioni AWS Organizations per più account, segui i passaggi indicati. Fase 2: creazione di un ruolo IAM (solo se si utilizza un'organizzazione) Se le autorizzazioni vengono concesse direttamente all'altro account anziché utilizzare Organizations, puoi saltare tale passaggio e procedere alla sezione Passaggio 4: creazione di un filtro di sottoscrizione.

Fase 2: creazione di un ruolo IAM (solo se si utilizza un'organizzazione)

Se nella sezione precedente hai creato la destinazione utilizzando una policy di accesso che concede le autorizzazioni all'organizzazione in cui è presente l'account 111111111111, invece di concederle direttamente all'account 111111111111, segui i passaggi descritti di seguito. In caso contrario, passa alla sezione Passaggio 4: creazione di un filtro di sottoscrizione.

I passaggi di questa sezione creano un ruolo IAM, che CloudWatch può presupporre e convalidare se l'account mittente dispone dell'autorizzazione per creare un filtro di abbonamento in base alla destinazione del destinatario.

Per l'account mittente, segui la procedura descritta in questa sezione. Il ruolo deve esistere nell'account mittente e devi specificare l'ARN di questo ruolo nel filtro di sottoscrizione. In questo esempio, l'account mittente è denominato `111111111111`.

Per creare il ruolo IAM necessario per le sottoscrizioni ai log tra account, utilizzare AWS Organizations

1. Crea la policy di attendibilità seguente in un file `/TrustPolicyForCWLSubscriptionFilter.json`. Utilizza un editor di testo per creare questo file di policy; non utilizzare la console IAM.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. Crea il ruolo IAM che utilizza questa policy. Prendi nota del valore `Arn` restituito dal comando, sarà necessario in seguito in questa procedura. In questo esempio, il ruolo in fase di creazione è denominato `CWLtoSubscriptionFilterRole`.

   ```
   aws iam create-role \
        --role-name CWLtoSubscriptionFilterRole \
        --assume-role-policy-document file://~/
   TrustPolicyForCWLSubscriptionFilter.json
   ```

3. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account.

   a. In primo luogo, utilizza un editor di testo per creare la policy di autorizzazione seguente in un file denominato `~/PermissionsForCWLSubscriptionFilter.json`.

      ```
      {
          "Statement": [
              {
      ```

```
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Inserisci il comando seguente per associare la policy di autorizzazione appena creata al ruolo creato nella fase 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Al termine dell'operazione, passa alla sezione Passaggio 4: creazione di un filtro di sottoscrizione.

Fase 3: autorizzazioni Add/validate IAM per la destinazione tra più account

In base alla AWS logica di valutazione delle politiche tra account, per accedere a qualsiasi risorsa tra account (come uno stream Kinesis o Firehose utilizzato come destinazione per un filtro di abbonamento) è necessario disporre di una politica basata sull'identità nell'account di invio che fornisca l'accesso esplicito alla risorsa di destinazione tra account. Per ulteriori informazioni sulla logica di valutazione delle politiche, vedere Logica di valutazione delle politiche. Cross-account

Puoi collegare la policy basata sull'identità al ruolo IAM o all'utente IAM che stai utilizzando per creare il filtro di sottoscrizione. Questa policy deve essere presente nell'account mittente. Se utilizzi il ruolo di amministratore per creare il filtro di sottoscrizione, puoi saltare questo passaggio e passare a Passaggio 4: creazione di un filtro di sottoscrizione.

Aggiunta o convalida delle autorizzazioni IAM necessarie per più account

1. Inserisci il seguente comando per verificare quale ruolo IAM o utente IAM viene utilizzato per eseguire i comandi di AWS log.

```
aws sts get-caller-identity
```

Il comando restituisce un output simile al seguente:

```
{
"UserId": "User ID",
"Account": "sending account id",
"Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
}
```

Prendi nota del valore rappresentato da *RoleName* o*UserName*.

2. Console di gestione AWS Accedi all'account di invio e cerca le policy allegate con il ruolo IAM o l'utente IAM restituito nell'output del comando inserito nel passaggio 1.
3. Verifica che le policy associate a questo ruolo o utente forniscano autorizzazioni esplicite per richiamare `logs:PutSubscriptionFilter` sulla risorsa di destinazione multi-account.

   La seguente policy fornisce le autorizzazioni per creare un filtro di sottoscrizione su qualsiasi risorsa di destinazione solo in un singolo AWS account, account999999999999:

   JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
        {
            "Sid": "AllowSubscriptionFiltersOnAccountResources",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

   La seguente politica fornisce le autorizzazioni per creare un filtro di abbonamento solo su una risorsa di destinazione specifica denominata `sampleDestination` in un singolo AWS account, account: `123456789012`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSubscriptionFiltersonAccountResource",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
}
```

Passaggio 4: creazione di un filtro di sottoscrizione

Una volta creata una destinazione, l'account del destinatario dei dati di log può condividere l'ARN di destinazione (arn:aws:logs:us-east-1:999999999999:destination:testDestination) con altri account AWS , perché questi possano inviare eventi di log alla stessa destinazione. Tali utenti di questi account di invio possono creare un filtro di sottoscrizione sui rispettivi gruppi di log sulla destinazione. Il filtro di sottoscrizione avvia immediatamente il flusso di dati di log in tempo reale dal gruppo di log selezionato alla destinazione specificata.

> **Note**
>
> Se stai concedendo le autorizzazioni per il filtro di sottoscrizione a un'intera organizzazione, dovrai utilizzare l'ARN del ruolo IAM che hai creato in Fase 2: creazione di un ruolo IAM (solo se si utilizza un'organizzazione).

Nell'esempio seguente, viene creato un filtro di sottoscrizione in un account di invio. Il filtro è associato a un gruppo di log contenente AWS CloudTrail eventi in modo che ogni attività registrata effettuata dalle AWS credenziali «Root» venga recapitata alla destinazione creata in precedenza. Tale destinazione incapsula uno stream chiamato "». RecipientStream

Il resto dei passaggi nelle sezioni seguenti presuppone che tu abbia seguito le istruzioni contenute in Invio di CloudTrail eventi ai CloudWatch registri nella Guida per l'AWS CloudTrail utente e abbia creato un gruppo di log che contiene i tuoi CloudTrail eventi. Questi passaggi presuppongono che il nome di questo gruppo di log sia `CloudTrail/logs`.

Quando inserisci il comando seguente, assicurati di aver effettuato l'accesso come utente IAM o di utilizzare il ruolo IAM per cui hai aggiunto la policy in Fase 3: autorizzazioni Add/validate IAM per la destinazione tra più account.

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Il gruppo di log e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la destinazione può puntare a una AWS risorsa come un flusso di Amazon Kinesis Data Streams che si trova in una regione diversa.

Convalida del flusso dei log eventi

Dopo aver creato il filtro di iscrizione, CloudWatch Logs inoltra tutti gli eventi di log in entrata che corrispondono al pattern di filtro allo stream incapsulato nello stream di destinazione chiamato "». RecipientStream Il proprietario della destinazione può verificare che ciò stia accadendo utilizzando il comando aws kinesis get-shard-iterator per acquisire uno shard Amazon Kinesis Data Streams e utilizzando il comando aws kinesis get-records per recuperare alcuni record di Amazon Kinesis Data Streams:

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}
```

```
aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Potrebbe essere necessario eseguire nuovamente il comando get-records alcune volte prima che Amazon Kinesis Data Streams inizi a restituire i dati.

Dovresti vedere una risposta con una serie di record di Amazon Kinesis Data Streams. L'attributo di dati nel record Amazon Kinesis Data Streams viene compresso in formato gzip e quindi codificato in base64. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando i seguenti comandi Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

I dati con codifica base64 e decompressi sono in formato JSON con la seguente struttura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
```

```
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Gli elementi chiave della struttura di dati sono i seguenti:

owner

L'ID dell' AWS account dei dati di registro di origine.

logGroup

Nome del gruppo di log dei dati di log originari.

logStream

Nome del flusso di log dei dati di log originari.

subscriptionFilters

Elenco dei nomi di filtro sottoscrizione che corrispondono con i dati di log originari.

messageType

I messaggi di dati usano il tipo "DATA_MESSAGE". A volte CloudWatch i log possono emettere record di Amazon Kinesis Data Streams con un tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

logEvents

I dati di log effettivi, rappresentati come una varietà di record di eventi di log. La proprietà ID è un identificatore univoco per ogni eventi di log.

Modifica dell'appartenenza alla destinazione in fase di runtime

Potrebbero verificarsi situazioni in cui potresti dover aggiungere o rimuovere l'adesione di alcuni utenti da una destinazione da te posseduta. Puoi utilizzare il commando `put-destination-policy` sulla destinazione con una nuova policy di accesso. Nell'esempio seguente, si impedisce

a un account aggiunto in precedenza 111111111111 di inviare ulteriori dati di log, mentre l'account 222222222222 viene abilitato.

1. Recupera la policy attualmente associata alla destinazione TestDestination e prendi nota di: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Aggiorna la policy per riflettere l'arresto di tale account 111111111111, mentre l'account 222222222222 viene abilitato. Inserisci questa politica nel file ~/: NewAccessPolicy.json

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "222222222222"
            },
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination"
```

```
        }
    ]
}
```

3. Chiama PutDestinationPolicy per associare la policy definita nel NewAccessPolicy.json file alla destinazione:

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Alla fine ciò disabiliterà gli eventi di log dall'ID account 111111111111. I log eventi provenienti dall'ID account 222222222222 iniziano a fluire verso la destinazione non appena il proprietario dell'account 222222222222 crea un filtro di sottoscrizione.

Aggiornamento di una sottoscrizione tra più account esistente

Se hai una sottoscrizione del log tra più account in cui l'account di destinazione concede le autorizzazioni solo a specifici account del mittente e desideri aggiornare questa sottoscrizione in modo che l'account di destinazione conceda l'accesso a tutti gli account di un'organizzazione, segui la procedura descritta in questa sezione.

Argomenti

- Fase 1: aggiornamento dei filtri di sottoscrizione
- Fase 2: aggiornamento della policy di accesso alla destinazione esistente

Fase 1: aggiornamento dei filtri di sottoscrizione

> **Note**
>
> Questo passaggio è necessario solo per le sottoscrizioni tra più account per i log creati dai servizi elencati in Abilita la registrazione da AWS services. Se non stai lavorando con log creati da uno di questi gruppi di log, puoi passare alla sezione Fase 2: aggiornamento della policy di accesso alla destinazione esistente.

In alcuni casi, devi aggiornare i filtri di sottoscrizione in tutti gli account del mittente che inviano log all'account di destinazione. L'aggiornamento aggiunge un ruolo IAM, che CloudWatch può

presupporre e convalidare che l'account mittente disponga dell'autorizzazione per inviare i log all'account del destinatario.

Segui la procedura descritta in questa sezione per ogni account del mittente che desideri aggiornare in modo da utilizzare l'ID dell'organizzazione per le autorizzazioni di sottoscrizione tra più account.

Negli esempi di questa sezione sono già stati creati dei filtri di sottoscrizione negli account 111111111111 e 222222222222 per l'invio di log all'account 999999999999. I valori del filtro di sottoscrizione esistenti sono i seguenti:

```
## Existing Subscription Filter parameter values
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Se è necessario trovare i valori dei parametri del filtro di sottoscrizione correnti, digita il comando seguente.

```
aws logs describe-subscription-filters
    \ --log-group-name "my-log-group-name"
```

Aggiornamento di un filtro di sottoscrizione per iniziare a utilizzare gli ID dell'organizzazione per le autorizzazioni del log tra più account

1. Crea la policy di attendibilità seguente in un file `~/TrustPolicyForCWL.json`. Utilizza un editor di testo per creare questo file di policy; non utilizzare la console IAM.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. Crea il ruolo IAM che utilizza questa policy. Prendi nota del valore `Arn` del valore `Arn` restituito dal comando, sarà necessario in seguito in questa procedura. In questo esempio, il ruolo in fase di creazione è denominato `CWLtoSubscriptionFilterRole`.

   ```
   aws iam create-role
   ```

```
    \ --role-name CWLtoSubscriptionFilterRole
    \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

3. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account.

   a. In primo luogo, utilizza un editor di testo per creare la policy di autorizzazione seguente in un file denominato `/PermissionsForCWLSubscriptionFilter.json`.

   ```
   {
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "logs:PutLogEvents",
               "Resource": "arn:aws:logs:region:111111111111:log-
   group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
           }
       ]
   }
   ```

   b. Inserisci il comando seguente per associare la policy di autorizzazione appena creata al ruolo creato nella fase 2.

   ```
   aws iam put-role-policy
       --role-name CWLtoSubscriptionFilterRole
       --policy-name Permissions-Policy-For-CWL-Subscription-filter
       --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
   ```

4. Inserisci il comando seguente per aggiornare il filtro di sottoscrizione.

```
aws logs put-subscription-filter
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn
 "arn:aws:logs:region:999999999999:destination:testDestination"
    \ --role-arn "arn:aws:iam::111111111111:role/CWLtoSubscriptionFilterRole"
```

Fase 2: aggiornamento della policy di accesso alla destinazione esistente

Dopo aver aggiornato i filtri di sottoscrizione in tutti gli account del mittente, è possibile aggiornare la policy di accesso alla destinazione nell'account del destinatario.

Negli esempi seguenti, l'account del destinatario è `999999999999` e la destinazione è denominata `testDestination`.

L'aggiornamento abilita tutti gli account che fanno parte dell'organizzazione con ID `o-1234567890` per l'invio di log all'account del destinatario. Solo gli account con filtri di sottoscrizione creati invieranno effettivamente log all'account del destinatario.

Aggiornamento della policy di accesso alla destinazione nell'account del destinatario per iniziare a utilizzare un ID dell'organizzazione per le autorizzazioni

1. Nell'account del destinatario, utilizza un editor di testo per creare un file `~/AccessPolicy.json` con i seguenti contenuti.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-1234567890"
                    ]
                }
            }
        }
    ]
}
```

2. Digita il seguente comando per collegare la policy appena creata alla destinazione esistente. Per aggiornare una destinazione in modo da utilizzare una politica di accesso con un ID dell'organizzazione anziché una politica di accesso che elenca ID AWS account specifici, includi il `force` parametro.

> **Warning**
>
> Se stai utilizzando i log inviati da un AWS servizio elencato in[Abilita la registrazione da AWS services](), prima di eseguire questo passaggio, devi prima aver aggiornato i filtri di abbonamento in tutti gli account mittente, come spiegato in[Fase 1: aggiornamento dei filtri di sottoscrizione]().

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Cross-account Condivisione dei dati di registro tra regioni tramite Firehose

Per condividere dati di log tra diversi account, è necessario stabilire un mittente e un ricevitore di dati di log:

- Mittente dei dati di registro: riceve le informazioni sulla destinazione dal destinatario e comunica a CloudWatch Logs che è pronto a inviare gli eventi di registro alla destinazione specificata. Nelle procedure descritte nel resto di questa sezione, il mittente dei dati di registro viene mostrato con un numero di conto fittizio pari AWS a 1111.
- Destinatario dei dati di log: imposta una destinazione che incapsula un flusso di Amazon Kinesis Data Streams e comunica a Logs che il destinatario desidera ricevere i dati di CloudWatch log. Il destinatario quindi condivide le informazioni su questa destinazione con il mittente. Nelle procedure descritte nel resto di questa sezione, al destinatario dei dati di log viene mostrato un numero di conto fittizio AWS pari a 2222.

L'esempio in questa sezione utilizza un flusso di distribuzione Firehose con storage Amazon S3. Puoi anche configurare i flussi di consegna Firehose con impostazioni diverse. Per ulteriori informazioni, consulta [Creazione di un flusso di consegna Firehose.]()

> **Note**
>
> Il gruppo di log e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la AWS risorsa a cui punta la destinazione può trovarsi in una regione diversa.

> **Note**
>
> È supportato il filtro di abbonamento Firehose per lo stesso account e lo stesso flusso di consegna tra regioni.

Argomenti

- Fase 1: Creare un flusso di consegna Firehose
- Fase 2: creazione di una destinazione
- Fase 3: autorizzazioni Add/validate IAM per la destinazione tra più account
- Passaggio 4: creazione di un filtro di sottoscrizione
- Convalida del flusso dei log eventi
- Modifica dell'appartenenza alla destinazione durante il runtime

Fase 1: Creare un flusso di consegna Firehose

> **Important**
>
> Prima di completare i passaggi seguenti, devi utilizzare una policy di accesso, in modo che Firehose possa accedere al tuo bucket Amazon S3. Per ulteriori informazioni, consulta Controlling Access nella Amazon Data Firehose Developer Guide.
> Tutti i passaggi in questa sezione (Fase 1) devono essere eseguiti nell'account del destinatario dei dati di log.
> La regione Stati Uniti orientali (Virginia settentrionale) viene utilizzata nei comandi di esempio. Sostituiscila con la Regione corretta per l'implementazione.

Per creare un flusso di consegna Firehose da utilizzare come destinazione

1. Crea un bucket Amazon S3:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-1
```

2. Crea il ruolo IAM che concede a Firehose il permesso di inserire dati nel bucket.

   a. In primo luogo, utilizza un editor di testo per creare una policy di attendibilità in un file *~/*`TrustPolicyForFirehose.json`.

   ```
   { "Statement": { "Effect": "Allow", "Principal": { "Service":
    "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
    { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
   ```

   b. Crea il ruolo IAM, specificando il file della policy di attendibilità appena creato.

   ```
   aws iam create-role \
       --role-name FirehosetoS3Role \
       --assume-role-policy-document file://~/TrustPolicyForFirehose.json
   ```

   c. L'output di questo comando risulterà simile al seguente: Annotare i Nome ruolo e ARN ruolo.

   ```
   {
       "Role": {
           "Path": "/",
           "RoleName": "FirehosetoS3Role",
           "RoleId": "AROAR3BXASEKW7K635M53",
           "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
           "CreateDate": "2021-02-02T07:53:10+00:00",
           "AssumeRolePolicyDocument": {
               "Statement": {
                   "Effect": "Allow",
                   "Principal": {
                       "Service": "firehose.amazonaws.com"
                   },
                   "Action": "sts:AssumeRole",
                   "Condition": {
                       "StringEquals": {
                           "sts:ExternalId": "222222222222"
                       }
                   }
               }
           }
       }
   ```

```
}
```

3. Crea una politica di autorizzazioni per definire le azioni che Firehose può eseguire nel tuo account.

   a. In primo luogo, utilizza un editor di testo per creare la policy di autorizzazione seguente in un file denominato `~/PermissionsForFirehose.json`. A seconda del caso d'uso, potresti dover aggiungere altre autorizzazioni a questo file.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
    }]
}
```

   b. Inserisci il comando seguente per associare la policy di autorizzazioni appena creata con il ruolo IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Inserisci il seguente comando per creare il flusso di consegna di Firehose. Sostituisci *my-role-arn* e *amzn-s3-demo-bucket2-arn* con i valori corretti per la tua distribuzione.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::amzn-s3-demo-bucket"}'
```

   L'output visualizzato dovrebbe essere simile al seguente:

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Fase 2: creazione di una destinazione

> **Important**
>
> Tutte le fasi di questa procedura devono essere eseguite nell'account del destinatario dei dati di log.

Quando viene creata la destinazione, CloudWatch Logs invia un messaggio di prova alla destinazione per conto dell'account del destinatario. Quando il filtro di sottoscrizione è attivo in un secondo momento, CloudWatch Logs invia gli eventi di registro alla destinazione per conto dell'account di origine.

Creazione di una destinazione

1. Attendi che lo stream Firehose in cui hai creato Fase 1: Creare un flusso di consegna Firehose diventi attivo. È possibile utilizzare il comando seguente per verificare la StreamDescription.StreamStatus proprietà.

   ```
   aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
   ```

   Inoltre, prendi nota del DeliveryStreamDescription.DeliveryStreamARN valore, perché dovrai utilizzarlo in un passaggio successivo. Esempio di output di questo comando:

   ```
   {
       "DeliveryStreamDescription": {
           "DeliveryStreamName": "my-delivery-stream",
           "DeliveryStreamARN": "arn:aws:firehose:us-
   east-1:222222222222:deliverystream/my-delivery-stream",
           "DeliveryStreamStatus": "ACTIVE",
           "DeliveryStreamEncryptionConfiguration": {
               "Status": "DISABLED"
           },
   ```

```
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

Potrebbero essere necessari uno o due minuti che il flusso di consegna venga visualizzato nello stato attivo.

2. Quando il flusso di distribuzione è attivo, crea il ruolo IAM che concederà a CloudWatch Logs il permesso di inserire dati nel tuo stream Firehose. Innanzitutto, devi creare una politica di fiducia in un file TrustPolicyForCWL.json ~/. Utilizza un editor di testo per creare questa policy. Per ulteriori informazioni sugli endpoint CloudWatch Logs, consulta Amazon CloudWatch Logs endpoint and quotas.

   Questa policy include una chiave di contesto della condizione globale `aws:SourceArn` che specifica il `sourceAccountId` per prevenire il problema di sicurezza noto come "confused deputy". Se non conosci ancora l'ID dell'account di origine nella prima chiamata, consigliamo di inserire l'ARN di destinazione nel campo ARN di origine. Nelle chiamate successive, è necessario impostare l'ARN di origine come l'ARN di origine effettivo raccolto dalla prima chiamata. Per ulteriori informazioni, consulta Prevenzione del "confused deputy".

   ```
   {
       "Statement": {
           "Effect": "Allow",
           "Principal": {
               "Service": "logs.region.amazonaws.com"
           },
           "Action": "sts:AssumeRole",
           "Condition": {
               "ArnLike": {
                   "aws:SourceArn": [
                       "arn:aws:logs:region:sourceAccountId:*",
                       "arn:aws:logs:region:recipientAccountId:*"
                   ]
               }
           }
        }
   }
   ```

3. Utilizza il comando aws iam create-role per creare il ruolo IAM, specificando il file della policy di attendibilità appena creato.

   ```
   aws iam create-role \
         --role-name CWLtoKinesisFirehoseRole \
         --assume-role-policy-document file://~/TrustPolicyForCWL.json
   ```

Di seguito è riportato un output di esempio. Prendi nota del valore `Role.Arn` restituito, perché dovrai usarlo in una fase successiva.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2021-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "ArnLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account. Innanzitutto, utilizza un editor di testo per creare una politica di autorizzazioni in un file ~/: PermissionsForCWL.json

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
```

```
}
```

5. Associa la policy di autorizzazioni al ruolo inserendo il comando seguente:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Dopo che il flusso di distribuzione di Firehose è attivo e hai creato il ruolo IAM, puoi creare la CloudWatch destinazione Logs.

   a. In questa fase alla tua destinazione non verrà associata una policy d'accesso predefinita e costituisce solo la prima delle due fasi necessarie per completare la creazione della destinazione. Annota dell'ARN della nuova destinazione restituito nel payload, perché lo utilizzerai come `destination.arn` in una fase successiva.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

   b. Dopo aver completato la fase precedente, nell'account del destinatario dei dati di log (222222222222) associa alla destinazione una policy d'accesso.

      Questa policy consente all'account mittente dei dati di log (111111111111) di accedere alla destinazione all'interno dell'account destinatario dei dati di log (222222222222). Puoi usare un editor di testo per inserire questa policy nel file ~/: AccessPolicy.json

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. In questo modo viene creata una policy che definisce chi ha accesso in scrittura alla destinazione. Questa policy deve specificare l'PutSubscriptionFilterazione logs: per accedere alla destinazione. Cross-account gli utenti utilizzeranno l'PutSubscriptionFilterazione per inviare gli eventi di registro alla destinazione:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Fase 3: autorizzazioni Add/validate IAM per la destinazione tra più account

In base alla AWS logica di valutazione delle politiche tra account, per accedere a qualsiasi risorsa tra account (come uno stream Kinesis o Firehose utilizzato come destinazione per un filtro di abbonamento) è necessario disporre di una politica basata sull'identità nell'account di invio che fornisca l'accesso esplicito alla risorsa di destinazione tra account. Per ulteriori informazioni sulla logica di valutazione delle politiche, vedere Logica di valutazione delle politiche. Cross-account

Puoi collegare la policy basata sull'identità al ruolo IAM o all'utente IAM che stai utilizzando per creare il filtro di sottoscrizione. Questa policy deve essere presente nell'account mittente. Se utilizzi il ruolo

di amministratore per creare il filtro di sottoscrizione, puoi saltare questo passaggio e passare a Passaggio 4: creazione di un filtro di sottoscrizione.

Aggiunta o convalida delle autorizzazioni IAM necessarie per più account

1. Inserisci il seguente comando per verificare quale ruolo IAM o utente IAM viene utilizzato per eseguire i comandi di AWS log.

   ```
   aws sts get-caller-identity
   ```

   Il comando restituisce un output simile al seguente:

   ```
   {
   "UserId": "User ID",
   "Account": "sending account id",
   "Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
   }
   ```

   Prendi nota del valore rappresentato da *RoleName* o*UserName*.

2. Console di gestione AWS Accedi all'account di invio e cerca le policy allegate con il ruolo IAM o l'utente IAM restituito nell'output del comando inserito nel passaggio 1.
3. Verifica che le policy associate a questo ruolo o utente forniscano autorizzazioni esplicite per richiamare `logs:PutSubscriptionFilter` sulla risorsa di destinazione multi-account.

   La seguente policy fornisce le autorizzazioni per creare un filtro di sottoscrizione su qualsiasi risorsa di destinazione solo in un singolo AWS account, account999999999999:

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [
           {
               "Sid":
    "AllowSubscriptionFiltersOnAnyResourceInOneSpecificAccount",
               "Effect": "Allow",
               "Action": "logs:PutSubscriptionFilter",
               "Resource": [
                   "arn:aws:logs:*:*:log-group:*",
   ```

```
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

La seguente politica fornisce le autorizzazioni per creare un filtro di abbonamento solo su una risorsa di destinazione specifica denominata `sampleDestination` in un singolo AWS account, account: `123456789012`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
        "Sid": "AllowSubscriptionFiltersOnSpecificResource",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

Passaggio 4: creazione di un filtro di sottoscrizione

Passare all'account di invio, che in questo esempio è 111111111111. Verrà ora creato il filtro di sottoscrizione nell'account di invio. In questo esempio, il filtro è associato a un gruppo di log contenente AWS CloudTrail eventi in modo che ogni attività registrata effettuata dalle AWS credenziali «Root» venga recapitata alla destinazione creata in precedenza. Per ulteriori informazioni su come inviare AWS CloudTrail eventi ai CloudWatch log, vedere Invio di CloudTrail eventi ai CloudWatch log nella Guida per l'AWS CloudTrail utente.

Quando inserisci il comando seguente, assicurati di aver effettuato l'accesso come utente IAM o di utilizzare il ruolo IAM per cui hai aggiunto la policy in Fase 3: autorizzazioni Add/validate IAM per la destinazione tra più account.

```
aws logs put-subscription-filter \
    --log-group-name "aws-cloudtrail-logs-111111111111-300a971e" \
    --filter-name "firehose_test" \
    --filter-pattern "{$.userIdentity.type = AssumedRole}" \
    --destination-arn "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
```

Il gruppo di log e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la destinazione può puntare a una AWS risorsa come uno stream Firehose che si trova in una regione diversa.

Convalida del flusso dei log eventi

Dopo aver creato il filtro di abbonamento, CloudWatch Logs inoltra tutti gli eventi di registro in entrata che corrispondono al modello di filtro al flusso di consegna di Firehose. I dati iniziano a comparire nel bucket Amazon S3 in base all'intervallo di tempo del buffer impostato nel flusso di consegna Firehose. Quando è trascorso tempo sufficiente, puoi verificare i dati controllando il bucket Amazon S3. Per controllare il bucket, inserisci il comando seguente:

```
aws s3api list-objects --bucket 'amzn-s3-demo-bucket'
```

L'output di questo comando sarà simile al seguente:

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2021-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
```

```
    ]
}
```

È quindi possibile recuperare un oggetto specifico dal bucket inserendo il seguente comando. Sostituisci il valore di `key` con il valore trovato nel comando precedente.

```
aws s3api get-object --bucket 'amzn-s3-demo-bucket' --key '2021/02/02/08/my-delivery-stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

I dati nell'oggetto di Amazon S3 vengono compressi nel formato gzip. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando uno dei seguenti comandi:

Linux:

```
zcat testfile.gz
```

macOS:

```
zcat <testfile.gz
```

Modifica dell'appartenenza alla destinazione durante il runtime

Potrebbero verificarsi situazioni in cui devi aggiungere o rimuovere mittenti di log da una destinazione da te posseduta. Puoi utilizzare l'PutDestinationPolicyazione sulla tua destinazione con una nuova politica di accesso. Nell'esempio seguente, si impedisce a un account aggiunto in precedenza 111111111111 di inviare ulteriori dati di log, mentre l'account 333333333333 viene abilitato.

1. Recupera la policy attualmente associata alla destinazione TestDestination e prendi nota di: AccessPolicy

   ```
   aws logs describe-destinations \
       --destination-name-prefix "testFirehoseDestination"

   {
       "destinations": [
           {
               "destinationName": "testFirehoseDestination",
               "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-delivery-stream",
   ```

```
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. Aggiorna la policy per riflettere che l'account 111111111111 è stoppato, mentre l'account 333333333333 viene abilitato. Inserisci questa politica nel file ~/: NewAccessPolicy.json

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement" : [
       {
         "Sid" : "",
         "Effect" : "Allow",
         "Principal" : {
           "AWS" : "333333333333 "
         },
         "Action" : "logs:PutSubscriptionFilter",
         "Resource" : "arn:aws:logs:us-
   east-1:222222222222:destination:testFirehoseDestination"
       }
     ]
   }
   ```

3. Utilizzate il comando seguente per associare la policy definita nel NewAccessPolicy.json file alla destinazione:

   ```
   aws logs put-destination-policy \
       --destination-name "testFirehoseDestination" \

       --access-policy file://~/NewAccessPolicy.json
   ```

Questo alla fine disabilita il log eventi dall'ID account 111111111111. I log eventi provenienti dall'ID account 333333333333 iniziano a fluire verso la destinazione non appena il proprietario dell'account 333333333333 crea un filtro di sottoscrizione.

Cross-account Abbonamenti a livello di account interregionali che utilizzano Amazon Kinesis Data Streams

Durante la creazione di una sottoscrizione tra più account, è possibile specificare un singolo account o un'organizzazione come mittente. Nel caso in cui si specifichi un'organizzazione, la procedura illustrata di seguito consente a tutti gli account dell'organizzazione di inviare log all'account del destinatario.

Per condividere dati di log tra diversi account, è necessario stabilire un mittente e un ricevitore di dati di log:

- Mittente dei dati di registro: riceve le informazioni sulla destinazione dal destinatario e comunica a CloudWatch Logs che è pronto per inviare gli eventi di registro alla destinazione specificata. Nelle procedure descritte nel resto di questa sezione, il mittente dei dati di registro viene mostrato con un numero di conto fittizio pari AWS a 1111.

  Nel caso in cui più account all'interno di un'organizzazione inviano log a un account del destinatario, è possibile creare una policy che conceda a tutti gli account dell'organizzazione l'autorizzazione per eseguire tale operazione. Devi comunque impostare filtri di sottoscrizione separati per ciascun account del mittente.
- Destinatario dei dati di log: imposta una destinazione che incapsula un flusso di Amazon Kinesis Data Streams e comunica a Logs che il destinatario desidera ricevere i dati di CloudWatch log. Il destinatario quindi condivide le informazioni su questa destinazione con il mittente. Nelle procedure descritte nel resto di questa sezione, al destinatario dei dati di log viene mostrato un numero di conto fittizio AWS pari a 9999.

Per iniziare a ricevere eventi di registro da utenti con più account, il destinatario dei dati di registro crea innanzitutto una destinazione Logs. CloudWatch Ogni destinazione è formata dai seguenti elementi chiave:

Nome della destinazione

Il nome della destinazione che intendi creare.

ARN di destinazione

L'Amazon Resource Name (ARN) della AWS risorsa che desideri utilizzare come destinazione del feed di abbonamento.

ARN del ruolo

Un ruolo AWS Identity and Access Management (IAM) che CloudWatch concede a Logs le autorizzazioni necessarie per inserire i dati nel flusso scelto.

Policy di accesso

Un documento della policy IAM (in formato JSON, scritto utilizzando la grammatica delle policy IAM) che controlla l'insieme degli utenti ai quali è concesso scrivere nella tua destinazione.

> **Note**
>
> Il gruppo di log e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la AWS risorsa a cui punta la destinazione può trovarsi in una regione diversa. Negli esempi delle sezioni seguenti, tutte le Region-specific risorse vengono create negli Stati Uniti orientali (Virginia settentrionale).

Argomenti

- Configurazione di una nuova sottoscrizione tra più account
- Aggiornamento di una sottoscrizione tra più account esistente

Configurazione di una nuova sottoscrizione tra più account

Segui la procedura riportata in queste sezioni per configurare una nuova sottoscrizione del log tra più account.

Argomenti

- Passaggio 1: creazione di una destinazione
- Fase 2: creazione di un ruolo IAM (solo se si utilizza un'organizzazione)
- Passaggio 3: crea una politica di filtro dell'abbonamento a livello di account
- Convalida del flusso dei log eventi
- Modifica dell'appartenenza alla destinazione in fase di runtime

Passaggio 1: creazione di una destinazione

> **Important**
>
> Tutte le fasi di questa procedura devono essere eseguite nell'account del destinatario dei dati di log.

Per questo esempio, l'account del destinatario dei dati di registro ha un ID AWS account di 9999, mentre l'ID dell'account del mittente AWS dei dati di registro è 1111.

Questo esempio crea una destinazione utilizzando un flusso di Amazon Kinesis Data Streams chiamato RecipientStream e un ruolo che consente a CloudWatch Logs di scrivere dati su di esso.

Quando viene creata la destinazione, CloudWatch Logs invia un messaggio di prova alla destinazione per conto dell'account del destinatario. Quando il filtro di sottoscrizione è attivo in un secondo momento, CloudWatch Logs invia gli eventi di registro alla destinazione per conto dell'account di origine.

Creazione di una destinazione

1. Nell'account del destinatario, crea uno stream di destinazione in Amazon Kinesis Data Streams. Al prompt dei comandi, digita:

   ```
   aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
   ```

2. Attendi finché il flusso non diventa attivo. Puoi utilizzare il comando aws kinesis describe-stream per verificare la proprietà. StreamDescription.StreamStatus Inoltre, prendi nota del StreamDescription.StreamARN valore perché lo passerai a Logs in un secondo momento: CloudWatch

   ```
   aws kinesis describe-stream --stream-name "RecipientStream"
   {
     "StreamDescription": {
       "StreamStatus": "ACTIVE",
       "StreamName": "RecipientStream",
       "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
       "Shards": [
         {
           "ShardId": "shardId-000000000000",
           "HashKeyRange": {
   ```

```
          "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702l8EXAMPLE"
        }
      }
    ]
  }
}
```

Potrebbero essere necessari uno o due minuti perché il flusso sia in stato attivo.

3. Crea il ruolo IAM che CloudWatch concede a Logs il permesso di inserire dati nel tuo stream. Innanzitutto, devi creare una politica di fiducia in un file TrustPolicyForCWL.json ~/. Utilizza un editor di testo per creare questo file di policy, non utilizzare la console IAM.

   Questa policy include una chiave di contesto della condizione globale `aws:SourceArn` che specifica il `sourceAccountId` per prevenire il problema di sicurezza noto come "confused deputy". Se non conosci ancora l'ID dell'account di origine nella prima chiamata, consigliamo di inserire l'ARN di destinazione nel campo ARN di origine. Nelle chiamate successive, è necessario impostare l'ARN di origine come l'ARN di origine effettivo raccolto dalla prima chiamata. Per ulteriori informazioni, consulta Prevenzione del "confused deputy".

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. Utilizza il comando aws iam create-role per creare il ruolo IAM, specificando il file della policy di attendibilità. Prendi nota del Role.Arn valore restituito perché verrà passato anche a CloudWatch Logs in un secondo momento:

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Crea una politica di autorizzazioni per definire quali azioni può eseguire CloudWatch Logs sul tuo account. Innanzitutto, utilizza un editor di testo per creare una politica di autorizzazioni in un file ~/: PermissionsForCWL.json

```
{
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Associa la policy delle autorizzazioni al ruolo utilizzando il comando aws iam put-role-policy:

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Dopo che lo stream è attivo e hai creato il ruolo IAM, puoi creare la destinazione CloudWatch Logs.

   a. In questa fase alla tua destinazione non si associa una policy d'accesso predefinita e costituisce solo la prima delle due fasi necessarie per completare la creazione della destinazione. Prendi nota di DestinationArn ciò che viene restituito nel payload:

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. Dopo aver completato la fase 7, nell'account del destinatario dei dati di log associa alla destinazione una policy d'accesso predefinita. Questa policy deve specificare i log: PutSubscriptionFilter action e concede all'account mittente l'autorizzazione ad accedere alla destinazione.

      La politica concede l'autorizzazione all' AWS account che invia i log. Puoi specificare solo questo account nella policy oppure, se l'account del mittente è membro di un'organizzazione, la policy può specificare l'ID dell'organizzazione. In questo modo, puoi

creare una sola policy per consentire a più account di un'organizzazione di inviare log a questo account di destinazione.

Utilizza un editor di testo per creare un file denominato `~/AccessPolicy.json` con una delle seguenti istruzioni di policy.

Questa prima policy di esempio consente a tutti gli account dell'organizzazione che hanno un ID di `o-1234567890` di inviare log all'account del destinatario.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-1234567890"
                    ]
                }
            }
        }
    ]
}
```

Nell'esempio seguente, solo l'account del mittente dei dati di log (111111111111) può inviare log all'account del destinatario dei dati di log.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111111111111"
            },
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination"
        }
    ]
}
```

c. Collega la policy creata nel passaggio precedente alla destinazione.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Questa politica di accesso consente agli utenti dell' AWS Account con ID 1111 di chiamare la destinazione con ARN arn:aws:logs PutSubscriptionFilter ::44449999:Destination:TestDestination. *region* Qualsiasi tentativo di chiamata da parte di un altro utente verso questa destinazione verrà rifiutato. PutSubscriptionFilter

Per convalidare i privilegi di un utente su una policy d'accesso predefinita, consulta Utilizzo dello strumento di validazione delle policy nella guida per l'utente IAM.

Al termine, se utilizzi le autorizzazioni AWS Organizations per più account, segui i passaggi indicati. Fase 2: creazione di un ruolo IAM (solo se si utilizza un'organizzazione) Se le autorizzazioni vengono concesse direttamente all'altro account anziché utilizzare Organizations, puoi saltare tale passaggio e procedere alla sezione Passaggio 3: crea una politica di filtro dell'abbonamento a livello di account.

Fase 2: creazione di un ruolo IAM (solo se si utilizza un'organizzazione)

Se nella sezione precedente hai creato la destinazione utilizzando una policy di accesso che concede le autorizzazioni all'organizzazione in cui è presente l'account 111111111111, invece di concederle direttamente all'account 111111111111, segui i passaggi descritti di seguito. In caso contrario, passa alla sezione Passaggio 3: crea una politica di filtro dell'abbonamento a livello di account.

I passaggi di questa sezione creano un ruolo IAM, che CloudWatch può presupporre e convalidare se l'account mittente dispone dell'autorizzazione per creare un filtro di abbonamento in base alla destinazione del destinatario.

Per l'account mittente, segui la procedura descritta in questa sezione. Il ruolo deve esistere nell'account mittente e devi specificare l'ARN di questo ruolo nel filtro di sottoscrizione. In questo esempio, l'account mittente è denominato 111111111111.

Per creare il ruolo IAM necessario per le sottoscrizioni ai log tra account, utilizzare AWS Organizations

1. Crea la policy di attendibilità seguente in un file `/TrustPolicyForCWLSubscriptionFilter.json`. Utilizza un editor di testo per creare questo file di policy; non utilizzare la console IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Crea il ruolo IAM che utilizza questa policy. Prendi nota del valore `Arn` restituito dal comando, sarà necessario in seguito in questa procedura. In questo esempio, il ruolo in fase di creazione è denominato `CWLtoSubscriptionFilterRole`.

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account.

a. In primo luogo, utilizza un editor di testo per creare la policy di autorizzazione seguente in un file denominato ~/PermissionsForCWLSubscriptionFilter.json.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Inserisci il comando seguente per associare la policy di autorizzazione appena creata al ruolo creato nella fase 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Al termine dell'operazione, passa alla sezione Passaggio 3: crea una politica di filtro dell'abbonamento a livello di account.

Passaggio 3: crea una politica di filtro dell'abbonamento a livello di account

Una volta creata una destinazione, l'account del destinatario dei dati di log può condividere l'ARN di destinazione (arn:aws:logs:us-east-1:999999999999:destination:testDestination) con altri account AWS , perché questi possano inviare eventi di log alla stessa destinazione. Tali utenti di questi account di invio possono creare un filtro di sottoscrizione sui rispettivi gruppi di log sulla destinazione. Il filtro di sottoscrizione avvia immediatamente il flusso di dati di log in tempo reale dal gruppo di log selezionato alla destinazione specificata.

> **Note**
>
> Se stai concedendo le autorizzazioni per il filtro di sottoscrizione a un'intera organizzazione, dovrai utilizzare l'ARN del ruolo IAM che hai creato in Fase 2: creazione di un ruolo IAM (solo se si utilizza un'organizzazione).

Nell'esempio seguente, in un account di invio viene creato un criterio di filtro delle sottoscrizioni a livello di account. Il filtro è associato all'account mittente in 111111111111 modo che ogni evento di registro che corrisponde al filtro e ai criteri di selezione venga recapitato alla destinazione creata in precedenza. Tale destinazione incapsula uno stream chiamato "». RecipientStream

Il `selection-criteria` campo è facoltativo, ma è importante per escludere i gruppi di log che possono causare una ricorsione infinita dei log da un filtro di sottoscrizione. Per ulteriori informazioni su questo problema e sulla determinazione dei gruppi di log da escludere, vedere. Prevenzione della ricorsione dei log Attualmente, NOT IN è l'unico operatore supportato per`selection-criteria`.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

I gruppi di log dell'account mittente e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la destinazione può puntare a una AWS risorsa come un flusso di Amazon Kinesis Data Streams che si trova in una regione diversa.

Convalida del flusso dei log eventi

Dopo aver creato la politica di filtro delle sottoscrizioni a livello di account, CloudWatch Logs inoltra tutti gli eventi di log in entrata che corrispondono al modello di filtro e ai criteri di selezione allo stream incapsulato nello stream di destinazione denominato "». RecipientStream Il proprietario della destinazione può verificare che ciò stia accadendo utilizzando il comando aws kinesis get-shard-iterator per acquisire uno shard Amazon Kinesis Data Streams e utilizzando il comando aws kinesis get-records per recuperare alcuni record di Amazon Kinesis Data Streams:

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Potrebbe essere necessario eseguire nuovamente il comando alcune volte prima che Amazon Kinesis Data Streams inizi a restituire i dati. `get-records`

Dovresti vedere una risposta con una serie di record di Amazon Kinesis Data Streams. L'attributo di dati nel record Amazon Kinesis Data Streams viene compresso in formato gzip e quindi codificato in base64. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando i seguenti comandi Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

I dati con codifica base64 e decompressi sono in formato JSON con la seguente struttura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
```

```
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Gli elementi chiave della struttura dei dati sono i seguenti:

messageType

I messaggi di dati utilizzeranno il tipo "DATA_MESSAGE". A volte CloudWatch i log possono emettere record di Amazon Kinesis Data Streams di tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

owner

L'ID dell' AWS account dei dati di registro di origine.

logGroup

Nome del gruppo di log dei dati di log originari.

logStream

Nome del flusso di log dei dati di log originari.

subscriptionFilters

Elenco dei nomi di filtro sottoscrizione che corrispondono con i dati di log originari.

logEvents

I dati di log effettivi, rappresentati come una varietà di record di eventi di log. La proprietà "id" è un identificatore univoco per ogni log eventi.

Livello di politica

Il livello al quale la politica è stata applicata. «ACCOUNT_LEVEL_POLICY» è la politica di filtro degli abbonamenti a livello `policyLevel` di account.

Modifica dell'appartenenza alla destinazione in fase di runtime

Potrebbero verificarsi situazioni in cui potresti dover aggiungere o rimuovere l'adesione di alcuni utenti da una destinazione da te posseduta. Puoi utilizzare il commando `put-destination-policy` sulla destinazione con una nuova policy di accesso. Nell'esempio seguente, si impedisce a un account aggiunto in precedenza 111111111111 di inviare ulteriori dati di log, mentre l'account 222222222222 viene abilitato.

1. Recupera la policy attualmente associata alla destinazione TestDestination e prendi nota di: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
```

```
}
```

2. Aggiorna la policy per riflettere l'arresto di tale account 111111111111, mentre l'account 222222222222 viene abilitato. Inserisci questa politica nel file ~/: NewAccessPolicy.json

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "222222222222"
            },
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination"
        }
    ]
}
```

3. Chiama PutDestinationPolicy per associare la policy definita nel NewAccessPolicy.json file alla destinazione:

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

   Alla fine ciò disabiliterà gli eventi di log dall'ID account 111111111111. I log eventi provenienti dall'ID account 222222222222 iniziano a fluire verso la destinazione non appena il proprietario dell'account 222222222222 crea un filtro di sottoscrizione.

Aggiornamento di una sottoscrizione tra più account esistente

Se hai una sottoscrizione del log tra più account in cui l'account di destinazione concede le autorizzazioni solo a specifici account del mittente e desideri aggiornare questa sottoscrizione in modo che l'account di destinazione conceda l'accesso a tutti gli account di un'organizzazione, segui la procedura descritta in questa sezione.

Argomenti

- Fase 1: aggiornamento dei filtri di sottoscrizione
- Fase 2: aggiornamento della policy di accesso alla destinazione esistente

Fase 1: aggiornamento dei filtri di sottoscrizione

> **Note**
>
> Questo passaggio è necessario solo per le sottoscrizioni tra più account per i log creati dai servizi elencati in Abilita la registrazione da AWS services. Se non stai lavorando con log creati da uno di questi gruppi di log, puoi passare alla sezione Fase 2: aggiornamento della policy di accesso alla destinazione esistente.

In alcuni casi, devi aggiornare i filtri di sottoscrizione in tutti gli account del mittente che inviano log all'account di destinazione. L'aggiornamento aggiunge un ruolo IAM, che CloudWatch può presupporre e convalidare che l'account mittente disponga dell'autorizzazione per inviare i log all'account del destinatario.

Segui la procedura descritta in questa sezione per ogni account del mittente che desideri aggiornare in modo da utilizzare l'ID dell'organizzazione per le autorizzazioni di sottoscrizione tra più account.

Negli esempi di questa sezione sono già stati creati dei filtri di sottoscrizione negli account 111111111111 e 222222222222 per l'invio di log all'account 999999999999. I valori del filtro di sottoscrizione esistenti sono i seguenti:

```
## Existing Subscription Filter parameter values
{
    "DestinationArn": "arn:aws:logs:region:999999999999:destination:testDestination",
    "FilterPattern": "{$.userIdentity.type = Root}",
    "Distribution": "Random"
}
```

Se è necessario trovare i valori dei parametri del filtro di sottoscrizione correnti, digita il comando seguente.

```
aws logs describe-account-policies \
--policy-type "SUBSCRIPTION_FILTER_POLICY" \
--policy-name "CrossAccountStreamsExamplePolicy"
```

Aggiornamento di un filtro di sottoscrizione per iniziare a utilizzare gli ID dell'organizzazione per le autorizzazioni del log tra più account

1. Crea la policy di attendibilità seguente in un file `~/TrustPolicyForCWL.json`. Utilizza un editor di testo per creare questo file di policy; non utilizzare la console IAM.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. Crea il ruolo IAM che utilizza questa policy. Prendi nota del valore `Arn` del valore `Arn` restituito dal comando, sarà necessario in seguito in questa procedura. In questo esempio, il ruolo in fase di creazione è denominato `CWLtoSubscriptionFilterRole`.

   ```
   aws iam create-role
       \ --role-name CWLtoSubscriptionFilterRole
       \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
   ```

3. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account.

   a. In primo luogo, utilizza un editor di testo per creare la policy di autorizzazione seguente in un file denominato `/PermissionsForCWLSubscriptionFilter.json`.

      ```
      {
          "Statement": [
              {
                  "Effect": "Allow",
                  "Action": "logs:PutLogEvents",
      ```

```
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Inserisci il comando seguente per associare la policy di autorizzazione appena creata al ruolo creato nella fase 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

4. Inserisci il seguente comando per aggiornare la politica del filtro degli abbonamenti.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "{$.userIdentity.type = Root}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Fase 2: aggiornamento della policy di accesso alla destinazione esistente

Dopo aver aggiornato i filtri di sottoscrizione in tutti gli account del mittente, è possibile aggiornare la policy di accesso alla destinazione nell'account del destinatario.

Negli esempi seguenti, l'account del destinatario è `999999999999` e la destinazione è denominata `testDestination`.

L'aggiornamento abilita tutti gli account che fanno parte dell'organizzazione con ID `o-1234567890` per l'invio di log all'account del destinatario. Solo gli account con filtri di sottoscrizione creati invieranno effettivamente log all'account del destinatario.

Aggiornamento della policy di accesso alla destinazione nell'account del destinatario per iniziare a utilizzare un ID dell'organizzazione per le autorizzazioni

1. Nell'account del destinatario, utilizza un editor di testo per creare un file `~/AccessPolicy.json` con i seguenti contenuti.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-1234567890"
                    ]
                }
            }
        }
    ]
}
```

2. Digita il seguente comando per collegare la policy appena creata alla destinazione esistente. Per aggiornare una destinazione in modo da utilizzare una politica di accesso con un ID dell'organizzazione anziché una politica di accesso che elenca ID AWS account specifici, includi il `force` parametro.

> **Warning**
>
> Se stai utilizzando i log inviati da un AWS servizio elencato in[Abilita la registrazione da AWS services](), prima di eseguire questo passaggio, devi prima aver aggiornato i filtri di

abbonamento in tutti gli account mittente, come spiegato in[Fase 1: aggiornamento dei filtri di sottoscrizione](#).

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Cross-account Abbonamenti a livello di account interregionale che utilizzano Firehose

Per condividere dati di log tra diversi account, è necessario stabilire un mittente e un ricevitore di dati di log:

- Mittente dei dati di registro: riceve le informazioni sulla destinazione dal destinatario e comunica a CloudWatch Logs che è pronto a inviare gli eventi di registro alla destinazione specificata. Nelle procedure descritte nel resto di questa sezione, il mittente dei dati di registro viene mostrato con un numero di conto fittizio pari AWS a 1111.
- Destinatario dei dati di log: imposta una destinazione che incapsula un flusso di Amazon Kinesis Data Streams e comunica a Logs che il destinatario desidera ricevere i dati di CloudWatch log. Il destinatario quindi condivide le informazioni su questa destinazione con il mittente. Nelle procedure descritte nel resto di questa sezione, al destinatario dei dati di log viene mostrato un numero di conto fittizio AWS pari a 2222.

L'esempio in questa sezione utilizza un flusso di distribuzione Firehose con storage Amazon S3. Puoi anche configurare i flussi di consegna Firehose con impostazioni diverse. Per ulteriori informazioni, consulta [Creazione di un flusso di consegna Firehose.](#)

> **Note**
>
> Il gruppo di log e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la AWS risorsa a cui punta la destinazione può trovarsi in una regione diversa.

> **Note**
>
> È supportato il filtro di abbonamento Firehose per lo stesso account e lo stesso flusso di consegna tra regioni.

Argomenti

- Fase 1: Creare un flusso di consegna Firehose
- Fase 2: creazione di una destinazione
- Passaggio 3: creare una politica di filtro dell'abbonamento a livello di account
- Convalida del flusso dei log eventi
- Modifica dell'appartenenza alla destinazione durante il runtime

Fase 1: Creare un flusso di consegna Firehose

> **Important**
>
> Prima di completare i passaggi seguenti, devi utilizzare una policy di accesso, in modo che Firehose possa accedere al tuo bucket Amazon S3. Per ulteriori informazioni, consulta Controlling Access nella Amazon Data Firehose Developer Guide.
> Tutti i passaggi in questa sezione (Fase 1) devono essere eseguiti nell'account del destinatario dei dati di log.
> La regione Stati Uniti orientali (Virginia settentrionale) viene utilizzata nei comandi di esempio. Sostituiscila con la Regione corretta per l'implementazione.

Per creare un flusso di consegna Firehose da utilizzare come destinazione

1. Crea un bucket Amazon S3:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-1
```

2. Crea il ruolo IAM che concede a Firehose il permesso di inserire dati nel bucket.

   a. In primo luogo, utilizza un editor di testo per creare una policy di attendibilità in un file `~/TrustPolicyForFirehose.json`.

```
{ "Statement": { "Effect": "Allow", "Principal": { "Service":
 "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
 { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
```

b. Crea il ruolo IAM, specificando il file della policy di attendibilità appena creato.

```
aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json
```

c. L'output di questo comando risulterà simile al seguente: Annotare i Nome ruolo e ARN ruolo.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. Crea una politica di autorizzazioni per definire le azioni che Firehose può eseguire nel tuo account.

a. In primo luogo, utilizza un editor di testo per creare la policy di autorizzazione seguente in un file denominato `~/PermissionsForFirehose.json`. A seconda del caso d'uso, potresti dover aggiungere altre autorizzazioni a questo file.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
    }]
}
```

b. Inserisci il comando seguente per associare la policy di autorizzazioni appena creata con il ruolo IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Inserisci il seguente comando per creare il flusso di consegna Firehose. Sostituisci *my-role-arn* e *amzn-s3-demo-bucket2-arn* con i valori corretti per la tua distribuzione.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::amzn-s3-demo-bucket"}'
```

L'output visualizzato dovrebbe essere simile al seguente:

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Fase 2: creazione di una destinazione

> **Important**
>
> Tutte le fasi di questa procedura devono essere eseguite nell'account del destinatario dei dati di log.

Quando viene creata la destinazione, CloudWatch Logs invia un messaggio di prova alla destinazione per conto dell'account del destinatario. Quando il filtro di sottoscrizione è attivo in un secondo momento, CloudWatch Logs invia gli eventi di registro alla destinazione per conto dell'account di origine.

Creazione di una destinazione

1. Attendi che lo stream Firehose in cui hai creato Fase 1: Creare un flusso di consegna Firehose diventi attivo. È possibile utilizzare il comando seguente per verificare la StreamDescription.StreamStatus proprietà.

   ```
   aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
   ```

   Inoltre, prendi nota del DeliveryStreamDescription.DeliveryStreamARN valore, perché dovrai utilizzarlo in un passaggio successivo. Esempio di output di questo comando:

   ```
   {
       "DeliveryStreamDescription": {
           "DeliveryStreamName": "my-delivery-stream",
           "DeliveryStreamARN": "arn:aws:firehose:us-
   east-1:222222222222:deliverystream/my-delivery-stream",
           "DeliveryStreamStatus": "ACTIVE",
           "DeliveryStreamEncryptionConfiguration": {
               "Status": "DISABLED"
           },
           "DeliveryStreamType": "DirectPut",
           "VersionId": "1",
           "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
           "Destinations": [
               {
                   "DestinationId": "destinationId-000000000001",
                   "S3DestinationDescription": {
   ```

```
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::amzn-s3-demo-bucket",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

Potrebbero essere necessari uno o due minuti che il flusso di consegna venga visualizzato nello stato attivo.

2. Quando il flusso di distribuzione è attivo, crea il ruolo IAM che concederà a CloudWatch Logs il permesso di inserire dati nel tuo stream Firehose. Innanzitutto, devi creare una politica di fiducia in un file TrustPolicyForCWL.json ~/. Utilizza un editor di testo per creare questa policy. Per

ulteriori informazioni sugli endpoint CloudWatch Logs, consulta Amazon CloudWatch Logs endpoint and quotas.

Questa policy include una chiave di contesto della condizione globale `aws:SourceArn` che specifica il `sourceAccountId` per prevenire il problema di sicurezza noto come "confused deputy". Se non conosci ancora l'ID dell'account di origine nella prima chiamata, consigliamo di inserire l'ARN di destinazione nel campo ARN di origine. Nelle chiamate successive, è necessario impostare l'ARN di origine come l'ARN di origine effettivo raccolto dalla prima chiamata. Per ulteriori informazioni, consulta Prevenzione del "confused deputy".

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
     }
}
```

3. Utilizza il comando aws iam create-role per creare il ruolo IAM, specificando il file della policy di attendibilità appena creato.

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

Di seguito è riportato un output di esempio. Prendi nota del valore `Role.Arn` restituito, perché dovrai usarlo in una fase successiva.

```
{
    "Role": {
        "Path": "/",
```

```
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2023-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "ArnLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Crea una politica di autorizzazioni per definire le azioni che CloudWatch Logs può eseguire sul tuo account. Innanzitutto, utilizza un editor di testo per creare una politica di autorizzazioni in un file ~/: PermissionsForCWL.json

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. Associa la policy di autorizzazioni al ruolo inserendo il comando seguente:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Dopo che il flusso di distribuzione di Firehose è attivo e hai creato il ruolo IAM, puoi creare la CloudWatch destinazione Logs.

   a. In questa fase alla tua destinazione non verrà associata una policy d'accesso predefinita e costituisce solo la prima delle due fasi necessarie per completare la creazione della destinazione. Annota dell'ARN della nuova destinazione restituito nel payload, perché lo utilizzerai come `destination.arn` in una fase successiva.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

   b. Dopo aver completato la fase precedente, nell'account del destinatario dei dati di log (222222222222) associa alla destinazione una policy d'accesso. Questa policy consente all'account mittente dei dati di log (111111111111) di accedere alla destinazione all'interno dell'account destinatario dei dati di log (222222222222). Puoi usare un editor di testo per inserire questa policy nel `~/AccessPolicy.json` file:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
```

```
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. In questo modo viene creata una policy che definisce chi ha accesso in scrittura alla destinazione. Questa politica deve specificare le `logs:PutAccountPolicy` azioni `logs:PutSubscriptionFilter` e le azioni per accedere alla destinazione. Cross-account gli utenti utilizzeranno `PutAccountPolicy` le azioni `PutSubscriptionFilter` e per inviare gli eventi di registro alla destinazione.

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Passaggio 3: creare una politica di filtro dell'abbonamento a livello di account

Passare all'account di invio, che in questo esempio è 111111111111. Ora creerai la politica di filtro delle sottoscrizioni a livello di account nell'account di invio. In questo esempio, il filtro fa sì che ogni evento di registro contenente la stringa `ERROR` in tutti i gruppi di log tranne due venga recapitato alla destinazione creata in precedenza.

```
aws logs put-account-policy \
    --policy-name "CrossAccountFirehoseExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"DestinationArn":"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination", "FilterPattern":
 "{$.userIdentity.type = AssumedRole}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

I gruppi di log dell'account mittente e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la destinazione può puntare a una AWS risorsa come uno stream Firehose che si trova in una regione diversa.

Convalida del flusso dei log eventi

Dopo aver creato il filtro di iscrizione, CloudWatch Logs inoltra tutti gli eventi di log in entrata che corrispondono al modello di filtro e ai criteri di selezione al flusso di consegna Firehose. I dati iniziano a comparire nel bucket Amazon S3 in base all'intervallo di tempo del buffer impostato nel flusso di consegna Firehose. Quando è trascorso tempo sufficiente, puoi verificare i dati controllando il bucket Amazon S3. Per controllare il bucket, inserisci il comando seguente:

```
aws s3api list-objects --bucket 'amzn-s3-demo-bucket'
```

L'output di questo comando sarà simile al seguente:

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2023-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

È quindi possibile recuperare un oggetto specifico dal bucket inserendo il seguente comando. Sostituisci il valore di `key` con il valore trovato nel comando precedente.

```
aws s3api get-object --bucket 'amzn-s3-demo-bucket' --key '2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

I dati nell'oggetto di Amazon S3 vengono compressi nel formato gzip. Puoi esaminare i dati non elaborati dalla riga di comando utilizzando uno dei seguenti comandi:

Linux:

```
zcat testfile.gz
```

macOS:

```
zcat <testfile.gz
```

Modifica dell'appartenenza alla destinazione durante il runtime

Potrebbero verificarsi situazioni in cui devi aggiungere o rimuovere mittenti di log da una destinazione da te posseduta. Puoi utilizzare le `PutAccountPolicy` azioni PutDestinationPolicy e sulla tua destinazione con la nuova politica di accesso. Nell'esempio seguente, si impedisce a un account aggiunto in precedenza 111111111111 di inviare ulteriori dati di log, mentre l'account 333333333333 viene abilitato.

1. Recupera la policy attualmente associata alla destinazione TestDestination e prendi nota di: AccessPolicy

   ```
   aws logs describe-destinations \
       --destination-name-prefix "testFirehoseDestination"
   ```

   I dati restituiti potrebbero avere questo aspetto.

   ```
   {
       "destinations": [
           {
               "destinationName": "testFirehoseDestination",
               "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
   my-delivery-stream",
               "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
               "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
   \" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
    \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
   \" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
   east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
               "arn": "arn:aws:logs:us-east-1:
    222222222222:destination:testFirehoseDestination",
               "creationTime": 1612256124430
           }
       ]
   }
   ```

2. Aggiorna la policy per riflettere che l'account 111111111111 è stoppato, mentre l'account 333333333333 viene abilitato. Inserisci questa policy nel NewAccessPolicy.json file ~/:

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. Utilizzate il comando seguente per associare la policy definita nel NewAccessPolicy.json file alla destinazione:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

Questo alla fine disabilita il log eventi dall'ID account 111111111111. I log eventi provenienti dall'ID account 333333333333 iniziano a fluire verso la destinazione non appena il proprietario dell'account 333333333333 crea un filtro di sottoscrizione.

Prevenzione del "confused deputy"

Il problema confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità maggiormente privilegiata a eseguire l'azione. Nel AWS, l'impersonificazione tra servizi può causare il confuso problema dei vice. Cross-service l'imitazione può avvenire quando un servizio (il servizio chiamante) chiama un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per

evitare che ciò accada, AWS fornisce strumenti che consentono di proteggere i dati per tutti i servizi ai quali è stato concesso l'accesso alle risorse del proprio account.

Ti consigliamo di utilizzare le chiavi di contesto della condizione globale `aws:SourceOrgPaths` `aws:SourceArn` `aws:SourceAccount` `aws:SourceOrgID`, e le chiavi di contesto della condizione globale nelle politiche delle risorse per limitare le autorizzazioni che forniscono un altro servizio alla risorsa. Utilizza `aws:SourceArn` per associare una sola risorsa all'accesso tra servizi. Utilizza `aws:SourceAccount` se desideri consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi. Utilizza `aws:SourceOrgID` se desideri consentire l'associazione di qualsiasi risorsa di qualsiasi account interno a un'organizzazione all'uso tra servizi. Utilizza `aws:SourceOrgPaths` per associare qualsiasi risorsa dagli account in un percorso AWS Organizations all'uso tra servizi. Per ulteriori informazioni sull'utilizzo e la comprensione dei percorsi, consulta Understand the AWS Organizations entity path.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio un ARN di un bucket Amazon S3, è necessario utilizzare sia `aws:SourceAccount` che `aws:SourceArn` per limitare le autorizzazioni.

Per proteggersi dal problema "confused deputy" su larga scala, nelle policy basate sulle risorse utilizza la chiave di contesto della condizione globale `aws:SourceOrgID` o `aws:SourceOrgPaths` con l'ID dell'organizzazione o il percorso dell'organizzazione della risorsa. Quando aggiungi, rimuovi o sposti degli account all'interno dell'organizzazione, le policy che includono la chiave `aws:SourceOrgID` o `aws:SourceOrgPaths` includono automaticamente anche gli account corretti e non necessitano dell'aggiornamento manuale.

Le politiche documentate per la concessione dell'accesso ai CloudWatch log in cui scrivere dati su Amazon Kinesis Data Streams e Firehose Passaggio 1: creazione di una destinazione e Fase 2: creazione di una destinazione mostrano come è possibile utilizzare la chiave di contesto `aws:SourceArn` globale delle condizioni per aiutare a prevenire il confuso problema dei sostituti.

Prevenzione della ricorsione dei log

Esiste il rischio di provocare una ricorsione infinita dei log con i filtri di abbonamento, il che può comportare un notevole aumento della fatturazione di inserimento sia nei CloudWatch log che nella destinazione, se non impedito. Ciò può verificarsi quando un filtro di abbonamento è associato a un gruppo di log che riceve eventi di registro come risultato del flusso di lavoro di consegna dell'abbonamento. I log inseriti nel gruppo di log verranno recapitati alla destinazione, facendo sì che il gruppo di log acquisisca altri log che verranno quindi inoltrati nuovamente alla destinazione, creando un ciclo di ricorsione.

Ad esempio, considera un filtro di abbonamento con destinazione Firehose, che invia gli eventi di log ad Amazon S3. Inoltre, esiste anche una funzione Lambda che elabora i nuovi eventi inviati ad Amazon S3 e produce autonomamente alcuni log. Se il filtro di sottoscrizione viene applicato al gruppo di log della funzione Lambda, gli eventi di log prodotti dalla funzione verranno inoltrati a Firehose e Amazon S3 a destinazione, che richiameranno nuovamente la funzione, generando altri log e inoltrandoli a Firehose e Amazon S3, provocando un'altra chiamata della funzione e così via. Ciò avverrà in un ciclo infinito, portando a un aumento inaspettato della fatturazione per l'ingestione di log, Firehose e Amazon S3.

Se la funzione Lambda è collegata a un VPC con i log di flusso abilitati per i log, anche il gruppo di CloudWatch log del VPC può causare una ricorsione dei log.

Ti consigliamo di non applicare filtri di abbonamento ai gruppi di log che fanno parte del flusso di lavoro di erogazione degli abbonamenti. Per i filtri di abbonamento a livello di account, utilizza il `selectionCriteria` parametro nell'`PutAccountPolicy`API per escludere questi gruppi di log dalla policy.

Quando escludi i gruppi di log, considera i seguenti AWS servizi che producono log e possono far parte dei flussi di lavoro di distribuzione degli abbonamenti:

- Amazon EC2 con Fargate
- Lambda
- AWS Funzioni Step
- Log di flusso di Amazon VPC abilitati per i CloudWatch log

> **Note**
>
> Gli eventi di log prodotti dal gruppo di log di una destinazione Lambda non verranno inoltrati alla funzione Lambda per una politica di filtro degli abbonamenti a livello di account. In questo caso, l'esclusione dell'utilizzo `selectionCriteria` del gruppo di log della funzione Lambda di destinazione non è richiesta per le politiche di sottoscrizione dell'account.

Sintassi del modello di filtro per filtri di parametri, filtri di sottoscrizione, filtri di log eventi e Live Tail

> **Note**
>
> Per informazioni su come interrogare i tuoi gruppi di log con il linguaggio di query Amazon CloudWatch Logs Insights, consultaCloudWatch Sintassi delle interrogazioni in linguaggio Logs Insights.

Con CloudWatch Logs, puoi utilizzare filtri metrici per trasformare i dati di registro in metriche utilizzabili, filtri di abbonamento per indirizzare gli eventi di registro ad altri AWS servizi, filtrare gli eventi di registro per cercare eventi di registro e Live Tail per visualizzare in modo interattivo i log in tempo reale man mano che vengono inseriti.

I modelli di filtro costituiscono la sintassi utilizzata dai filtri metrici, dai filtri di sottoscrizione, dagli eventi di registro e da Live Tail per abbinare i termini negli eventi di registro. I termini possono essere parole, frasi esatte o valori numerici. Le espressioni regolari (regex) possono essere utilizzate per creare modelli di filtro autonomi o incorporate con modelli di filtri JSON delimitati da spazi.

Crea modelli di filtro con i termini che desideri associare. I modelli di filtro restituiscono solo i log eventi che contengono i termini definiti. Puoi testare i modelli di filtro nella CloudWatch console.

Argomenti

- Sintassi delle espressioni regolari (regex) supportate
- Utilizzo dei modelli di filtro per verificare la presenza dei termini in un'espressione regolare (regex)
- Utilizzo dei modelli di filtro per associare termini nei log eventi non strutturati
- Utilizzo dei modelli di filtro per associare termini nei log eventi
- Utilizzo dei modelli di filtro per associare termini nei log eventi delimitati da spazi

Sintassi delle espressioni regolari (regex) supportate

Sintassi regex supportata

Quando si utilizzano espressioni regolari per cercare e filtrare i dati di un log, è necessario racchiudere le espressioni tra `%`.

I modelli di filtro con regex possono includere solo quanto segue:

- Caratteri alfanumerici: un carattere alfanumerico è un carattere costituito da una lettera (dalla A alla Z o dalla a alla z) o una cifra (da 0 a 9).
- Caratteri simbolici supportati: ", `:` ", `_` ", # ", = ", `@` ", `/` ", `;` ", `,` " e '`-`'. Ad esempio, `%something!%` verrebbe rifiutato poiché '`!`' non è supportato.
- Operatori supportati: questi includono: '^', '$', '?', '[', ']', '{', '}', '|', '\', '*', '+' e '`.`'.

Gli operatori (e) non sono supportati. Non è possibile utilizzare le parentesi per definire un sottomodello.

Multi-byte i caratteri non sono supportati.

> **Note**
>
> Quote
> Quando si creano filtri di parametri o filtri di sottoscrizione, per ogni gruppo di log sono disponibili al massimo 5 modelli di filtro contenenti espressioni regolari.
> Quando si crea un modello di filtro delimitato o JSON per filtri di parametri e filtri di sottoscrizione o quando si filtrano log eventi, è possibile utilizzare al massimo due espressioni regolari per ogni modello di filtro.

Utilizzo degli operatori supportati

- ^: fissa la corrispondenza all'inizio di una stringa. Ad esempio, `%^[hc]at%` restituisce "hat" e "cat", ma solo all'inizio di una stringa.
- $: fissa la corrispondenza all'inizio di una stringa. Ad esempio, `%[hc]at$%` corrisponde a "hat" e "cat", ma solo alla fine di una stringa.
- ?: corrisponde a zero o a un'occorrenza del termine precedente. Ad esempio, `%colou?r%` può restituire entrambe le varianti "color" e "colour".

- `[]`: definisce una classe di caratteri. Corrisponde all'elenco o all'intervallo di caratteri racchiuso tra parentesi quadre. Ad esempio, `%[abc]%` restituisce ad "a", "b" o "c", `%[a-z]%` restituisce qualsiasi lettera minuscola da "a" a "z" e `%[abcx-z]%` restituisce ad "a", "b", "c", "x", "y" oppure "z".
- `{m, n}`: restituisce il termine precedente almeno m e non più di n volte. Ad esempio, `%a{3,5}%` restituisce solo "aaa", "aaaa" e "aaaaa".

> **Note**
>
> È possibile omettere m o n se si sceglie di non definire un minimo o un massimo.

- `|`: "or" booleano, che restituisce il termine a sinistra o a destra della barra verticale. Ad esempio:
  - `%gray|grey%`può corrispondere a «grigio» o «grigio»
  - `%^starting|^initializing|^shutting down%`può corrispondere a «start... «, o «inizializzazione... «, o «spegnimento», ma non corrisponderà a «saltare l'inizializzazione...»
  - `%abcc|ab[^c]$%`può corrispondere a «abcc...» e «aba», ma non corrisponderà a «aac...»
- `\`: carattere di escape, che consente di utilizzare il significato letterale di un operatore anziché il suo significato speciale. Ad esempio, `%\[.\]%` restituisce qualsiasi carattere singolo racchiuso tra "[" e "]" poiché le parentesi quadre sono ignorate, come nel caso di "[a]", "[b]", "[7]", "[@]", "[]]" e "[ ]".

> **Note**
>
> `%10\.10\.0\.1%` è il modo corretto per creare un'espressione regolare che restituisca l'indirizzo IP 10.10.0.1.

- `*`: corrisponde a zero o più istanze del termine precedente. Ad esempio, `%ab*c%` può restituire "ac", "abc" e "abbbc"; `%ab[0-9]*%` può restituire "ab", "ab0" e "ab129".
- `+`: corrisponde a una o più istanze del termine precedente. Ad esempio, `%ab+c%` può restituire "abc", "abbc" e "abbbc", ma non "ac".
- `.`: corrisponde a qualsiasi carattere singolo. Ad esempio, `%.at%` restituisce qualsiasi stringa di tre caratteri che termina con "at", tra cui "hat", "cat", "bat", "4at", "#at" e " at" (che inizia con uno spazio).

> **Note**
>
> Quando si crea un'espressione regolare per verificare la presenza di indirizzi IP, è importante evitare l'operatore `.`. Ad esempio, `%10.10.0.1%` può restituire "10010,051", che potrebbe non essere lo scopo effettivo previsto dell'espressione.

- `\d`,`\D`: Corrisponde a un carattere a una digit/non cifra. Ad esempio, `%\d%` è equivalente a `%[0-9]%` e `%\D%` è equivalente a `%[^0-9]%`.

> **Note**
>
> L'operatore maiuscolo indica l'inverso della sua controparte minuscola.

- `\s`,`\S`: Corrisponde a un carattere whitespace -whitespace character/non.

> **Note**
>
> L'operatore maiuscolo indica l'inverso della sua controparte minuscola. I caratteri di spaziatura includono i caratteri tab (`\t`), spazio ( ) e newline (`\n`).

- `\w`,`\W`: Corrisponde a un carattere alfanumerico -alfanumerico. character/non Ad esempio, `%\w%` è equivalente a `%[a-zA-Z_0-9]%` e `%\W%` è equivalente a `%[^a-zA-Z_0-9]%`.

> **Note**
>
> L'operatore maiuscolo indica l'inverso della sua controparte minuscola.

- `\xhh`: corrisponde alla mappatura ASCII per un carattere esadecimale a due cifre. `\x`è la sequenza di escape che indica che i seguenti caratteri rappresentano il valore esadecimale per ASCII. hhspecifica le due cifre esadecimali (0-9 e A-F) che puntano a un carattere nella tabella ASCII.

> **Note**
>
> È possibile utilizzare `\xhh` per restituire caratteri simbolici che non sono supportati dal modello di filtro. Ad esempio, `%\x3A%` corrisponde a `:` e `%\x28%` corrisponde a `(`.

Utilizzo dei modelli di filtro per verificare la presenza dei termini in un'espressione regolare (regex)

Corrispondenza dei termini usando regex

È possibile verificare la presenza di termini nei log eventi utilizzando un modello di espressioni regolari racchiuso tra % (segni percentuali prima e dopo il modello di espressione regolare). Il seguente frammento di codice mostra un esempio di un modello di filtro che restituisce tutti i log eventi che comprendono la parola chiave AUTHORIZED.

Per un elenco delle espressioni regolari supportate, consulta la sezione Espressioni regolari supportate.

```
%AUTHORIZED%
```

Questo modello di filtro restituisce messaggi di log eventi come i seguenti:

- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[SUCCESS 200] AUTHORIZED REQUEST`

Utilizzo dei modelli di filtro per associare termini nei log eventi non strutturati

Corrispondenza dei termini in log eventi non strutturati

Gli esempi seguenti contengono frammenti di codice che mostrano come è possibile utilizzare i modelli di filtro per associare i termini nei log eventi non strutturati.

> **Note**
>
> I modelli di filtro fanno distinzione tra maiuscole e minuscole. Racchiudi frasi e termini esatti che includono caratteri non alfanumerici tra virgolette doppie ("").

Example: Match a single term

Il seguente frammento di codice mostra un esempio del modello di filtro di un termine singolo che restituisce tutti i log eventi in cui i messaggi contengono la parola ERROR.

```
ERROR
```

Questo modello di filtro corrisponde ai messaggi di log eventi come i seguenti:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match multiple terms

Il seguente frammento di codice mostra un esempio del modello di filtro di più termini che restituisce tutti i log eventi in cui i messaggi contengono le parole ERROR e ARGUMENTS.

```
ERROR ARGUMENTS
```

Il filtro restituisce i messaggi del log eventi, come i seguenti:

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Questo modello di filtro non restituisce i seguenti messaggi del log eventi perché non contengono entrambi i termini specificati nel modello di filtro.

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Example: Match optional terms

È possibile utilizzare la corrispondenza di modelli per creare modelli di filtro che restituiscono log eventi contenenti termini facoltativi. Inserisci un punto interrogativo ("?") prima dei termini che desideri associare. Il seguente frammento di codice mostra un esempio di un modello di filtro che restituisce tutti i log eventi in cui i messaggi contengono la parola ERRORE o ARGOMENTI.

```
?ERROR ?ARGUMENTS
```

Questo modello di filtro corrisponde ai messaggi di log eventi come i seguenti:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

> **Note**
>
> Non è possibile combinare il punto interrogativo («?») con altri modelli di filtro, ad esempio termini di inclusione ed esclusione. Se combini «?» con altri modelli di filtro, tutti i termini con punto interrogativo verranno ignorati.
> Ad esempio, il seguente schema di filtro corrisponde a tutti gli eventi che contengono la parolaREQUEST, tranne il punto interrogativo («?») i termini di filtro vengono ignorati e non hanno alcun effetto.
>
> ```
> ?ERROR ?ARGUMENTS REQUEST
> ```
>
> Corrispondenze di log eventi
>
> - `[INFO] REQUEST FAILED`
> - `[WARN] UNAUTHORIZED REQUEST`
> - `[ERROR] 400 BAD REQUEST`

Example: Match exact phrases

Il seguente frammento di codice mostra un esempio di un modello di filtro che restituisce log eventi in cui i messaggi contengono la frase esatta INTERNAL SERVER ERROR.

```
"INTERNAL SERVER ERROR"
```

Questo modello di filtro restituisce il seguente messaggio di log eventi:

- `[ERROR 500] INTERNAL SERVER ERROR`

Example: Include and exclude terms

Puoi creare modelli di filtro che restituiscono log eventi in cui i messaggi includono alcuni termini ed escludono altri termini. Posiziona un simbolo meno ("-") prima dei termini che desideri escludere. Il seguente frammento di codice mostra un esempio di un modello di filtro che restituisce log eventi in cui i messaggi contengono il termine ERROR ed escludono il termine ARGUMENTS.

```
ERROR -ARGUMENTS
```

Questo modello di filtro restituisce messaggi di log eventi come i seguenti:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Questo modello di filtro non restituisce i seguenti messaggi di log eventi perché contengono la parola ARGUMENTS.

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match everything

Puoi utilizzare la corrispondenza totale nei log eventi usando le virgolette doppie. Il seguente frammento di codice mostra un esempio di un modello di filtro che restituisce tutti i log eventi.

```
" "
```

Utilizzo dei modelli di filtro per associare termini nei log eventi

Scrittura di modelli di filtro per log eventi JSON

Negli esempi seguenti viene descritto l'utilizzo della sintassi per i filtri di modelli che verificano la presenza di termini JSON contenenti stringhe e valori numerici.

Writing filter patterns that match strings

È possibile creare modelli di filtro per associare le stringhe nei log eventi JSON. Il seguente frammento di codice mostra un esempio della sintassi per il modello di filtro basato su stringa.

```
{ PropertySelector EqualityOperator String }
```

Racchiudi i modelli di filtro tra parentesi graffe («{}»). String-based i modelli di filtro devono contenere le seguenti parti:

- Property selector (Selettore di proprietà)

  Imposta i selettori di proprietà con il simbolo del dollaro seguito da un punto ("$."). I selettori di proprietà sono stringhe alfanumeriche che supportano anche i caratteri trattino alto ("-") e basso ("_"). Le stringhe non supportano la notazione scientifica. I selettori di proprietà puntano ai nodi di valore nei log eventi JSON. I nodi di valore possono essere stringhe o numeri. Posiziona le matrici dopo i selettori di proprietà. Gli elementi negli array seguono un sistema di numerazione a base zero, il che significa che il primo elemento dell'array è l'elemento 0, il secondo elemento è l'elemento 1 e così via. Racchiudi gli elementi tra parentesi quadre ("[]"). Se un selettore di proprietà punta a una matrice o a un oggetto, il modello di filtro non corrisponde al formato

di log. Se la proprietà JSON contiene un punto (''.''), è possibile utilizzare la notazione a parentesi per selezionarla.

> **Note**
>
> Selettore di carattere jolly
> È possibile utilizzare il carattere jolly JSON per selezionare qualsiasi elemento dell'array o qualsiasi campo oggetto JSON.
> Quote
> È possibile utilizzare un solo selettore di caratteri jolly in un selettore di proprietà.

- Equality operator (Operatori di uguaglianza)

  Imposta gli operatori di uguaglianza con uno dei seguenti simboli: uguale ("=") o non uguale ("!=") . Gli operatori di uguaglianza restituiscono un valore booleano (vero o falso).

- String

  Puoi racchiudere le stringhe tra virgolette doppie (""). Le stringhe che non contengono caratteri alfanumerici e il simbolo del trattino basso devono essere collocate tra virgolette doppie. Usa l'asterisco ("*") come carattere jolly per associare il testo.

> **Note**
>
> È possibile utilizzare qualsiasi espressione regolare condizionale durante la creazione di modelli di filtro che verifichino la presenza di termini nei log eventi JSON. Per un elenco delle espressioni regolari supportate, consulta la sezione Espressioni regolari supportate.

Il seguente frammento di codice contiene un esempio di un modello di filtro che mostra come è possibile formattarne uno in modo che corrisponda a un termine JSON con una stringa.

```
{ $.eventType = "UpdateTrail" }
```

Writing filter patterns that match numeric values

È possibile creare modelli di filtro per associare valori numerici nei log eventi JSON. Il seguente frammento di codice mostra un esempio della sintassi per i modelli di filtro che corrispondono a valori numerici.

```
{ PropertySelector NumericOperator Number }
```

Racchiudi i modelli di filtro tra parentesi graffe ("{}"). I modelli di filtro che corrispondono a valori numerici devono avere le seguenti parti:

- Property selector (Selettore di proprietà)

  Imposta i selettori di proprietà con il simbolo del dollaro seguito da un punto ("$."). I selettori di proprietà sono stringhe alfanumeriche che supportano anche i caratteri trattino alto ("-") e basso ("_"). Le stringhe non supportano la notazione scientifica. I selettori di proprietà puntano ai nodi di valore nei log eventi JSON. I nodi di valore possono essere stringhe o numeri. Posiziona le matrici dopo i selettori di proprietà. Gli elementi negli array seguono un sistema di numerazione a base zero, il che significa che il primo elemento dell'array è l'elemento 0, il secondo elemento è l'elemento 1 e così via. Racchiudi gli elementi tra parentesi quadre ("[]"). Se un selettore di proprietà punta a una matrice o a un oggetto, il modello di filtro non corrisponde al formato di log. Se la proprietà JSON contiene un punto (" **.** "), è possibile utilizzare la notazione a parentesi per selezionarla.

  > **Note**
  >
  > Selettore di carattere jolly
  > È possibile utilizzare il carattere jolly JSON per selezionare qualsiasi elemento dell'array o qualsiasi campo oggetto JSON.
  > Quote
  > È possibile utilizzare un solo selettore di caratteri jolly in un selettore di proprietà.

- Operazioni numeriche

  Imposta le operazioni numeriche con uno dei seguenti simboli: maggiore di (">"), minore di ("<"), uguale ("="), non uguale ("!="), maggiore o uguale a (">=") oppure minore o uguale a ("<=").

- Numero

  Puoi utilizzare numeri interi che contengono i simboli più ("+") o meno ("-") e seguire la notazione scientifica. Usa l'asterisco ("*") come carattere jolly per associare i numeri.

Il seguente frammento di codice contiene esempi che mostrano come è possibile formattare i modelli di filtro in modo che corrispondano a termini JSON con valori numerici.

```
// Filter pattern with greater than symbol
{ $.bandwidth > 75 }
// Filter pattern with less than symbol
{ $.latency < 50 }
// Filter pattern with greater than or equal to symbol
{ $.refreshRate >= 60 }
// Filter pattern with less than or equal to symbol
{ $.responseTime <= 5 }
// Filter pattern with equal sign
{ $.errorCode = 400}
// Filter pattern with not equal sign
{ $.errorCode != 500 }
// Filter pattern with scientific notation and plus symbol
{ $.number[0] = 1e+3 }
// Filter pattern with scientific notation and minus symbol
{ $.number[0] != 1e-3 }
```

Corrispondenza dei termini in log eventi JSON utilizzando espressioni semplici

Gli esempi seguenti contengono frammenti di codice che mostrano come i modelli di filtro possono corrispondere ai termini in un log eventi JSON.

> **Note**
>
> Se testi il modello di filtro di esempio con il log eventi JSON di esempio, devi inserire il log JSON di esempio su una singola riga.

Log eventi JSON

```
{
      "eventType": "UpdateTrail",
```

```
      "sourceIPAddress": "111.111.111.111",
      "arrayKey": [
            "value",
            "another value"
      ],
      "objectList": [
           {
             "name": "a",
             "id": 1
           },
           {
             "name": "b",
             "id": 2
           }
      ],
      "SomeObject": null,
      "cluster.name": "c"
}
```

Example: Filter pattern that matches string values

Questo modello di filtro restituisce la stringa "UpdateTrail" nella proprietà "eventType".

```
{ $.eventType = "UpdateTrail" }
```

Example: Filter pattern that matches string values (IP address)

Questo modello di filtro contiene un carattere jolly e corrisponde alla proprietà "sourceIPAddress" perché non contiene un numero con il prefisso "123.123.".

```
{ $.sourceIPAddress != 123.123.* }
```

Example: Filter pattern that matches a specific array element with a string value

Questo modello di filtro restituisce l'elemento "value" nell'array "arrayKey".

```
{ $.arrayKey[0] = "value" }
```

Example: Filter pattern that matches a string using regex

Questo modello di filtro restituisce la stringa `"Trail"` nella proprietà `"eventType"`.

```
{ $.eventType = %Trail% }
```

Example: Filter pattern that uses a wildcard to match values of any element in the array using regex

Questo modello di filtro contiene espressioni regolari che restituiscono l'elemento `"value"` dell'array `"arrayKey"`.

```
{ $.arrayKey[*] = %val.{2}% }
```

Example: Filter pattern that uses a wildcard to match values of any element with a specific prefix and subnet using regex (IP address)

Questo modello di filtro contiene espressioni regolari che restituiscono l'elemento `"111.111.111.111"` della proprietà `"sourceIPAddress"`.

```
{ $.* = %111\.111\.111\.1[0-9]{1,2}% }
```

> **Note**
>
> Quote
> È possibile utilizzare un solo selettore di caratteri jolly in un selettore di proprietà.

Example: Filter pattern that matches a JSON property with a period (.) in the key

È possibile creare modelli di filtro che corrispondano ai nomi di proprietà JSON che contengono un punto (`.`) utilizzando la notazione tra parentesi. Racchiudere il nome della proprietà tra

virgolette singole (). ' Le virgolette doppie (") non sono supportate all'interno della notazione tra parentesi e causano il fallimento del pattern di filtro con un errore di sintassi. Il seguente schema di filtro corrisponde alla proprietà. `cluster.name`

```
{ $.['cluster.name'] = "c" }
```

Example: Filter pattern that matches JSON logs using IS

È possibile creare modelli di filtro che restituiscono le corrispondenze con i campi dei log JSON con la variabile `IS`. La variabile `IS` può corrispondere a campi che contengono i valori `NULL`, `TRUE` oppure `FALSE`. Il seguente modello di filtro restituisce i log JSON in cui il valore di `SomeObject` è `NULL`.

```
{ $.SomeObject IS NULL }
```

Example: Filter pattern that matches JSON logs using NOT EXISTS

È possibile creare modelli di filtro con la `NOT EXISTS` variabile per restituire log JSON che non contengono campi specifici nei dati di registro. Il seguente modello di filtro utilizza `NOT EXISTS` per restituire i log JSON che non contengono il campo `SomeOtherObject`.

```
{ $.SomeOtherObject NOT EXISTS }
```

> **Note**
>
> Le variabili `IS NOT` e `EXISTS` al momento non sono supportate.

Corrispondenza di termini negli oggetti JSON utilizzando espressioni composte

È possibile utilizzare gli operatori logici AND ("&&") e OR ("||") nei modelli di filtro per creare espressioni composte che corrispondono ai log eventi in cui due o più condizioni sono vere. Le

espressioni composte supportano l'uso di parentesi ("()") e il seguente ordine standard di operazioni: () > && > ||. Gli esempi seguenti contengono frammenti di codice che mostrano come utilizzare i modelli di filtro con le espressioni composte in modo da associare i termini in un oggetto JSON.

Oggetto JSON

```
{
    "user": {
        "id": 1,
        "email": "John.Stiles@example.com"
    },
    "users": [
        {
         "id": 2,
         "email": "John.Doe@example.com"
        },
        {
         "id": 3,
         "email": "Jane.Doe@example.com"
        }
    ],
    "actions": [
        "GET",
        "PUT",
        "DELETE"
    ],
    "coordinates": [
        [0, 1, 2],
        [4, 5, 6],
        [7, 8, 9]
    ]
}
```

Example: Expression that matches using AND (&&)

Questo modello di filtro contiene un'espressione composta che corrisponde a `"id"` in `"user"` con un valore numerico di 1 e `"email"` nel primo elemento dell'array `"users"` con la stringa `"John.Doe@example.com"`.

```
{ ($.user.id = 1) && ($.users[0].email = "John.Doe@example.com") }
```

Example: Expression that matches using OR (||)

Questo modello di filtro contiene un'espressione composta che corrisponde a `"email"` in `"user"` con la stringa `"John.Stiles@example.com"`.

```
{ $.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch" &&
 $.actions[2] = "nonmatch" }
```

Example: Expression that doesn't match using AND (&&)

Questo modello di filtro contiene un'espressione composta che non trova una corrispondenza perché l'espressione non corrisponde alla terza operazione in `"actions"`.

```
{ ($.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch") &&
 $.actions[2] = "nonmatch" }
```

> **Note**
>
> Quote
> È possibile utilizzare un solo selettore di caratteri jolly in un selettore di proprietà e fino a tre selettori di caratteri jolly in un modello di filtro con espressioni composte.

Example: Expression that doesn't match using OR (||)

Questo modello di filtro contiene un'espressione composta che non trova una corrispondenza perché l'espressione non corrisponde alla prima proprietà in `"users"` o alla terza operazione in `"actions"`.

```
{ ($.user.id = 2 && $.users[0].email = "nonmatch") || $.actions[2] = "GET" }
```

Utilizzo dei modelli di filtro per associare termini nei log eventi delimitati da spazi

Scrittura di modelli di filtro per log eventi delimitati da spazi

È possibile creare modelli di filtro per verificare la presenza di termini in log eventi delimitati da spazi. Di seguito viene fornito un esempio di log eventi delimitato da spazi e viene illustrato come scrivere la sintassi per i modelli di filtro che verificano la presenza di termini in un log eventi delimitato da spazi.

> **Note**
>
> È possibile utilizzare qualsiasi espressione regolare condizionale durante la creazione di modelli di filtro che verifichino la presenza di termini nei log eventi delimitati da spazi. Per un elenco delle espressioni regolari supportate, consulta la sezione Espressioni regolari supportate.

Example: Space-delimited log event

Il seguente frammento di codice mostra un log eventi delimitato da spazi che contiene sette campi: `ip`, `user`, `username`, `timestamp`, `request`, `status_code` e `bytes`.

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

> **Note**
>
> I caratteri tra parentesi quadre ("[]") e virgolette doppie ("") sono considerati campi singoli.

Writing filter patterns that match terms in a space-delimited log event

Per creare un modello di filtro che verifichi la presenza di termini in un log eventi delimitato da spazi, racchiudi il modello di filtro tra parentesi quadre ("[]") e specifica i campi con nomi separati da virgole (","). Il seguente modello di filtro analizza sette campi.

```
[ip=%127\.0\.0\.[1-9]%, user, username, timestamp, request =*.html*, status_code =
 4*, bytes]
```

È possibile utilizzare operazioni numeriche (**>**, **<**, **=**, **!=**, **>=** oppure **<=**) e l'asterisco (*) come carattere jolly per fornire le condizioni del modello di filtro. Nel modello di filtro di esempio, `ip` utilizza espressioni regolari che restituiscono l'intervallo di indirizzi IP 127.0.0.1 - 127.0.0.9, `request` contiene un carattere jolly che indica che deve estrarre un valore con `.html` e `status_code` contiene un carattere jolly che indica che deve estrarre un valore che inizia con `4`.

Se non conosci il numero di campi che stai analizzando in un log eventi delimitato da spazi, puoi usare i puntini di sospensione (...) per fare riferimento a qualsiasi campo senza nome. Questi possono fare riferimento a tutti i campi necessari. L'esempio seguente mostra un modello di filtro con puntini di sospensione che rappresentano i primi quattro campi senza nome mostrati nel modello di filtro di esempio precedente.

```
[..., request =*.html*, status_code = 4*, bytes]
```

Puoi anche utilizzare gli operatori logici AND (&&) e OR (||) per creare espressioni composte. Il seguente modello di filtro contiene un'espressione composta che indica che il valore di `status_code` deve essere `404` o `410`.

```
[ip, user, username, timestamp, request =*.html*, status_code = 404 || status_code =
 410, bytes]
```

Corrispondenza di termini in log eventi delimitati da spazi utilizzando la corrispondenza dei modelli

È possibile utilizzare la corrispondenza di modelli per creare modelli di filtro delimitati da spazi che associano termini in un ordine specifico. Specifica l'ordine dei termini con gli indicatori. Utilizza w1 per rappresentare il primo termine e w2 e così via per rappresentare l'ordine dei termini successivi. Inserisci una virgola (",") tra i termini. Gli esempi seguenti contengono frammenti di codice che mostrano come è possibile utilizzare la corrispondenza di modelli con modelli di filtro delimitati da spazi.

> **Note**
>
> È possibile utilizzare qualsiasi espressione regolare condizionale durante la creazione di modelli di filtro che verifichino la presenza di termini nei log eventi delimitati da spazi. Per un elenco delle espressioni regolari supportate, consulta la sezione Espressioni regolari supportate.

Space-delimited evento di registro

```
INFO 09/25/2014 12:00:00 GET /service/resource/67 1200
INFO 09/25/2014 12:00:01 POST /service/resource/67/part/111 1310
WARNING 09/25/2014 12:00:02 Invalid user request
ERROR 09/25/2014 12:00:02 Failed to process request
```

Example: Match terms in order

Il seguente modello di filtro delimitato da spazi restituisce log eventi in cui la prima parola è ERROR.

```
[w1=ERROR, w2]
```

> **Note**
>
> Quando crei modelli di filtro delimitati da spazi che utilizzano la corrispondenza di modelli, devi includere un indicatore vuoto dopo aver specificato l'ordine dei termini. Ad esempio,

per creare un modello di filtro che restituisca i log eventi in cui la prima parola è ERROR, includi un indicatore vuoto w2 dopo il termine w1.

Example: Match terms with AND (&&) and OR (||)

È possibile utilizzare gli operatori logici AND ("&&") e OR ("||") per creare modelli di filtro delimitati da spazi che contengono condizioni. Il seguente modello di filtro restituisce log eventi in cui la prima parola è ERROR o WARNING.

```
[w1=ERROR || w1=WARNING, w2]
```

Example: Exclude terms from matches

È possibile creare modelli di filtro delimitati da spazi che restituiscono log eventi escludendo uno o più termini. Inserisci un simbolo non uguale ("!=") prima del termine o dei termini che desideri escludere. Il seguente frammento di codice mostra un esempio di un modello di filtro che restituisce log eventi in cui le prime parole non sono ERROR e WARNING.

```
[w1!=ERROR && w1!=WARNING, w2]
```

Example: Match the top level item in a resource URI

Il seguente frammento di codice mostra un esempio di un modello di filtro che restituisce l'elemento di livello superiore in un URI di risorsa che utilizza regex.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+$%, response_time]
```

Example: Match the child level item in a resource URI

Il seguente frammento di codice mostra un esempio di un modello di filtro che corrisponde all'elemento di livello inferiore in un URI di risorsa che utilizza regex.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+/part/[0-9]+$%,
 response_time]
```

Abilita la registrazione da AWS services

Utilizza il catalogo ricercabile per trovare un AWS servizio e apri la relativa guida alla configurazione della registrazione. Per ulteriori informazioni sulle destinazioni e sui modelli di autorizzazione per ciascun servizio, consulta Destinazioni e autorizzazioni di registro supportate.

Molti servizi pubblicano i log solo su CloudWatch Logs, ma altri utilizzano il vended log delivery per inviare i log direttamente ad Amazon Simple Storage Service o Amazon Data Firehose. La consegna diretta è utile quando il requisito principale è l'archiviazione o l'elaborazione a lungo termine in una di queste destinazioni.

Anche quando pubblichi i log direttamente su Amazon S3 o CloudWatch Firehose, vengono applicate le spese di spedizione. Se invii log ad Amazon S3, gli addebiti *AWS_REGION*`-S3-Egress-Bytes` vengono visualizzati in Cost Explorer o sulla fattura. Se invii i log a Firehose, *AWS_REGION*`-FH-Egress-Bytes` vengono visualizzati gli addebiti. Per ulteriori informazioni sui prezzi dei log venduti, consulta la scheda Logs in Amazon Pricing. CloudWatch

Alcuni servizi richiedono autorizzazioni aggiuntive prima di poter fornire i log. Senza queste autorizzazioni, la consegna dei log non riesce. Nella tabella di confronto, i servizi che utilizzano il modello di autorizzazioni originale sono etichettati Supportati (autorizzazioni V1). I servizi che utilizzano il modello corrente sono etichettati Supportati (autorizzazioni V2). Ogni etichetta rimanda alle politiche richieste.
<integration-catalog>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>

<integration></integration>
<integration></integration>
<integration></integration>

<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>

<integration></integration>
<integration></integration>
</integration-catalog>

Destinazioni e autorizzazioni di registro supportate

La tabella seguente mostra le destinazioni supportate da ciascun AWS servizio per l'invio dei log e la versione delle autorizzazioni richiesta.

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Log di accesso ad Amazon API Gateway	Registri venduti	Supportato (autorizzazioni V1)			
Log di accesso di Application Load Balancer	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	
AWS AppSync logs	Registri personalizzati	Supportata			
Registri MySQL di Amazon Aurora	Registri personalizzati	Supportata			

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Amazon Bedrock Registrazione delle basi di conoscenza	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)
Amazon Bedrock Registrazione degli agenti	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Amazon Bedrock AgentCore Runtime	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)
Amazon Bedrock AgentCore Gateway	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)
Amazon Bedrock AgentCore Identità	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Amazon Bedrock AgentCore Memoria	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)
Amazon Bedrock AgentCore Pagamenti	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)
Amazon Bedrock AgentCore Strumenti	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)
Log dei parametri di qualità multimediale di Amazon Chime e log dei messaggi SIP	Registri venduti	Supportato (autorizzazioni V1)			
CloudFront: registri di accesso	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
AWS CloudHSM registri di controllo	Registri personalizzati	Supportata			
CloudWatch Evidentemente i registri degli eventi di valutazione	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)		
CloudWatch Registri di Internet Monitor	Registri venduti		Supportato (autorizzazioni V1)		
the section called "Crea una fonte di consegna per i log di esecuzione delle query di CloudWatch Logs Insights"	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
CloudTrail logs	Registri personalizzati	Supportata			
AWS CodeBuild logs	Registri personalizzati	Supportata			

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Amazon CodeWhisperer registri degli eventi	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Amazon Cognito logs	Registri venduti	Supportato (autorizzazioni V1)			
Registri clienti Amazon Connect	Registri personalizzati	Supportata			
AWS DataSync logs	Registri personalizzati	Supportata			
AWS DevOps Agent logs	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Registri di Amazon ElastiCache (Redis OSS)	Registri venduti	Supportati (autorizzazioni V1)		Supportato (autorizzazioni V1)	
AWS Elastic Beanstalk logs	Registri personalizzati	Supportata			
Registri di Amazon Elastic Container Service	Registri personalizzati	Supportata			
Registri in modalità automatica di Amazon Elastic Kubernetes Service	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri delle funzionalità di Amazon Elastic Kubernetes Service	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Log del piano di controllo di Amazon Elastic Kubernetes Service	Registri venduti	Supportata			

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
AWS Elemental MediaPackage registri di accesso	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
AWS Elemental MediaTailor logs	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
AWS Entity Resolution logs	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Amazon EventBridge Registrazione delle tubazioni	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
Amazon EventBridge bus per eventi	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Amazon EventBridge registri degli event bus (personalizzati)	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
AWS Fargate logs	Registri personalizzati	Supportata			
AWS Fault Injection Service registri degli esperimenti	Registri venduti		Supportato (autorizzazioni V1)		
Amazon FinSpace	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
AWS Global Accelerator registri di flusso	Registri venduti		Supportato (autorizzazioni V1)		

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
AWS Glue registri di lavoro	Registri personalizzati	Supportata			
registri degli errori di IAM Identity Center	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri delle chat di Amazon Interactive Video Service	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
AWS IoT logs	Registri personalizzati	Supportata			
AWS IoT FleetWise logs	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
AWS Lambda logs	Registri venduti	Supportata	Supportato	Supportata	

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Registri di Amazon Macie	Registri personalizzati	Supportata			
Modernizzazione del mainframe AWS	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
Amazon Managed Service per i log di Prometheus	Registri venduti	Supportato (autorizzazioni V1)			
Registri del broker Amazon MSK	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
Registri di Amazon MSK Connect	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Registri Amazon MQ	Registri personalizzati	Supportata			
AWS Registri del Network Firewall	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
AWS Registri proxy del Network Firewall	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri di accesso al Network Load Balancer	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
OpenSearch logs	Registri personalizzati	Supportata			

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Registri OpenSearch di ingestione di Amazon Service	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
AWS PCS logs	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri dei connettori Amazon Q Business	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri delle conversazioni di Amazon Q Business	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri eventi degli agenti Amazon Q in Connect AI	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Registri Amazon Quick	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri PostgreSQL di Amazon Relational Database Service	Registri personalizzati	Supportata			
Registri delle query di Amazon Route 53 Global Resolver	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri di query DNS pubblici di Amazon Route 53	Registri venduti	Supportata			
Registri delle query del resolver Amazon Route 53	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)		
AWS Registri RTB Fabric	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
AWS Security Hub CSPM	Registri venduti	Supportato (autorizzazioni V2)			
AWS Security Hub	Registri venduti	Supportato (autorizzazioni V2)			
AWS Registri del flusso di attacco Shield Advanced	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Log di accesso al server Amazon Simple Storage Service	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Eventi Amazon SageMaker AI	Registri venduti	Supportato (autorizzazioni V1)			

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Eventi per i lavoratori di Amazon SageMaker AI	Registri venduti	Supportato (autorizzazioni V1)			
AWS Site-to-Site Registri VPN	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
Registri di Amazon Simple Email Service	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
Registri di Amazon Simple Notification Service	Registri personalizzati	Supportata			
Registri delle politiche di protezione e dei dati di Amazon Simple Notification Service	Registri personalizzati	Supportata			

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
File di feed di dati di EC2 Spot Instance	Registri venduti		Supportato (autorizzazioni V1)		
AWS Step Functions Registri Express Workflow e Standard Workflow	Registri venduti	Supportato (autorizzazioni V1)			
Registri di controllo e registri di integrità di Storage Gateway	Registri venduti	Supportato (autorizzazioni V1)			
AWS Transfer Family logs	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
Accesso verificato da AWS logs	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	

Origine del log	Tipo di log	the section called "Registri inviati a CloudWatch Logs"	the section called "Log inviati ad Amazon S3"	the section called "Log inviati a Firehose"	the section called "Tracce inviate a X-Ray"
Registri di flusso di Amazon Virtual Private Cloud	Registri venduti	Supportata	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
Log di accesso di Amazon VPC Lattice	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	
Server di routing Amazon VPC	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	
AWS WAF logs	Registri venduti	Supportato (autorizzazioni V1)	Supportato (autorizzazioni V1)	Supportata	
Amazon WorkMail registri di controllo	Registri venduti	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	Supportato (autorizzazioni V2)	

Registrazione che richiede autorizzazioni aggiuntive [V1]

Alcuni AWS servizi utilizzano un'infrastruttura comune per inviare i propri log a CloudWatch Logs, Amazon S3 o Firehose. Per consentire ai AWS servizi elencati nella tabella precedente di inviare i propri log a queste destinazioni, è necessario accedere come utente con determinate autorizzazioni.

Inoltre, è necessario concedere le autorizzazioni AWS per consentire l'invio dei log. AWS può creare automaticamente tali autorizzazioni al momento della configurazione dei registri oppure è possibile crearle personalmente prima di configurare la registrazione. Per la distribuzione su più account, è necessario creare manualmente le politiche di autorizzazione.

Se si sceglie di impostare AWS automaticamente le autorizzazioni e le politiche relative alle risorse necessarie quando l'utente o un membro dell'organizzazione configurano per la prima volta l'invio dei log, l'utente che sta configurando l'invio dei log deve disporre di determinate autorizzazioni, come spiegato più avanti in questa sezione. In alternativa, è possibile creare autonomamente le policy delle risorse e quindi gli utenti che impostano l'invio dei log non necessitano di altrettante autorizzazioni.

I seguenti argomenti forniscono ulteriori dettagli per ciascuna di queste destinazioni.

Argomenti

- Registri inviati a CloudWatch Logs
- Log inviati ad Amazon S3
- Log inviati a Firehose

Registri inviati a CloudWatch Logs

> **Important**
>
> Quando si impostano i tipi di registro nell'elenco seguente per l'invio ai CloudWatch registri, AWS crea o modifica le politiche delle risorse associate al gruppo di log che riceve i registri, se necessario. Continua a leggere questa sezione per vedere i dettagli.

Questa sezione si applica quando i tipi di log elencati nella tabella della sezione precedente vengono inviati a Logs: CloudWatch

Autorizzazioni degli utenti

Per poter configurare l'invio di uno di questi tipi di log a CloudWatch Logs per la prima volta, è necessario accedere a un account con le seguenti autorizzazioni.

- `logs:CreateLogDelivery`
- `logs:PutResourcePolicy`
- `logs:DescribeResourcePolicies`
- `logs:DescribeLogGroups`

> **Note**
>
> Quando specificate l'`logs:PutResourcePolicy`autorizzazione`logs:DescribeLogGroups`, `ologs:DescribeResourcePolicies`, assicuratevi di impostare l'ARN della relativa `Resource` riga in modo che utilizzi un carattere * jolly, invece di specificare solo il nome di un singolo gruppo di log. Ad esempio, `"Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*"`

Se uno di questi tipi di log viene già inviato a un gruppo di log in CloudWatch Logs, per configurare l'invio di un altro di questi tipi di log allo stesso gruppo di log è sufficiente l'autorizzazione. `logs:CreateLogDelivery`

Policy delle risorse del gruppo di log

Il gruppo di log in cui vengono inviati i log deve disporre di una policy delle risorse che includa determinate autorizzazioni. Se il gruppo di log attualmente non dispone di un criterio in materia di risorse e l'utente che configura la `logs:PutResourcePolicy` registrazione dispone dei `logs:DescribeLogGroups` permessi e per il gruppo di log, crea AWS automaticamente la seguente politica quando si inizia a inviare i log a CloudWatch Logs. `logs:DescribeResourcePolicies` Per le sottoscrizioni appena create, le politiche delle risorse sono configurate a livello di gruppo di log e hanno una dimensione massima di 51.200 byte. Se una politica di risorse esistente a livello di account concede già le autorizzazioni tramite wildcard, non verrebbe creata una politica a livello di gruppo di log separata. Per verificare la politica Group-level delle risorse di registro per uno specifico gruppo di log, utilizzare il `describe-resource-policies` comando con il `--resource-arn` parametro impostato sul gruppo di log ARN e il `--policy-scope` parametro impostato su. `RESOURCE`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:my-log-group:log-
stream:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Il limite delle politiche relative alle risorse del gruppo di log è di 51.200 byte. Una volta raggiunto questo limite, AWS non può aggiungere nuove autorizzazioni. Ciò richiede che i clienti modifichino manualmente la policy per concedere al `delivery.logs.amazonaws.com` servizio le autorizzazioni principali sulle azioni `logs:CreateLogStream` e `logs:PutLogEvents`. I

clienti devono utilizzare un prefisso per il nome del gruppo di log con caratteri jolly come `/aws/vendedlogs/*` e utilizzare questo nome per le future creazioni di Delivery.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:my-log-group/aws/
vendedlogs/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Log inviati ad Amazon S3

Quando imposti i log da inviare ad Amazon S3 AWS , crea o modifica le politiche delle risorse associate al bucket S3 che riceve i log, se necessario.

I log pubblicati in Amazon S3 vengono pubblicati in un bucket esistente da te specificato. Uno o più file di log vengono creati ogni cinque minuti nel bucket specificato.

Quando invii i log per la prima volta a un bucket Amazon S3, il servizio che consegna i log registra il proprietario del bucket per assicurarsi che i log vengano consegnati solo a un bucket appartenente a questo account. Di conseguenza, per modificare il proprietario del bucket Amazon S3, devi ricreare o aggiornare la sottoscrizione del log nel servizio di origine.

Note

CloudFront utilizza un modello di autorizzazioni diverso rispetto agli altri servizi che inviano i log forniti a S3. Per ulteriori informazioni, consulta Autorizzazioni richieste per configurare la registrazione standard e per accedere ai file di log.
Inoltre, se utilizzi lo stesso bucket S3 per i log di CloudFront accesso e un'altra fonte di log, l'abilitazione dell'ACL sul bucket concede CloudFront anche l'autorizzazione a tutte le altre fonti di log che utilizzano questo bucket.

Important

Se invii log a un bucket Amazon S3 e la policy del bucket contiene `NotAction` un elemento `NotPrincipal` or, l'aggiunta automatica delle autorizzazioni di consegna dei log al bucket e la creazione di un abbonamento di log avranno esito negativo. Per creare correttamente un abbonamento di log, devi aggiungere manualmente le autorizzazioni di consegna dei log alla policy del bucket, quindi creare l'abbonamento di log. Per ulteriori informazioni, consulta le istruzioni in questa sezione.
Se il bucket dispone di una crittografia lato server che utilizza una AWS KMS chiave gestita dal cliente, è necessario aggiungere anche la policy relativa alla chiave gestita dal cliente. Per ulteriori informazioni, consulta Amazon S3.
Se il bucket di destinazione dispone SSE-KMS di una Bucket Key abilitata, la politica di chiave KMS gestita dal cliente allegata non funziona più come previsto per tutte le richieste. Per ulteriori informazioni, consulta Ridurre il costo di SSE-KMS con Amazon S3 Bucket Keys.

Se utilizzi registri venduti e crittografia S3 con una AWS KMS chiave gestita dal cliente, devi utilizzare una chiave AWS KMS ARN completamente qualificata anziché un ID chiave quando configuri il bucket. Per ulteriori informazioni, consulta put-bucket-encryption.

Autorizzazioni degli utenti

Per poter configurare l'invio di uno di questi tipi di log ad Amazon S3 per la prima volta, devi aver effettuato l'accesso a un account con le seguenti autorizzazioni.

- `logs:CreateLogDelivery`
- `S3:GetBucketPolicy`
- `S3:PutBucketPolicy`

Se uno qualsiasi di questi tipi di log è già stato inviato a un bucket Amazon S3, quindi per impostare l'invio di un altro di questi tipi di log allo stesso bucket è sufficiente disporre del autorizzazione `logs:CreateLogDelivery`.

Policy delle risorse di bucket S3

Il bucket S3 in cui vengono inviati i log deve disporre di una policy delle risorse che include determinate autorizzazioni. Se il bucket al momento non dispone di una policy delle risorse e l'utente che imposta la registrazione ha le autorizzazioni `S3:GetBucketPolicy` e `S3:PutBucketPolicy` per il bucket, AWS crea automaticamente la seguente policy quando inizi a inviare i log ad Amazon S3.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
```

```
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Nella policy precedente, per `aws:SourceAccount`, specifica l'elenco degli ID account per i quali i log vengono consegnati a questo bucket. Per `aws:SourceArn`, specifica l'elenco di ARN della risorsa che genera i log, nel formato `arn:aws:logs:`*source-region*`:`*source-account-id*`:*`.

Se il bucket dispone di una policy delle risorse, ma tale criterio non contiene l'istruzione mostrata nella policy precedente e l'utente che imposta la registrazione ha le autorizzazioni `S3:GetBucketPolicy` e `S3:PutBucketPolicy` per il bucket, tale istruzione viene aggiunta alla policy delle risorse del bucket.

> **Note**
>
> In alcuni casi, è possibile che vengano visualizzati `AccessDenied` degli errori AWS CloudTrail se l'`s3:ListBucket`autorizzazione non è stata concessa. `delivery.logs.amazonaws.com` Per evitare questi errori nei CloudTrail log, è necessario concedere l'`s3:ListBucket`autorizzazione `delivery.logs.amazonaws.com` e includere `Condition` i parametri mostrati con l'`s3:GetBucketAcl`autorizzazione impostata nella precedente policy del bucket. Per renderlo più semplice, invece di creare una nuova `Statement`, puoi aggiornar direttamente`AWSLogDeliveryAclCheck` per essere `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Crittografia lato server di bucket Amazon S3

Puoi proteggere i dati nel tuo bucket Amazon S3 abilitando la crittografia lato server con chiavi S3-managed Amazon SSE-S3 () o la crittografia lato server con AWS KMS una chiave archiviata in (). AWS Key Management Service SSE-KMS Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server.

Se lo desideri, non è richiesta alcuna configurazione SSE-S3 aggiuntiva. Amazon S3 gestisce la chiave di crittografia.

> **Warning**
>
> Se si sceglie SSE-KMS, è necessario utilizzare una chiave gestita dal cliente, poiché l'utilizzo di una chiave AWS gestita non è supportato in questo scenario. Se si configura la crittografia utilizzando una chiave AWS gestita, i log verranno consegnati in un formato illeggibile.

Quando utilizzi una AWS KMS chiave gestita dal cliente, puoi specificare l'Amazon Resource Name (ARN) della chiave gestita dal cliente quando abiliti la crittografia dei bucket. È necessario aggiungere quanto segue alla policy della chiave per la chiave gestita dal cliente (non alla policy di bucket per il bucket S3), in modo che l'account di consegna del log possa scrivere nel bucket S3.

Se lo desideri SSE-KMS, devi utilizzare una chiave gestita dal cliente, poiché l'utilizzo di una chiave AWS gestita non è supportato in questo scenario. Quando utilizzi una AWS KMS chiave gestita dal cliente, puoi specificare l'Amazon Resource Name (ARN) della chiave gestita dal cliente quando abiliti la crittografia dei bucket. È necessario aggiungere quanto segue alla policy della chiave per la chiave gestita dal cliente (non alla policy di bucket per il bucket S3), in modo che l'account di consegna del log possa scrivere nel bucket S3.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
        }
}
```

Per `aws:SourceAccount`, specifica l'elenco degli ID account per i quali i log vengono consegnati a questo bucket. Per `aws:SourceArn`, specifica l'elenco di ARN della risorsa che genera i registri, nel formato `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Log inviati a Firehose

Questa sezione si applica quando i tipi di log elencati nella tabella della sezione precedente vengono inviati a Firehose:

Autorizzazioni degli utenti

Per poter configurare l'invio di uno di questi tipi di log a Firehose per la prima volta, è necessario accedere a un account con le seguenti autorizzazioni.

- `logs:CreateLogDelivery`
- `firehose:TagDeliveryStream`
- `iam:CreateServiceLinkedRole`

Se uno di questi tipi di log viene già inviato a Firehose, per configurare l'invio di un altro di questi tipi di log a Firehose è necessario disporre solo delle autorizzazioni and. `logs:CreateLogDelivery` `firehose:TagDeliveryStream`

Ruoli IAM utilizzati per le autorizzazioni

Poiché Firehose non utilizza policy relative alle risorse, AWS utilizza i ruoli IAM per configurare questi log da inviare a Firehose. AWS crea un ruolo collegato al servizio denominato. AWSServiceRoleForLogDelivery Questo ruolo collegato al servizio include le autorizzazioni seguenti.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Questo ruolo collegato al servizio concede l'autorizzazione per tutti i flussi di distribuzione Firehose con il tag impostato su. `LogDeliveryEnabled` `true` AWS assegna questo tag al flusso di consegna di destinazione quando si configura la registrazione.

Questo ruolo collegato al servizio dispone inoltre di una policy di attendibilità che autorizzi il principale del servizio `delivery.logs.amazonaws.com` di assumere il ruolo collegato al servizio necessario. Questa policy di attendibilità è la seguente:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Registrazione che richiede autorizzazioni aggiuntive [V2]

Alcuni AWS servizi utilizzano un nuovo metodo per inviare i propri log. Si tratta di un metodo flessibile che consente di configurare la consegna dei log da questi servizi verso una o più delle seguenti destinazioni: CloudWatch Logs, Amazon S3 o X-Ray Firehose e per la consegna tramite traccia.

Una consegna di log funzionante è composta da tre elementi:

- `ADeliverySource`, che è un oggetto logico che rappresenta le risorse che effettivamente inviano i log.
- `ADeliveryDestination`, che è un oggetto logico che rappresenta la destinazione di consegna effettiva.
- `ADelivery`, che collega una fonte di consegna alla destinazione di consegna

Per configurare la consegna dei log tra un AWS servizio supportato e una destinazione, devi fare quanto segue:

- Crea una fonte di consegna con PutDeliverySource.
- Crea una destinazione di consegna con PutDeliveryDestination.
- Se distribuisci log su più account, puoi utilizzare solo destinazioni Amazon S3 e Firehose. È necessario utilizzarlo PutDeliveryDestinationPolicynell'account di destinazione per assegnare una IAM policy alla destinazione. Questa politica autorizza la creazione di una consegna dalla fonte di consegna nell'account A alla destinazione di consegna nell'account B. Per la consegna tra account, devi creare manualmente le politiche di autorizzazione. Per esempi di configurazione, consulta. Cross-account esempio di consegna
- Crea una consegna associando esattamente una fonte di consegna e una destinazione di consegna, utilizzando CreateDelivery.

Esempi di configurazione della consegna dei log

Gli esempi seguenti creano l'origine e la destinazione di consegna nello stesso AWS account. Sostituisci l'ARN della risorsa di origine e il tipo di log con i valori supportati dal servizio che genera i log. Questi esempi non richiedono una politica sulla destinazione di consegna.

Crea una fonte di consegna per i log di esecuzione delle query di CloudWatch Logs Insights

L'esempio seguente crea una fonte di consegna per i log di esecuzione delle query di CloudWatch Logs Insights. Per i requisiti ARN completi `logType` e relativi alle risorse, vedere. PutDeliverySource

> **Note**
>
> Il carattere jolly finale `log-group:*` è obbligatorio perché i log di esecuzione delle query non sono legati a un gruppo di log. AWS non supporta un gruppo di log ARN specifico per questo tipo di registro.

```
aws logs put-delivery-source \
    --name insights-query-logs \
    --resource-arn arn:aws:logs:region:account-id:log-group:* \
    --log-type INSIGHTS_QUERY_LOGS
```

Crea una fonte di consegna

Crea la fonte di consegna, quindi utilizza uno degli esempi di destinazione che seguono.

```
aws logs put-delivery-source \
    --name my-delivery-source \
    --resource-arn source-resource-arn \
    --log-type log-type
```

Crea una consegna a CloudWatch Logs

Crea una destinazione di consegna per un gruppo di log esistente.

```
aws logs put-delivery-destination \
    --name my-cwl-delivery-destination \
    --delivery-destination-configuration \
    "destinationResourceArn=arn:aws:logs:region:account-id:log-group:log-group-name"
```

Crea la consegna.

```
aws logs create-delivery \
    --delivery-source-name my-delivery-source \
    --delivery-destination-arn arn:aws:logs:region:account-id:delivery-destination:my-
cwl-delivery-destination
```

Creare una consegna su Amazon S3

Crea una destinazione di consegna per un bucket esistente.

```
aws logs put-delivery-destination \
    --name my-s3-delivery-destination \
    --delivery-destination-configuration \
    "destinationResourceArn=arn:aws:s3:::bucket-name"
```

Crea la consegna.

```
aws logs create-delivery \
    --delivery-source-name my-delivery-source \
    --delivery-destination-arn arn:aws:logs:region:account-id:delivery-destination:my-
s3-delivery-destination
```

Per configurare un prefisso di destinazione, un percorso di suffisso o un Hive-compatible percorso, vedi. Formato chiave oggetto Amazon S3

Creare una consegna a Firehose

Crea una destinazione di consegna per un flusso di DirectPut consegna esistente.

```
aws logs put-delivery-destination \
    --name my-firehose-delivery-destination \
    --delivery-destination-configuration \
    "destinationResourceArn=arn:aws:firehose:region:account-id:deliverystream/delivery-stream-name"
```

Crea la consegna.

```
aws logs create-delivery \
    --delivery-source-name my-delivery-source \
    --delivery-destination-arn arn:aws:logs:region:account-id:delivery-destination:my-firehose-delivery-destination
```

Crea una consegna a X-Ray

Per la consegna tramite traccia, utilizza un servizio di origine e un tipo di registro che supporti la consegna a X-Ray. Crea la destinazione logica X-Ray di consegna.

```
aws logs put-delivery-destination \
    --name my-xray-delivery-destination \
    --delivery-destination-type XRAY
```

Crea la consegna.

```
aws logs create-delivery \
    --delivery-source-name my-delivery-source \
    --delivery-destination-arn arn:aws:logs:region:account-id:delivery-destination:my-xray-delivery-destination
```

Per verificare la consegna, usa il seguente comando.

```
aws logs describe-deliveries \
```

```
    --delivery-source-name-prefix my-delivery-source
```

Le sezioni seguenti forniscono i dettagli delle autorizzazioni necessarie quando si effettua l'accesso per configurare la consegna dei log a ciascun tipo di destinazione utilizzando il processo V2. Queste autorizzazioni possono essere concesse a un ruolo IAM con cui hai effettuato l'accesso.

> **Important**
>
> È responsabilità dell'utente rimuovere le risorse di consegna dei log dopo aver eliminato la risorsa che genera i log. A tale scopo, segui questi passaggi.
>
> 1. Eliminare il `Delivery` utilizzando l'DeleteDeliveryoperazione.
> 2. Eliminare il `DeliverySource` utilizzando l'DeleteDeliverySourceoperazione.
> 3. Se l'`DeliveryDestination`elemento associato a `DeliverySource` quello che hai appena eliminato viene utilizzato solo per questo specifico`DeliverySource`, puoi rimuoverlo utilizzando l'DeleteDeliveryDestinationsoperazione.

Indice

- Registri inviati a CloudWatch Logs
  - Autorizzazioni degli utenti
  - Policy delle risorse del gruppo di log
- Log inviati ad Amazon S3
  - Autorizzazioni degli utenti
  - Politica delle risorse del bucket Amazon S3
  - Crittografia lato server di bucket Amazon S3
  - Formato chiave oggetto Amazon S3
- Log inviati a Firehose
  - Autorizzazioni degli utenti
  - Ruoli IAM utilizzati per le autorizzazioni delle risorse
- Tracce inviate a X-Ray
  - Autorizzazioni degli utenti
  - X-Ray politica delle risorse
  - Abilitazione di Transaction Search

Registri inviati a CloudWatch Logs

Per un esempio di AWS CLI, vedi. Crea una consegna a CloudWatch Logs

Autorizzazioni degli utenti

Per abilitare l'invio di log a CloudWatch Logs, devi aver effettuato l'accesso con le seguenti autorizzazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
            "arn:aws:logs:us-east-1:444455556666:delivery-source:*",
            "arn:aws:logs:us-east-1:777788889999:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
```

```
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyCWL",
            "Effect": "Allow",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:123456789012:*"
            ]
        }
    ]
}
```

Policy delle risorse del gruppo di log

Il gruppo di log in cui vengono inviati i log deve disporre di una policy delle risorse che includa determinate autorizzazioni. Se al momento il gruppo di log non dispone di una politica in materia di risorse e l'utente che configura la `logs:PutResourcePolicy` registrazione dispone `logs:DescribeLogGroups` delle autorizzazioni relative al gruppo di log, crea AWS automaticamente la seguente politica quando si inizia a inviare i log a Logs. `logs:DescribeResourcePolicies` CloudWatch Per le sottoscrizioni appena create, le politiche delle risorse sono configurate a livello di gruppo di log e hanno una dimensione massima di 51.200 byte. Se una politica di risorse esistente a livello di account concede già le autorizzazioni tramite wildcard, non verrebbe creata una politica a livello di gruppo di log separata. Per verificare la politica Group-level delle risorse di registro per uno specifico gruppo di log, utilizzare il `describe-resource-policies` comando con il `--resource-arn` parametro impostato sul gruppo di log ARN e il `--policy-scope` parametro impostato su. `RESOURCE`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:my-log-group:log-
stream:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Il limite delle politiche relative alle risorse del gruppo di log è di 51.200 byte. Una volta raggiunto questo limite, AWS non può aggiungere nuove autorizzazioni. Ciò richiede che i clienti modifichino manualmente la policy per concedere al `delivery.logs.amazonaws.com` servizio le autorizzazioni principali sulle azioni `logs:CreateLogStream` e `logs:PutLogEvents`. I

clienti devono utilizzare un prefisso per il nome del gruppo di log con caratteri jolly come `/aws/vendedlogs/*` e utilizzare questo nome per le future creazioni di Delivery.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite20150319",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:my-log-group/aws/
vendedlogs/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Log inviati ad Amazon S3

Per un esempio di AWS CLI, vedi. Creare una consegna su Amazon S3

Autorizzazioni degli utenti

Per abilitare l'invio di log ad Amazon S3, devi aver effettuato l'accesso con le seguenti autorizzazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-source:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
```

```
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyS3",
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        }
    ]
}
```

Politica delle risorse del bucket Amazon S3

Il bucket S3 in cui vengono inviati i log deve disporre di una policy delle risorse che include determinate autorizzazioni. Se il bucket al momento non dispone di una policy delle risorse e l'utente che imposta la registrazione ha le autorizzazioni `S3:GetBucketPolicy` e `S3:PutBucketPolicy` per il bucket, AWS crea automaticamente la seguente policy quando inizi a inviare i log ad Amazon S3.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/AWSLogs/account-ID/*",
            "Condition": {
```

```
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:delivery-source:*"
                    ]
                }
            }
        }
    ]
}
```

Nella policy precedente, per `aws:SourceAccount`, specifica l'elenco degli ID account per i quali i log vengono consegnati a questo bucket. Per `aws:SourceArn`, specifica l'elenco di ARN della risorsa che genera i log, nel formato `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Se il bucket dispone di una policy delle risorse, ma tale criterio non contiene l'istruzione mostrata nella policy precedente e l'utente che imposta la registrazione ha le autorizzazioni `S3:GetBucketPolicy` e `S3:PutBucketPolicy` per il bucket, tale istruzione viene aggiunta alla policy delle risorse del bucket.

> **Note**
>
> In alcuni casi, potresti riscontrare `AccessDenied` degli errori AWS CloudTrail se l'`s3:ListBucket`autorizzazione non è stata concessa. `delivery.logs.amazonaws.com` Per evitare questi errori nei CloudTrail log, è necessario concedere l'`s3:ListBucket`autorizzazione `delivery.logs.amazonaws.com` e includere `Condition` i parametri mostrati con l'`s3:GetBucketAcl`autorizzazione impostata nella precedente policy del bucket. Per renderlo più semplice, invece di creare una nuova `Statement`, puoi aggiornar direttamenteAWSLogDeliveryAclCheck per essere `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Crittografia lato server di bucket Amazon S3

Puoi proteggere i dati nel tuo bucket Amazon S3 abilitando la crittografia lato server. Puoi usare Amazon S3-managed keys (SSE-S3) o una AWS KMS chiave memorizzata in AWS Key Management Service (SSE-KMS). Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server.

Se lo desideri SSE-S3, non è richiesta alcuna configurazione aggiuntiva. Amazon S3 gestisce la chiave di crittografia.

> **⚠ È richiesta una chiave gestita dal cliente**
>
> Se lo desideri SSE-KMS, devi utilizzare una chiave gestita dal cliente. Non puoi usare una chiave AWS gestita. Se configuri la crittografia con una chiave AWS gestita, CloudWatch Logs fornisce i log in un formato illeggibile.

Per SSE-KMS, specifica l'Amazon Resource Name (ARN) della chiave quando abiliti la crittografia dei bucket. Aggiungi quanto segue alla politica chiave (non alla politica del bucket per il tuo bucket S3), in modo che l'account di consegna dei log possa scrivere sul tuo bucket S3.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["012345678901"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:012345678901:delivery-source:*"]
```

```
        }
        }
}
```

Per`aws:SourceAccount`, specifica gli ID degli account i cui log vengono consegnati a questo bucket. Per`aws:SourceArn`, specifica gli ARN della fonte di consegna nel seguente formato:. `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:delivery-source:*`

Formato chiave oggetto Amazon S3

Per le consegne che utilizzano le autorizzazioni V2, la chiave oggetto Amazon S3 è determinata dal prefisso di destinazione, dal tipo di log, dal percorso del suffisso della consegna e dall'eventuale abilitazione dei percorsi. Hive-compatible Il percorso esatto definito dal servizio e le variabili di suffisso supportate variano in base al tipo di registro.

Prefisso di destinazione

Un percorso opzionale che aggiungi al bucket ARN quando chiami. PutDeliveryDestination Ad esempio, `arn:aws:s3:::`*`bucket-name`*`/`*`MyLogPrefix`*. Gli oggetti consegnati iniziano con questo prefisso, seguito dal percorso definito dal servizio del tipo di registro, ad esempio. `AWSLogs/`*`source-account-id`*`/`*`service-name`*`/` Per la registrazione CloudFront standard (v2), il prefisso di destinazione sostituisce, anziché precede, il percorso predefinito.

Percorso del suffisso

Un percorso opzionale che configuri per una singola consegna nel suo S3DeliveryConfiguration. Un suffisso può contenere testo e variabili statici. Per trovare le variabili supportate da un tipo di registro, chiamate DescribeConfigurationTemplatese controllate`allowedSuffixPathFields`. Se non specifichi un percorso di suffisso, il percorso del suffisso predefinito del tipo di registro viene utilizzato quando disponibile.

Hive-compatible percorso

Quando `enableHiveCompatiblePath` è così`true`, le variabili supportate in modo esplicito `suffixPath` vengono renderizzate come. *`key`*`=`*`value`* Per tipi di log come i log di accesso di Application Load Balancer, questa formattazione modifica anche il segmento di account definito dal servizio. Ad esempio `AWSLogs/`*`source-account-id`*`/` diventa `AWSLogs/aws-account-id=`*`source-account-id`*`/`.

Se si omette`suffixPath`, la Hive-compatible formattazione viene applicata solo quando il percorso del suffisso predefinito del tipo di registro la supporta. Ad esempio, i log di accesso di

Application Load Balancer supportano questa formattazione. Altri tipi di log, inclusi i log di accesso ai server Amazon S3, mantengono il suffisso predefinito originale in formato diverso da Hive, incluse le directory di risorse e date.

Per un tipo di registro il cui valore predefinito non supporta la Hive-compatible formattazione, specifica un file valido che contenga variabili supportate. `suffixPath` Un suffisso esplicito sostituisce il suffisso predefinito. Includi le variabili di risorsa e data di cui hai bisogno, utilizzando il tipo di registro from. `allowedSuffixPathFields` DescribeConfigurationTemplates

Gli esempi seguenti mostrano l'inizio di una chiave oggetto del log di accesso di Application Load Balancer per l'account in. `111122223333` `us-east-1` Se non diversamente specificato, gli esempi non presuppongono alcun prefisso di destinazione.

Configurazione	Inizio della chiave dell'oggetto
Hive-compatible percorso disabilitato, suffisso omesso	`AWSLogs/111122223333/elasticloadbalancing/us-east-1/2026/09/10/`
Hive-compatible percorso abilitato, suffisso omesso	`AWSLogs/aws-account-id=111122223333/elasticloadbalancing/region=us-east-1/year=2026/month=09/day=10/`
Hive-compatible percorso abilitato, suffisso `myFolder/{yyyy}/{MM}/{dd}`	`AWSLogs/aws-account-id=111122223333/elasticloadbalancing/myFolder/year=2026/month=09/day=10/`
Hive-compatible percorso abilitato, suffisso `myFolder/`	`AWSLogs/111122223333/elasticloadbalancing/myFolder/`
prefisso di destinazioneMyLogPrefix , Hive-compatible percorso disabilitato, suffisso omesso	`MyLogPrefix/AWSLogs/111122223333/elasticloadbalancing/us-east-1/2026/09/10/`

Percorsi dei suffissi statici e autorizzazioni dei bucket

Per tipi di log come i log di accesso di Application Load Balancer, un registro `suffixPath` che contiene solo testo statico non abilita la Hive-compatible formattazione del prefisso dell'account definito dal servizio. Ad esempio, `myFolder/` utilizza il semplice prefisso dell'account mostrato nella tabella. La consegna fallisce se CloudWatch Logs tenta di scrivere con il semplice prefisso dell'account e la policy del bucket consente solo il prefisso. Hive-compatible

Quando si crea una distribuzione di Application Load Balancer con `enableHiveCompatiblePath` set to`true`, la bucket policy creata automaticamente concede `s3:PutObject` il prefisso. Hive-compatible Includi almeno una variabile supportata dal tipo di registro `allowedSuffixPathFields` in `suffixPath` o omettila `suffixPath` per un tipo di registro il cui percorso del suffisso predefinito supporta la formattazione. Hive-compatible

CloudFront standard logging (v2) formatta il suo prefisso predefinito indipendentemente dalle variabili in. `suffixPath`

Note

CloudFront documenta il comportamento del percorso di registrazione standard (v2) e gli esempi in Send logs to Amazon S3.

Note

L'aggiornamento del percorso o dell' Hive-compatible impostazione del suffisso influisce sulle consegne successive dopo la propagazione della configurazione. Gli oggetti esistenti non vengono spostati. Non è possibile modificare l'ARN di una destinazione, incluso il prefisso, mentre le consegne attive fanno riferimento a tale destinazione.

La policy del bucket deve consentire `s3:PutObject` il prefisso risultante. Quando gestisci la policy del bucket, mantieni attive le `aws:SourceArn` condizioni `aws:SourceAccount` e mostrate Politica delle risorse del bucket Amazon S3 nella policy del bucket di Amazon S3 e concedi l'accesso solo al prefisso richiesto.

Log inviati a Firehose

Per un esempio di AWS CLI, vedi. Creare una consegna a Firehose

Autorizzazioni degli utenti

Per abilitare l'invio di log a Firehose, è necessario accedere con le seguenti autorizzazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-source:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
```

```
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyFH",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/*"
            ]
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        }
    ]
}
```

Ruoli IAM utilizzati per le autorizzazioni delle risorse

Poiché Firehose non utilizza policy relative alle risorse, AWS utilizza i ruoli IAM per configurare questi log da inviare a Firehose. AWS crea un ruolo collegato al servizio denominato. AWSServiceRoleForLogDelivery Questo ruolo collegato al servizio include le autorizzazioni seguenti.

JSON

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

```
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Questo ruolo collegato al servizio concede l'autorizzazione per tutti i flussi di distribuzione Firehose con il tag impostato su. `LogDeliveryEnabled` `true` AWS assegna questo tag al flusso di consegna di destinazione quando si configura la registrazione.

Questo ruolo collegato al servizio dispone inoltre di una policy di attendibilità che autorizzi il principale del servizio `delivery.logs.amazonaws.com` di assumere il ruolo collegato al servizio necessario. Questa policy di attendibilità è la seguente:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Tracce inviate a X-Ray

Per un esempio di AWS CLI, vedi. Crea una consegna a X-Ray

Autorizzazioni degli utenti

Per abilitare l'invio di tracce a AWS X-Ray, devi aver effettuato l'accesso con le seguenti autorizzazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
              "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
            "arn:aws:logs:us-east-1:111122223333:delivery-source:*",
            "arn:aws:logs:us-east-1:111122223333:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
```

```
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyXRay",
            "Effect": "Allow",
            "Action": [
                "xray:PutResourcePolicy",
                "xray:ListResourcePolicies",
                "xray:GetTraceSegmentDestination"
            ],
            "Resource": "*"
        }
    ]
}
```

X-Ray politica delle risorse

L'account di destinazione a cui vengono inviate le tracce deve disporre di una politica delle risorse che includa determinate autorizzazioni. Quando l'utente che configura il tracciamento dispone `xray:PutResourcePolicy` `xray:ListResourcePolicies` delle autorizzazioni necessarie per l'account, crea AWS automaticamente la politica delle risorse quando inizi a inviare le tracce a. X-Ray La politica creata dipende dal servizio di origine:

Amazon Bedrock AgentCore resources

AWS crea una politica delle risorse per tipo di risorsa. La policy utilizza modelli di caratteri jolly relativi ai confini dell'account e coprono tutte le risorse dello stesso tipo di Amazon Bedrock AgentCore risorsa presenti nell'account. Ad esempio, se una risorsa di Amazon Bedrock AgentCorememoria è abilitata per l'invio di tracce, la policy copre tutte le risorse di memoria di quell'account, incluse le risorse di memoria create in futuro.

JSON

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

```
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "xray:PutTraceSegments",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ForAllValues:ArnLike": {
          "logs:LogGeneratingResourceArns": "arn:aws:bedrock-agentcore:us-
east-1:123456789012:memory/*"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:delivery-
source:*"
        }
      }
    }
  ]
}
```

Altri AWS servizi

Per altri servizi che supportano la distribuzione delle tracce, AWS crea una politica delle risorse relativa alla risorsa di origine specifica.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "xray:PutTraceSegments",
      "Resource": "*",
      "Condition": {
```

```
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ForAllValues:ArnLike": {
          "logs:LogGeneratingResourceArns": "arn:aws:bedrock:us-
east-1:123456789012:knowledge-base/KnowledgeBaseId"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:delivery-
source:xray-test"
        }
      }
    }
  ]
}
```

Abilitazione di Transaction Search

Per abilitare l'invio di tracce a X-Ray, è necessario abilitare la ricerca delle transazioni.

Service-specific autorizzazioni

Oltre alle autorizzazioni specifiche della destinazione elencate nelle sezioni precedenti, alcuni servizi richiedono un'autorizzazione esplicita per consentire ai clienti di inviare i log dalle proprie risorse, come ulteriore livello di sicurezza. Questa politica autorizza l'`AllowVendedLogDeliveryForResource`azione per le risorse che vendono i log all'interno di quel servizio. Per questi servizi, utilizzate la seguente politica e sostituite lo spazio dei nomi del servizio (*`service`*), la regione (*`region`*), l'ID AWS account () e il tipo di risorsa (*`account-idresource-type`*) con i valori appropriati per la risorsa. Non tutti i servizi che vendono log utilizzano questa azione. Conferma le autorizzazioni e i valori richiesti, incluso l'eventuale applicabilità dell'`AllowVendedLogDeliveryForResource`azione, nella documentazione dei vendded logs relativa al servizio a cui concedi l'accesso. L'esempio seguente mostra la politica per i registri di vendita di Amazon SES.

JSON

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

```
        {
            "Sid": "ServiceLevelAccessForLogDelivery",
            "Effect": "Allow",
            "Action": [
                "ses:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:ses:us-east-1:123456789012:resource-type/*"
        }
    ]
}
```

Console-specific autorizzazioni

Oltre alle autorizzazioni elencate nelle sezioni precedenti, se si sta configurando la consegna dei log utilizzando la console anziché le API, sono necessarie anche le seguenti autorizzazioni aggiuntive:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogDeliveryActionsConsoleCWL",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleS3",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleFH",
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "firehose:DescribeDeliveryStream"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Cross-account esempio di consegna

In questo esempio, sono coinvolti due account. L'account con la risorsa che genera log è Account A, ID: e l'account con la risorsa che consuma log è Account B*123456789012*, ID:. *111122223333*

L'account A desidera inserire i log della Amazon Bedrock knowledge base nel proprio account con l'ARN *us-east-1* arn:aws:bedrock: ::knowledge-base/. *123456789012 kb-12345678*

Per questo esempio, l'account A necessita delle seguenti autorizzazioni:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowVendedLogDeliveryForKnowledgeBase",
            "Effect": "Allow",
            "Action": [
                "bedrock:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:bedrock:us-east-1:123456789012:knowledge-
base/XXXXXXXXXX"
        },
```

```
        {
            "Sid": "CreateLogDeliveryPermissions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliverySource",
                "logs:CreateDelivery"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:123456789012:delivery-source:*",
    "arn:aws:logs:us-east-1:123456789012:delivery:*",
    "arn:aws:logs:us-east-1:444455556666:delivery-destination:*"
            ]
        }
    ]
}
```

Crea una fonte di consegna

Per iniziare, l'account A crea una fonte di consegna con la sua base di conoscenze di base:

```
aws logs put-delivery-source --name my-delivery-source --log-type APPLICATION_LOGS --
resource-arn arn:aws:bedrock:region:AAAAAAAAAAAA:knowledge-base/XXXXXXXXXX
```

Successivamente, l'account B deve creare la destinazione di consegna utilizzando uno dei flussi seguenti:

- Configurare la consegna a un bucket Amazon S3
- Configurare la consegna a uno stream Firehose

Configurare la consegna a un bucket Amazon S3

L'account B desidera ricevere i log nel proprio bucket S3 con l'ARN arn:aws:s3: :amzn-s3-demo-bucket Per questo esempio, l'account B avrà bisogno delle seguenti autorizzazioni:

JSON

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

```
        {
            "Sid": "PutLogDestinationPermissions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliveryDestination",
                "logs:PutDeliveryDestinationPolicy"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:*"
        }
    ]
}
```

Il bucket avrà bisogno delle seguenti autorizzazioni nella sua politica relativa al bucket:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogsDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/AWSLogs/123456789012/
*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                    "123456789012"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:delivery-source:my-
delivery-source"
```

```
                    ]
                }
            }
        }
    ]
}
```

Se il bucket è crittografato con SSE-KMS, assicurati che la policy della AWS KMS chiave disponga delle autorizzazioni appropriate. Ad esempio, se la chiave KMS è`arn:aws:kms:`*`us-east-1`*`:`*`111122223333`*`:key/`*`1234abcd-12ab-34cd-56ef-1234567890ab`*, usa quanto segue:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogsGenerateDataKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                    "123456789012"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:delivery-source:my-
delivery-source"
                    ]
                }
            }
        }
```

```
    ]
}
```

L'account B può quindi creare una destinazione di consegna con il bucket S3 come risorsa di destinazione:

```
aws logs put-delivery-destination --name my-s3-delivery-destination --delivery-
destination-configuration "destinationResourceArn=arn:aws:s3:::amzn-s3-demo-bucket"
```

Successivamente, l'Account B crea una politica sulla destinazione di consegna sulla destinazione di consegna appena creata, che autorizzerà l'Account A a creare una consegna registrata. La politica che verrà aggiunta alla destinazione di consegna appena creata è la seguente:

JSON

```
    {
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {
                "AWS": "123456789012"
            },
            "Action": [
                "logs:CreateDelivery"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:amzn-s3-demo-bucket"
        }
    ]
}
```

Questa politica verrà salvata nel computer dell'Account B come `destination-policy-s3.json` Per collegare questa risorsa, l'Account B eseguirà il seguente comando:

```
aws logs put-delivery-destination-policy --delivery-destination-name my-s3-delivery-
destination --delivery-destination-policy file://destination-policy-s3.json
```

Infine, l'Account A crea la consegna, che collega la fonte di consegna nell'Account A alla destinazione di consegna nell'Account B.

```
aws logs create-delivery --delivery-source-name my-delivery-source --delivery-
destination-arn arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:my-s3-delivery-
destination
```

Configurare la consegna a uno stream Firehose

In questo esempio, l'account B desidera ricevere i log nel proprio stream Firehose. Lo stream Firehose ha il seguente ARN ed è configurato per utilizzare il DirectPut tipo di flusso di distribuzione:

`arn:aws:firehose:us-east-1:111122223333:deliverystream/log-delivery-stream`

Per questo esempio, l'account B necessita delle seguenti autorizzazioni:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFirehoseCreateSLR",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        },
        {
            "Sid": "AllowFirehoseTagging",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": "arn:aws:firehose:us-
east-1:111122223333:deliverystream/X"
```

```
        },
        {
            "Sid": "AllowFirehoseDeliveryDestination",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliveryDestination",
                "logs:PutDeliveryDestinationPolicy"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:*"
        }
    ]
}
```

Lo stream Firehose deve avere il tag `LogDeliveryEnabled` impostato su. `true`

L'account B creerà quindi una destinazione di consegna con lo stream Firehose come risorsa di destinazione:

```
aws logs put-delivery-destination --name my-fh-delivery-destination --delivery-
destination-configuration
 "destinationResourceArn=arn:aws:firehose:region:BBBBBBBBBBBB:deliverystream/X"
```

Successivamente, l'Account B crea una politica sulla destinazione di consegna sulla destinazione di consegna appena creata, che autorizzerà l'Account A a creare una consegna di log. La politica da aggiungere alla destinazione di consegna appena creata è la seguente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {
            "AWS": "123456789012"
            },
            "Action": [
                "logs:CreateDelivery"
            ],
```

```
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:amzn-s3-demo-bucket"
        }
    ]
}
```

Questa politica verrà salvata nel computer dell'Account B poiché `destination-policy-fh.json` Per collegare questa risorsa, l'Account B esegue il seguente comando:

```
aws logs put-delivery-destination-policy --delivery-destination-name my-fh-delivery-
destination --delivery-destination-policy file://destination-policy-fh.json
```

Infine, l'Account A crea la consegna, che collega la fonte di consegna nell'Account A alla destinazione di consegna nell'Account B.

```
aws logs create-delivery --delivery-source-name my-delivery-source --delivery-
destination-arn arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:my-fh-delivery-
destination
```

Cross-service confusa prevenzione sostitutiva

Il problema confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità maggiormente privilegiata a eseguire l'azione. Nel AWS, l'impersonificazione intersettoriale può portare al confuso problema del vicesceriffo. Cross-service l'impersonificazione può verificarsi quando un servizio (il servizio chiamante) chiama un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare che ciò accada, AWS mette a disposizione strumenti che consentono di proteggere i dati relativi a tutti i servizi con responsabili del servizio a cui è stato concesso l'accesso alle risorse dell'account.

Ti consigliamo di utilizzare le chiavi contestuali `aws:SourceArn`aws:SourceAccountaws:SourceOrgID,, e `aws:SourceOrgPaths`global condition nelle politiche delle risorse per limitare le autorizzazioni che CloudWatch Logs concede a un altro servizio alla risorsa. Utilizza `aws:SourceArn` per associare una sola risorsa all'accesso tra servizi. Utilizza `aws:SourceAccount` se desideri consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi. Utilizza `aws:SourceOrgID` se desideri consentire l'associazione

di qualsiasi risorsa di qualsiasi account interno a un'organizzazione all'uso tra servizi. Da utilizzare `aws:SourceOrgPaths` per associare qualsiasi risorsa proveniente dagli account all'interno di un AWS Organizations percorso all'utilizzo tra servizi. Per ulteriori informazioni sull'utilizzo e la comprensione dei percorsi, vedi Comprendere il percorso dell' AWS Organizations entità.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio un ARN di un bucket Amazon S3, è necessario utilizzare sia `aws:SourceAccount` che `aws:SourceArn` per limitare le autorizzazioni.

Per proteggersi dal problema "confused deputy" su larga scala, nelle policy basate sulle risorse utilizza la chiave di contesto della condizione globale `aws:SourceOrgID` o `aws:SourceOrgPaths` con l'ID dell'organizzazione o il percorso dell'organizzazione della risorsa. Quando aggiungi, rimuovi o sposti degli account all'interno dell'organizzazione, le policy che includono la chiave `aws:SourceOrgID` o `aws:SourceOrgPaths` includono automaticamente anche gli account corretti e non necessitano dell'aggiornamento manuale.

Le politiche documentate per concedere l'accesso ai CloudWatch log per scrivere dati su Kinesis Data Streams e Passaggio 1: creazione di una destinazione Fase 2: creazione di una destinazione Firehose mostrano come utilizzare la chiave global condition context per prevenire il `aws:SourceArn` confuso problema del vice.

Automatizza l'abilitazione dei log con le regole di abilitazione della telemetria

Puoi utilizzare le regole di abilitazione della telemetria per configurare automaticamente la raccolta dei log per le tue risorse. AWS Le regole ti aiutano a standardizzare la raccolta dei log in tutta l'organizzazione o negli account e a garantire una copertura di monitoraggio coerente.

Le regole di abilitazione consentono di:

- Abilita automaticamente la registrazione per le risorse nuove ed esistenti che corrispondono all'ambito della regola

- Applica le regole a livello di organizzazione, unità organizzativa (OU) o singolo account
- Filtra le risorse influenzate da una regola utilizzando i tag

Le regole di abilitazione attualmente supportano le seguenti fonti di AWS telemetria: Amazon VPC Flow Logs AWS WAF , Logs, Route 53 Resolver Query Logs, NLB Access Logs, Control Plane Logs, CloudTrail Data and Management Events, Amazon Bedrock Logs, Amazon EC2 Detailed Metrics, Security Hub, Amazon Bedrock Gateway AgentCore , Amazon Bedrock Memory e Distribution. AWS AgentCore AgentCore CloudFront

Per dettagli completi sulle regole di abilitazione, tra cui il comportamento delle regole, le regole di gestione, la risoluzione dei problemi e considerazioni specifiche del servizio, consulta le regole di abilitazione della telemetria nella Amazon User Guide. CloudWatch

Gerarchia di valutazione delle regole

Le regole di abilitazione vengono valutate gerarchicamente: prima le regole organizzative, poi le regole, quindi OU-level le regole a livello di account. Le regole di livello superiore forniscono la telemetria di base. Lower-level le regole possono aggiungere telemetria aggiuntiva ma non possono ridurla. Se esistono regole in conflitto nello stesso ambito, non viene applicata alcuna regola fino alla risoluzione del conflitto.

Creazione di una regola di abilitazione

Per creare una regola di abilitazione della telemetria

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione, scegli Configurazione della telemetria.
3. Scegli la scheda Regole di abilitazione.
4. Scegli Aggiungi regola.
5. Specificare il nome della regola, l'ambito (organizzazione, unità organizzativa o account), l'origine dati e il tipo di telemetria.
6. Facoltativamente, aggiungi tag per filtrare le risorse interessate dalla regola.
7. Scegli Crea regola.

Abilita la registrazione da fonti di terze parti

CloudWatch Logs supporta l'acquisizione di log da fonti di terze parti tramite integrazioni API dirette e integrazioni di bucket Amazon S3. Puoi anche ricevere ulteriori risultati di sicurezza di terze parti tramite AWS Security Hub CSPM.

Integrazioni dirette di terze parti

CloudWatch Logs fornisce integrazioni dirette con le seguenti fonti di terze parti. Queste integrazioni utilizzano connessioni API dirette o integrazioni di bucket Amazon S3 per inserire i log nei log: CloudWatch

- CrowdStrike Falcon
- Microsoft Office 365
- Okta Auth0
- ID Microsoft Entra
- Palo Alto Networks NGFW
- Eventi Microsoft Windows
- Wiz
- Accesso a Internet Zscaler
- Okta SSO
- SentinelOne Sicurezza degli endpoint
- GitHub Registri di controllo
- ServiceNow CMDB
- Ombrello Cisco

Per istruzioni di configurazione, consulta la guida alla configurazione dell'integrazione di terze parti nella Amazon CloudWatch User Guide.

Fonti aggiuntive tramite AWS Security Hub CSPM

Oltre alle integrazioni dirette, i partner di sicurezza di terze parti inviano i risultati a AWS Security Hub CSPM, che sono quindi disponibili come fonti di dati in Logs. CloudWatch La tabella seguente elenca le integrazioni dei partner CSPM di Security Hub e il loro tipo di integrazione.

Per abilitare i risultati CSPM di Security Hub come origine dati nei CloudWatch registri, crea una regola di abilitazione della telemetria per Security Hub nella console. AWS CloudWatch La regola di abilitazione si configura per inserire automaticamente i CloudWatch risultati da Security Hub CSPM in un gruppo di log gestito. Per step-by-step istruzioni, consulta le regole di abilitazione della telemetria nella Amazon User Guide. CloudWatch

Partner	Integrazione
CORESec 3 — NTA	Invia i risultati tramite Security Hub CSPM
Alert Logic — SIEMless Gestione delle minacce	Invia i risultati tramite Security Hub CSPM
Aqua Security: piattaforma di sicurezza nativa per il cloud	Invia i risultati tramite Security Hub CSPM
Aqua Security — Kube-bench	Invia i risultati tramite Security Hub CSPM
Armatura: armatura ovunque	Invia i risultati tramite Security Hub CSPM
AttackIQ	Invia i risultati tramite Security Hub CSPM
Barracuda Networks — Cloud Security Guardian	Invia i risultati tramite Security Hub CSPM
BigID — BigID Enterprise	Invia i risultati tramite Security Hub CSPM
Esagono blu	Invia i risultati tramite Security Hub CSPM
Check Point — CloudGuard IaaS	Invia i risultati tramite Security Hub CSPM

Partner	Integrazione
Check Point — Gestione della postura CloudGuard	Invia i risultati tramite Security Hub CSPM
Clarity — XDome	Invia i risultati tramite Security Hub CSPM
Sicurezza dell'archiviazione nel cloud: antivirus per Amazon S3	Invia i risultati tramite Security Hub CSPM
Contrast Security — Contrast Assessment	Invia i risultati tramite Security Hub CSPM
CrowdStrike — Falco CrowdStrike	Invia i risultati tramite Security Hub CSPM
CyberArk — Analisi privilegiata delle minacce	Invia i risultati tramite Security Hub CSPM
Teorema dei dati	Invia i risultati tramite Security Hub CSPM
Dati	Invia i risultati tramite Security Hub CSPM
Forcepoint — CASB	Invia i risultati tramite Security Hub CSPM
Forcepoint — Gateway di sicurezza nel cloud	Invia i risultati tramite Security Hub CSPM
Forcepoint — DLP	Invia i risultati tramite Security Hub CSPM
Forcepoint — NGFW	Invia i risultati tramite Security Hub CSPM
Fuga	Invia i risultati tramite Security Hub CSPM

Partner	Integrazione
Guardicore — Centra	Invia i risultati tramite Security Hub CSPM
HackerOne — Intelligenza sulle vulnerabilità	Invia i risultati tramite Security Hub CSPM
JFrog — Radiografia	Invia i risultati tramite Security Hub CSPM
Juniper Networks: firewall vSRx di nuova generazione	Invia i risultati tramite Security Hub CSPM
k9 Security — Analizzatore di accesso	Invia i risultati tramite Security Hub CSPM
Merletti	Invia i risultati tramite Security Hub CSPM
McAfee — MVISION CNAPP	Invia i risultati tramite Security Hub CSPM
NETSCOUT — Investigatore informatico	Invia i risultati tramite Security Hub CSPM
Orca — Piattaforma di sicurezza cloud	Invia i risultati tramite Security Hub CSPM
Palo Alto Networks — Prisma Cloud Compute	Invia i risultati tramite Security Hub CSPM
Palo Alto Networks — Prisma Cloud Enterprise	Invia i risultati tramite Security Hub CSPM
Plerion — Piattaforma di sicurezza cloud	Invia i risultati tramite Security Hub CSPM
che si aggira	Invia i risultati tramite Security Hub CSPM

Partner	Integrazione
Qualys — Gestione delle vulnerabilità	Invia i risultati tramite Security Hub CSPM
Rapid7 — InsightVM	Invia i risultati tramite Security Hub CSPM
SentinelOne	Invia i risultati tramite Security Hub CSPM
Snyk	Invia i risultati tramite Security Hub CSPM
Sonrai Security — Sonrai Dig	Invia i risultati tramite Security Hub CSPM
Sophos — Protezione dei server	Invia i risultati tramite Security Hub CSPM
StackRox — Sicurezza Kubernetes	Invia i risultati tramite Security Hub CSPM
Sumo Logic — Analisi dei dati delle macchine	Invia i risultati tramite Security Hub CSPM
Symantec — Protezione del carico di lavoro nel cloud	Invia i risultati tramite Security Hub CSPM
Tenable.io	Invia i risultati tramite Security Hub CSPM
Trend Micro — Cloud One	Invia i risultati tramite Security Hub CSPM
Vectra — Cognito Detect	Invia i risultati tramite Security Hub CSPM
Wiz	Invia i risultati tramite Security Hub CSPM

Partner	Integrazione
Caveonix — Caveonix Cloud	Invia e riceve i risultati tramite Security Hub CSPM
Cloud Custodian	Invia e riceve i risultati tramite Security Hub CSPM
DisruptOps	Invia e riceve i risultati tramite Security Hub CSPM
Kion	Invia e riceve i risultati tramite Security Hub CSPM
Turbot	Invia e riceve i risultati tramite Security Hub CSPM

Note

Questo elenco riflette le integrazioni dei partner di Security Hub che inviano i risultati al momento della stesura. Poiché AWS Security Hub aggiunge regolarmente nuove integrazioni per i partner, consulta la sezione Integrazioni di prodotti di terze parti con Security Hub CSPM nella AWS Security Hub User Guide per l' up-to-dateelenco più completo dei partner disponibili.

Esportazione di dati di log in Amazon S3

Questo capitolo fornisce informazioni che consentono di esportare i dati di log dai gruppi di log in un bucket Amazon S3 per l'elaborazione e l'analisi personalizzate o per caricarli su altri sistemi. Puoi esportare in un bucket S3 nello stesso account o in un account diverso.

Puoi fare quanto segue:

- Esporta i dati di registro in bucket S3 crittografati da SSE-KMS in AWS Key Management Service ()AWS KMS
- Esportazione dei dati di log in bucket S3 per i quali S3 Object Lock è abilitato con un periodo di conservazione

Ti consigliamo di non esportare regolarmente in Amazon S3 per archiviare continuamente i log. In questo caso d'uso, ti consigliamo invece di utilizzare gli abbonamenti. Per ulteriori informazioni sugli abbonamenti, consulta. Real-time trattamento dei dati di registro con abbonamenti

Per iniziare il processo di esportazione, è necessario creare un bucket S3 per archiviare i dati di log esportati. Puoi archiviare i file esportati nel bucket S3 e definire regole del ciclo di vita Amazon S3 per archiviare o eliminare i file esportati automaticamente.

Puoi esportare in bucket S3 crittografati con o con AES-256 . SSE-KMS L'esportazione in bucket crittografati con DSSE-KMS non è supportata.

Puoi esportare i log da più gruppi di log o da più intervalli di tempo nello stesso bucket S3. Per organizzare i dati esportati, specifica un prefisso per ogni attività di esportazione che verrà utilizzato come prefisso chiave di Amazon S3 per tutti gli oggetti esportati. Ad esempio, `prod/app-logs/2026-01-03/` o `log-group-name/backup/`

> **Note**
>
> Time-based l'ordinamento in base a blocchi di dati di log all'interno di un file esportato non è garantito. Puoi ordinare i dati del campo di log esportati utilizzando le utilità Linux. Ad esempio, il seguente comando di utilità ordina gli eventi in tutti i file `.gz` in una singola cartella.
>
> ```
> find . -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
> ```

Il seguente comando di utilità ordina i file .gz da più sottocartelle.

```
find ./*/ -type f -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
```

Inoltre, puoi usare un altro comando `stdout` per reindirizzare l'output ordinato a un altro file per salvarlo.

Potrebbero essere necessarie fino a 12 ore affinché i dati di log diventino disponibili per l'esportazione. Le attività di esportazione scadono dopo 24 ore. Se le tue attività di esportazione sono scadute, riduci l'intervallo di tempo quando crei l'attività di esportazione.

Per un'analisi quasi in tempo reale dei dati di log, consulta Analisi dei dati di registro con CloudWatch Logs Insights o Real-time trattamento dei dati di registro con abbonamenti.

Indice

- Concetti
- Esportazione di dati di log in Amazon S3 tramite la console
- Esporta i dati di registro su Amazon S3 utilizzando il AWS CLI
- Descrivere le attività di esportazione (CLI)
- Annullare un'attività di esportazione (CLI)

Concetti

Prima di iniziare, acquisisci familiarità con i seguenti concetti di esportazione:

nome gruppo di log

Il nome del gruppo di log associato a un'attività di esportazione. I dati di log di questo gruppo di log verranno esportati nel bucket S3 specificato.

da (timestamp)

Un timestamp obbligatorio espresso come il numero di millisecondi a partire dal 1° gennaio 1970 00:00:00 UTC. Tutti gli eventi di registro nel gruppo di log che sono stati inseriti in questo periodo o dopo tale periodo verranno esportati.

a (timestamp)

Un timestamp obbligatorio espresso come il numero di millisecondi a partire dal 1° gennaio 1970 00:00:00 UTC. Tutti gli eventi di log nel gruppo di log inseriti prima di questo momento saranno esportati.

bucket di destinazione

Il nome del bucket S3 associato a un'attività di esportazione. Questo bucket viene utilizzato per esportare i dati di log dal gruppo di log specificato.

prefisso di destinazione

Un attributo facoltativo utilizzato come prefisso della chiave Amazon S3 per tutti gli oggetti esportati. Questo ti aiuta a creare un'organizzazione con stile cartella nel bucket.

Esportazione di dati di log in Amazon S3 tramite la console

Negli esempi seguenti, utilizzi la CloudWatch console Amazon per esportare tutti i dati da un gruppo di log di Amazon CloudWatch Logs denominato in un `my-log-group` bucket Amazon S3 denominato. `amzn-s3-demo-bucket`

È supportata l'esportazione dei dati di log in bucket S3 crittografati da. SSE-KMS L'esportazione in bucket crittografati con non è supportata. DSSE-KMS

I dettagli su come configuri l'esportazione dipendono dal fatto che il bucket Amazon S3 in cui desideri esportare si trovi nello stesso account dei log che vengono esportati o in un account diverso.

Argomenti

- Same-account esportazione (console)
- Cross-account esporta (console)

Same-account esportazione (console)

Se il bucket Amazon S3 si trova nello stesso account dei log che vengono esportati, segui le istruzioni in questa sezione.

Argomenti

- Crea un bucket Amazon S3 (console)

- Configura le autorizzazioni di accesso (console)
- Imposta le autorizzazioni su un bucket Amazon S3 (console)
- (Facoltativo) Esportazione verso un bucket Amazon S3 di destinazione crittografato SSE-KMS con (console)
- Crea un'attività di esportazione (console)

Crea un bucket Amazon S3 (console)

Ti consigliamo di utilizzare un bucket creato appositamente per i log. CloudWatch Tuttavia, se intendi utilizzare un bucket esistente, puoi passare alla fase 2.

> **Note**
>
> Il bucket Amazon S3 deve risiedere nella stessa Regione dei dati di registro da esportare. CloudWatch Logs non supporta l'esportazione di dati in bucket Amazon S3 in un'altra regione.

Come creare un bucket Amazon S3.

1. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Se necessario, modificare la regione . Dalla barra di navigazione, scegli la regione in cui risiedono CloudWatch i log.
3. Scegli Crea bucket.
4. In Bucket Name (Nome bucket), immettere il nome del bucket.
5. Per Regione, seleziona la regione in cui risiedono i dati CloudWatch dei registri.
6. Scegli Crea.

Configura le autorizzazioni di accesso (console)

Per creare l'attività di esportazione, devi accedere con il ruolo `AmazonS3ReadOnlyAccess` IAM e disporre delle seguenti autorizzazioni:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`

- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in Centro identità AWS IAM:

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di Centro identità AWS IAM .
- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Create a role for a third-party identity provider (federation) della Guida per l'utente IAM.
- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella pagina Create a role for an IAM user della Guida per l'utente IAM.
  - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

Imposta le autorizzazioni su un bucket Amazon S3 (console)

Per impostazione predefinita, tutti gli oggetti e i bucket Amazon S3 sono privati. Solo il proprietario della risorsa, colui Account AWS che ha creato il bucket, può accedere al bucket e a tutti gli oggetti in esso contenuti. Tuttavia, il proprietario della risorsa può concedere le autorizzazioni di accesso ad altre risorse e ad altri utenti mediante una policy di accesso.

Quando si imposta la policy, consigliamo di includere una stringa generata in modo casuale come prefisso per il bucket, in modo che solo i flussi di log vengano esportati nel bucket.

> **Important**
>
> Per rendere più sicure le esportazioni verso i bucket Amazon S3, ora ti chiediamo di specificare l'elenco degli account di origine autorizzati a esportare i dati di log nel tuo bucket S3.
> Nell'esempio seguente, l'elenco degli ID account nella `aws:SourceAccount` chiave sarebbe costituito dagli account da cui un utente può esportare i dati di registro nel bucket Amazon

S3. La chiave `aws:SourceArn` è la risorsa per la quale viene intrapresa l'azione. Puoi limitarla a un gruppo di log specifico o utilizzare un jolly come mostrato in questo esempio. Ti consigliamo di includere anche l'ID dell'account in cui è stato creato il bucket S3 per consentire l'esportazione all'interno dello stesso account.

Impostazione delle autorizzazioni su un bucket Amazon S3

1. Nella console Amazon S3, scegli il bucket che hai creato.
2. Selezionare Permissions (Autorizzazioni), Add bucket policy (Aggiungi policy bucket).
3. In Bucket Policy Editor (Editor della policy del bucket), aggiungi la seguente policy. Cambia `amzn-s3-demo-bucket` nel nome del bucket S3. Assicurati di specificare l'endpoint corretto della regione, come `us-west-1`, per Principale.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
          "Sid": "AllowCloudWatchLogsGetBucketAcl",
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "123456789012",
                    "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:123456789012:log-group:*",
                        "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
            }
          }
```

```
        },
        {
            "Sid": "AllowCloudWatchLogsPutObject",
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "123456789012",
                      "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:us-east-1:123456789012:log-group:*",
                          "arn:aws:logs:us-east-1:111122223333:log-group:*"
                      ]
              }
            }
        }
    ]
}
```

4. Seleziona Save (Salva) per impostare la policy appena aggiunta come la policy di accesso all'interno del bucket. Questa policy consente a CloudWatch Logs di esportare i dati di log nel tuo bucket Amazon S3. Il proprietario del bucket dispone di autorizzazioni complete su tutti gli oggetti esportati.

> **Warning**
>
> Se al bucket esistente sono già associate una o più politiche, aggiungi le istruzioni per CloudWatch Logs access a quella o alle politiche. Consigliamo di valutare il set di autorizzazioni risultante per verificare che siano adeguate agli utenti che accederanno al bucket.

(Facoltativo) Esportazione verso un bucket Amazon S3 di destinazione crittografato SSE-KMS con (console)

Questo passaggio è necessario solo se si esegue l'esportazione in un bucket Amazon S3 che utilizza la crittografia lato server con. AWS KMS keys Questa crittografia è nota come. SSE-KMS

Esportare in un bucket crittografato con SSE-KMS

1. Apri la AWS KMS console all'indirizzo https://console.aws.amazon.com/kms.
2. Per modificare la Regione AWS, usa il selettore della regione nell'angolo superiore destro della pagina.
3. Nella barra di navigazione a sinistra, scegli Customer managed keys (Chiavi gestite dal cliente).

   Scegli Create Key (Crea chiave).
4. Alla voce Key type (Tipo di chiave), scegliere Symmetric (Simmetrica).
5. Per Key usage (Utilizzo della chiave), scegli Encrypt and decrypt (Crittografa e decrittografa), quindi scegli Next (Avanti).
6. In Add labels (Aggiungi etichette), inserisci un alias per la chiave e, facoltativamente, aggiungi una descrizione o dei tag. Quindi scegli Successivo.
7. In Key administrators (Amministratori delle chiavi), seleziona chi può amministrare questa chiave, quindi scegli Next (Avanti).
8. In Define key usage permissions (Definisci le autorizzazioni per utilizzare la chiave), non apportare modifiche e scegli Next (Avanti).
9. Esamina le impostazioni e scegli Finish (Fine).
10. Torna alla pagina Customer managed keys (Chiavi gestite dal cliente) e scegli il nome della chiave che hai appena creato.
11. Nella sezione Key policy (Policy chiave), scegli Switch to policy view (Passa alla visualizzazione della policy).
12. Nella sezione Key policy (Policy chiave), scegli Edit (Modifica).
13. Aggiungi la seguente istruzione all'elenco delle istruzioni della policy chiave. Quando lo fai, *`Region`* sostituiscilo con la regione dei tuoi log e sostituiscilo *`account-ARN`* con l'ARN dell'account che possiede la chiave KMS.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

14. Scegli Save changes (Salva modifiche).
15. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
16. Cerca il bucket che hai creato in Creare un bucket S3 (CLI) e scegli il nome del bucket.

17. Scegliere la scheda Properties (Proprietà). Quindi, in Default Encryption (Crittografia predefinita), scegli Edit (Modifica).
18. In Server-side Crittografia, scegli Abilita.
19. In Tipo di crittografia, scegli AWS Key Management Service chiave (SSE-KMS).
20. Scegli tra AWS KMS le tue chiavi e trova la chiave che hai creato.
21. Per Bucket key (Chiave bucket), scegli Enable (Abilita).
22. Scegli Save changes (Salva modifiche).

Crea un'attività di esportazione (console)

In questa procedura, si crea l'attività di esportazione per esportare i log da un gruppo di log.

Per esportare dati su Amazon S3 utilizzando la console CloudWatch

1. Accedi con le autorizzazioni sufficienti come documentato in Configura le autorizzazioni di accesso (console).
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
4. Nella schermata Gruppi di log, scegliere il nome del gruppo di log.
5. Scegli Actions (Operazioni), Export data to Amazon S3 (Esporta dati in Amazon S3.
6. Nella schermata Export data to Amazon S3 (Esporta dati in Amazon S3), in Define data export (Definizione dell'esportazione dei dati), impostare l'intervallo di tempo per i dati da esportare in From (Da) e To (A).
7. Se il gruppo di log dispone di più flussi di log, puoi fornire un prefisso del flusso di log per limitare i dati del gruppo di log a un flusso specifico. Scegliere Advanced (Avanzato) e immettere il prefisso del flusso di log in Stream prefix (Prefisso flusso).
8. In Choose S3 bucket (Scegli bucket S3), seleziona l'account associato al bucket S3.
9. Per S3 bucket name (Nome bucket S3), scegli un bucket Amazon S3.
10. Per Prefisso bucket S3, immettere la stringa generata in modo casuale specificata nella policy del bucket.
11. Seleziona Esporta per esportare i dati di log in Amazon S3.
12. Per visualizzare lo stato dei dati di log esportati in Amazon S3, scegli Operazioni, quindi Visualizza tutte le esportazioni in Amazon S3.

Cross-account esporta (console)

Se il bucket Amazon S3 si trova in un account diverso da quello dei log che vengono esportati, segui le istruzioni in questa sezione.

Argomenti

- Crea un bucket Amazon S3 per l'esportazione tra account (console)
- Configura le autorizzazioni di accesso per l'esportazione tra account (console)
- Imposta le autorizzazioni su un bucket S3 per l'esportazione tra account (console)
- (Facoltativo) Esportazione verso un bucket Amazon S3 di destinazione crittografato per l'esportazione tra SSE-KMS account (console)
- Crea un'attività di esportazione per l'esportazione tra account diversi (console)

Crea un bucket Amazon S3 per l'esportazione tra account (console)

Ti consigliamo di utilizzare un bucket creato appositamente per i log. CloudWatch Tuttavia, se desideri utilizzare un bucket esistente, puoi saltare questa procedura.

> **Note**
>
> Il bucket Amazon S3 deve risiedere nella stessa Regione dei dati di registro da esportare. CloudWatch Logs non supporta l'esportazione di dati in bucket Amazon S3 in un'altra regione.

Come creare un bucket Amazon S3.

1. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Se necessario, modificare la regione . Dalla barra di navigazione, scegli la regione in cui risiedono CloudWatch i log.
3. Scegli Crea bucket.
4. In Bucket Name (Nome bucket), immettere il nome del bucket.
5. Per Regione, seleziona la regione in cui risiedono i dati CloudWatch dei registri.
6. Scegli Crea.

Configura le autorizzazioni di accesso per l'esportazione tra account (console)

Innanzitutto, devi creare una nuova policy IAM per consentire a CloudWatch Logs di avere l'`s3:PutObject`azione per il bucket Amazon S3 di destinazione nell'account di destinazione.

Oltre all'`s3:PutObject`azione, le azioni aggiuntive incluse nella policy dipendono dal fatto che il bucket di destinazione utilizzi la AWS KMS crittografia o che gli ACL siano abilitati utilizzando l'impostazione S3 Object Ownership.

- Se utilizzi la crittografia KMS, aggiungi le `kms:Decrypt` azioni `kms:GenerateDataKey` and per la risorsa chiave
- Se gli ACL sono abilitati sul bucket, aggiungi l'`s3:PutObjectAcl`azione per la risorsa bucket

Passa `amzn-s3-demo-bucket` al nome del bucket S3 di destinazione nelle seguenti politiche.

Creazione di una policy IAM per esportare i log in un bucket Amazon S3

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Se il bucket di destinazione non utilizza la AWS KMS crittografia, incolla la seguente politica nell'editor.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Se il bucket di destinazione utilizza la AWS KMS crittografia, incolla la seguente politica nell'editor.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ]
}
```

Se gli ACL sono abilitati sul bucket di destinazione, aggiungi il blocco s3: PutObjectAcl to s3: PutObject Action nelle politiche precedenti.

JSON

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

```
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

6. Scegli Avanti.
7. Inserire un nome per la policy. Utilizzerai questo nome per collegare la policy al tuo ruolo IAM.
8. Quindi, per salvare la nuova policy, seleziona Crea policy.

Per creare un'attività di esportazione, devi accedere con un ruolo IAM a cui è associata la policy `AmazonS3ReadOnlyAccess` gestita, la policy IAM creata sopra e anche con le seguenti autorizzazioni:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in Centro identità AWS IAM:

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di Centro identità AWS IAM .
- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Create a role for a third-party identity provider (federation) della Guida per l'utente IAM.
- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella pagina Create a role for an IAM user della Guida per l'utente IAM.
  - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

Imposta le autorizzazioni su un bucket S3 per l'esportazione tra account (console)

Per impostazione predefinita, tutti i bucket e gli oggetti S3 sono privati. Solo il proprietario della risorsa, colui Account AWS che ha creato il bucket, può accedere al bucket e a tutti gli oggetti in esso contenuti. Tuttavia, il proprietario della risorsa può concedere le autorizzazioni di accesso ad altre risorse e ad altri utenti mediante una policy di accesso.

Quando si imposta la policy, consigliamo di includere una stringa generata in modo casuale come prefisso per il bucket, in modo che solo i flussi di log vengano esportati nel bucket.

> **Important**
>
> Per rendere più sicure le esportazioni verso i bucket S3, ora ti chiediamo di specificare l'elenco degli account di origine autorizzati a esportare i dati di log nel tuo bucket S3. Nell'esempio seguente, l'elenco degli ID account nella `aws:SourceAccount` chiave sarebbe costituito dagli account da cui un utente può esportare i dati di registro nel bucket S3. La chiave `aws:SourceArn` è la risorsa per la quale viene intrapresa l'azione. Puoi limitarla a un gruppo di log specifico o utilizzare un jolly come mostrato in questo esempio.
> Ti consigliamo di includere anche l'ID dell'account in cui è stato creato il bucket S3 per consentire l'esportazione all'interno dello stesso account.

Impostazione delle autorizzazioni su un bucket Amazon S3

1. Nella console Amazon S3, scegli il bucket che hai creato.
2. Selezionare Permissions (Autorizzazioni), Add bucket policy (Aggiungi policy bucket).
3. In Bucket Policy Editor (Editor della policy del bucket), aggiungi la seguente policy. Cambia `amzn-s3-demo-bucket` nel nome del bucket S3. Assicurati di specificare l'endpoint corretto della regione, come `us-east-1`, per Principale.

   JSON

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

```
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                      "123456789012",
                      "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:us-east-1:123456789012:log-group:*",
                          "arn:aws:logs:us-east-1:111122223333:log-group:*"
                      ]
              }
            }
        },
        {
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "123456789012",
                      "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:us-east-1:123456789012:log-group:*",
                          "arn:aws:logs:us-east-1:111122223333:log-group:*"
                      ]
              }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
              "AWS": "arn:aws:iam::111122223333:role/role_name"
            },
            "Action": "s3:PutObject",
```

```
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control"
              }
            }
        }
    ]
}
```

4. Seleziona Save (Salva) per impostare la policy appena aggiunta come la policy di accesso all'interno del bucket. Questa politica consente a CloudWatch Logs di esportare i dati di registro nel bucket S3. Il proprietario del bucket dispone di autorizzazioni complete su tutti gli oggetti esportati.

> **Warning**
>
> Se al bucket esistente sono già associate una o più politiche, aggiungi le istruzioni per CloudWatch Logs access a quella o più policy. Consigliamo di valutare il set di autorizzazioni risultante per verificare che siano adeguate agli utenti che accederanno al bucket.

(Facoltativo) Esportazione verso un bucket Amazon S3 di destinazione crittografato per l'esportazione tra SSE-KMS account (console)

Questa procedura è necessaria solo se si esegue l'esportazione in un bucket S3 che utilizza la crittografia lato server con. AWS KMS keys Questa crittografia è nota come. SSE-KMS

Esportare in un bucket crittografato con SSE-KMS

1. Apri la AWS KMS console all'indirizzo https://console.aws.amazon.com/kms.
2. Per modificare la Regione AWS, usa il selettore della regione nell'angolo superiore destro della pagina.
3. Nella barra di navigazione a sinistra, scegli Customer managed keys (Chiavi gestite dal cliente).

   Scegli Create Key (Crea chiave).
4. Alla voce Key type (Tipo di chiave), scegliere Symmetric (Simmetrica).

5. Per Key usage (Utilizzo della chiave), scegli Encrypt and decrypt (Crittografa e decrittografa), quindi scegli Next (Avanti).
6. In Add labels (Aggiungi etichette), inserisci un alias per la chiave e, facoltativamente, aggiungi una descrizione o dei tag. Quindi scegli Successivo.
7. In Key administrators (Amministratori delle chiavi), seleziona chi può amministrare questa chiave, quindi scegli Next (Avanti).
8. In Define key usage permissions (Definisci le autorizzazioni per utilizzare la chiave), non apportare modifiche e scegli Next (Avanti).
9. Esamina le impostazioni e scegli Finish (Fine).
10. Torna alla pagina Customer managed keys (Chiavi gestite dal cliente) e scegli il nome della chiave che hai appena creato.
11. Nella sezione Key policy (Policy chiave), scegli Switch to policy view (Passa alla visualizzazione della policy).
12. Nella sezione Key policy (Policy chiave), scegli Edit (Modifica).
13. Aggiungi la seguente istruzione all'elenco delle istruzioni della policy chiave. Quando lo fai, *us-east-1* sostituiscilo con la regione dei tuoi log, *account-ARN* con l'ARN dell'account che possiede la chiave KMS, con il numero di account che possiede la chiave KMS*123456789012*, con l'ID della chiave *key_id* kms *role_name* e con il ruolo utilizzato per creare l'attività di esportazione.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:123456789012:key/key-id"
        }
    ]
}
```

14. Scegli Save changes (Salva modifiche).
15. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
16. Cerca il bucket che hai creato in Creare un bucket S3 (CLI) e scegli il nome del bucket.
17. Scegliere la scheda Properties (Proprietà). Quindi, in Default Encryption (Crittografia predefinita), scegli Edit (Modifica).
18. In Server-side Crittografia, scegli Abilita.
19. In Tipo di crittografia, scegli AWS Key Management Service chiave (SSE-KMS).
20. Scegli tra AWS KMS le tue chiavi e trova la chiave che hai creato.
21. Per Bucket key (Chiave bucket), scegli Enable (Abilita).

22. Scegli Save changes (Salva modifiche).

Crea un'attività di esportazione per l'esportazione tra account diversi (console)

In questa procedura, si crea l'attività di esportazione per esportare i log da un gruppo di log.

Per esportare dati su Amazon S3 utilizzando la console CloudWatch

1. Accedi con le autorizzazioni sufficienti come documentato in Configura le autorizzazioni di accesso (console).
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
4. Nella schermata Gruppi di log, scegliere il nome del gruppo di log.
5. Scegli Actions (Operazioni), Export data to Amazon S3 (Esporta dati in Amazon S3.
6. Nella schermata Export data to Amazon S3 (Esporta dati in Amazon S3), in Define data export (Definizione dell'esportazione dei dati), impostare l'intervallo di tempo per i dati da esportare in From (Da) e To (A).
7. Se il gruppo di log dispone di più flussi di log, puoi fornire un prefisso del flusso di log per limitare i dati del gruppo di log a un flusso specifico. Scegliere Advanced (Avanzato) e immettere il prefisso del flusso di log in Stream prefix (Prefisso flusso).
8. In Choose S3 bucket (Scegli bucket S3), seleziona l'account associato al bucket S3.
9. In S3 bucket name (Nome bucket S3), seleziona un bucket S3.
10. Per Prefisso bucket S3, immettere la stringa generata in modo casuale specificata nella policy del bucket.
11. Seleziona Esporta per esportare i dati di log in Amazon S3.
12. Per visualizzare lo stato dei dati di log esportati in Amazon S3, scegli Operazioni, quindi Visualizza tutte le esportazioni in Amazon S3.

Esporta i dati di registro su Amazon S3 utilizzando il AWS CLI

Nell'esempio seguente, si utilizza un'attività di esportazione per esportare tutti i dati da un gruppo di log CloudWatch Logs denominato in un `my-log-group` bucket Amazon S3 denominato. `amzn-s3-demo-bucket` Questo esempio presuppone che tu abbia già creato il gruppo di log `my-log-group`.

È supportata l'esportazione dei dati di registro in bucket S3 crittografati da. AWS KMS L'esportazione in bucket crittografati con non è supportata. DSSE-KMS

I dettagli su come configuri l'esportazione dipendono dal fatto che il bucket Amazon S3 in cui desideri esportare si trovi nello stesso account dei log che vengono esportati o in un account diverso.

Argomenti

- Same-account esportazione (CLI)
- Cross-account esportazione (CLI)

Same-account esportazione (CLI)

Se il bucket Amazon S3 si trova nello stesso account dei log che vengono esportati, segui le istruzioni in questa sezione.

Argomenti

- Creare un bucket S3 (CLI)
- Configurare le autorizzazioni di accesso (CLI)
- Imposta le autorizzazioni su un bucket S3 (CLI)
- (Facoltativo) Esportazione verso un bucket Amazon S3 di destinazione crittografato con SSE-KMS (CLI)
- Creare un'attività di esportazione (CLI)

Creare un bucket S3 (CLI)

Ti consigliamo di utilizzare un bucket creato appositamente per i log. CloudWatch Tuttavia, se desideri utilizzare un bucket esistente, puoi saltare questa procedura.

> **Note**
>
> Il bucket S3 deve trovarsi nella stessa Regione dei dati di log da esportare. CloudWatch I registri non supportano l'esportazione di dati in bucket S3 in una regione diversa.

Per creare un bucket S3 utilizzando il AWS CLI

Al prompt dei comandi, eseguire il seguente comando create-bucket, in cui `LocationConstraint` è la regione in cui esportare i dati di log.

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-2
```

Di seguito è riportato un output di esempio.

```
{
    "Location": "/amzn-s3-demo-bucket"
}
```

Configurare le autorizzazioni di accesso (CLI)

Per creare l'attività di esportazione in un secondo momento, dovrai accedere con il ruolo `AmazonS3ReadOnlyAccess` IAM e disporre delle seguenti autorizzazioni:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in Centro identità AWS IAM:

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di Centro identità AWS IAM .
- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Create a role for a third-party identity provider (federation) della Guida per l'utente IAM.
- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella pagina Create a role for an IAM user della Guida per l'utente IAM.

- (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

Imposta le autorizzazioni su un bucket S3 (CLI)

Per impostazione predefinita, tutti i bucket e gli oggetti S3 sono privati. Solo il proprietario della risorsa, l'account che ha creato il bucket, può accedere al bucket e agli oggetti in esso contenuti. Tuttavia, il proprietario della risorsa può concedere le autorizzazioni di accesso ad altre risorse e ad altri utenti mediante una policy di accesso.

> **Important**
>
> Per rendere più sicure le esportazioni verso i bucket S3, ora ti chiediamo di specificare l'elenco degli account di origine autorizzati a esportare i dati di log nel tuo bucket S3. Nell'esempio seguente, l'elenco degli ID account nella `aws:SourceAccount` chiave sarebbe costituito dagli account da cui un utente può esportare i dati di registro nel bucket S3. La chiave `aws:SourceArn` è la risorsa per la quale viene intrapresa l'azione. Puoi limitarla a un gruppo di log specifico o utilizzare un jolly come mostrato in questo esempio.
> Ti consigliamo di includere anche l'ID dell'account in cui è stato creato il bucket S3 per consentire l'esportazione all'interno dello stesso account.

Impostazione delle autorizzazioni su un bucket S3

1. Crea il file `policy.json` e aggiungi la seguente policy di accesso, modificando `amzn-s3-demo-bucket` con il nome del bucket S3 e `Principal` con l'endpoint della regione di esportazione dei dati di log, come `us-east-1`. Utilizza un editor di testo per creare questo file di policy. Non utilizzare la console IAM.

   JSON

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

```
            "Action": "s3:GetBucketAcl",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                      "123456789012",
                      "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:us-east-1:123456789012:log-group:*",
                          "arn:aws:logs:us-east-1:111122223333:log-group:*"
                      ]
              }
            }
        },
        {
            "Sid": "AllowPutObject",
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "123456789012",
                      "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:us-east-1:123456789012:log-group:*",
                          "arn:aws:logs:us-east-1:111122223333:log-group:*"
                      ]
              }
            }
        }
    ]
}
```

2. Impostare la policy appena aggiunta come policy di accesso nel bucket tramite il comando put-bucket-policy. Questa politica consente a CloudWatch Logs di esportare i dati di registro nel bucket S3. Il proprietario del bucket disporrà di autorizzazioni complete su tutti gli oggetti esportati.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://
policy.json
```

> **Warning**
>
> Se al bucket esistente sono già associate una o più politiche, aggiungi le istruzioni per CloudWatch Logs access a quella o più policy. Consigliamo di valutare il set di autorizzazioni risultante per verificare che siano adeguate agli utenti che accederanno al bucket.

(Facoltativo) Esportazione verso un bucket Amazon S3 di destinazione crittografato con SSE-KMS (CLI)

Questa procedura è necessaria solo se si esegue l'esportazione in un bucket S3 che utilizza la crittografia lato server con. AWS KMS keys Questa crittografia è nota come. SSE-KMS

Esportare in un bucket crittografato con SSE-KMS

1. Utilizza un editor di testo per creare un file denominato `key_policy.json` e aggiungi la seguente policy di accesso. Quando aggiungi la policy della chiave, apporta le modifiche seguenti:

   - Sostituisci *`Region`* con la regione dei tuoi log.
   - Sostituisci *`account-ARN`* con l'ARN dell'account che possiede la chiave KMS.

   JSON

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

```
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

2. Immetti il comando seguente:

```
aws kms create-key --policy file://key_policy.json
```

Di seguito è fornito un esempio dell'output di questo comando:

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account-ARN:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
```

```
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. Utilizza un editor di testo per creare un file denominato `bucketencryption.json` con il seguente contenuto.

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Immettete il seguente comando, sostituendolo *amzn-s3-demo-bucket* con il nome del bucket in cui state esportando i log.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-
encryption-configuration file://bucketencryption.json
```

Se il comando non restituisce un errore, il processo ha esito positivo.

Creare un'attività di esportazione (CLI)

Usare il comando seguente per creare l'attività di esportazione. Dopo averla creata, potrebbero essere necessari da pochi secondi ad alcune ore per l'attività di esportazione, in base alla dimensione dei dati da esportare.

Per esportare i dati in Amazon S3 utilizzando AWS CLI

1. Accedi con le autorizzazioni sufficienti come documentato in Configurare le autorizzazioni di accesso (CLI).
2. Al prompt dei comandi, utilizzare il seguente comando create-export-task per creare l'attività di esportazione.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --to 1441494000000 --destination "amzn-s3-demo-bucket" --destination-prefix "export-task-output"
```

Di seguito è riportato un output di esempio.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Cross-account esportazione (CLI)

Se il bucket Amazon S3 si trova in un account diverso da quello dei log che vengono esportati, segui le istruzioni in questa sezione.

Argomenti

- Crea un bucket S3 per l'esportazione tra account (CLI)
- Configurare le autorizzazioni di accesso per l'esportazione tra account (CLI)
- Imposta le autorizzazioni su un bucket S3 per l'esportazione tra account (CLI)
- (Facoltativo) Esportazione verso un bucket Amazon S3 di destinazione crittografato SSE-KMS per l'esportazione tra account (CLI)
- Creare un'attività di esportazione per l'esportazione tra account (CLI)

Crea un bucket S3 per l'esportazione tra account (CLI)

Ti consigliamo di utilizzare un bucket creato appositamente per i registri. CloudWatch Tuttavia, se intendi utilizzare un bucket esistente, puoi passare alla fase 2.

> **Note**
>
> Il bucket S3 deve trovarsi nella stessa Regione dei dati di log da esportare. CloudWatch Logs non supporta l'esportazione di dati in bucket S3 in una regione diversa.

Per creare un bucket S3 utilizzando il AWS CLI

Al prompt dei comandi, eseguire il seguente comando create-bucket, in cui `LocationConstraint` è la regione in cui esportare i dati di log.

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-2
```

Di seguito è riportato un output di esempio.

```
{
    "Location": "/amzn-s3-demo-bucket"
}
```

Configurare le autorizzazioni di accesso per l'esportazione tra account (CLI)

Innanzitutto, devi creare una nuova policy IAM per consentire a CloudWatch Logs di avere l'`s3:PutObject`azione per il bucket Amazon S3 di destinazione nell'account di destinazione.

Oltre all'`s3:PutObject`azione, le azioni aggiuntive incluse nella policy dipendono dal fatto che il bucket di destinazione utilizzi la AWS KMS crittografia o che gli ACL siano abilitati utilizzando l'impostazione S3 Object Ownership.

- Se utilizzi la crittografia KMS, aggiungi le `kms:Decrypt` azioni `kms:GenerateDataKey` and per la risorsa chiave
- Se gli ACL sono abilitati sul bucket, aggiungi l'`s3:PutObjectAcl`azione per la risorsa bucket

Passa `amzn-s3-demo-bucket` al nome del bucket S3 di destinazione nelle seguenti politiche.

La policy creata dipende dal fatto che il bucket di destinazione utilizzi la crittografia AWS KMS . Se non utilizza la AWS KMS crittografia, crea una politica con i seguenti contenuti.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Se il bucket di destinazione utilizza AWS KMS la crittografia, create una policy con i seguenti contenuti.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Se gli ACL sono abilitati sul bucket di destinazione, aggiungi il blocco s3: PutObjectAcl a s3: PutObject Action nelle politiche precedenti.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "s3:PutObject",
               "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Per creare un'attività di esportazione, devi accedere con un ruolo IAM a cui è associata la policy `AmazonS3ReadOnlyAccess` gestita, la policy IAM creata sopra e anche con le seguenti autorizzazioni:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in Centro identità AWS IAM:

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di Centro identità AWS IAM .
- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Create a role for a third-party identity provider (federation) della Guida per l'utente IAM.
- Utenti IAM:

- Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella pagina Create a role for an IAM user della Guida per l'utente IAM.
- (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

Imposta le autorizzazioni su un bucket S3 per l'esportazione tra account (CLI)

Per impostazione predefinita, tutti i bucket e gli oggetti S3 sono privati. Solo il proprietario della risorsa, l'account che ha creato il bucket, può accedere al bucket e agli oggetti in esso contenuti. Tuttavia, il proprietario della risorsa può concedere le autorizzazioni di accesso ad altre risorse e ad altri utenti mediante una policy di accesso.

> **Important**
>
> Per rendere più sicure le esportazioni verso i bucket S3, ora ti chiediamo di specificare l'elenco degli account di origine autorizzati a esportare i dati di log nel tuo bucket S3. Nell'esempio seguente, l'elenco degli ID account nella `aws:SourceAccount` chiave sarebbe costituito dagli account da cui un utente può esportare i dati di registro nel bucket S3. La chiave `aws:SourceArn` è la risorsa per la quale viene intrapresa l'azione. Puoi limitarla a un gruppo di log specifico o utilizzare un jolly come mostrato in questo esempio.
> Ti consigliamo di includere anche l'ID dell'account in cui è stato creato il bucket S3 per consentire l'esportazione all'interno dello stesso account.

Impostazione delle autorizzazioni su un bucket S3

1. Crea un file denominato `policy.json` e aggiungi la seguente politica di accesso, modificando `amzn-s3-demo-bucket` il nome del bucket S3 di destinazione, `Principal` all'endpoint della regione in cui stai esportando i dati di registro, ad esempio. `us-west-1` Utilizza un editor di testo per creare questo file di policy. Non utilizzare la console IAM.

   JSON

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

```
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                "123456789012",
                "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:log-group:*",
                    "arn:aws:logs:us-east-1:111122223333:log-group:*"
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                "123456789012",
                "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:log-group:*",
                    "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
            }
          }
      },
      {
          "Effect": "Allow",
```

```
            "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/role_name"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::>amzn-s3-demo-bucket/*",
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control"
              }
            }
        }
    ]
}
```

2. Impostare la policy appena aggiunta come policy di accesso nel bucket tramite il comando put-bucket-policy. Questa policy consente a CloudWatch Logs di esportare i dati di log nel tuo bucket S3. Il proprietario del bucket disporrà di autorizzazioni complete su tutti gli oggetti esportati.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://
policy.json
```

> **Warning**
>
> Se al bucket esistente sono già associate una o più politiche, aggiungi le istruzioni per CloudWatch Logs access a quella o più policy. Consigliamo di valutare il set di autorizzazioni risultante per verificare che siano adeguate agli utenti che accederanno al bucket.

(Facoltativo) Esportazione verso un bucket Amazon S3 di destinazione crittografato SSE-KMS per l'esportazione tra account (CLI)

Questa procedura è necessaria solo se si esegue l'esportazione in un bucket S3 che utilizza la crittografia lato server con. AWS KMS keys Questa crittografia è nota come. SSE-KMS

Esportare in un bucket crittografato con SSE-KMS

1. Utilizza un editor di testo per creare un file denominato `key_policy.json` e aggiungi la seguente policy di accesso. Quando aggiungi la policy della chiave, apporta le modifiche seguenti:

- Sostituisci *us-east-1* con la regione dei tuoi log.
- Sostituisci *account-ARN* con l'ARN dell'account che possiede la chiave KMS.
- Sostituisci *123456789012* con il numero di account che possiede la chiave KMS.
- *key_id*con l'ID della chiave kms.
- *role_name*con il ruolo utilizzato per creare l'attività di esportazione.

JSON

```
{
"Version":"2012-10-17",
"Statement": [
    {
        "Sid": "AllowCWLServicePrincipalUsage",
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
        },
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "*"
    },
    {
        "Sid": "EnableIAMUserPermissions",
        "Effect": "Allow",
        "Principal": {
            "AWS": "account-ARN"
        },
        "Action": [
            "kms:GetKeyPolicy*",
            "kms:PutKeyPolicy*",
            "kms:DescribeKey*",
            "kms:CreateAlias*",
            "kms:ScheduleKeyDeletion*",
            "kms:Decrypt"
        ],
        "Resource": "*"
```

```
        },
        {
            "Sid": "EnableIAMRolePermissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

2. Immetti il comando seguente:

```
aws kms create-key --policy file://key_policy.json
```

Di seguito è fornito un esempio dell'output di questo comando:

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-1:123456789012:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. Utilizza un editor di testo per creare un file denominato `bucketencryption.json` con il seguente contenuto.

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Immettete il seguente comando, sostituendolo *amzn-s3-demo-bucket* con il nome del bucket in cui state esportando i log.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-
encryption-configuration file://bucketencryption.json
```

   Se il comando non restituisce un errore, il processo ha esito positivo.

Creare un'attività di esportazione per l'esportazione tra account (CLI)

Usare il comando seguente per creare l'attività di esportazione. Dopo averla creata, potrebbero essere necessari da pochi secondi ad alcune ore per l'attività di esportazione, in base alla dimensione dei dati da esportare.

Per esportare i dati in Amazon S3 utilizzando AWS CLI

1. Accedi con le autorizzazioni sufficienti come documentato in Configurare le autorizzazioni di accesso (CLI).
2. Al prompt dei comandi, utilizzare il seguente comando create-export-task per creare l'attività di esportazione.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-
group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --
```

```
to 1441494000000 --destination "amzn-s3-demo-bucket" --destination-prefix "export-
task-output"
```

Di seguito è riportato un output di esempio.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Descrivere le attività di esportazione (CLI)

Dopo aver creato un'attività di esportazione, puoi ottenere lo stato corrente dell'attività.

Per descrivere le attività di esportazione utilizzando AWS CLI

Al prompt dei comandi, utilizzare il seguente comando describe-export-tasks.

```
aws logs --profile CWLExportUser describe-export-tasks --task-id
 "cda45419-90ea-4db5-9833-aade86253e66"
```

Di seguito è riportato un output di esempio.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "RUNNING",
         "message": "Started Successfully"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "tTo": 1441494000000
   }]
```

```
}
```

Puoi utilizzare il comando `describe-export-tasks` in tre diversi modi:

- Senza filtri: elenca tutte le attività di esportazione, in ordine inverso di creazione.
- Filtro su ID attività: elenca l'attività di esportazione, se esistente, con l'ID specificato.
- Filtro su stato dell'attività: elenca le attività di esportazione con lo stato specificato.

Ad esempio, utilizzare il seguente comando per applicare un filtro allo stato `FAILED`.

```
aws logs --profile CWLExportUser describe-export-tasks --status-code "FAILED"
```

Di seguito è riportato un output di esempio.

```
{
   "exportTasks": [
   {
      "destination": "amzn-s3-demo-bucket",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "completionTime": 1441498600000
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "FAILED",
         "message": "FAILED"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "to": 1441494000000
   }]
}
```

Annullare un'attività di esportazione (CLI)

Puoi annullare un'attività di esportazione se il relativo stato è `PENDING` o `RUNNING`.

Per annullare un'operazione di esportazione utilizzando il AWS CLI

Al prompt dei comandi, utilizza il seguente comando cancel-export-task:

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

Puoi utilizzare il comando describe-export-tasks per verificare che l'attività sia stata annullata.

Streaming CloudWatch registra i dati su Amazon Service OpenSearch

Puoi configurare un gruppo di log in Amazon CloudWatch Logs, in modo da trasmettere i dati al tuo cluster Amazon OpenSearch Service quasi in tempo reale. Per ulteriori informazioni, consulta Real-time trattamento dei dati di registro con abbonamenti.

> **Note**
>
> Lo streaming su OpenSearch Service è supportato solo per i gruppi di log nella classe di log Standard. Per ulteriori informazioni sulle classi di log, consultaClassi di registro.

A seconda della quantità di dati di log trasmessi in streaming, valuta la possibilità di impostare un limite di concorrenza a livello di funzione. Per ulteriori informazioni, consulta Lambda function scaling (Dimensionamento della funzione Lambda).

> **Note**
>
> Poiché lo streaming di grandi quantità di dati di CloudWatch log su OpenSearch Service potrebbe comportare costi di utilizzo elevati, ti consigliamo di creare un budget nella console. Gestione costi e fatturazione AWS Per ulteriori informazioni, consulta Gestire i costi con i AWS budget.

Questa sezione descrive i prerequisiti da completare prima di sottoscrivere un gruppo di log al Servizio. OpenSearch Descrive inoltre come sottoscrivere un gruppo di log al OpenSearch Servizio.

Prerequisiti

Prima di iniziare, crea un dominio OpenSearch di servizio. Il dominio può avere accesso pubblico o accesso VPC, ma non puoi modificare il tipo di accesso dopo aver creato il dominio. Potresti voler rivedere le impostazioni del dominio di OpenSearch servizio in un secondo momento e modificare la configurazione del cluster in base alla quantità di dati che il cluster elaborerà. Per istruzioni su come creare un dominio, consulta Creazione di domini OpenSearch di servizio.

Per ulteriori informazioni sul OpenSearch servizio, consulta l'Amazon OpenSearch Service Developer Guide.

Sottoscrivi un gruppo di log a OpenSearch Service

È possibile utilizzare la CloudWatch console per sottoscrivere un gruppo di log al OpenSearch Servizio.

Per iscrivere un gruppo di log al OpenSearch Servizio

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Selezionare il nome del gruppo di registro.
4. Scegli Azioni, Filtri di abbonamento, Crea filtro OpenSearch di abbonamento Amazon Service.
5. Scegli se eseguire lo streaming in un cluster di questo account o di un altro account.
   - Se hai scelto questo account, selezionare il dominio creato nella fase precedente.
   - Se hai scelto un altro account, fornire il dominio ARN e l'endpoint.
6. Per Lambda IAM Execution Role, scegli il ruolo IAM che Lambda deve utilizzare per eseguire le chiamate. OpenSearch

   Il ruolo IAM scelto deve soddisfare i seguenti requisiti:

   - Deve avere `lambda.amazonaws.com` nella relazione di trust.
   - Deve includere la policy seguente:

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOpenSearchStreamingAccess",
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:us-
east-1:123456789012:domain/cloudwatch-logs/*"
```

```
            }
        ]
}
```

- Se il dominio OpenSearch di servizio di destinazione utilizza l'accesso VPC, al ruolo deve essere associata la AWSLambdaVPCAccessExecutionRolepolicy. Questa politica gestita da Amazon concede l'accesso Lambda al VPC del cliente, consentendo a Lambda di scrivere sull'endpoint nel VPC. OpenSearch

7. Per Log format (Formato log) scegli un formato per il registro.
8. In Subscription filter pattern (Modello del filtro sottoscrizioni) digita i termini o i modelli da individuare nei tuoi registri eventi. Ciò garantisce l'invio al cluster solo dei dati che ti interessano. OpenSearch Per ulteriori informazioni, consulta Creazione di parametri da log eventi mediante filtri.
9. (Opzionale) In Select log data to test (Seleziona i dati di registro per il test), seleziona un flusso di registro, quindi scegli Test pattern (Modello di test) per verificare che il filtro di ricerca restituisca i risultati previsti.
10. Quindi scegli Start streaming (Avvia streaming).

Esempi di codice per l'utilizzo di CloudWatch Logs AWS SDK

I seguenti esempi di codice mostrano come utilizzare CloudWatch Logs con un kit di sviluppo AWS software (SDK).

Le azioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica chiamando più funzioni all'interno dello stesso servizio o combinate con altri Servizi AWS.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di codice

- Esempi di base per l'utilizzo dei CloudWatch log AWS SDK
  - CloudWatch Ciao Logs
  - Azioni per l'utilizzo CloudWatch dei log AWS SDK
    - Utilizza AssociateKmsKeycon un AWS SDK
    - Utilizzare CancelExportTaskcon un AWS SDK
    - Utilizzare CreateExportTaskcon un AWS SDK
    - Utilizzare CreateLogGroupcon un AWS SDK o CLI
    - Utilizzare CreateLogStreamcon un AWS SDK o CLI
    - Utilizzare DeleteLogGroupcon un AWS SDK o CLI
    - Utilizzare DeleteSubscriptionFiltercon un AWS SDK
    - Utilizzare DeleteSyslogConfigurationcon un AWS SDK
    - Utilizzare DescribeExportTaskscon un AWS SDK
    - Utilizzare DescribeLogGroupscon un AWS SDK o CLI
    - Utilizzare DescribeLogStreamscon un AWS SDK o CLI
    - Utilizzare DescribeSubscriptionFilterscon un AWS SDK
    - Utilizzare GetLogEventscon un AWS SDK o CLI

    - Utilizzare GetQueryResultscon un AWS SDK
    - Utilizzare ListSyslogConfigurationscon un AWS SDK
    - Utilizzare PutResourcePolicycon un AWS SDK
    - Utilizzare PutSubscriptionFiltercon un AWS SDK
    - Utilizzare PutSyslogConfigurationcon un AWS SDK
    - Utilizzare StartLiveTailcon un AWS SDK o CLI
    - Utilizzare StartQuerycon un AWS SDK
- Scenari per l'utilizzo CloudWatch dei log AWS SDK
    - Configura la connettività del servizio container
    - Creazione della tua prima funzione serverless
    - Scopri le nozioni di base sull'ingestione di CloudWatch Logs syslog utilizzando un AWS SDK
    - Usa CloudWatch Logs per eseguire una query di grandi dimensioni
    - Utilizzo degli eventi pianificati per invocare una funzione Lambda

Esempi di base per l'utilizzo dei CloudWatch log AWS SDK

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Amazon CloudWatch Logs con AWS SDK.

Esempi

- CloudWatch Ciao Logs
- Azioni per l'utilizzo CloudWatch dei log AWS SDK
    - Utilizza AssociateKmsKeycon un AWS SDK
    - Utilizzare CancelExportTaskcon un AWS SDK
    - Utilizzare CreateExportTaskcon un AWS SDK
    - Utilizzare CreateLogGroupcon un AWS SDK o CLI
    - Utilizzare CreateLogStreamcon un AWS SDK o CLI
    - Utilizzare DeleteLogGroupcon un AWS SDK o CLI
    - Utilizzare DeleteSubscriptionFiltercon un AWS SDK
    - Utilizzare DeleteSyslogConfigurationcon un AWS SDK
    - Utilizzare DescribeExportTaskscon un AWS SDK

- Utilizzare DescribeLogGroupscon un AWS SDK o CLI
- Utilizzare DescribeLogStreamscon un AWS SDK o CLI
- Utilizzare DescribeSubscriptionFilterscon un AWS SDK
- Utilizzare GetLogEventscon un AWS SDK o CLI
- Utilizzare GetQueryResultscon un AWS SDK
- Utilizzare ListSyslogConfigurationscon un AWS SDK
- Utilizzare PutResourcePolicycon un AWS SDK
- Utilizzare PutSubscriptionFiltercon un AWS SDK
- Utilizzare PutSyslogConfigurationcon un AWS SDK
- Utilizzare StartLiveTailcon un AWS SDK o CLI
- Utilizzare StartQuerycon un AWS SDK

CloudWatch Ciao Logs

Il seguente esempio di codice mostra come iniziare a utilizzare CloudWatch Logs.

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 hello_cloudwatch_logs() -> None:
    """
    Verifies connectivity to CloudWatch Logs by calling DescribeLogGroups
    and displaying the first page of log groups with name, ARN, and
    creation time.
    """
    logs_client = boto3.client("logs")

    try:
        response = logs_client.describe_log_groups()
        log_groups = response.get("logGroups", list())
```

```
        if not log_groups:
            print("No log groups found in this Region for your account.")
        else:
            print(f"Found {len(log_groups)} log group(s):\n")
            for lg in log_groups:
                name = lg.get("logGroupName", "N/A")
                arn = lg.get("arn", "N/A")
                creation_ms = lg.get("creationTime", 0)
                creation_dt = datetime.fromtimestamp(
                    creation_ms / 1000, tz=timezone.utc
                )
                print(f"  Name: {name}")
                print(f"  ARN:  {arn}")
                print(f"  Created: {creation_dt.strftime('%Y-%m-%d %H:%M:%S
UTC')}")
                print()

    except ClientError as error:
        logger.error(
            "Failed to describe log groups: %s",
            error.response["Error"]["Message"],
        )
        raise
```

- Per i dettagli sull'API, consulta DescribeLogGroups AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Azioni per l'utilizzo CloudWatch dei log AWS SDK

I seguenti esempi di codice mostrano come eseguire singole azioni di CloudWatch Logs con gli SDK. AWS Ogni esempio include un collegamento a GitHub, dove è possibile trovare le istruzioni per la configurazione e l'esecuzione del codice.

Questi estratti richiamano l'API CloudWatch Logs e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. È possibile visualizzare le azioni nel contesto in Scenari per l'utilizzo CloudWatch dei log AWS SDK.

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per un elenco completo, consulta l'Amazon CloudWatch Logs API Reference.

Esempi

- Utilizza AssociateKmsKeycon un AWS SDK
- Utilizzare CancelExportTaskcon un AWS SDK
- Utilizzare CreateExportTaskcon un AWS SDK
- Utilizzare CreateLogGroupcon un AWS SDK o CLI
- Utilizzare CreateLogStreamcon un AWS SDK o CLI
- Utilizzare DeleteLogGroupcon un AWS SDK o CLI
- Utilizzare DeleteSubscriptionFiltercon un AWS SDK
- Utilizzare DeleteSyslogConfigurationcon un AWS SDK
- Utilizzare DescribeExportTaskscon un AWS SDK
- Utilizzare DescribeLogGroupscon un AWS SDK o CLI
- Utilizzare DescribeLogStreamscon un AWS SDK o CLI
- Utilizzare DescribeSubscriptionFilterscon un AWS SDK
- Utilizzare GetLogEventscon un AWS SDK o CLI
- Utilizzare GetQueryResultscon un AWS SDK
- Utilizzare ListSyslogConfigurationscon un AWS SDK
- Utilizzare PutResourcePolicycon un AWS SDK
- Utilizzare PutSubscriptionFiltercon un AWS SDK
- Utilizzare PutSyslogConfigurationcon un AWS SDK
- Utilizzare StartLiveTailcon un AWS SDK o CLI
- Utilizzare StartQuerycon un AWS SDK

Utilizza **`AssociateKmsKey`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `AssociateKmsKey`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to associate an AWS Key Management Service (AWS KMS) key with
    /// an Amazon CloudWatch Logs log group.
    /// </summary>
    public class AssociateKmsKey
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string kmsKeyId = "arn:aws:kms:us-west-2:<account-
number>:key/7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
            string groupName = "cloudwatchlogs-example-loggroup";

            var request = new AssociateKmsKeyRequest
            {
                KmsKeyId = kmsKeyId,
                LogGroupName = groupName,
            };

            var response = await client.AssociateKmsKeyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
```

```
            {
                Console.WriteLine($"Successfully associated KMS key ID:
{kmsKeyId} with log group: {groupName}.");
            }
            else
            {
                Console.WriteLine("Could not make the association between:
{kmsKeyId} and {groupName}.");
            }
        }
    }
```

- Per i dettagli sull'API, consulta la AssociateKmsKeysezione AWS SDK per .NET API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CancelExportTask`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `CancelExportTask`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

```
/// <summary>
/// Shows how to cancel an Amazon CloudWatch Logs export task.
/// </summary>
public class CancelExportTask
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();
        string taskId = "exampleTaskId";

        var request = new CancelExportTaskRequest
        {
            TaskId = taskId,
        };

        var response = await client.CancelExportTaskAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"{taskId} successfully canceled.");
        }
        else
        {
            Console.WriteLine($"{taskId} could not be canceled.");
        }
    }
}
```

- Per i dettagli sull'API, consulta la CancelExportTasksezione AWS SDK per .NET API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateExportTask`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `CreateExportTask`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Export Task to export the contents of the Amazon
    /// CloudWatch Logs to the specified Amazon Simple Storage Service (Amazon
S3)
    /// bucket.
    /// </summary>
    public class CreateExportTask
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string taskName = "export-task-example";
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string destination = "amzn-s3-demo-bucket";
            var fromTime = 1437584472382;
            var toTime = 1437584472833;

            var request = new CreateExportTaskRequest
            {
```

```
                From = fromTime,
                To = toTime,
                TaskName = taskName,
                LogGroupName = logGroupName,
                Destination = destination,
            };

            var response = await client.CreateExportTaskAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"The task, {taskName} with ID: " +
                                  $"{response.TaskId} has been created
successfully.");
            }
        }
    }
```

- Per i dettagli sull'API, consulta la CreateExportTasksezione AWS SDK per .NET API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateLogGroup`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateLogGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Configura la connettività dei servizi container
- Scopri le nozioni di base sull'ingestione di syslog

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Amazon CloudWatch Logs log group.
    /// </summary>
    public class CreateLogGroup
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string logGroupName = "cloudwatchlogs-example-loggroup";

            var request = new CreateLogGroupRequest
            {
                LogGroupName = logGroupName,
            };

            var response = await client.CreateLogGroupAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully create log group with ID:
{logGroupName}.");
            }
```

```
            else
            {
                Console.WriteLine("Could not create log group.");
            }
        }
    }
```

- Per i dettagli sull'API, consulta la CreateLogGroupsezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Il comando seguente crea un gruppo di log denominato `my-logs`.

```
aws logs create-log-group --log-group-name my-logs
```

- Per i dettagli sull'API, consulta CreateLogGroup AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { CreateLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new CreateLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });
```

```
  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Per i dettagli sull'API, consulta la CreateLogGroupsezione AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`CreateLogStream`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateLogStream`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to create an Amazon CloudWatch Logs stream for a CloudWatch
/// log group.
```

```
    /// </summary>
    public class CreateLogStream
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string logStreamName = "cloudwatchlogs-example-logstream";

            var request = new CreateLogStreamRequest
            {
                LogGroupName = logGroupName,
                LogStreamName = logStreamName,
            };

            var response = await client.CreateLogStreamAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"{logStreamName} successfully created for
{logGroupName}.");
            }
            else
            {
                Console.WriteLine("Could not create stream.");
            }
        }
    }
```

- Per i dettagli sull'API, consulta la CreateLogStreamsezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Il comando seguente crea un flusso di log denominato `20150601` nel gruppo di log `my-logs`.

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- Per i dettagli sull'API, consulta CreateLogStream AWS CLICommand Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DeleteLogGroup`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteLogGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Configura la connettività dei servizi container
- Creazione della tua prima funzione serverless
- Scopri le nozioni di base sull'ingestione di syslog

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Uses the Amazon CloudWatch Logs Service to delete an existing
/// CloudWatch Logs log group.
/// </summary>
```

```
    public class DeleteLogGroup
    {
        public static async Task Main()
        {
            var client = new AmazonCloudWatchLogsClient();
            string logGroupName = "cloudwatchlogs-example-loggroup";

            var request = new DeleteLogGroupRequest
            {
                LogGroupName = logGroupName,
            };

            var response = await client.DeleteLogGroupAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully deleted CloudWatch log group,
{logGroupName}.");
            }
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteLogGroupsezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Il comando seguente elimina un gruppo di log denominato `my-logs`.

```
aws logs delete-log-group --log-group-name my-logs
```

- Per i dettagli sull'API, consulta DeleteLogGroup AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import { DeleteLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Per i dettagli sull'API, consulta la DeleteLogGroupsezione AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DeleteSubscriptionFilter`**con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`DeleteSubscriptionFilter`.

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.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DeleteSubscriptionFilterRequest.h>
#include <iostream>
```

Eliminare il filtro di sottoscrizione.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DeleteSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetLogGroupName(log_group);

        auto outcome = cwl.DeleteSubscriptionFilter(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Failed to delete CloudWatch log subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        } else {
            std::cout << "Successfully deleted CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Per i dettagli sull'API, consulta la DeleteSubscriptionFiltersezione AWS SDK per C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DeleteSubscriptionFilterRequest;

/**
 * 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 DeleteSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <logGroup>

                Where:
                  filter - The name of the subscription filter (for example,
 MyFilter).
                  logGroup - The name of the log group. (for example, testgroup).
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String filter = args[0];
        String logGroup = args[1];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .build();

        deleteSubFilter(logs, filter, logGroup);
        logs.close();
    }

    public static void deleteSubFilter(CloudWatchLogsClient logs, String filter,
 String logGroup) {
        try {
            DeleteSubscriptionFilterRequest request =
 DeleteSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .logGroupName(logGroup)
                    .build();

            logs.deleteSubscriptionFilter(request);
            System.out.printf("Successfully deleted CloudWatch logs subscription
 filter %s", filter);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DeleteSubscriptionFiltersezione 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 { DeleteSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteSubscriptionFilterCommand({
    // The name of the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Per i dettagli sull'API, consulta la DeleteSubscriptionFiltersezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
```

```
  filterName: "FILTER",
  logGroupName: "LOG_GROUP",
};

cwl.deleteSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la DeleteSubscriptionFiltersezione 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 deleteSubFilter(
    filter: String?,
    logGroup: String?,
) {
    val request =
        DeleteSubscriptionFilterRequest {
            filterName = filter
            logGroupName = logGroup
        }

    CloudWatchLogsClient.fromEnvironment { region = "us-west-2" }.use { logs ->
        logs.deleteSubscriptionFilter(request)
        println("Successfully deleted CloudWatch logs subscription filter named
 $filter")
```

```
    }
}
```

- Per i dettagli sull'API, DeleteSubscriptionFilterconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DeleteSyslogConfiguration`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `DeleteSyslogConfiguration`.

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:

- Scopri le nozioni di base sull'ingestione di syslog

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    def delete_syslog_configuration(
        self, log_group_identifier: str, vpc_endpoint_id: str
    ) -> None:
        """
        Deletes a syslog configuration for a log group. After deletion, syslog
        data is no longer ingested through the specified VPC endpoint.

        :param log_group_identifier: The name or ARN of the log group.
        :param vpc_endpoint_id: The ID of the VPC endpoint associated with the
            syslog configuration.
        :raises ClientError: If the syslog configuration does not exist (in which
```

```
            case the error is logged but not re-raised during cleanup).
        """
        try:
            self.logs_client.delete_syslog_configuration(
                logGroupIdentifier=log_group_identifier,
                vpcEndpointId=vpc_endpoint_id,
            )
            logger.info(
                "Deleted syslog configuration for log group '%s' and VPC endpoint
'%s'.",
                log_group_identifier,
                vpc_endpoint_id,
            )
        except ClientError as error:
            if error.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.info(
                    "Syslog configuration does not exist or was already deleted:
%s",
                    error.response["Error"]["Message"],
                )
            else:
                logger.error(
                    "Failed to delete syslog configuration: %s",
                    error.response["Error"]["Message"],
                )
                raise
```

- Per i dettagli sull'API, consulta DeleteSyslogConfiguration AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DescribeExportTasks`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `DescribeExportTasks`.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to retrieve a list of information about Amazon CloudWatch
/// Logs export tasks.
/// </summary>
public class DescribeExportTasks
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();

        var request = new DescribeExportTasksRequest
        {
            Limit = 5,
        };

        var response = new DescribeExportTasksResponse();

        do
        {
            response = await client.DescribeExportTasksAsync(request);
            response.ExportTasks.ForEach(t =>
            {
```

```
                    Console.WriteLine($"{t.TaskName} with ID: {t.TaskId} has
status: {t.Status}");
                });
            }
            while (response.NextToken is not null);
        }
    }
```

- Per i dettagli sull'API, consulta la DescribeExportTaskssezione AWS SDK per .NET API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DescribeLogGroups`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeLogGroups`.

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:

- Scopri le nozioni di base sull'ingestione di syslog

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

```
    /// <summary>
    /// Retrieves information about existing Amazon CloudWatch Logs log groups
    /// and displays the information on the console.
    /// </summary>
    public class DescribeLogGroups
    {
        public static async Task Main()
        {
            // Creates a CloudWatch Logs client using the default
            // user. If you need to work with resources in another
            // AWS Region than the one defined for the default user,
            // pass the AWS Region as a parameter to the client constructor.
            var client = new AmazonCloudWatchLogsClient();

            bool done = false;
            string newToken = null;

            var request = new DescribeLogGroupsRequest
            {
                Limit = 5,
            };

            DescribeLogGroupsResponse response;

            do
            {
                if (newToken is not null)
                {
                    request.NextToken = newToken;
                }

                response = await client.DescribeLogGroupsAsync(request);

                response.LogGroups.ForEach(lg =>
                {
                    Console.WriteLine($"{lg.LogGroupName} is associated with the
key: {lg.KmsKeyId}.");
                    Console.WriteLine($"Created on:
{lg.CreationTime.Date.Date}");
                    Console.WriteLine($"Date for this group will be stored for:
{lg.RetentionInDays} days.\n");
                });
```

```
                if (response.NextToken is null)
                {
                    done = true;
                }
                else
                {
                    newToken = response.NextToken;
                }
            }
            while (!done);
        }
    }
```

- Per i dettagli sull'API, consulta la DescribeLogGroupssezione AWS SDK per .NET API Reference.

CLI

AWS CLI

Il comando seguente descrive un gruppo di log denominato `my-logs`.

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

Output:

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- Per i dettagli sull'API, consulta DescribeLogGroups AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
import {
  paginateDescribeLogGroups,
  CloudWatchLogsClient,
} from "@aws-sdk/client-cloudwatch-logs";

const client = new CloudWatchLogsClient({});

export const main = async () => {
  const paginatedLogGroups = paginateDescribeLogGroups({ client }, {});
  const logGroups = [];

  for await (const page of paginatedLogGroups) {
    if (page.logGroups?.every((lg) => !!lg)) {
      logGroups.push(...page.logGroups);
    }
  }

  console.log(logGroups);
  return logGroups;
};
```

- Per i dettagli sull'API, consulta la DescribeLogGroupssezione AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DescribeLogStreams`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeLogStreams`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione della tua prima funzione serverless

CLI

AWS CLI

Il comando seguente mostra tutti i flussi di log che iniziano con il prefisso `2015` nel gruppo di log `my-logs`.

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-
prefix 2015
```

Output:

```
{
    "logStreams": [
        {
            "creationTime": 1433189871774,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150531",
            "logStreamName": "20150531",
            "storedBytes": 0
        },
        {
            "creationTime": 1433189873898,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150601",
            "logStreamName": "20150601",
            "storedBytes": 0
        }
    ]
}
```

- Per i dettagli sull'API, consulta DescribeLogStreams 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.

Cerca i flussi di log all'interno di un gruppo di log specificato che corrispondono a un determinato prefisso.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CloudWatchLogsSearch {

    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <logGroupName> <logStreamName>

                Where:
                  logGroupName - The name of the log group (for example,
 WeathertopJavaContainerLogs).
                  logStreamName - The name of the log stream (for example,
 weathertop-java-stream).
                  pattern - the pattern to use (for example, INFO)

                """;

        if (args.length != 3) {
            System.out.print(usage);
```

```
            System.exit(1);
        }

        String logGroupName = args[0] ;
        String logStreamName = args[1] ;
        String pattern = args[2] ;

        CloudWatchLogsClient cwlClient = CloudWatchLogsClient.builder()
                .region(Region.US_EAST_1)
                .build();

        searchLogStreamsAndFilterEvents(cwlClient, logGroupName, logStreamName,
pattern);
    }

    /**
     * Searches for log streams with a specific prefix within a log group and
filters log events based on a specified pattern.
     *
     * @param cwlClient       the CloudWatchLogsClient used to interact with AWS
CloudWatch Logs
     * @param logGroupName    the name of the log group to search within
     * @param logStreamPrefix the prefix of the log streams to search for
     * @param pattern         the pattern to filter log events by
     */
    public static void searchLogStreamsAndFilterEvents(CloudWatchLogsClient
cwlClient, String logGroupName, String logStreamPrefix, String pattern) {
        DescribeLogStreamsRequest describeLogStreamsRequest =
DescribeLogStreamsRequest.builder()
                .logGroupName(logGroupName)
                .logStreamNamePrefix(logStreamPrefix)
                .build();

        DescribeLogStreamsResponse describeLogStreamsResponse =
cwlClient.describeLogStreams(describeLogStreamsRequest);
        List<LogStream> logStreams = describeLogStreamsResponse.logStreams();

        for (LogStream logStream : logStreams) {
            String logStreamName = logStream.logStreamName();
            System.out.println("Searching in log stream: " + logStreamName);

            FilterLogEventsRequest filterLogEventsRequest =
FilterLogEventsRequest.builder()
                    .logGroupName(logGroupName)
```

```
                    .logStreamNames(logStreamName)
                    .filterPattern(pattern)
                    .build();

            FilterLogEventsResponse filterLogEventsResponse =
 cwlClient.filterLogEvents(filterLogEventsRequest);

            for (FilteredLogEvent event : filterLogEventsResponse.events()) {
                System.out.println(event.message());
            }


 System.out.println("-------------------------------------------------"); //
 Separator for better readability
        }
    }
}
```

Stampa i metadati relativi al flusso di log più recente in un gruppo di log specificato.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CloudWatchLogQuery {
    public static void main(final String[] args) {
        final String usage = """
                Usage:
                  <logGroupName>

                Where:
                  logGroupName - The name of the log group (for example, /aws/
lambda/ChatAIHandler).
                """;

        if (args.length != 1) {
```

```
            System.out.print(usage);
            System.exit(1);
        }

        String logGroupName = "/aws/lambda/ChatAIHandler" ; //args[0];
        CloudWatchLogsClient logsClient = CloudWatchLogsClient.builder()
                .region(Region.US_EAST_1)
                .build();

        describeMostRecentLogStream(logsClient, logGroupName);
    }

    /**
     * Describes and prints metadata about the most recent log stream in the
specified log group.
     *
     * @param logsClient   the CloudWatchLogsClient used to interact with AWS
CloudWatch Logs
     * @param logGroupName the name of the log group
     */
    public static void describeMostRecentLogStream(CloudWatchLogsClient
logsClient, String logGroupName) {
        DescribeLogStreamsRequest streamsRequest =
DescribeLogStreamsRequest.builder()
                .logGroupName(logGroupName)
                .orderBy(OrderBy.LAST_EVENT_TIME)
                .descending(true)
                .limit(1)
                .build();

        try {
            DescribeLogStreamsResponse streamsResponse =
logsClient.describeLogStreams(streamsRequest);
            List<LogStream> logStreams = streamsResponse.logStreams();

            if (logStreams.isEmpty()) {
                System.out.println("No log streams found for log group: " +
logGroupName);
                return;
            }

            LogStream stream = logStreams.get(0);
            System.out.println("Most Recent Log Stream:");
            System.out.println("  Name: " + stream.logStreamName());
```

```
            System.out.println("  ARN: " + stream.arn());
            System.out.println("  Creation Time: " + stream.creationTime());
            System.out.println("  First Event Time: " +
 stream.firstEventTimestamp());
            System.out.println("  Last Event Time: " +
 stream.lastEventTimestamp());
            System.out.println("  Stored Bytes: " + stream.storedBytes());
            System.out.println("  Upload Sequence Token: " +
 stream.uploadSequenceToken());

        } catch (CloudWatchLogsException e) {
            System.err.println("Failed to describe log stream: " +
 e.awsErrorDetails().errorMessage());
        }
    }
}
```

- Per i dettagli sull'API, consulta la sezione DescribeLogStreams AWS SDK for Java 2.xAPI Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`DescribeSubscriptionFilters`**con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`DescribeSubscriptionFilters`.

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.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DescribeSubscriptionFiltersRequest.h>
#include <aws/logs/model/DescribeSubscriptionFiltersResult.h>
#include <iostream>
#include <iomanip>
```

Elencare i filtri di sottoscrizione.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DescribeSubscriptionFiltersRequest request;
        request.SetLogGroupName(log_group);
        request.SetLimit(1);

        bool done = false;
        bool header = false;
        while (!done) {
            auto outcome = cwl.DescribeSubscriptionFilters(
                    request);
            if (!outcome.IsSuccess()) {
                std::cout << "Failed to describe CloudWatch subscription filters
 "
                    << "for log group " << log_group << ": " <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header) {
                std::cout << std::left << std::setw(32) << "Name" <<
                    std::setw(64) << "FilterPattern" << std::setw(64) <<
                    "DestinationArn" << std::endl;
                header = true;
            }

            const auto &filters = outcome.GetResult().GetSubscriptionFilters();
            for (const auto &filter : filters) {
                std::cout << std::left << std::setw(32) <<
                    filter.GetFilterName() << std::setw(64) <<
                    filter.GetFilterPattern() << std::setw(64) <<
                    filter.GetDestinationArn() << std::endl;
            }
```

```
            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Per i dettagli sull'API, consulta la DescribeSubscriptionFilterssezione AWS SDK per C++ 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.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersResponse
import software.amazon.awssdk.services.cloudwatchlogs.model.SubscriptionFilter;

/**
 * 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 DescribeSubscriptionFilters {
    public static void main(String[] args) {
```

```
        final String usage = """

                Usage:
                  <logGroup>

                Where:
                  logGroup - A log group name (for example, myloggroup).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String logGroup = args[0];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        describeFilters(logs, logGroup);
        logs.close();
    }

    public static void describeFilters(CloudWatchLogsClient logs, String
logGroup) {
        try {
            boolean done = false;
            String newToken = null;

            while (!done) {
                DescribeSubscriptionFiltersResponse response;
                if (newToken == null) {
                    DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                            .logGroupName(logGroup)
                            .limit(1).build();

                    response = logs.describeSubscriptionFilters(request);
                } else {
                    DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                            .nextToken(newToken)
                            .logGroupName(logGroup)
                            .limit(1).build();
```

```
                    response = logs.describeSubscriptionFilters(request);
                }

                for (SubscriptionFilter filter : response.subscriptionFilters())
 {
                    System.out.printf("Retrieved filter with name %s, " +
 "pattern %s " + "and destination arn %s",
                            filter.filterName(),
                            filter.filterPattern(),
                            filter.destinationArn());
                }

                if (response.nextToken() == null) {
                    done = true;
                } else {
                    newToken = response.nextToken();
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.printf("Done");
    }
}
```

- Per i dettagli sull'API, consulta la DescribeSubscriptionFilterssezione 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 { DescribeSubscriptionFiltersCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  // This will return a list of all subscription filters in your account
  // matching the log group name.
  const command = new DescribeSubscriptionFiltersCommand({
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
    limit: 1,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Per i dettagli sull'API, consulta la DescribeSubscriptionFilterssezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
```

```
  logGroupName: "GROUP_NAME",
  limit: 5,
};

cwl.describeSubscriptionFilters(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.subscriptionFilters);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la DescribeSubscriptionFilterssezione 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 describeFilters(logGroup: String) {
    val request =
        DescribeSubscriptionFiltersRequest {
            logGroupName = logGroup
            limit = 1
        }

    CloudWatchLogsClient.fromEnvironment { region = "us-west-2" }.use { cwlClient
 ->
        val response = cwlClient.describeSubscriptionFilters(request)
        response.subscriptionFilters?.forEach { filter ->
            println("Retrieved filter with name  ${filter.filterName} pattern
 ${filter.filterPattern} and destination ${filter.destinationArn}")
        }
```

```
    }
}
```

- Per i dettagli sull'API, DescribeSubscriptionFiltersconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`GetLogEvents`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`GetLogEvents`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Creazione della tua prima funzione serverless

CLI

AWS CLI

Il comando seguente recupera gli eventi del log da un flusso di log denominato `20150601` nel gruppo di log `my-logs`.

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

Output:

```
{
    "nextForwardToken":
 "f/31961209122447488583055879464742346735121166569214640130",
    "events": [
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
```

```
            "ingestionTime": 1433190516679,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184358,
            "message": "Example Event 2"
        }
    ],
    "nextBackwardToken":
 "b/31961209122358285602261756944988674324553373268216709120"
}
```

- Per i dettagli sull'API, consulta GetLogEvents 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.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsResponse;
import software.amazon.awssdk.services.cloudwatchlogs.model.GetLogEventsRequest;
import software.amazon.awssdk.services.cloudwatchlogs.model.GetLogEventsResponse;

import java.time.Instant;
import java.time.temporal.ChronoUnit;

/**
 * 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 GetLogEvents {

    public static void main(String[] args) {

        final String usage = """

                Usage:
                  <logGroupName> <logStreamName>

                Where:
                  logGroupName - The name of the log group (for example,
 myloggroup).
                  logStreamName - The name of the log stream (for example,
 mystream).

                """;

       // if (args.length != 2) {
       //     System.out.print(usage);
       //     System.exit(1);
//        }

        String logGroupName = "WeathertopJavaContainerLogs" ; //args[0];
        String logStreamName = "weathertop-java-stream" ; //args[1];

        Region region = Region.US_EAST_1 ;
        CloudWatchLogsClient cloudWatchLogsClient =
 CloudWatchLogsClient.builder()
                .region(region)
                .build();

        getCWLogEvents(cloudWatchLogsClient, logGroupName, logStreamName);
        cloudWatchLogsClient.close();
    }

    public static void getCWLogEvents(CloudWatchLogsClient cloudWatchLogsClient,
                                      String logGroupName,
                                      String logStreamPrefix) {
```

```
        try {
            // First, find the exact log stream name
            DescribeLogStreamsRequest describeRequest =
DescribeLogStreamsRequest.builder()
                    .logGroupName(logGroupName)
                    .logStreamNamePrefix(logStreamPrefix)
                    .limit(1) // get the first matching stream
                    .build();

            DescribeLogStreamsResponse describeResponse =
cloudWatchLogsClient.describeLogStreams(describeRequest);

            if (describeResponse.logStreams().isEmpty()) {
                System.out.println("No matching log streams found for prefix: " +
logStreamPrefix);
                return;
            }

            String exactLogStreamName =
describeResponse.logStreams().get(0).logStreamName();
            System.out.println("Using exact log stream: " + exactLogStreamName);

            long startTime = Instant.now().minus(7,
ChronoUnit.DAYS).toEpochMilli();
            long endTime = Instant.now().toEpochMilli();

            GetLogEventsRequest getLogEventsRequest =
GetLogEventsRequest.builder()
                    .logGroupName(logGroupName)
                    .logStreamName(exactLogStreamName) // <-- exact name, not
prefix
                    .startTime(startTime)
                    .endTime(endTime)
                    .startFromHead(true)
                    .build();

            GetLogEventsResponse response =
cloudWatchLogsClient.getLogEvents(getLogEventsRequest);

            if (response.events().isEmpty()) {
                System.out.println("No log events found in the past 7 days.");
            } else {
                response.events().forEach(e -> System.out.println(e.message()));
            }
```

```
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la GetLogEventssezione AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`GetQueryResults`**con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`GetQueryResults`.

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:

- Eseguire una query di grandi dimensioni

.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>
    /// Gets the results of a CloudWatch Logs Insights query.
    /// </summary>
    /// <param name="queryId">The ID of the query.</param>
```

```
    /// <returns>The query results response.</returns>
    public async Task<GetQueryResultsResponse?> GetQueryResultsAsync(string
queryId)
    {
        try
        {
            var request = new GetQueryResultsRequest
            {
                QueryId = queryId
            };

            var response = await
_amazonCloudWatchLogs.GetQueryResultsAsync(request);
            return response;
        }
        catch (ResourceNotFoundException ex)
        {
            _logger.LogError($"Query not found: {ex.Message}");
            return null;
        }
        catch (Exception ex)
        {
            _logger.LogError($"An error occurred while getting query results:
{ex.Message}");
            return null;
        }
    }
```

- Per i dettagli sull'API, consulta la GetQueryResultssezione AWS SDK per .NET 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.

```
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
```

- Per i dettagli sull'API, consulta la GetQueryResultssezione 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.

```
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
```

```
                return results.get("results", [])
```

- Per i dettagli sull'API, consulta GetQueryResults AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_cwl->getqueryresults(
          iv_queryid = iv_query_id ).

        " Display query status and result count
        DATA(lv_status) = oo_result->get_status( ).
        DATA(lt_results) = oo_result->get_results( ).
        DATA(lv_result_count) = lines( lt_results ).

        MESSAGE |Query status: { lv_status }. Retrieved { lv_result_count } log
event(s).| TYPE 'I'.
      CATCH /aws1/cx_cwlinvalidparameterex.
        MESSAGE 'Invalid parameter.' TYPE 'E'.
      CATCH /aws1/cx_cwlresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'E'.
      CATCH /aws1/cx_cwlserviceunavailex.
        MESSAGE 'Service unavailable.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, GetQueryResultsconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`ListSyslogConfigurations`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `ListSyslogConfigurations`.

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:

- Scopri le nozioni di base sull'ingestione di syslog

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
    def list_syslog_configurations(
        self,
        log_group_identifier: Optional[str] = None,
        vpc_endpoint_id: Optional[str] = None,
    ) -> list[dict[str, Any]]:
        """
        Lists syslog configurations, optionally filtered by log group or VPC
endpoint.

        Handles pagination by following nextToken until all results are returned.

        :param log_group_identifier: Optional log group name or ARN to filter by.
        :param vpc_endpoint_id: Optional VPC endpoint ID to filter by.
        :return: A list of syslog configuration dictionaries.
        :raises ClientError: If a filter parameter is invalid.
        """
        try:
            configurations = list()
```

```
            params = dict()
            if log_group_identifier is not None:
                params["logGroupIdentifier"] = log_group_identifier
            if vpc_endpoint_id is not None:
                params["vpcEndpointId"] = vpc_endpoint_id

            while True:
                response = self.logs_client.list_syslog_configurations(**params)
                configurations.extend(response.get("syslogConfigurations",
list()))
                next_token = response.get("nextToken", None)
                if next_token is None:
                    break
                params["nextToken"] = next_token

            logger.info("Listed %d syslog configuration(s).",
len(configurations))
            return configurations
        except ClientError as error:
            if error.response["Error"]["Code"] == "InvalidParameterException":
                logger.error(
                    "Invalid filter parameter: %s. Ensure the log group
identifier "
                    "or VPC endpoint ID is correctly formatted.",
                    error.response["Error"]["Message"],
                )
            else:
                logger.error(
                    "Failed to list syslog configurations: %s",
                    error.response["Error"]["Message"],
                )
            raise
```

- Per i dettagli sull'API, consulta ListSyslogConfigurations AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **PutResourcePolicy**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `PutResourcePolicy`.

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:

- Scopri le nozioni di base sull'ingestione di syslog

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 put_resource_policy(
        self,
        log_group_arn: str,
        vpc_endpoint_id: str,
    ) -> dict[str, Any]:
        """
        Creates or updates a resource-scoped policy on a log group that grants
        the syslog.logs.amazonaws.com service principal permission to call
        logs:PutLogEvents and logs:CreateLogStream.

        A resource-scoped policy (one attached to a specific log group) is
        specified with resourceArn and policyDocument. The PutResourcePolicy
        API does not allow policyName to be combined with resourceArn.

        :param log_group_arn: The ARN of the log group (with trailing :*).
        :param vpc_endpoint_id: The VPC endpoint ID to scope the condition.
        :return: The resource policy response.
        """
        policy_document = json.dumps(
            {
                "Version":"2012-10-17",
                "Statement": [
```

```
                {
                    "Sid": "SyslogIngestPermissions",
                    "Effect": "Allow",
                    "Principal": {"Service": "syslog.logs.amazonaws.com"},
                    "Action": [
                        "logs:PutLogEvents",
                        "logs:CreateLogStream",
                    ],
                    "Resource": log_group_arn,
                    "Condition": {
                        "StringEquals": {"aws:sourceVpce": vpc_endpoint_id}
                    },
                }
            ],
        }
    )
    try:
        response = self.logs_client.put_resource_policy(
            policyDocument=policy_document,
            resourceArn=log_group_arn,
        )
        logger.info("Put resource policy on log group '%s'.", log_group_arn)
        return response
    except ClientError as error:
        logger.error(
            "Failed to put resource policy: %s",
            error.response["Error"]["Message"],
        )
        raise
```

- Per i dettagli sull'API, consulta PutResourcePolicy AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`PutSubscriptionFilter`**con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`PutSubscriptionFilter`.

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.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/PutSubscriptionFilterRequest.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Creare il filtro di sottoscrizione.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::PutSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetFilterPattern(filter_pattern);
        request.SetLogGroupName(log_group);
        request.SetDestinationArn(dest_arn);
        auto outcome = cwl.PutSubscriptionFilter(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch logs subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Per i dettagli sull'API, consulta la PutSubscriptionFiltersezione AWS SDK per C++ 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.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.PutSubscriptionFilterRequest;

/**
 * Before running this code example, you need to grant permission to CloudWatch
 * Logs the right to execute your Lambda function.
 * To perform this task, you can use this CLI command:
 *
 * aws lambda add-permission --function-name "lamda1" --statement-id "lamda1"
 * --principal "logs.us-west-2.amazonaws.com" --action "lambda:InvokeFunction"
 * --source-arn "arn:aws:logs:us-west-2:111111111111:log-group:testgroup:*"
 * --source-account "111111111111"
 *
 * Make sure you replace the function name with your function name and replace
 * '111111111111' with your account details.
 * For more information, see "Subscription Filters with AWS Lambda" in the
 * Amazon CloudWatch Logs Guide.
 *
 *
 * Also, 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 PutSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <pattern> <logGroup> <functionArn>\s

                Where:
                  filter - A filter name (for example, myfilter).
                  pattern - A filter pattern (for example, ERROR).
                  logGroup - A log group name (testgroup).
                  functionArn - An AWS Lambda function ARN (for example,
 arn:aws:lambda:us-west-2:111111111111:function:lambda1) .
                """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String filter = args[0];
        String pattern = args[1];
        String logGroup = args[2];
        String functionArn = args[3];
        Region region = Region.US_WEST_2;
        CloudWatchLogsClient cwl = CloudWatchLogsClient.builder()
                .region(region)
                .build();

        putSubFilters(cwl, filter, pattern, logGroup, functionArn);
        cwl.close();
    }

    public static void putSubFilters(CloudWatchLogsClient cwl,
            String filter,
            String pattern,
            String logGroup,
            String functionArn) {
```

```
        try {
            PutSubscriptionFilterRequest request =
 PutSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .filterPattern(pattern)
                    .logGroupName(logGroup)
                    .destinationArn(functionArn)
                    .build();

            cwl.putSubscriptionFilter(request);
            System.out.printf(
                    "Successfully created CloudWatch logs subscription filter
 %s",
                    filter);

        } catch (CloudWatchLogsException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la PutSubscriptionFiltersezione 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 { PutSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new PutSubscriptionFilterCommand({
```

```
    // An ARN of a same-account Kinesis stream, Kinesis Firehose
    // delivery stream, or Lambda function.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
SubscriptionFilters.html
    destinationArn: process.env.CLOUDWATCH_LOGS_DESTINATION_ARN,

    // A name for the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,

    // A filter pattern for subscribing to a filtered stream of log events.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
FilterAndPatternSyntax.html
    filterPattern: process.env.CLOUDWATCH_LOGS_FILTER_PATTERN,

    // The name of the log group. Messages in this group matching the filter
 pattern
    // will be sent to the destination ARN.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Per i dettagli sull'API, consulta la PutSubscriptionFiltersezione AWS SDK per JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
```

```
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  destinationArn: "LAMBDA_FUNCTION_ARN",
  filterName: "FILTER_NAME",
  filterPattern: "ERROR",
  logGroupName: "LOG_GROUP",
};

cwl.putSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK per JavaScript.
- Per i dettagli sull'API, consulta la PutSubscriptionFiltersezione AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`PutSyslogConfiguration`**con un AWS SDK

L'esempio di codice seguente mostra come utilizzare `PutSyslogConfiguration`.

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:

- Scopri le nozioni di base sull'ingestione di syslog

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 put_syslog_configuration(
        self, log_group_identifier: str, vpc_endpoint_id: str
    ) -> None:
        """
        Creates or updates a syslog configuration for a log group. This enables
        ingestion of syslog data through the specified VPC endpoint.

        :param log_group_identifier: The name or ARN of the log group to
associate
            with the syslog configuration.
        :param vpc_endpoint_id: The ID of the VPC endpoint to use for syslog
            ingestion.
        :raises ClientError: If the log group or VPC endpoint does not exist.
        """
        try:
            self.logs_client.put_syslog_configuration(
                logGroupIdentifier=log_group_identifier,
                vpcEndpointId=vpc_endpoint_id,
            )
            logger.info(
                "Created syslog configuration for log group '%s' with VPC
endpoint '%s'.",
                log_group_identifier,
                vpc_endpoint_id,
            )
        except ClientError as error:
            if error.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error(
                    "The specified log group or VPC endpoint does not exist: %s",
                    error.response["Error"]["Message"],
                )
            else:
```

```
            logger.error(
                "Failed to put syslog configuration: %s",
                error.response["Error"]["Message"],
            )
        raise
```

- Per i dettagli sull'API, consulta PutSyslogConfiguration AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`StartLiveTail`**con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`StartLiveTail`.

.NET

SDK per .NET

Includere i file richiesti.

```
using Amazon;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

Avvia la sessione Live Tail.

```
        var client = new AmazonCloudWatchLogsClient();
        var request = new StartLiveTailRequest
        {
            LogGroupIdentifiers = logGroupIdentifiers,
            LogStreamNames = logStreamNames,
            LogEventFilterPattern = filterPattern,
        };

        var response = await client.StartLiveTailAsync(request);
```

```
            // Catch if request fails
            if (response.HttpStatusCode != System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine("Failed to start live tail session");
                return;
            }
```

È possibile gestire gli eventi della sessione Live Tail in due modi.

```
            /* Method 1
            * 1). Asynchronously loop through the event stream
            * 2). Set a timer to dispose the stream and stop the Live Tail
session at the end.
            */
            var eventStream = response.ResponseStream;
            var task = Task.Run(() =>
            {
                foreach (var item in eventStream)
                {
                    if (item is LiveTailSessionUpdate liveTailSessionUpdate)
                    {
                        foreach (var sessionResult in
liveTailSessionUpdate.SessionResults)
                        {
                            Console.WriteLine("Message : {0}",
sessionResult.Message);
                        }
                    }
                    if (item is LiveTailSessionStart)
                    {
                        Console.WriteLine("Live Tail session started");
                    }
                    // On-stream exceptions are processed here
                    if (item is CloudWatchLogsEventStreamException)
                    {
                        Console.WriteLine($"ERROR: {item}");
                    }
                }
            });
            // Close the stream to stop the session after a timeout
            if (!task.Wait(TimeSpan.FromSeconds(10))){
```

```
                eventStream.Dispose();
                Console.WriteLine("End of line");
            }
```

```
            /* Method 2
            * 1). Add event handlers to each event variable
            * 2). Start processing the stream and wait for a timeout using
AutoResetEvent
            */
            AutoResetEvent endEvent = new AutoResetEvent(false);
            var eventStream = response.ResponseStream;
            using (eventStream) // automatically disposes the stream to stop the
session after execution finishes
            {
                eventStream.SessionStartReceived += (sender, e) =>
                {
                    Console.WriteLine("LiveTail session started");
                };
                eventStream.SessionUpdateReceived += (sender, e) =>
                {
                    foreach (LiveTailSessionLogEvent logEvent in
e.EventStreamEvent.SessionResults){
                        Console.WriteLine("Message: {0}", logEvent.Message);
                    }
                };
                // On-stream exceptions are captured here
                eventStream.ExceptionReceived += (sender, e) =>
                {
                    Console.WriteLine($"ERROR:
{e.EventStreamException.Message}");
                };

                eventStream.StartProcessing();
                // Stream events for this amount of time.
                endEvent.WaitOne(TimeSpan.FromSeconds(10));
                Console.WriteLine("End of line");
            }
```

- Per i dettagli sull'API, consulta StartLiveTail AWS SDK per .NETAPI Reference.

CLI

AWS CLI

Il comando seguente avvia una sessione Live Tail su un gruppo di log denominatomy-logs:

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs
```

Il comando seguente avvia una sessione Live Tail su un gruppo di log denominato my-logs in modalità interattiva:

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs --mode interactive
```

In modalità interattiva è possibile evidenziare fino a cinque termini nei registri di coda. I codici di gravità sono evidenziati per impostazione predefinita. Premi h per accedere alla modalità di evidenziazione, quindi digita i termini da evidenziare, uno alla volta, e premi invio. Premi c per cancellare i termini evidenziati. tPremete per alternare la formattazione tra i testi. JSON/ Plain Premi Esc per uscire dalla sessione di Live Tail. Premi i down tastiup/per scorrere verso l'alto o verso il basso tra gli eventi del registro, usaCtrl + u/Ctrl + dper scorrere più velocemente. qPremete per scorrere fino agli ultimi eventi del registro.

- Per i dettagli sull'API, consulta StartLiveTail AWS CLICommand Reference.

Go

SDK per Go V2

Includere i file richiesti.

```
import (
 "context"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)
```

Gestisce gli eventi della sessione Live Tail.

```
func handleEventStreamAsync(stream *cloudwatchlogs.StartLiveTailEventStream) {
 eventsChan := stream.Events()
 for {
  event := <-eventsChan
  switch e := event.(type) {
  case *types.StartLiveTailResponseStreamMemberSessionStart:
   log.Println("Received SessionStart event")
  case *types.StartLiveTailResponseStreamMemberSessionUpdate:
   for _, logEvent := range e.Value.SessionResults {
    log.Println(*logEvent.Message)
   }
  default:
   // Handle on-stream exceptions
   if err := stream.Err(); err != nil {
    log.Fatalf("Error occured during streaming: %v", err)
   } else if event == nil {
    log.Println("Stream is Closed")
    return
   } else {
    log.Fatalf("Unknown event type: %T", e)
   }
  }
 }
}
```

Avvia la sessione Live Tail.

```
 cfg, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  panic("configuration error, " + err.Error())
 }
 client := cloudwatchlogs.NewFromConfig(cfg)

 request := &cloudwatchlogs.StartLiveTailInput{
  LogGroupIdentifiers:   logGroupIdentifiers,
  LogStreamNames:        logStreamNames,
  LogEventFilterPattern: logEventFilterPattern,
 }

 response, err := client.StartLiveTail(context.TODO(), request)
```

```
// Handle pre-stream Exceptions
if err != nil {
 log.Fatalf("Failed to start streaming: %v", err)
}

// Start a Goroutine to handle events over stream
stream := response.GetStream()
go handleEventStreamAsync(stream)
```

Arresta la sessione Live Tail dopo un determinato periodo di tempo.

```
// Close the stream (which ends the session) after a timeout
time.Sleep(10 * time.Second)
stream.Close()
log.Println("Event stream closed")
```

- Per i dettagli sull'API, StartLiveTailconsulta AWS SDK per Go API Reference.

Java

SDK per Java 2.x

Includere i file richiesti.

```
import io.reactivex.FlowableSubscriber;
import io.reactivex.annotations.NonNull;
import org.reactivestreams.Subscription;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsAsyncClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionStart;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionUpdate;
import software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseHandler;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
```

```
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseStream;

import java.util.Date;
import java.util.List;
import java.util.concurrent.atomic.AtomicReference;
```

Gestisce gli eventi della sessione Live Tail.

```
   private static StartLiveTailResponseHandler
 getStartLiveTailResponseStreamHandler(
           AtomicReference<Subscription> subscriptionAtomicReference) {
       return StartLiveTailResponseHandler.builder()
           .onResponse(r -> System.out.println("Received initial response"))
           .onError(throwable -> {
               CloudWatchLogsException e = (CloudWatchLogsException)
 throwable.getCause();
               System.err.println(e.awsErrorDetails().errorMessage());
               System.exit(1);
           })
           .subscriber(() -> new FlowableSubscriber<>() {
               @Override
               public void onSubscribe(@NonNull Subscription s) {
                   subscriptionAtomicReference.set(s);
                   s.request(Long.MAX_VALUE);
               }

               @Override
               public void onNext(StartLiveTailResponseStream event) {
                   if (event instanceof LiveTailSessionStart) {
                       LiveTailSessionStart sessionStart =
 (LiveTailSessionStart) event;
                       System.out.println(sessionStart);
                   } else if (event instanceof LiveTailSessionUpdate) {
                       LiveTailSessionUpdate sessionUpdate =
 (LiveTailSessionUpdate) event;
                       List<LiveTailSessionLogEvent> logEvents =
 sessionUpdate.sessionResults();
                       logEvents.forEach(e -> {
                           long timestamp = e.timestamp();
                           Date date = new Date(timestamp);
                           System.out.println("[" + date + "] " + e.message());
```

```
                    });
                } else {
                    throw CloudWatchLogsException.builder().message("Unknown
event type").build();
                }
            }

            @Override
            public void onError(Throwable throwable) {
                System.out.println(throwable.getMessage());
                System.exit(1);
            }

            @Override
            public void onComplete() {
                System.out.println("Completed Streaming Session");
            }
        })
        .build();
   }
```

Avvia la sessione Live Tail.

```
       CloudWatchLogsAsyncClient cloudWatchLogsAsyncClient =
               CloudWatchLogsAsyncClient.builder()
                   .credentialsProvider(ProfileCredentialsProvider.create())
                   .build();

       StartLiveTailRequest request =
               StartLiveTailRequest.builder()
                   .logGroupIdentifiers(logGroupIdentifiers)
                   .logStreamNames(logStreamNames)
                   .logEventFilterPattern(logEventFilterPattern)
                   .build();

       /* Create a reference to store the subscription */
       final AtomicReference<Subscription> subscriptionAtomicReference = new
AtomicReference<>(null);

       cloudWatchLogsAsyncClient.startLiveTail(request,
getStartLiveTailResponseStreamHandler(subscriptionAtomicReference));
```

Arresta la sessione Live Tail dopo un determinato periodo di tempo.

```
        /* Set a timeout for the session and cancel the subscription. This will:
         * 1). Close the stream
         * 2). Stop the Live Tail session
         */
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        if (subscriptionAtomicReference.get() != null) {
            subscriptionAtomicReference.get().cancel();
            System.out.println("Subscription to stream closed");
        }
```

- Per i dettagli sull'API, StartLiveTailconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Includere i file richiesti.

```
import { CloudWatchLogsClient, StartLiveTailCommand } from "@aws-sdk/client-
cloudwatch-logs";
```

Gestisce gli eventi della sessione Live Tail.

```
async function handleResponseAsync(response) {
    try {
      for await (const event of response.responseStream) {
        if (event.sessionStart !== undefined) {
          console.log(event.sessionStart);
        } else if (event.sessionUpdate !== undefined) {
          for (const logEvent of event.sessionUpdate.sessionResults) {
            const timestamp = logEvent.timestamp;
            const date = new Date(timestamp);
            console.log("[" + date + "] " + logEvent.message);
```

```
          }
        } else {
            console.error("Unknown event type");
        }
      }
    } catch (err) {
        // On-stream exceptions are captured here
        console.error(err)
    }
}
```

Avvia la sessione Live Tail.

```
    const client = new CloudWatchLogsClient();

    const command = new StartLiveTailCommand({
        logGroupIdentifiers: logGroupIdentifiers,
        logStreamNames: logStreamNames,
        logEventFilterPattern: filterPattern
    });
    try{
        const response = await client.send(command);
        handleResponseAsync(response);
    } catch (err){
        // Pre-stream exceptions are captured here
        console.log(err);
    }
```

Arresta la sessione Live Tail dopo un determinato periodo di tempo.

```
    /* Set a timeout to close the client. This will stop the Live Tail session.
 */
    setTimeout(function() {
        console.log("Client timeout");
        client.destroy();
      }, 10000);
```

- Per i dettagli sull'API, consulta la sezione AWS SDK per JavaScript API StartLiveTailReference.

Kotlin

SDK per Kotlin

Includere i file richiesti.

```
import aws.sdk.kotlin.services.cloudwatchlogs.CloudWatchLogsClient
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailRequest
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailResponseStream
import kotlinx.coroutines.flow.takeWhile
```

Avvia la sessione Live Tail.

```
    val client = CloudWatchLogsClient.fromEnvironment()

    val request = StartLiveTailRequest {
        logGroupIdentifiers = logGroupIdentifiersVal
        logStreamNames = logStreamNamesVal
        logEventFilterPattern = logEventFilterPatternVal
    }

    val startTime = System.currentTimeMillis()

    try {
        client.startLiveTail(request) { response ->
            val stream = response.responseStream
            if (stream != null) {
                /* Set a timeout to unsubcribe from the flow. This will:
                * 1). Close the stream
                * 2). Stop the Live Tail session
                */
                stream.takeWhile { System.currentTimeMillis() - startTime <
10000 }.collect { value ->
                    if (value is StartLiveTailResponseStream.SessionStart) {
                        println(value.asSessionStart())
                    } else if (value is
StartLiveTailResponseStream.SessionUpdate) {
                        for (e in value.asSessionUpdate().sessionResults!!) {
                            println(e)
                        }
                    } else {
                        throw IllegalArgumentException("Unknown event type")
```

```
                    }
                }
            } else {
                throw IllegalArgumentException("No response stream")
            }
        }
    } catch (e: Exception) {
        println("Exception occurred during StartLiveTail: $e")
        System.exit(1)
    }
```

- Per i dettagli sull'API, StartLiveTailconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

Includere i file richiesti.

```
import boto3
import time
from datetime import datetime
```

Avvia la sessione Live Tail.

```
    # Initialize the client
    client = boto3.client('logs')

    start_time = time.time()

    try:
        response = client.start_live_tail(
            logGroupIdentifiers=log_group_identifiers,
            logStreamNames=log_streams,
            logEventFilterPattern=filter_pattern
        )
        event_stream = response['responseStream']
        # Handle the events streamed back in the response
        for event in event_stream:
            # Set a timeout to close the stream.
```

```
            # This will end the Live Tail session.
            if (time.time() - start_time >= 10):
                event_stream.close()
                break
            # Handle when session is started
            if 'sessionStart' in event:
                session_start_event = event['sessionStart']
                print(session_start_event)
            # Handle when log event is given in a session update
            elif 'sessionUpdate' in event:
                log_events = event['sessionUpdate']['sessionResults']
                for log_event in log_events:
                    print('[{date}]
{log}'.format(date=datetime.fromtimestamp(log_event['timestamp']/1000),log=log_event['me
            else:
                # On-stream exceptions are captured here
                raise RuntimeError(str(event))
    except Exception as e:
        print(e)
```

- Per i dettagli sull'API, consulta StartLiveTail AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzare **`StartQuery`**con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`StartQuery`.

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:

- Eseguire una query di grandi dimensioni

.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>
    /// Starts a CloudWatch Logs Insights query.
    /// </summary>
    /// <param name="logGroupName">The name of the log group to query.</param>
    /// <param name="queryString">The CloudWatch Logs Insights query string.</
param>
    /// <param name="startTime">The start time for the query (seconds since
 epoch).</param>
    /// <param name="endTime">The end time for the query (seconds since epoch).</
param>
    /// <param name="limit">The maximum number of results to return.</param>
    /// <returns>The query ID if successful, null otherwise.</returns>
    public async Task<string?> StartQueryAsync(
        string logGroupName,
        string queryString,
        long startTime,
        long endTime,
        int limit = 10000)
    {
        try
        {
            var request = new StartQueryRequest
            {
                LogGroupName = logGroupName,
                QueryString = queryString,
                StartTime = startTime,
                EndTime = endTime,
                Limit = limit
            };

            var response = await _amazonCloudWatchLogs.StartQueryAsync(request);
            return response.QueryId;
```

```
        }
        catch (InvalidParameterException ex)
        {
            _logger.LogError($"Invalid parameter for query: {ex.Message}");
            return null;
        }
        catch (ResourceNotFoundException ex)
        {
            _logger.LogError($"Log group not found: {ex.Message}");
            return null;
        }
        catch (Exception ex)
        {
            _logger.LogError($"An error occurred while starting query:
{ex.Message}");
            return null;
        }
    }
```

- Per i dettagli sull'API, consulta la StartQuerysezione AWS SDK per .NET 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.

```
/**
 * Wrapper for the StartQueryCommand. Uses a static query string
 * for consistency.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 * @returns {Promise<{ queryId: string }>}
 */
async _startQuery([startDate, endDate], maxLogs = 10000) {
  try {
    return await this.client.send(
```

```
        new StartQueryCommand({
          logGroupNames: this.logGroupNames,
          queryString: "fields @timestamp, @message | sort @timestamp asc",
          startTime: startDate.valueOf(),
          endTime: endDate.valueOf(),
          limit: maxLogs,
        }),
      );
    } catch (err) {
      /** @type {string} */
      const message = err.message;
      if (message.startsWith("Query's end date and time")) {
        // This error indicates that the query's start or end date occur
        // before the log group was created.
        throw new DateOutOfBoundsError(message);
      }

      throw err;
    }
  }
```

- Per i dettagli sull'API, consulta la StartQuerysezione 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.

```
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
```

```
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_group,
                    startTime=start_time,
                    endTime=end_time,
                    queryString=self.query_string,
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.
```

```
        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_group,
                startTime=start_time,
                endTime=end_time,
                queryString=self.query_string,
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")
```

- Per i dettagli sull'API, consulta StartQuery 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.
       " iv_log_group_name = '/aws/lambda/my-function'
       " iv_query_string = 'fields @timestamp, @message | sort @timestamp desc |
limit 20'
       " iv_start_time and iv_end_time must be in Unix epoch milliseconds (ms
since Jan 1, 1970 00:00:00 UTC)
       oo_result = lo_cwl->startquery(
         iv_loggroupname = iv_log_group_name
         iv_starttime    = iv_start_time
         iv_endtime      = iv_end_time
         iv_querystring  = iv_query_string
         iv_limit        = iv_limit ).

       " Display the query ID for tracking
       DATA(lv_query_id) = oo_result->get_queryid( ).
       MESSAGE |Query started successfully with ID: { lv_query_id }| TYPE 'I'.
     CATCH /aws1/cx_cwlinvalidparameterex.
       MESSAGE 'Invalid parameter.' TYPE 'E'.
     CATCH /aws1/cx_cwllimitexceededex.
       MESSAGE 'Limit exceeded.' TYPE 'E'.
     CATCH /aws1/cx_cwlmalformedqueryex.
       MESSAGE 'Malformed query.' TYPE 'E'.
     CATCH /aws1/cx_cwlresourcenotfoundex.
       MESSAGE 'Resource not found.' TYPE 'E'.
     CATCH /aws1/cx_cwlserviceunavailex.
       MESSAGE 'Service unavailable.' TYPE 'E'.
   ENDTRY.
```

- Per i dettagli sulle API, StartQueryconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per l'utilizzo CloudWatch dei log AWS SDK

I seguenti esempi di codice mostrano come implementare scenari comuni in CloudWatch Logs with SDK. AWS Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno di CloudWatch Logs o combinandole con altre. Servizi AWS Ogni scenario include un

collegamento al codice sorgente completo, dove è possibile trovare le istruzioni su come configurare ed eseguire il codice.

Gli scenari sono relativi a un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- Configura la connettività del servizio container
- Creazione della tua prima funzione serverless
- Scopri le nozioni di base sull'ingestione di CloudWatch Logs syslog utilizzando un AWS SDK
- Usa CloudWatch Logs per eseguire una query di grandi dimensioni
- Utilizzo degli eventi pianificati per invocare una funzione Lambda

Configura la connettività del servizio container

L'esempio di codice seguente mostra come:

- Crea l'infrastruttura VPC
- Configurare la registrazione
- Crea il cluster ECS
- Configura i ruoli IAM
- Crea il servizio con Service Connect
- Verifica della distribuzione
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# ECS Service Connect Tutorial Script v4 - Modified to use Default VPC
# This script creates an ECS cluster with Service Connect and deploys an nginx
 service
# Uses the default VPC to avoid VPC limits

set -e  # Exit on any error

# Configuration
SCRIPT_NAME="ECS Service Connect Tutorial"
LOG_FILE="ecs-service-connect-tutorial-v4-default-vpc.log"
REGION=${AWS_DEFAULT_REGION:-${AWS_REGION:-$(aws configure get region 2>/dev/
null)}}
if [ -z "$REGION" ]; then
    echo "ERROR: No AWS region configured."
    echo "Set one with: aws configure set region us-east-1"
    exit 1
fi
ENV_PREFIX="tutorial"
CLUSTER_NAME="${ENV_PREFIX}-cluster"
NAMESPACE_NAME="service-connect"

# Generate random suffix for unique resource names
RANDOM_SUFFIX=$(openssl rand -hex 6)

# Arrays to track created resources for cleanup
declare -a CREATED_RESOURCES=()

# Logging function
log() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}

# Error handling function
handle_error() {
    log "ERROR: Script failed at line $1"
    log "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Set up error handling
```

```
trap 'handle_error $LINENO' ERR

# Function to add resource to tracking array
track_resource() {
    CREATED_RESOURCES+=("$1")
    log "Tracking resource: $1"
}

# Function to check if command output contains actual errors
check_for_errors() {
    local output="$1"
    local command_name="$2"

    # Check for specific AWS CLI error patterns, not just any occurrence of
 "error"
    if echo "$output" | grep -qi "An error occurred\|InvalidParameterException\|
AccessDenied\|ValidationException\|ResourceNotFoundException"; then
        log "ERROR in $command_name: $output"
        return 1
    fi
    return 0
}

# Function to get AWS account ID
get_account_id() {
    ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
    log "Using AWS Account ID: $ACCOUNT_ID"
}

# Function to wait for resources to be ready
wait_for_resource() {
    local resource_type="$1"
    local resource_id="$2"

    case "$resource_type" in
        "cluster")
            log "Waiting for cluster $resource_id to be active..."
            local attempt=1
            local max_attempts=30
            while [ $attempt -le $max_attempts ]; do
                local status=$(aws ecs describe-clusters --clusters
 "$resource_id" --query 'clusters[0].status' --output text)
                if [ "$status" = "ACTIVE" ]; then
                    log "Cluster is now active"
```

```
                    return 0
                fi
                log "Cluster status: $status (attempt $attempt/$max_attempts)"
                sleep 10
                ((attempt++))
            done
            log "ERROR: Cluster did not become active within expected time"
            return 1
            ;;
        "service")
            log "Waiting for service $resource_id to be stable..."
            aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$resource_id"
            ;;
        "nat-gateway")
            log "Waiting for NAT Gateway $resource_id to be available..."
            aws ec2 wait nat-gateway-available --nat-gateway-ids "$resource_id"
            ;;
    esac
}

# Function to use default VPC infrastructure
setup_default_vpc_infrastructure() {
    log "Using default VPC infrastructure..."

    # Get default VPC
    VPC_ID=$(aws ec2 describe-vpcs --filters "Name=isDefault,Values=true" --query
 'Vpcs[0].VpcId' --output text)
    if [[ "$VPC_ID" == "None" || -z "$VPC_ID" ]]; then
        log "ERROR: No default VPC found. Please create a default VPC first."
        exit 1
    fi
    log "Using default VPC: $VPC_ID"

    # Get default subnets
    SUBNETS=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC_ID"
 "Name=default-for-az,Values=true" --query 'Subnets[].SubnetId' --output text)
    SUBNET_ARRAY=($SUBNETS)

    if [ ${#SUBNET_ARRAY[@]} -lt 2 ]; then
        log "ERROR: Need at least 2 subnets for ECS Service Connect. Found:
 ${#SUBNET_ARRAY[@]}"
        exit 1
    fi
```

```
    PUBLIC_SUBNET1=${SUBNET_ARRAY[0]}
    PUBLIC_SUBNET2=${SUBNET_ARRAY[1]}

    log "Using subnets: $PUBLIC_SUBNET1, $PUBLIC_SUBNET2"

    # Create security group for ECS tasks
    SG_OUTPUT=$(aws ec2 create-security-group \
        --group-name "${ENV_PREFIX}-ecs-sg-${RANDOM_SUFFIX}" \
        --description "Security group for ECS Service Connect tutorial" \
        --vpc-id "$VPC_ID" \
        --tag-specifications 'ResourceType=security-
group,Tags=[{Key=project,Value=doc-smith},{Key=tutorial,Value=amazon-ecs-service-
connect}]' 2>&1)
    check_for_errors "$SG_OUTPUT" "create-security-group"
    SECURITY_GROUP_ID=$(echo "$SG_OUTPUT" | grep -o '"GroupId": "[^"]*"' | cut -
d'"' -f4)
    track_resource "SG:$SECURITY_GROUP_ID"
    log "Created security group: $SECURITY_GROUP_ID"

    # Add inbound rules to security group
    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 80 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 443 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    log "Default VPC infrastructure setup completed"
}

# Function to create CloudWatch log groups
create_log_groups() {
    log "Creating CloudWatch log groups..."

    # Create log group for nginx container
    aws logs create-log-group --log-group-name "/ecs/service-connect-nginx"
 --tags project=doc-smith,tutorial=amazon-ecs-service-connect 2>&1 | grep -v
 "ResourceAlreadyExistsException" || {
```

```
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-nginx created"
            track_resource "LOG_GROUP:/ecs/service-connect-nginx"
        else
            log "Log group /ecs/service-connect-nginx already exists"
        fi
    }

    # Create log group for service connect proxy
    aws logs create-log-group --log-group-name "/ecs/service-connect-proxy"
 --tags project=doc-smith,tutorial=amazon-ecs-service-connect 2>&1 | grep -v
 "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-proxy created"
            track_resource "LOG_GROUP:/ecs/service-connect-proxy"
        else
            log "Log group /ecs/service-connect-proxy already exists"
        fi
    }
}

# Function to create ECS cluster with Service Connect
create_ecs_cluster() {
    log "Creating ECS cluster with Service Connect..."

    CLUSTER_OUTPUT=$(aws ecs create-cluster \
        --cluster-name "$CLUSTER_NAME" \
        --service-connect-defaults namespace="$NAMESPACE_NAME" \
        --tags key=Environment,value=tutorial key=project,value=doc-smith
 key=tutorial,value=amazon-ecs-service-connect 2>&1)
    check_for_errors "$CLUSTER_OUTPUT" "create-cluster"

    track_resource "CLUSTER:$CLUSTER_NAME"
    log "Created ECS cluster: $CLUSTER_NAME"

    wait_for_resource "cluster" "$CLUSTER_NAME"

    # Track the Service Connect namespace that gets created
    # Wait a moment for the namespace to be created
    sleep 5
    NAMESPACE_ID=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")
```

```
    if [[ -n "$NAMESPACE_ID" && "$NAMESPACE_ID" != "None" ]]; then
        track_resource "NAMESPACE:$NAMESPACE_ID"
        log "Service Connect namespace created: $NAMESPACE_ID"
    fi
}

# Function to create IAM roles
create_iam_roles() {
    log "Creating IAM roles..."

    # Check if ecsTaskExecutionRole exists
    if aws iam get-role --role-name ecsTaskExecutionRole >/dev/null 2>&1; then
        log "IAM role ecsTaskExecutionRole exists"
    else
        log "Creating ecsTaskExecutionRole..."
        aws iam create-role \
            --role-name ecsTaskExecutionRole \
            --assume-role-policy-document '{
                "Version":"2012-10-17",
                "Statement": [{
                    "Effect": "Allow",
                    "Principal": {"Service": "ecs-tasks.amazonaws.com"},
                    "Action": "sts:AssumeRole"
                }]
            }' >/dev/null 2>&1
        aws iam attach-role-policy \
            --role-name ecsTaskExecutionRole \
            --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy >/dev/null 2>&1
        track_resource "ROLE:ecsTaskExecutionRole"
        aws iam tag-role --role-name ecsTaskExecutionRole --tags
 Key=project,Value=doc-smith Key=tutorial,Value=amazon-ecs-service-connect
        log "Created ecsTaskExecutionRole"
        sleep 10
    fi

    # Check if ecsTaskRole exists, create if not
    if aws iam get-role --role-name ecsTaskRole >/dev/null 2>&1; then
        log "IAM role ecsTaskRole exists"
    else
        log "IAM role ecsTaskRole does not exist, will create it"

        # Create trust policy for ECS tasks
```

```
        cat > /tmp/ecs-task-trust-policy.json << EOF
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF

        aws iam create-role \
            --role-name ecsTaskRole \
            --assume-role-policy-document file:///tmp/ecs-task-trust-policy.json
 >/dev/null

        track_resource "IAM_ROLE:ecsTaskRole"
        aws iam tag-role --role-name ecsTaskRole --tags Key=project,Value=doc-
smith Key=tutorial,Value=amazon-ecs-service-connect
        log "Created ecsTaskRole"

        # Wait for role to be available
        sleep 10
    fi
}

# Function to create task definition
create_task_definition() {
    log "Creating task definition..."

    # Create task definition JSON
    cat > /tmp/task-definition.json << EOF
{
    "family": "service-connect-nginx",
    "networkMode": "awsvpc",
    "requiresCompatibilities": ["FARGATE"],
    "cpu": "256",
    "memory": "512",
    "executionRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskExecutionRole",
    "taskRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskRole",
```

```
    "containerDefinitions": [
        {
            "name": "nginx",
            "image": "public.ecr.aws/docker/library/nginx:latest",
            "portMappings": [
                {
                    "containerPort": 80,
                    "protocol": "tcp",
                    "name": "nginx-port"
                }
            ],
            "essential": true,
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/service-connect-nginx",
                    "awslogs-region": "${REGION}",
                    "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ]
}
EOF

    TASK_DEF_OUTPUT=$(aws ecs register-task-definition --cli-input-json file:///
tmp/task-definition.json 2>&1)
    check_for_errors "$TASK_DEF_OUTPUT" "register-task-definition"

    TASK_DEF_ARN=$(echo "$TASK_DEF_OUTPUT" | grep -o '"taskDefinitionArn":
 "[^"]*"' | cut -d'"' -f4)
    track_resource "TASK_DEF:service-connect-nginx"
    log "Created task definition: $TASK_DEF_ARN"

    # Clean up temporary file
    rm -f /tmp/task-definition.json
}

# Function to create ECS service with Service Connect
create_ecs_service() {
    log "Creating ECS service with Service Connect..."

    # Create service definition JSON
    cat > /tmp/service-definition.json << EOF
```

```
{
    "serviceName": "service-connect-nginx-service",
    "cluster": "${CLUSTER_NAME}",
    "taskDefinition": "service-connect-nginx",
    "desiredCount": 1,
    "launchType": "FARGATE",
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": ["${PUBLIC_SUBNET1}", "${PUBLIC_SUBNET2}"],
            "securityGroups": ["${SECURITY_GROUP_ID}"],
            "assignPublicIp": "ENABLED"
        }
    },
    "serviceConnectConfiguration": {
        "enabled": true,
        "namespace": "${NAMESPACE_NAME}",
        "services": [
            {
                "portName": "nginx-port",
                "discoveryName": "nginx",
                "clientAliases": [
                    {
                        "port": 80,
                        "dnsName": "nginx"
                    }
                ]
            }
        ],
        "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
                "awslogs-group": "/ecs/service-connect-proxy",
                "awslogs-region": "${REGION}",
                "awslogs-stream-prefix": "ecs-service-connect"
            }
        }
    },
    "tags": [
        {
            "key": "Environment",
            "value": "tutorial"
        },
        {
            "key": "project",
```

```
            "value": "doc-smith"
        },
        {
            "key": "tutorial",
            "value": "amazon-ecs-service-connect"
        }
    ]
}
EOF

    SERVICE_OUTPUT=$(aws ecs create-service --cli-input-json file:///tmp/service-
definition.json 2>&1)
    check_for_errors "$SERVICE_OUTPUT" "create-service"

    track_resource "SERVICE:service-connect-nginx-service"
    log "Created ECS service: service-connect-nginx-service"

    wait_for_resource "service" "service-connect-nginx-service"

    # Clean up temporary file
    rm -f /tmp/service-definition.json
}

# Function to verify deployment
verify_deployment() {
    log "Verifying deployment..."

    # Check service status
    SERVICE_STATUS=$(aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].status' --output text)
    log "Service status: $SERVICE_STATUS"

    # Check running tasks
    RUNNING_COUNT=$(aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].runningCount' --output text)
    log "Running tasks: $RUNNING_COUNT"

    # Get task ARN
    TASK_ARN=$(aws ecs list-tasks \
        --cluster "$CLUSTER_NAME" \
```

```
        --service-name "service-connect-nginx-service" \
        --query 'taskArns[0]' --output text)

    if [[ "$TASK_ARN" != "None" && -n "$TASK_ARN" ]]; then
        log "Task ARN: $TASK_ARN"

        # Try to get task IP address
        TASK_IP=$(aws ecs describe-tasks \
            --cluster "$CLUSTER_NAME" \
            --tasks "$TASK_ARN" \
            --query 'tasks[0].attachments[0].details[?
name==`privateIPv4Address`].value' \
            --output text 2>/dev/null || echo "")

        if [[ -n "$TASK_IP" && "$TASK_IP" != "None" ]]; then
            log "Task IP address: $TASK_IP"
        else
            log "Could not retrieve task IP address"
        fi
    fi

    # Check Service Connect namespace
    NAMESPACE_STATUS=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

    if [[ -n "$NAMESPACE_STATUS" && "$NAMESPACE_STATUS" != "None" ]]; then
        log "Service Connect namespace '$NAMESPACE_NAME' is active"
    else
        log "Service Connect namespace '$NAMESPACE_NAME' not found or not active"
    fi

    # Display Service Connect configuration
    log "Service Connect configuration:"
    aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].serviceConnectConfiguration' 2>/dev/null || true
}

# Function to display created resources
display_resources() {
    echo ""
```

```
    echo "========================================="
    echo "CREATED RESOURCES"
    echo "========================================="
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "- $resource"
    done
    echo "========================================="
    echo ""
}

# Function to cleanup resources
cleanup_resources() {
    log "Starting cleanup process..."

    # Delete resources in reverse order of creation
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        resource="${CREATED_RESOURCES[i]}"
        resource_type=$(echo "$resource" | cut -d':' -f1)
        resource_id=$(echo "$resource" | cut -d':' -f2)

        log "Cleaning up $resource_type: $resource_id"

        case "$resource_type" in
            "SERVICE")
                aws ecs update-service --cluster "$CLUSTER_NAME" --service
 "$resource_id" --desired-count 0 2>&1 | grep -qi "error" && log "Warning: Failed
 to scale down service $resource_id"
                aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$resource_id" 2>/dev/null || true
                aws ecs delete-service --cluster "$CLUSTER_NAME" --service
 "$resource_id" --force 2>&1 | grep -qi "error" && log "Warning: Failed to delete
 service $resource_id"
                ;;
            "TASK_DEF")
                TASK_DEF_ARNS=$(aws ecs list-task-definitions --family-prefix
 "$resource_id" --query 'taskDefinitionArns' --output text 2>/dev/null)
                for arn in $TASK_DEF_ARNS; do
                    aws ecs deregister-task-definition --task-definition "$arn"
 >/dev/null 2>&1 || true
                done
                ;;
            "ROLE")
```

```
                aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
                aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete role $resource_id"
                ;;
            "IAM_ROLE")
                aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
                aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete role $resource_id"
                ;;
            "CLUSTER")
                aws ecs delete-cluster --cluster "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete cluster $resource_id"
                ;;
            "SG")
                for attempt in 1 2 3 4 5; do
                    if aws ec2 delete-security-group --group-id "$resource_id"
 2>/dev/null; then
                        break
                    fi
                    log "Security group $resource_id still has dependencies,
 retrying in 30s ($attempt/5)..."
                    sleep 30
                done
                ;;
            "LOG_GROUP")
                aws logs delete-log-group --log-group-name "$resource_id" 2>&1 |
 grep -qi "error" && log "Warning: Failed to delete log group $resource_id"
                ;;
            "NAMESPACE")
                # First, delete any services in the namespace
                NAMESPACE_SERVICES=$(aws servicediscovery list-services \
                    --filters Name=NAMESPACE_ID,Values="$resource_id" \
                    --query 'Services[].Id' --output text 2>/dev/null || echo "")

                if [[ -n "$NAMESPACE_SERVICES" && "$NAMESPACE_SERVICES" !=
 "None" ]]; then
                    for service_id in $NAMESPACE_SERVICES; do
                        aws servicediscovery delete-service --id "$service_id" >/
dev/null 2>&1 || true
                        sleep 2
```

```
                    done
                fi

                # Then delete the namespace
                aws servicediscovery delete-namespace --id "$resource_id" >/dev/
null 2>&1 || true
                ;;
        esac

        sleep 2  # Brief pause between deletions
    done

    # Clean up temporary files
    rm -f /tmp/ecs-task-trust-policy.json
    rm -f /tmp/task-definition.json
    rm -f /tmp/service-definition.json

    log "Cleanup completed"
}

# Main execution
main() {
    log "Starting $SCRIPT_NAME v4 (Default VPC)"
    log "Region: $REGION"
    log "Log file: $LOG_FILE"

    # Get AWS account ID
    get_account_id

    # Setup infrastructure using default VPC
    setup_default_vpc_infrastructure

    # Create CloudWatch log groups
    create_log_groups

    # Create ECS cluster
    create_ecs_cluster

    # Create IAM roles
    create_iam_roles

    # Create task definition
    create_task_definition
```

```
    # Create ECS service
    create_ecs_service

    # Verify deployment
    verify_deployment

    log "Tutorial completed successfully!"

    # Display created resources
    display_resources

    # Ask user if they want to clean up
    echo ""
    echo "==========================================="
    echo "CLEANUP CONFIRMATION"
    echo "==========================================="
    echo "Do you want to clean up all created resources? (y/n): "
    CLEANUP_CHOICE="y"

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        cleanup_resources
        log "All resources have been cleaned up"
    else
        log "Resources left intact. You can clean them up later by running the
 cleanup function."
        echo ""
        echo "To clean up resources later, you can use the AWS CLI commands or
 the AWS Management Console."
        echo "Remember to delete resources in the correct order to avoid
 dependency issues."
    fi
}

# Make script executable and run
chmod +x "$0"
main "$@"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress

- CreateCluster
- CreateLogGroup
- CreateRole
- CreateSecurityGroup
- CreateService
- DeleteCluster
- DeleteLogGroup
- DeleteNamespace
- DeleteRole
- DeleteSecurityGroup
- DeleteService (AWS Cloud Map)
- DeleteService (Amazon ECS)
- DeregisterTaskDefinition
- DescribeClusters
- DescribeServices
- DescribeSubnets
- DescribeTasks
- DescribeVpcs
- DetachRolePolicy
- GetCallerIdentity
- GetRole
- ListNamespaces
- ListServices
- ListTaskDefinitions
- ListTasks
- RegisterTaskDefinition
- UpdateService
- Attendi (Amazon EC2)
- Attendi (Amazon ECS)

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Creazione della tua prima funzione serverless

L'esempio di codice seguente mostra come:

- Crea un ruolo IAM per Lambda
- Crea codice funzionale
- Creazione di una funzione Lambda
- Testa la tua funzione Lambda
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash
# AWS Lambda - Create Your First Function
# This script creates a Lambda function, invokes it with a test event,
# views CloudWatch logs, and cleans up all resources.
#
# Source: https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html
#
# Resources created:
#   - IAM role (Lambda execution role with basic logging permissions)
#   - Lambda function (Python 3.13 or Node.js 22.x runtime)
#   - CloudWatch log group (created automatically by Lambda on invocation)

set -eE -o pipefail
```

```
################################################################################
# Setup
################################################################################

UNIQUE_ID=$(head -c 8 /dev/urandom | od -An -tx1 | tr -d ' ')
FUNCTION_NAME="my-lambda-function-${UNIQUE_ID}"
ROLE_NAME="lambda-execution-role-${UNIQUE_ID}"
LOG_GROUP_NAME="/aws/lambda/${FUNCTION_NAME}"

TEMP_DIR=$(mktemp -d)
readonly TEMP_DIR
LOG_FILE="${TEMP_DIR}/lambda-gettingstarted.log"

exec > >(tee -a "$LOG_FILE") 2>&1

declare -a CREATED_RESOURCES

################################################################################
# Helper functions
################################################################################

cleanup_resources() {
    # Disable error trap to prevent recursion during cleanup
    trap - ERR
    set +eE

    echo ""
    echo "Cleaning up resources..."
    echo ""

    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        local RESOURCE="${CREATED_RESOURCES[$i]}"
        local TYPE="${RESOURCE%%:*}"
        local NAME="${RESOURCE#*:}"

        case "$TYPE" in
            log-group)
                echo "Deleting CloudWatch log group: ${NAME}"
                aws logs delete-log-group \
                    --log-group-name "$NAME" 2>&1 || echo "  WARNING: Could not
 delete log group ${NAME}."
                ;;
            lambda-function)
                echo "Deleting Lambda function: ${NAME}"
```

```
                aws lambda delete-function \
                    --function-name "$NAME" 2>&1 || echo "  WARNING: Could not
 delete Lambda function ${NAME}."
                echo "  Waiting for function deletion to complete..."
                local DELETE_WAIT=0
                while aws lambda get-function --function-name "$NAME" > /dev/null
 2>&1; do
                    sleep 2
                    DELETE_WAIT=$((DELETE_WAIT + 2))
                    if [ "$DELETE_WAIT" -ge 60 ]; then
                        echo "  WARNING: Timed out waiting for function
 deletion."
                        break
                    fi
                done
                ;;
            iam-role-policy)
                local ROLE_PART="${NAME%%|*}"
                local POLICY_PART="${NAME#*|}"
                echo "Detaching policy from role: ${ROLE_PART}"
                aws iam detach-role-policy \
                    --role-name "$ROLE_PART" \
                    --policy-arn "$POLICY_PART" 2>&1 || echo "  WARNING: Could
 not detach policy from role ${ROLE_PART}."
                ;;
            iam-role)
                echo "Deleting IAM role: ${NAME}"
                aws iam delete-role \
                    --role-name "$NAME" 2>&1 || echo "  WARNING: Could not delete
 IAM role ${NAME}."
                ;;
        esac
    done

    if [ -d "$TEMP_DIR" ]; then
        rm -rf "$TEMP_DIR"
    fi

    echo ""
    echo "Cleanup complete."
}

handle_error() {
    echo ""
```

```
    echo "========================================="
    echo "ERROR: Script failed at $1"
    echo "========================================="
    echo ""
    if [ ${#CREATED_RESOURCES[@]} -gt 0 ]; then
        echo "Attempting to clean up ${#CREATED_RESOURCES[@]} resource(s)..."
        cleanup_resources
    fi
    exit 1
}

trap 'handle_error "line $LINENO"' ERR

wait_for_resource() {
    local DESCRIPTION="$1"
    local COMMAND="$2"
    local TARGET_VALUE="$3"
    local TIMEOUT=300
    local ELAPSED=0
    local INTERVAL=5

    echo "Waiting for ${DESCRIPTION}..."
    while true; do
        local RESULT
        RESULT=$(eval "$COMMAND" 2>&1) || true
        if echo "$RESULT" | grep -q "$TARGET_VALUE"; then
            echo "  ${DESCRIPTION} is ready."
            return 0
        fi
        if [ "$ELAPSED" -ge "$TIMEOUT" ]; then
            echo "ERROR: Timed out waiting for ${DESCRIPTION} after ${TIMEOUT}
 seconds."
            return 1
        fi
        sleep "$INTERVAL"
        ELAPSED=$((ELAPSED + INTERVAL))
    done
}

validate_input() {
    local input="$1"
    local pattern="$2"
    if ! [[ "$input" =~ $pattern ]]; then
        echo "ERROR: Invalid input: $input"
```

```
        return 1
    fi
    return 0
}

################################################################################
# Region pre-check
################################################################################

CONFIGURED_REGION=$(aws configure get region 2>/dev/null || true)
if [ -z "$CONFIGURED_REGION" ] && [ -z "$AWS_DEFAULT_REGION" ] && [ -z
 "$AWS_REGION" ]; then
    echo "ERROR: No AWS region configured."
    echo "Run 'aws configure set region <region>' or export AWS_DEFAULT_REGION."
    exit 1
fi

################################################################################
# Runtime selection
################################################################################

echo ""
echo "========================================="
echo "AWS Lambda - Create Your First Function"
echo "========================================="
echo ""
echo "Select a runtime for your Lambda function:"
echo "  1) Python 3.13"
echo "  2) Node.js 22.x"
echo ""
echo "Using default: Python 3.13"
RUNTIME_CHOICE="1"

case "$RUNTIME_CHOICE" in
    1)
        RUNTIME="python3.13"
        HANDLER="lambda_function.lambda_handler"
        CODE_FILE="lambda_function.py"
        cat > "${TEMP_DIR}/${CODE_FILE}" << 'PYTHON_EOF'
import json
import logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
def lambda_handler(event, context):
```

```
    if not isinstance(event, dict) or 'length' not in event or 'width' not in
 event:
        raise ValueError('Event must contain length and width')
    try:
        length = float(event['length'])
        width = float(event['width'])
        if length < 0 or width < 0:
            raise ValueError('Length and width must be non-negative')
        area = calculate_area(length, width)
        print(f'The area is {area}')
        logger.info(f'CloudWatch logs group: {context.log_group_name}')
        return json.dumps({'area': area})
    except (TypeError, ValueError) as e:
        logger.error(f'Error processing input: {str(e)}')
        raise
def calculate_area(length, width):
    return length * width
PYTHON_EOF
        echo "Selected runtime: Python 3.13"
        ;;
    2)
        RUNTIME="nodejs22.x"
        HANDLER="index.handler"
        CODE_FILE="index.mjs"
        cat > "${TEMP_DIR}/${CODE_FILE}" << 'NODEJS_EOF'
export const handler = async (event, context) => {
  if (!event || typeof event.length !== 'number' || typeof event.width !==
 'number') {
    throw new Error('Event must contain numeric length and width');
  }
  if (event.length < 0 || event.width < 0) {
    throw new Error('Length and width must be non-negative');
  }
  const area = event.length * event.width;
  console.log(`The area is ${area}`);
  console.log('CloudWatch log group: ', context.logGroupName);
  return JSON.stringify({area});
};
NODEJS_EOF
        echo "Selected runtime: Node.js 22.x"
        ;;
    *)
        echo "ERROR: Invalid choice. Please enter 1 or 2."
        exit 1
```

```
        ;;
esac

################################################################################
# Step 1: Create IAM execution role
################################################################################

echo ""
echo "=========================================="
echo "Step 1: Create IAM execution role"
echo "=========================================="
echo ""

TRUST_POLICY='{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'

echo "Creating IAM role: ${ROLE_NAME}"
ROLE_OUTPUT=$(aws iam create-role \
    --role-name "$ROLE_NAME" \
    --assume-role-policy-document "$TRUST_POLICY" \
    --query 'Role.Arn' \
    --output text 2>&1)

if ! validate_input "$ROLE_OUTPUT" "^arn:aws:iam::[0-9]+:role/"; then
    echo "ERROR: Failed to create IAM role"
    exit 1
fi

echo "$ROLE_OUTPUT"
ROLE_ARN="$ROLE_OUTPUT"
CREATED_RESOURCES+=("iam-role:${ROLE_NAME}")
echo "Role ARN: ${ROLE_ARN}"

aws iam tag-role \
```

```
    --role-name "$ROLE_NAME" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=lambda-gettingstarted

echo ""
echo "Attaching AWSLambdaBasicExecutionRole policy..."
aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>&1
CREATED_RESOURCES+=("iam-role-policy:${ROLE_NAME}|arn:aws:iam::aws:policy/
service-role/AWSLambdaBasicExecutionRole")
echo "Policy attached."

# IAM roles can take a few seconds to propagate
echo "Waiting for IAM role to propagate..."
sleep 10

###############################################################################
# Step 2: Create Lambda function
###############################################################################

echo ""
echo "==========================================="
echo "Step 2: Create Lambda function"
echo "==========================================="
echo ""

echo "Creating deployment package..."
ORIGINAL_DIR=$(pwd)
cd "$TEMP_DIR" || exit 1
zip -j function.zip "$CODE_FILE" > /dev/null 2>&1 || {
    echo "ERROR: Failed to create deployment package"
    exit 1
}
cd "$ORIGINAL_DIR" || exit 1

if [ ! -f "${TEMP_DIR}/function.zip" ]; then
    echo "ERROR: Deployment package creation failed"
    exit 1
fi

echo "Creating Lambda function: ${FUNCTION_NAME}"
echo "  Runtime: ${RUNTIME}"
echo "  Handler: ${HANDLER}"
```

```
echo ""

CREATE_OUTPUT=$(aws lambda create-function \
    --function-name "$FUNCTION_NAME" \
    --runtime "$RUNTIME" \
    --role "$ROLE_ARN" \
    --handler "$HANDLER" \
    --architectures x86_64 \
    --zip-file "fileb://${TEMP_DIR}/function.zip" \
    --tags project=doc-smith,tutorial=lambda-gettingstarted \
    --query '[FunctionName, FunctionArn, Runtime, State]' \
    --output text 2>&1)

if [ -z "$CREATE_OUTPUT" ]; then
    echo "ERROR: Failed to create Lambda function"
    exit 1
fi

echo "$CREATE_OUTPUT"
CREATED_RESOURCES+=("lambda-function:${FUNCTION_NAME}")

wait_for_resource "Lambda function to become Active" \
    "aws lambda get-function-configuration --function-name ${FUNCTION_NAME} --
query State --output text" \
    "Active"

################################################################################
# Step 3: Invoke the function
################################################################################

echo ""
echo "========================================="
echo "Step 3: Invoke the function"
echo "========================================="
echo ""

TEST_EVENT='{"length": 6, "width": 7}'
echo "Invoking function with test event: ${TEST_EVENT}"
echo ""

echo "$TEST_EVENT" > "${TEMP_DIR}/test-event.json"

if ! validate_input "$TEST_EVENT" '"length": [0-9]+, "width": [0-9]+'; then
    echo "ERROR: Invalid test event format"
```

```
    exit 1
fi

INVOKE_OUTPUT=$(aws lambda invoke \
    --function-name "$FUNCTION_NAME" \
    --payload "fileb://${TEMP_DIR}/test-event.json" \
    --cli-read-timeout 30 \
    "${TEMP_DIR}/response.json" 2>&1)
echo "$INVOKE_OUTPUT"

if [ ! -f "${TEMP_DIR}/response.json" ]; then
    echo "ERROR: No response file generated"
    exit 1
fi

RESPONSE=$(cat "${TEMP_DIR}/response.json")
echo ""
echo "Function response: ${RESPONSE}"
echo ""

if echo "$INVOKE_OUTPUT" | grep -qi "functionerror"; then
    echo "WARNING: Function returned an error."
fi

################################################################################
# Step 4: View CloudWatch logs
################################################################################

echo ""
echo "========================================="
echo "Step 4: View CloudWatch Logs"
echo "========================================="
echo ""

echo "Log group: ${LOG_GROUP_NAME}"
echo ""

echo "Waiting for CloudWatch logs to be available..."

LOG_STREAMS=""
for i in $(seq 1 6); do
    LOG_STREAMS=$(aws logs describe-log-streams \
        --log-group-name "$LOG_GROUP_NAME" \
        --order-by LastEventTime \
```

```
        --descending \
        --query 'logStreams[0].logStreamName' \
        --output text 2>/dev/null) || true
    if [ -n "$LOG_STREAMS" ] && [ "$LOG_STREAMS" != "None" ]; then
        break
    fi
    LOG_STREAMS=""
    sleep 5
done

if [ -n "$LOG_STREAMS" ] && [ "$LOG_STREAMS" != "None" ]; then
    echo "Latest log stream: ${LOG_STREAMS}"
    echo ""
    echo "--- Log events ---"
    LOG_EVENTS=$(aws logs get-log-events \
        --log-group-name "$LOG_GROUP_NAME" \
        --log-stream-name "$LOG_STREAMS" \
        --query 'events[].message' \
        --output text 2>&1) || true
    echo "$LOG_EVENTS"
    echo "--- End of log events ---"
else
    echo "No log streams found yet. Logs may take a moment to appear."
    echo "You can view them in the CloudWatch console:"
    echo "  Log group: ${LOG_GROUP_NAME}"
fi

CREATED_RESOURCES+=("log-group:${LOG_GROUP_NAME}")

aws logs tag-log-group \
    --log-group-name "$LOG_GROUP_NAME" \
    --tags project=doc-smith,tutorial=lambda-gettingstarted

################################################################################
# Summary and cleanup
################################################################################

echo ""
echo "========================================="
echo "SUMMARY"
echo "========================================="
echo ""
echo "Resources created:"
echo "  IAM role:          ${ROLE_NAME}"
```

```
echo "  Lambda function:   ${FUNCTION_NAME}"
echo "  CloudWatch logs:   ${LOG_GROUP_NAME}"
echo ""
echo "========================================="
echo "CLEANUP"
echo "========================================="
echo ""
echo "Cleaning up all created resources..."
cleanup_resources

echo ""
echo "Done."
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - AttachRolePolicy
  - CreateFunction
  - CreateRole
  - DeleteFunction
  - DeleteLogGroup
  - DeleteRole
  - DescribeLogStreams
  - DetachRolePolicy
  - GetFunction
  - GetFunctionConfiguration
  - GetLogEvents
  - Invoke

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUsare Logs con un CloudWatch AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scopri le nozioni di base sull'ingestione di CloudWatch Logs syslog utilizzando un AWS SDK

L'esempio di codice seguente mostra come:

- Implementa uno CloudFormation stack che effettua il provisioning di un endpoint VPC syslog.
- Crea un gruppo di log e aggiungi una politica delle risorse per il servizio syslog.
- Crea una configurazione syslog con. PutSyslogConfiguration
- Elenca le configurazioni syslog con e senza filtri.
- Pulisci eliminando la configurazione syslog, il gruppo di log e lo 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 SyslogIngestionScenario:
    """Interactive scenario demonstrating CloudWatch Logs syslog ingestion."""

    STACK_NAME = "syslog-demo-stack"

    def __init__(
        self,
        cloudwatch_logs_wrapper: CloudWatchLogsWrapper,
        cf_client: boto3.client,
    ) -> None:
        """
        :param cloudwatch_logs_wrapper: A CloudWatchLogsWrapper instance.
        :param cf_client: A Boto3 CloudFormation client.
        """
        self.wrapper = cloudwatch_logs_wrapper
        self.cf_client = cf_client
        self.vpc_endpoint_id: Optional[str] = None
        self.log_group_name: Optional[str] = None
        self.log_group_arn: Optional[str] = None
        self.stack_deployed = False

    # --------------------------------------------------------------------
```

```
    # Setup phase
    # ------------------------------------------------------------------
    def deploy_cfn_stack(self) -> str:
        """
        Deploys the CloudFormation stack that provisions the VPC endpoint
        prerequisite. Waits for CREATE_COMPLETE and returns the VpcEndpointId
        output.

        :return: The VPC endpoint ID.
        """
        print("\n" + "=" * 68)
        print("Step 1: Deploy the CloudFormation prerequisite stack")
        print("=" * 68)
        print(
            f"\nDeploying stack '{self.STACK_NAME}' to create a VPC, subnet, "
            "security group, and syslog VPC endpoint.\n"
            "This can take several minutes..."
        )

        try:
            self.cf_client.create_stack(
                StackName=self.STACK_NAME,
                TemplateBody=_load_cfn_template(),
                Capabilities=["CAPABILITY_IAM"],
            )
            # Only this run's own stack should be torn down in cleanup. Mark it
            # as soon as creation is initiated so a stack that fails partway
            # (e.g. ROLLBACK) is still cleaned up.
            self.stack_deployed = True
        except ClientError as error:
            if error.response["Error"]["Code"] == "AlreadyExistsException":
                # The stack pre-existed this run. Reuse it but do NOT mark it
                # for cleanup — deleting a stack we did not create could destroy
                # infrastructure the user set up themselves.
                print(
                    f"Stack '{self.STACK_NAME}' already exists. Reusing it; it "
                    "will not be deleted during cleanup."
                )
            else:
                raise

        waiter = self.cf_client.get_waiter("stack_create_complete")
        try:
            waiter.wait(
```

```
                StackName=self.STACK_NAME,
                WaiterConfig={"Delay": 30, "MaxAttempts": 40},
            )
        except WaiterError:
            # The waiter fails both when the stack genuinely failed
            # (CREATE_FAILED, ROLLBACK_COMPLETE, ...) and, harmlessly, when the
            # stack already existed in a completed state before this run.
            # Inspect the actual status so a real failure surfaces clearly
            # instead of as a confusing "missing outputs" error later.
            status = self.cf_client.describe_stacks(StackName=self.STACK_NAME)[
                "Stacks"
            ][0]["StackStatus"]
            if status not in ("CREATE_COMPLETE", "UPDATE_COMPLETE"):
                raise RuntimeError(
                    f"CloudFormation stack '{self.STACK_NAME}' did not reach a "
                    f"successful state (status: {status}). Check the stack events
"
                    "in the CloudFormation console for the failure reason."
                )

        response = self.cf_client.describe_stacks(StackName=self.STACK_NAME)
        outputs = response["Stacks"][0].get("Outputs", list())
        vpc_endpoint_id = None
        for output in outputs:
            if output["OutputKey"] == "VpcEndpointId":
                vpc_endpoint_id = output["OutputValue"]
                break

        if vpc_endpoint_id is None:
            raise RuntimeError(
                "VpcEndpointId not found in stack outputs. "
                "Check the CloudFormation stack for errors."
            )

        self.vpc_endpoint_id = vpc_endpoint_id
        print(f"\nStack deployed. VPC Endpoint ID: {vpc_endpoint_id}")
        return vpc_endpoint_id

    def create_and_verify_log_group(self) -> None:
        """
        Prompts the user for a log group name, creates it, and verifies it
        with DescribeLogGroups.
        """
        print("\n" + "=" * 68)
```

```
        print("Step 2: Create and verify a log group")
        print("=" * 68)

        name_pattern = re.compile(r"^[a-zA-Z0-9_\-/.#]{1,512}$")

        while True:
            log_group_name = q.ask(
                "\nEnter a name for the syslog log group (e.g. /syslog/demo): "
            )
            if name_pattern.match(log_group_name):
                break
            print("Invalid name. Use 1-512 characters: a-z, A-Z, 0-9, " "_,
-, /, ., #")

        self.log_group_name = log_group_name
        self.wrapper.create_log_group(log_group_name)

        log_groups = self.wrapper.describe_log_groups(
            log_group_name_prefix=log_group_name
        )
        for lg in log_groups:
            if lg.get("logGroupName") == log_group_name:
                creation_ms = lg.get("creationTime", 0)
                creation_dt = datetime.fromtimestamp(
                    creation_ms / 1000, tz=timezone.utc
                )
                self.log_group_arn = lg.get("logGroupArn", "")
                print(f"\nLog group verified:")
                print(f"  Name:    {lg['logGroupName']}")
                print(f"  ARN:     {lg.get('arn', 'N/A')}")
                print(f"  Created: {creation_dt.strftime('%Y-%m-%d %H:%M:%S
UTC')}")
                return

        print(f"\nWARNING: Log group '{log_group_name}' not found in describe
results.")

    def add_resource_policy(self) -> None:
        """
        Adds a resource policy to the log group so the syslog service can
        write to it.
        """
        print("\n" + "=" * 68)
        print("Step 3: Add a resource policy for syslog ingestion")
```

```
        print("=" * 68)

        if self.log_group_arn is None or self.vpc_endpoint_id is None:
            print(
                "WARNING: Cannot add resource policy - missing log group ARN or
VPC endpoint."
            )
            return

        self.wrapper.put_resource_policy(
            log_group_arn=self.log_group_arn,
            vpc_endpoint_id=self.vpc_endpoint_id,
        )
        print(
            "\nResource policy created. The "
            "syslog.logs.amazonaws.com service principal can now call "
            "logs:PutLogEvents and logs:CreateLogStream on the log group."
        )

    # ------------------------------------------------------------------
    # Create syslog configuration phase
    # ------------------------------------------------------------------
    def create_syslog_configuration(self) -> None:
        """
        Creates a syslog configuration that associates the VPC endpoint with
        the log group.
        """
        print("\n" + "=" * 68)
        print("Step 4: Create the syslog configuration")
        print("=" * 68)

        print(
            "\nAssociating the VPC endpoint with the log group so that syslog "
            "messages arriving at the endpoint are stored in the log group."
        )
        print(
            "CloudWatch Logs parses common syslog formats (RFC 5424, "
            "RFC 3164, Cisco FTD/ASA) and extracts fields such as "
            "facility, severity, hostname, and appName.\n"
        )

        self.wrapper.put_syslog_configuration(
            log_group_identifier=self.log_group_name,
            vpc_endpoint_id=self.vpc_endpoint_id,
```

```
        )

        print(
            f"\nSyslog ingestion is now enabled for log group "
            f"'{self.log_group_name}' through VPC endpoint "
            f"'{self.vpc_endpoint_id}'."
        )
        print("\nSupported transport options:")
        print("  - TCP with TLS on port 6514 (recommended)")
        print("  - Plaintext TCP on port 1514")
        print("  - UDP on port 514")

    # ------------------------------------------------------------------
    # List syslog configurations phase
    # ------------------------------------------------------------------
    def _display_configurations(self, configs: list) -> None:
        """Helper to display a list of syslog configurations."""
        if not configs:
            print("  (no configurations found)")
            return
        for cfg in configs:
            created_ms = cfg.get("createdAt", 0)
            created_dt = datetime.fromtimestamp(created_ms / 1000,
tz=timezone.utc)
            print(f"  Log Group ARN:    {cfg.get('logGroupArn', 'N/A')}")
            print(f"  VPC Endpoint ID:  {cfg.get('vpcEndpointId', 'N/A')}")
            print(f"  Source Type:      {cfg.get('sourceType', 'N/A')}")
            print(f"  Created At:       {created_dt.strftime('%Y-%m-%d %H:%M:%S
UTC')}")
            print()

    def list_all_configurations(self) -> None:
        """Lists all syslog configurations in the account (no filter)."""
        print("\n" + "=" * 68)
        print("Step 5a: List all syslog configurations (no filter)")
        print("=" * 68)

        configs = self.wrapper.list_syslog_configurations()
        print(f"\nAll syslog configurations ({len(configs)} found):\n")
        self._display_configurations(configs)

    def list_configurations_by_log_group(self) -> None:
        """Lists syslog configurations filtered by the scenario's log group."""
        print("\n" + "=" * 68)
```

```
        print("Step 5b: List configurations filtered by log group")
        print("=" * 68)

        configs = self.wrapper.list_syslog_configurations(
            log_group_identifier=self.log_group_name,
        )
        print(
            f"\nConfigurations for log group '{self.log_group_name}' "
            f"({len(configs)} found):\n"
        )
        self._display_configurations(configs)

    def list_configurations_by_vpc_endpoint(self) -> None:
        """Lists syslog configurations filtered by VPC endpoint."""
        print("\n" + "=" * 68)
        print("Step 5c: List configurations filtered by VPC endpoint")
        print("=" * 68)

        configs = self.wrapper.list_syslog_configurations(
            vpc_endpoint_id=self.vpc_endpoint_id,
        )
        print(
            f"\nConfigurations for VPC endpoint '{self.vpc_endpoint_id}' "
            f"({len(configs)} found):\n"
        )
        self._display_configurations(configs)

    # ------------------------------------------------------------------
    # Cleanup phase
    # ------------------------------------------------------------------
    def cleanup(self, prompt: bool = True) -> None:
        """
        Cleans up the resources created during the scenario. Tolerates
        resources that were never created or are already gone, so it is safe to
        call after a partial failure.

        :param prompt: When True (the normal end-of-scenario path), ask the user
            to confirm before deleting. When False (best-effort cleanup after an
            error), skip the prompt and delete whatever was created.
        """
        print("\n" + "=" * 68)
        print("Cleanup")
        print("=" * 68)
```

```
        if prompt:
            do_cleanup = q.ask(
                "\nDo you want to delete all resources created during this "
                "scenario? (y/n) ",
                q.is_yesno,
            )
            # q.is_yesno converts the answer to a bool (True when the user
            # answered 'y'); check it directly rather than calling string
methods.
            if not do_cleanup:
                print("Skipping cleanup. Resources remain in your account.")
                return
        else:
            print(
                "\nAn error occurred; attempting best-effort cleanup of any "
                "resources that were created..."
            )

        # Delete syslog configuration
        if self.log_group_name and self.vpc_endpoint_id:
            print(
                f"\nDeleting syslog configuration for log group "
                f"'{self.log_group_name}'..."
            )
            self.wrapper.delete_syslog_configuration(
                log_group_identifier=self.log_group_name,
                vpc_endpoint_id=self.vpc_endpoint_id,
            )
            print(
                "  After deletion, syslog data is no longer ingested through "
                "the VPC endpoint into this log group."
            )

            # Verify deletion
            configs = self.wrapper.list_syslog_configurations(
                log_group_identifier=self.log_group_name,
            )
            if not configs:
                print("  Verified: configuration has been removed.")
            else:
                print(
                    f"  WARNING: Still found {len(configs)} configuration(s). "
                    "They may take a moment to be fully removed."
                )
```

```
        # Delete log group
        if self.log_group_name:
            print(f"\nDeleting log group '{self.log_group_name}'...")
            self.wrapper.delete_log_group(self.log_group_name)
            print(
                "  The log group and all archived log events have been "
                "permanently deleted."
            )

        # Delete CloudFormation stack
        if self.stack_deployed:
            print(f"\nDeleting CloudFormation stack '{self.STACK_NAME}'...")
            try:
                self.cf_client.delete_stack(StackName=self.STACK_NAME)
                waiter = self.cf_client.get_waiter("stack_delete_complete")
                waiter.wait(
                    StackName=self.STACK_NAME,
                    WaiterConfig={"Delay": 30, "MaxAttempts": 40},
                )
                print("  Stack deleted successfully.")
            except (ClientError, WaiterError) as error:
                logger.warning("Stack deletion issue: %s", error)
                print(
                    f"  WARNING: Stack deletion may still be in progress. "
                    f"Check the CloudFormation console."
                )

    # ------------------------------------------------------------------
    # Run the full scenario
    # ------------------------------------------------------------------
    def run_scenario(self) -> None:
        """Runs all phases of the syslog ingestion scenario."""
        print("\n" + "=" * 68)
        print("CloudWatch Logs Syslog Ingestion Scenario")
        print("=" * 68)
        print(
            "\nThis scenario demonstrates CloudWatch Logs managed syslog "
            "ingestion, which lets you route syslog data from firewalls, "
            "routers, switches, and Linux servers directly into CloudWatch "
            "Logs through a VPC endpoint - without running a separate "
            "collection tier.\n"
        )
```

```
        succeeded = False
        try:
            # Setup
            self.deploy_cfn_stack()
            self.create_and_verify_log_group()
            self.add_resource_policy()

            # Create syslog configuration
            self.create_syslog_configuration()

            q.ask("\nPress Enter to continue to listing configurations...")

            # List syslog configurations
            self.list_all_configurations()
            self.list_configurations_by_log_group()
            self.list_configurations_by_vpc_endpoint()
            succeeded = True

            # On success, prompt before cleaning up.
            self.cleanup(prompt=True)
        finally:
            # On failure, run best-effort cleanup without prompting so that
            # partially-created resources are not left behind.
            if not succeeded:
                self.cleanup(prompt=False)

        print("\n" + "=" * 68)
        print("Scenario complete!")
        print("=" * 68)
        print(
            "\nYou have successfully demonstrated CloudWatch Logs managed "
            "syslog ingestion:\n"
            "  - Deployed a CloudFormation stack with a syslog VPC endpoint.\n"
            "  - Created a log group and authorized the syslog service.\n"
            "  - Associated a syslog configuration with PutSyslogConfiguration.
\n"
            "  - Listed configurations with and without filters.\n"
            "  - Cleaned up all resources.\n"
        )
```

Crea una classe che racchiuda le operazioni di CloudWatch Logs.

```
class CloudWatchLogsWrapper:
    """Encapsulates Amazon CloudWatch Logs actions."""

    def __init__(self, logs_client: boto3.client) -> None:
        """
        Initializes the CloudWatchLogsWrapper with a CloudWatch Logs client.

        :param logs_client: A Boto3 CloudWatch Logs client.
        """
        self.logs_client = logs_client

    @classmethod
    def from_client(cls) -> "CloudWatchLogsWrapper":
        """
        Creates a CloudWatchLogsWrapper instance with a default CloudWatch Logs
 client.

        :return: An instance of CloudWatchLogsWrapper.
        """
        logs_client = boto3.client("logs")
        return cls(logs_client)
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - CreateLogGroup
  - DeleteLogGroup
  - DeleteSyslogConfiguration
  - DescribeLogGroups
  - ListSyslogConfigurations
  - PutResourcePolicy
  - PutSyslogConfiguration

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Usa CloudWatch Logs per eseguire una query di grandi dimensioni

I seguenti esempi di codice mostrano come utilizzare CloudWatch Logs per interrogare più di 10.000 record.

.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.

Questo è il flusso di lavoro principale che illustra lo scenario di query di grandi dimensioni.

```
using System.Diagnostics;
using System.Text.RegularExpressions;
using Amazon.CloudFormation;
using Amazon.CloudFormation.Model;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
using CloudWatchLogsActions;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;

namespace CloudWatchLogsScenario;

public class LargeQueryWorkflow
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.
    This .NET code example performs the following tasks for the CloudWatch Logs
 Large Query workflow:

    1. Prepare the Application:
       - Prompt the user to deploy CloudFormation stack and generate sample logs.
       - Deploy the CloudFormation template for resource creation.
       - Generate 50,000 sample log entries using CloudWatch Logs API.
```

```
       - Wait 5 minutes for logs to be fully ingested.

    2. Execute Large Query:
       - Perform recursive queries to retrieve all logs using binary search.
       - Display progress for each query executed.
       - Show total execution time and logs found.

    3. Clean up:
       - Prompt the user to delete the CloudFormation stack and all resources.
       - Destroy the CloudFormation stack and wait until removed.
    */

    public static ILogger<LargeQueryWorkflow> _logger = null!;
    public static CloudWatchLogsWrapper _wrapper = null!;
    public static IAmazonCloudFormation _amazonCloudFormation = null!;

    private static string _logGroupName = "/workflows/cloudwatch-logs/large-
query";
    private static string _logStreamName = "stream1";
    private static long _queryStartDate;
    private static long _queryEndDate;

    public static bool _interactive = true;
    public static string _stackName = "CloudWatchLargeQueryStack";
    private static string _stackResourcePath = "../../../../../../../scenarios/
features/cloudwatch_logs_large_query/resources/stack.yaml";

    public static async Task Main(string[] args)
    {
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter("Microsoft", LogLevel.Information))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonCloudWatchLogs>()
                    .AddAWSService<IAmazonCloudFormation>()
                    .AddTransient<CloudWatchLogsWrapper>()
            )
            .Build();

        if (_interactive)
        {
            _logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
                .CreateLogger<LargeQueryWorkflow>();
```

```
            _wrapper = host.Services.GetRequiredService<CloudWatchLogsWrapper>();
            _amazonCloudFormation =
host.Services.GetRequiredService<IAmazonCloudFormation>();
        }

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the CloudWatch Logs Large Query
Scenario.");
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("This scenario demonstrates how to perform large-scale
queries on");
        Console.WriteLine("CloudWatch Logs using recursive binary search to
retrieve more than");
        Console.WriteLine("the 10,000 result limit.");
        Console.WriteLine();

        try
        {
            Console.WriteLine(new string('-', 80));
            var prepareSuccess = await PrepareApplication();
            Console.WriteLine(new string('-', 80));

            if (prepareSuccess)
            {
                Console.WriteLine(new string('-', 80));
                await ExecuteLargeQuery();
                Console.WriteLine(new string('-', 80));
            }

            Console.WriteLine(new string('-', 80));
            await Cleanup();
            Console.WriteLine(new string('-', 80));
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, "There was a problem with the scenario,
initiating cleanup...");
            _interactive = false;
            await Cleanup();
        }

        Console.WriteLine("CloudWatch Logs Large Query scenario completed.");
    }
```

```
   /// <summary>
   /// Runs the scenario workflow. Used for testing.
   /// </summary>
   public static async Task RunScenario()
   {
       Console.WriteLine(new string('-', 80));
       Console.WriteLine("Welcome to the CloudWatch Logs Large Query
Scenario.");
       Console.WriteLine(new string('-', 80));
       Console.WriteLine("This scenario demonstrates how to perform large-scale
queries on");
       Console.WriteLine("CloudWatch Logs using recursive binary search to
retrieve more than");
       Console.WriteLine("the 10,000 result limit.");
       Console.WriteLine();

       try
       {
           Console.WriteLine(new string('-', 80));
           var prepareSuccess = await PrepareApplication();
           Console.WriteLine(new string('-', 80));

           if (prepareSuccess)
           {
               Console.WriteLine(new string('-', 80));
               await ExecuteLargeQuery();
               Console.WriteLine(new string('-', 80));
           }

           Console.WriteLine(new string('-', 80));
           await Cleanup();
           Console.WriteLine(new string('-', 80));
       }
       catch (Exception ex)
       {
           _logger.LogError(ex, "There was a problem with the scenario,
initiating cleanup...");
           _interactive = false;
           await Cleanup();
       }

       Console.WriteLine("CloudWatch Logs Large Query scenario completed.");
   }
```

```
    /// <summary>
    /// Prepares the application by creating the necessary resources.
    /// </summary>
    /// <returns>True if the application was prepared successfully.</returns>
    public static async Task<bool> PrepareApplication()
    {
        Console.WriteLine("Preparing the application...");
        Console.WriteLine();

        try
        {
            var deployStack = !_interactive || GetYesNoResponse(
                "Would you like to deploy the CloudFormation stack and generate
sample logs? (y/n) ");

            if (deployStack)
            {
                if (_interactive)
                {
                    Console.Write(
                        $"Enter a path for the CloudFormation stack
resource .yaml file (or press Enter for default '{_stackResourcePath}'): ");
                    string? inputPath = Console.ReadLine();
                    if (!string.IsNullOrWhiteSpace(inputPath))
                    {
                        _stackResourcePath = inputPath;
                    }
                }

                _stackName = PromptUserForStackName();

                var deploySuccess = await DeployCloudFormationStack(_stackName);

                if (deploySuccess)
                {
                    Console.WriteLine();
                    Console.WriteLine("Generating 50,000 sample log entries...");
                    var generateSuccess = await GenerateSampleLogs();

                    if (generateSuccess)
                    {
                        Console.WriteLine();
```

```
                        Console.WriteLine("Sample logs created. Waiting 5 minutes
for logs to be fully ingested...");
                        await WaitWithCountdown(300);

                        Console.WriteLine("Application preparation complete.");
                        return true;
                    }
                }
            }
            else
            {
                _logGroupName = PromptUserForInput("Enter the log group name ",
_logGroupName);
                _logStreamName = PromptUserForInput("Enter the log stream name ",
_logStreamName);

                var startDateMs = PromptUserForLong("Enter the query start date
(milliseconds since epoch): ");
                var endDateMs = PromptUserForLong("Enter the query end date
(milliseconds since epoch): ");

                _queryStartDate = startDateMs / 1000;
                _queryEndDate = endDateMs / 1000;

                Console.WriteLine("Application preparation complete.");
                return true;
            }
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, "An error occurred while preparing the
application.");
        }

        Console.WriteLine("Application preparation failed.");
        return false;
    }

    /// <summary>
    /// Deploys the CloudFormation stack with the necessary resources.
    /// </summary>
    /// <param name="stackName">The name of the CloudFormation stack.</param>
    /// <returns>True if the stack was deployed successfully.</returns>
    private static async Task<bool> DeployCloudFormationStack(string stackName)
```

```
    {
        Console.WriteLine($"\nDeploying CloudFormation stack: {stackName}");

        try
        {
            var request = new CreateStackRequest
            {
                StackName = stackName,
                TemplateBody = await File.ReadAllTextAsync(_stackResourcePath)
            };

            var response = await _amazonCloudFormation.CreateStackAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"CloudFormation stack creation started:
{stackName}");

                bool stackCreated = await
WaitForStackCompletion(response.StackId);

                if (stackCreated)
                {
                    Console.WriteLine("CloudFormation stack created
successfully.");
                    return true;
                }
                else
                {
                    _logger.LogError($"CloudFormation stack creation failed:
{stackName}");
                    return false;
                }
            }
            else
            {
                _logger.LogError($"Failed to create CloudFormation stack:
{stackName}");
                return false;
            }
        }
        catch (AlreadyExistsException)
        {
```

```
            _logger.LogWarning($"CloudFormation stack '{stackName}' already
exists. Please provide a unique name.");
            var newStackName = PromptUserForStackName();
            return await DeployCloudFormationStack(newStackName);
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"An error occurred while deploying the
CloudFormation stack: {stackName}");
            return false;
        }
    }

    /// <summary>
    /// Waits for the CloudFormation stack to be in the CREATE_COMPLETE state.
    /// </summary>
    /// <param name="stackId">The ID of the CloudFormation stack.</param>
    /// <returns>True if the stack was created successfully.</returns>
    private static async Task<bool> WaitForStackCompletion(string stackId)
    {
        int retryCount = 0;
        const int maxRetries = 30;
        const int retryDelay = 10000;

        while (retryCount < maxRetries)
        {
            var describeStacksRequest = new DescribeStacksRequest
            {
                StackName = stackId
            };

            var describeStacksResponse = await
_amazonCloudFormation.DescribeStacksAsync(describeStacksRequest);

            if (describeStacksResponse.Stacks.Count > 0)
            {
                if (describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.CREATE_COMPLETE)
                {
                    return true;
                }
                if (describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.CREATE_FAILED ||
```

```
                    describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.ROLLBACK_COMPLETE)
                {
                    return false;
                }
            }

            Console.WriteLine("Waiting for CloudFormation stack creation to
complete...");
            await Task.Delay(retryDelay);
            retryCount++;
        }

        _logger.LogError("Timed out waiting for CloudFormation stack creation to
complete.");
        return false;
    }

    /// <summary>
    /// Generates sample logs directly using CloudWatch Logs API.
    /// Creates 50,000 log entries spanning 5 minutes.
    /// </summary>
    /// <returns>True if logs were generated successfully.</returns>
    private static async Task<bool> GenerateSampleLogs()
    {
        const int totalEntries = 50000;
        const int entriesPerBatch = 10000;
        const int fiveMinutesMs = 5 * 60 * 1000;

        try
        {
            // Calculate timestamps
            var startTimeMs = DateTimeOffset.UtcNow.ToUnixTimeMilliseconds();
            var timestampIncrement = fiveMinutesMs / totalEntries;

            Console.WriteLine($"Generating {totalEntries} log entries...");

            var entryCount = 0;
            var currentTimestamp = startTimeMs;
            var numBatches = totalEntries / entriesPerBatch;

            // Generate and upload logs in batches
            for (int batchNum = 0; batchNum < numBatches; batchNum++)
            {
```

```
                var logEvents = new List<InputLogEvent>();

                for (int i = 0; i < entriesPerBatch; i++)
                {
                    logEvents.Add(new InputLogEvent
                    {
                        Timestamp =
 DateTimeOffset.FromUnixTimeMilliseconds(currentTimestamp).UtcDateTime,
                        Message = $"Entry {entryCount}"
                    });

                    entryCount++;
                    currentTimestamp += timestampIncrement;
                }

                // Upload batch
                var success = await _wrapper.PutLogEventsAsync(_logGroupName,
 _logStreamName, logEvents);
                if (!success)
                {
                    _logger.LogError($"Failed to upload batch {batchNum + 1}/
{numBatches}");
                    return false;
                }

                Console.WriteLine($"Uploaded batch {batchNum + 1}/{numBatches}");
            }

            // Set query date range (convert milliseconds to seconds for query
 API)
            _queryStartDate = startTimeMs / 1000;
            _queryEndDate = (currentTimestamp - timestampIncrement) / 1000;

            Console.WriteLine($"Query start date:
 {DateTimeOffset.FromUnixTimeSeconds(_queryStartDate):yyyy-MM-
ddTHH:mm:ss.fffZ}");
            Console.WriteLine($"Query end date:
 {DateTimeOffset.FromUnixTimeSeconds(_queryEndDate):yyyy-MM-ddTHH:mm:ss.fffZ}");
            Console.WriteLine($"Successfully uploaded {totalEntries} log
 entries");

            return true;
        }
        catch (Exception ex)
```

```
        {
            _logger.LogError(ex, "An error occurred while generating sample
logs.");
            return false;
        }
    }

    /// <summary>
    /// Executes the large query workflow.
    /// </summary>
    public static async Task ExecuteLargeQuery()
    {
        Console.WriteLine("Starting recursive query to retrieve all logs...");
        Console.WriteLine();

        var queryLimit = PromptUserForInteger("Enter the query limit (max 10000)
", 10000);
        if (queryLimit > 10000) queryLimit = 10000;

        var queryString = "fields @timestamp, @message | sort @timestamp asc";

        var stopwatch = Stopwatch.StartNew();
        var allResults = await PerformLargeQuery(_logGroupName, queryString,
_queryStartDate, _queryEndDate, queryLimit);
        stopwatch.Stop();

        Console.WriteLine();
        Console.WriteLine($"Queries finished in
{stopwatch.Elapsed.TotalSeconds:F3} seconds.");
        Console.WriteLine($"Total logs found: {allResults.Count}");

        // Check for duplicates
        Console.WriteLine();
        Console.WriteLine("Checking for duplicate logs...");
        var duplicates = FindDuplicateLogs(allResults);
        if (duplicates.Count > 0)
        {
            Console.WriteLine($"WARNING: Found {duplicates.Count} duplicate log
entries!");
            Console.WriteLine("Duplicate entries (showing first 10):");
            foreach (var dup in duplicates.Take(10))
            {
                Console.WriteLine($"  [{dup.Timestamp}] {dup.Message} (appears
{dup.Count} times)");
```

```
            }

            var uniqueCount = allResults.Count - duplicates.Sum(d => d.Count -
1);
            Console.WriteLine($"Unique logs: {uniqueCount}");
        }
        else
        {
            Console.WriteLine("No duplicates found. All logs are unique.");
        }
        Console.WriteLine();

        var viewSample = !_interactive || GetYesNoResponse("Would you like to see
a sample of the logs? (y/n) ");
        if (viewSample)
        {
            Console.WriteLine();
            Console.WriteLine($"Sample logs (first 10 of {allResults.Count}):");
            for (int i = 0; i < Math.Min(10, allResults.Count); i++)
            {
                var timestamp = allResults[i].Find(f => f.Field ==
"@timestamp")?.Value ?? "N/A";
                var message = allResults[i].Find(f => f.Field ==
"@message")?.Value ?? "N/A";
                Console.WriteLine($"[{timestamp}] {message}");
            }
        }
    }

    /// <summary>
    /// Performs a large query using recursive binary search.
    /// </summary>
    private static async Task<List<List<ResultField>>> PerformLargeQuery(
        string logGroupName,
        string queryString,
        long startTime,
        long endTime,
        int limit)
    {
        var queryId = await _wrapper.StartQueryAsync(logGroupName, queryString,
startTime, endTime, limit);
        if (queryId == null)
        {
            return new List<List<ResultField>>();
```

```
        }

        var results = await PollQueryResults(queryId);
        if (results == null || results.Count == 0)
        {
            return new List<List<ResultField>>();
        }

        var startDate =
 DateTimeOffset.FromUnixTimeSeconds(startTime).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
        var endDate = DateTimeOffset.FromUnixTimeSeconds(endTime).ToString("yyyy-
MM-ddTHH:mm:ss.fffZ");
        Console.WriteLine($"Query date range: {startDate} ({startTime}s) to
 {endDate} ({endTime}s). Found {results.Count} logs.");

        if (results.Count < limit)
        {
            Console.WriteLine($"  -> Returning {results.Count} logs (less than
 limit of {limit})");
            return results;
        }

        Console.WriteLine($"  -> Hit limit of {limit}. Need to split and
 recurse.");

        // Get the timestamp of the last log (sorted to find the actual last one)
        var lastLogTimestamp = GetLastLogTimestamp(results);
        if (lastLogTimestamp == null)
        {
            Console.WriteLine($"  -> No timestamp found in results. Returning
 {results.Count} logs.");
            return results;
        }

        Console.WriteLine($"  -> Last log timestamp: {lastLogTimestamp}");

        // Parse the timestamp and add 1 millisecond to avoid querying the same
 log again
        var lastLogDate = DateTimeOffset.Parse(lastLogTimestamp + " +0000");
        Console.WriteLine($"  -> Last log as DateTimeOffset: {lastLogDate:yyyy-
MM-ddTHH:mm:ss.fffZ} ({lastLogDate.ToUnixTimeSeconds()}s)");

        var offsetLastLogDate = lastLogDate.AddMilliseconds(1);
```

```
        Console.WriteLine($"  -> Offset timestamp (last
 + 1ms): {offsetLastLogDate:yyyy-MM-ddTHH:mm:ss.fffZ}
 ({offsetLastLogDate.ToUnixTimeSeconds()}s)");

        // Convert to seconds, but round UP to the next second to avoid
 overlapping with logs in the same second
        // This ensures we don't re-query logs that share the same second as the
 last log
        var offsetLastLogTime = offsetLastLogDate.ToUnixTimeSeconds();
        if (offsetLastLogDate.Millisecond > 0)
        {
            offsetLastLogTime++; // Move to the next full second
            Console.WriteLine($"  -> Adjusted to next full second:
 {offsetLastLogTime}s
 ({DateTimeOffset.FromUnixTimeSeconds(offsetLastLogTime):yyyy-MM-
ddTHH:mm:ss.fffZ})");
        }

        Console.WriteLine($"  -> Comparing:
 offsetLastLogTime={offsetLastLogTime}s vs endTime={endTime}s");
        Console.WriteLine($"  -> End time as date:
 {DateTimeOffset.FromUnixTimeSeconds(endTime):yyyy-MM-ddTHH:mm:ss.fffZ}");

        // Check if there's any time range left to query
        if (offsetLastLogTime >= endTime)
        {
            Console.WriteLine($"  -> No time range left to query. Offset time
 ({offsetLastLogTime}s) >= end time ({endTime}s)");
            return results;
        }

        // Split the remaining date range in half
        var (range1Start, range1End, range2Start, range2End) =
 SplitDateRange(offsetLastLogTime, endTime);

        var range1StartDate =
 DateTimeOffset.FromUnixTimeSeconds(range1Start).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
        var range1EndDate =
 DateTimeOffset.FromUnixTimeSeconds(range1End).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
        var range2StartDate =
 DateTimeOffset.FromUnixTimeSeconds(range2Start).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
```

```
        var range2EndDate =
 DateTimeOffset.FromUnixTimeSeconds(range2End).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");

        Console.WriteLine($"  -> Splitting remaining range:");
        Console.WriteLine($"     Range 1: {range1StartDate} ({range1Start}s) to
 {range1EndDate} ({range1End}s)");
        Console.WriteLine($"     Range 2: {range2StartDate} ({range2Start}s) to
 {range2EndDate} ({range2End}s)");

        // Query both halves recursively
        Console.WriteLine($"  -> Querying range 1...");
        var results1 = await PerformLargeQuery(logGroupName, queryString,
 range1Start, range1End, limit);
        Console.WriteLine($"  -> Range 1 returned {results1.Count} logs");

        Console.WriteLine($"  -> Querying range 2...");
        var results2 = await PerformLargeQuery(logGroupName, queryString,
 range2Start, range2End, limit);
        Console.WriteLine($"  -> Range 2 returned {results2.Count} logs");

        // Combine all results
        var allResults = new List<List<ResultField>>(results);
        allResults.AddRange(results1);
        allResults.AddRange(results2);

        Console.WriteLine($"  -> Combined total: {allResults.Count} logs
 ({results.Count} + {results1.Count} + {results2.Count})");

        return allResults;
    }

    /// <summary>
    /// Gets the timestamp string of the most recent log from a list of logs.
    /// Sorts timestamps to find the actual last one.
    /// </summary>
    private static string? GetLastLogTimestamp(List<List<ResultField>> logs)
    {
        var timestamps = logs
            .Select(log => log.Find(f => f.Field == "@timestamp")?.Value)
            .Where(t => !string.IsNullOrEmpty(t))
            .OrderBy(t => t)
            .ToList();
```

```
        if (timestamps.Count == 0)
        {
            return null;
        }

        return timestamps[timestamps.Count - 1];
    }

    /// <summary>
    /// Splits a date range in half.
    /// Range 2 starts at midpoint + 1 second to avoid overlap.
    /// </summary>
    private static (long range1Start, long range1End, long range2Start, long
range2End) SplitDateRange(long startTime, long endTime)
    {
        var midpoint = startTime + (endTime - startTime) / 2;
        // Range 2 starts at midpoint + 1 to avoid querying the same second twice
        return (startTime, midpoint, midpoint + 1, endTime);
    }

    /// <summary>
    /// Polls for query results until complete.
    /// </summary>
    private static async Task<List<List<ResultField>>?> PollQueryResults(string
queryId)
    {
        int retryCount = 0;
        const int maxRetries = 60;
        const int retryDelay = 1000;

        while (retryCount < maxRetries)
        {
            var response = await _wrapper.GetQueryResultsAsync(queryId);
            if (response == null)
            {
                return null;
            }

            if (response.Status == QueryStatus.Complete)
            {
                return response.Results;
            }

            if (response.Status == QueryStatus.Failed ||
```

```
                response.Status == QueryStatus.Cancelled ||
                response.Status == QueryStatus.Timeout ||
                response.Status == QueryStatus.Unknown)
            {
                _logger.LogError($"Query failed with status: {response.Status}");
                return null;
            }

            await Task.Delay(retryDelay);
            retryCount++;
        }

        _logger.LogError("Timed out waiting for query results.");
        return null;
    }

    /// <summary>
    /// Cleans up the resources created during the scenario.
    /// </summary>
    public static async Task<bool> Cleanup()
    {
        var cleanup = !_interactive || GetYesNoResponse(
            "Do you want to delete the CloudFormation stack and all resources?
(y/n) ");

        if (cleanup)
        {
            try
            {
                var stackDeleteSuccess = await
DeleteCloudFormationStack(_stackName, false);
                return stackDeleteSuccess;
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, "An error occurred while cleaning up the
resources.");
                return false;
            }
        }

        Console.WriteLine($"Resources will remain. Stack name: {_stackName}, Log
group: {_logGroupName}");
```

```
        _logger.LogInformation("CloudWatch Logs Large Query scenario is
complete.");
        return true;
    }

    /// <summary>
    /// Deletes the CloudFormation stack and waits for confirmation.
    /// </summary>
    private static async Task<bool> DeleteCloudFormationStack(string stackName,
bool forceDelete)
    {
        var request = new DeleteStackRequest
        {
            StackName = stackName,
        };

        if (forceDelete)
        {
            request.DeletionMode = DeletionMode.FORCE_DELETE_STACK;
        }

        await _amazonCloudFormation.DeleteStackAsync(request);
        Console.WriteLine($"CloudFormation stack '{stackName}' is being deleted.
This may take a few minutes.");

        bool stackDeleted = await WaitForStackDeletion(stackName, forceDelete);

        if (stackDeleted)
        {
            Console.WriteLine($"CloudFormation stack '{stackName}' has been
deleted.");
            return true;
        }
        else
        {
            _logger.LogError($"Failed to delete CloudFormation stack
'{stackName}'.");
            return false;
        }
    }

    /// <summary>
    /// Waits for the stack to be deleted.
    /// </summary>
```

```
   private static async Task<bool> WaitForStackDeletion(string stackName, bool
forceDelete)
   {
       int retryCount = 0;
       const int maxRetries = 30;
       const int retryDelay = 10000;

       while (retryCount < maxRetries)
       {
           var describeStacksRequest = new DescribeStacksRequest
           {
               StackName = stackName
           };

           try
           {
               var describeStacksResponse = await
_amazonCloudFormation.DescribeStacksAsync(describeStacksRequest);

               if (describeStacksResponse.Stacks.Count == 0 ||
                   describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.DELETE_COMPLETE)
               {
                   return true;
               }

               if (!forceDelete && describeStacksResponse.Stacks[0].StackStatus
== StackStatus.DELETE_FAILED)
               {
                   return await DeleteCloudFormationStack(stackName, true);
               }
           }
           catch (AmazonCloudFormationException ex) when (ex.ErrorCode ==
"ValidationError")
           {
               return true;
           }

           Console.WriteLine($"Waiting for CloudFormation stack '{stackName}' to
be deleted...");
           await Task.Delay(retryDelay);
           retryCount++;
       }
```

```
        _logger.LogError($"Timed out waiting for CloudFormation stack
'{stackName}' to be deleted.");
        return false;
    }

    /// <summary>
    /// Waits with a countdown display.
    /// </summary>
    private static async Task WaitWithCountdown(int seconds)
    {
        for (int i = seconds; i > 0; i--)
        {
            Console.Write($"\rWaiting: {i} seconds remaining...  ");
            await Task.Delay(1000);
        }
        Console.WriteLine("\rWait complete.                      ");
    }

    /// <summary>
    /// Helper method to get a yes or no response from the user.
    /// </summary>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null && ynResponse.Equals("y",
StringComparison.InvariantCultureIgnoreCase);
        return response;
    }

    /// <summary>
    /// Prompts the user for a stack name.
    /// </summary>
    private static string PromptUserForStackName()
    {
        if (_interactive)
        {
            Console.Write($"Enter a name for the CloudFormation stack (press
Enter for default '{_stackName}'): ");
            string? input = Console.ReadLine();
            if (!string.IsNullOrWhiteSpace(input))
            {
                var regex = "[a-zA-Z][-a-zA-Z0-9]*";
                if (!Regex.IsMatch(input, regex))
```

```
                {
                    Console.WriteLine($"Invalid stack name. Using default:
{_stackName}");
                    return _stackName;
                }
                return input;
            }
        }
        return _stackName;
    }

    /// <summary>
    /// Prompts the user for input with a default value.
    /// </summary>
    private static string PromptUserForInput(string prompt, string defaultValue)
    {
        if (_interactive)
        {
            Console.Write($"{prompt}(press Enter for default '{defaultValue}'):
");
            string? input = Console.ReadLine();
            return string.IsNullOrWhiteSpace(input) ? defaultValue : input;
        }
        return defaultValue;
    }

    /// <summary>
    /// Prompts the user for an integer value.
    /// </summary>
    private static int PromptUserForInteger(string prompt, int defaultValue)
    {
        if (_interactive)
        {
            Console.Write($"{prompt}(press Enter for default '{defaultValue}'):
");
            string? input = Console.ReadLine();
            if (string.IsNullOrWhiteSpace(input) || !int.TryParse(input, out var
result))
            {
                return defaultValue;
            }
            return result;
        }
        return defaultValue;
```

```
    }

    /// <summary>
    /// Prompts the user for a long value.
    /// </summary>
    private static long PromptUserForLong(string prompt)
    {
        if (_interactive)
        {
            Console.Write(prompt);
            string? input = Console.ReadLine();
            if (long.TryParse(input, out var result))
            {
                return result;
            }
        }
        return 0;
    }

    /// <summary>
    /// Finds duplicate log entries based on timestamp and message.
    /// </summary>
    private static List<(string Timestamp, string Message, int Count)>
FindDuplicateLogs(List<List<ResultField>> logs)
    {
        var logSignatures = new Dictionary<string, int>();

        foreach (var log in logs)
        {
            var timestamp = log.Find(f => f.Field == "@timestamp")?.Value ?? "";
            var message = log.Find(f => f.Field == "@message")?.Value ?? "";
            var signature = $"{timestamp}|{message}";

            if (logSignatures.ContainsKey(signature))
            {
                logSignatures[signature]++;
            }
            else
            {
                logSignatures[signature] = 1;
            }
        }

        return logSignatures
```

```
            .Where(kvp => kvp.Value > 1)
            .Select(kvp =>
            {
                var parts = kvp.Key.Split('|');
                return (Timestamp: parts[0], Message: parts[1], Count:
 kvp.Value);
            })
            .OrderByDescending(x => x.Count)
            .ToList();
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per .NET .
  - GetQueryResults
  - StartQuery

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.

Questo è il punto di ingresso.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { CloudWatchLogsClient } from "@aws-sdk/client-cloudwatch-logs";
import { CloudWatchQuery } from "./cloud-watch-query.js";

console.log("Starting a recursive query...");

if (!process.env.QUERY_START_DATE || !process.env.QUERY_END_DATE) {
  throw new Error(
    "QUERY_START_DATE and QUERY_END_DATE environment variables are required.",
  );
```

```
}

const cloudWatchQuery = new CloudWatchQuery(new CloudWatchLogsClient({}), {
  logGroupNames: ["/workflows/cloudwatch-logs/large-query"],
  dateRange: [
    new Date(Number.parseInt(process.env.QUERY_START_DATE)),
    new Date(Number.parseInt(process.env.QUERY_END_DATE)),
  ],
});

await cloudWatchQuery.run();

console.log(
  `Queries finished in ${cloudWatchQuery.secondsElapsed} seconds.\nTotal logs
 found: ${cloudWatchQuery.results.length}`,
);
```

Questa è una classe che suddivide le query in più fasi, se necessario.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  StartQueryCommand,
  GetQueryResultsCommand,
} from "@aws-sdk/client-cloudwatch-logs";
import { splitDateRange } from "@aws-doc-sdk-examples/lib/utils/util-date.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

class DateOutOfBoundsError extends Error {}

export class CloudWatchQuery {
  /**
   * Run a query for all CloudWatch Logs within a certain date range.
   * CloudWatch logs return a max of 10,000 results. This class
   * performs a binary search across all of the logs in the provided
   * date range if a query returns the maximum number of results.
   *
   * @param {import('@aws-sdk/client-cloudwatch-logs').CloudWatchLogsClient}
 client
   * @param {{ logGroupNames: string[], dateRange: [Date, Date], queryConfig:
 { limit: number } }} config
   */
```

```
  constructor(client, { logGroupNames, dateRange, queryConfig }) {
    this.client = client;
    /**
     * All log groups are queried.
     */
    this.logGroupNames = logGroupNames;

    /**
     * The inclusive date range that is queried.
     */
    this.dateRange = dateRange;

    /**
     * CloudWatch Logs never returns more than 10,000 logs.
     */
    this.limit = queryConfig?.limit ?? 10000;

    /**
     * @type {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]}
     */
    this.results = [];
  }

  /**
   * Run the query.
   */
  async run() {
    this.secondsElapsed = 0;
    const start = new Date();
    this.results = await this._largeQuery(this.dateRange);
    const end = new Date();
    this.secondsElapsed = (end - start) / 1000;
    return this.results;
  }

  /**
   * Recursively query for logs.
   * @param {[Date, Date]} dateRange
   * @returns {Promise<import("@aws-sdk/client-cloudwatch-logs").ResultField[]
[]>}
   */
  async _largeQuery(dateRange) {
    const logs = await this._query(dateRange, this.limit);
```

```
    console.log(
      `Query date range: ${dateRange
        .map((d) => d.toISOString())
        .join(" to ")}. Found ${logs.length} logs.`,
    );

    if (logs.length < this.limit) {
      return logs;
    }

    const lastLogDate = this._getLastLogDate(logs);
    const offsetLastLogDate = new Date(lastLogDate);
    offsetLastLogDate.setMilliseconds(lastLogDate.getMilliseconds() + 1);
    const subDateRange = [offsetLastLogDate, dateRange[1]];
    const [r1, r2] = splitDateRange(subDateRange);
    const results = await Promise.all([
      this._largeQuery(r1),
      this._largeQuery(r2),
    ]);
    return [logs, ...results].flat();
  }

  /**
   * Find the most recent log in a list of logs.
   * @param {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]} logs
   */
  _getLastLogDate(logs) {
    const timestamps = logs
      .map(
        (log) =>
          log.find((fieldMeta) => fieldMeta.field === "@timestamp")?.value,
      )
      .filter((t) => !!t)
      .map((t) => `${t}Z`)
      .sort();

    if (!timestamps.length) {
      throw new Error("No timestamp found in logs.");
    }

    return new Date(timestamps[timestamps.length - 1]);
  }

  /**
```

```
   * Simple wrapper for the GetQueryResultsCommand.
   * @param {string} queryId
   */
  _getQueryResults(queryId) {
    return this.client.send(new GetQueryResultsCommand({ queryId }));
  }

  /**
   * Starts a query and waits for it to complete.
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   */
  async _query(dateRange, maxLogs) {
    try {
      const { queryId } = await this._startQuery(dateRange, maxLogs);
      const { results } = await this._waitUntilQueryDone(queryId);
      return results ?? [];
    } catch (err) {
      /**
       * This error is thrown when StartQuery returns an error indicating
       * that the query's start or end date occur before the log group was
       * created.
       */
      if (err instanceof DateOutOfBoundsError) {
        return [];
      }
      throw err;
    }
  }

  /**
   * Wrapper for the StartQueryCommand. Uses a static query string
   * for consistency.
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   * @returns {Promise<{ queryId: string }>}
   */
  async _startQuery([startDate, endDate], maxLogs = 10000) {
    try {
      return await this.client.send(
        new StartQueryCommand({
          logGroupNames: this.logGroupNames,
          queryString: "fields @timestamp, @message | sort @timestamp asc",
          startTime: startDate.valueOf(),
```

```
          endTime: endDate.valueOf(),
          limit: maxLogs,
        }),
      );
    } catch (err) {
      /** @type {string} */
      const message = err.message;
      if (message.startsWith("Query's end date and time")) {
        // This error indicates that the query's start or end date occur
        // before the log group was created.
        throw new DateOutOfBoundsError(message);
      }

      throw err;
    }
  }

  /**
   * Call GetQueryResultsCommand until the query is done.
   * @param {string} queryId
   */
  _waitUntilQueryDone(queryId) {
    const getResults = async () => {
      const results = await this._getQueryResults(queryId);
      const queryDone = [
        "Complete",
        "Failed",
        "Cancelled",
        "Timeout",
        "Unknown",
      ].includes(results.status);

      return { queryDone, results };
    };

    return retry(
      { intervalInMs: 1000, maxRetries: 60, quiet: true },
      async () => {
        const { queryDone, results } = await getResults();
        if (!queryDone) {
          throw new Error("Query not done.");
        }

        return results;
```

```
      },
    );
  }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per JavaScript .
  - GetQueryResults
  - StartQuery

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.

Questo file richiama un modulo di esempio per la gestione di CloudWatch query che superano i 10.000 risultati.

```
import logging
import os
import sys

import boto3
from botocore.config import Config

from cloudwatch_query import CloudWatchQuery
from date_utilities import DateUtilities

# Configure logging at the module level.
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(filename)s:%(lineno)d - %(message)s",
)

DEFAULT_QUERY_LOG_GROUP = "/workflows/cloudwatch-logs/large-query"
```

```
class CloudWatchLogsQueryRunner:
    def __init__(self):
        """
        Initializes the CloudWatchLogsQueryRunner class by setting up date
 utilities
        and creating a CloudWatch Logs client with retry configuration.
        """
        self.date_utilities = DateUtilities()
        self.cloudwatch_logs_client = self.create_cloudwatch_logs_client()

    def create_cloudwatch_logs_client(self):
        """
        Creates and returns a CloudWatch Logs client with a specified retry
 configuration.

        :return: A CloudWatch Logs client instance.
        :rtype: boto3.client
        """
        try:
            return boto3.client("logs", config=Config(retries={"max_attempts":
 10}))
        except Exception as e:
            logging.error(f"Failed to create CloudWatch Logs client: {e}")
            sys.exit(1)

    def fetch_environment_variables(self):
        """
        Fetches and validates required environment variables for query start and
 end dates.
        Fetches the environment variable for log group, returning the default
 value if it
        does not exist.

        :return: Tuple of query start date and end date as integers and the log
 group.
        :rtype: tuple
        :raises SystemExit: If required environment variables are missing or
 invalid.
        """
        try:
            query_start_date = int(os.environ["QUERY_START_DATE"])
            query_end_date = int(os.environ["QUERY_END_DATE"])
```

```
        except KeyError:
            logging.error(
                "Both QUERY_START_DATE and QUERY_END_DATE environment variables
are required."
            )
            sys.exit(1)
        except ValueError as e:
            logging.error(f"Error parsing date environment variables: {e}")
            sys.exit(1)

        try:
            log_group = os.environ["QUERY_LOG_GROUP"]
        except KeyError:
            logging.warning("No QUERY_LOG_GROUP environment variable, using
default value")
            log_group = DEFAULT_QUERY_LOG_GROUP

        return query_start_date, query_end_date, log_group

    def convert_dates_to_iso8601(self, start_date, end_date):
        """
        Converts UNIX timestamp dates to ISO 8601 format using DateUtilities.

        :param start_date: The start date in UNIX timestamp.
        :type start_date: int
        :param end_date: The end date in UNIX timestamp.
        :type end_date: int
        :return: Start and end dates in ISO 8601 format.
        :rtype: tuple
        """
        start_date_iso8601 =
self.date_utilities.convert_unix_timestamp_to_iso8601(
            start_date
        )
        end_date_iso8601 = self.date_utilities.convert_unix_timestamp_to_iso8601(
            end_date
        )
        return start_date_iso8601, end_date_iso8601

    def execute_query(
        self,
        start_date_iso8601,
        end_date_iso8601,
        log_group="/workflows/cloudwatch-logs/large-query",
```

```
        query="fields @timestamp, @message | sort @timestamp asc"
    ):
        """
        Creates a CloudWatchQuery instance and executes the query with provided
 date range.

        :param start_date_iso8601: The start date in ISO 8601 format.
        :type start_date_iso8601: str
        :param end_date_iso8601: The end date in ISO 8601 format.
        :type end_date_iso8601: str
        :param log_group: Log group to search: "/workflows/cloudwatch-logs/large-
query"
        :type log_group: str
        :param query: Query string to pass to the CloudWatchQuery instance
        :type query: str
        """
        cloudwatch_query = CloudWatchQuery(
            log_group=log_group,
            query_string=query
        )
        cloudwatch_query.query_logs((start_date_iso8601, end_date_iso8601))
        logging.info("Query executed successfully.")
        logging.info(
            f"Queries completed in {cloudwatch_query.query_duration} seconds.
 Total logs found: {len(cloudwatch_query.query_results)}"
        )


def main():
    """
    Main function to start a recursive CloudWatch logs query.
    Fetches required environment variables, converts dates, and executes the
 query.
    """
    logging.info("Starting a recursive CloudWatch logs query...")
    runner = CloudWatchLogsQueryRunner()
    query_start_date, query_end_date, log_group =
 runner.fetch_environment_variables()
    start_date_iso8601 = DateUtilities.convert_unix_timestamp_to_iso8601(
        query_start_date
    )
    end_date_iso8601 =
 DateUtilities.convert_unix_timestamp_to_iso8601(query_end_date)
```

```
    runner.execute_query(start_date_iso8601, end_date_iso8601,
 log_group=log_group)


if __name__ == "__main__":
    main()
```

Questo modulo elabora le CloudWatch interrogazioni che superano i 10.000 risultati.

```
import logging
import time
from datetime import datetime
import threading
import boto3

from date_utilities import DateUtilities

DEFAULT_QUERY = "fields @timestamp, @message | sort @timestamp asc"
DEFAULT_LOG_GROUP = "/workflows/cloudwatch-logs/large-query"

class DateOutOfBoundsError(Exception):
    """Exception raised when the date range for a query is out of bounds."""

    pass


class CloudWatchQuery:
    """
    A class to query AWS CloudWatch logs within a specified date range.

    :vartype date_range: tuple
    :ivar limit: Maximum number of log entries to return.
    :vartype limit: int
    :log_group str: Name of the log group to query
    :query_string str: query
    """

    def __init__(self, log_group: str = DEFAULT_LOG_GROUP, query_string:
 str=DEFAULT_QUERY) -> None:
        self.lock = threading.Lock()
        self.log_group = log_group
        self.query_string = query_string
```

```
        self.query_results = []
        self.query_duration = None
        self.datetime_format = "%Y-%m-%d %H:%M:%S.%f"
        self.date_utilities = DateUtilities()
        self.limit = 10000

    def query_logs(self, date_range):
        """
        Executes a CloudWatch logs query for a specified date range and
calculates the execution time of the query.

        :return: A batch of logs retrieved from the CloudWatch logs query.
        :rtype: list
        """
        start_time = datetime.now()

        start_date, end_date = self.date_utilities.normalize_date_range_format(
            date_range, from_format="unix_timestamp", to_format="datetime"
        )

        logging.info(
            f"Original query:"
            f"\n       START:     {start_date}"
            f"\n       END:       {end_date}"
            f"\n       LOG GROUP: {self.log_group}"
        )
        self.recursive_query((start_date, end_date))
        end_time = datetime.now()
        self.query_duration = (end_time - start_time).total_seconds()

    def recursive_query(self, date_range):
        """
        Processes logs within a given date range, fetching batches of logs
recursively if necessary.

        :param date_range: The date range to fetch logs for, specified as a tuple
(start_timestamp, end_timestamp).
        :type date_range: tuple
        :return: None if the recursive fetching is continued or stops when the
final batch of logs is processed.
                 Although it doesn't explicitly return the query results, this
method accumulates all fetched logs
                 in the `self.query_results` attribute.
        :rtype: None
```

```
        """
        batch_of_logs = self.perform_query(date_range)
        # Add the batch to the accumulated logs
        with self.lock:
            self.query_results.extend(batch_of_logs)
        if len(batch_of_logs) == self.limit:
            logging.info(f"Fetched {self.limit}, checking for more...")
            most_recent_log = self.find_most_recent_log(batch_of_logs)
            most_recent_log_timestamp = next(
                item["value"]
                for item in most_recent_log
                if item["field"] == "@timestamp"
            )
            new_range = (most_recent_log_timestamp, date_range[1])
            midpoint = self.date_utilities.find_middle_time(new_range)

            first_half_thread = threading.Thread(
                target=self.recursive_query,
                args=((most_recent_log_timestamp, midpoint),),
            )
            second_half_thread = threading.Thread(
                target=self.recursive_query, args=((midpoint, date_range[1]),)
            )

            first_half_thread.start()
            second_half_thread.start()

            first_half_thread.join()
            second_half_thread.join()

    def find_most_recent_log(self, logs):
        """
        Search a list of log items and return most recent log entry.
        :param logs: A list of logs to analyze.
        :return: log
        :type :return List containing log item details
        """
        most_recent_log = None
        most_recent_date = "1970-01-01 00:00:00.000"

        for log in logs:
            for item in log:
                if item["field"] == "@timestamp":
```

```
                    logging.debug(f"Compared: {item['value']} to
{most_recent_date}")
                    if (
                        self.date_utilities.compare_dates(
                            item["value"], most_recent_date
                        )
                        == item["value"]
                    ):
                        logging.debug(f"New most recent: {item['value']}")
                        most_recent_date = item["value"]
                        most_recent_log = log
        logging.info(f"Most recent log date of batch: {most_recent_date}")
        return most_recent_log

    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_group,
                    startTime=start_time,
                    endTime=end_time,
                    queryString=self.query_string,
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
```

```
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_group,
                startTime=start_time,
                endTime=end_time,
                queryString=self.query_string,
                limit=max_logs,
            )
            return response["queryId"]
```

```
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")


    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - GetQueryResults
  - StartQuery

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo degli eventi pianificati per invocare una funzione Lambda

I seguenti esempi di codice mostrano come creare una AWS Lambda funzione richiamata da un evento EventBridge pianificato di Amazon.

Java

SDK per Java 2.x

Mostra come creare un evento EventBridge pianificato da Amazon che richiami una AWS Lambda funzione. Configura EventBridge per utilizzare un'espressione cron per pianificare quando viene richiamata la funzione Lambda. In questo esempio, viene creata una funzione Lambda utilizzando l'API di runtime Lambda Java. Questo esempio richiama diversi AWS servizi per eseguire un caso d'uso specifico. Questo esempio dimostra come creare un'app che invia un messaggio di testo via mobile ai tuoi dipendenti che si congratula con loro alla data dell'anniversario di un anno.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- CloudWatch Registri
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

JavaScript

SDK per JavaScript (v3)

Mostra come creare un evento EventBridge pianificato da Amazon che richiami una AWS Lambda funzione. Configura EventBridge per utilizzare un'espressione cron per pianificare quando viene richiamata la funzione Lambda. In questo esempio, crei una funzione Lambda utilizzando l'API JavaScript Lambda runtime. Questo esempio richiama diversi AWS servizi per eseguire un caso d'uso specifico. Questo esempio dimostra come creare un'app che invia un messaggio di testo via mobile ai tuoi dipendenti che si congratula con loro alla data dell'anniversario di un anno.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Questo esempio è anche disponibile nella Guida per lo sviluppatore di AWS SDK per JavaScript v3.

Servizi utilizzati in questo esempio

- CloudWatch Registri
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

Python

SDK per Python (Boto3)

Questo esempio mostra come registrare una AWS Lambda funzione come destinazione di un EventBridge evento Amazon pianificato. Il gestore Lambda scrive un messaggio intuitivo e i dati completi dell'evento su Amazon CloudWatch Logs per recuperarli in un secondo momento.

- Distribuzione di una funzione Lambda.
- Crea un evento EventBridge pianificato e rende la funzione Lambda la destinazione.
- Concede il permesso di EventBridge invocare la funzione Lambda.
- Stampa i dati più recenti dai CloudWatch registri per mostrare il risultato delle chiamate pianificate.
- Elimina tutte le risorse create durante la demo.

Questo esempio è visualizzato al meglio su. GitHub Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- CloudWatch Registri
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Usare Logs con un CloudWatch AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Crittografa le tabelle di ricerca nei CloudWatch registri utilizzando AWS Key Management Service

I dati della tabella di ricerca sono sempre crittografati nei registri. CloudWatch Per impostazione predefinita, CloudWatch Logs utilizza la crittografia lato server con Advanced Encryption Standard Galois/Counter Mode (AES-GCM) a 256 bit per crittografare i dati della tabella di ricerca inattivi. In alternativa, è possibile utilizzare AWS Key Management Service per questa crittografia. In tal caso, la crittografia viene eseguita utilizzando una chiave. AWS KMS L'utilizzo della crittografia AWS KMS è abilitato a livello di tabella di ricerca, associando una chiave KMS a una tabella di ricerca, quando si crea la tabella di ricerca o quando la si aggiorna.

> **Important**
>
> CloudWatch Logs supporta solo chiavi KMS simmetriche. Non utilizzare una chiave asimmetrica per crittografare i dati nelle tabelle di ricerca. Per ulteriori informazioni, consulta la sezione relativa all'uso di chiavi simmetriche e asimmetriche.

Dopo aver associato una chiave KMS a una tabella di ricerca, tutti i dati memorizzati nella tabella di ricerca vengono crittografati utilizzando questa chiave. CloudWatch I registri decrittografano questi dati ogni volta che vengono richiesti. CloudWatch I log devono disporre delle autorizzazioni per la chiave KMS ogni volta che vengono richiesti dati crittografati.

Se successivamente si dissocia una chiave KMS da una tabella di ricerca, CloudWatch Logs crittografa i dati utilizzando il metodo di crittografia predefinito di Logs. CloudWatch Tuttavia, se la chiave viene disabilitata o eliminata prima di dissociarla, CloudWatch Logs non è in grado di leggere i dati crittografati con quella chiave.

Per informazioni generali su come CloudWatch Logs utilizza per AWS KMS crittografare i dati di registro, vedere. Crittografa i dati di registro in CloudWatch Logs utilizzando AWS Key Management Service

Come viene utilizzato CloudWatch Logs per le tabelle di ricerca AWS KMS

CloudWatch Logs utilizza la crittografia a AWS KMS busta per proteggere i dati della tabella di ricerca. Quando associ una chiave KMS a una tabella di ricerca, CloudWatch Logs invia una richiesta a. `GenerateDataKey` AWS KMS AWS KMS genera una chiave di crittografia dei dati (DEK) unica e restituisce sia una copia in testo semplice che una copia crittografata del DEK. CloudWatch Logs utilizza il DEK in testo semplice per crittografare i dati della tabella di ricerca, quindi archivia il DEK crittografato insieme ai dati crittografati. Il DEK in testo semplice non viene archiviato e viene eliminato dalla memoria dopo l'uso.

Quando CloudWatch Logs deve leggere i dati della tabella di ricerca, invia una `Decrypt` richiesta a con il DEK crittografato. AWS KMS AWS KMS decrittografa il DEK e restituisce il DEK in testo semplice a CloudWatch Logs, che quindi lo utilizza per decrittografare i dati della tabella di ricerca.

CloudWatch Logs utilizza il seguente contesto di crittografia quando effettua richieste a: AWS KMS

```
{
    "aws:logs:arn": "arn:aws:logs:region:account-id:lookup-table:lookup-table-name"
}
```

È possibile utilizzare questo contesto di crittografia nelle politiche IAM e nelle politiche AWS KMS chiave per controllare l'accesso alla chiave KMS. Per ulteriori informazioni, consulta AWS KMS chiavi e contesto di crittografia.

Autorizzazioni richieste

Per utilizzare AWS KMS la crittografia con le tabelle di ricerca, il principale IAM deve disporre delle seguenti AWS KMS autorizzazioni sulla chiave KMS:

- `kms:Decrypt`
- `kms:GenerateDataKey`

L'`kms:Decrypt`autorizzazione è richiesta quando si richiama `GetLookupTable` una tabella di ricerca crittografata con una chiave KMS, in modo che CloudWatch Logs possa decrittografare i dati per conto dell'utente. L'`kms:Decrypt`autorizzazione è richiesta anche

sulla chiave (la chiave KMS utilizzata per crittografare la tabella di ricerca) quando si chiama `StartQuery` con una query che utilizza il comando su una tabella di ricerca crittografata. `lookup` L'`kms:GenerateDataKey`autorizzazione è richiesta durante la chiamata `CreateLookupTable` o `UpdateLookupTable` con una chiave KMS, in modo che CloudWatch Logs possa generare una chiave di crittografia dei dati per crittografare i dati della tabella di ricerca.

Inoltre, il servizio CloudWatch Logs deve disporre dell'autorizzazione per utilizzare la chiave KMS. Puoi concedere queste autorizzazioni aggiungendo una dichiarazione politica alla politica chiave KMS, come descritto nella sezione seguente.

Passaggio 1: creare una chiave AWS KMS

Per creare una chiave KMS simmetrica, usa il seguente comando create-key:

```
aws kms create-key
```

L'output contiene l'ID chiave e l'Amazon Resource Name (ARN) della chiave. Di seguito è riportato un output di esempio:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Registra la chiave ARN. Ne hai bisogno nei seguenti passaggi.

Fase 2: Impostazione delle autorizzazioni sulla chiave KMS

Per impostazione predefinita, tutte AWS KMS le chiavi sono private. Solo il proprietario della risorsa può utilizzarla per crittografare e decrittare i dati. È necessario concedere al servizio CloudWatch Logs l'autorizzazione principale per utilizzare la chiave e concedere anche al ruolo chiamante l'autorizzazione a utilizzare la chiave.

Innanzitutto, salva la politica predefinita per la tua chiave KMS `policy.json` utilizzando il seguente comando: get-key-policy

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Apri il `policy.json` file in un editor di testo e aggiungi la seguente istruzione per concedere al servizio CloudWatch Logs l'autorizzazione principale all'uso della chiave. Questo esempio utilizza una `Condition` sezione che corrisponde al contesto di crittografia per limitare la chiave a una tabella di ricerca specifica.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logs.region.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:lookup-table:lookup-table-name",
            "aws:SourceAccount": "account-id",
            "aws:SourceArn": "arn:aws:logs:region:account-id:lookup-table:lookup-table-
name"
        }
    }
}
```

Quindi, aggiungi le autorizzazioni al ruolo che chiamerà i CloudWatch `CreateLookupTable` Logs o. `UpdateLookupTable` APIs CloudWatch Registra gli usi `kms:ViaService` per effettuare chiamate per AWS KMS conto del cliente. Per ulteriori informazioni, vedere kms:. ViaService

```
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::account-id:role/role-name"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": [
                "logs.region.amazonaws.com"
            ]
        }
    }
}
```

Infine, aggiungi la politica aggiornata utilizzando il seguente put-key-policycomando:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Fase 3: Associare una chiave KMS a una tabella di ricerca

Puoi associare una chiave KMS a una tabella di ricerca quando la crei utilizzando l'`CreateLookupTable`API o aggiornare una tabella di ricerca esistente utilizzando l'API. `UpdateLookupTable` Entrambi APIs fanno parte di. AWSLogs ConfigService

Per associare la chiave KMS a una tabella di ricerca al momento della creazione

Usa l'`CreateLookupTable`API e specifica il `kmsKeyArn` parametro con l'ARN della tua chiave KMS:

```
aws logs create-lookup-table \
```

```
    --lookup-table-name my-lookup-table \
    --kms-key-arn "arn:aws:kms:region:account-id:key/key-id"
```

Per associare la chiave KMS a una tabella di ricerca esistente

Usa l'`UpdateLookupTable`API e specifica il `kmsKeyArn` parametro con l'ARN della tua chiave KMS:

```
aws logs update-lookup-table \
    --lookup-table-name my-lookup-table \
    --kms-key-arn "arn:aws:kms:region:account-id:key/key-id"
```

Considerazioni

- Dopo aver associato o dissociato una chiave KMS da una tabella di ricerca, possono essere necessari fino a cinque minuti prima che l'operazione abbia effetto.
- Se si revoca l'accesso CloudWatch dei log a una chiave associata o si elimina una chiave KMS associata, i dati crittografati della tabella di ricerca in Logs non possono più essere recuperati. CloudWatch
- Per eseguire i passaggi descritti in questo argomento, è necessario disporre delle seguenti autorizzazioni:, `kms:CreateKey` `kms:GetKeyPolicy``kms:PutKeyPolicy`, e delle autorizzazioni di registro appropriate per chiamare o. CloudWatch `CreateLookupTable` `UpdateLookupTable`

Sicurezza in Amazon CloudWatch Logs

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. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili WorkSpaces, consulta AWS Servizi nell'ambito del programma di conformitàAWS .
- 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 ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi Amazon CloudWatch Logs. Ti mostra come configurare Amazon CloudWatch Logs per soddisfare i tuoi obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse di CloudWatch Logs.

Indice

- Protezione dei dati in Amazon CloudWatch Logs
- Gestione delle identità e degli accessi per Amazon CloudWatch Logs
- Convalida della conformità per Amazon Logs CloudWatch
- Resilienza in Amazon Logs CloudWatch
- Sicurezza dell'infrastruttura in Amazon CloudWatch Logs
- Utilizzo dei CloudWatch log con endpoint VPC di interfaccia

Protezione dei dati in Amazon CloudWatch Logs

> **Note**
>
> Oltre alle seguenti informazioni sulla protezione generale dei dati in AWS, CloudWatch Logs consente anche di proteggere i dati sensibili negli eventi di registro mascherandoli. Per ulteriori informazioni, consulta Incremento della protezione dei dati di log sensibili con il mascheramento.

Il modello di responsabilità AWS condivisa (modello di) si applica alla protezione dei dati in Amazon CloudWatch Logs. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati . Per informazioni sulla protezione dei dati in Europa, consulta il General Data Protection Regulation (GDPR) Center.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e di configurare i singoli utenti con Centro identità AWS IAM or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con CloudWatch Logs o altro Servizi AWS utilizzando la console, l'API o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Crittografia dei dati a riposo

CloudWatch Logs protegge i dati archiviati utilizzando la crittografia. Tutti i gruppi di log sono crittografati. Per impostazione predefinita, il servizio CloudWatch Logs gestisce la crittografia lato server e utilizza la crittografia lato server con Advanced Encryption Standard Galois/Counter Mode () AES-GCM a 256 bit per crittografare i dati di registro inattivi.

Se desideri gestire le chiavi utilizzate per crittografare e decrittografare i log, utilizza le chiavi. AWS KMS Per ulteriori informazioni, consulta Crittografa i dati di registro in CloudWatch Logs utilizzando AWS Key Management Service.

Crittografia dei dati in transito

CloudWatch I registri utilizzano la crittografia end-to-end dei dati in transito. Il servizio CloudWatch Logs gestisce le chiavi di crittografia lato server.

Gestione delle identità e degli accessi per Amazon CloudWatch Logs

L'accesso ad Amazon CloudWatch Logs richiede credenziali che AWS possono essere utilizzate per autenticare le tue richieste. Tali credenziali devono disporre delle autorizzazioni per accedere alle AWS risorse, ad esempio per recuperare i dati di CloudWatch Logs sulle risorse cloud. Le sezioni seguenti forniscono dettagli su come utilizzare AWS Identity and Access Management (IAM) e CloudWatch Logs per proteggere le risorse controllando chi può accedervi:

- Autenticazione
- Controllo accessi

Autenticazione

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in Centro identità AWS IAM:

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di Centro identità AWS IAM .
- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Create a role for a third-party identity provider (federation) della Guida per l'utente IAM.
- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella pagina Create a role for an IAM user della Guida per l'utente IAM.
  - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

Controllo accessi

Puoi disporre di credenziali valide per autenticare le tue richieste, ma se non disponi delle autorizzazioni non puoi creare o accedere alle CloudWatch risorse di Logs. Ad esempio, è necessario disporre delle autorizzazioni per creare flussi di log, gruppi di log e così via.

Le sezioni seguenti descrivono come gestire le autorizzazioni per i log. CloudWatch Consigliamo di leggere prima la panoramica.

- Panoramica sulla gestione delle autorizzazioni di accesso alle risorse Logs CloudWatch
- Utilizzo di politiche basate sull'identità (politiche IAM) per i log CloudWatch
- CloudWatch Riferimento alle autorizzazioni dei log

Panoramica sulla gestione delle autorizzazioni di accesso alle risorse Logs CloudWatch

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in Centro identità AWS IAM:

  Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina Create a permission set (Creazione di un set di autorizzazioni) nella Guida per l'utente di Centro identità AWS IAM .

- Utenti gestiti in IAM tramite un provider di identità:

  Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Create a role for a third-party identity provider (federation) della Guida per l'utente IAM.

- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella pagina Create a role for an IAM user della Guida per l'utente IAM.
  - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

Argomenti

- CloudWatch Registra risorse e operazioni
- Informazioni sulla proprietà delle risorse
- Gestione dell'accesso alle risorse
- Specifica degli elementi delle policy: operazioni, effetti e principali
- Specifica delle condizioni in una policy

CloudWatch Registra risorse e operazioni

In CloudWatch Logs le risorse principali sono i gruppi di log, i flussi di log e le destinazioni. CloudWatch Logs non supporta le risorse secondarie (altre risorse da utilizzare con la risorsa principale).

Alle risorse e alle risorse secondarie sono associati Amazon Resource Name (ARN) univoci, come illustrato nella tabella seguente.

Tipo di risorsa	Formato ARN
Gruppo di log	

Tipo di risorsa	Formato ARN
	arn:aws:logs: ::log-group: *region account-id log_group_name*
Flusso di log	arn:aws:logs: ::log-group *region* ::log-stream: *account-id log_group_name log-stream-name*
Destinazione	arn:aws:logs: *region* ::destinazione: *account-id destination_name*

Per ulteriori informazioni sugli ARN consulta la sezione relativa agli ARN nella Guida per l'utente di IAM. Per informazioni sui CloudWatch Logs ARN, consulta Amazon Resource Names (ARN) in. Riferimenti generali di Amazon Web Services Per un esempio di policy che copre CloudWatch i log, consulta. Utilizzo di politiche basate sull'identità (politiche IAM) per i log CloudWatch

CloudWatch Logs fornisce una serie di operazioni per utilizzare le risorse CloudWatch Logs. Per un elenco di operazioni disponibili, consulta la sezione CloudWatch Riferimento alle autorizzazioni dei log.

Informazioni sulla proprietà delle risorse

L' AWS account è il proprietario delle risorse create nell'account, indipendentemente da chi le ha create. In particolare, il proprietario della risorsa è l' AWS account dell'entità principale (ovvero l'account root, un utente o un ruolo IAM) che autentica la richiesta di creazione della risorsa. Negli esempi seguenti viene illustrato il funzionamento:

- Se utilizzi le credenziali dell'account root del tuo AWS account per creare un gruppo di log, il tuo AWS account è il proprietario della CloudWatch risorsa Logs.
- Se crei un utente nel tuo AWS account e concedi le autorizzazioni per creare risorse CloudWatch Logs a quell'utente, l'utente può creare risorse CloudWatch Logs. Tuttavia, il tuo AWS account, a cui appartiene l'utente, possiede le risorse CloudWatch Logs.
- Se crei un ruolo IAM nel tuo AWS account con le autorizzazioni per creare risorse CloudWatch Logs, chiunque possa assumere il ruolo può creare risorse CloudWatch Logs. Il tuo AWS account, a cui appartiene il ruolo, possiede le risorse CloudWatch Logs.

Gestione dell'accesso alle risorse

La policy delle autorizzazioni descrive chi ha accesso a cosa. Nella sezione seguente vengono descritte le opzioni disponibili per la creazione di policy relative alle autorizzazioni.

> **Note**
>
> Questa sezione illustra l'utilizzo di IAM nel contesto dei CloudWatch log. Non vengono fornite informazioni dettagliate sul servizio IAM. Per la documentazione di IAM completa, consulta la pagina Che cos'è IAM? nella Guida per l'utente di IAM. Per informazioni sulla sintassi delle policy IAM e le rispettive descrizioni, consultare Riferimento alle policy IAM di nella Guida per l'utente di IAM.

Le politiche associate a un'identità IAM sono denominate politiche basate sull'identità (politiche IAM) e le politiche associate a una risorsa sono denominate politiche basate sulle risorse. CloudWatch Logs supporta politiche basate sull'identità e politiche basate sulle risorse per le destinazioni, utilizzate per abilitare gli abbonamenti tra account. Per ulteriori informazioni, consulta Cross-account Abbonamenti interregionali.

Argomenti

- Autorizzazioni del gruppo di log e Contributor Insights
- Resource-based politiche

Autorizzazioni del gruppo di log e Contributor Insights

Contributor Insights è una funzionalità CloudWatch che consente di analizzare i dati dai gruppi di log e creare serie temporali che visualizzano i dati dei collaboratori. Puoi visualizzare i parametri relative ai primi N collaboratori, al numero totale di collaboratori univoci e al loro utilizzo. Per ulteriori informazioni, consulta Utilizzo di Contributor Insights per analizzare i dati. High-Cardinality

Le operazioni di Contributor Insights utilizzano il namespace `cloudwatch:` IAM. CloudWatch Le autorizzazioni di logs (`logs:`) non vengono valutate quando Contributor Insights elabora i dati dei gruppi di log. Per informazioni dettagliate sul modello di autorizzazioni, sulle chiavi di condizione e sulle best practice di Contributor Insights, consulta Usare le chiavi di condizione per limitare l'accesso degli utenti di Contributor Insights ai gruppi di log nella Amazon User Guide. CloudWatch

Se i tuoi gruppi di log contengono dati sensibili, utilizza le politiche di protezione dei dati di CloudWatch Logs per mascherare tali dati prima che Contributor Insights o qualsiasi altra funzionalità possano elaborarli. Per ulteriori informazioni, consulta Incremento della protezione dei dati di log sensibili con il mascheramento.

Resource-based politiche

CloudWatch Logs supporta politiche basate sulle risorse per le destinazioni, che puoi utilizzare per abilitare gli abbonamenti tra account. Per ulteriori informazioni, consulta Passaggio 1: creazione di una destinazione. Le destinazioni possono essere create utilizzando l'PutDestinationAPI ed è possibile aggiungere una politica sulle risorse alla destinazione utilizzando l'API. PutDestinationPolicy L'esempio seguente permette a un altro account AWS con l'ID account 111122223333 di sottoscrivere i propri gruppi di log nella destinazione `arn:aws:logs:us-east-1:123456789012:destination:testDestination`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111122223333"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:123456789012:destination:testDestination"
    }
  ]
}
```

Specifica degli elementi delle policy: operazioni, effetti e principali

Per ogni risorsa CloudWatch Logs, il servizio definisce una serie di operazioni API. Per concedere le autorizzazioni per queste operazioni API, CloudWatch Logs definisce una serie di azioni che è possibile specificare in una policy. Alcune operazioni API possono richiedere le autorizzazioni per più di un'operazione al fine di eseguire l'operazione API. Per ulteriori informazioni sulle risorse e sulle

operazioni delle API, consulta CloudWatch Registra risorse e operazioni e CloudWatch Riferimento alle autorizzazioni dei log.

Di seguito sono elencati gli elementi di base di una policy:

- Risorsa - Usa un Amazon Resource Name (ARN) per identificare la risorsa a cui si applica la policy. Per ulteriori informazioni, consulta CloudWatch Registra risorse e operazioni.
- Operazione: utilizzi le parole chiave per identificare le operazioni sulla risorsa da permettere o rifiutare. Ad esempio, l'autorizzazione `logs.DescribeLogGroups` concede all'utente le autorizzazioni per eseguire l'operazione `DescribeLogGroups`.
- Effetto: specifica l'effetto, ovvero l'autorizzazione o il rifiuto, quando l'utente richiede l'operazione specifica. USe non concedi esplicitamente (consenti) l'accesso a una risorsa, l'accesso viene implicitamente rifiutato. Puoi anche rifiutare esplicitamente l'accesso a una risorsa per garantire che un utente non possa accedervi, anche se l'accesso viene concesso da un'altra policy.
- Principale: nelle policy basate su identità (policy IAM), l'utente a cui la policy è collegata è il principale implicito. Per le politiche basate sulle risorse, specifichi l'utente, l'account, il servizio o l'altra entità a cui desideri ricevere le autorizzazioni (si applica solo alle politiche basate sulle risorse). CloudWatch Logs supporta le politiche basate sulle risorse per le destinazioni.

Per ulteriori informazioni sulla sintassi e le descrizioni delle policy IAM, consulta AWS Riferimento alle policy IAM nella Guida per l'utente di IAM.

Per una tabella che mostra tutte le azioni dell'API CloudWatch Logs e le risorse a cui si applicano, consulta. CloudWatch Riferimento alle autorizzazioni dei log

Specifica delle condizioni in una policy

Quando concedi le autorizzazioni, puoi utilizzare la sintassi della policy di accesso per specificare le condizioni in base a cui la policy deve essere applicata. Ad esempio, potresti decidere che una policy venga applicata solo dopo una data specifica. Per ulteriori informazioni su come specificare le condizioni in un linguaggio di policy, consulta la sezione Condizione nella Guida per l'utente di IAM.

Per esprimere le condizioni è necessario utilizzare chiavi di condizione predefinite. Per un elenco delle chiavi di contesto supportate da ciascun AWS servizio e un elenco di chiavi di policy a AWS livello globale, consulta Azioni, risorse e chiavi di condizione per AWS i servizi e chiavi di contesto di condizione AWS globali.

> **Note**
>
> Puoi utilizzare i tag per controllare l'accesso alle risorse di CloudWatch Logs, inclusi i gruppi di log e le destinazioni. L'accesso ai flussi di log è controllato a livello di gruppo di log per via della relazione gerarchica tra i gruppi di log e i flussi di log. Per ulteriori informazioni sull'utilizzo dei tag per controllare l'accesso alle risorse, consulta Controllo dell'accesso alle risorse di Amazon Web Services utilizzando i tag.

Utilizzo di politiche basate sull'identità (politiche IAM) per i log CloudWatch

In questo argomento vengono forniti esempi di policy basate su identità in cui un amministratore account può collegare policy di autorizzazione a identità IAM, ovvero utenti, gruppi e ruoli.

> **Important**
>
> Ti consigliamo di esaminare innanzitutto gli argomenti introduttivi che spiegano i concetti e le opzioni di base disponibili per gestire l'accesso alle tue risorse Logs. CloudWatch Per ulteriori informazioni, consulta Panoramica sulla gestione delle autorizzazioni di accesso alle risorse Logs CloudWatch.

Questo argomento comprende quanto segue:

- Autorizzazioni necessarie per utilizzare la console CloudWatch
- AWS politiche gestite (predefinite) per i log CloudWatch
- Esempi di policy gestite dal cliente

Di seguito è riportato un esempio di policy delle autorizzazioni:

JSON

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

```
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Questa policy dispone di una dichiarazione che concede autorizzazioni per creare gruppi e flussi di log, caricare eventi di log ai flussi di log ed elencare dettagli relativi ai flussi di log.

Il carattere jolly (*) nel `Resource` valore concede l'autorizzazione per le azioni elencate su qualsiasi risorsa Logs. CloudWatch

- Per limitare le autorizzazioni a specifiche azioni di gruppi di log (ad esempio,`DescribeLogStreams`)`CreateLogGroup`, sostituisci il carattere jolly con il gruppo di log ARN— `arn:aws:logs:`*`us-west-2`*`:`*`123456789012`*`:log-group:`*`MyLogGroup`*
- Per limitare le autorizzazioni a specifiche azioni del flusso di log (ad esempio,`PutLogEvents`)`CreateLogStream`, sostituisci il carattere jolly con l'ARN— `arn:aws:logs:`*`us-west-2`*`:`*`123456789012`*`:log-group:`*`MyLogGroup`*`:*` (corrisponde a tutti gli stream) o (stream specifico) `arn:aws:logs:`*`us-west-2`*`:`*`123456789012`*`:log-group:`*`MyLogGroup`*`:log-stream:`*`MyStream`*

Consulta il riferimento all'autorizzazione del servizio per il tipo di risorsa richiesto da ciascuna API.

Autorizzazioni necessarie per utilizzare la console CloudWatch

Affinché un utente possa CloudWatch utilizzare Logs nella CloudWatch console, deve disporre di un set minimo di autorizzazioni che gli consenta di descrivere altre AWS risorse nel proprio AWS account. Per utilizzare CloudWatch Logs nella CloudWatch console, è necessario disporre delle autorizzazioni dei seguenti servizi:

- CloudWatch
- CloudWatch Log

- OpenSearch Servizio
- IAM
- Kinesis
- Lambda
- Simple Storage Service (Amazon S3)

Se decidi di creare una policy IAM più restrittiva delle autorizzazioni minime richieste, la console non funzionerà come previsto per gli utenti con tale policy IAM. Per garantire che tali utenti possano ancora utilizzare la CloudWatch console, allega anche la policy `CloudWatchReadOnlyAccess` gestita all'utente, come descritto in AWS politiche gestite (predefinite) per i log CloudWatch.

Non è necessario consentire le autorizzazioni minime della console per gli utenti che effettuano chiamate solo all'API AWS CLI o all'API CloudWatch Logs.

Il set completo di autorizzazioni necessarie per utilizzare la CloudWatch console per un utente che non utilizza la console per gestire le sottoscrizioni ai log è:

- cloudwatch: GetMetricData
- orologio cloud: ListMetrics
- registri: CancelExportTask
- registri: CreateExportTask
- registri: CreateLogGroup
- registri: CreateLogStream
- registri: DeleteLogGroup
- registri: DeleteLogStream
- registri: DeleteMetricFilter
- registri: DeleteQueryDefinition
- registri: DeleteRetentionPolicy
- registri: DeleteSubscriptionFilter
- registri: DescribeExportTasks
- registri: DescribeLogGroups
- registri: DescribeLogStreams
- registri: DescribeMetricFilters

- registri: DescribeQueryDefinitions
- registri: DescribeQueries
- registri: DescribeSubscriptionFilters
- registri: FilterLogEvents
- registri: GetLogEvents
- registri: GetLogGroupFields
- registri: GetLogRecord
- registri: GetQueryResults
- registri: PutMetricFilter
- registri: PutQueryDefinition
- registri: PutRetentionPolicy
- registri: StartQuery
- registri: StopQuery
- registri: PutSubscriptionFilter
- registri: TestMetricFilter

Per un utente che intende utilizzare anche la console per gestire le sottoscrizioni ai log, sono necessarie anche le seguenti autorizzazioni:

- es: DescribeElasticsearchDomain
- es: ListDomainNames
- sono: AttachRolePolicy
- sono: CreateRole
- sono: GetPolicy
- sono: GetPolicyVersion
- Sono: GetRole
- sono: ListAttachedRolePolicies
- sono: ListRoles
- cinesia: DescribeStreams
- cinesi: ListStreams
- lambda: AddPermission

- lambda: CreateFunction
- lambda: GetFunctionConfiguration
- lambda: ListAliases
- lambda: ListFunctions
- lambda: ListVersionsByFunction
- lambda: RemovePermission
- s3: ListBuckets

AWS politiche gestite (predefinite) per i log CloudWatch

AWS affronta molti casi d'uso comuni fornendo policy IAM autonome create e amministrate da. AWS Le policy gestite concedono le autorizzazioni necessarie per i casi di utilizzo comune in modo da non dover cercare quali sono le autorizzazioni richieste. Per ulteriori informazioni, consulta Policy gestite da AWS nella Guida per l'utente di IAM.

Le seguenti politiche AWS gestite, che puoi allegare agli utenti e ai ruoli del tuo account, sono specifiche CloudWatch dei log:

- CloudWatchLogsFullAccess— Consente l'accesso completo ai CloudWatch log.
- CloudWatchLogsReadOnlyAccess— Concede l'accesso in sola lettura ai log. CloudWatch

CloudWatchLogsFullAccess

La CloudWatchLogsFullAccess politica garantisce l'accesso completo ai log. CloudWatch La policy include le `cloudwatch:GenerateQueryResultsSummary` autorizzazioni `cloudwatch:GenerateQuery` and, in modo che gli utenti con questa policy possano generare una stringa di query CloudWatch Logs Insights da un prompt in linguaggio naturale. Per visualizzare il contenuto completo della policy, consulta CloudWatchLogsFullAccess la AWS Managed Policy Reference Guide.

CloudWatchLogsReadOnlyAccess

La CloudWatchLogsReadOnlyAccess policy garantisce l'accesso in sola lettura ai log. CloudWatch Include le `cloudwatch:GenerateQueryResultsSummary` autorizzazioni `cloudwatch:GenerateQuery` and, in modo che gli utenti con questa policy possano generare una stringa di query CloudWatch Logs Insights da un prompt in linguaggio naturale. Per visualizzare

il contenuto completo della policy, consulta CloudWatchLogsReadOnlyAccess la AWS Managed Policy Reference Guide.

CloudWatchOpenSearchDashboardsFullAccess

La CloudWatchOpenSearchDashboardsFullAccess policy concede l'accesso per creare, gestire ed eliminare integrazioni con il OpenSearch Servizio e per creare, eliminare e gestire dashboard di log vended in tali integrazioni.

Per vedere il contenuto completo della policy, consulta la Managed Policy Reference CloudWatchOpenSearchDashboardsFullAccess Guide. AWS

CloudWatchOpenSearchDashboardAccess

La CloudWatchOpenSearchDashboardAccess policy consente l'accesso alla visualizzazione delle dashboard dei log vendute create con l'analisi. Amazon OpenSearch Service

Per vedere il contenuto completo della policy, consulta la AWS Managed Policy CloudWatchOpenSearchDashboardAccess Reference Guide.

CloudWatchLogsCrossAccountSharingConfiguration

La CloudWatchLogsCrossAccountSharingConfiguration policy concede l'accesso per creare, gestire e visualizzare i link di Observability Access Manager per la condivisione delle risorse CloudWatch Logs tra account. Per maggiori informazioni, consulta la sezione Osservabilità su più account di CloudWatch .

Per vedere il contenuto completo della policy, consulta CloudWatchLogsCrossAccountSharingConfiguration la AWS Managed Policy Reference Guide.

CloudWatchLogsAPIKeyAccess

La CloudWatchLogsAPIKeyAccess policy abilita l'autenticazione con chiave API CloudWatch Logs e l'inserimento di log crittografati. Questa policy concede le autorizzazioni per l'autenticazione tramite token al portatore e la scrittura di eventi di registro nei log, con AWS KMS autorizzazioni aggiuntive per la decrittografia e la generazione di chiavi di dati quando i CloudWatch log sono crittografati.

La policy concede le seguenti autorizzazioni:

- `logs`— Consente ai principali di autenticarsi tramite token API key bearer e di scrivere eventi di log nei flussi di Logs. CloudWatch

- `kms`— Consente ai principali di leggere i metadati delle chiavi, generare AWS KMS chiavi di dati per la crittografia e decrittografare i dati. Queste autorizzazioni supportano i CloudWatch log crittografati consentendo al servizio di crittografare i dati di registro utilizzando chiavi gestite dal cliente. AWS KMS L'accesso è limitato alle operazioni eseguite tramite il servizio Logs. CloudWatch

Per visualizzare ulteriori dettagli sulla policy, inclusa la versione più recente del documento di policy JSON, consulta CloudWatchLogsAPIKeyAccess nella Guida di riferimento alle policy gestite da AWS .

CloudWatch Registra gli aggiornamenti di AWS policy gestite

Visualizza i dettagli sugli aggiornamenti delle policy AWS gestite per CloudWatch i log da quando questo servizio ha iniziato a tracciare queste modifiche. Per ricevere avvisi automatici sulle modifiche apportate a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei documenti di CloudWatch log.

Modifica	Descrizione	Data
CloudWatchLogsAPIKeyAccess— Nuova policy.	CloudWatch Logs ha aggiunto una nuova policy gestita. CloudWatchLogsAPIKeyAccess Questa policy consente l'autenticazione con chiave API CloudWatch Logs e l'inserimento di log crittografati, concedendo le autorizzazioni per l'autenticazione utilizzando i token del portatore e scrivendo gli eventi di registro nei log. CloudWatch	17 febbraio 2026
		02 dicembre 2025

Modifica	Descrizione	Data
CloudWatchLogsFullAccess: aggiorna a una policy esistente.	CloudWatch I registri hanno aggiunto autorizzazioni a. CloudWatchLogsFullAccess Le autorizzazioni per le azioni di amministrazione dell'osservabilità sono state aggiunte per consentire l'accesso in sola lettura alle pipeline di telemetria e alle integrazioni delle tabelle S3.	
CloudWatchLogsReadOnlyAccess: aggiorna a una policy esistente.	CloudWatch I registri hanno aggiunto autorizzazioni a. CloudWatchLogsReadOnlyAccess Le autorizzazioni per le azioni di amministrazione dell'osservabilità sono state aggiunte per consentire l'accesso in sola lettura alle pipeline di telemetria e alle integrazioni delle tabelle S3.	02 dicembre 2025

Modifica	Descrizione	Data
CloudWatchLogsFullAccess: aggiorna a una policy esistente.	CloudWatch I registri hanno aggiunto autorizzazioni a. CloudWatchLogsFullAccess Le autorizzazioni per `cloudwatch:GenerateQueryResultsSummary` sono state aggiunte per consentire la generazione di un riepilogo in linguaggio naturale dei risultati della query.	20 maggio 2025
CloudWatchLogsReadOnlyAccess: aggiorna a una policy esistente.	CloudWatch I log hanno aggiunto le autorizzazioni a. CloudWatchLogsReadOnlyAccess Le autorizzazioni per `cloudwatch:GenerateQueryResultsSummary` sono state aggiunte per consentire la generazione di un riepilogo in linguaggio naturale dei risultati della query.	20 maggio 2025

Modifica	Descrizione	Data
CloudWatchLogsFullAccess: aggiorna a una policy esistente.	CloudWatch I log hanno aggiunto le autorizzazioni a. CloudWatchLogsFullAccess Sono state aggiunte le autorizzazioni per Amazon OpenSearch Service e IAM, per abilitare l'integrazione CloudWatch dei log con il OpenSearch servizio per alcune funzionalità.	1 dicembre 2024
CloudWatchOpenSearchDashboardsFullAccess— Nuova policy IAM.	CloudWatch Logs ha aggiunto una nuova policy IAM, CloudWatchOpenSearchDashboardsFullAccess .- Questa policy consente l'accesso per creare, gestire ed eliminare integrazioni con il OpenSearch Servizio e per creare, gestire ed eliminare le dashboard di log vendute in tali integrazioni.	1 dicembre 2024
CloudWatchOpenSearchDashboardAccess— Nuova politica IAM.	CloudWatch Logs ha aggiunto una nuova policy IAM, CloudWatchOpenSearchDashboardAccess .- Questa policy consente l'accesso alle dashboard di visualizzazione dei log vendute fornite da. Amazon OpenSearch Service	1 dicembre 2024

Modifica	Descrizione	Data
CloudWatchLogsFullAccess: aggiorna a una policy esistente.	CloudWatch I log hanno aggiunto un'autorizzazione a. CloudWatchLogsFullAccess L'`cloudwatch:GenerateQuery` autorizzazione è stata aggiunta, in modo che gli utenti con questa policy possano generare una stringa di query CloudWatch Logs Insights da un prompt in linguaggio naturale.	27 novembre 2023
CloudWatchLogsReadOnlyAccess: aggiorna a una policy esistente.	CloudWatch ha aggiunto un'autorizzazione a CloudWatchLogsReadOnlyAccess. L'`cloudwatch:GenerateQuery` autorizzazione è stata aggiunta, in modo che gli utenti con questa policy possano generare una stringa di query CloudWatch Logs Insights da un prompt in linguaggio naturale.	27 novembre 2023

Modifica	Descrizione	Data
CloudWatchLogsReadOnlyAccess: aggiornamento di una policy esistente	CloudWatch I log hanno aggiunto le autorizzazioni a. CloudWatchLogsReadOnlyAccess Le `logs:StopLiveTail` autorizzazioni `logs:StartLiveTail` and sono state aggiunte in modo che gli utenti con questa policy possano utilizzare la console per avviare e interrompere le sessioni live tail di CloudWatch Logs. Per ulteriori informazioni, consulta Use live tail to view logs in near real time.	6 giugno 2023
CloudWatchLogsCrossAccountSharingConfiguration: nuova policy	CloudWatch Logs ha aggiunto una nuova policy per consentirti di gestire i link di osservabilità CloudWatch tra più account che condividono i gruppi di CloudWatch log di Logs. Per ulteriori informazioni, consulta CloudWatch Osservabilità tra account	27 novembre 2022

Modifica	Descrizione	Data
CloudWatchLogsRead OnlyAccess: aggiornamento di una policy esistente	CloudWatch I log hanno aggiunto autorizzazioni a. CloudWatchLogsRead OnlyAccess Le `oam:ListAttachedLinks` autorizzazioni `oam:ListSinks` and sono state aggiunte in modo che gli utenti con questa policy possano utilizzare la console per visualizzare i dati condivisi dagli account di origine con osservabilità CloudWatch tra più account.	27 novembre 2022

Esempi di policy gestite dal cliente

Puoi creare politiche IAM personalizzate per consentire le autorizzazioni per le azioni e le risorse di CloudWatch Logs. Puoi collegare queste policy personalizzate agli utenti o ai gruppi che richiedono le autorizzazioni.

In questa sezione, puoi trovare esempi di politiche utente che concedono le autorizzazioni per varie azioni di CloudWatch Logs. Queste politiche funzionano quando si utilizzano l'API CloudWatch Logs, AWS gli SDK o il. AWS CLI

Esempi

- Esempio 1: consenti l'accesso completo ai log CloudWatch
- Esempio 2: consenti l'accesso in sola lettura ai log CloudWatch
- Esempio 3: consentire l'accesso a un gruppo di log o flusso di log

Esempio 1: consenti l'accesso completo ai log CloudWatch

La seguente politica consente a un utente di accedere a tutte le azioni di CloudWatch Logs.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Esempio 2: consenti l'accesso in sola lettura ai log CloudWatch

AWS fornisce una CloudWatchLogsReadOnlyAccess policy che consente l'accesso in sola lettura ai dati dei log. CloudWatch Questa policy include le seguenti autorizzazioni:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
```

```
    ]
}
```

Esempio 3: consentire l'accesso a un gruppo di log o flusso di log

CloudWatch Logs ha due tipi di risorse con diversi formati ARN:

- Gruppo di log: `arn:aws:logs:region:account:log-group:LogGroupName`
- Stream di log: `arn:aws:logs:region:account:log-group:LogGroupName:log-stream:StreamName`

Quando si scrivono policy IAM, il formato ARN utilizzato deve corrispondere al tipo di risorsa su cui l'API è autorizzata. Consulta il riferimento all'autorizzazione del servizio per il tipo di risorsa richiesto da ciascuna API.

Esempio 3a: consentire l'accesso alle azioni a livello di gruppo di log su un gruppo di log specifico

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:DeleteLogGroup",
        "logs:PutRetentionPolicy",
        "logs:PutSubscriptionFilter",
        "logs:DescribeLogStreams"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName"
      }
   ]
}
```

Queste azioni autorizzano il tipo di risorsa. `log-group` È supportato il formato ARN standard (senza `:*`).

Esempio 3b: consentire l'accesso alle azioni a livello di log stream su un gruppo di log specifico

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName:*"
      }
   ]
}
```

Queste azioni autorizzano il tipo di risorsa. `log-stream` Il `:*` suffisso è obbligatorio per corrispondere a tutti i flussi di log all'interno del gruppo di log o per specificare un flusso specifico con. `:log-stream:`*`StreamName`*

Esempio 3c: politica combinata per le azioni log-group e log-stream

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
        "Action": [
          "logs:DeleteLogGroup",
          "logs:PutRetentionPolicy",
          "logs:DescribeLogStreams",
          "logs:FilterLogEvents"
        ],
        "Effect": "Allow",
        "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName"
      },
      {
        "Action": [
          "logs:CreateLogStream",
          "logs:PutLogEvents",
          "logs:GetLogEvents"
        ],
        "Effect": "Allow",
        "Resource": "arn:aws:logs:us-west-2:123456789012:log-
group:SampleLogGroupName:*"
      }
```

```
    ]
}
```

Utilizzo del tagging e delle policy IAM per il controllo a livello di gruppo di log

È possibile concedere agli utenti l'accesso a determinati gruppi di log evitando che accedano ad altri gruppi di log. Per farlo, aggiungere un tag ai gruppi di log e usare policy IAM che fanno riferimento a tali tag. Per applicare i tag a un gruppo di log, è necessario disporre dell'autorizzazione `logs:TagResource` o `logs:TagLogGroup`. Ciò vale sia se si assegnano tag al gruppo di log al momento della creazione, sia se li si assegna successivamente.

Per ulteriori informazioni sull'applicazione di tag ai gruppi di log, consulta Contrassegna i gruppi di log in Amazon CloudWatch Logs.

Quando si aggiunge un tag ai gruppi di log, è possibile concedere una policy IAM a un utente per consentire l'accesso solo ai gruppi di log con un determinato tag. Ad esempio, la seguente istruzione di policy garantisce l'accesso solo ai gruppi di log con il valore `Green` per la chiave di tag `Team`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

Le operazioni StopQuery e StopLiveTail API non interagiscono con le AWS risorse nel senso tradizionale. Non restituiscono né inseriscono alcun dato, né modificano una risorsa in alcun

modo. Invece, operano solo su una determinata sessione live tail o su una determinata query di CloudWatch Logs Insights, che non sono classificate come risorse. Di conseguenza, quando per queste operazioni si specifica il campo `Resource` nelle policy IAM, è necessario impostare il valore del campo `Resource` come *, analogamente all'esempio di seguito.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
        [ {
            "Effect": "Allow",
            "Action": [
                "logs:StopQuery",
                "logs:StopLiveTail"
            ],
            "Resource": "*"
            }
        ]
}
```

Per ulteriori informazioni sull'utilizzo di istruzioni di policy IAM, consulta Controllo dell'accesso tramite le policy nella Guida per l'utente di IAM.

CloudWatch Riferimento alle autorizzazioni dei log

Quando configuri Controllo accessi e scrivi policy di autorizzazioni che puoi collegare a un'identità IAM (policy basate su identità), puoi usare la tabella seguente come riferimento. La tabella elenca ogni operazione dell'API CloudWatch Logs e le azioni corrispondenti per le quali è possibile concedere le autorizzazioni per eseguire l'azione. Puoi specificare le operazioni nel campo `Action` della policy. Per il `Resource` campo, puoi specificare l'ARN di un gruppo di log o di un flusso di log oppure specificare di rappresentare tutte * le risorse CloudWatch Logs.

È possibile utilizzare chiavi di condizione AWS a livello globale nelle politiche di CloudWatch Logs per esprimere le condizioni. Per un elenco completo delle chiavi AWS-wide, consulta AWS Global e IAM Condition Condition Context Keys nella IAM User Guide.

> **Note**
>
> Per specificare un'operazione, utilizza il prefisso `logs:` seguito dal nome dell'operazione API. Ad esempio:`logs:CreateLogGroup`,`logs:CreateLogStream`, o `logs:*` (per tutte le azioni di CloudWatch Logs).

CloudWatch Registra le operazioni API e le autorizzazioni richieste per le azioni

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
CancelExportTask	`logs:CancelExportTask` Necessaria per eliminare un'attività di esportazione in esecuzione o pendente.
CreateExportTask	`logs:CreateExportTask` Necessaria per esportare dati da un gruppo di log a un bucket Amazon S3.
CreateLogGroup	`logs:CreateLogGroup` Necessaria per creare un nuovo gruppo di log.
CreateLogStream	`logs:CreateLogStream` Necessaria per creare un nuovo flusso di log in un gruppo di log.
DeleteDestination	`logs:DeleteDestination` Necessaria per eliminare una destinazione di log e disabilita tutti i filtri di sottoscrizione.
DeleteLogGroup	`logs:DeleteLogGroup`

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
	Necessario per eliminare un gruppo di log e tutti i log eventi archiviati associati.
DeleteLogStream	`logs:DeleteLogStream` Necessaria per eliminare un flusso di log e tutti gli eventi di log archiviati associati.
DeleteMetricFilter	`logs:DeleteMetricFilter` Necessaria per eliminare un filtro parametri associato a un gruppo di log.
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Necessario per eliminare una definizione di query salvata in CloudWatch Logs Insights.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Necessario per eliminare una policy relativa alle risorse di CloudWatch Logs.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Necessaria per eliminare una policy di retention di un gruppo di log.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Necessaria per eliminare un filtro sottoscrizioni associato a un gruppo di log.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
DescribeDestinations	`logs:DescribeDestinations` Necessaria per visualizzare tutte le destinazioni associate all'account.
DescribeExportTasks	`logs:DescribeExportTasks` Necessaria per visualizzare tutte le attività di esportazione associate all'account.
DescribeLogGroups	`logs:DescribeLogGroups` Necessaria per visualizzare tutti i gruppi di log associati all'account.
DescribeLogStreams	`logs:DescribeLogStreams` Necessaria per visualizzare tutti i flussi di log associati a un gruppo di log.
DescribeMetricFilters	`logs:DescribeMetricFilters` Necessario per visualizzare tutti i parametri associati a un gruppo di log.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Necessario per visualizzare l'elenco delle definizioni delle query salvate in CloudWatch Logs Insights.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
DescribeQueries	`logs:DescribeQueries` Necessario per visualizzare l'elenco delle query di CloudWatch Logs Insights pianificate, in esecuzione o eseguite di recente.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Necessario per visualizzare un elenco delle politiche relative alle risorse di CloudWatch Logs.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Necessaria per visualizzare tutti i filtri di sottoscrizione associati a un gruppo di log.
FilterLogEvents	`logs:FilterLogEvents` Necessaria per ordinare log eventi in base a un modello filtro di gruppo di log.
GetLogEvents	`logs:GetLogEvents` Necessaria per recuperare eventi di log da un flusso di log.
GetLogGroupFields	`logs:GetLogGroupFields` Necessaria per recuperare l'elenco dei campi inclusi negli eventi di log in un gruppo di log.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
GetLogRecord	`logs:GetLogRecord` Necessaria per recuperare i dettagli da un singolo log eventi.
GetLogObject	`logs:GetLogRecord` Necessario per recuperare il contenuto di ampie porzioni di eventi di registro che sono stati inseriti tramite l'API. PutOpenTelemetryLogs
GetQueryResults	`logs:GetQueryResults` Necessario per recuperare i risultati delle query di Logs Insights. CloudWatch
ListEntitiesForLogGroup (autorizzazione solo per CloudWatch console)	`logs:ListEntitiesForLogGroup` Necessario per trovare le entità associate a un gruppo di log. Necessario per esplorare i log correlati all'interno della CloudWatch console.
ListLogGroupsForEntity (autorizzazione CloudWatch solo per la console)	`logs:ListLogGroupsForEntity` Necessario per trovare i gruppi di log associati a un'entità. Necessario per esplorare i log correlati all'interno della CloudWatch console.
ListTagsLogGroup	`logs:ListTagsLogGroup` Necessaria per elencare i tag associati a un gruppo di log.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
ListLogGroups	`logs:DescribeLogGroups` Necessaria per visualizzare tutti i gruppi di log associati all'account.
ProcessWithPipeline (solo autorizzazione)	`logs:ProcessWithPipeline` Necessario per elaborare e trasformare gli eventi di registro tramite trasformatori di pipeline prima dell'archiviazione. Questa autorizzazione deve essere concessa al ruolo IAM passato come ruolo di origine di CloudWatch Logs in una configurazione di pipeline. La risorsa è il gruppo di log ARN.
PutDestination	`logs:PutDestination` Necessaria per creare o aggiornare un flusso di log di destinazione (ad esempio, un flusso Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Necessaria per creare o aggiornare una policy di accesso predefinita associata a una destinazi one di log esistente.
PutLogEvents	`logs:PutLogEvents` Necessaria per caricare un batch di eventi di log in un flusso di eventi.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
PutMetricFilter	`logs:PutMetricFilter` Necessaria per creare o aggiornare un filtro di parametri e associarlo a un gruppo di log.
PutQueryDefinition	`logs:PutQueryDefinition` Necessario per salvare una query in CloudWatch Logs Insights, incluse le query salvate con parametri.
PutResourcePolicy	`logs:PutResourcePolicy` Necessario per creare una policy sulle risorse CloudWatch Logs.
PutRetentionPolicy	`logs:PutRetentionPolicy` Necessaria per impostare il numero di giorni per conservare log eventi (retention) in un gruppo di log.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Necessaria per creare o aggiornare un filtro sottoscrizioni e associarlo a un gruppo di log.
StartQuery	`logs:StartQuery` Necessario per avviare le query di CloudWatch Logs Insights. Per eseguire una query salvata con parametri, è inoltre necessario `logs:DescribeQueryDefinitions` .

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
StopQuery	`logs:StopQuery` Necessario per interrompere una query CloudWatch Logs Insights in corso.
TagLogGroup	`logs:TagLogGroup` Necessaria per aggiungere o aggiornare i tag dei gruppi di log.
TestMetricFilter	`logs:TestMetricFilter` Necessaria per verificare un modello di filtro su un campionamento di messaggi di log eventi.

Utilizzo di ruoli collegati ai servizi per i log CloudWatch

Amazon CloudWatch Logs utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente a Logs. CloudWatch Service-linked i ruoli sono predefiniti da CloudWatch Logs e includono tutte le autorizzazioni necessarie al servizio per chiamare altri AWS servizi per conto dell'utente.

Un ruolo collegato al servizio rende la configurazione CloudWatch dei log più efficiente perché non è necessario aggiungere manualmente le autorizzazioni necessarie. CloudWatch Logs definisce le autorizzazioni dei relativi ruoli collegati al servizio e, se non diversamente definito, solo CloudWatch Logs può assumere tali ruoli. Le autorizzazioni definite includono policy di attendibilità e di autorizzazioni. Queste ultime non possono essere collegate a nessun'altra entità IAM.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi consulta Servizi AWS che funzionano con IAM. 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 i registri CloudWatch

CloudWatch Logs utilizza il ruolo collegato al servizio denominato. AWSServiceRoleForLogDelivery CloudWatch Logs utilizza questo ruolo collegato al servizio per scrivere i log direttamente su Firehose. Per ulteriori informazioni, consulta Abilita la registrazione da AWS services.

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi AWSServiceRoleForLogDelivery considera attendibili i seguenti servizi:

- `logs.amazonaws.com`

La politica delle autorizzazioni dei ruoli consente a CloudWatch Logs di completare le seguenti azioni sulle risorse specificate. AWSServiceRoleForLogDeliveryPolicy Per il documento completo sulla politica JSON, consulta AWSServiceRoleForLogDeliveryPolicy la AWS Managed Policy Reference Guide.

- Azione: `firehose:PutRecordfirehose:PutRecordBatch`, e `firehose:ListTagsForDeliveryStream` su tutti i flussi di distribuzione Firehose che hanno un tag con una `LogDeliveryEnabled` chiave con un valore di. `true` Questo tag viene automaticamente allegato a un flusso di consegna Firehose quando si crea un abbonamento per inviare i log a Firehose.
- Azione: `kms:GenerateDataKey` e `kms:Decrypt` su tutte le AWS KMS chiavi, ma solo quando la richiesta viene effettuata tramite Firehose (applicata dalla chiave di `kms:ViaService` condizione impostata su). `firehose.*.amazonaws.com` Queste autorizzazioni consentono a Logs di inviare CloudWatch i log ai flussi di distribuzione Firehose che utilizzano la crittografia lato server con chiavi gestite dal cliente (). SSE-CMK La policy AWS KMS chiave del cliente deve inoltre concedere in modo indipendente l'accesso al ruolo collegato al servizio.

Per consentire a un'entità IAM, di creare, modificare o eliminare un ruolo collegato ai servizi, devi configurare le autorizzazioni. Questa entità può essere un utente, un gruppo o un ruolo. Per ulteriori informazioni, consulta Service-Linked Role Permissions nella IAM User Guide.

Creazione di un ruolo collegato al servizio per i log CloudWatch

Non è necessario creare manualmente un ruolo collegato ai servizi. Quando imposti i log da inviare direttamente a uno stream Firehose nell'API Console di gestione AWS, nel o nell' AWS API AWS CLI, CloudWatch Logs crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando configuri nuovamente i log per l'invio diretto a uno stream Firehose, CloudWatch Logs crea nuovamente il ruolo collegato al servizio per te.

Modifica di un ruolo collegato al servizio per Logs CloudWatch

CloudWatch Logs non consente di modificare né modificare AWSServiceRoleForLogDelivery alcun altro ruolo collegato al servizio dopo averlo creato. Non è possibile modificare il nome del ruolo poiché varie entità possono farvi riferimento. Tuttavia, utilizzando IAM è possibile modificarne la descrizione. Per ulteriori informazioni, consulta Modifica di un Service-Linked ruolo nella IAM User Guide.

Eliminazione di un ruolo collegato al servizio per i log CloudWatch

Se non è più necessario utilizzare una caratteristica o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare quel ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare manualmente.

> **Note**
>
> Se il servizio CloudWatch Logs utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse CloudWatch Logs utilizzate da AWSServiceRoleForLogDelivery Ruolo collegato al servizio

- Smetti di inviare i log direttamente agli stream Firehose.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Usa la console IAM AWS CLI, l'API o l' AWS API per eliminare il ruolo collegato al AWSServiceRoleForLogDelivery servizio. Per ulteriori informazioni, consulta Eliminazione di un ruolo Service-Linked

Regioni supportate per i ruoli collegati al servizio CloudWatch Logs

CloudWatch Logs supporta l'utilizzo di ruoli collegati ai servizi in tutte le AWS regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta CloudWatch Logs Regions and Endpoints.

CloudWatch Registra gli aggiornamenti di AWS ruoli collegati al servizio

Visualizza i dettagli sugli aggiornamenti del ruolo collegato al AWS servizio per CloudWatch i log da quando questo servizio ha iniziato a tracciare queste modifiche. Per ricevere avvisi automatici sulle modifiche apportate a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei documenti di CloudWatch log.

Modifica	Descrizione	Data
AWSServiceRoleForLogDelivery politica relativa ai ruoli collegata al servizio: aggiornamento a una politica esistente	CloudWatch Log aggiunti `kms:GenerateDataKey` e `kms:Decrypt` autorizzazioni alla policy IAM associate al ruolo collegato al AWSServiceRoleForLogDelivery servizio. Queste autorizzazioni sono coperte dalla chiave di `kms:ViaService` condizione per consentirne l'uso solo tramite Firehose. Questa modifica consente a CloudWatch Logs di inviare i log ai flussi di distribuzione Firehose che utilizzano la crittografia lato server con chiavi gestite dal cliente (). SSE-CMK	15 maggio 2026
AWSServiceRoleForLogDelivery politica relativa ai ruoli collegati ai servizi:	CloudWatch I log hanno modificato le autorizzazioni	15 luglio 2021

Modifica	Descrizione	Data
aggiornamento di una politica esistente	nella policy IAM associata al ruolo collegato al AWSServiceRoleForLogDelivery servizio. È stata apportata la seguente modifica: • La chiave di condizione `firehose:ResourceTag/LogDeliveryEnabled": "true"` è stata modificata in `aws:ResourceTag/LogDeliveryEnabled": "true"` .	
CloudWatch I log hanno iniziato a tenere traccia delle modifiche	CloudWatch Logs ha iniziato a tenere traccia delle modifiche relative alle politiche AWS gestite.	10 giugno 2021

Convalida della conformità per Amazon Logs CloudWatch

La nostra nuova esperienza AWS di iscrizione non è progettata per carichi di lavoro regolamentati. Se stai utilizzando la nostra nuova esperienza di AWS registrazione, ma desideri utilizzarla AWS per carichi di lavoro regolamentati, puoi iscriverti a AWS (avanzato) o attivare funzionalità avanzate per il tuo ambiente. AWS

Per sapere se un Servizio AWS rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program e scegli il programma di conformità che ti interessa. Per informazioni generali, vedere Programmi di AWS conformità Programmi di

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, vedere Scaricamento dei report in AWS Artifact .

La responsabilità dell'utente in materia di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità dell'utente in materia di conformità durante l'utilizzo Servizi AWS, consulta AWS la Documentazione sulla sicurezza.

Resilienza in Amazon Logs CloudWatch

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.

Sicurezza dell'infrastruttura in Amazon CloudWatch Logs

In quanto servizio gestito, Amazon CloudWatch Logs 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.

Si utilizzano chiamate API AWS pubblicate per accedere ai CloudWatch log attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Utilizzo dei CloudWatch log con endpoint VPC di interfaccia

Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e Logs. CloudWatch Puoi utilizzare questa connessione per inviare log a CloudWatch Logs senza inviarli tramite Internet. CloudWatch Logs supporta gli endpoint IPv4 VPC in tutte le regioni e IPv6 supporta gli endpoint in tutte le regioni.

Amazon VPC è un AWS servizio che puoi utilizzare per avviare AWS risorse in una rete virtuale definita dall'utente. Con un VPC;, detieni il controllo delle impostazioni della rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Per connettere il tuo VPC ai CloudWatch log, definisci un endpoint VPC di interfaccia per Logs. CloudWatch Questo tipo di endpoint consente di collegare il VPC ai servizi AWS . L'endpoint fornisce una connettività affidabile e scalabile ai CloudWatch log senza richiedere un gateway Internet, un'istanza NAT (Network Address Translation) o una connessione VPN. Per ulteriori informazioni, consulta Che cos'è Amazon VPC? nella Guida per l'utente di Amazon VPC.

Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, una AWS tecnologia che consente la comunicazione privata tra AWS i servizi utilizzando un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, vedere New — AWS PrivateLink for AWS Services.

Le fasi seguenti sono per gli utenti Amazon VPC. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base nella Guida per l'utente di Amazon VPC.

Disponibilità

CloudWatch Logs attualmente supporta gli endpoint VPC in AWS tutte le regioni, incluse le regioni. AWS GovCloud (US)

Creazione di un endpoint VPC per i log CloudWatch

Per iniziare a utilizzare CloudWatch Logs con il tuo VPC, crea un endpoint VPC di interfaccia per Logs. CloudWatch Il servizio da scegliere è com.amazonaws. ***Region***.registri. Per connettersi con un endpoint FIPS, il servizio da scegliere è. `com.amazonaws.Region.logs-fips` Non è necessario modificare alcuna impostazione per CloudWatch i registri. Per ulteriori informazioni, consulta Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

Alcuni CloudWatch log APIs, come StartLiveTail e GetLogObject, sono ospitati su un endpoint e un endpoint VPC diversi:. `stream-logs.Region.amazonaws.com` Per creare un endpoint VPC di interfaccia per questi APIs, il servizio da scegliere è com.amazonaws. ***Region***.stream-logs. Per connettersi con un endpoint FIPS, il servizio da scegliere è. `com.amazonaws.Region.stream-logs-fips`

Verifica della connessione tra il tuo VPC e Logs CloudWatch

Dopo aver creato l'endpoint, è possibile testare la connessione.

Per testare la connessione tra il tuo VPC e l'endpoint Logs CloudWatch

1. Connettiti a un'istanza Amazon EC2 che risiede nel tuo VPC. Per informazioni sulla connessione, consulta Connessione all'istanza Linux o Connessione all'istanza Windows nella documentazione Amazon EC2.
2. Dall'istanza, usa AWS CLI per creare una voce di registro in uno dei gruppi di log esistenti.

   Prima di tutto, creare un file JSON con un evento di log. Il timestamp deve essere specificato come numero di millisecondi dopo il 1° gennaio 1970 alle 00:00:00 UTC.

   ```
   [
     {
       "timestamp": 1533854071310,
       "message": "VPC Connection Test"
     }
   ]
   ```

   Quindi, utilizzare il comando `put-log-events` per creare la voce di log:

   ```
   aws logs put-log-events --log-group-name LogGroupName --log-stream-
   name LogStreamName --log-events file://JSONFileName
   ```

   Se la risposta al comando include `nextSequenceToken`, il comando è riuscito e l'endpoint VPC sta funzionando.

Controllo dell'accesso all'endpoint VPC CloudWatch Logs

Una policy endpoint VPC è una policy della risorsa IAM che viene collegata a un endpoint durante la creazione o la modifica dell'endpoint. Se non colleghi una policy durante la creazione di un endpoint, viene collegata una policy predefinita che consente l'accesso completo al servizio. Una policy endpoint non esclude né sostituisce policy IAM o policy specifiche del servizio. Si tratta di una policy separata per controllare l'accesso dall'endpoint al servizio specificato.

Le policy endpoint devono essere scritte in formato JSON.

Per ulteriori informazioni, consultare Controllo degli accessi ai servizi con endpoint VPC nella Guida per l'utente di Amazon VPC.

Di seguito è riportato un esempio di policy sugli endpoint per Logs. CloudWatch Questa policy consente agli utenti che si connettono ai CloudWatch log tramite il VPC di creare flussi di log e inviare log CloudWatch ai log, e impedisce loro di eseguire altre azioni di log. CloudWatch

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Per modificare la policy degli endpoint VPC per Logs CloudWatch

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione, seleziona Endpoint.
3. Se non hai ancora creato l'endpoint for CloudWatch Logs, scegli Crea endpoint. Quindi seleziona com.amazonaws. *Region*.logs e scegli Crea endpoint.
4. Seleziona com.amazonaws. ***Region***.logs endpoint e scegli la scheda Policy nella metà inferiore dello schermo.
5. Scegli Edit Policy (Modifica policy) e apporta le modifiche alla policy.

Supporto delle chiavi di contesto VPC

CloudWatch Logs supporta `aws:SourceVpc` le chiavi di `aws:SourceVpce` contesto che possono limitare l'accesso a endpoint VPC specifici VPCs o specifici. Queste chiavi funzionano solo se l'utente utilizza endpoint VPC. Per ulteriori informazioni, consulta Chiavi disponibili per alcuni servizi nella Guida per l'utente di IAM.

Logging CloudWatch Logs, API e operazioni della console in AWS CloudTrail

Amazon CloudWatch Logs è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o un Servizio AWS. CloudTrail acquisisce le chiamate API per i CloudWatch log come eventi. Le chiamate acquisite includono chiamate dalla console CloudWatch Logs e chiamate di codice alle operazioni dell'API CloudWatch Logs. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata a CloudWatch Logs, l'indirizzo IP da cui è stata effettuata, quando è stata effettuata e ulteriori dettagli.

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 utente root o utente.
- Se la richiesta è stata effettuata per conto di un utente del Centro identità IAM.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro Servizio AWS.

CloudTrail è attivo nel tuo account Account AWS quando crei l'account e hai automaticamente accesso alla cronologia degli CloudTrail eventi. La cronologia CloudTrail degli eventi fornisce un record visualizzabile, ricercabile, scaricabile e immutabile degli ultimi 90 giorni di eventi di gestione registrati in un. Regione AWSPer ulteriori informazioni, consulta Lavorare con la cronologia degli CloudTrail eventi nella Guida per l'utente.AWS CloudTrail Non sono CloudTrail previsti costi per la visualizzazione della cronologia degli eventi.

Per una registrazione continua degli eventi degli Account AWS ultimi 90 giorni, crea un trail o un data store di eventi CloudTrailLake.

CloudTrail sentieri

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Tutti i percorsi creati utilizzando il Console di gestione AWS sono multiregionali. È possibile creare un trail per una singola Regione o per più Regioni tramite AWS CLI. La creazione di un percorso multiregionale è consigliata in quanto consente di registrare l'intera attività del proprio Regioni AWS account. Se si crea un trail per una singola Regione, è possibile visualizzare solo gli eventi registrati nella

Regione AWS del trail. Per ulteriori informazioni sui trail, consulta Creating a trail for your Account AWS e Creating a trail for an organization nella Guida per l'utente di AWS CloudTrail .

Puoi inviare gratuitamente una copia dei tuoi eventi di gestione in corso al tuo bucket Amazon S3 CloudTrail creando un percorso, tuttavia ci sono costi di storage di Amazon S3. Per ulteriori informazioni sui CloudTrail prezzi, consulta la pagina Prezzi.AWS CloudTrail Per informazioni sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

CloudTrail Archivi di dati sugli eventi di Lake

CloudTrail Lake ti consente di eseguire query basate su SQL sui tuoi eventi. CloudTrail Lake converte gli eventi esistenti in formato JSON basato su righe in formato Apache ORC. ORC è un formato di archiviazione a colonne ottimizzato per il recupero rapido dei dati. Gli eventi vengono aggregati in archivi di dati degli eventi, che sono raccolte di eventi immutabili basate sui criteri selezionati applicando i selettori di eventi avanzati. I selettori applicati a un archivio di dati degli eventi controllano quali eventi persistono e sono disponibili per l'esecuzione della query. Per ulteriori informazioni su CloudTrail Lake, consulta Working with AWS CloudTrail Lake nella Guida per l'utente.AWS CloudTrail

CloudTrail Gli archivi e le richieste di dati sugli eventi di Lake comportano dei costi. Quando crei un datastore di eventi, scegli l'opzione di prezzo da utilizzare per tale datastore. L'opzione di prezzo determina il costo per l'importazione e l'archiviazione degli eventi, nonché il periodo di conservazione predefinito e quello massimo per il datastore di eventi. Per ulteriori informazioni sui CloudTrail prezzi, consulta Prezzi.AWS CloudTrail

CloudWatch Logs supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- AssociateKmsKey
- CancelExportTask
- CreateDelivery
- CreateExportTask
- CreateLogAnomalyDetector
- CreateLogGroup
- CreateLogStream
- DeleteAccountPolicy
- DeleteDataProtectionPolicy

- DeleteDelivery
- DeleteDeliveryDestination
- DeleteDeliveryDestinationPolicy
- DeleteDeliverySource
- DeleteDestination
- DeleteIndexPolicy
- DeleteIntegration
- DeleteLogAnomalyDetector
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteQueryDefinition
- DeleteResourcePolicy
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- DeleteTransformer
- DescribeAccountPolicies
- DescribeConfigurationTemplates
- DescribeDeliveries
- DescribeDeliveryDestinations
- DescribeDeliverySources
- DescribeDestinations
- DescribeExportTasks
- DescribeFieldIndexes
- DescribeIndexPolicies
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries

- DescribeQueryDefinitions
- DescribeResourcePolicies
- DescribeSubscriptionFilters
- DisassociateKmsKey
- FilterLogEvents
- GetDataProtectionPolicy
- GetDelivery
- GetDeliveryDestination
- GetDeliveryDestinationPolicy
- GetDeliverySource
- GetIntegration
- GetLogAnomalyDetector
- GetLogEvents
- GetLogGroupFields
- GetLogRecord
- GetQueryResults
- GetTransformer
- ListAnomalies
- ListIntegrations
- ListLogAnomalyDetectors
- ListLogGroups
- ListLogGroupsForQuery
- ListTagsForResource
- ListTagsLogGroup
- PutAccountPolicy
- PutDataProtectionPolicy
- PutDeliveryDestination
- PutDeliveryDestinationPolicy
- PutDeliverySource

- PutDestination
- PutDestinationPolicy
- PutIndexPolicy
- PutIntegration
- PutMetricFilter
- PutQueryDefinition
- PutResourcePolicy
- PutRetentionPolicy
- PutSubscriptionFilter
- PutTransformer
- StartLiveTail
- StartQuery
- StopQuery
- TagResource
- TestMetricFilter
- TestTransformer
- UntagResource
- UpdateAnomaly
- UpdateDeliveryConfiguration
- UpdateLogAnomalyDetector

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.

Informazioni sulla generazione di query in CloudTrail

CloudTrail È supportata anche la registrazione degli eventi della console del generatore di query. Il generatore di query è attualmente supportato per CloudWatch Logs Insights e CloudWatch Metric Insights. In questi CloudTrail eventi, lo è. `eventSource` `monitoring.amazonaws.com`

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'GenerateQueryazione in CloudWatch Logs Insights.


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId": "123456789012",
                "userName": "SAMPLE_NAME"
            },
            "attributes": {
                "creationDate": "2020-04-08T21:43:24Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "monitoring.amazonaws.com",
    "eventName": "GenerateQuery",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "exampleUserAgent",
    "requestParameters": {
        "query_ask": "***",
        "query_type": "LogsInsights",
        "logs_insights": {
            "fields": "***",
            "log_group_names": ["yourloggroup"]
```

```
        },
        "include_description": true
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Informazioni sulle voci dei file di log di

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

La seguente voce del file di registro mostra che un utente ha chiamato l'azione CloudWatch Logs. CreateExportTask

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
```

```
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

Monitoraggio con CloudWatch metriche

Puoi utilizzare le tabelle in questa sezione per esaminare i parametri che Amazon CloudWatch Logs invia ad Amazon CloudWatch ogni minuto.

CloudWatch Parametri dei log

Lo spazio dei nomi `AWS/Logs` include le metriche descritte di seguito.

Metrica	Description
`CallCount`	Il numero di operazioni API specificate eseguite nel tuo account. `CallCount` è una metrica di CloudWatch utilizzo del servizio Logs. Per ulteriori informazioni, consulta CloudWatch Registra le metriche di utilizzo del servizio. Dimensioni valide: Classe, Risorsa, Servizio, Tipo Statistiche valide: Sum Unità: nessuna
`DeliveryErrors`	Il numero di eventi di registro per i quali CloudWatch Logs ha ricevuto un errore durante l'inoltro dei dati alla destinazione dell'abbonamento. Se il servizio di destinazione restituisce un errore riutilizzabile, ad esempio un'eccezione di limitazione o un'eccezione del servizio riutilizzabile (ad esempio HTTP 5xx), CloudWatch Logs continua a ritentare la consegna per un massimo di 24 ore. CloudWatch Logs non tenta di ripetere la consegna se l'errore è un errore irreversibile, ad esempio o. `AccessDeniedException` `ResourceNotFoundException` Dimensioni valide:,,, LogGroupName DestinationType FilterName PolicyLevel Statistiche valide: Sum Unità: nessuna

Metrica	Description
`DeliveryThrottling`	Il numero di eventi di registro per i quali CloudWatch Logs è stato limitato durante l'inoltro dei dati alla destinazione dell'abbonamento. Se il servizio di destinazione restituisce un errore riutilizzabile, ad esempio un'eccezione di limitazione o un'eccezione del servizio riutilizz abile (ad esempio HTTP 5xx), Logs continua a ritentare la consegna per un massimo di 24 ore. CloudWatch CloudWatch Logs non tenta di ripetere la consegna se l'errore è un errore irreversibile, ad esempio o. `AccessDeniedException` `ResourceNotFoundException` Dimensioni valide:,,, LogGroupName DestinationType FilterName PolicyLevel Statistiche valide: Sum Unità: nessuna
`EMFDisabledErrors`	Il numero di eventi di registro in formato EMF che sono stati ignorati a causa di una policy di account attiva di tipo METRIC_EXTRACTION_ POLICY che corrisponde al gruppo di log. Per ulteriori informazioni sulle politiche di estrazione delle metriche, vedere. PutAccountPolicy Dimensioni valide: `LogGroupName` Statistiche valide: Sum Unità: nessuna

Metrica	Description
`EMFParsingErrors`	Il numero di errori di analisi riscontrati durante l'elaborazione dei log in formato dei parametri incorporato. Tali errori si verificano quando i registri vengono identificati come formato dei parametri incorporato ma non seguono il formato corretto. Per ulteriori informazioni sul formato dei parametri incorporato, consulta la sezione Specification: Embedded metric format (Specifica: formato dei parametri incorporato). Dimensioni valide: `LogGroupName` Statistiche valide: Sum Unità: nessuna
`EMFValidationErrors`	Il numero di errori di convalida riscontrati durante l'elaborazione dei log in formato dei parametri incorporato. Questi errori si verificano quando le definizioni dei parametri nei log in formato dei parametri incorporato non aderiscono al formato dei parametri incorporato e alle specifiche di `MetricDatum` . Per informazioni sul formato metrico incorporato, vedere Specificazione: formato metrico incorporato. CloudWatch Per informazioni sul tipo di dati`MetricDatum` , MetricDatumconsulta Amazon CloudWatch API Reference. **Note** Alcuni errori di convalida possono causare la mancata pubblicazione di più metriche all'interno di un log EMF. Ad esempio, tutti i parametri impostati con un namespace non valido verranno eliminati. Dimensioni valide: `LogGroupName` Statistiche valide: Sum Unità: nessuna

Metrica	Description
`ErrorCount`	Il numero di operazioni API eseguite nel tuo account che sono risultate in errori. `ErrorCount` è una metrica di utilizzo del servizio CloudWatch Logs. Per ulteriori informazioni, consulta CloudWatch Registra le metriche di utilizzo del servizio. Dimensioni valide: Classe, Risorsa, Servizio, Tipo Statistiche valide: Sum Unità: nessuna
`ForwardedBytes`	Il volume dei log eventi in byte compressi inoltrati alla destinazione di sottoscrizione. Dimensioni valide: LogGroupName,, DestinationType FilterName Statistiche valide: Sum Unità: byte
`Forwarded LogEvents`	Il numero dei log eventi inoltrati alla destinazione di sottoscrizione. Dimensioni valide: LogGroupName, DestinationType, FilterName, PolicyLevel Statistiche valide: Sum Unità: nessuna

Metrica	Description
`IncomingBytes`	Il volume degli eventi di registro in byte non compressi caricati in Logs. CloudWatch Se utilizzato con la dimensione `LogGroupName` , corrispon de al volume degli eventi di log in byte non compressi caricati nel gruppo di log. Dimensioni valide: LogGroupName Statistiche valide: Sum Unità: byte
`IncomingL ogEvents`	Il numero di eventi di registro caricati in CloudWatch Logs. Se utilizzato con la dimensione `LogGroupName` , corrisponde al numero di eventi di log caricati nel gruppo di log. Dimensioni valide: LogGroupName Statistiche valide: Sum Unità: nessuna
`LogEvents WithFindings`	Il numero di eventi di registro corrispondenti a una stringa di dati che si sta controllando utilizzando la funzionalità di protezione dei dati dei CloudWatch registri. Per ulteriori informazioni, consulta Incremento della protezione dei dati di log sensibili con il mascheramento. Dimensioni valide: nessuna Statistiche valide: Sum Unità: nessuna

Metrica	Description
`ThrottleCount`	Il numero di operazioni API eseguite nell'account che sono state limitate a causa delle quote di utilizzo. `ThrottleCount` è una metrica di utilizzo del CloudWatch servizio Logs. Per ulteriori informazioni, consulta CloudWatch Registra le metriche di utilizzo del servizio. Dimensioni valide: Classe, Risorsa, Servizio, Tipo Statistiche valide: Sum Unità: nessuna

Dimensioni per le metriche di Logs CloudWatch

Le dimensioni che è possibile utilizzare con la maggior parte delle metriche di CloudWatch Logs sono elencate nella tabella seguente.

Dimensione	Description
`LogGroupName`	Il nome del gruppo di CloudWatch log Logs per il quale visualizzare le metriche.
`DestinationType`	La destinazione dell'abbonamento per i dati di CloudWatch Logs, che può essere AWS Lambda Amazon Kinesis Data Streams o Amazon Data Firehose.
`FilterName`	Nome del filtro di sottoscrizione che sta inviando dati dal gruppo di log alla destinazione. Il nome del filtro di abbonamento viene convertito CloudWatch automaticamente in ASCII e tutti i caratteri non supportati vengono sostituiti con un punto interrogativo (?).

Dimensioni metriche del filtro di abbonamento

Le dimensioni per le metriche relative ai filtri di sottoscrizione a livello di account sono elencate nella tabella seguente.

Dimensione	Description
`PolicyLevel`	Il livello a cui si applica la politica. Attualmente, l'unico valore valido per questa dimensione è `AccountPolicy`
`DestinationType`	La destinazione dell'abbonamento per i dati di CloudWatch Logs, che può essere AWS Lambda Amazon Kinesis Data Streams o Amazon Data Firehose.
`FilterName`	Nome del filtro di sottoscrizione che sta inviando dati dal gruppo di log alla destinazione. Il nome del filtro di abbonamento viene convertito CloudWatch automaticamente in ASCII e tutti i caratteri non supportati vengono sostituiti con un punto interroga tivo (?).

Metriche e dimensioni del trasformatore di log

CloudWatch Logs pubblica le seguenti metriche del trasformatore di log nel namespace. CloudWatch `AWS/Logs`

Metrica	Description
`TransformationErrors`	Il numero di errori riscontrati durante la trasformazione degli eventi di registro con il trasformatore specificato. Unità: nessuna Statistica valida: Sum
`TransformedBytes`	Il volume dell'output degli eventi di registro trasformati, in byte non compressi. Unità: byte Statistica valida: Sum

Metrica	Description
`TransformedLogEvents`	Il numero di eventi di registro trasformati. Unità: nessuna Statistica valida: Sum

Le seguenti dimensioni vengono utilizzate dalle metriche dei trasformatori.

Dimensione	Description
`LogGroupname`	Questa dimensione viene utilizzata solo per log-group-level i trasformatori.
`PolicyLevel`	Questa dimensione viene utilizzata solo per i trasformatori a livello di account. Attualmente l'unico valore valido per questa dimensione è `AccountPolicy`

Metriche e dimensioni di centralizzazione

Per il monitoraggio centralizzato su più account e regioni, puoi utilizzare CloudWatch Logs Centralization per consolidare i dati e le metriche dei log in una posizione centrale. Per ulteriori informazioni, consulta Cross-account Centralizzazione dei log interregionali.

CloudWatch Logs pubblica le seguenti metriche di centralizzazione nel namespace. CloudWatch `AWS/Logs` Queste metriche consentono di monitorare la replica dei dati di registro dagli account di origine agli account di destinazione quando si utilizzano le regole di centralizzazione di Logs. CloudWatch

> **Note**
>
> CloudWatch inizia a riportare le metriche di centralizzazione poco dopo la creazione di una regola di centralizzazione. Le metriche vengono pubblicate al meglio e tengono traccia solo dei nuovi eventi di registro che arrivano dopo la creazione della regola di centralizzazione.

Metrica	Description	Pubblicato in
`IncomingCopiedBytes`	Il volume di dati di registro in byte non compressi che è stato replicato nell'account di destinazione. Questa metrica si applica solo ai nuovi eventi di registro che arrivano dopo la creazione della regola di centralizzazione. Dimensioni valide:, SourceRegion SourceAccount Statistiche valide: Sum Unità: byte	Account di destinazione
`IncomingCopiedLogEvents`	Il numero di eventi di registro che sono stati replicati nell'account di destinazione. Questa metrica si applica solo ai nuovi eventi di registro che arrivano dopo la creazione della regola di centralizzazione. Dimensioni valide:, SourceRegion SourceAccount Statistiche valide: Sum Unità: nessuna	Account di destinazione
`OutgoingCopiedBytes`	Il volume di dati di registro in byte non compressi che è stato inviato dagli account di origine all'account di destinazione. Questa metrica si applica solo ai nuovi eventi di registro che vengono copiati nella destinazione dopo la creazione della regola di centralizzazione. Dimensioni valide: DestinationRegion Statistiche valide: Sum	Account di origine

Metrica	Description	Pubblicato in
	Unità: byte	
`OutgoingCopiedLogEvents`	Il numero di eventi di registro inviati dagli account di origine all'account di destinazione. Questa metrica si applica solo ai nuovi eventi di registro che vengono copiati nella destinazione dopo la creazione della regola di centralizzazione. Dimensioni valide: DestinationRegion Statistiche valide: Sum Unità: nessuna	Account di origine
`CentralizationError`	Il numero di errori riscontrati durante la replica dei dati di registro. Gli errori possono verificarsi a causa di vari motivi, ad esempio problemi di autorizzazione delle chiavi KMS, limiti di quota per i gruppi di log o mancate corrispondenze tra i livelli di registro. Per identificare gruppi o flussi di log specifici la cui replica non è riuscita e i relativi motivi, monitora la dimensione. `ErrorType` Dimensioni valide: ErrorType Statistiche valide: Sum Unità: nessuna	Account di destinazione

Metrica	Description	Pubblicato in
`CentralizationThrottled`	Il numero di volte in cui l'elaborazione della centralizzazione è stata limitata. La limitazione comporta un rallentamento dell'elaborazione della centralizzazione, ma non impedisce la replica dei dati di registro. Dimensioni valide: DestinationRegion Statistiche valide: Sum Unità: nessuna	Account di origine

Le seguenti dimensioni vengono utilizzate dalle metriche di centralizzazione.

Dimensione	Description
`SourceRegion`	La AWS regione in cui hanno avuto origine i dati del registro di origine.
`SourceAccount`	L'ID dell' AWS account da cui hanno avuto origine i dati del registro di origine.
`DestinationRegion`	La AWS regione in cui vengono replicati i dati di registro.
`ErrorType`	Il tipo di errore riscontrato durante la centralizzazione. I valori possibili includono: • `LogGroupQuotaExceeded` : L'account di destinazione ha raggiunto la quota del gruppo di log. • `InvalidKMS` : La chiave KMS è mancante, eliminata, non abilitata o è asimmetrica anziché simmetrica. • `AccessDenied` : l'accesso è stato negato durante il tentativo di replicare i dati di registro. Ciò può verificarsi a causa di autorizzazioni insufficienti, ad esempio autorizzazioni errate per le chiavi KMS, autorizzazioni IAM mancanti o politiche restrittive in materia di risorse.

Dimensione	Description
	• `LogTierMismatch` : il livello di registro dei gruppi di log di origine e di destinazione non corrisponde. • `InvalidLogStream` : parametro non valido rilevato durante la creazione o la scrittura in un flusso di log, ad esempio quando il nome del flusso di log supera la lunghezza massima. • `InvalidLogGroup` : parametro non valido rilevato durante la creazione o la configurazione di un gruppo di log nell'acco unt di destinazione. • `DestinationEncryptionMismatch` : Il gruppo di log di destinazione esiste già con una configurazione di crittogra fia KMS diversa da quella specificata dalla regola di centraliz zazione.

CloudWatch Registra le metriche di utilizzo del servizio

CloudWatch Logs invia metriche CloudWatch che tengono traccia dell'utilizzo delle operazioni dell'API CloudWatch Logs. Queste metriche corrispondono alle quote di servizio. AWS Il monitoraggio di questi parametri consente di gestire in modo proattivo le tue quote. Per ulteriori informazioni, consulta Service Quotas Integration and Usage Metrics (Integrazione di quote di servizio e parametri di utilizzo).

Ad esempio, è possibile tenere traccia dei parametri `ThrottleCount` o imposta un allarme per tale parametro. Se il valore di questo parametro aumenta, è consigliabile richiedere un aumento di quota per l'operazione API che viene limitata. Per ulteriori informazioni sulle quote del servizio CloudWatch Logs, vedere. CloudWatch Registra le quote

CloudWatch Logs pubblica le metriche di utilizzo delle quote di servizio ogni minuto sia nel namespace che nel namespace. `AWS/Usage` `AWS/Logs`

La tabella seguente elenca le metriche sull'utilizzo del servizio pubblicate da Logs. CloudWatch Questi parametri non hanno un'unità specificata. La statistica più utile per questi parametri è `SUM`, che rappresenta il conteggio totale delle operazioni per il periodo di 1 minuto.

Ciascuno di questi parametri vengono pubblicati con i valori per tutte le dimensioni di `Service`, `Class`, `Type`, e `Resource`. Vengono inoltre pubblicati con una singola dimensione chiamata `Account Metrics`. Utilizzare la dimensione `Account Metrics` per visualizzare la somma dei parametri per tutte le operazioni API nel tuo account. Utilizzare le altre dimensioni e specificare il nome di un'operazione API per la dimensione `Resource` per trovare i parametri per quella particolare API.

Parametri

Metrica	Description
`CallCount`	Il numero di operazioni specificate eseguite nel tuo account. `CallCount` è pubblicato in entrambi spazi dei nomi `AWS/Usage` e `AWS/Logs`.
`ErrorCount`	Il numero di operazioni API eseguite nel tuo account che sono risultate in errori. `ErrorCount` è pubblicato solo nel `AWS/Logs`.
`ThrottleCount`	Il numero di operazioni API eseguite nell'account che sono state limitate a causa delle quote di utilizzo. `ThrottleCount` è pubblicato solo nel `AWS/Logs`.

Dimensioni

Dimensione	Description
`Account metrics`	Utilizza questa dimensione per ottenere una somma della metrica in tutti i log. CloudWatch APIs Se si desidera visualizzare i parametri per una determinata API, utilizzar e le altre dimensioni elencate in questa tabella e specificare il nome API come valore di `Resource`.

Dimensione	Description
`Service`	Il nome del AWS servizio che contiene la risorsa. Per le metriche di utilizzo dei CloudWatch log, il valore per questa dimensione è. `Logs`
`Class`	La classe di risorsa monitorata. CloudWatch Le metriche di utilizzo dell'API di Logs utilizzano questa dimensione con un valore di. `None`
`Type`	Il tipo di risorsa monitorata. Attualmente, quando la dimensione `Service` è `Logs`, l'unico valore valido per `Type` è `API`.
`Resource`	Il nome dell'operazione API. I valori validi includono tutti i nomi delle operazioni API elencati in Actions (Operazioni). Ad esempio, `PutLogEvents`

CloudWatch Registra le quote

Puoi utilizzare la tabella in questa sezione per esaminare le quote di servizio predefinite, note anche come limiti, per un AWS account in Amazon CloudWatch Logs. La maggior parte delle quote di servizio, ma non tutte, sono elencate nello spazio dei nomi Amazon CloudWatch Logs nella console Service Quotas.

> **Note**
>
> Se desideri richiedere un aumento della quota per una di queste quote, consulta la procedura descritta in questa sezione.

Nome	Predefinita	Adatta e	Description
Attività di esportazione attiva	Ogni regione supportata: 1	No	Il numero di attività di esportazione (in esecuzione o in attesa) attive per account
AssociateKmsKey limite di acceleraz ione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate associate-kms-key al secondo per regione account/per
AssociateSourceToS3TableIntegration limite di accelerazione nelle transazioni al secondo	Ogni Regione supportata: 5	No	Il numero massimo di AssociateSourceToS 3TableIntegration chiamate al secondo per regione account/per
AssociateSourceToS3TableIntegration limite di accelerazione delle transazioni al secondo in una raffica	Ogni regione supportata: 10	No	Il numero massimo di AssociateSourceToS 3TableIntegration chiamate al secondo per

Nome	Predefinita	Adatta e	Description
			account/per regione in una raffica
Dimensione batch	Ogni regione supportata: 1 MB	No	La dimensione massima del batch in MB di una richiesta put-log-events
CancelExportTask limite di acceleraz ione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate cancel-export-task al secondo per regione account/per
CancelImportTask limite di acceleraz ione nelle transazioni al secondo	Ogni regione supportata: 1	No	Il numero massimo di CancelImportTask chiamate al secondo per regione account/per
CancelImportTask limite di acceleraz ione delle transazioni al secondo in una raffica	Ogni regione supportata: 1	No	Il numero massimo di CancelImportTask chiamate al secondo per account/per regione in una raffica
CreateExportTask limite di acceleraz ione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate create-export-task al secondo per regione account/per
CreateImportTask limite di acceleraz ione nelle transazioni al secondo	Ogni regione supportata: 1	No	Il numero massimo di CreateImportTask chiamate al secondo per regione account/per

Nome	Predefinita	Adatta e	Description
CreateImportTask limite di acceleraz ione delle transazioni al secondo in una raffica	Ogni regione supportata: 1	No	Il numero massimo di CreateImportTask chiamate al secondo per account/per regione in una raffica
CreateLogGroup limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 10 al secondo	Sì	Il numero massimo di chiamate create-log-group al secondo per regione account/per
CreateLogStream limite di acceleraz ione nelle transazioni al secondo	Ogni regione supportata: 50 al secondo	Sì	Il numero massimo di chiamate create-log-stream al secondo per regione account/per
Archiviazione di dati	Ogni regione supportata: 5 GB	No	La dimensione massima in GB di archiviazione dei dati gratuita
DeleteDataProtectionPolicy limite di accelerazione nelle transazioni al secondo	Ogni Regione supportata: 5	No	Il numero massimo di chiamate delete-da ta-protection-policy al secondo per regione account/per
DeleteDestination limite di acceleraz ione delle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate di eliminazi one di destinazione al secondo per regione account/per

Nome	Predefinita	Adatta e	Description
DeleteLogGroup limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 10 al secondo	Sì	Il numero massimo di chiamate delete-log-group al secondo per regione account/per
DeleteLogStream limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 15 al secondo	Sì	Il numero massimo di chiamate delete-log-stream al secondo per regione account/per
DeleteMetricFilter limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate delete-metric-filter al secondo per regione account/per
DeleteRetentionPolicy limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate delete-retention-policy al secondo per regione account/per
DeleteSubscriptionFilter limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate delete-subscription-filter al secondo per regione account/per
DeleteTransformer limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate delete-transformer al secondo per regione account/per
DescribeDestinations limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate describe-destination al secondo per regione account/per

Nome	Predefinita	Adatta e	Description
DescribeExportTasks limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate describe-export-tasks al secondo per regione account/per
DescribeImportTaskBatches limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 10	No	Il numero massimo di DescribeImportTask Batches chiamate al secondo per regione account/per
DescribeImportTaskBatches limite di accelerazione delle transazioni al secondo in una raffica	Ogni regione supportata: 10	No	Il numero massimo di DescribeImportTask Batches chiamate al secondo per account/per regione in una raffica
DescribeImportTasks limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 10	No	Il numero massimo di DescribeImportTasks chiamate al secondo per regione account/per
DescribeImportTasks limite di accelerazione delle transazioni al secondo in una raffica	Ogni regione supportata: 10	No	Il numero massimo di DescribeImportTasks chiamate al secondo per account/per regione in una raffica
DescribeLogGroups limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 10 al secondo	Sì	Il numero massimo di chiamate describe-log-groups al secondo per regione account/per

Nome	Predefinita	Adatta e	Description
DescribeLogStreams limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 25 al secondo	Sì	Il numero massimo di chiamate describe-log-streams al secondo per regione account/per
DescribeMetricFilters limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate describe-metric-filters al secondo per regione account/per
DescribeSubscriptionFilters limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate describe-subscription-filters al secondo per regione account/per
DisassociateSourceFromS3TableIntegration limite di accelerazione nelle transazioni al secondo	Ogni Regione supportata: 5	No	Il numero massimo di DisassociateSourceFromS3TableIntegration chiamate al secondo per regione account/per
DisassociateSourceFromS3TableIntegration limite di accelerazione delle transazioni al secondo in una raffica	Ogni regione supportata: 10	No	Il numero massimo di DisassociateSourceFromS3TableIntegration chiamate al secondo per account/per regione in una raffica
Dimensioni dell'evento	Ogni regione supportata: 1.024 KB	No	La dimensione massima dell'evento di log in KB

Nome	Predefinita	Adatta e	Description
FilterLogEvents limite di accelerazione nelle transazioni al secondo	us-east-1: 25 al secondo ap-northeast-3: 5 al secondo ap-southeast-3:5 al secondo ca-west-1:5 al secondo eu-central-1: 5 al secondo il-central-1:5 al secondo Ogni altra regione supportata: 10 al secondo	No	Il numero massimo di chiamate filter-log-events al secondo per regione account/per
GetDataProtectionPolicy limite di accelerazione nelle transazioni al secondo	Ogni Regione supportata: 5	No	Il numero massimo di chiamate get-data-protection-policy al secondo per regione account/per

Nome	Predefinita	Adattabile	Description
GetLogEvents limite di accelerazione delle transazioni al secondo	us-west-2: 10 al secondo ap-northeast-3: 10 al secondo ap-southeast-3:10 al secondo ca-west-1:10 al secondo eu-central-1: 10 al secondo us-west-1: 10 al secondo us-west-3: 30 al secondo il-central-1:10 al secondo Ogni altra regione supportata: 25 al secondo	No	Il numero massimo di chiamate get-log-events al secondo per regione account/per
GetQueryResults limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 10	No	Il numero massimo di chiamate get-query-results al secondo per regione account/per

Nome	Predefinita	Adatta e	Description
GetTransformer limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate get-transformer al secondo per regione account/per
ListSourcesForS3TableIntegration limite di accelerazione nelle transazioni al secondo	Ogni Regione supportata: 5	No	Il numero massimo di ListSourcesForS3TableIntegration chiamate al secondo per regione account/per
ListSourcesForS3TableIntegration limite di accelerazione delle transazioni al secondo in una raffica	Ogni regione supportata: 10	No	Il numero massimo di ListSourcesForS3TableIntegration chiamate al secondo per account/per regione in una raffica
ListTagsForResource limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 30 al secondo	No	Il numero massimo di chiamate list-tags-for-resource al secondo per regione account/per
ListTagsLogGroup limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate list-tags-log-group al secondo per regione account/per
Limite di sessioni simultanee di Live Tail	Ogni regione supportata: 15	Sì	Il numero massimo di sessioni Live Tail attive simultanee per account
Gruppi di log	Ogni regione supportata: 1.000.000	Sì	Il numero massimo di gruppi di log che può avere un account.

Nome	Predefinita	Adattabile	Description
Gruppi di log scansionati per sessione di Live Tail	Ogni regione supportata: 10	No	Il numero massimo di gruppi di log che possono essere scansionati per sessione di Live Tail
Filtri di parametri per gruppo di log	Ogni regione supportata: 100	No	Il numero di filtri di parametri per gruppo di log
PutBearerTokenAuthentication limite di accelerazione nelle transazioni al secondo	Ogni Regione supportata: 5	No	Il numero massimo di chiamate di autenticazione put-bearer-token al secondo per regione account/per
PutBearerTokenAuthentication limite di accelerazione delle transazioni al secondo in una raffica	Ogni regione supportata: 10	No	Il numero massimo di chiamate di autenticazione put-bearer-token al secondo per regione in una raffica account/per
PutDataProtectionPolicy limite di accelerazione nelle transazioni al secondo	Ogni Regione supportata: 5	No	Il numero massimo di chiamate put-data-protection-policy al secondo per regione account/per
PutDestination limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate put-destination al secondo per regione account/per

Nome	Predefinita	Adatta e	Description
PutDestinationPolicy limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate put-destination-policy al secondo per regione account/per
PutLogEvents limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5.000 al secondo	Sì	Il numero massimo di chiamate put-log-events al secondo per regione account/per
PutMetricFilter limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate put-metric-filter al secondo per regione account/per
PutRetentionPolicy limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate put-retention-policy al secondo per regione account/per
PutSubscriptionFilter limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate put-subscription-filter al secondo per regione account/per
PutTransformer limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate put-transformer al secondo per regione account/per
Policy delle risorse	Ogni regione supportata: 10	No	Il numero massimo di politiche relative alle risorse per regione account/per

Nome	Predefinita	Adatta e	Description
StartLiveTail limite di accelerazione delle transazioni al secondo	Ogni Regione supportata: 5	No	Il numero massimo di chiamate start-live-tail al secondo per regione. account/per Questo limite si applica indipende ntemente sia alla console che all'API
StartQuery limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 10	No	Il numero massimo di chiamate di start-query al secondo per regione account/per
Filtri di sottoscrizione per gruppo di log	Ogni Regione supportata: 5	No	Il numero di filtri di sottoscrizione per gruppo di log
TagLogGroup limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate tag-log-group al secondo per regione account/per
TagResource limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate tag-resource al secondo per regione account/per
TestMetricFilter limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate test-metric-filter al secondo per regione account/per

Nome	Predefinita	Adatta e	Description
TestTransformer limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate test-transformer al secondo per regione account/per
UntagLogGroup limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate untag-log-group al secondo per regione account/per
UntagResource limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di chiamate untag-resource al secondo per regione account/per
registra i rilevatori di anomalie	Ogni regione supportata: 500	Sì	Il numero massimo di rilevatori di anomalie logaritmiche attivi

Gestione delle quote del servizio CloudWatch Logs

CloudWatch Logs si è integrato con Service Quotas, AWS un servizio che consente di visualizzare e gestire le quote da una posizione centrale. Per ulteriori informazioni, consulta Cos'è Service Quotas? nella Guida per l'utente di Service Quotas.

Service Quotas semplifica la ricerca del valore delle quote del servizio CloudWatch Logs.

Console di gestione AWS

Per visualizzare le quote del servizio CloudWatch Logs utilizzando la console

1. Apri la console Service Quotas all'indirizzo https://console.aws.amazon.com/servicequotas/.
2. Nel pannello di navigazione, scegliere servizi AWS .
3. Dall'elenco dei AWS servizi, cerca e seleziona Amazon CloudWatch Logs.

Nell'elenco Service Quotas, è possibile visualizzare il nome della quota di servizio, il valore applicato (se è disponibile), la quota predefinita AWS e se il valore della quota è adattabile.

4. Per visualizzare ulteriori informazioni su una quota di servizio, ad esempio la descrizione, scegli il nome della quota.
5. (Facoltativo) Per richiedere un aumento della quota, selezionare la quota che si desidera aumentare, selezionare Richiedi aumento quota, immettere o selezionare le informazioni richieste e selezionare Richiedi.

Per maggiori informazioni sulle quote di servizio utilizzando la console, consulta Guida per l'utente di Service Quotas. Per richiedere un aumento delle quote, consulta Richiesta di aumento delle quote nella Guida per l'utente di Service Quotas.

AWS CLI

Per visualizzare le quote del servizio CloudWatch Logs, utilizza AWS CLI

Eseguire il comando seguente per visualizzare le quote di CloudWatch registro predefinite.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code logs \
    --output table
```

Per utilizzare meglio le quote di servizio utilizzando il AWS CLI, vedere Service Quotas Command AWS CLI Reference. Per richiedere un aumento delle quote, consultare il comando `request-service-quota-increase` nella Documentazione di riferimento sui comandi AWS CLI.

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti in ciascuna versione della CloudWatch Logs User Guide, a partire da giugno 2018. Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile iscriversi a un feed RSS.

Modifica	Descrizione	Data
Nuovo comando di interrogazione delle stime per CloudWatch Logs Insights	È ora possibile utilizzare il `estimate` comando in CloudWatch Logs Insights per restituire i byte stimati che la query analizzerà nei gruppi di log e nell'intervallo di tempo selezionati, senza eseguire la query. Il `estimate` comando non comporta alcun costo per le query di CloudWatch Logs Insights. Per ulteriori informazioni, consulta Sintassi delle query nel linguaggio CloudWatch Logs Insights.	22 luglio 2026
Politica aggiornata AWSServiceRoleForLogDelivery sui ruoli collegati ai servizi	CloudWatch Log aggiunti `kms:GenerateDataKey` e `kms:Decrypt` autorizzazioni alla politica dei ruoli collegati al AWSServiceRoleForLogDelivery servizio. Queste autorizzazioni consentono l'invio dei log ai flussi di distribuzione Firehose che utilizzano la crittografia lato server con chiavi gestite dal cliente (). SSE-CMK Per ulteriori informazioni,	15 maggio 2026

	consulta CloudWatch Registra gli aggiornamenti dei ruoli collegati al servizio. AWS	
Query salvate con parametri per CloudWatch Logs Insights	Ora puoi creare modelli di query riutilizzabili con parametri denominati in CloudWatch Logs Insights. Definisci un singolo modello e passa valori diversi in fase di esecuzione invece di mantenere più query salvate quasi identiche. Per ulteriori informazioni, vedete Utilizzo delle interrogazioni salvate con parametri.	26 marzo 2026
Nuova politica gestita dai CloudWatch log CloudWatchLogsAPIKeyAccess	CloudWatch Logs ha rilasciato una nuova policy gestita `CloudWatchLogsAPIKeyAccess` che consente l'autenticazione con chiave API CloudWatch Logs e l'inserimento di log crittografati. Per informazioni, vedi Aggiornamenti CloudWatch dei log alle policy AWS gestite.	17 febbraio 2026
Politica aggiornata CloudWatchLogsReadOnlyAccess	Sono state aggiunte le autorizzazioni per le azioni di amministrazione dell'osservabilità per consentire l'accesso in sola lettura CloudWatchLogsReadOnlyAccess alle pipeline di telemetria e alle integrazioni delle tabelle S3.	2 dicembre 2025

Politica aggiornata CloudWatchLogsFullAccess	Sono state aggiunte le autorizzazioni per le azioni di amministrazione dell'osservabilità per consentire l'accesso in sola lettura CloudWatchLogsFullAccess alle pipeline di telemetria e alle integrazioni delle tabelle S3.	2 dicembre 2025
Nuova funzionalità di gestione dei log in CloudWatch Logs	CloudWatch consolida la gestione dei log in un unico servizio con funzionalità di governance integrate senza archiviare e mantenere più copie degli stessi dati su strumenti e archivi dati diversi. Per ulteriori informazioni, consulta Gestione dei https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/LogManagement.html log.	2 dicembre 2025
Nuova funzionalità di inserimento e normalizzazione dei dati in Logs CloudWatch	CloudWatch raccoglie automaticamente i log AWS distribuiti tra account e AWS regioni attraverso l'integrazione con le AWS organizzazioni e connettori predefiniti per fonti di terze parti. Per ulteriori informazioni, consulta Transform logs during ingestion.	2 dicembre 2025

Nuova funzionalità di analisi e interrogazione basata sulle fonti di dati in CloudWatch Logs Insights	CloudWatch Insights introduce delle sfaccettature per le interrogazioni. I facet sono utili per analizzare i log in quanto consentono di filtrare e raggruppare i dati in modo interattivo senza eseguire query. Un facet è un campo dei log (ad esempio ServiceName or StatusCode) che consente il filtraggio, l'aggregazione e l'analisi tra i gruppi di log. Per ulteriori informazioni, consulta Usare i facets per raggruppare ed esplorare i log.	2 dicembre 2025
Nuova integrazione con le tabelle S3	L'integrazione con S3 Tables CloudWatch consente di accedere ai dati di log inseriti CloudWatch utilizzando motori di analisi come Amazon Athena, Amazon Redshift e strumenti di terze parti che supportano la connessione agli Iceberg-compatible store Apache. Per ulteriori informazioni, consulta Access logs with S3 Tables Integration. https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/s3-tables-integration.html	2 dicembre 2025

CloudWatch Log ha aggiunto il supporto per la centraliz zazione dei Cross-account log tra regioni	CloudWatch Logs ha aggiunto il supporto per la centraliz zazione dei log Cross-account interregionali che consente a Logs di raccogliere CloudWatc h i dati di registro dagli account dei AWS Organizat ions membri in un account centrale per l'analisi. Per ulteriori informazioni, consulta Centralizzazione dei log Cross-account tra regioni.	17 settembre 2025
CloudWatch Politica di esempio aggiornata per il rilevamento delle anomalie nei registri	CloudWatch Logs ha aggiornat o la politica di esempio del KMS per ridurne l'ambito e migliorare la sicurezza aggiungendo condizioni per `aws:SourceAccount` e. `aws:SourceArn` Per ulteriori informazioni, consulta Crittografare un rilevatore di anomalie e i relativi risultati con KMS AWS	14 luglio 2025
CloudWatch Logs Insights aggiunge il supporto per i log di Amazon VPC Route Server	CloudWatch Logs Insights aggiunge il supporto per i log di Amazon VPC Route Server. Per ulteriori informazi oni, consulta Abilitare la registrazione dai AWS servizi. La nuova fonte di log per Amazon VPC Route Server `EVENT_LOGS` è documentata in PutDeliverySource	12 giugno 2025

CloudWatch Logs Insights aggiunge il supporto per i log AWS PCS	CloudWatch Logs Insights aggiunge il supporto per AWS PCS i log. Per ulteriori informazioni, consulta Abilitare la registrazione dai AWS servizi. Le nuove fonti di log per AWS PCS `PCS_SCHEDULER_LOGS` e `PCS_JOBCOMP_LOGS` sono documentate in PutDeliverySource	12 giugno 2025
CloudWatch Logs Insights aggiunge il supporto per i AWS Entity Resolution log	CloudWatch Logs Insights aggiunge il supporto per AWS Entity Resolution i log. Per ulteriori informazioni, consulta Abilitare la registrazione dai AWS servizi. La nuova fonte di log per AWS Entity Resolution, `WORKFLOW_LOGS` è documentata in PutDeliverySource	22 maggio 2025

CloudWatch Registra gli aggiornamenti delle policy gestiti per supportare i riepiloghi in linguaggio naturale	Le autorizzazioni per `cloudwatch:GenerateQueryResultsSummary` sono state aggiunte CloudWatchLogsFullAccess e CloudWatchLogsReadOnlyAccess consentono la generazione di un riepilogo in linguaggio naturale dei risultati della query. Per vedere il contenuto delle policy, consulta CloudWatchLogsFullAccess e CloudWatchLogsReadOnlyAccess nella AWS Managed Policy Reference Guide.	20 maggio 2025
CloudWatch Logs Insights aggiunge il supporto per la generazione di riepiloghi in linguaggio naturale dai risultati delle query di CloudWatch Logs Insights	È stato aggiunto il supporto per i riepiloghi in linguaggio naturale in CloudWatch Logs Insights. Questa funzione genera riepiloghi leggibili dall'uomo dei risultati delle query, attualmente disponibili negli Stati Uniti orientali (Virginia settentrionale). Per ulteriori informazioni, consulta Generare riepiloghi in linguaggio naturale dai risultati delle query di Logs Insights. CloudWatch	20 maggio 2025

CloudWatchOpenSearchDashboardAccesse CloudWatchOpenSearchDashboardsFullAccess sono nuove politiche IAM	CloudWatch Logs ha aggiunto due nuove policy IAM CloudWatchOpenSearchDashboardsFullAccess e CloudWatchOpenSearchDashboardAccess. CloudWatchOpenSearchDashboardsFullAccessconcede il permesso di creare e gestire integrazioni con OpenSearch il servizio. CloudWatchOpenSearchDashboardAccessconcede l'accesso per visualizzare i dashboard dei log di vendita creati in queste integrazioni.	1 dicembre 2024
CloudWatchLogsFullAccesspolitica aggiornata	CloudWatch I log hanno aggiunto le autorizzazioni per Amazon OpenSearch Service e IAM alla CloudWatchLogsFullAccess policy, per abilitare l'integrazione CloudWatch dei log con il OpenSearch servizio per alcune funzionalità.	1 dicembre 2024
CloudWatch Logs Insights aggiunge nuovi tipi di struttura alla sintassi della query	CloudWatch Logs Insights aggiunge il `unnest` comando e due funzioni JSON, che consentono di utilizzare stringhe JSON come mappe ed elenchi. Per ulteriori informazioni, consulta Tipi di struttura.	21 novembre 2024

CloudWatch Logs supporta la trasformazione dei log durante l'inserimento dei log	È possibile creare trasformatori di log in grado di modificare gli eventi dei log al momento dell'inserimento, aiutandovi a normalizzare i log in diversi formati e provenienti da diverse fonti in formati coerenti e ricchi di contesto. Per ulteriori informazioni, consulta Trasformare i log durante l'inserimento. https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CloudWatch-Logs-Transformation.html	20 novembre 2024
CloudWatch Logs Insights aggiunge l'indicizzazione dei campi	CloudWatch Logs Insights ha aggiunto il supporto per l'indicizzazione dei log sui campi. Quando poi si utilizza un indice di campo in una query CloudWatch Logs Insights, la query tenta di ignorare l'elaborazione di eventi di log che notoriamente non includono il campo indicizzato. Per ulteriori informazioni, consulta Creare indici di campo per migliorare le prestazioni delle query e ridurre il volume di scansione.	20 novembre 2024

CloudWatch Il supporto di Logs Insights per la generazione di query in linguaggio naturale è generalmente disponibile	CloudWatch Logs Insights supporta il linguaggio naturale per generare e aggiornare le query. Per ulteriori informazioni, consulta Usa il linguaggio naturale per generare e aggiornare le query di CloudWatch Logs Insights.	20 giugno 2024
CloudWatchLogsReadOnlyAccesspolitica aggiornata	CloudWatch I log hanno aggiunto l'`cloudwatch:GenerateQuery` autorizzazione a CloudWatchLogsReadOnlyAccess, in modo che gli utenti con questa policy possano generare una stringa di query CloudWatch Logs Insights da un prompt in linguaggio naturale.	26 novembre 2023
CloudWatchLogsFullAccesspolitica aggiornata	CloudWatch I log hanno aggiunto l'`cloudwatch:GenerateQuery` autorizzazione a CloudWatchLogsFullAccess, in modo che gli utenti con questa policy possano generare una stringa di query CloudWatch Logs Insights da un prompt in linguaggio naturale.	26 novembre 2023

CloudWatch Logs aggiunge l'analisi del modello di registro	CloudWatch Logs ora analizza i pattern negli eventi di registro ogni volta che si esegue una query CloudWatch Logs Insights. Per ulteriori informazioni, consulta Analisi dei https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CWL_AnalyzeLogData_Patterns.html modelli.	26 novembre 2023
CloudWatch Logs aggiunge il rilevamento delle anomalie nei log	È possibile creare un rilevatore di anomalie nei log per un gruppo di log. Il rilevatore di anomalie analizza gli eventi di registro inseriti nel gruppo di log e trova le anomalie nei dati di registro. Per ulteriori informazioni, consulta Rilevamento delle anomalie di log.	26 novembre 2023
CloudWatch Logs aggiunge la funzionalità di confronto	Ora puoi utilizzare CloudWatch Logs Insights per confrontare le modifiche negli eventi di registro nel tempo. Per ulteriori informazioni, consulta Confronta (diff) con intervalli di tempo precedenti.	26 novembre 2023

CloudWatch Logs aggiunge una nuova classe di log	CloudWatch Logs supporta due classi di gruppi di log, in modo da offrire un'opzion e conveniente per i log a cui si accede raramente e un'opzione completa per i log che richiedono monitoraggio in tempo reale o altre funzional ità. Per ulteriori informazioni, consulta Classi di log.	26 novembre 2023
CloudWatch Logs Insights supporta la generazione di query in linguaggio naturale	CloudWatch Logs Insights supporta il linguaggio naturale per generare e aggiornare le query. Per ulteriori informazi oni, consulta Usa il linguaggi o naturale per generare e aggiornare le query di CloudWatch Logs Insights.	26 novembre 2023
CloudWatch Logs aggiunge il supporto per la sintassi dei modelli di filtro per le espressio ni regolari per Live Tail	Ora puoi personalizzare ulteriormente le operazioni di ricerca e corrispondenza per soddisfare le tue esigenze con espressioni regolari flessibili all'interno dei modelli di filtro Live Tail. Per ulteriori informazi oni, consulta la sintassi del modello di filtro nella Amazon CloudWatch Logs User Guide.	13 novembre 2023

CloudWatch Logs aggiunge il supporto della sintassi dei pattern di filtro con espressioni regolari per filtri metrici, filtri di sottoscrizione ed eventi di log dei filtri	Ora puoi personalizzare ulteriormente le operazioni di ricerca e corrispondenza per soddisfare le tue esigenze con espressioni regolari flessibili all'interno di modelli di filtro. Per ulteriori informazioni, consulta la sintassi dei modelli di filtro nella Amazon CloudWatch Logs User Guide.	05 settembre 2023
CloudWatch Logs Insights aggiunge un comando pattern	È ora possibile utilizzare pattern nelle query di CloudWatch Logs Insights per raggruppare automaticamente i dati di registro in modelli. Un pattern è una struttura di testo condivisa che ricorre tra i campi dei log. Per ulteriori informazioni, consulta pattern nella Amazon CloudWatch Logs User Guide.	17 luglio 2023
CloudWatch Logs Insights aggiunge un comando di dedup	Ora puoi utilizzare dedup nelle tue query CloudWatch Logs Insights per rimuovere i risultati duplicati in base a valori specifici nei campi specificati. Per ulteriori informazioni, consulta dedup nella Amazon Logs User Guide. CloudWatch	20 giugno 2023

Account-level politiche di protezione dei dati	Ora puoi impostare le policy di protezione dei dati a livello di account. Queste policy a livello di account possono verificar e e mascherare le informazi oni riservate dei log eventi in tutti i gruppi di log dell'acco unt. Per ulteriori informazioni, consulta Aiutare a protegger e i dati di log sensibili con il mascheramento nella Amazon CloudWatch Logs User Guide.	8 giugno 2023
Funzionalità Live Tail aggiunta	CloudWatch I log hanno aggiunto la funzionalità Live Tail, in modo da poter scansionare i log man mano che vengono inseriti per facilitare la risoluzione dei problemi. Facoltativamente, puoi filtrare il flusso di log eventi visualizzato in base a termini specificati ed evidenzia re anche i log eventi con termini specifici. Per ulteriori informazioni, consulta Use live tail to view logs in near real time.	6 giugno 2023

CloudWatchLogsRead OnlyAccesspolitica aggiornata	CloudWatch I registri hanno aggiunto autorizzazioni a. CloudWatchLogsRead OnlyAccess Le `logs:StopLiveTail` autorizzazioni `logs:StartLiveTail` and sono state aggiunte in modo che gli utenti con questa policy possano utilizzare la console per avviare e interrompere le sessioni live tail di CloudWatch Logs. Per ulteriori informazioni, consulta Use live tail to view logs in near real time.	6 giugno 2023
CloudWatch È stato rilasciato Logs Insights	Puoi utilizzare CloudWatch Logs Insights per cercare e analizzare in modo interattivo i tuoi dati di registro. Per ulteriori informazioni, consulta la sezione Analyze Log Data with CloudWatch Logs Insights nella Amazon CloudWatch Logs User Guide	27 novembre 2018
Supporto per gli endpoint Amazon VPC	Ora puoi stabilire una connessione privata tra il tuo VPC e CloudWatch Logs. Per ulteriori informazioni, consulta Using CloudWatch Logs with Interface VPC Endpoints nella Amazon CloudWatch Logs User Guide.	28 giugno 2018

La tabella seguente descrive le modifiche importanti alla Amazon CloudWatch Logs User's Guide.

Modifica	Description	Data di rilascio
Endpoint VPC di interfaccia	In alcune regioni, puoi utilizzare un endpoint VPC di interfaccia per impedire al traffico tra Amazon VPC e CloudWatch Logs di lasciare la rete Amazon. Per ulteriori informazioni, consulta Utilizzo dei CloudWatch log con endpoint VPC di interfaccia.	07 marzo 2018
Log di query DNS di Route 53	Puoi utilizzare CloudWatch Logs per archiviare i log relativi alle richieste DNS ricevute da Route 53. Per ulteriori informazioni, consulta Che cos'è Amazon CloudWatch Logs? o Registrazione delle query DNS nella Guida per gli sviluppatori di Amazon Route 53.	07 settembre 2017
Assegnazione di tag ai gruppi di log	Puoi utilizzare i tag per categorizzare i gruppi di log. Per ulteriori informazioni, consulta Contrassegna i gruppi di log in Amazon CloudWatch Logs.	13 dicembre 2016
Miglioramenti della console	Puoi spostarti dai grafici di parametri ai gruppo di log associati. Per ulteriori informazioni, consulta Cambiare da parametri a log.	7 Novembre 2016
Miglioramenti della fruibilità della console	Migliorata per rendere più facili la ricerche, l'applicazione di filtri e la risoluzioni dei problemi. Ad esempio, è ora possibile filtrare i dati di log per un intervallo di data e orari. Per ulteriori informazioni, consulta Visualizza i dati di registro inviati ai registri CloudWatch.	29 agosto 2016
È stato aggiunto AWS CloudTrail il supporto per Amazon CloudWatch Logs e le nuove	È stato aggiunto AWS CloudTrail il supporto per i CloudWatch log. Per ulteriori informazioni, consulta Logging CloudWatch Logs, API e operazioni della console in AWS CloudTrail.	10 marzo 2016

Modifica	Description	Data di rilascio
CloudWatch metriche di Logs		
È stato aggiunto il supporto per l'esportazione CloudWatch dei log in Amazon S3	È stato aggiunto il supporto per l'esportazione dei dati di CloudWatch log in Amazon S3. Per ulteriori informazioni, consulta Esportazione di dati di log in Amazon S3.	7 dicembre 2015
È stato aggiunto il supporto per gli eventi AWS CloudTrail registrati in Amazon Logs CloudWatch	Puoi creare allarmi CloudWatch e ricevere notifiche relative a particolari attività dell'API acquisite da CloudTrail e utilizzare la notifica per eseguire la risoluzione dei problemi.	10 novembre 2014
È stato aggiunto il supporto per Amazon Logs CloudWatch	Puoi utilizzare Amazon CloudWatch Logs per monitorare, archiviare e accedere al sistema, all'applicazione e ai file di registro personalizzati da istanze Amazon Elastic Compute Cloud (Amazon EC2) o da altre fonti. Puoi quindi recuperare i dati di log associati da CloudWatch Logs utilizzando la CloudWatch console Amazon, i comandi CloudWatch Logs in AWS CLI, o il Logs SDK. CloudWatch Per ulteriori informazioni, consulta Che cos'è Amazon CloudWatch Logs?.	10 luglio 2014

AWS Glossario

Per la AWS terminologia più recente, consultate il AWS glossario nella sezione Reference.Glossario AWS

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.