aws

Guida per l'utente

CloudWatch Registri Amazon

CloudWatch Registri Amazon: Guida per l'utente

Table of Contents

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 della 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 di API attività particolari rilevate da 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.
- Archivia i dati di registro: puoi utilizzare CloudWatch Logs per archiviare i dati di registro in uno spazio di archiviazione 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 DNS interrogazioni su Route 53: è possibile utilizzare CloudWatch Logs per registrare le informazioni sulle query ricevute da Route 53. DNS Per ulteriori informazioni, consulta la sezione Logging DNS Queries nella Amazon Route 53 Developer Guide.

Servizi correlati AWS

I seguenti servizi vengono utilizzati insieme ai registri: CloudWatch

- AWS CloudTrailè un servizio Web che consente di monitorare le chiamate effettuate ai CloudWatch registri API dell'account, incluse le chiamate effettuate da AWS Management Console, AWS Command Line Interface (AWS CLI) e altri servizi. Quando CloudTrail la registrazione è attivata, CloudTrail acquisisce le API chiamate 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, vedereRegistrazione dei CloudWatch log API e delle operazioni della console AWS CloudTrail.
- AWS Identity and Access Management (IAM) è un servizio web che consente di controllare in modo sicuro l'accesso alle AWS risorse per gli 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 VPC flusso di Amazon 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 avere la codifica -8. UTF

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.

Fatturazione e costi di File di log Amazon CloudWatch

Per ulteriori informazioni su come analizzare i costi e l'utilizzo di File di log CloudWatch e CloudWatch e per le best practice su come ridurre i costi, consulta la sezione CloudWatch billing and cost.

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

Quando effettui la registrazione ad AWS, puoi iniziare a utilizzare CloudWatch Logs gratuitamente tramite il piano gratuito AWS.

Si applicano le tariffe standard per i log archiviati da altri servizi che utilizzano File di log CloudWatch, ad esempio i flussi di log 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.)	
OpenSearch SQLUsare OpenSearch PPL o eseguire interrogazioni in CloudWatch Logs Insights;	Sì ✓	No	
CloudWatch I campi rilevati da Logs Insights	Sì ✓	No	
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	

Funzionalità	Standard	Accesso poco frequente	
Filtri di abbonamento	Sì ✓	No	
Esportazione in Amazon S3	Sì ✓	No	
GetLogEvents e FilterLogEvents APIoperazioni	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	
Protezione dei dati sensibili con mascheramento	Sì ✓	No	
Formato di metriche incorporato	Sì ✓	No	

Guida introduttiva ai CloudWatch registri

Per raccogliere i log dalle tue EC2 istanze Amazon e dai server locali in CloudWatch Logs, usa 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

Per utilizzare Amazon CloudWatch Logs è necessario un AWS account. Il tuo AWS account consente di utilizzare servizi (ad esempio AmazonEC2) per generare log che è possibile visualizzare nella CloudWatch console, un'interfaccia basata sul Web. Inoltre, è possibile installare e configurare AWS Command Line Interface (AWS CLI).

Iscriviti per un Account AWS

Se non hai un Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, un Utente root dell'account AWSviene creato. L'utente root ha accesso a tutti Servizi AWS e le risorse presenti nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato per un Account AWS, proteggi il tuo Utente root dell'account AWS, abilita AWS IAM Identity Center e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi a AWS Management Consolecome proprietario dell'account selezionando Utente root e inserendo il Account AWS indirizzo email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso tramite utente root, consulta Accesso come utente root in Accedi ad AWS Guida per l'utente.
2. Attiva l'autenticazione a più fattori (MFA) per il tuo utente root.

   Per istruzioni, consulta Abilitare un MFA dispositivo virtuale per il Account AWS utente root (console) nella Guida per l'IAMutente.

Crea un utente con accesso amministrativo

1. Abilita IAM Identity Center.

Per istruzioni, vedi Abilitazione AWS IAM Identity Center nella AWS IAM Identity Center Guida per l'utente.

2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, vedi Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella AWS IAM Identity Center Guida per l'utente.

Accesso come utente amministratore

- Per accedere con il tuo utente IAM Identity Center, utilizza l'accesso URL che ti è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso tramite un utente di IAM Identity Center, vedi Accesso a AWS accedere al portale in Accedi ad AWS Guida per l'utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Per istruzioni, vedere Creare un set di autorizzazioni nella AWS IAM Identity Center Guida per l'utente.

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella AWS IAM Identity Center Guida per l'utente.

Impostazione dell'interfaccia a riga di comando

Puoi utilizzare il plugin AWS CLI per eseguire operazioni di registro CloudWatch .

Per informazioni su come installare e configurare AWS CLI, vedi Getting Up Up with the AWS Interfaccia a riga di comando in AWS Command Line Interface Guida per l'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 CloudWatch Logs, consulta Collect Metrics and Logs from Amazon Instances and On-Premises Servers with the CloudWatch Agent nella EC2 Amazon User Guide. CloudWatch Per installare, configurare e avviare l'agente, completa la procedura di seguito riportata. Se non utilizzi l'agente per raccogliere anche i CloudWatch parametri, puoi ignorare tutte le sezioni che fanno riferimento ai 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 i log che le metriche dalle istanze e dai EC2 server locali. Il vecchio agente che utilizza solo i log è obsoleto e non è più supportato.
> Per informazioni sulla migrazione dal precedente 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 EC2 istanze Amazon 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 Agent CloudWatch nella Amazon User Guide. CloudWatch

Indice

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

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

> **Important**
>
> Il vecchio logs agent è obsoleto. CloudWatch include un agente unificato in grado di raccogliere sia i log che le metriche dalle istanze e dai server locali. EC2 Per ulteriori informazioni, consulta Guida introduttiva ai CloudWatch registri.

Per informazioni sulla migrazione dal precedente 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 utilizzaIMDSv2, è necessario utilizzare l'agente unificato più recente anziché il vecchio 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 EC2 istanze 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 utilizzaIMDSv2, 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 Linux in esecuzione EC2

È possibile utilizzare il programma di installazione dell'agente CloudWatch Logs su un'EC2istanza esistente per installare e configurare l'agente CloudWatch Logs. 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, è possibile pubblicare i dati di registro anche utilizzando Logs o CloudWatch Logs. AWS CLI SDK CloudWatch API AWS CLI È più adatto per la pubblicazione di dati sulla riga di comando o tramite script. The CloudWatch Logs SDK è la soluzione ideale per pubblicare i dati di registro direttamente dalle applicazioni o per creare un'applicazione di pubblicazione dei log personalizzata.

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

L'agente CloudWatch Logs supporta IAM ruoli e utenti. Se alla tua istanza è già associato un IAM ruolo, assicurati di includere la IAM policy riportata di seguito. Se non hai già assegnato un IAM ruolo alla tua istanza, puoi utilizzare IAM le tue credenziali per i passaggi successivi oppure puoi assegnare un IAM ruolo a quell'istanza. Per ulteriori informazioni, consulta Associare un IAM ruolo a un'istanza.

Per configurare il IAM ruolo o l'utente per i registri CloudWatch

1. Apri la IAM console 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. Scegli la JSONscheda e digita il seguente documento di JSON policy.

```
{
  "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 Summary (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'EC2istanza Amazon 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 da Amazon Linux AMI 2014.09, l'agente CloudWatch Logs è disponibile come RPM installazione 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 programma di installazione di CloudWatch Logs, l'istanza riceve regolarmente aggiornamenti e patch dei pacchetti senza dover reinstallare manualmente l'agente Logs. RPM AWS CloudWatch

> **Warning**
>
> Non aggiornate l'agente CloudWatch Logs utilizzando il metodo di RPM installazione 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. 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 .

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

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

1. Connect alla tua EC2 istanza. 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 corrente e `awslogs-agent-setup` sulla cronologia delle versioni di, consulta CHANGELOG .txt.

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

Elemento	Descrizione
AWS ID della chiave di accesso	Premi Invio se utilizzi un IAM ruolo. Altrimenti, inserisci l'ID della tua chiave di AWS accesso.

Elemento	Descrizione
AWS chiave di accesso segreta	Premi Invio se usi un IAM ruolo. 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. 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: installa e configura l'agente CloudWatch Logs su un'istanza EC2 Linux all'avvio

> **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 EC2 nuove istanze per impostazione predefinita. Per ulteriori informazioni sull'agente unificato, vedere 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 EC2 utente di Amazon, una funzionalità di Amazon EC2 che consente il trasferimento di informazioni parametriche all'istanza all'avvio, per installare e configurare l'agente CloudWatch Logs su quell'istanza. Per trasmettere le informazioni di installazione e configurazione dell'agente CloudWatch Logs ad AmazonEC2, 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
```

Per configurare il ruolo IAM

1. Apri la console all'IAMindirizzo. 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 politiche personalizzate, consulta IAMPolicies 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:

```
{
    "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. Scegliere Create Policy (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 Seleziona il tipo di ruolo, scegli 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 IAM politica che hai creato, quindi scegli Passaggio successivo.
12. Selezionare Create Role (Crea ruolo).

    Per ulteriori informazioni su utenti e criteri, vedere IAMUtenti e gruppi e Gestione delle IAM politiche nella Guida per l'IAMutente.

Per avviare una nuova istanza e abilitare CloudWatch i registri

1. Apri la EC2 console Amazon 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 Fase 1: Scegli una Amazon Machine Image (AMI), seleziona il tipo di istanza Linux da avviare, quindi nella pagina Passaggio 2: Scegli un tipo di istanza, scegli Avanti: Configura i dettagli dell'istanza.

   Assicurati che cloud-init sia incluso nella tua Amazon Machine Image (). AMI Amazon Linux e AMIs per Ubuntu includono RHEL già cloud-initAMIs, ma CentOS e altri AMIs potrebbero non esserlo. Marketplace AWS

4. Nella pagina Fase 3: Configurazione dei dettagli dell'istanza, per IAMruolo, seleziona il IAM ruolo che hai 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: EC2 consenti alle istanze Amazon che eseguono Windows Server 2016 di inviare log a Logs utilizzando l' CloudWatch agente Logs CloudWatch

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

Consenti alle EC2 istanze Amazon che eseguono Windows Server 2016 di inviare log a Logs utilizzando il CloudWatch vecchio 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 per JSON CloudWatch
- Creare un IAM ruolo 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 per JSON CloudWatch

È possibile determinare a quali registri 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
- Passaggio 2: configura le impostazioni per CloudWatch
- Fase 3: configurazione dei dati da inviare
- Fase 4: configurazione del controllo di flusso
- Fase 5: JSON Salvare i contenuti

Fase 1: Abilitare i log CloudWatch

Nella parte superiore del JSON file, cambia «false» in «true» per`IsEnabled`:

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

Passaggio 2: configura 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 JSON file, individua la `CloudWatchLogs` sezione.

```
{
    "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`. Le credenziali vengono configurate utilizzando un IAM ruolo.
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 della CloudWatch console.
5. Per `LogStream` digita il flusso di log di destinazione. Questo nome appare 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 a CloudWatch Logs.

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

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

```
{
    "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 JSON file, individua la `SecurityEventLog` sezione.

```
{
    "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 JSON file, individuate la `SystemEventLog` sezione.

```
{
    "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 JSON file, aggiungi 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(Event Tracing per Windows) fornisce un meccanismo di registrazione efficiente e dettagliato su cui le applicazioni possono scrivere i log. Ciascuno di essi ETW è controllato da un gestore di sessione che può avviare e interrompere la sessione di registrazione. Ogni sessione dispone di un fornitore e di uno o più consumatori.

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

```
{
    "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 JSON file, individua 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, vedere l'argomento Stringhe di formato di data e ora personalizzate suMSDN.

> **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. Per`Encoding`, digitate la codifica del file da utilizzare (ad esempio, UTF -8). Per un elenco dei valori supportati, consultate l'argomento Encoding Class 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, consultate l'argomento FileSystemWatcherFilter Proprietà suMSDN.
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 in merito, vedere la `Language tag` colonna della tabella nell'argomento Comportamento del prodotto suMSDN.

> **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 IIS registro hanno intestazioni praticamente identiche. È possibile immettere **5** per leggere le prime tre righe dell'intestazione del file di log per identificarlo. Nei file di IIS registro, la terza riga è l'indicatore di data e ora, ma non è sempre garantito che il timestamp sia diverso tra i file di registro. 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 IIS registro ai registri CloudWatch

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

```
{
    "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. Ad esempio`LogDirectoryPath`, digita la cartella in cui sono archiviati IIS i log di un singolo sito (ad esempio,`C:\inetpub\logs\LogFiles\W3SVC`*n*).

   > **Note**
   >
   > È supportato solo il formato di log W3C. IISNCSA, e i formati personalizzati non sono supportati.

3. In `TimestampFormat`, digita il formato timestamp da utilizzare. Per ulteriori informazioni sui valori supportati, vedere l'argomento Stringhe di formato di data e ora personalizzate suMSDN.
4. Per`Encoding`, digitate la codifica del file da utilizzare (ad esempio, UTF -8). Per ulteriori informazioni sui valori supportati, consultate l'argomento Encoding Class 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, consultate l'argomento FileSystemWatcherFilter Proprietà suMSDN.
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 sui valori supportati, consultate la `Language tag` colonna della tabella nell'argomento Comportamento del prodotto suMSDN.

   > **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 IIS registro hanno intestazioni praticamente identiche. È possibile inserire **5** per leggere le prime cinque righe dell'intestazione del file di log per identificarlo. Nei

file di IIS registro, la terza riga è l'indicatore di data e ora, ma non è sempre garantito che il timestamp sia diverso tra i file di registro. 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 e il ETW registro di sistema a CloudWatch Logs, aggiungili `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` alla sezione. `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: JSON Salvare i contenuti

Ora hai finito di modificare il JSON file. 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 IAM ruolo per Systems Manager

Un IAM ruolo, ad esempio le credenziali, è necessario quando si utilizza Systems Manager Run Command. 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 associare un IAM ruolo 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 Internet Gateways nella Amazon VPC User Guide.

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. L'SSMagente sull'istanza elabora il comando e configura l'istanza come specificato.

Per configurare l'integrazione con CloudWatch i registri utilizzando Run Command

1. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
2. Apri la SSM console 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. Per le proprietà, copia e incolla il JSON contenuto creato nelle attività precedenti.
9. Completare i restanti campi opzionali e scegliere Esegui.

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

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.

Quick Start: EC2 consenti alle istanze Amazon 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 e dai server locali. EC2 Ti consigliamo di utilizzare il nuovo agente unificato. CloudWatch Per ulteriori informazioni, consulta Guida introduttiva ai CloudWatch registri. Il resto di questa sezione spiega l'uso del vecchio agente CloudWatch Logs.

Consenti alle EC2 istanze Amazon 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

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

Configura il JSON file per CloudWatch

È possibile determinare a quali registri inviare i dati CloudWatch specificando le proprie scelte nel JSON 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
- Passaggio 2: configura le impostazioni per CloudWatch
- Fase 3: configurazione dei dati da inviare
- Fase 4: configurazione del controllo di flusso

Fase 1: Abilitare i log CloudWatch

Nella parte superiore del JSON file, cambia «false» in «true» per`IsEnabled`:

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

Passaggio 2: configura 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 JSON file, individua la `CloudWatchLogs` sezione.

```
{
    "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`. Le credenziali vengono configurate utilizzando un IAM ruolo.
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 della CloudWatch console.
5. Per `LogStream` digita il flusso di log di destinazione. Questo nome appare 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 a CloudWatch Logs.

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

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

```
{
    "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 JSON file, individua la `SecurityEventLog` sezione.

```
{
    "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 JSON file, individuate la `SystemEventLog` sezione.

```
{
    "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 JSON file, aggiungi 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(Event Tracing per Windows) fornisce un meccanismo di registrazione efficiente e dettagliato su cui le applicazioni possono scrivere i log. Ciascuno di essi ETW è controllato da un gestore di sessione che può avviare e interrompere la sessione di registrazione. Ogni sessione dispone di un fornitore e di uno o più consumatori.

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

```
{
    "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 JSON file, individua 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, vedere l'argomento Stringhe di formato di data e ora personalizzate suMSDN.

> **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. Per`Encoding`, digitate la codifica del file da utilizzare (ad esempio, UTF -8). Per ulteriori informazioni sui valori supportati, consultate l'argomento Encoding Class 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, consultate l'argomento FileSystemWatcherFilter Proprietà suMSDN.
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 sui valori supportati, consultate la `Language tag` colonna della tabella nell'argomento Comportamento del prodotto suMSDN.

   > **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 IIS registro hanno intestazioni praticamente identiche. È possibile immettere **5** per leggere le prime tre righe dell'intestazione del file di log per identificarlo. Nei file di IIS registro, la terza riga è l'indicatore di data e ora, ma non è sempre garantito che il timestamp sia diverso tra i file di registro. 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 IIS registro ai registri CloudWatch

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

```
{
    "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. Ad esempioLogDirectoryPath, digita la cartella in cui sono archiviati IIS i log di un singolo sito (ad esempio,C:\inetpub\logs\LogFiles\W3SVC*n*).

> **Note**
>
> È supportato solo il formato di log W3C. IISNCSA, e i formati personalizzati non sono supportati.

3. In TimestampFormat, digita il formato timestamp da utilizzare. Per ulteriori informazioni sui valori supportati, vedere l'argomento Stringhe di formato di data e ora personalizzate suMSDN.
4. PerEncoding, digitate la codifica del file da utilizzare (ad esempio, UTF -8). Per ulteriori informazioni sui valori supportati, consultate l'argomento Encoding Class 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, consultate l'argomento FileSystemWatcherFilter Proprietà suMSDN.
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 sui valori supportati, consultate la Language tag colonna della tabella nell'argomento Comportamento del prodotto suMSDN.

> **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 IIS registro hanno intestazioni praticamente identiche. È possibile inserire **5** per leggere le prime cinque righe dell'intestazione del file di log per identificarlo. Nei file di IIS registro, la terza riga è l'indicatore di data e ora, ma non è sempre garantito che il timestamp sia diverso tra i file di registro. 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 e il ETW registro di sistema a CloudWatch Logs, aggiungili `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` alla sezione. `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`).

Ora hai finito di modificare il JSON file. Lo utilizzerai in una fase successiva.

Avvia l'agente

Per consentire a un'EC2istanza Amazon che esegue Windows Server 2012 o Windows Server 2008 di inviare log a CloudWatch Logs, usa il EC2Config servizio (. `EC2Config.exe`) La tua istanza deve avere la EC2Config versione 4.0 o successiva e puoi usare questa procedura. Per ulteriori informazioni sull'utilizzo di una versione precedente diEC2Config, consulta Use EC2Config 3.x o precedente per configurare CloudWatch nella Amazon EC2 User Guide

Per configurare CloudWatch utilizzando 4.x EC2Config

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

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

Dopo il riavvio, l'SSMagente rileva il file di configurazione e configura l'istanza per l'integrazione. CloudWatch Se si modificano i parametri e le impostazioni nel file di configurazione locale, è necessario riavviare l'SSMagente per rilevare le modifiche. Per disabilitare CloudWatch l'integrazione sull'istanza, `IsEnabled` apporta `false` e salva le modifiche nel file di configurazione.

Avvio rapido: installa l'agente CloudWatch Logs utilizzando AWS OpsWorks and Chef

Puoi installare l'agente CloudWatch Logs e creare flussi di log utilizzando and Chef, uno AWS OpsWorks strumento di automazione dei sistemi e dell'infrastruttura cloud di terze parti. Chef utilizza le "ricette", che puoi scrivere per installare e configurare software sul tuo computer, e i "libri di ricette", che sono raccolte di ricette, per eseguire attività di configurazione e distribuzione delle policy. Per ulteriori informazioni, consulta Chef.

Gli esempi di ricette Chef riportati di seguito mostrano come monitorare un file di registro per ogni EC2 istanza. Le ricette utilizzano il nome dello stack come gruppo di log e il nome host dell'istanza come nome del flusso di log. Per monitorare più file di log, è necessario estendere le ricette per creare più gruppi di log e flussi di log.

Fase 1: creazione di ricette personalizzate

Crea un archivio per archiviare le tue ricette. AWS OpsWorks supporta Git e Subversion oppure puoi archiviare un archivio in Amazon S3. La struttura del repository dei libri di ricette è descritta nella sezione Repository dei libri di ricette nella AWS OpsWorks Guida per l'utente. Gli esempi di seguito presuppongono che il libro di ricette sia denominato `logs`. La ricetta install.rb installa l'agente Logs. CloudWatch Puoi anche scaricare l'esempio del libro di cucina (-Cookbooks.zip). CloudWatchLogs

Crea un file denominato metadata.rb che contiene il seguente codice:

```
#metadata.rb

name             'logs'
version          '0.0.1'
```

Crea il file di configurazione CloudWatch dei registri:

```
#config.rb

template "/tmp/cwlogs.cfg" do
  cookbook "logs"
  source "cwlogs.cfg.erb"
  owner "root"
  group "root"
  mode 0644
end
```

Scarica e installa l'agente CloudWatch Logs:

```
# install.rb

directory "/opt/aws/cloudwatch" do
  recursive true
end
```

```
remote_file "/opt/aws/cloudwatch/awslogs-agent-setup.py" do
  source "https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py"
  mode "0755"
end

  execute "Install CloudWatch Logs agent" do
  command "/opt/aws/cloudwatch/awslogs-agent-setup.py -n -r region -c /tmp/cwlogs.cfg"
  not_if { system "pgrep -f aws-logs-agent-setup" }
end
```

> **Note**
>
> Nell'esempio precedente, sostituisci *region* con uno dei seguenti: us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, eu-central-1 eu-west-1 o sa-east-1.

Se l'installazione dell'agente ha esito negativo, verifica che il pacchetto `python-dev` sia installato. In caso contrario, utilizza il comando seguente, quindi riprova a eseguire l'installazione dell'agente:

```
sudo apt-get -y install python-dev
```

Questa ricetta utilizza un file del modello cwlogs.cfg.erb che puoi modificare per specificare diversi attributi, ad esempio quali file registrare. Per ulteriori informazioni su questi attributi, consulta CloudWatch Registra il riferimento dell'agente.

```
[general]
# Path to the AWSLogs agent's state file. Agent uses this file to maintain
# client side state across its executions.
state_file = /var/awslogs/state/agent-state


## Each log file is defined in its own section. The section name doesn't
## matter as long as its unique within this file.
#
#[kern.log]
#
## Path of log file for the agent to monitor and upload.
#
#file = /var/log/kern.log
#
```

```
## Name of the destination log group.
#
#log_group_name = kern.log
#
## Name of the destination log stream.
#
#log_stream_name = {instance_id}
#
## Format specifier for timestamp parsing.
#
#datetime_format = %b %d %H:%M:%S
#
#

[<%= node[:opsworks][:stack][:name] %>]
datetime_format = [%Y-%m-%d %H:%M:%S]
log_group_name = <%= node[:opsworks][:stack][:name].gsub(' ','_') %>
file = <%= node[:cwlogs][:logfile] %>
log_stream_name = <%= node[:opsworks][:instance][:hostname] %>
```

Il modello ottiene il nome dello stack e il nome host facendo riferimento agli attributi corrispondenti nella configurazione e nella distribuzione dello JSON stack. L'attributo che specifica il file da registrare è definito nel file di attributi default.rb del libro di cucina cwlogs (.rb). logs/attributes/default

```
default[:cwlogs][:logfile] = '/var/log/aws/opsworks/opsworks-agent.statistics.log'
```

AWS OpsWorks Fase 2: Creare una pila

1. Apri la AWS OpsWorks console all'indirizzo https://console.aws.amazon.com/opsworks/.
2. Nella OpsWorks Dashboard, scegli Aggiungi stack per creare uno AWS OpsWorks stack.
3. Nella schermata Add stack (Aggiungi stack), scegli Chef 11 stack (Stack Chef 11).
4. In Stack name (Nome stack), inserisci un nome.
5. In Use custom Chef Cookbooks (Utilizza i libri di ricette di Chef personalizzati), scegli Yes (Sì).
6. In Repository type (Tipo di repository), seleziona il tipo di repository che utilizzerai. Se utilizzi l'esempio sopra riportato, scegli Http Archive (Archivio Http).
7. Per Repository URL, inserisci l'archivio in cui hai archiviato il ricettario che hai creato nel passaggio precedente. Se utilizzi l'esempio precedente, immetti **`https://s3.amazonaws.com/aws-cloudwatch/downloads/CloudWatchLogs-Cookbooks.zip`**.

8. Scegli Add Stack (Aggiungi stack) per creare lo stack.

Fase 3: Estendi il tuo ruolo IAM

Per utilizzare CloudWatch i log con le AWS OpsWorks istanze, è necessario estendere il IAM ruolo utilizzato dalle istanze.

1. Apri la console all'IAMindirizzo. 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), sotto Create Your Own Policy (Crea la tua policy), scegli Select (Seleziona). Per ulteriori informazioni sulla creazione di politiche personalizzate, consulta IAMPolicies 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:

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

6. Scegliere Create Policy (Crea policy).
7. Nel riquadro di navigazione, scegli Ruoli, quindi nel riquadro dei contenuti, in Nome ruolo, seleziona il nome del ruolo dell'istanza utilizzato dallo AWS OpsWorks stack. Puoi trovare quello utilizzato dal tuo stack nelle impostazioni dello stack (il valore predefinito è `aws-opsworks-ec2-role`).

> **Note**
>
> Seleziona il nome del ruolo, non la casella di controllo.

8. Nella scheda Permissions (Autorizzazioni), in Managed Policies (Policy gestite), seleziona Attach Policy (Collega policy).
9. Nella pagina Attach Policy (Collega policy), nell'intestazione della tabella (accanto a Filter (Filtro) e Search (Cerca)), scegli Policy Type (Tipo di policy), quindi Customer Managed Policies (Policy gestite dal cliente).
10. Per Customer Managed Policies, seleziona la IAM policy che hai creato sopra e scegli Allega policy.

    Per ulteriori informazioni su utenti e politiche, consulta IAMUtenti e gruppi e Gestione delle IAM politiche nella Guida per l'IAMutente.

Fase 4: aggiungere un livello

1. Apri la AWS OpsWorks console all'indirizzo https://console.aws.amazon.com/opsworks/.
2. Nel riquadro di navigazione scegli Layers (Livelli).
3. Nel riquadro dei contenuti, seleziona un livello e scegli Add layer (Aggiungi livello).
4. Nella OpsWorksscheda, per Tipo di livello, scegli Personalizzato.
5. Nei campi Name (Nome) e Short Name (Nome breve), inserisci il nome lungo e il nome breve del livello, quindi scegli Add layer (Aggiungi livello).
6. Nella scheda Recipes, in Custom Chef Recipes, ci sono diverse rubriche, Setup, Configure, Deploy, Undeploy e Shutdown, che corrispondono agli eventi del ciclo di vita. AWS OpsWorks AWS OpsWorks attiva questi eventi in questi punti chiave del ciclo di vita dell'istanza, che esegue le ricette associate.

   > **Note**
   >
   > Se le intestazioni sopra riportate non risultano visibili, sotto Custom Chef Recipes (Ricette di Chef personalizzate), scegli edit (modifica).

7. Inserisci logs::config, logs::install accanto a Setup (Installa), scegli + per aggiungerlo all'elenco, quindi scegli Save (Salva).

AWS OpsWorks esegue questa ricetta su ciascuna delle nuove istanze di questo livello, subito dopo l'avvio dell'istanza.

Fase 5: aggiungere un'istanza

Il livello controlla solo la configurazione delle istanze. Sarà ora necessario aggiungere al livello delle istanze e avviarle.

1. Apri la AWS OpsWorks console all'indirizzo https://console.aws.amazon.com/opsworks/.
2. Nel riquadro di navigazione scegli Instances (Istanze), quindi, sotto il livello, scegli + Instance (+ istanza).
3. Accetta le impostazioni predefinite e scegli Aggiungi istanza per aggiungere l'istanza al livello.
4. Nella colonna Actions (Operazioni) della riga, fai clic su start (avvia) per avviare l'istanza.

   AWS OpsWorks avvia una nuova EC2 istanza e configura i log. CloudWatch Quando è pronto, lo stato dell'istanza diventa online.

Fase 6: visualizzazione dei log

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 .

Segnala lo stato dell'agente CloudWatch Logs

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

Indicare lo stato dell'agente

1. Connect alla tua EC2 istanza. 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'agente CloudWatch Logs.

Avvia l'agente Logs CloudWatch

Se l'agente CloudWatch Logs sull'EC2istanza non si è avviato automaticamente dopo l'installazione, o se l'agente è stato interrotto, è possibile utilizzare la procedura seguente per avviare l'agente.

Avvio dell'agente

1. Connect alla tua EC2 istanza. 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' CloudWatch agente Logs

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

Arresto dell'agente

1. Connect alla tua EC2 istanza. 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
```

Avvio rapido: utilizzalo AWS CloudFormation per iniziare a usare CloudWatch Logs

AWS CloudFormation consente di descrivere e fornire le AWS risorse in JSON formato. 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 AWS 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. AWS 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 .

Utilizzo dei CloudWatch log con un SDK AWS

AWS i kit di sviluppo software (SDKs) 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 for C++	AWS SDK for C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice
AWS SDK for Java	AWS SDK for Java esempi di codice
AWS SDK for JavaScript	AWS SDK for JavaScript esempi di codice
SDK AWS for Kotlin	SDK AWS for Kotlin esempi di codice
AWS SDK for .NET	AWS SDK for .NET esempi di codice
AWS SDK for PHP	AWS SDK for PHP esempi di codice
AWS Tools for PowerShell	Strumenti per esempi di PowerShell codice
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) esempi di codice
AWS SDK for Ruby	AWS SDK for Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

Per esempi specifici per CloudWatch Logs, vedereEsempi di codice per l'utilizzo di CloudWatch Logs AWS SDKs.

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

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 le query per rispondere in modo rapido ed efficiente a problemi operativi. Se si verifica un problema, puoi utilizzare CloudWatch Logs Insights per identificare potenziali cause e convalidare le correzioni implementate.

CloudWatch Logs Insights supporta tre 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.
- (Nuovo) OpenSearch Service Piped Processing Language (PPL). OpenSearch PPL consente di analizzare i registri 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.
- (Nuovo) OpenSearch Service Structured Query Language (SQL). Con le query OpenSearch SQL, è possibile 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 operazioni JOINs su più 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.

CloudWatch Logs Insights offre le seguenti funzionalità che sono 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 notoriamente non includono 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.
- Salvataggio delle interrogazioni, visualizzazione della cronologia delle query e riesecuzione delle interrogazioni salvate. Ciò ti consente di eseguire query complesse quando necessario, senza doverle ricreare ogni volta che desideri eseguirle.
- Aggiungere interrogazioni alle dashboard.
- Crittografia dei risultati delle query con. AWS Key Management Service

Le seguenti funzionalità di CloudWatch Logs Insights sono supportate solo quando si utilizza Logs Insights QL.

- Generazione di query utilizzando il linguaggio naturale.
- Interrogazione dei log nella classe di log Infrequent Access.
- 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.
- Il *`filterIndex`*comando, che impone alla query di tentare di analizzare solo gli eventi di registro che contengono un indice di campo specificato dall'utente.

> **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 funzionalità assistita dall'intelligenza artificiale genera una query in base al prompt dell'utente e fornisce una spiegazione del funzionamento della query. line-by-line 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 sette giorni.

CloudWatch Le query di Logs Insights comportano costi in base alla quantità di dati interrogati, 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 È possibile utilizzare le funzionalità di interrogazione di CloudWatch Logs Insights utilizzando. APIs Per ulteriori informazioni, StartQueryconsulta Amazon CloudWatch Logs API Reference.

Indice

Linguaggi di interrogazione supportati

Nelle sezioni seguenti sono elencati i comandi supportati in ogni linguaggio di interrogazione. Descrivono inoltre il formato della sintassi e forniscono interrogazioni di esempio.

Argomenti

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.

Argomenti

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`

`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 l'indice di quel 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 in base al nome del gruppo di log, al prefisso, agli identificatori di account e alla classe del gruppo di log da includere in una query. Questo comando è supportato solo quando si crea una query in AWS CLI o a livello di programmazione, 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".
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.
Altre operazioni e funzioni	CloudWatch Logs Insights supporta anche molte funzioni e operazion i 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
- display
- campi
- filter
- FilterIndex
- SOURCE
- pattern
- diff
- parse
- sort
- statistiche
- limit

- dedup
- unmask
- unnest
- Funzioni booleane, di confronto, numeriche, datetime e altre
- 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 trannepattern,, difffilterIndex, e. unmask

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
```

Utilizza `fields` invece di `display`. Quando desideri utilizzare diverse funzioni e operazioni supportate da `fields` per la modifica dei valori dei campi e la creazione di nuovi campi che possono essere utilizzati 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 hai creato indici di campo per un gruppo di log, puoi sfruttare tali indici di campo per rendere le tue `filter` query più efficienti e ridurre il volume di scansioni. 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 sfrutterà gli indici di campo per tentare di analizzare il minor numero di eventi di registro per trovare eventi con `requestId` 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 registro 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
```

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 i 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.

`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 cinque prefissi nell'elenco.

- `accountIdentifiers`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.
- `logGroupClass`esegue 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 sfruttano 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(accountIdentifiers:['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(accountIdentifiers:['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"
```

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 dei log. È possibile utilizzarlo `pattern` per evidenziare le tendenze emergenti, monitorare gli errori noti e identificare le righe di registro ricorrenti 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 richieste. IDs

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`.

Examples (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

Utilizza `parse` per estrarre i dati da un campo di log e creare un campo estratto che puoi elaborare nella query. **`parse`** supporta sia la modalità glob con i caratteri jolly che le espressioni regolari. Per informazioni sulla sintassi delle espressioni regolari, vedere. Sintassi delle espressioni regolari (regex) supportate

Puoi analizzare i campi JSON annidati con un'espressione regolare.

Esempio: analisi di un campo JSON annidato

Il frammento di codice mostra come analizzare un log eventi JSON che è stato dissociato durante l'acquisizione.

```
{'fieldsA': 'logs', 'fieldsB': [{'fA': 'a1'}, {'fA': 'a2'}]}
```

Il frammento di codice mostra una query con un'espressione regolare che estrae i valori per `fieldsA` e `fieldsB` per creare i campi estratti `fld` e `array`.

```
parse @message "'fieldsA': '*', 'fieldsB': ['*']" as fld, array
```

Gruppi di acquisizione denominati

Quando si utilizza **`parse`** con un'espressione regolare, è possibile utilizzare gruppi di acquisizione denominati per acquisire un pattern in un campo. La sintassi è `parse @message (?<Name>pattern)`.

L'esempio seguente utilizza un gruppo di acquisizione su un log di flusso VPC per estrarre l'ENI in un campo denominato `NetworkInterface`.

```
parse @message /(?<NetworkInterface>eni-.*?) / | display NetworkInterface, @message
```

> **Note**
>
> I log eventi JSON vengono appiattiti durante l'acquisizione. Attualmente, l'analisi di campi JSON annidati con un'espressione glob non è supportata. È possibile analizzare solo i log eventi JSON che includono non più di 200 campi di log eventi. Quando si analizzano i campi JSON annidati, è necessario formattare l'espressione regolare nella query in modo che corrisponda al formato del log eventi JSON.

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
```

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
```

statistiche

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)
```

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.

Examples (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.

Examples (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 fino a due comandi `stats` in un'unica query. Ciò consente di eseguire un'aggregazione aggiuntiva sull'output della prima aggregazione.

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 due comandi `stats` con altri comandi come `filter` e `fields` in un'unica 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 count_distinct(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 al massimo due comandi `stats`. Questa quota non può essere modificata.

Se si utilizza un comando `sort` o `limit`, questo deve apparire dopo il secondo comando `stats`. Se è precedente al secondo comando `stats`, la query non è valida.

Quando una query ha due comandi `stats`, i risultati parziali della query non iniziano a essere visualizzati fino al completamento della prima aggregazione `stats`.

Nel secondo comando `stats` di un'unica query, è possibile fare riferimento solo ai campi definiti nel primo comando `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 il primo comando `stats` devono essere definiti in tale primo comando `stats`.

```
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 utilizzare `bin` nel secondo comando `stats` senza utilizzare il primo comando `stats` per propagare il campo `timestamp`. 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
```

Definisci invece il campo `@timestamp` nel primo comando `stats`, quindi potrai utilizzarlo con `dateceil` nel secondo comando `stats`, 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	Descrizione
`avg(fieldName: NumericLogField)`	number	La media dei valori nel campo specificato.

`count()`
 	
`count(fieldName: LogField)`
	number	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.

`count_distinct(fieldName: LogField)`	number	Restituisce il numero di valori univoci per il campo. Se il campo dispone di alta cardinalità (contiene molti valori univoci), il valore restituito da `count_distinct` è solo un'approssimazione.

Funzione	Tipo di risultato	Descrizione
`max(fieldName: LogField)`	LogFieldValue	Il numero massimo di valori per questo campo di log nei log di query.
`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)`	number	La deviazione standard dei valori nel campo specificato.
`sum(fieldName: NumericLogField)`	number	La somma dei valori nel campo specificato.

Funzioni di non aggregazione delle statistiche

Utilizza funzioni di non aggregazione nel comando `stats` e come argomenti per altre funzioni.

Funzione	Tipo di risultato	Descrizione
`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.

Funzione	Tipo di risultato	Descrizione
`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`, la query restituirà fino a 10.000 eventi di registro nei risultati.

L'esempio seguente restituisce solo i 25 log eventi più recenti

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

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`.

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 `MAP`numbers, viene considerato come un elenco contenente un elemento`unnest`.

Struttura dei comandi

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
```

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.

Operazione	Descrizione
`a + b`	Addizione
`a - b`	Sottrazione
`a * b`	Moltiplicazione
`a / b`	Divisione
`a ^ b`	Elevamento a potenza (`2 ^ 3` restituisce 8)
`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	Descrizione
=	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.

Operazione	Tipo di risultato	Descrizione
`abs(a: number)`	number	Valore assoluto
`ceil(a: number)`	number	Arrotonda per eccesso (l'intero più piccolo che è maggiore del valore di a)
`floor(a: number)`	number	Arrotonda per difetto (l'intero più grande che è minore del valore di a)

Operazione	Tipo di risultato	Descrizione
`greatest(a: number, ...numbers: number[])`	number	Restituisce il valore più grande
`least(a: number, ...numbers: number[])`	number	Restituisce il valore di piccolo
`log(a: number)`	number	Log naturale
`sqrt(a: number)`	number	Radice quadrata

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

Si usa `jsonParse` 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"
```

Se nel nome della chiave sono presenti caratteri speciali, racchiudi i caratteri speciali nei backtick (``).

```
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	Descrizione

`bin(period: Period)`	Timestamp	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)`
		`| 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,

Funzione	Tipo di risultato	Descrizione
		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)`	Timestamp	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)`	Timestamp	Arrotonda il time stamp al periodo specifica to e quindi tronca. Ad esempio, `dateceil( @timestamp, 1h)` tronca tutti i valori di `@timestamp` all'inizio di ogni ora.
`fromMillis(fieldNa me: number)`	Timestamp	Interpreta il campo di input come il numero di millisecondi dall'epoca Unix e lo converte in un timestamp.

Funzione	Tipo di risultato	Descrizione
`toMillis(fieldName: Timestamp)`	number	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` .

> **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	Descrizione
`ispresent(fieldName: LogField)`	Booleano	Restituisce `true` se il campo esiste
`coalesce(fieldName: LogField, ...fieldNames: LogField[ ])`	LogField	Restituisce il primo valore non nullo dall'elenco.

Funzioni JSON

Funzioni JSON

Usa le funzioni JSON nei `filter` comandi `fields` and e come argomenti per altre funzioni.

Funzione	Tipo di risultato	Descrizione
`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.

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	Descrizione
`isValidIp(fieldName: string)`	booleano	Restituisce `true` se il campo è un IPv6 indirizzo IPv4 o valido.
`isValidIpV4(fieldName: string)`	booleano	Restituisce `true` se il campo è un IPv4 indirizzo valido.
`isValidIpV6(fieldName: string)`	booleano	Restituisce `true` se il campo è un IPv6 indirizzo valido.
`isIpInSubnet(fieldName: string, subnet: string)`	booleano	Restituisce `true` se il campo è un IPv6 indirizzo IPv4 or valido all'interno della sottorete v4 o v6 specificata. Quando si specifica la sottorete, utilizzare la notazione CIDR, come

Funzione	Tipo di risultato	Descrizione
		`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 IPv4 indirizzo 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 IPv6 indirizzo 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.

Funzioni stringa

Funzioni stringa

Utilizza le funzioni stringa nei comandi `fields` e `filter`, oltre che come argomenti per altre funzioni.

Funzione	Tipo di risultato	Descrizione
`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.

Funzione	Tipo di risultato	Descrizione
`concat(str: string, ...strings: string[])`	string	Concatena le stringhe.

`ltrim(str: string)`
 	
`ltrim(str: string, trimChars: string)`
	string	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)`
	string	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, non rimuove gli spazi vuoti. Invece, rimuove i caratteri di `trimChars` dalla destra di `str`. Ad esempio, `rtrim("xyZfooxyxyZ","xyZ")` restituisce `"xyZfoo"`.

Funzione	Tipo di risultato	Descrizione

`trim(str: string)`
 	
`trim(str: string, trimChars: string)`
	string	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)`	number	Restituisce la lunghezza della stringa in punti di codice Unicode.
`toupper(str: string)`	string	Converte la stringa in maiuscolo.
`tolower(str: string)`	string	Converte la stringa in minuscolo.

Funzione	Tipo di risultato	Descrizione

`substr(str: string, startIndex: number)`
 	
`substr(str: string, startIndex: number, length: number)`
	string	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)`	string	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`.
`strcontains(str: string, searchValue: string)`	number	Restituisce 1 se `str` contiene `searchValue` e 0 in caso contrario.

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 (`-`).

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 e riesegui 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. Seleziona 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

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 log per ogni servizio, tipo di evento e Regione AWS .

```
stats count(*) by eventSource, eventName, awsRegion
```

Trova gli EC2 host Amazon 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 ARNs gli 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
```

Interrogazioni per Amazon API Gateway

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 Amazon API Gateway richieste che richiedono più tempo nel tuo gruppo di log di accesso Amazon API Gateway

```
fields @timestamp, status, ip, path, httpMethod, responseLatency
| sort responseLatency desc
| limit 10
```

Restituisci l'elenco dei percorsi API più popolari nel tuo Amazon API Gateway gruppo di log di accesso

```
stats count(*) as requestCount by path
| sort requestCount desc
| limit 10
```

Crea un rapporto sulla latenza di integrazione per il tuo gruppo di log di Amazon API Gateway accesso

```
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 count_distinct(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
```

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 la pagina CloudWatch dei prezzi di Amazon.

Per eseguire una query di confronto

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Logs, 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.

Utilizza il linguaggio naturale per generare e aggiornare le query di Logs Insights CloudWatch

CloudWatch Logs supporta una funzionalità di interrogazione in linguaggio naturale per aiutarti a generare e aggiornare le query per CloudWatch Logs Insights e Metrics Insights. CloudWatch

Con questa funzionalità, puoi porre domande o descrivere i dati di CloudWatch Logs che stai cercando in un inglese semplice. La funzionalità del linguaggio naturale genera un'interrogazione in base a un prompt immesso e fornisce una line-by-line spiegazione del funzionamento della query. Puoi anche aggiornare la tua query per analizzare ulteriormente i dati.

A seconda dell'ambiente, è possibile inserire richieste 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 generalmente disponibile in 10 regioni. Per alcune regioni, la funzionalità effettua chiamate interregionali verso le regioni degli Stati Uniti d'America per elaborare le richieste di interrogazione. La tabella seguente elenca le regioni supportate e mostra dove ciascuna regione elabora i propri prompt.

Regione supportata	Regione in cui viene elaborata la richiesta
Stati Uniti orientali (Virginia settentrionale)	Stati Uniti orientali (Virginia settentrionale)

Regione supportata	Regione in cui viene elaborata la richiesta
Stati Uniti orientali (Ohio)	Stati Uniti orientali (Virginia settentrionale)
US West (Oregon)	US West (Oregon)
Asia Pacifico (Hong Kong)	US West (Oregon)
Asia Pacifico (Singapore)	US West (Oregon)
Asia Pacifico (Sydney)	US West (Oregon)
Asia Pacifico (Tokyo)	Asia Pacifico (Tokyo)
Europa (Francoforte)	Europa (Francoforte)
Europa (Irlanda)	Stati Uniti orientali (Virginia settentrionale)
Europa (Stoccolma)	Stati Uniti orientali (Virginia settentrionale)

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 eseguire la query 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.

Esempio: 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 una query che esegue 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 riportato un esempio di query generata dalla funzionalità di 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.

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 della query aggiornata. 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 Logs Insights. CloudWatch 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 .

OpenSearch Linguaggio 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.

È possibile utilizzare OpenSearch PPL solo per le interrogazioni dei gruppi di log nella Standard Log Class.

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

Comando o funzione	Query di esempio	Descrizione
campi	`fields field1, field2`	Visualizza un insieme di campi che necessitano di proiezione.
dove	`where field1="success" | where field2 != "i-023fe0a90929d8822" | fields field3, field4, field5,field6 | head 1000`	Filtra i dati in base alle condizioni specificate.
statistiche	`stats count(), count(field1), min(field1), 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.

Comando o funzione	Query di esempio	Descrizione
		Il pattern estratto può essere ulteriormente utilizzato per creare nuovi campi o filtrare i dati.
sort	`stats count(), count(field1), min(field1) as field1Alias, max(`field1`), avg(`field1`) by field2 | sort -field1Alias | head 1000`	Ordina i risultati visualizzati in base al nome di un campo. Usa sort - FieldName per ordinare in ordine decrescente.
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 modificare matematicamente una colonna, applicare funzioni di stringa a una colonna o applicare funzioni di data a una colonna.

Comando o funzione	Query di esempio	Descrizione
rinominare	`rename field2 as field1 | fields field1;`	Rinomina uno o più campi nel risultato della ricerca.
head	`fields `@message` | head 20`	Limita i risultati dell'interrogazione visualizzati alle prime N righe.
top	`top 2 field1 by field2`	Trova i valori più frequenti per un campo.
dedup	`dedup field1 | fields field1, field2, field3`	Rimuove le voci duplicate in base ai campi specificati.
rari	`rare field1 by field2`	Trova i valori meno frequenti di tutti i campi nell'elenco dei campi.
linea di tendenza	`trendline sma(2, field1) as field1Alias`	Calcola le medie mobili dei campi.

Comando o funzione	Query di esempio	Descrizione
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.
riepilogo del campo	`where field1 != 200 | fieldsummary includefields= field1 nulls=true`	Calcola le statistiche di base per ogni campo (conteggi o, conteggio distinto, min, max, avg, stddev e mean).
grok	`grok email '.+@%{HOSTNAME:host}' | fields email, host`	Analizza un campo di testo con un pattern grok e aggiunge i risultati al risultato della ricerca.

Comando o funzione	Query di esempio	Descrizione
Funzioni stringa	`eval field1Len = LENGTH(field1) | fields field1Len`	Funzioni integrate in PPL che possono 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.
Funzioni matematiche	`eval field2 = ACOS(field1) | fields field1`	Funzioni integrate per eseguire calcoli e trasformazioni matematiche nelle query PPL. Ad esempio, abs (valore assoluto), round (arrotonda i numeri), sqrt (radice quadrata), pow (calcolo della potenza) e ceil (arrotonda al numero intero più vicino).

Comando o funzione	Query di esempio	Descrizione
Funzioni di data	`eval newDate = ADDDATE(DATE('2020-08-26'), 1) | fields newDate`	Funzioni integrate per la gestione e la trasformazione dei dati di data e ora nelle query PPL. Ad esempio, date_add, date_format, datediff e current_date.
Funzioni di condizione	`eval field2 = isnull(field1) | fields field2, field1, field3`	Funzioni integrate che controllano le condizioni specifiche dei campi e valutano le espressioni in modo condizionale. Ad esempio, se field1 è nullo, restituisci field2.

Comando o funzione	Query di esempio	Descrizione
Funzioni matematiche	`eval field2 = ACOS(field1) | fields field1`	Funzioni integrate 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

OpenSearch Linguaggio 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 JOINs più 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.

È possibile utilizzare OpenSearch SQL solo per le query dei gruppi di log nella Standard Log Class.

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.

Comandi SQL supportati

Comando o funzione	Query di esempio	Descrizione
SELECT	`SELECT \`@message\`, Operation FROM \`LogGroupA\``	Visualizza i valori proiettati.
FROM	`SELECT \`@message\`, Operation FROM \`LogGroupA\``	Clausola incorporata 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.
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.

Comando o funzione	Query di esempio	Descrizione
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
LIMIT	`Select * from `LogGroupA` limit 10`	Limita i risultati della query visualizzati alle prime N righe.

Comando o funzione	Query di esempio	Descrizione
Funzioni stringa	`SELECT upper(Operation) , lower(Operation), Operation FROM `LogGroupA``	Funzioni integrate 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``	Funzioni integrate 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	Descrizione
Funzioni condizionali	`SELECT Operation, IF(Error > 0, 'High', 'Low') as error_category FROM `LogGroup A`;`	Funzioni integrate 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``	Funzioni integrate 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	Descrizione
Funzioni JSON	`SELECT get_json_object(json_column, '$.name') as name FROM `LogGroupA``	Funzioni integrate per l'analisi, l'estrazione, la modifica e l'interrogazione di dati in formato JSON 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	Descrizione
Funzioni di array	`SELECT scores, size(scores) as length, array_contains(scores, 90) as has_90 FROM `LogGroupA`;`	Funzioni integrate 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).

Comando o funzione	Query di esempio	Descrizione
Funzioni finestra	`SELECT field1, field2, RANK() OVER (ORDER BY field2 DESC) as field2Rank FROM` `` `LogGroupA` ``;	Funzioni integrate che eseguono calcoli su un insieme specifico di righe relative alla riga (finestra) corrente, abilitando operazioni come la classific azione, l'esecuzi one dei totali e le medie mobili. Ad esempio, ROW_NUMBE R, RANK, LAG e LEAD

Comando o funzione	Query di esempio	Descrizione
Funzioni di conversione	`SELECT CAST('123' AS INT) as converted_number, CAST(123 AS STRING) as converted_string FROM `LogGroupA``	Funzioni integrate per la conversione dei 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`;`	Funzioni integrate che valutano le condizioni e restituiscono valori booleani (vero/falso) in base a criteri o modelli specifici. Ad esempio, IN, LIKE, BETWEEN, IS NULL ed EXISTS.

Comando o funzione	Query di esempio	Descrizione
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

SQL supportato per le multi-log-group interrogazioni

Per supportare il caso d'uso per l'interrogazione di più gruppi di log in SQL, è possibile utilizzare il `logGroups` comando. 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, usa ARNs al posto dei `LogGroup` 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'
```

Restrizioni

Le seguenti restrizioni si applicano quando si utilizza OpenSearch SQL per eseguire query in CloudWatch Logs Insights.

- È possibile includere un solo JOIN in un'istruzione SELECT.
- 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) non sono supportate. FIELD1

  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 colonne 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;
```

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 log Lambda contiene un ID di traccia X-Ray, 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.

> **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

È possibile creare indici di campo dei campi negli 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.

Puoi anche sfruttare 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 I registri indicizzano 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 come creare indici di campo. Per informazioni sul riferimento agli indici di campo nelle query, consulta e. FilterIndex filter

Argomenti

Sintassi e quote degli indici 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.

Le politiche di indicizzazione dei campi a livello di gruppo di log hanno la precedenza sulle politiche di indice dei campi a livello di account: se crei una politica di indicizzazione a livello di gruppo di log, quel gruppo di log utilizza solo quella politica e ignora le politiche a livello di account.

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`

Puoi avere fino a 20 politiche di indicizzazione a livello di account. Se hai più politiche di indicizzazione a livello di account filtrate per registrare i prefissi dei nomi dei gruppi di log, nessuno di essi 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`

Se disponi 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.

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.

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`
- `@xraySetmentId`

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.

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. L'interrogazione 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 tua nuova politica.
6. Per Select log groups, effettuate 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 Seleziona i gruppi di log per prefisso match per applicare la policy 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.
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. 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.

8. Ripeti il passaggio precedente per aggiungere fino a 20 indici di campo.
9. 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. Ripeti il passaggio precedente per aggiungere fino a 20 indici di campo.
8. 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 riquadro di navigazione, scegli Logs, Logs Insights.
3. 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.

4. 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.

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 richieste. IDs 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 riquadro di navigazione, scegli Logs, Logs Insights.

   L'editor di query della pagina Logs Insights contiene una query predefinita che restituisce gli ultimi 20 log eventi.

3. Rimuovi la `| limit 20` riga nella casella della query, in modo che la query abbia l'aspetto seguente:

```
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, usapattern @fieldname.
- Le query con output non di registro, ad esempio le query con il stats comando, non restituiscono risultati di pattern.

Salva e riesegui 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 livello specifico della regione, non a un livello specifico dell'utente. Se si crea e si salva un'interrogazione, 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.

Per salvare una 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. Nell'editor di query, crea una query.
4. Seleziona Salva.

   Se non vedi il pulsante Salva, devi passare al nuovo design della console CloudWatch Logs. A tale scopo:

   a. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
   b. Scegliere Try the new design (Prova il nuovo design).
   c. Nel riquadro di spostamento scegliere Insights (Informazioni dettagliate) e tornare alla fase 3 di questa procedura.
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. Seleziona Salva.

> **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, consulta PutQueryDefinition.

Per eseguire una query salvata

1. Aprire 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. Seleziona Esegui.

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. Seleziona 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).

Aggiunta di query a pannello di controllo o esportazione dei risultati della query

Dopo aver eseguito una query, puoi aggiungere la query a un CloudWatch pannello di controllo 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 30 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 30 query simultanee di CloudWatch Logs Insights per account, incluse le query aggiunte ai dashboard. Solo 15 di queste 30 query possono utilizzare Service PPL o OpenSearch Service SQL. OpenSearch

Per visualizzare la cronologia delle query recenti

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 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 quei risultati 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 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.

Limiti

- 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

Fase 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.

Innanzitutto, salva la politica predefinita per la tua chiave KMS `policy.json` utilizzando il seguente get-key-policycomando:

```
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 `Condition` sezione di questo esempio limita l'uso della AWS KMS chiave ai risultati della query CloudWatch Logs Insights nell'account specificato.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
```

```
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_ID: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:region:account_ID:query-result:*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "Your_account_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 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 segue:

```
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, usa il seguente disassociate-kms-keycomando:

```
aws logs disassociate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-result:*"
```

Rilevamento delle anomalie nei registri

È possibile creare un rilevatore di anomalie di registro per ogni 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. Il rilevamento delle anomalie utilizza l'apprendimento automatico e il riconoscimento dei pattern per stabilire le linee di base dei contenuti tipici dei log.

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*> Il *string* è una descrizione del tipo di dati rappresentato dal token. Il *number* mostra dove nel pattern 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*Se Logs non è in grado di dedurre il tipo di dati rappresentato da un token dinamico, visualizza il token come 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 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. ListAnomaliesAPI 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. Gruppi di log con eventi che contengono parole chiave relative al livello di registro o alla gravità INFO, come ERROR, e DEBUGsono particolarmente adatti per il rilevamento di anomalie nei log.

Il rilevamento delle anomalie nei log non è adatto per: Registra eventi con JSON strutture estremamente lunghe, come i registri. 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 VPC flusso, 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.

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, vedere. CreateLogAnomalyDetector

Per creare un rilevatore di anomalie nel 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 la chiave in. ARN 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, vedere. 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 apportate agli 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 mostra un istogramma che mostra 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 apportate agli 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 mostra un istogramma che mostra 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 ogni volta ***your-metric-name*** è. , 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 SNS argomento Amazon a cui inviare una notifica quando l'allarme passa allo `INSUFFICIENT_DATA` stato `ALARMOK`, o.

    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 non inviare SNS notifiche Amazon, scegli Rimuovi.
11. (Facoltativo) Se desideri che l'allarme esegua azioni per Amazon EC2 Auto ScalingEC2, Amazon, ticket oppure AWS Systems Manager 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 UTF -8 caratteri e non può contenere caratteri ASCII di controllo. La descrizione può includere la formattazione markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella CloudWatch console. Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.

> **Tip**
>
> Il nome dell'allarme deve contenere solo UTF -8 caratteri. Non può contenere caratteri ASCII di controllo.

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 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 rilevatore di anomalie, associando una KMS chiave a un rilevatore di anomalie.

> **Important**
>
> CloudWatch Logs supporta solo chiavi simmetriche. KMS 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.

Limiti

- 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 associata, i dati crittografati contenuti nei CloudWatch registri non possono più essere recuperati. KMS

Fase 1: Creare una chiave AWS KMS

Per creare una KMS chiave, usa il seguente comando create-key:

```
aws kms create-key
```

L'output contiene l'ID della chiave e 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"
        ]
    }
}
```

Passaggio 2: imposta le autorizzazioni sulla chiave KMS

Per impostazione predefinita, tutte le AWS KMS chiavi sono private. Solo il proprietario della risorsa può utilizzarla per crittografare e decrittare i dati. Tuttavia, il proprietario della risorsa può concedere le autorizzazioni per accedere alla KMS chiave ad altri utenti e risorse. 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 KMS chiave.

Come procedura ottimale, si consiglia di limitare l'uso della KMS chiave solo agli AWS account o ai rilevatori di anomalie specificati.

Innanzitutto, salvate la politica predefinita per la KMS chiave `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.

```
{
  "Version": "2012-10-17",
  "Id": "key-default-1",
  "Statement": [
    {
```

```
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::Your_account_ID:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.REGION.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "kms:EncryptionContext:aws:logs:arn":
"arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
        }
      }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.REGION.amazonaws.com"
        },
        "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
        ],
        "Resource": "*",
        "Condition": {
            "ArnLike": {
                "kms:EncryptionContext:aws-crypto-ec:aws:logs:arn":
"arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
```

```
            }
        }
    }
  ]
}
```

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 KMS chiave a un rilevatore di anomalie

È possibile associare una KMS chiave a un rilevatore di anomalie quando la si crea 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.

> **Note**
>
> 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).

Avvia una sessione di 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, vedere start-live-tail

È possibile utilizzarlo `start-live-tail` in due modalità:

- 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}'
```

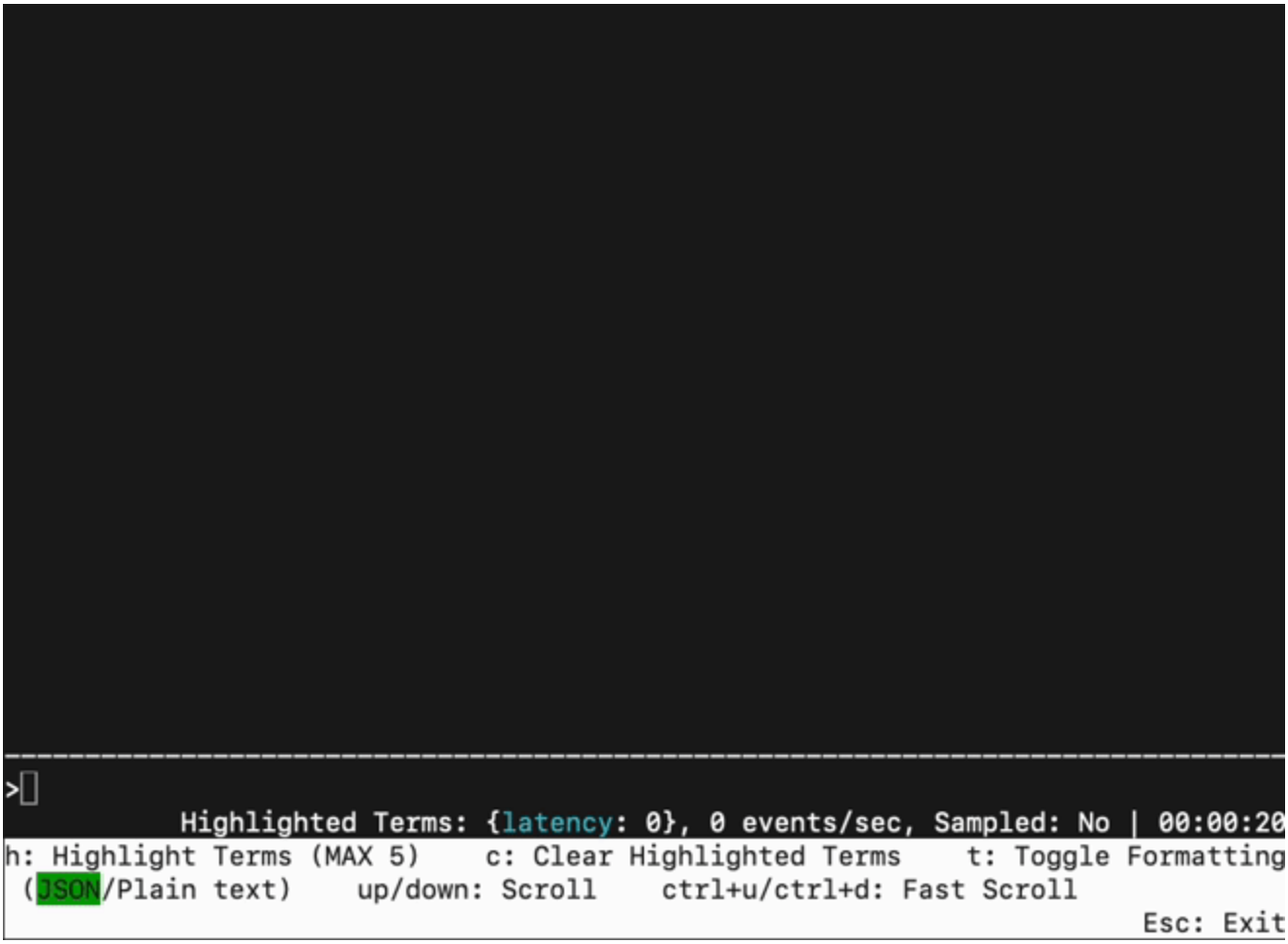
interattivo

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.



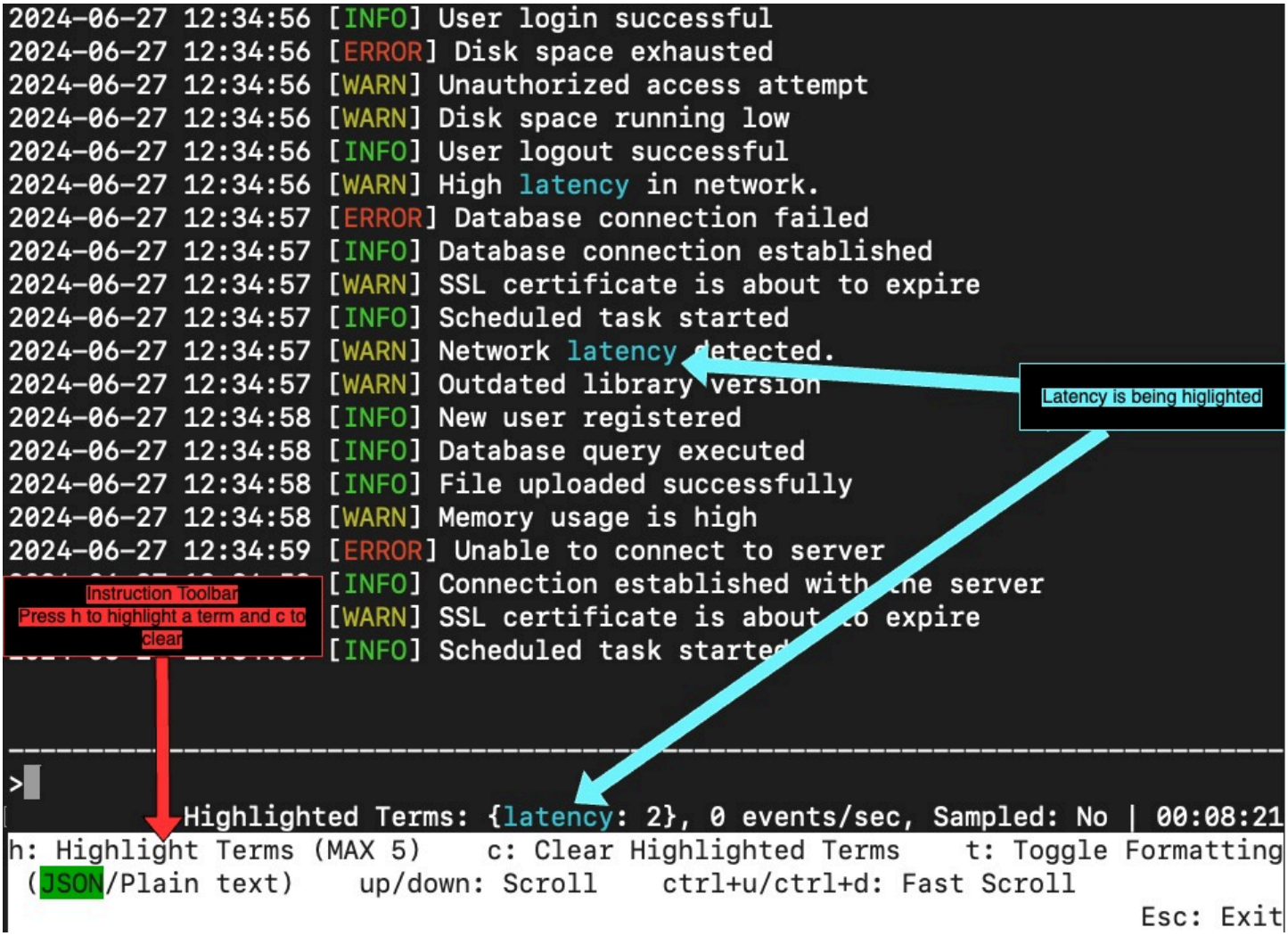
Per evidenziare un termine nei log di streaming, premi h e inserisci il termine. Di seguito viene mostrata la schermata dopo l'evidenziazione del termine`latency`.

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 AWS CLI o APIs. Per ulteriori informazioni, consulta Avvia una sessione di 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.
   - eventi/sec mostra quanti log eventi importati al secondo corrispondono ai filtri impostati.
   - Per evitare che la sessione scorra troppo velocemente perché molti eventi corrispondono ai filtri, nei CloudWatch registri potrebbero essere visualizzati solo alcuni eventi corrispondenti. In tal caso, la percentuale di eventi corrispondenti visualizzati sullo schermo viene mostrata in % visualizzata.

8. Per sospendere il flusso di eventi e analizzare ciò che è attualmente visualizzato, fai clic in un punto qualsiasi della 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).

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.

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'EC2istanza Amazon utilizzando i passaggi nelle sezioni precedenti della Amazon CloudWatch Logs User Guide, il gruppo di log viene creato come parte di tale processo. Puoi anche creare un gruppo di log direttamente nella CloudWatch console.

Creazione di un gruppo di log

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 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 Servers with the Agent CloudWatch nella Amazon User Guide. CloudWatch
- AWS CLIput-log-events—Carica batch di eventi di registro su Logs. CloudWatch
- A livello di codice: PutLogEventsAPIconsente 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 pannello di navigazione, selezionare Log groups (Gruppi di log).
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). Puoi anche passare da un fuso orario locale all'altro. UTC

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 pannello di navigazione, selezionare Log groups (Gruppi di log).
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 pannello di navigazione, selezionare Log groups (Gruppi di log).

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. Inoltre, questi eventi di registro contrassegnati per l'eliminazione non sono inclusi quando si utilizza an API per recuperare il `storedBytes` valore 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.

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

È possibile utilizzare AWS CloudFormation AWS CLI, o CloudWatch Logs API 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.

Taggare i gruppi di log utilizzando i Logs CloudWatch API

È possibile aggiungere, elencare e rimuovere tag utilizzando i CloudWatch registri. API 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 Advanced Encryption Standard Galois/ Counter Mode (-) a 256 bit per crittografare i dati di registro inattivi. AES GCM 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 KMS chiave a un gruppo di log, al momento della creazione del gruppo di log o dopo la sua esistenza.

Important

CloudWatch Logs ora supporta il contesto di crittografia, utilizzando `kms:EncryptionContext:aws:logs:arn` come chiave e il gruppo ARN di log come valore per quella chiave. Se disponi di gruppi di log che hai già crittografato con una KMS chiave e desideri limitare l'utilizzo della chiave con un singolo account e gruppo di log, dovresti assegnare una nuova KMS chiave che includa una condizione nella IAM policy. Per ulteriori informazioni, consulta AWS KMS chiavi e contesto di crittografia.

Important

CloudWatch Ora il supporto Logs `kms:ViaService` consente ai registri di effettuare AWS KMS chiamate per conto dell'utente. Dovresti aggiungerlo ai tuoi ruoli che chiamano CloudWatch Logs in Key Policy o in. IAM Per maggiori informazioni, vedi kms:. ViaService

Dopo aver associato una KMS chiave a un gruppo di log, tutti i dati appena inseriti per il gruppo di log vengono crittografati utilizzando 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 KMS chiave ogni volta che vengono richiesti dati crittografati.

Se successivamente si dissocia una KMS chiave 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 e crittografati con la chiave rimangono crittografati con la chiave. KMS KMS CloudWatch I log possono comunque restituire quei dati dopo che la KMS chiave 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 simmetriche. KMS 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.

Limiti

- 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 si revoca l'accesso CloudWatch dei log a una chiave associata o si elimina una chiave associata, i dati crittografati in CloudWatch Logs non possono più essere recuperati. KMS
- Non è possibile associare una KMS chiave a un gruppo di log utilizzando la console. CloudWatch

Passaggio 1: creare una AWS KMS chiave

Per creare una KMS chiave, usa il seguente comando create-key:

```
aws kms create-key
```

L'output contiene l'ID della chiave e 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"
        ]
    }
}
```

Passaggio 2: imposta le autorizzazioni sulla chiave KMS

Per impostazione predefinita, tutte le AWS KMS chiavi sono private. Solo il proprietario della risorsa può utilizzarla per crittografare e decrittare i dati. Tuttavia, il proprietario della risorsa può concedere le autorizzazioni per accedere alla KMS chiave ad altri utenti e risorse. Con questo passaggio, si concede al responsabile del servizio CloudWatch Logs e al ruolo del chiamante l'autorizzazione a utilizzare la chiave. L'entità del servizio deve trovarsi nella stessa AWS regione in cui è memorizzata la KMS chiave.

Come procedura ottimale, si consiglia di limitare l'uso della KMS chiave solo agli AWS account o ai gruppi di log specificati.

Innanzitutto, salvate la politica predefinita per la KMS chiave `policy.json` utilizzando il seguente get-key-policycomando:

```
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 la chiave a un singolo gruppo ARN di log.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID: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": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id: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.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
```

```
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID: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": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:*"
                }
            }
        }
    ]
}
```

Quindi, aggiungi le autorizzazioni al ruolo che chiamerà i CloudWatch log. Puoi farlo aggiungendo una dichiarazione aggiuntiva alla AWS KMS Key Policy o inserendo IAM il ruolo stesso. CloudWatch Registra gli usi `kms:ViaService` per effettuare chiamate per AWS KMS conto del cliente. Per ulteriori informazioni, vedere kms:. ViaService

Per aggiungere autorizzazioni nella politica AWS KMS chiave, aggiungi la seguente dichiarazione aggiuntiva alla tua politica chiave. Se utilizzi questo metodo, come best practice, applichi la policy solo ai ruoli che interagiranno con i gruppi di log AWS KMS 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"
      ]
    }
  }
}
```

In alternativa, se desideri gestire le autorizzazioni dei ruoli inIAM, puoi aggiungere autorizzazioni equivalenti tramite la seguente politica. Questo può essere aggiunto a una politica di ruolo esistente o allegato 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 IAM le politiche.

```
{
   "Version": "2012-10-17",
   "Statement" : [
     {
        "Effect": "Allow",
        "Action": [
            "kms:Encrypt",
            "kms:ReEncrypt*",
            "kms:Decrypt",
            "kms:GenerateDataKey*
            ",
            "kms:Describe*"
        ],
        "Condition":{
            "StringEquals":{
```

```
            "kms:ViaService": [
                "logs.region.amazonaws.com"
                ]
             }
        },
        "Resource": "arn:aws:kms:region:account_id: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 3: Associare una KMS chiave a un gruppo di log

È possibile associare una KMS chiave a un gruppo di log al momento della creazione o dopo che esiste.

Per scoprire se a un gruppo di log è già associata una KMS chiave, utilizzate il seguente describe-log-groupscomando:

```
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.

Per associare la KMS chiave 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"
```

Per associare la KMS chiave 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"
```

Fase 4: Dissociazione di una chiave da un gruppo di log

Per dissociare la KMS chiave associata a un gruppo di log, utilizzate il seguente disassociate-kms-keycomando:

```
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 condizioni IAM politiche per limitare l'accesso alla AWS KMS chiave per AWS account e gruppo di log. Per ulteriori informazioni, vedere Contesto di crittografia ed elementi IAM JSON della politica: condizione.

Si consiglia di utilizzare KMS chiavi diverse per ciascuno dei gruppi di log crittografati.

Se disponi di un gruppo di log crittografato in precedenza e ora desideri modificare il gruppo di log per utilizzare una nuova KMS chiave che funzioni solo per quel gruppo di log, segui questi passaggi.

Per convertire un gruppo di log crittografato in modo che utilizzi una KMS chiave con una politica che lo limiti a quel gruppo di log

1. Immettete il seguente comando per trovare la ARN chiave corrente del gruppo di log:

   ```
   aws logs describe-log-groups
   ```

   L'output include la seguente riga. Prendi nota delARN. È necessario utilizzarlo nel passaggio 7.

   ```
   ...
   "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
   cdef-0123-456789abcdef"
   ...
   ```

2. Immettete il seguente comando per creare una nuova KMS chiave:

   ```
   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:

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT-ID: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": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn":
 "arn:aws:logs:REGION:ACCOUNT-ID:log-
group:LOG-GROUP-NAME"
                }
            }
        }
    ]
```

```
}
```

5. Immettete il seguente comando per aggiungere la politica aggiornata alla nuova KMS chiave:

```
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`

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID: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'`logs:Unmask`IAMautorizzazione 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 solo per i gruppi di log nella classe di log Standard. Se crei una politica di protezione dei dati per tutti i gruppi di log del tuo account, questa si applica solo ai gruppi di log nella classe di log Standard. Per ulteriori informazioni sulle classi di log, consultaClassi 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 /Logs. AWS 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 di identificazione personale (PII) come patenti di guida o numeri di previdenza sociale

- Informazioni sanitarie protette (PHI) come l'assicurazione sanitaria o i numeri di identificazione medica
- Identificatori del dispositivo, come indirizzi o MAC indirizzi IP

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

Argomenti

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.

JSONproprietà per la politica 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": [
        ...
  ]
}
```

JSONproprietà per una dichiarazione politica

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 che i CloudWatch registri devono ricercare. Ad esempio, nome, indirizzo o numero di telefono.
- Funzionamento: le azioni successive, Audit o De-identity. CloudWatch Logs esegue queste azioni quando rileva dati sensibili.

```
{
  ...
  "Statement": [
    {
      "Sid": "audit-policy",
      "DataIdentifier": [
        "arn:aws:dataprotection::aws:data-identifier/Address"
      ],
      "Operation": {
        "Audit": {
          "FindingsDestination": {}
        }
      }
    },
```

JSONproprietà per un'operazione relativa a una dichiarazione politica

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 corrispondenti incrementano la LogEventsWithFindingsmetrica che CloudWatch Logs pubblica nello spazio dei nomi /Logs. AWS 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.

IAMautorizzazioni necessarie per creare o utilizzare una politica 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.

Operazione	IAMautorizzazione 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`	*

Operazione	IAMautorizzazione necessaria	Risorsa
	`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:::deliverystream/` *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*

Operazione	IAMautorizzazione necessaria	Risorsa
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.

Operazione	IAMautorizzazione necessaria	Risorsa
Crea una policy di protezione dei dati senza destinazioni di verifica	`logs:PutDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Operazione	IAMautorizzazione necessaria	Risorsa
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 firehose:TagDeliveryStream	arn:aws:logs:::log-group: *YOUR_LOG_GROUP* :* * 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*

Operazione	IAMautorizzazione necessaria	Risorsa
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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "YOUR_SID_1",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:DescribeLogGroups",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "YOUR_SID_2",
```

```
            "Effect": "Allow",
            "Action": [
                "logs:GetDataProtectionPolicy",
                "logs:DeleteDataProtectionPolicy",
                "logs:PutDataProtectionPolicy",
                "s3:PutBucketPolicy",
                "firehose:TagDeliveryStream",
                "s3:GetBucketPolicy"
            ],
            "Resource": [
                "arn:aws:firehose:::deliverystream/YOUR_DELIVERY_STREAM",
                "arn:aws:s3:::YOUR_BUCKET",
                "arn:aws:logs:::log-group:YOUR_LOG_GROUP:*"
            ]
        }
    ]
}
```

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 nostre AWS CLI operazioni API

Quando si crea una politica di protezione JSON dei dati da utilizzare in un AWS CLI comando o in un'APIoperazione, la politica deve includere due JSON blocchi:

- 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 nostre AWS CLI operazioni API

Quando si crea una politica di protezione JSON dei dati da utilizzare in un AWS CLI comando o in un'APIoperazione, la politica deve includere due JSON blocchi:

- 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 IAM politica 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 chiavi gestite da Amazon S3 (-S3) o la crittografia lato server con chiavi (SSE-). KMS SSE KMS Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server nella Guida per l'utente di Amazon S3.

Se invii i risultati dell'audit a un bucket protetto con -S3, non è richiesta alcuna configurazione aggiuntiva. SSE Amazon S3 gestisce la chiave di crittografia.

Se invii i risultati dell'audit a un bucket protetto con SSE -KMS, devi aggiornare la policy chiave della chiave in modo che l'account di consegna dei log possa scrivere nel tuo KMS 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

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 informazioni relative a credenziali, dispositivi, finanziarie, mediche e sanitarie protette (PHI) che CloudWatch Logs è in grado di rilevare utilizzando identificatori di dati gestiti. Questi si aggiungono a determinati tipi di dati che potrebbero anche essere considerati informazioni di identificazione personale (). PII

Identificatori supportati indipendenti dalla lingua e dall'area geografica

Identificatore	Categoria
`Address`	Personale

Identificatore	Categoria
`AwsSecretKey`	Credenziali
`CreditCardExpiration`	Servizi finanziari
`CreditCardNumber`	Servizi finanziari
`CreditCardSecurityCode`	Servizi finanziari
`EmailAddress`	Personale
`IpAddress`	Personale
`LatLong`	Personale
`Name`	Personale
`OpenSshPrivateKey`	Credenziali
`PgpPrivateKey`	Credenziali
`PkcsPrivateKey`	Credenziali
`PuttyPrivateKey`	Credenziali
`VehicleIdentificationNumber`	Personale

Gli identificatori di dati dipendenti dalla regione devono includere il nome dell'identificatore, quindi un trattino e quindi i codici a due lettere (3166-1 alfa-2). ISO 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

Identificatore	Categoria	Paesi e lingue
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
HealthInsuranceCardNumber	Integrità	UE
HealthInsuranceClaimNumber	Integrità	US
HealthInsuranceNumber	Integrità	FR
HealthcareProcedureCode	Integrità	US
IndividualTaxIdentificationNumber	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

Identificatore	Categoria	Paesi e lingue
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
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
Aprire la chiave SSH privata	`OpenSSHPrivateKey`	Nessuno	Tutti

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni
PGPchiave privata	`PgpPrivateKey`	Nessuno	Tutti
Chiave privata Pkcs	`PkcsPrivateKey`	Nessuno	Tutti
Chiave TTY privata Pu	`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
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.

Identificatore dei dati del dispositivo ARN
`arn:aws:dataprotection::aws:data-identifier/IpAddress`

Informazioni finanziarie

CloudWatch Logs Data Protection può 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 alfanumer

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
				ici, inclusi elementi come i codici dei paesi.
Data di scadenza della carta di credito	`CreditCardExpiration`	`exp d`, `exp m`, `exp y`, `expiration` , `expiry`	Tutti	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero di carta di credito	`CreditCardNumber`	`account number`, `american express`, `amex`, `bank card`, `card`, `card number`, `card 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`	Tutti	Il rilevamento richiede 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

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
				di credito: American Express, Dankort, 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`

Identificatore di dati finanziari ARNs
`arn:aws:dataprotection::aws:data-identifier/CreditCardExpiration`
`arn:aws:dataprotection::aws:data-identifier/CreditCardNumber`
`arn:aws:dataprotection::aws:data-identifier/CreditCardSecurityCode`

Informazioni sanitarie protette () 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 della Drug Enforcement Agency () DEA	`DrugEnforcementAgencyNumber`	`dea number`, `dea registration`	Stati Uniti
Numero della tessera sanitaria (EHIC)	`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`	Unione Europea

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni
		card number, health insurance card, health insurance number, insurance card number, krankenversicherungskarte , krankenversicherungsnummer , medical account number, numero conto medico, 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	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni
Numero di richiesta di assicurazione sanitaria (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
Codice del sistema di codifica delle procedure comuni sanitarie (HCPCS)	`HealthcareProcedureCode`	`current procedural terminology` , `hcpcs`, `healthcare common procedure coding system`	Stati Uniti
Numero del beneficiario Medicare () MBN	`MedicareBeneficiaryNumber`	`mbi`, `medicare beneficiary`	Stati Uniti
Codice nazionale sulle droghe () NDC	`NationalDrugCode`	`national drug code`, `ndc`	Stati Uniti
Identificatore nazionale del fornitore () NPI	`NationalProviderId`	`hipaa`, `n.p.i.`, `national provider`, `npi`	Stati Uniti
Numero del Servizio Sanitario Nazionale (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 dei 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 delle informazioni sanitarie protette (PHI).

PHIidentificatore di dati ARNs
`arn:aws:dataprotection::aws:data-identifier/DrugEnforcementAgencyNumber-US`
`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 di identificazione personale () PII

CloudWatch I registri di protezione dei dati possono 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 component i della

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
				data possono essere separati da spazi, barre (/) o trattini (-).
Code de Endereçamento Postal () CEP	`CepCode`	`cep`, `código de endereçamento postal`, `codigo de endereçamento postal`	Brasile	
Catastro Nacional da Pessoa Jurídica () CNPJ	`Cnpj`	`cadastro nacional da pessoa jurídica`, `cadastro nacional da pessoa juridica`, `cnpj`	Brasile	
Catastro 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
Istituto nazionale di statistica e studi economici () 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.	Germani Italia, Spagna	Ciò include gli identificatori del Documento Nacional de Identidad (DNI) (Spagna), i codici del Codice fiscale (Italia) e i numeri della carta d'identità nazionale (tedesco).

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Numero di previdenza nazionale () NINO	`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	
Numero di identificazione fiscale 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	
Numero di previdenza sociale () 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
Numero di previdenza sociale (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	Ciò include TIN (Francia), Steueridentifikationsnummer (Germania), (Spagna) CIF e (Regno Unito). TRN UTR

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
ZIPcodice	`ZipCode`	`zip code`, `zip+4`	Stati Uniti	Codice postale degli Stati Uniti.

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
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 ZIP codice o codice postale.
Indirizzo e-mail	`EmailAddress`	Nessuno	Qualsias	

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
Coordinate del Global Positioning System (GPS)	`LatLong`	`coordinate` , `coordinates` , `lat long`, `latitude longitude` , `location`, `position`	Qualsias	CloudWatch I log possono rilevare GPS le coordinate se le coordinate di latitudine e longitudine sono memorizzate come coppia e sono in formato DD (Decimal Degrees), ad esempio 41.948614 , -87,655311. Support non

Tipo di dato	ID dell'identificatore di dati	Parola chiave obbligatoria	Paesi e Regioni	Note
				include le coordinate in formato Degrees Decimal Minutes (DDM), ad esempio 41°56.9168'N 87°39.3187'W, o Degrees, Minutes, Seconds (), ad esempio 41°56'55.0104"N 87°39'19.1196"W. DMS

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 identificazione 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 registri 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. Ciò include gli identificatori del Documento Nacional de

Identidad (DNI) (Spagna), i codici dell'Istituto nazionale francese di statistica e studi economici (INSEE), i numeri delle carte d'identità nazionali tedesche e i numeri del Registro Geral (RG) (Brasile).

La tabella seguente elenca le parole chiave che CloudWatch Logs riconosce per paesi e aree geografiche 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 identificatori di dati Amazon Resource Names (ARNs) per le informazioni di identificazione personale (PII) che puoi aggiungere alle tue politiche di protezione dei dati.

PIIidentificatore di dati 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`

PIIidentificatore di dati ARNs
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GB
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

PIIidentificatore di dati 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

PIIidentificatore di dati 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?

Gli identificatori di dati personalizzati (CDIs) consentono di definire espressioni regolari personalizzate che possono essere utilizzate nella politica di protezione dei dati. Utilizzando identificatori di dati personalizzati, puoi indirizzare le informazioni di identificazione personale specifiche dell'azienda (PII) a casi d'uso che gli identificatori di dati gestiti non sono in grado di fornire. Ad esempio, puoi 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: ('_' | '#' | '=' | '@' |'/' | ';' | ',' | '-' | ' ')
  - RegEx caratteri riservati: ('^' | '$' | '?' | '[' | ']' | '{' | '}' | '|' | '\\' | '*' | '+' | '.')
- 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 AWS CLI o AWS API per specificare un identificatore di dati personalizzato, è necessario includere il nome dell'identificatore di dati e l'espressione regolare nella JSON politica 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

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.

Oltre a trasformarli in diversi formati, puoi anche arricchire i log con un contesto aggiuntivo, ad esempio ID account, regione e 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. Utilizzando la console, è possibile creare trasformatori solo per singoli gruppi di log. Queste istruzioni si trovano in questa sezione. Per informazioni sulla creazione di trasformatori a livello di account, consulta. PutAccountPolicy

Argomenti

Creare e gestire 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.

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.

Per ulteriori informazioni su tutti i processori supportati e sulla relativa sintassi, consulta. Processori che è possibile utilizzare

Argomenti

Crea un trasformatore di log partendo da zero

Usa questi passaggi per creare da zero un trasformatore di log per un gruppo di log.

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 il quale 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 in[Processori 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. Per informazioni sui parametri di configurazione, consultate la sezione relativa al processore in[Processori che è possibile utilizzare](#).

   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`. Di seguito è riportato un esempio che crea un trasformatore che include i processori `parseJSON` e `addKeys`:

```
 aws logs put-transformer \
--transformer-config '[{"parseJSON":{}},{"addKeys":{"entries":
[{"key":"metadata.transformed_in","value":"CloudWatchLogs"},
{"key":"feature","value":"Transformation"}]}},{"trimString":{"withKeys":
["status"]}}]' \
--log-group-identifier my-log-group-name
```

Crea un trasformatore di log 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. Scegliete 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 di log

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. Per informazioni sui parametri di configurazione, consultate la sezione relativa al processore inProcessori che è possibile utilizzare.

   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.

Eliminate un trasformatore di log

Usa questi passaggi per eliminare un trasformatore di log.

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 che è possibile utilizzare

Questa sezione contiene informazioni su ciascun processore che è possibile utilizzare in un trasformatore di eventi di registro. I processori possono essere classificati in parser, mutatori di stringhe, mutatori JSON e processori di data.

Indice

Processori configurabili di tipo parser

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.

Il `@message` contenuto originale non viene modificato, le nuove chiavi vengono aggiunte al messaggio.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
source	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

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
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 utilizzare il pattern matching per analizzare e strutturare dati non strutturati. Questo processore può anche estrarre campi dai messaggi di registro.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
source	Percorso del campo nel registro dell'evento a cui applicare la corrispondenza grok	No	`@message`	Lunghezza massima: 128. Profondità massima della chiave annidata: 3
match	Il pattern grok da confrontare con l'evento log. I pattern grok supportati sono elencati alla fine di questa sezione.	Sì		Lunghezza massima: 128. Massimo 5 pattern grok. i pattern grok non supporteranno conversioni di tipo. Per i modelli di formato di log più comuni (APACHE_ACCESS_LOG , NGINX_ACCESS_LOG,,) è supportato solo l'inclusione dei modelli GREEDYDATA o DATA dopo il modello di log comune. SYSLOG5424

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"
}
```

Esempio 2

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 3: 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"
}
```

Schemi grok supportati

Le tabelle seguenti elencano i modelli supportati dal `grok` processore.

Schemi di gruppo generali

Pattern	Esempio	Descrizione
NOME UTENTE o UTENTE	Inserimento: `user123.name-TEST` Modello: `%{USERNAME:name}` Uscita: `{"name": "user123.name-TEST"}`	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 (-)
INT	Input: `-456` Modello: `%{INT:num}` Uscita: `{"num": "-456"}`	Corrisponde a un segno più o meno facoltativo seguito da una o più cifre.
BASE10 NUM	Ingresso: `-0.67` Modello: `%{BASE10NUM:num}` Uscita: `{"num": "-0.67"}`	Corrisponde a un numero intero o a virgola mobile con segno e virgola decimale opzionali.
BASE16NUM	Ingresso: `+0xA1B2`	Corrisponde ai numeri decimali ed esadecimali con un segno opzionale (+ o -) e un prefisso 0x opzionale .

Pattern	Esempio	Descrizione
	Modello: `%{BASE16NUM:num}` Uscita: `{"num": "+0xA1B2"}`	
PUNTO	Ingresso: `123` Modello: `%{POSINT:num}` Uscita: `{"num": "123"}`	Corrisponde a numeri interi positivi senza zeri iniziali, composti da una o più cifre (1-9 seguito da 0-9).
NON NEGINT	Ingresso: `008` Modello: `%{NONNEGINT:num}` Uscita: `{"num": "008"}`	Corrisponde a qualsiasi numero intero (composto da una o più cifre da 0 a 9) compresi zero e numeri con zeri iniziali.
PAROLA	Ingresso: `abc def` Modello: `%{WORD:user}` Uscita: `{"user": "user_123"}`	Corrisponde a parole intere composte da uno o più caratteri di parola (\ w), incluse lettere, cifre e caratteri di sottolineatura.

Pattern	Esempio	Descrizione
NOTSPACE	Ingresso: `hello_world123` Modello: `%{NOTSPACE:msg}` Uscita: `{"msg": "hello_world123"}`	Corrisponde a uno o più caratteri diversi dagli spazi bianchi.
SPACE	Ingresso: `" "` Modello: `%{SPACE:extra}` Uscita: `{"extra": " "}`	Corrisponde a zero o più caratteri di spazio bianco.
DATA	Ingresso: `abc def` Modello: `%{DATA:data}` Uscita: `{"data": "abc def"}`	Corrisponde a qualsiasi carattere (eccetto la nuova riga) zero o più volte, non è avido.
DATI AVIDI	Ingresso: `abc def` Modello: `%{GREEDYDATA:data}` Uscita: `{"data": "abc def"}`	Corrisponde a qualsiasi carattere (eccetto la nuova riga) zero o più volte, greedy.

Pattern	Esempio	Descrizione
STRINGA TRA VIRGOLETTE	Ingresso: `"Hello, world!` Modello: `%{QUOTEDSTRING:msg}` Uscita: `{"msg": "Hello, world!"}`	Corrisponde alle stringhe tra virgolette (virgolette singole o doppie) ai caratteri con escape.
UUID	Ingresso: `550e8400-e29b-41d4-a716-446655440000` Modello: `%{UUID:id}` Uscita: `{"id": "550e8400-e29b-41d4-a716-446655440000"}`	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.

Pattern	Esempio	Descrizione
BRUCIA	Ingresso: `urn:isbn:0451450523` Modello: `%{URN:urn}` Uscita: `{"urn": "urn:isbn:0451450523"}`	Corrisponde alla sintassi URN (Uniform Resource Name).

AWS modelli grok

Pattern	Esempio	Descrizione
ARN	Ingresso: `arn:aws:iam:us-east-1:123456789012:user/johndoe` Modello: `%{ARN:arn}` Uscita: `{"arn": "arn:aws:iam:us-east-1:123456789012:user/johndoe"}`	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 identificatori gerarchici di risorse separati da barre. Non ARNs corrisponderà alle informazioni mancanti tra i due punti.

Schemi di gruppo di rete

Pattern	Esempio	Descrizione
CISCOMAC	Ingresso: `0123.4567.89AB` Modello: `%{CISCOMAC:MacAddress}` Uscita: `{"MacAddress": "0123.4567.89AB"}`	Corrisponde a un indirizzo MAC in formato esadecimale 4-4-4.
WINDOWS MAC	Ingresso: `01-23-45-67-89-AB` Modello: `%{WINDOWSMAC:MacAddress}` Uscita: `{"MacAddress": "01-23-45-67-89-AB"}`	Corrisponde a un indirizzo MAC in formato esadecimale con trattini.
MAC COMUNE	Ingresso: `01:23:45:67:89:AB`	Corrisponde a un indirizzo MAC in formato esadecimale con due punti.

Pattern	Esempio	Descrizione
	Modello: `%{COMMONMAC:MacAddress}` Uscita: `{"MacAddress": "01:23:45:67:89:AB"}`	
MAC	Ingresso: `01:23:45:67:89:AB` Modello: `%{MAC:m1}` Uscita: `{"m1": "01:23:45:67:89:AB"}`	Corrisponde a qualsiasi modello CISCOMAC, WINDOWSMAC o COMMONMAC.

Pattern	Esempio	Descrizione
IPV6	Ingresso: `2001:db8:3333:4444:5555:6666:7777:8888` Modello: `%{IPV6:ip}` Uscita: `{"ip":"2001:db8:3333:4444:5555:6666:7777:8888"}`	Corrisponde IPv6 agli indirizzi, inclusi i moduli compressi e gli indirizzi IPv4 mappati IPv6 .
IPV4	Ingresso: `192.168.0.1` Modello: `%{IPV4:ip}` Uscita: `{"ip":"192.168.0.1"}`	Corrisponde IPv4 agli indirizzi.
IP	Ingresso: `192.168.0.1` Modello: `%{IP:ip}` Uscita: `{"ip":"192.168.0.1"}`	IPv6 Corrisponde agli indirizzi supportati dal IPV6pattern o IPv4 agli indirizzi supportati dal IPV4pattern.

Pattern	Esempio	Descrizione
HOSTNAME o HOST	Ingresso: `server-01.internal-network.local` Modello: `%{HOST:host}` Uscita: `{"host": "server-01.internal-network.local"}`	Corrisponde ai nomi di dominio, compresi i sottodomini.
IPORHOST	Ingresso: `2001:db8:3333:4444:5555:6666:7777:8888` Modello: `%{IPORHOST:ip}` Uscita: `{"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}`	Corrisponde a un nome host come supportato nel pattern HOSTNAME o a un indirizzo IP come supportato nel pattern IP.

Pattern	Esempio	Descrizione

HOSTPORT	Ingresso:
	`192.168.0.1:8080`
	
	Modello:
	`%{HOSTPORT:ip}`
	
	Uscita:
	`{"ip":"192.168.0.1:8080","PORT":"8080"}`
		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, e nell'output la porta viene memorizzata come «PORT».

URIHOST	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"}`
		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, acquisendo la porta come «porta», se presente.

Schemi di gruppi di percorsi

Pattern	Esempio	Descrizione
UNIXPATH	Ingresso: `/search?q=regex` Modello: `%{UNIXPATH:path}` Uscita: `{"path":"/search?q=regex"}`	Corrisponde ai percorsi degli URL, inclusi potenzialmente i parametri di query.
WINPATH	Ingresso: `C:\Users\John\Documents\file.txt` Modello: `%{WINPATH:path}` Uscita: `{"path":"C:\\Users\\John\\Documents\\file.txt"}`	Corrisponde ai percorsi dei file di Windows.
PATH	Ingresso: `/search?q=regex` Modello: `%{PATH:path}` Uscita: `{"path":"`	Corrisponde ai percorsi dei file URL o Windows.

Pattern	Esempio	Descrizione
	`/search?q=regex"}`	
TTY	Ingresso: `/dev/tty1` Modello: `%{TTY:path}` Uscita: `{"path":"/dev/tty1"}`	Corrisponde ai percorsi dei dispositivi Unix per terminali e pseudo-terminali.
URIPROTO	Ingresso: `web+transformer` Modello: `%{URIPROTO:protocol}` Uscita: `{"protocol":"web+transformer"}`	Corrisponde alle lettere, seguite facoltativamente da un carattere più (+) e lettere aggiuntive o caratteri più (+).

Pattern	Esempio	Descrizione
AURIPATH	Ingresso: `/category/sub-category/product_name` Modello: `%{URIPATH:path}` Uscita: `{"path":"/category/sub-category/product_name"}`	Corrisponde al componente del percorso di un URI.
URIPARAM	Ingresso: `?param1=value1&param2=value2` Modello: `%{URIPARAM:url}` Uscita: `{"url":"?param1=value1&param2=value2"}`	Corrisponde ai parametri della query URL.

Pattern	Esempio	Descrizione
URIPATHPARAM	Ingresso: `/category/sub-category/product?id=12345&color=red` Modello: `%{URIPATHPARAM:path}` Uscita: `{"path":"/category/sub-category/product?id=12345&color=red"}`	Corrisponde a un percorso URI seguito facoltativamente da parametri di query.

Pattern	Esempio	Descrizione
URI	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"}`	Corrisponde a un URI completo.

Schemi di data e ora

Pattern	Esempio	Descrizione
MESE	Ingresso: `Jan` Modello: `%{MONTH:month}`	Corrisponde ai nomi completi o abbreviati dei mesi in inglese come parole intere.

Pattern	Esempio	Descrizione
	Uscita: `{"month": "Jan"}` Ingresso: `January` Modello: `%{MONTH:month}` Uscita: `{"month": "January"}`	
NUMERO DI MESI	Ingresso: `5` Modello: `%{MONTHNUM:month}` Uscita: `{"month":"5"}` Ingresso: `05` Modello: `%{MONTHNUM:month}` Uscita: `{"month": "05"}`	Corrisponde ai numeri dei mesi da 1 a 12, con lo zero iniziale opzionale per i mesi a una cifra.

Pattern	Esempio	Descrizione
MONTHNUM2	Ingresso: `05` Modello: `%{MONTHNUM2:month}` Uscita: `{"month": "05"}`	Corrisponde ai numeri dei mesi a due cifre compresi tra 01 e 12.
MESE/GIORNO	Ingresso: `31` Modello: `%{MONTHDAY:monthDay}` Uscita: `{"monthDay":"31"}`	Corrisponde al giorno del mese compreso tra 1 e 31, con zero iniziale opzionale.
ANNO	Ingresso: `2024` Modello: `%{YEAR:year}` Uscita: `{"year":"2024"}` Ingresso: `24` Modello: `%{YEAR:year}` Uscita: `{"year":"24"}`	Corrisponde agli anni in formati a due o quattro cifre.

Pattern	Esempio	Descrizione
GIORNO	Ingresso: `Tuesday` Modello: `%{DAY:day}` Uscita: `{"day":"Tuesday"}`	Corrisponde ai nomi dei giorni completi o abbreviati.
ORA	Ingresso: `22` Modello: `%{HOUR:hour}` Uscita: `{"hour":"22"}`	Corrisponde all'ora nel formato a 24 ore con uno zero iniziale opzionale (0) 0-23.
MINUTO	Ingresso: `59` Modello: `%{MINUTE:min}` Uscita: `{"min":"59"}`	Corrisponde ai minuti (00-59).

Pattern	Esempio	Descrizione
SECOND	Ingresso: `3` Modello: `%{SECOND: second}` Uscita: `{"second" :"3"}` Ingresso: `30.5` Modello: `%{SECOND: fractionalSeconds}` Uscita: `{"minSec" :"30.5"}` Ingresso: `30:5` Modello: `%{SECOND: fractionalSeconds}` Uscita: `{"minSec" :"30:5"}`	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.

Pattern	Esempio	Descrizione
TIME	Ingresso: `09:45:32` Modello: `%{TIME:time}` Uscita: `{"time":"09:45:32"}`	Corrisponde a un formato dell'ora con ore, minuti e secondi, dove lo schema HOUR corrisponde all'ora, lo schema MINUTE corrisponde al minuto e lo schema SECOND corrisponde al secondo, in genere nel formato `(H)H:mm:(s)s` . I secondi includono i secondi intercalari (0) 0-60.
DATA_US	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"}`	Corrisponde a una data nel formato `(M)M/(d)d/(yy)yy` o `(M)M-(d)d-(yy)yy` , dove il modello MONTHNUM corrisponde al mese, lo schema MONTHDAY corrisponde al giorno e il modello YEAR corrisponde all'anno.

Pattern	Esempio	Descrizione
DATE_EU	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"}`	Corrisponde a una data nel formato di `(d)d/(M)M/(yy)yy` , o `(d)d-(M)M-(yy)yy` `(d)d.(M)M.(yy)yy`, dove il modello MONTHNUM corrisponde al mese, lo schema MONTHDAY corrisponde al giorno e il modello YEAR corrisponde all'anno.

Pattern	Esempio	Descrizione

DATE	Ingresso:
	`11/29/2024`
	
	Modello:
	`%{DATE:date}`
	
	Uscita:
	`{"date":"11/29/2024"}`
	
	Ingresso:
	`29.11.2024`
	
	Modello:
	`%{DATE:date}`
	
	Uscita:
	`{"date":"29.11.2024"}`
		Corrisponde a una data nel formato USA o nel formato UE, come nei modelli DATE_US e DATE_EU.

DATESTAMP	Ingresso:
	`29-11-2024 14:30:00`
	
	Modello:
	`%{DATESTAMP:dateTime}`
	
	Uscita:
	`{"dateTime":"29-11-2024 14:30:00"}`
		Corrisponde a uno schema DATE seguito da uno schema TIME, separato da uno spazio o da un trattino.

Pattern	Esempio	Descrizione
TZ	Ingresso: `PDT` Modello: `%{TZ:tz}` Uscita: `{"tz":"PDT"}`	Corrisponde alle abbreviazioni dei fusi orari comuni (PST, PDT, MST, MDT, CST CDT, EST, EDT, UTC).

Pattern	Esempio	Descrizione
ISO8601_FUSO ORARIO	Ingresso: `+05:30` Modello: `%{ISO8601 _TIMEZONE :tz}` Uscita: `{"tz":"+0 5:30"}` Ingresso: `-530` Modello: `%{ISO8601 _TIMEZONE :tz}` Uscita: `{"tz":"-530"}` Ingresso: `Z` Modello: `%{ISO8601 _TIMEZONE :tz}` Uscita: `{"tz":"Z"}`	Corrisponde all'offset UTC 'Z' o all'offset del fuso orario con due punti facoltativi in questo formato: `[+-](H)H(:)mm` w qui lo schema HOUR corrispon de all'ora e lo schema MINUTE corrisponde ai minuti.

Pattern	Esempio	Descrizione
ISO8601_SECONDI	Ingresso: `60` Modello: `%{ISO8601_SECOND:second}` Uscita: `{"second":"60"}`	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.

Pattern	Esempio	Descrizione
TIMESTAMP_ 01 ISO86	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`	Corrisponde al formato data/ora ISO86 01 `(yy)yy-(M)M-(d)dT(H)H:mm:((s)s)(Z|[+-](H)H:mm)` con secondi e fuso orario opzionali.

Pattern	Esempio	Descrizione
	Modello: `%{TIMESTAMP_ISO8601:timestamp}` Uscita: `{"timestamp":"23-5-1T1:25Z"}`	

Pattern	Esempio	Descrizione
DATESTAMP_RFC2822	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:timestamp}` Uscita: `{"dateTime":"Monday, 15 Jan 23 14:30:00 Z"}`	Corrisponde al RFC2822 formato data-ora: `Day, (d)d MonthName (yy)yy (H)H:mm:(s)s Z|[+-](H)H:mm` `Day`il modello viene utilizzato per abbinare un giorno intero o abbreviato, ad esempio «lunedì» o «lunedì». `MonthName` il pattern viene utilizzato per abbinare i nomi completi o abbreviati dei mesi inglesi, come «Jan» o «January». `Timezone`il pattern viene utilizzato per abbinare l'offset UTC (Z) o l'offset del fuso orario con due punti facoltativi.

Pattern	Esempio	Descrizione
DATESTAMP_OTHER	Ingresso: `Mon May 15 14:30:00 PST 2023` Modello: `%{DATESTAMP_OTHER:dateTime}` Uscita: `{"dateTime":"Mon May 15 14:30:00 PST 2023"}`	Corrisponde a data e ora nel seguente formato: `Day MonthName (d)d (H)H:mm:(s)s Timezone (yy)yy` Lo schema del giorno viene utilizzato per abbinare un giorno intero o abbreviato, ad esempio «lunedì» o «lunedì». Il MonthNamemodello viene utilizzato per abbinare i nomi completi o abbreviati dei mesi inglesi, ad esempio «Jan» o «January». `Day`il modello viene utilizzato per abbinare un giorno completo o abbreviato, ad esempio «lunedì» o «lunedì». `MonthName` il pattern viene utilizzato per abbinare i nomi completi o abbreviati dei mesi inglesi, come «Jan» o «January». `Timezone`il pattern corrispon de a qualsiasi fuso orario supportato dal pattern TZ grok.
DATESTAMP_EVENTLOG	Ingresso: `20230515143000` Modello: `%{DATESTAMP_EVENTLOG:dateTime}` Uscita: `{"dateTime":"20230515143000"}`	Corrisponde a un formato datetime compatto senza separatori: `(yy)yyMM(d)d(H)Hmm(s)s`

Schemi di gruppo logaritmici

Pattern	Esempio	Descrizione
LOGLEVEL	Ingresso: `INFO` Modello: `%{LOGLEVEL:logLevel}` Uscita: `{"logLevel":"INFO"}`	Corrisponde ai livelli di registro 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/Emergency/EMERG/EMERGENCY`
HTTPDATE	Ingresso: `23/Nov/2024:14:30:00 +0640` Modello: `%{HTTPDATE:date}` Uscita: `{"date":"23/Nov/2024:14:30:00 +0640"}`	Corrisponde al formato di data e ora spesso utilizzato nei file di registro. Formato: `(d)d/MonthName/(yy)yy:(H)H:mm:(s)s Timezone` `MonthName` corrisponde ai nomi completi o abbreviati dei mesi inglesi, ad esempio «Jan» o «January», e `Timezone` corrisponde allo schema INT grok.
SYSLOGTIMESTAMP	Ingresso: `Nov 29 14:30:00` Modello: `%{SYSLOGTIMESTAMP:dateTime}` Uscita: `{"dateTim`	Corrisponde al formato della data con`MonthName (d)d (H)H:mm:(s)s` . `MonthName` corrisponde ai nomi completi o abbreviati dei mesi in inglese, ad esempio «Jan» o «January»

Pattern	Esempio	Descrizione
	`e":"Nov 29 14:30:00"}`	
PROG	Ingresso: `user.profile/settings-page` Modello: `%{PROG:program}` Uscita: `{"program":"user.profile/settings-page"}`	Corrisponde al nome di un programma composto da una stringa di lettere, cifre, punti, trattini bassi, barra, segno di percentuale e trattino.

Pattern	Esempio	Descrizione
SYSLOGPROG	Ingresso: `user.profile/settings-page[1234]` Modello: `%{SYSLOGPROG:programWithId}` Uscita: `{"programWithId":"user.profile/settings-page[1234]","program":"user.profile/settings-page","pid":"1234"}`	Corrisponde al pattern PROG grok seguito facoltativamente da un ID di processo tra parentesi quadre.

Pattern	Esempio	Descrizione
SYSLOGHOST	Ingresso: `2001:db8:3333:4444:5555:6666:7777:8888` Modello: `%{SYSLOGHOST:ip}` Uscita: `{"ip":"2001:db8:3333:4444:5555:6666:7777:8888"}`	Corrisponde a un pattern HOST o IP.
STRUTTURA SYSLOG	Ingresso: `<13.6>` Modello: `%{SYSLOGFACILITY:syslog}` Uscita: `{"syslog":"<13.6>","facility":"13","priority":"6"}`	Corrisponde alla priorità syslog in formato decimale. Il valore deve essere racchiuso tra parentesi angolari (**<>**).

Schemi di gruppo logaritmici comuni

È possibile utilizzare modelli grok personalizzati predefiniti che possono essere utilizzati 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 il pattern GREEDYDATA o DATA.

Pattern	Descrizione	Limite di utilizzo all'interno del campo **`match`**
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 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:

```
{
  "remote_host": "127.0.0.1",
  "ident": "-",
  "auth_user": "-",
  "timestamp": "2023-08-03T12:34:56Z",
  "http_method": "GET",
  "request": "/page.html",
  "http_version": 1.1,
  "status_code": 200,
  "response_size": 1234
}
```

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:

```
{
  "remote_host": "192.168.1.100",
  "ident": "-",
  "auth_user": "Foo",
  "timestamp": "2023-08-03T12:34:56Z",
  "http_method": "GET",
  "request": "/account/login.html",
  "http_version": 1.1,
  "status_code": 200,
  "response_size": 42,
  "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"
}
```

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	Descrizione	Obbligatorio?	Predefinita	Limiti
source	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
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

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"
}
```

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	Descrizione	Obbligatorio?	Predefinita	Limiti
source	Percorso del campo nel registro dell'evento che verrà analizzato	No	`@message`	Lunghezza massima: 128. Profondità massima della chiave annidata: 3
destinazione	Il campo di destinazione in cui inserire le coppie chiave-valore estratte	No		Lunghezza massima: 128.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
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

Analizza WAF

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 registro 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 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 Amazon Route 53 Public Data Plane i log forniti da 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"
}
```

Processori di mutazione delle stringhe

lowerCaseString

Il `lowerCaseString` processore converte una stringa nella sua versione minuscola.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
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, da utilizzare `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	Descrizione	Obbligatorio?	Predefinita	Limiti
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, da utilizzare `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 divide un campo in un array utilizzando un carattere delimitante.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
voci	Matrice di voci. Ogni elemento dell'array deve contenere `source` `delimiter` campi.	Sì		Numero massimo di iscrizioni: 100
source	La chiave per dividere	Sì		Lunghezza massima: 128.
delimitatore	I caratteri separatori responsabili della divisione	Sì		Lunghezza massima: 1

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
```

```
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, da utilizzare `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"
    ]
  }
}
```

Stringa sostitutiva

Il `substituteString` processore confronta il valore di una chiave con un'espressione regolare e sostituisce tutte le corrispondenze con una stringa sostitutiva.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
voci	Matrice di voci. Ogni elemento dell'array deve contenere `source` to campi e. `from`	Sì		Numero massimo di iscrizioni: 10

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
source	La chiave 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. Per ulteriori informazioni, vedere Class Pattern sul sito Web di Oracle.	Sì		Lunghezza massima: 128.
in	La stringa da sostituire per ogni corrispondenza di `from`	Sì		Lunghezza massima: 128.

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key1": "[]",
        "inner_key2": "123-345-567"
    }
}
```

La configurazione del trasformatore è questa, da utilizzare `substituteString` con`parseJSON`:

```
[
   "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"
            }
        ]
    }
]
```

L'evento di registro trasformato sarebbe il seguente.

```
{
  "outer_key": {
    "inner_key1": "value1",
    "inner_key2": "xxx-xxx-xxx"
  }
}
```

TrimString

Il `trimString` processore rimuove gli spazi bianchi dall'inizio e dalla fine di una chiave.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
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, da utilizzare `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

Aggiungi chiavi

Utilizzate il `addKeys` processore per aggiungere nuove coppie chiave-valore all'evento di registro.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
voci	Matrice di voci. Ogni elemento dell'array può contenere `overwriteIfExists` campi `keyvalue`, e.	Sì		Numero massimo di iscrizioni: 5
Chiave	La chiave della nuova voce da aggiungere	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
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	

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, da utilizzare `addKeys` con`parseJSON`:

```
[
   "parseJSON": {},
   "addKeys": {
        "entries": [
            {
                "source": "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	Descrizione	Obbligatorio?	Predefinita	Limiti
Con chiavi	L'elenco delle chiavi da eliminare.	Sì		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, da utilizzare `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	Descrizione	Obbligatorio?	Predefinita	Limiti
voci	Matrice di voci. Ogni elemento dell'array può contenere `overwriteIfExists` campi `sourcetarget`, e.	Sì		Numero massimo di iscrizioni: 5
source	La chiave per muoversi	Sì		Lunghezza massima: 128. 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 `sutrue`, il valore esistente viene sovrascritto se `key` già presente nell'evento. Il valore predefinito è `false`.	No	false	

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, da utilizzare `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	Descrizione	Obbligatorio?	Predefinita	Limiti
voci	Matrice di voci. Ogni elemento dell'array può contenere `overwriteIfExists` campi `keytarget`, e.	Sì		Numero massimo di iscrizioni: 5

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
Chiave	La chiave per rinominare	Sì		Lunghezza massima: 128.
target	Il nuovo nome della chiave	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	

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, da utilizzare `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	Descrizione	Obbligatorio?	Predefinita	Limiti
voci	Matrice di voci. Ogni elemento dell'array può contenere `overwriteIfExists` campi `sourcetarget`, e.	Sì		Numero massimo di iscrizioni: 5
source	La chiave da copiare	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3
target	La chiave in cui copiare il valore	Sì		Lunghezza massima: 128. Profondità massima della chiave annidata: 3

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
overwriteIfExists	Se lo impostate su `true`, il valore esistente viene sovrascritto se `key` già presente nell'evento. Il valore predefinito è `false`.	No	false	

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configurazione del trasformatore è questa, da utilizzare `copyValue` con`parseJSON`:

```
[
   "parseJSON": {},
   "copyValue": {
        "entries": [
            {
                "source": "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	Descrizione	Obbligatorio?	Predefinita	Limiti
source	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	Descrizione	Obbligatorio?	Predefinita	Limiti
	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

Convertitore di tipo

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	Descrizione	Obbligatorio?	Predefinita	Limiti
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.

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
				Profondità massima della chiave annidata: 3
tipo	Il tipo in cui eseguire la conversione. I valori validi sono `integer`, `double`, `string` e `boolean`.	Sì		

Esempio

Prendiamo il seguente esempio di evento di registro:

```
{
    "name": "value",
    "status": "200"
}
```

La configurazione del trasformatore è questa, da utilizzare `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	Descrizione	Obbligatorio?	Predefinita	Limiti
source	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. 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.	No	UTC	Lunghezza minima:1

Campo	Descrizione	Obbligatorio?	Predefinita	Limiti
	Per un elenco dei valori possibili, consulta Java Supported Zone Ids and Offsets.			
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, da utilizzare `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.

Analizza con Amazon OpenSearch Service

CloudWatch Logs si integra con Amazon OpenSearch Service per consentirti di creare dashboard automatiche e curate che mostrano le metriche chiave che il OpenSearch Servizio ricava dai log forniti dai servizi. AWS Sono disponibili le seguenti dashboard:

- Una dashboard di Amazon VPC flow logs acquisisce i dati del flusso di rete per Amazon. VPC Ti aiuta ad analizzare il traffico di rete, rilevare schemi insoliti e monitorare l'utilizzo delle risorse. Le metriche chiave visualizzate includono quanto segue:
  - Flussi totali e accettazione e rifiuto di tali flussi
  - Modelli di traffico nel tempo
  - Un diagramma di Sankey che illustra il flusso di dati tra origine e destinazione IPs (top talker)
  - Primo per byte e IPs pacchetti trasferiti

  > **Note**
  >
  > Attualmente è supportato solo il formato dei campi della VPC versione 2.

- Una dashboard AWS WAF dei registri fornisce informazioni sul traffico web monitorato da AWS WAF. Questa dashboard ti aiuta a identificare i modelli di traffico, le richieste bloccate e le potenziali minacce provenienti da regioni specifiche oIPs. Le metriche chiave visualizzate includono quanto segue:
  - Richieste totali, inclusi i conteggi «ALLOW» e «BLOCK».
  - Cronologia delle richieste nel tempo, visualizzazione delle richieste consentite e bloccate.
  - Suddivisione delle richieste per ACL nome Web, richieste bloccate per regola di terminazione e origine. IPs
  - Una distribuzione geografica delle origini delle richieste.
  - Principali regole relative IPs ai client e alle terminazioni in base al numero di richieste.
- Una dashboard CloudTrail dei log fornisce una panoramica delle API attività all'interno dell' AWS ambiente utilizzando CloudTrail i log. È utile per monitorare le API attività, controllare le azioni e identificare potenziali problemi di sicurezza o conformità. Le metriche chiave visualizzate includono quanto segue:
  - Numero totale di eventi e cronologia degli eventi nel tempo
  - Una suddivisione degli eventi per accountIDs, categorie e regioni.

- PrincipaliAPIs, servizi e fonti IPs coinvolti nella generazione di eventi.
- Una tabella dei principali utenti che generano eventi, con informazioni dettagliate sugli account utente e conteggi degli eventi.

Le metriche visualizzate in queste dashboard curate derivano dall'analisi. Amazon OpenSearch Service

Prima di poter visualizzare questi dashboard, devi creare un IAM ruolo ed eseguire un'integrazione una tantum di Logs con. CloudWatch Amazon OpenSearch Service Questa integrazione unica configura Amazon OpenSearch Service le risorse necessarie per creare e renderizzare la dashboard. Ti verranno addebitati dei costi per i servizi utilizzati. OpenSearch Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

Puoi creare queste dashboard curate solo per i gruppi di log della Standard Log Class.

> **Important**
>
> Non utilizzare i trasformatori di log per i gruppi di log per i quali desideri creare dashboard di log venduti. La trasformazione degli eventi di registro farà sì che i dashboard abbiano dati vuoti.

Argomenti

Fase 1: Creare l'integrazione con Service OpenSearch

Il primo passo è creare l'integrazione con OpenSearch Service, che devi eseguire una sola volta. La creazione dell'integrazione creerà le seguenti risorse nel tuo account.

- Una raccolta OpenSearch Service di serie temporali senza elevata disponibilità.

Una raccolta è un insieme di indici di OpenSearch servizio che interagiscono per supportare un carico di lavoro.

- Due politiche di sicurezza per la raccolta. Una definisce il tipo di crittografia, che può essere con una AWS KMS chiave gestita dal cliente o una chiave di proprietà del servizio. L'altra politica definisce l'accesso alla rete, consentendo all'applicazione OpenSearch Service di accedere alla raccolta. Per ulteriori informazioni, consulta Encryption of data at rest for Amazon OpenSearch Service.
- Una politica di accesso ai dati del OpenSearch Servizio che definisce chi può accedere ai dati della raccolta.
- Un'origine dati OpenSearch Service Direct Query con CloudWatch Logs definiti come origine.
- Un'applicazione OpenSearch di servizio con il nome`aws-analytics`. L'applicazione verrà configurata per consentire la creazione di uno spazio di lavoro. Se esiste `aws-analytics` già un'applicazione denominata, verrà aggiornata per aggiungere questa raccolta come fonte di dati.
- Un'area OpenSearch di lavoro di servizio che ospiterà le dashboard e consentirà a tutti coloro a cui è stato concesso l'accesso di leggere dall'area di lavoro.

Argomenti

Autorizzazioni richieste

Per creare l'integrazione, devi accedere a un account con la IAM policy CloudWatchOpenSearchDashboardsFullAccessgestita o autorizzazioni equivalenti, mostrate qui. È inoltre necessario disporre di queste autorizzazioni per eliminare l'integrazione, creare, modificare ed eliminare i dashboard e aggiornare manualmente la dashboard.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "CloudWatchOpenSearchDashboardsIntegration",
            "Effect": "Allow",
            "Action": [
                "logs:ListIntegrations",
                "logs:GetIntegration",
```

```
                "logs:DeleteIntegration",
                "logs:PutIntegration",
                "logs:DescribeLogGroups",
                "opensearch:ApplicationAccessAll",
                "iam:ListRoles",
                "iam:ListUsers"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLogsOpensearchReadAPIs",
            "Effect": "Allow",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "es:ListApplications"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsOpensearchCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "opensearchservice.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsObservabilityCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
```

```
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
observability.aoss.amazonaws.com/AWSServiceRoleForAmazonOpenSearchServerless",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "observability.aoss.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionRequestAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateCollection"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsApplicationRequestAccess",
            "Effect": "Allow",
            "Action": [
                "es:CreateApplication"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/OpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
```

```
                    "aws:TagKeys": "OpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionResourceAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:DeleteCollection"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsApplicationResourceAccess",
            "Effect": "Allow",
            "Action": [
                "es:UpdateApplication",
                "es:GetApplication"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/OpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateSecurityPolicy",
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
```

```
                "aoss:DeleteSecurityPolicy",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAPIAccessAll",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsIndexPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:GetAccessPolicy",
                "aoss:CreateLifecyclePolicy",
                "aoss:DeleteLifecyclePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:index": "cloudwatch-logs-*",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
```

```
            "Sid": "CloudWatchLogsDQSRequestQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsStartDirectQueryAccess",
            "Effect": "Allow",
            "Action": [
                "opensearch:StartDirectQuery",
                "opensearch:GetDirectQuery"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*"
        },
        {
            "Sid": "CloudWatchLogsDQSResourceQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:GetDirectQueryDataSource",
                "es:DeleteDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
```

```
        {
            "Sid": "CloudWatchLogsPassRoleAccess",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService":
"directquery.opensearchservice.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAossTagsAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:TagResource"
            ],
            "Resource": "arn:aws:aoss:*:*:collection/*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsEsApplicationTagsAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddTags"
            ],
            "Resource": "arn:aws:opensearch:*:*:application/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/OpenSearchIntegration": [
```

```
                        "Dashboards"
                    ],
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "OpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsEsDataSourceTagsAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddTags"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ],
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        }
    ]
}
```

Crea l'integrazione

Usa questi passaggi per creare l'integrazione.

Per integrare CloudWatch Logs con Amazon OpenSearch Service

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, scegli Logs Insights, quindi scegli la OpenSearch scheda Analizza con.
3. Scegli Crea integrazione.
4. In Nome dell'integrazione, inserisci un nome per l'integrazione.

5. (Facoltativo) Per crittografare i dati scritti su OpenSearch Service Serverless, inserisci nella AWS KMS KMS chiave la chiave che desideri utilizzare. ARN ARN Per ulteriori informazioni, consulta Encryption at rest nella Amazon OpenSearch Service Developer Guide.
6. Per la conservazione dei dati, inserisci il periodo di tempo per cui desideri che gli indici dei dati del OpenSearch Servizio vengano conservati. Questo definisce anche il periodo di tempo massimo per il quale è possibile visualizzare i dati nei dashboard. La scelta di un periodo di conservazione dei dati più lungo comporterà costi di ricerca e indicizzazione aggiuntivi. Per ulteriori informazioni, consulta OpenSearch Service Serverless Pricing.

   Il periodo di conservazione massimo è di 30 giorni.

   La durata di conservazione dei dati verrà utilizzata anche per creare la politica del ciclo di vita della raccolta dei OpenSearch servizi.
7. Per IAM quanto riguarda il ruolo di scrittura nella OpenSearch raccolta, crea un nuovo IAM ruolo o seleziona un IAM ruolo esistente da utilizzare per scrivere nella raccolta OpenSearch Service.

   La creazione di un nuovo ruolo è il metodo più semplice e il ruolo verrà creato con le autorizzazioni necessarie.

   > **Note**
   >
   > Se crei un ruolo, avrà le autorizzazioni per la lettura da tutti i gruppi di log dell'account.

   Se desideri selezionare un ruolo esistente, dovrebbe avere le autorizzazioni elencate in. Autorizzazioni necessarie per l'integrazione In alternativa, puoi scegliere Usa un ruolo esistente e quindi nella sezione Verifica le autorizzazioni di accesso del ruolo selezionato puoi scegliere Crea ruolo. In questo modo puoi utilizzare le autorizzazioni elencate Autorizzazioni necessarie per l'integrazione come modello e modificarle. Ad esempio, se si desidera specificare un controllo più preciso dei gruppi di log.
8. Per IAMi ruoli e gli utenti che possono visualizzare i dashboard, è possibile selezionare come concedere l'accesso ai IAM ruoli e IAM agli utenti per l'accesso alla dashboard dei registri venduti:
   - Per limitare l'accesso alla dashboard solo ad alcuni utenti, scegli Seleziona IAM ruoli e utenti che possono visualizzare le dashboard, quindi nella casella di testo cerca e seleziona IAM i ruoli e IAM gli utenti a cui desideri concedere l'accesso.

- Per concedere l'accesso alla dashboard a tutti gli utenti, scegli Consenti a tutti i ruoli e agli utenti di questo account di visualizzare le dashboard.

> **Important**
>
> La selezione dei ruoli o degli utenti, o la selezione di tutti gli utenti, li aggiunge solo alla politica di accesso ai dati necessaria per accedere alla raccolta di OpenSearch servizi che memorizza i dati del dashboard. Per consentire loro di visualizzare i dashboard dei registri venduti, devi inoltre concedere a tali ruoli e utenti la CloudWatchOpenSearchDashboardAccess politica gestita. IAM

9. Scegli Crea integrazione

   La creazione dell'integrazione richiederà alcuni minuti.

Fase 2: Creare dashboard per i registri venduti

Dopo aver creato l'integrazione, puoi creare dashboard. Le dashboard sono disponibili per i log di VPC flusso, i log e i CloudTrail log di Amazon. AWS WAF

Per creare una dashboard di log venduta con metriche derivate dal servizio OpenSearch

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione a sinistra, scegli Logs Insights, quindi scegli la OpenSearch scheda Analizza con.
3. Seleziona Crea pannello di controllo.
4. Scegli il tipo di log per cui creare la dashboard AWS WAF, Amazon VPC flow logs o. CloudTrail
5. Inserisci un nome per la dashboard e, facoltativamente, inserisci una descrizione.
6. Per Frequenza di sincronizzazione dei dati, inserisci la frequenza con cui desideri che OpenSearch Service esegua le query CloudWatch in modo che le metriche e gli indici creati in OpenSearch Service possano essere sincronizzati e aggiornati con nuovi dati. OpenSearch Il servizio crea metriche e indici nei log per il rendering della dashboard.

   La scelta di un periodo di tempo più breve mantiene i dati più aggiornati e comporta costi più elevati.
7. Seleziona i gruppi di log da cui raccogliere i dati per questa dashboard. Assicurati di selezionare i gruppi di log che corrispondono al tipo di dashboard che stai creando.

Puoi utilizzare il pulsante Sfoglia i gruppi di log e l'opzione Visualizza esempi di log dai gruppi di log selezionati mentre effettui queste scelte, per assicurarti di ottenere i gruppi di log che desideri.

8. Seleziona Crea pannello di controllo.

   Inizialmente, la dashboard viene visualizzata senza dati. Dopo alcuni minuti, i dati verranno visualizzati nella dashboard. Quando i dati vengono visualizzati per la prima volta, si riferiranno agli ultimi 15 minuti di voci di registro.

Visualizza, modifica o elimina i dashboard dei log venduti

Visualizza i dashboard dei log venduti in Log or Service CloudWatch OpenSearch

Per poter visualizzare i dashboard, devi aver effettuato l'accesso a un IAM responsabile che dispone della politica. CloudWatchOpenSearchDashboardAccessIAM

Per visualizzare i dashboard dei log forniti

1. Apri la CloudWatch console all'indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel riquadro di navigazione a sinistra, scegli Logs Insights, quindi scegli la OpenSearch scheda Analizza con.
3. Scegli la dashboard nella casella delle OpenSearch dashboard.
4. (Facoltativo) In alto a destra, scegli Visualizza in OpenSearch.

   La console OpenSearch di servizio si apre e lì viene visualizzata la stessa dashboard. Nella console di OpenSearch servizio, puoi apportare modifiche alla dashboard e ai relativi widget e tali modifiche saranno visibili anche quando visualizzi la dashboard in CloudWatch Logs.

Concedi l'accesso alla visualizzazione della dashboard a IAM ruoli o utenti aggiuntivi IAM

Per concedere l'accesso a IAM principi aggiuntivi dopo aver creato l'integrazione, procedi nel seguente modo.

Per concedere l'accesso alla dashboard dei log forniti a IAM ruoli o utenti aggiuntivi

1. Modifica la politica di accesso ai dati per la raccolta per aggiungere questi ruoli o utenti. Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai dati per Amazon OpenSearch Service Serverless nella OpenSearch Service Developer Guide.
2. Concedili CloudWatchOpenSearchDashboardAccessa questi utenti. Per ulteriori informazioni sui contenuti di questa politica, vedereCloudWatchOpenSearchDashboardAccess.

Modifica la configurazione del pannello di controllo

Puoi modificare il nome, la descrizione e la frequenza di sincronizzazione dei dashboard dei log dei fornitori esistenti.

Per modificare un pannello di controllo dei registri venduti

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, scegli Logs Insights, quindi scegli la OpenSearch scheda Analizza con.
3. Scegli la dashboard nella casella delle OpenSearch dashboard.
4. Scegli Azioni, Modifica i dettagli della dashboard.
5. Apporta le modifiche, quindi scegli Conferma modifiche.

Elimina un pannello di controllo del registro venduto

È possibile eliminare un pannello di controllo dei registri venduti. In tal caso, la dashboard, le metriche e gli indici creati nella raccolta OpenSearch Service vengono tutti eliminati.

> **Note**
>
> Dopo aver eliminato una dashboard di log venduta, attendi almeno sei ore prima di provare a ricreare la stessa dashboard. Se non aspetti, la dashboard ricreata non funzionerà correttamente.

Per eliminare una dashboard di registro venduta

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione a sinistra, scegli Logs Insights, quindi scegli la OpenSearch scheda Analizza con.
3. Scegli la dashboard nella casella delle OpenSearch dashboard.
4. Scegli Operazioni > Elimina.
5. Conferma la tua decisione inserendo**`delete`**, quindi scegli Elimina.

Elimina tutti i log venduti, l'integrazione del pannello di controllo con Service OpenSearch

Puoi eliminare l'intera OpenSearch integrazione. Se lo fai, tutti i log venduti, i dashboard e i dati in essi visualizzati vengono eliminati.

> **Important**
>
> Per evitare costi continui, ti consigliamo vivamente di eliminare manualmente le seguenti risorse prima di eliminare l'integrazione. L'eliminazione dell'integrazione non elimina automaticamente queste risorse e, dopo aver eliminato l'integrazione, non sarà possibile accedere a queste risorse per eliminarle. Per trovare i nomi delle risorse da eliminare, consulta la procedura seguente.
>
> - La fonte dei dati
> - La collezione
> - La politica di accesso ai dati
> - La politica di crittografia
> - La politica di rete
> - La politica del ciclo di vita

Per eliminare l'intera integrazione del pannello di controllo dei registri venduti con Service OpenSearch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra scegliere Impostazioni.
3. Scegliere la scheda Log.
4. Nella sezione OpenSearch integrazione, scegli Elimina integrazione.

La schermata successiva mostra i nomi delle risorse del OpenSearch servizio che è necessario eliminare prima di eliminare l'integrazione.

5. Conferma la tua decisione inserendo**`delete`**, quindi scegli Elimina integrazione.

IAMpolitiche per gli utenti

CloudWatch Logs ha creato due IAM politiche CloudWatchOpenSearchDashboardsFullAccesse CloudWatchOpenSearchDashboardAccess. La tabella seguente elenca le azioni abilitate da ciascuna di queste politiche.

Azione	Policy IAM	Sono necessarie autorizza zioni aggiuntive
Crea integrazione	CloudWatchOpenSear chDashboardsFullAccess	
Eliminare l'integrazione	CloudWatchOpenSear chDashboardsFullAccess	
Crea dashboard	CloudWatchOpenSear chDashboardsFullAccess	
Modifica dashboard	CloudWatchOpenSear chDashboardsFullAccess	
Elimina dashboard	CloudWatchOpenSear chDashboardsFullAccess	
Aggiorna la dashboard usando Sincronizza ora	CloudWatchOpenSear chDashboardsFullAccess	
Visualizza l'integrazione nelle Impostazioni	CloudWatchOpenSear chDashboardAccess o CloudWatchOpenSear chDashboardsFullAccess	
Visualizza la dashboard	CloudWatchOpenSear chDashboardAccess o	Specificate il ruolo o l'utente quando create l'integrazione

Azione	Policy IAM	Sono necessarie autorizzazioni aggiuntive
	CloudWatchOpenSearchDashboardsFullAccess	oppure modificate la politica di accesso ai dati per la raccolta per aggiungere questi ruoli o utenti. Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai dati per Amazon OpenSearch Service Serverless nella OpenSearch Service Developer Guide.
Visualizza la dashboard nella console OpenSearch di servizio	CloudWatchOpenSearchDashboardAccess o CloudWatchOpenSearchDashboardsFullAccess	Specificate il ruolo o l'utente quando create l'integrazione oppure modificate la politica di accesso ai dati per la raccolta per aggiungere questi ruoli o utenti. Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai dati per Amazon OpenSearch Service Serverless nella OpenSearch Service Developer Guide.

Autorizzazioni necessarie per l'integrazione

Se crei un IAM ruolo da utilizzare per l'integrazione, invece di consentire a CloudWatch Logs di creare il ruolo, questa deve includere le seguenti autorizzazioni e criteri di fiducia. Per ulteriori informazioni su come creare un IAM ruolo, consulta Creare un ruolo per delegare le autorizzazioni a un servizio. AWS

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

```
    {
      "Sid": "CloudWatchLogsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:StartQuery",
        "logs:GetLogGroupFields",
        "logs:GetQueryResults"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "CloudWatchLogsDescribeLogGroupsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    },
    {
        "Sid": "AmazonOpenSearchCollectionAccess",
        "Effect": "Allow",
        "Action": [
            "aoss:APIAccessAll"
        ],
        "Resource": "*",
        "Condition": {
            "StringLike": {
                "aoss:collection": "cloudwatch-logs-*"
            }
        }
    }
  ]
}

//Trust Policy
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TrustPolicyForAmazonOpenSearchDirectQueryService",
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "directquery.opensearchservice.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:opensearch:us-
east-1:123456789012:datasource/cloudwatch_logs_*"
                }
            }
        }
    ]
}
```

Note

Il ruolo precedente consente l'accesso alla lettura da tutti i gruppi di log dell'account, per consentire la creazione di dashboard per qualsiasi account di registro, inclusi i gruppi di registro tra account. Se desideri limitare l'accesso a gruppi di log specifici e creare dashboard solo per quei gruppi di log, puoi aggiornare la prima dichiarazione di quella politica nel modo seguente:

```
{
      "Sid": "CloudWatchLogsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:StartQuery",
        "logs:GetLogGroupFields",
        "logs:GetQueryResults"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:123456789012:log-group:myLogGroup:*",
        "arn:aws:logs:us-east-1:123456789012:log-group:myLogGroup"
      ]
}
```

Creazione di parametri da log eventi mediante filtri

Puoi cercare e filtrare i dati di registro 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 registro quando vengono inviati ai registri. CloudWatch 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 un gruppo di log con un abbonamento utilizza la trasformazione dei log, lo schema di filtro viene applicato alle versioni trasformate degli eventi di registro. 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, consultateClassi di registro.

Puoi utilizzare qualsiasi tipo di CloudWatch statistica, incluse le statistiche sui percentili, durante la visualizzazione di queste metriche o l'impostazione degli 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. I risultati filtrati restituiscono le prime 50 righe, che non verranno visualizzate se il timestamp sui risultati filtrati è antecedente al tempo di creazione dei parametri.

Indice

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 un parametro, ogni volta che una coppia nome/valore unica viene estratta dai log, crei una nuova variante di detto parametro.

Modello di filtro

Una descrizione simbolica di come CloudWatch Logs deve 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 su cui devono essere pubblicate le informazioni di registro monitorate. Ad esempio, puoi pubblicare su una metrica chiamata. 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 CloudWatch query di 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, consultaCloudWatch 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 metrica Logs. CloudWatch Puoi definire i termini e i modelli da cercare nei dati di registro man mano che vengono inviati ai

registri. 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, MetricDatumconsulta Amazon CloudWatch API Reference.

Argomenti

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 disomogenee. 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 registro delimitati da spazi. Le dimensioni sono coppie nome/valore e sono disponibili solo per modelli di filtro JSON e 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' CloudWatch utente di Amazon
- 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 aiutare a prevenire addebiti accidentali elevati, Amazon potrebbe disabilitare un filtro di parametri se genera 1000 coppie nome/valore diverse per le dimensioni specificate entro 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 monitorare gli addebiti stimati. AWS

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 Creare un gruppo di log 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 errate. 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) 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.
7. Scegli Successivo e poi inserisci un nome per il filtro.
8. In Dettagli metrici, per Metric namespace, 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.
9. Per Metric name (Nome parametro) inserisci un nome per il nuovo parametro.
10. 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`.
11. (Facoltativo) Per Unit (Unità), seleziona un'unità da assegnare al parametro. Se non specifichi un'unità, l'unità viene impostata come `None`.
12. (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.

13. 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 MyApp /access.log per creare la metrica nello spazio dei nomi EventCount . 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 il 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 il 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 /message.log. MyApp CloudWatch Logs pubblica un punto

dati nella metrica CloudWatch personalizzata ErrorCount nello spazio dei nomi MyApp/message.log con un valore di «1" per ogni evento contenente Error. Se nessun evento contiene la parola Error, il valore pubblicato è 0. Quando rappresenti graficamente questi dati nella console CloudWatch, assicurati di utilizzare la statistica di 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 il 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

Utilizzando CloudWatch Logs, è possibile monitorare quante volte i 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
- Timestamp
- 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. Scegli Avanti, quindi in Nome filtro digita 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 Nome della metrica, inserisci HTTP4xxErrori.
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 il 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 il 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}'
```

È possibile utilizzare i seguenti dati nelle put-log-event chiamate per testare 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 Delete (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
```

Elaborazione in tempo reale dei dati di log con le sottoscrizioni

Puoi utilizzare gli abbonamenti per accedere a un feed di eventi di log in tempo reale da CloudWatch Logs e distribuirlo ad altri servizi come uno stream Amazon Kinesis, uno stream Amazon Data Firehose AWS Lambda o 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.

Per iniziare la sottoscrizione a log eventi, crea la risorsa ricevente, ad esempio un flusso Kinesis Data Streams, in cui verranno distribuiti gli eventi. Un filtro di sottoscrizione definisce lo schema di filtro da utilizzare per filtrare gli eventi di registro 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 registro. 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 può essere associati fino a due filtri di sottoscrizione.

> **Note**
>
> 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 l'invio 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 In questi casi il filtro di sottoscrizione è disabilitato per un massimo di 10 minuti, quindi CloudWatch Logs riprova a inviare i registri alla destinazione. Durante questo periodo di disabilitazione, i log vengono ignorati.

CloudWatch Logs produce anche CloudWatch metriche sull'inoltro degli eventi di registro agli abbonamenti. 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 OpenSearch Service. 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, consultateClassi 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.

Indice

Concetti

Ogni filtro sottoscrizione è formato dai seguenti elementi chiave.

Modello di filtro

Una descrizione simbolica di come CloudWatch Logs deve interpretare i dati in ogni evento di registro, insieme a espressioni di filtro che limitano ciò che viene consegnato 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) del flusso Kinesis Data Streams, dello stream 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 i log possono ottenere le autorizzazioni necessarie dalle impostazioni di controllo degli accessi sulla funzione Lambda stessa.

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 gli abbonamenti 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 impedire loop di log infiniti. Per ulteriori informazioni sul problema del loop di log infinito, vedereRegistra la prevenzione della ricorsione.

I criteri di selezione hanno un limite di dimensione di 25 KB.

Registra i filtri di abbonamento a livello di gruppo

Puoi utilizzare un filtro di abbonamento con Amazon Kinesis Data AWS Lambda Streams o Amazon Data Firehose. I log inviati a un servizio tramite un filtro di sottoscrizione sono codificati in base64 e compressi con il formato gzip. Questa sezione fornisce esempi che è possibile seguire per creare un filtro di sottoscrizione CloudWatch Logs che invii i dati di registro a Firehose, Lambda e Kinesis Data Streams.

> **Note**
>
> Se desideri effettuare una ricerca nei dati di registro, consulta Filter and pattern syntax.

Esempi

- Esempio 1: filtri di sottoscrizione con Kinesis Data Streams
- Esempio 2: filtri di abbonamento con AWS Lambda
- Esempio 3: filtri di abbonamento con Amazon Data Firehose

Esempio 1: filtri di sottoscrizione con Kinesis Data Streams

L'esempio seguente associa un filtro di sottoscrizione a un gruppo di log contenente AWS CloudTrail eventi. Il filtro di sottoscrizione invia ogni attività registrata effettuata dalle AWS credenziali «Root» a un flusso in Kinesis Data Streams chiamato "». RootAccess Per ulteriori informazioni su come inviare AWS CloudTrail eventi ai registri, consulta Invio CloudTrail di eventi ai CloudWatch registri nella 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:

- `random`Specificare `distribution` quando si crea il filtro di sottoscrizione con PutSubscriptionFiltero. put-subscription-filter Per impostazione predefinita, la distribuzione del filtro di flusso avviene per flusso di log 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 i quali CloudWatch 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 on demand per il tuo flusso in Kinesis Data Streams. La modalità on demand si adatta istantaneamente ai carichi di lavoro siano essi aumentati o diminuiti. Per ulteriori informazioni sulla modalità di capacità on demand, consulta Modalità on demand.
- Limita il modello di filtro dell' CloudWatch abbonamento in modo che corrisponda alla capacità del tuo stream in Kinesis Data Streams. Se invii una quantità eccessiva di dati al flusso, potrebbe essere necessario ridurre le dimensioni del filtro o modificarne i criteri.

Creare un filtro di sottoscrizione per 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. È possibile utilizzare il seguente comando Kinesis Data Streams describe-stream per controllare il. StreamDescription StreamStatusproprietà. Inoltre, prendi nota del valore StreamDescription.StreamArn, poiché ti servirà in un passaggio successivo:

```
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 l'autorizzazione a 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": {
           "StringLike": { "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 valore Role.Arn restituito, poiché ne avrai bisogno anche in una fase successiva:

```
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 quali azioni 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 politica 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 nello 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 di abbonamento, CloudWatch Logs inoltra tutti gli eventi di registro in entrata che corrispondono al modello di filtro allo stream. Puoi verificare che questo stia effettivamente avvenendo acquisendo un'iteratore di partizione di Kinesis Data Streams e utilizzando il comando Kinesis get-records per recuperare alcuni log di 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"
```

Potresti dover effettuare questa chiamata più volte prima che Kinesis Data Streams inizi a restituire dati.

In genere si visualizzerà una risposta con una matrice di record. L'attributo Dati in un record Kinesis Data Streams usa la codifica base64 e la compressione nel 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 Kinesis Data Streams 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 2: filtri di abbonamento con AWS Lambda

In questo esempio, creerai un filtro di sottoscrizione 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 helloWorld.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. Concedi a CloudWatch Logs il permesso 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 AWS dell'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 flusso di distribuzione di Amazon Data Firehose. I dati inviati dai CloudWatch log ad Amazon Data Firehose sono già compressi con la compressione gzip di livello 6, quindi non è necessario utilizzare la compressione all'interno del flusso di distribuzione Firehose. È quindi possibile utilizzare la funzionalità di decompressione di Firehose per decomprimere automaticamente i registri. Per ulteriori informazioni, vedere Send CloudWatch Logs to Firehose.

> **Note**
>
> Prima di creare lo stream Firehose, calcolate il volume di dati di registro che verranno generati. Assicurati 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 del 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 i log. 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 valore Role.Arn restituito, poiché ne avrai bisogno anche 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 Firehose può eseguire 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. Associate la politica delle autorizzazioni al ruolo utilizzando il seguente comando: put-role-policy

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Crea un flusso di distribuzione Firehose di destinazione come segue, sostituendo i valori segnaposto per ROLearn e bucketArn con il ruolo e il bucket che hai creato: ARNs

```
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 YYYY/MM/DD/HH UTC per gli oggetti Amazon S3 consegnati. 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). È possibile utilizzare il describe-delivery-streamcomando Firehose per controllare il. DeliveryStreamDescription DeliveryStreamStatusproprietà. Inoltre, nota il DeliveryStreamDescription. DeliveryStreamValore ARN, in quanto sarà necessario 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 concede a CloudWatch Logs l'autorizzazione a inserire i 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": {
         "StringLike": {
             "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. Annota il valore Role.Arn restituito, poiché ne avrai bisogno anche in una fase successiva:

```
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 quali azioni 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 politica delle autorizzazioni al ruolo utilizzando il 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 sottoscrizione 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 configurato il filtro di abbonamento, CloudWatch Logs inoltrerà tutti gli eventi di log in entrata che corrispondono allo schema di filtro al flusso di distribuzione di Amazon Data Firehose. I tuoi dati inizieranno a comparire in Amazon S3 in base all'intervallo di tempo impostato nel flusso di distribuzione 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
```

Filtri di abbonamento a livello di account

> **Important**
>
> Esiste il rischio di creare un ciclo ricorsivo infinito con i filtri di abbonamento, che può portare a un forte aumento della fatturazione di importazione se non viene risolto il problema. Per mitigare questo rischio, si consiglia di utilizzare criteri di selezione nei filtri di abbonamento a livello di account per escludere i gruppi di log che acquisiscono i dati di registro dalle risorse che fanno parte del flusso di lavoro di distribuzione dell'abbonamento. Per ulteriori informazioni su questo problema e per determinare quali gruppi di log escludere, vedere. Registra la prevenzione della ricorsione

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 AWS Lambda Streams o Amazon Data Firehose. I log inviati a un servizio tramite una

politica di abbonamento a livello di account sono codificati in base64 e compressi con il formato gzip. Questa sezione fornisce esempi che puoi seguire per creare un abbonamento a livello di account per Kinesis Data Streams, Lambda e Firehose.

> **Note**
>
> Per visualizzare un elenco di tutte le politiche di filtro degli abbonamenti presenti nel tuo account, usa il `describe-account-policies` comando con il valore per il parametro. `SUBSCRIPTION_FILTER_POLICY` `--policy-type` Per ulteriori informazioni, vedere describe-account-policies¶.

Esempi

- Esempio 1: filtri di sottoscrizione con Kinesis Data Streams
- Esempio 2: filtri di abbonamento con AWS Lambda
- Esempio 3: filtri di abbonamento con Amazon Data Firehose

Esempio 1: filtri di sottoscrizione con Kinesis Data Streams

Prima di creare un flusso di dati Kinesis Data Streams da utilizzare con una politica 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 ha abbastanza shard, viene limitato. Per ulteriori informazioni sui limiti del volume dello stream, consulta Quote e limiti nella documentazione di Kinesis Data Streams.

> **Warning**
>
> Poiché gli eventi di registro 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 Kinesis Data Streams CloudWatch con le metriche. Questo ti aiuta a identificare le limitazioni e ad adattare 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 on demand per il tuo flusso in Kinesis Data Streams. La modalità on demand si adatta istantaneamente ai carichi di lavoro siano essi aumentati o diminuiti. Per ulteriori informazioni, consulta la modalità On-demand.
- Limita il modello di filtro dell'abbonamento CloudWatch Logs in modo che corrisponda alla capacità del tuo stream in 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 politica di sottoscrizione a livello di account per inoltrare tutti gli eventi di registro a uno stream in Kinesis Data Streams. Lo schema di filtro abbina tutti gli eventi di registro al testo `Test` e li inoltra allo stream in Kinesis Data Streams.

Per creare una politica di sottoscrizione a livello di account per 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 che lo stream diventi attivo. Puoi verificare se lo stream è attivo utilizzando il comando describe-stream per controllare il. StreamDescription StreamStatusproprietà.

```
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 l'autorizzazione a 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": {
        "StringLike": { "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 valore Role.Arn restituito, poiché ne avrai bisogno anche in una fase successiva:

```
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 quali azioni 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/TestStream"
    }
  ]
}
```

6. Associa la politica delle autorizzazioni al ruolo utilizzando il seguente put-role-policycomando:

```
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 nello stato Attivo e hai creato il ruolo IAM, puoi creare la politica di filtro di sottoscrizione CloudWatch Logs. La policy avvia immediatamente il flusso di dati di registro

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` and`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 di abbonamento, CloudWatch Logs inoltra tutti gli eventi di registro in entrata che corrispondono al modello di filtro e ai criteri di selezione allo stream.

   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 per determinare quali gruppi di log escludere, vedere. Registra la prevenzione della ricorsione Attualmente, NOT IN è l'unico operatore supportato per`selection-criteria`.

   È possibile verificare il flusso di eventi di registro utilizzando un iteratore di shard Kinesis Data Streams e utilizzando il `get-records` comando Kinesis Data Streams per recuperare alcuni record 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 Kinesis Data Streams inizi a restituire dati.

In genere si visualizzerà una risposta con una matrice di record. L'attributo Dati in un record Kinesis Data Streams usa la codifica base64 e la compressione nel 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 Kinesis Data Streams di tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

owner

L' AWS ID dell'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 è stata applicata la politica. «ACCOUNT_LEVEL_POLICY» è il criterio 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 degli abbonamenti a livello di account CloudWatch Logs che invia i dati di registro alla tua funzione. AWS Lambda

> **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 degli abbonamenti 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 helloWorld.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. Concedi a CloudWatch Logs il permesso di eseguire la tua 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 degli abbonamenti 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 and. `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 di abbonamento, CloudWatch Logs inoltra tutti gli eventi di registro in entrata che corrispondono al modello di filtro e ai criteri di selezione allo stream.

   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 per determinare quali gruppi di log escludere, vedere. Registra la prevenzione della ricorsione 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 sottoscrizione a livello di account non verrà applicato al gruppo di log della funzione Lambda di destinazione. Questo serve a prevenire una ricorsione infinita dei log che può portare a un aumento della fatturazione di importazione. Per ulteriori informazioni su questo problema, vedere. Registra la prevenzione della ricorsione

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 Kinesis Data Streams di tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

owner

L' AWS ID dell'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 è stata applicata la politica. «ACCOUNT_LEVEL_POLICY» è il criterio di filtro degli abbonamenti a livello `policyLevel` di account.

Esempio 3: filtri di abbonamento con Amazon Data Firehose

In questo esempio, creerai una politica di filtro di abbonamento CloudWatch Logs a livello di account che invia gli eventi di registro in entrata che corrispondono ai filtri definiti al flusso di distribuzione di Amazon Data Firehose. I dati inviati dai CloudWatch log ad Amazon Data Firehose sono già compressi con la compressione gzip di livello 6, quindi non è necessario utilizzare la compressione all'interno del flusso di distribuzione Firehose. È quindi possibile utilizzare la funzionalità di decompressione di Firehose per decomprimere automaticamente i registri. Per ulteriori informazioni, vedere Writing to Kinesis Data CloudWatch Firehose Using Logs.

> **Warning**
>
> Prima di creare lo stream Firehose, calcolate il volume di dati di registro che verranno generati. Assicurati 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 del 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 i log. 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 valore Role.Arn restituito, poiché ti servirà in un passaggio successivo:

```
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 Firehose può eseguire 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. Associate la politica delle autorizzazioni al ruolo utilizzando il seguente comando: put-role-policy

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Crea un flusso di distribuzione Firehose di destinazione come segue, sostituendo i valori segnaposto per ROLearn e bucketArn con il ruolo e il bucket che hai creato: ARNs

```
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 YYYY/MM/DD/HH UTC per gli oggetti Amazon S3 consegnati. 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 che lo stream diventi attivo. È possibile utilizzare il describe-delivery-streamcomando Firehose per controllare il. DeliveryStreamDescription DeliveryStreamStatusproprietà. Inoltre, nota il DeliveryStreamDescription. DeliveryStreamValore ARN, in quanto sarà necessario 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 concede a CloudWatch Logs l'autorizzazione a inserire i 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": {
            "StringLike": {
                "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 valore Role.Arn 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 quali azioni i CloudWatch log possono 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 politica delle autorizzazioni al ruolo utilizzando il 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 policy di filtro di sottoscrizione a livello di account 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 configurato il filtro di abbonamento, CloudWatch Logs inoltra gli eventi di log in entrata che corrispondono al modello di filtro al flusso di distribuzione 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 dei log da un filtro di sottoscrizione. Per ulteriori informazioni su questo problema e per determinare quali gruppi di log escludere, vedere. Registra la prevenzione della ricorsione 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 impostato nel flusso di distribuzione 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
```

Abbonamenti tra più account e più regioni

Puoi collaborare con il proprietario di un altro AWS account e ricevere i relativi eventi di registro sulle tue AWS risorse, ad esempio uno stream Amazon Kinesis o Amazon Data Firehose (questa operazione è nota come condivisione di dati tra account). Ad esempio, i dati di questo registro degli eventi possono essere letti da un flusso Kinesis Data Streams o Firehose centralizzato 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 di dati di evento in tempo reale tra questi account può

essere assemblato e distribuito ai gruppi di sicurezza di informazioni che possono utilizzare Kinesis Data Streams per collegare i dati ai propri sistemi di analisi di sicurezza esistenti.

> **Note**
>
> Il gruppo di log e la destinazione devono trovarsi nella stessa regione. AWS Tuttavia, la AWS risorsa a cui punta la destinazione può trovarsi in una regione diversa. Negli esempi riportati nelle sezioni seguenti, tutte le risorse specifiche della regione vengono create negli Stati Uniti orientali (Virginia settentrionale).

Argomenti

Condivisione dei dati di registro tra account e aree geografiche tramite 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:

- Registra il mittente dei dati: ottiene 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 illustrate nel resto di questa sezione, il mittente dei dati di registro viene visualizzato con un numero di account fittizio pari a 1111 AWS .

  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 registro: imposta una destinazione che incapsula un flusso Kinesis Data Streams e CloudWatch fa sapere a Logs che il destinatario desidera ricevere i dati di registro. Il destinatario quindi condivide le informazioni su questa destinazione con il mittente. Nelle procedure illustrate nel resto di questa sezione, il destinatario dei dati di registro viene mostrato con un numero di account fittizio pari a 9999. AWS

Per iniziare a ricevere gli 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 concede a CloudWatch 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 regione. AWS Tuttavia, la risorsa AWS a cui punta la destinazione può trovarsi in una Regione diversa. Negli esempi delle sezioni seguenti, tutte le risorse specifiche della Regione vengono create in Stati Uniti orientali (Virginia settentrionale).

Argomenti

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

> **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 mittente AWS dei dati di registro è 1111.

Questo esempio crea una destinazione utilizzando un flusso Kinesis Data RecipientStream Streams chiamato e un ruolo CloudWatch che consente a 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 un flusso di destinazione in 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 usare il comando aws kinesis describe-stream per controllare. StreamDescription StreamStatusproprietà. Inoltre, prendi nota del valore StreamDescription.StreamArn perché lo passerai a Logs in un secondo CloudWatch momento:

```
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":
 "49551135218688818456679503831981458784591352702 18EXAMPLE"
        }
      }
    ]
  }
}
```

Potrebbero essere necessari uno o due minuti perché il flusso sia in stato attivo.

3. Crea il ruolo IAM che concede a CloudWatch Logs l'autorizzazione a inserire dati nel tuo stream. Per prima cosa, devi creare una politica di fiducia in un file TrustPolicyFor~/ CWL.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": {
            "StringLike": {
                "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 valore Role.Arn restituito perché verrà passato anche a Logs in un secondo momento: CloudWatch

```
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 i CloudWatch log possono eseguire sul tuo account. Innanzitutto, usa un editor di testo per creare una politica di autorizzazioni in un file ~/ CWL.json: PermissionsFor

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Associa la politica 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 nello stato 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 DestinationArnciò 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 politica deve specificare i log: PutSubscriptionFilter action e concede l'autorizzazione all'account mittente di 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.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region: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.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" :
 "arn:aws:logs:region: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 effettuare chiamate PutSubscriptionFilterverso la destinazione con ARN arn:aws:logs ::9999:destination:testDestination. *region* Qualsiasi tentativo di chiamata 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 la procedura riportata di seguito. 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 descritti in questa sezione creano un ruolo IAM, che CloudWatch può presupporre e verificare se l'account mittente è autorizzato a creare un filtro di sottoscrizione 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.

Creazione del ruolo IAM necessario per le sottoscrizioni del log tra più account utilizzando 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.

Passaggio 3: aggiunta/convalida delle autorizzazioni IAM per la destinazione multi-account

In base alla AWS logica di valutazione dei criteri tra account, per accedere a qualsiasi risorsa tra account (ad esempio 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 diversi. Per ulteriori informazioni sulla logica di valutazione delle policy, consulta la pagina Cross-account policy evaluation logic.

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 log AWS .

   ```
   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 o. *RoleName* *UserName*

2. AWS Management Console 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. Le seguenti policy di esempio mostrano le autorizzazioni suggerite.

   La seguente politica fornisce le autorizzazioni per creare un filtro di sottoscrizione su qualsiasi risorsa di destinazione solo in un singolo AWS account, account123456789012:

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "Allow subscription filters on any resource in one specific
    account",
               "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 sottoscrizione solo su una risorsa di destinazione specifica denominata `sampleDestination` in AWS account singolo, account: `123456789012`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "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 consegnata alla destinazione creata in precedenza. Tale destinazione incapsula un flusso chiamato "». RecipientStream

Il resto dei passaggi descritti nelle sezioni seguenti presuppone che l'utente abbia seguito le istruzioni riportate in Invio di CloudTrail eventi ai CloudWatch registri nella Guida per l'AWS CloudTrail utente e abbia creato un gruppo di log contenente gli eventi. CloudTrail 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 Passaggio 3: aggiunta/convalida delle autorizzazioni IAM per la destinazione multi-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 Kinesis Data Streams che si trova in una regione diversa.

Convalida del flusso dei log eventi

Dopo aver creato il filtro di sottoscrizione, CloudWatch Logs inoltra tutti gli eventi di registro in entrata che corrispondono al modello di filtro allo stream incapsulato nel flusso di destinazione denominato "». RecipientStream Il proprietario della destinazione può verificare che ciò stia accadendo utilizzando il get-shard-iterator comando aws kinesis per acquisire uno shard Kinesis Data Streams e utilizzando il comando aws kinesis get-records per recuperare alcuni record 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**
>
> Potresti dover rieseguire il comando get-records più volte prima che il flusso Kinesis Data Streams inizi a restituire dati.

Dovresti visualizzare una risposta con un array di log di Kinesis Data Streams. L'attributo dati nel log di Kinesis Data Streams usa la compressione nel formato gzip e la codifica 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

AWS L'ID dell'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 Kinesis Data Streams 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 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 politica 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 ~/ .json: NewAccessPolicy

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
```

```
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Chiama PutDestinationPolicyper associare la politica definita nel NewAccessPolicyfile.json 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 dai servizi AWS. 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 sia autorizzato a 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"
```

Per aggiornare un filtro di sottoscrizione e iniziare a utilizzare l'organizzazione IDs per le autorizzazioni di registro tra account diversi

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.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region: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 un AWS account specifico IDs, includi il `force` parametro.

> **Warning**
>
> Se utilizzi i log inviati da un AWS servizio elencato inAbilita la registrazione dai servizi AWS, prima di eseguire questo passaggio devi 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
```

Condivisione dei dati di registro tra più account 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: ottiene 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 illustrate nel resto di questa sezione, il mittente dei dati di registro viene visualizzato con un numero di account fittizio pari a 1111 AWS .
- Destinatario dei dati di registro: imposta una destinazione che incapsula un flusso Kinesis Data Streams e CloudWatch fa sapere a Logs che il destinatario desidera ricevere i dati di registro. 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 registro viene mostrato con un numero di account fittizio di 222222222222. AWS

L'esempio in questa sezione utilizza un flusso di distribuzione Firehose con storage Amazon S3. È inoltre possibile configurare i flussi di distribuzione di Firehose con impostazioni diverse. Per ulteriori informazioni, vedere Creating a Firehose Delivery Stream.

> **Note**
>
> Il gruppo di log e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la risorsa AWS 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 distribuzione tra regioni.

Argomenti

Fase 1: Creare un flusso di distribuzione Firehose

> **Important**
>
> Prima di completare i seguenti passaggi, è necessario utilizzare una policy di accesso, in modo che Firehose possa accedere al 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 distribuzione 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 l'autorizzazione a 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 sul 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. Immettete il seguente comando per creare il flusso di distribuzione di Firehose. Sostituisci *my-role-arn* e *amzn-s3-demo-bucket2-arn* con i valori corretti per la tua implementazione.

```
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 distribuzione Firehose diventi attivo. È possibile utilizzare il seguente comando per controllare ilStreamDescription. StreamStatusproprietà.

   ```
   aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
   ```

   Inoltre, prendi nota del DeliveryStreamDescription. DeliveryStreamValore ARN, perché sarà necessario 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 l'autorizzazione a inserire dati nel tuo flusso Firehose. Per prima cosa, devi creare una policy di fiducia in un file TrustPolicyFor~/ CWL.json. Utilizza un editor di testo per creare questa policy. Per ulteriori informazioni sugli endpoint di Amazon CloudWatch Logs, consulta Endpoints e quote di Amazon CloudWatch Logs.

   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": {
               "StringLike": {
                   "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": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Crea una politica di autorizzazioni per definire quali azioni CloudWatch Logs può eseguire sul tuo account. Innanzitutto, usa un editor di testo per creare una politica di autorizzazioni in un file ~/ CWL.json: PermissionsFor

```
{
    "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 nello stato attivo e hai creato il ruolo IAM, puoi creare la destinazione CloudWatch 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:

```
{
  "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 politica deve specificare i log: PutSubscriptionFilter action per accedere alla destinazione. Gli utenti con più account utilizzeranno l'PutSubscriptionFilterazione per inviare gli eventi di registro alla destinazione:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Passaggio 3: aggiunta/convalida delle autorizzazioni IAM per la destinazione multi-account

In base alla AWS logica di valutazione dei criteri tra account, per accedere a qualsiasi risorsa tra account (ad esempio 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 diversi. Per ulteriori informazioni sulla logica di valutazione delle policy, consulta la pagina Cross-account policy evaluation logic.

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 log AWS .

```
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 o. *RoleName* *UserName*

2. AWS Management Console 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. Le seguenti policy di esempio mostrano le autorizzazioni suggerite.

   La seguente politica fornisce le autorizzazioni per creare un filtro di sottoscrizione su qualsiasi risorsa di destinazione solo in un singolo AWS account, account`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on any resource in one specific
 account",
            "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 sottoscrizione solo su una risorsa di destinazione specifica denominata `sampleDestination` in AWS account singolo, account: `123456789012`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "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

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 consegnata alla destinazione creata in precedenza. Per ulteriori informazioni su come inviare AWS CloudTrail eventi ai CloudWatch registri, vedere Invio di CloudTrail eventi ai CloudWatch registri nella Guida per l'utente.AWS CloudTrail

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 Passaggio 3: aggiunta/convalida delle autorizzazioni IAM per la destinazione multi-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 un flusso Firehose che si trova in una regione diversa.

Convalida del flusso dei log eventi

Dopo aver creato il filtro di sottoscrizione, CloudWatch Logs inoltra tutti gli eventi di registro in entrata che corrispondono allo schema di filtro al flusso di distribuzione di Firehose. I dati iniziano a comparire nel bucket Amazon S3 in base all'intervallo di tempo impostato nel flusso di distribuzione di 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 politica 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 ~/ .json: NewAccessPolicy

```
{
  "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. Usa il seguente comando per associare la politica definita nel NewAccessPolicyfile.json 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.

Abbonamenti a livello di account interregionali con 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:

- Registra il mittente dei dati: ottiene 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 illustrate nel resto di questa sezione, il mittente dei dati di registro viene visualizzato con un numero di account fittizio pari a 1111 AWS .

  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 registro: imposta una destinazione che incapsula un flusso Kinesis Data Streams e CloudWatch fa sapere a Logs che il destinatario desidera ricevere i dati di registro. Il destinatario quindi condivide le informazioni su questa destinazione con il mittente. Nelle procedure illustrate nel resto di questa sezione, il destinatario dei dati di registro viene mostrato con un numero di account fittizio pari a 9999. AWS

Per iniziare a ricevere gli 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 concede a CloudWatch 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 regione. AWS Tuttavia, la risorsa AWS a cui punta la destinazione può trovarsi in una Regione diversa. Negli esempi delle sezioni seguenti, tutte le risorse specifiche della Regione vengono create in Stati Uniti orientali (Virginia settentrionale).

Argomenti

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

> **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 mittente AWS dei dati di registro è 1111.

Questo esempio crea una destinazione utilizzando un flusso Kinesis Data RecipientStream Streams chiamato e un ruolo CloudWatch che consente a 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 un flusso di destinazione in 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 usare il comando aws kinesis describe-stream per controllare. StreamDescription StreamStatusproprietà. Inoltre, prendi nota del valore StreamDescription.StreamArn perché lo passerai a Logs in un secondo CloudWatch momento:

```
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":
 "49551135218688818456679503831981458784591352702 18EXAMPLE"
        }
      }
    ]
  }
}
```

Potrebbero essere necessari uno o due minuti perché il flusso sia in stato attivo.

3. Crea il ruolo IAM che concede a CloudWatch Logs l'autorizzazione a inserire dati nel tuo stream. Per prima cosa, devi creare una politica di fiducia in un file TrustPolicyFor~/ CWL.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": {
            "StringLike": {
                "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 valore Role.Arn restituito perché verrà passato anche a Logs in un secondo momento: CloudWatch

```
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 i CloudWatch log possono eseguire sul tuo account. Innanzitutto, usa un editor di testo per creare una politica di autorizzazioni in un file ~/ CWL.json: PermissionsFor

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Associa la politica 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 nello stato 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 DestinationArnciò 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 politica deve specificare i log: PutSubscriptionFilter action e concede l'autorizzazione all'account mittente di 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.

```
{
```

```
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region: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.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" :
 "arn:aws:logs:region: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 effettuare chiamate PutSubscriptionFilterverso la destinazione con ARN arn:aws:logs ::9999:destination:testDestination. *region* Qualsiasi tentativo di chiamata 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 la procedura riportata di seguito. 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 degli abbonamenti 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 degli abbonamenti a livello di account.

I passaggi descritti in questa sezione creano un ruolo IAM, che CloudWatch può presupporre e verificare se l'account mittente è autorizzato a creare un filtro di sottoscrizione 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`.

Creazione del ruolo IAM necessario per le sottoscrizioni del log tra più account utilizzando 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 degli abbonamenti a livello di account.

Passaggio 3: Crea una politica di filtro degli abbonamenti 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 di sottoscrizione 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 inviato alla destinazione creata in precedenza. Tale destinazione incapsula un flusso 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 per determinare quali gruppi di log escludere, vedere. Registra la prevenzione della ricorsione 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 Kinesis Data Streams che si trova in una regione diversa.

Convalida del flusso dei log eventi

Dopo aver creato la politica di filtro dell'abbonamento a livello di account, CloudWatch Logs inoltra tutti gli eventi di registro in entrata che corrispondono al modello di filtro e ai criteri di selezione allo stream incapsulato nel flusso di destinazione denominato "». RecipientStream Il proprietario della destinazione può verificare che ciò stia accadendo utilizzando il get-shard-iterator comando aws kinesis per acquisire uno shard Kinesis Data Streams e utilizzando il comando aws kinesis get-records per recuperare alcuni record 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 `get-records` comando alcune volte prima che Kinesis Data Streams inizi a restituire dati.

Dovresti visualizzare una risposta con un array di log di Kinesis Data Streams. L'attributo dati nel log di Kinesis Data Streams usa la compressione nel formato gzip e la codifica 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 Kinesis Data Streams di tipo «CONTROL_MESSAGE», principalmente per verificare se la destinazione è raggiungibile.

owner

L' AWS ID dell'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 è stata applicata la politica. «ACCOUNT_LEVEL_POLICY» è il criterio 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 politica 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 ~/ .json: NewAccessPolicy

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Chiama PutDestinationPolicyper associare la politica definita nel NewAccessPolicyfile.json 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

> **Note**
>
> Questo passaggio è necessario solo per le sottoscrizioni tra più account per i log creati dai servizi elencati in Abilita la registrazione dai servizi AWS. 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 sia autorizzato a 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"
```

Per aggiornare un filtro di sottoscrizione e iniziare a utilizzare l'organizzazione IDs per le autorizzazioni di registro tra account diversi

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 di 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.

```
{
"Version" : "2012-10-17",
    "Statement" : [
        {
"Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region: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 un AWS account specifico IDs, includi il `force` parametro.

> **Warning**
>
> Se utilizzi i log inviati da un AWS servizio elencato in[Abilita la registrazione dai servizi AWS](), prima di eseguire questo passaggio devi 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
```

Abbonamenti a livello di account per più account con più account 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: ottiene 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 illustrate nel resto di questa sezione, il mittente dei dati di registro viene visualizzato con un numero di account fittizio pari a 1111 AWS .
- Destinatario dei dati di registro: imposta una destinazione che incapsula un flusso Kinesis Data Streams e CloudWatch fa sapere a Logs che il destinatario desidera ricevere i dati di registro. 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 registro viene mostrato con un numero di account fittizio di 222222222222. AWS

L'esempio in questa sezione utilizza un flusso di distribuzione Firehose con storage Amazon S3. È inoltre possibile configurare i flussi di distribuzione di Firehose con impostazioni diverse. Per ulteriori informazioni, vedere Creating a Firehose Delivery Stream.

> **Note**
>
> Il gruppo di log e la destinazione devono trovarsi nella stessa AWS regione. Tuttavia, la risorsa AWS 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 distribuzione tra regioni.

Argomenti

Fase 1: Creare un flusso di distribuzione Firehose

> **Important**
>
> Prima di completare i seguenti passaggi, è necessario utilizzare una policy di accesso, in modo che Firehose possa accedere al 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 distribuzione 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 l'autorizzazione a 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 sul 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. Immettete il seguente comando per creare il flusso di distribuzione di Firehose. Sostituisci *my-role-arn* e *amzn-s3-demo-bucket2-arn* con i valori corretti per la tua implementazione.

```
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 distribuzione Firehose diventi attivo. È possibile utilizzare il seguente comando per controllare ilStreamDescription. StreamStatusproprietà.

   ```
   aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
   ```

   Inoltre, prendi nota del DeliveryStreamDescription. DeliveryStreamValore ARN, perché sarà necessario 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 l'autorizzazione a inserire dati nel tuo flusso Firehose. Per prima cosa, devi creare una policy di fiducia in un file TrustPolicyFor~/ CWL.json. Utilizza un editor di testo per creare questa policy. Per ulteriori informazioni sugli endpoint di Amazon CloudWatch Logs, consulta Endpoints e quote di Amazon CloudWatch Logs.

   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": {
            "StringLike": {
                "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": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Crea una politica di autorizzazioni per definire quali azioni CloudWatch Logs può eseguire sul tuo account. Innanzitutto, usa un editor di testo per creare una politica di autorizzazioni in un file ~/CWL.json: PermissionsFor

```
{
    "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 nello stato attivo e hai creato il ruolo IAM, puoi creare la destinazione CloudWatch 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). È possibile utilizzare un editor di testo per inserire questa politica nel `~/AccessPolicy.json` file:

```
{
  "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 per accedere alla destinazione. Gli utenti con più account 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 degli abbonamenti a livello di account

Passare all'account di invio, che in questo esempio è 111111111111. Ora creerai la politica di filtro degli abbonamenti 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 un flusso Firehose che si trova in una regione diversa.

Convalida del flusso dei log eventi

Dopo aver creato il filtro di sottoscrizione, CloudWatch Logs inoltra tutti gli eventi di registro in entrata che corrispondono allo schema di filtro e ai criteri di selezione al flusso di distribuzione di Firehose. I dati iniziano a comparire nel bucket Amazon S3 in base all'intervallo di tempo impostato nel flusso di distribuzione di 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 PutDestinationPolicye 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 politica 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 politica nel file ~/ NewAccessPolicy .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. Usa il seguente comando per associare la politica definita nel NewAccessPolicyfile.json 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"

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel AWS, l'impersonificazione tra servizi può portare al confuso problema del vice. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a 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 del vostro 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 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 Comprendere il percorso dell' AWS Organizations entità.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn`con l'ARN completo della risorsa. Se non conosci l'ARN completo della risorsa o scegli più risorse, utilizza 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 Firehose mostrano come utilizzare la chiave SourceArn aws: global condition context per aiutare Fase 2: creazione di una destinazione a prevenire il confuso problema del vice.

Registra la prevenzione della ricorsione

Esiste il rischio di causare una ricorsione infinita dei log con i filtri di abbonamento, che può portare a un notevole aumento della fatturazione di importazione sia CloudWatch nei registri che nella destinazione, se non viene impedita. Ciò può verificarsi quando un filtro di sottoscrizione è associato a un gruppo di log che riceve eventi di registro come risultato del flusso di lavoro di distribuzione degli abbonamenti. I log inseriti nel gruppo di log verranno recapitati alla destinazione, facendo sì che il gruppo di log inserisca altri log che verranno poi nuovamente inoltrati alla destinazione, creando un ciclo di ricorsione.

Ad esempio, considera un filtro di sottoscrizione con destinazione Firehose, che invia gli eventi di registro ad Amazon S3. Inoltre, esiste anche una funzione Lambda che elabora i nuovi eventi

distribuiti ad Amazon S3 e produce automaticamente 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 alla destinazione, che richiamerà nuovamente la funzione, generando e inoltrando altri log a Firehose e Amazon S3, provocando un'altra chiamata della funzione e così via. Ciò avverrà in un ciclo infinito, con conseguente aumento imprevisto della fatturazione per l'ingestione dei 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 sottoscrizione ai gruppi di log che fanno parte del flusso di lavoro di distribuzione 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, prendi in considerazione i seguenti AWS servizi che producono registri e possono far parte dei flussi di lavoro di distribuzione degli abbonamenti:

- Amazon EC2 con Fargate
- Lambda
- AWS Step Functions
- Log di flusso di Amazon VPC abilitati per i log CloudWatch

> **Note**
>
> Gli eventi di registro prodotti dal gruppo di log di una destinazione Lambda non verranno inoltrati alla funzione Lambda per una politica di filtro di sottoscrizione a livello di account. In questo caso, l'esclusione dell'utilizzo del gruppo di log della funzione Lambda di destinazione non `selectionCriteria` 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

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.

I caratteri multibyte 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 più istanze 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. Per esempio:
  - `%gra|ey%`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`: restituisce un carattere numerico/non numerico. 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`: restituisce un carattere di spazio bianco/carattere diverso da uno spazio bianco.

> **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`: restituisce un carattere alfanumerico/non alfanumerico. 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`: restituisce la mappatura ASCII per un carattere esadecimale a due cifre. `\x` è la sequenza di escape che indica che i caratteri seguenti rappresentano il valore esadecimale per ASCII. `hh` specifica 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 puoi combinare il punto interrogativo ("?") con altri modelli di filtro, come termini di inclusione ed esclusione. Se combini "?" con altri modelli di filtro, il punto interrogativo ("?") viene ignorato.
> Ad esempio, il seguente schema di filtro corrisponde a tutti gli eventi che contengono la parola `REQUEST`, ma il punto interrogativo ("?") il filtro viene ignorato e non ha 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 ("{}"). I modelli di filtro basati su stringhe 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

```
{ $.['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.

Log eventi delimitato da spazi

```
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 dai servizi AWS

Sebbene molti servizi pubblichino i log solo nei CloudWatch log, alcuni AWS servizi possono pubblicare i log direttamente su Amazon Simple Storage Service o Amazon Data Firehose. Se il tuo requisito principale per i log è l'archiviazione o l'elaborazione in uno di questi servizi, puoi facilmente fare in modo che il servizio che produce i log li invii direttamente ad Amazon S3 o Firehose senza ulteriori configurazioni.

Anche quando i log vengono pubblicati direttamente su Amazon S3 o Firehose, vengono applicati dei costi. Per ulteriori informazioni, consulta Vending Logs nella scheda Logs di Amazon Pricing. CloudWatch

Alcuni AWS servizi utilizzano un'infrastruttura comune per inviare i propri log. Per abilitare la registrazione da questi servizi, è necessario accedere come utente con determinate autorizzazioni. Inoltre, è necessario concedere le autorizzazioni AWS per consentire l'invio dei log.

Per i servizi che richiedono queste autorizzazioni, sono necessarie due versioni delle autorizzazioni. I servizi che richiedono queste autorizzazioni aggiuntive sono indicati come [Autorizzazioni V1] supportate e [Autorizzazioni V2] supportate nella tabella. Per informazioni su queste autorizzazioni richieste, consulta le sezioni dopo la tabella.

Origine del log	Tipo di log	CloudWatch Logs	Amazon S3	Firehose
Log di accesso Gateway Amazon API	Registri venduti	[Autorizzazioni V1] supportate		
AWS AppSync logs	Registri personalizzati	Supportato		
Log di Amazon Aurora MySQL	Registri personalizzati	Supportato		

Origine del log	Tipo di log	CloudWatch Logs	Amazon S3	Firehose
Amazon Bedrock Registrazione delle basi di conoscenza	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate
Log di parametri di qualità dei supporti Amazon Chime e log dei messaggi SIP	Registri venduti	[Autorizzazioni V1] supportate		
CloudFront: registri di accesso	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate
AWS CloudHSM registri di controllo	Registri personalizzati	Supportato		
CloudWatch Evidentemente i registri degli eventi di valutazione	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	
CloudWatch Registri di Internet Monitor	Registri venduti		[Autorizzazioni V1] supportate	
CloudTrail registri	Registri personalizzati	Supportato		
AWS CodeBuild logs	Registri personalizzati	Supportato		
Amazon CodeWhisperer registri degli eventi	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate

Origine del log	Tipo di log	CloudWatch Logs	Amazon S3	Firehose
Amazon Cognito logs	Registri venduti	[Autorizzazioni V1] supportate		
Log di Amazon Connect	Registri personalizzati	Supportato		
AWS DataSync logs	Registri personalizzati	Supportato		
Registri di Amazon ElastiCache (Redis OSS)	Registri venduti	[Autorizzazioni V1] supportate		[Autorizzazioni V1] supportate
AWS Elastic Beanstalk logs	Registri personalizzati	Supportato		
Log di Amazon Elastic Container Service	Registri personalizzati	Supportato		
Log del piano di controllo (control-plane) Amazon Elastic Kubernetes Service	Registri venduti	Supportato		
AWS Elemental MediaPackage registri di accesso	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate
AWS Elemental MediaTailor log	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate

Origine del log	Tipo di log	CloudWatch Logs	Amazon S3	Firehose
Amazon EventBridge registrazione delle tubazioni	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
AWS Fargate logs	Registri personalizzati	Supportato		
AWS Fault Injection Service registri degli esperimenti	Registri venduti		[Autorizzazioni V1] supportate	
Amazon FinSpace	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
AWS Global Accelerator registri di flusso	Registri venduti		[Autorizzazioni V1] supportate	
AWS Glue registri di lavoro	Registri personalizzati	Supportato		
registri degli errori di IAM Identity Center	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate
Log delle chat di Amazon Interactive Video Service Chat	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
AWS IoT logs	Registri personalizzati	Supportato		

Origine del log	Tipo di log	CloudWatch Logs	Amazon S3	Firehose
AWS IoT FleetWise logs	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
AWS Lambda logs	Registri personalizzati	Supportato		
Log di Amazon Macie	Registri personalizzati	Supportato		
Registrazione del gestore di posta Amazon SES	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate
Modernizzazione del mainframe AWS	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
Log di Amazon Managed Service for Prometheus	Registri venduti	[Autorizzazioni V1] supportate		
Log di broker Amazon MSK	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
Log di Amazon MSK Connect	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
Log generali e di controllo di Amazon MQ	Registri personalizzati	Supportato		

Origine del log	Tipo di log	CloudWatch Logs	Amazon S3	Firehose
AWS Registri del Network Firewall	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
Log di accesso di Network Load Balancer	Registri venduti		[Autorizzazioni V1] supportate	
OpenSearch registri	Registri personalizzati	Supportato		
Registri OpenSearch di ingestione di Amazon Service	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
AWS OpsWorks logs	Registri personalizzati	Supportato		
Registri ServicePostgre SQL di Amazon Relational Database	Registri personalizzati	Supportato		
Registri delle conversazioni di Amazon Q Business	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate
AWS RoboMaker logs	Registri personalizzati	Supportato		
Log di query DNS pubblici di Amazon Route 53	Registri venduti	Supportato		

Origine del log	Tipo di log	CloudWatch Logs	Amazon S3	Firehose
Log delle query del risolutore Amazon Route 53	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	
Eventi Amazon SageMaker AI	Registri venduti	[Autorizzazioni V1] supportate		
Eventi per i lavoratori di Amazon SageMaker AI	Registri venduti	[Autorizzazioni V1] supportate		
AWS Registri VPN da sito a sito	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
Registri di Amazon Simple Email Service	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate
Log di Amazon Simple Notification Service	Registri personalizzati	Supportato		
Log delle policy di protezione dei dati di Amazon Simple Notification Service	Registri personalizzati	Supportato		
EC2 File di feed di dati di Spot Instance	Registri venduti		[Autorizzazioni V1] supportate	
AWS Step Functions Registri di Express Workflow e Standard Workflow	Registri venduti	[Autorizzazioni V1] supportate		

Origine del log	Tipo di log	CloudWatch Logs	Amazon S3	Firehose
Log di controllo e log di integrità del Gateway di archiviazione	Registri venduti	[Autorizzazioni V1] supportate		
AWS Transfer Family logs	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
Accesso verificato da AWS logs	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
Flussi di log Amazon Virtual Private Cloud	Registri venduti	Supportato	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
Log di accesso Amazon VPC Lattice	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate
AWS WAF logs	Registri venduti	[Autorizzazioni V1] supportate	[Autorizzazioni V1] supportate	Supportato
Amazon WorkMail registri di controllo	Registri venduti	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate	[Autorizzazioni V2] supportate

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 abilitare i servizi AWS elencati nella tabella seguente per inviare i log a queste destinazioni, devi essere connesso come un utente che dispone di 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.

Nella tabella seguente vengono riepilogati i tipi di log e le destinazioni di log a cui si applicano le informazioni contenute in questa sezione.

Nelle sezioni seguenti vengono fornite ulteriori dettagli per ciascuna di queste destinazioni.

Registri inviati a Logs CloudWatch

> **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`

```
{
  "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-1:0123456789:log-group:my-log-group:log-stream:*"
```

```
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

Se il gruppo di log dispone di una policy delle risorse, ma tale policy non contiene l'istruzione mostrata nel policy precedente e l'utente che imposta la registrazione ha le autorizzazioni `logs:PutResourcePolicy`, `logs:DescribeResourcePolicies`, e `logs:DescribeLogGroups` per il gruppo di log, tale istruzione viene aggiunta al policy delle risorse del gruppo di log.

Considerazioni relative al limite delle dimensioni delle policy delle risorse del gruppo di log

Questi servizi devono elencare ogni gruppo di log a cui inviano i log nella politica delle risorse e le politiche delle risorse di CloudWatch Logs sono limitate a 5120 caratteri. Un servizio che invia log a un gran numero di gruppi di log potrebbe raggiungere questo limite.

Per mitigare questo problema, CloudWatch Logs monitora la dimensione delle politiche relative alle risorse utilizzate dal servizio che invia i log e, quando rileva che una policy si avvicina al limite di dimensione di 5120 caratteri, CloudWatch Logs `/aws/vendedlogs/*` abilita automaticamente la politica delle risorse per quel servizio. Quindi puoi iniziare a utilizzare gruppi di log con nomi che iniziano con `/aws/vendedlogs/` come destinazioni per i log di questi servizi.

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 stai inviando 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 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 attualmente non dispone di una politica delle risorse e l'utente che configura la registrazione dispone delle autorizzazioni `S3:GetBucketPolicy` e `S3:PutBucketPolicy` delle autorizzazioni per il bucket, crea AWS automaticamente la seguente politica quando inizi a inviare i log ad Amazon S3.

```
{
    "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-1:0123456789:*"]
                }
            }
        },
        {
            "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-1:0123456789:*"]
                }
            }
        }
    ]
}
```

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 ARNs della risorsa che genera i log, nel modulo. `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 gestite da Amazon S3 (SSE-S3) o la crittografia lato server con una chiave archiviata in (SSE-KMS). AWS KMS AWS Key Management Service Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server.

Se si sceglie SSE-S3, non è richiesta alcuna configurazione aggiuntiva. Amazon S3 gestisce la chiave di crittografia.

> **Warning**
>
> Se scegli SSE-KMS, devi utilizzare una chiave gestita dal cliente, poiché l'utilizzo di una chiave gestita non è supportato in questo scenario. AWS 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 scegli 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 ARNs della risorsa che genera i log, nel modulo. `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 seguenti autorizzazioni.

```
{
    "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:

```
{
  "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 Firehose.

La consegna di un 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 stai consegnando i log su più account, devi utilizzarli PutDeliveryDestinationPolicynell'account di destinazione per assegnare una IAM politica 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, è necessario creare manualmente le politiche di autorizzazione.
- Crea una consegna associando esattamente una fonte di consegna e una destinazione di consegna, utilizzando. CreateDelivery

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

Per abilitare l'invio dei log ai CloudWatch registri, è necessario accedere con le seguenti autorizzazioni.

```
{
    "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:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id: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:region:account-id:*"
            ]
        }
    ]
}
```

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

```
{
  "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-1:0123456789:log-group:my-log-group:log-stream:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

Considerazioni relative al limite delle dimensioni delle policy delle risorse del gruppo di log

Questi servizi devono elencare ogni gruppo di log a cui inviano i log nella politica delle risorse e le politiche delle risorse di CloudWatch Logs sono limitate a 5120 caratteri. Un servizio che invia i log a un numero elevato di gruppi di log potrebbe rientrare in questo limite.

Per mitigare questo problema, CloudWatch Logs monitora la dimensione delle politiche relative alle risorse utilizzate dal servizio che invia i log e, quando rileva che una policy si avvicina al limite di dimensione di 5120 caratteri, CloudWatch Logs abilita automaticamente la politica delle risorse per quel servizio. `/aws/vendedlogs/*` Quindi puoi iniziare a utilizzare gruppi di log con nomi che iniziano con `/aws/vendedlogs/` come destinazioni per i log di questi servizi.

Log inviati ad Amazon S3

Autorizzazioni degli utenti

Per abilitare l'invio di log ad Amazon S3, devi aver effettuato l'accesso con le seguenti autorizzazioni.

```
{
    "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:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id: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"
```

```
        }
    ]
}
```

Il bucket S3 in cui vengono inviati i log deve disporre di una policy delle risorse che include determinate autorizzazioni. Se il bucket attualmente non dispone di una politica delle risorse e l'utente che configura la registrazione dispone delle autorizzazioni `S3:GetBucketPolicy` e `S3:PutBucketPolicy` delle autorizzazioni per il bucket, crea AWS automaticamente la seguente politica quando inizi a inviare i log ad Amazon S3.

```
{
    "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-1:0123456789: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 ARNs della risorsa che genera i log, nel modulo. `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 gestite da Amazon S3 (SSE-S3) o la crittografia lato server con una chiave archiviata in (SSE-KMS). AWS KMS AWS Key Management Service Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server.

Se si sceglie SSE-S3, non è richiesta alcuna configurazione aggiuntiva. Amazon S3 gestisce la chiave di crittografia.

> **Warning**
>
> Se scegli SSE-KMS, devi utilizzare una chiave gestita dal cliente, poiché l'utilizzo di una chiave gestita non è supportato in questo scenario. AWS 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 scegli 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:delivery-source:*"]
        }
        }
}
```

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 ARNs della risorsa che genera i log, nel modulo. `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`

Log inviati a Firehose

Autorizzazioni degli utenti

Per abilitare l'invio di log a Firehose, è necessario accedere con le seguenti autorizzazioni.

```
{
    "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:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id: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:region:account-id:deliverystream/*"
            ]
```

```
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::account-id: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 seguenti autorizzazioni.

```
{
    "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:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Autorizzazioni specifiche del servizio

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. Autorizza l'`AllowVendedLogDeliveryForResource`azione per le risorse che vendono i log all'interno di quel servizio. Per questi servizi, utilizza la seguente politica *`service`* e sostituiscila *`resource-type`* con i valori appropriati. Per i valori specifici del servizio per questi campi, consulta la pagina della documentazione di tali servizi per i registri forniti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ServiceLevelAccessForLogDelivery",
            "Effect": "Allow",
            "Action": [
                "service:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:service:region:account-id:resource-type/*"
        }
    ]
}
```

Autorizzazioni specifiche per la console

Oltre alle autorizzazioni elencate nelle sezioni precedenti, se si configura la consegna dei log utilizzando la console anziché il APIs, sono necessarie anche le seguenti autorizzazioni aggiuntive:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogDeliveryActionsConsoleCWL",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "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": [
                "*"
            ]
        }
    ]
}
```

Esempio di distribuzione tra account

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*AAAAAAAAAAAA*, ID:. *BBBBBBBBBBB*

L'account A desidera inserire i log della Amazon Bedrock knowledge base nel proprio account con l'ARN *region* arn:aws:bedrock: ::knowledge-base/. *AAAAAAAAAAA XXXXXXXXX*

Per questo esempio, l'account A necessita delle seguenti autorizzazioni:

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowVendedLogDeliveryForKnowledgeBase",
            "Effect": "Allow",
            "Action": [
                "bedrock:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:bedrock:region:AAAAAAAAAAAA:knowledge-base/XXXXXXXXX"
        },
        {
            "Sid": "CreateLogDeliveryPermissions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliverySource",
                "logs:CreateDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:AAAAAAAAAAA:delivery-source:*",
                "arn:aws:logs:region:AAAAAAAAAAA:delivery:*",
                "arn:aws:logs:region:BBBBBBBBBBB: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:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PutLogDestinationPermissions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliveryDestination",
                "logs:PutDeliveryDestinationPolicy"
            ],
            "Resource": "arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:*"
        }
    ]
}
```

Il bucket avrà bisogno delle seguenti autorizzazioni nella sua politica relativa al bucket:

```
{
    "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/AAAAAAAAAAAA/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": ["AAAAAAAAAAAA"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:region:AAAAAAAAAAAA:delivery-
source:my-delivery-source"]
                }
            }
        }
    ]
}
```

Se il bucket è crittografato con SSE-KMS, assicurati che la policy chiave disponga delle AWS KMS autorizzazioni appropriate. Ad esempio, se la chiave KMS è, usa quanto segue: `arn:aws:kms:region:BBBBBBBBBBBB:key/X`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogsGenerateDataKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            }
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:region:BBBBBBBBBBBB:key/X",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": ["AAAAAAAAAAAA"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:region:AAAAAAAAAAAA: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:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {
                "AWS": "AAAAAAAAAAAA"
            },
            "Action": [
                "logs:CreateDelivery"
            ],
            "Resource": "arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:my-s3-
delivery-destination"
        }
    ]
}
```

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:region:BBBBBBBBBBBB:deliverystream/X`

Per questo esempio, l'account B necessita delle seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFirehoseCreateSLR",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::BBBBBBBBBBBB:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery",
        },
        {
            "Sid": "AllowFirehoseTagging",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": "arn:aws:firehose:region:BBBBBBBBBBBB:deliverystream/X"
        },
        {
            "Sid": "AllowFirehoseDeliveryDestination",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliveryDestination",
                "logs:PutDeliveryDestinationPolicy"
            ],
            "Resource": "arn:aws:logs:region:BBBBBBBBBBBB: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:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {
                "AWS": "AAAAAAAAAAAA"
            },
            "Action": [
                "logs:CreateDelivery"
            ],
            "Resource": "arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:my-fh-
delivery-destination"
        }
    ]
}
```

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
```

Prevenzione del confused deputy tra servizi

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 frattempo AWS, l'impersonificazione tra diversi servizi può portare al confuso problema del vicesceriffo. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a 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 ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Si consiglia di utilizzare le chiavi contestuali `aws:SourceArnaws:SourceAccount`, `aws: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. 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, vedi Comprendere il percorso dell' AWS Organizations entità.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn`con l'ARN completo della risorsa. Se non conosci l'ARN completo della risorsa o scegli più risorse, utilizza 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 policy nelle sezioni precedenti di questa pagina mostrano come è possibile utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` per prevenire il problema "confused deputy".

CloudWatch Registra gli aggiornamenti delle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti alle politiche AWS gestite per CloudWatch Logs da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei documenti di CloudWatch Logs.

Modifica	Descrizione	Data
AWSServiceRoleForLogDelivery politica relativa ai ruoli collegati ai servizi: aggiornamento a una politica esistente	CloudWatch I log hanno modificato le autorizzazioni nella politica IAM associata a AWSServiceRoleForLogDeliveryruolo collegato al servizio. È stata apportata la seguente modifica: • La chiave di condizione `firehose:ResourceTag/LogDeliveryEnabled": "true"` è stata modificata in `aws:ResourceTag/LogDeliveryEnabled": "true"` .	15 luglio 2021

Modifica	Descrizione	Data
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

Esportazione di dati di log in Amazon S3

Questo capitolo fornisce informazioni per esportare i dati di log dai gruppi di log in un bucket Amazon S3 per elaborazioni e analisi personalizzate o per caricarli su altri sistemi. È possibile eseguire l'esportazione in un bucket nello stesso account o in un account diverso.

Puoi eseguire le operazioni indicate di seguito:

- Esporta i dati di log 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

> **Note**
>
> L'esportazione in Amazon S3 è supportata solo per i gruppi di log nella classe di log Standard. Per ulteriori informazioni sulle classi di log, consultaClassi di registro.

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. Elaborazione in tempo reale dei dati di log con le sottoscrizioni

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 eseguire l'esportazione in bucket S3 che sono crittografati con AES-256 o con 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 separare i dati di log per ogni attività di esportazione, puoi specificare un prefisso che verrà utilizzato come prefisso della chiave Amazon S3 per tutti gli oggetti esportati.

> **Note**
>
> L'ordinamento basato sul tempo su 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 Elaborazione in tempo reale dei dati di log con le sottoscrizioni.

Indice

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. `my-exported-logs`

L'esportazione dei dati di log in bucket S3 crittografati da SSE-KMS è supportata. L'esportazione in bucket crittografati con DSSE-KMS non è supportata.

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

- Esportazione nello stesso account
- Esportazione in account diversi

Esportazione nello stesso account

Se il bucket Amazon S3 si trova nello stesso account dei log che vengono esportati, segui le istruzioni in questa sezione.

Argomenti

Fase 1: creazione di un bucket Amazon S3

Ti consigliamo di utilizzare un bucket creato appositamente per Logs. CloudWatch Tuttavia, se intendi utilizzare un bucket esistente, puoi passare alla fase 2.

> **Note**
>
> Il bucket S3 deve risiedere nella stessa regione dei dati di registro da esportare. CloudWatch I registri non supportano l'esportazione di dati in bucket S3 in una regione diversa.

Per creare un bucket 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 Create (Crea) .

Fase 2: configurare le autorizzazioni di accesso

Per creare l'attività di esportazione nella fase 5, è necessario essere registrati con il ruolo IAM `AmazonS3ReadOnlyAccess` e 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: AWS IAM Identity Center

  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 AWS IAM Identity Center .
- 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.

Passaggio 3: Impostazione delle autorizzazioni su un bucket S3

Per impostazione predefinita, tutti i bucket e gli oggetti S3 sono privati. Solo il proprietario della risorsa, l' Account AWS 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.

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 di account IDs 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, scegliere il bucket creato nella fase 1.
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 `my-exported-logs` nel nome del bucket S3. Assicurati di specificare l'endpoint corretto della regione, come `us-west-1`, per Principale.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
```

```
                        ...
                    ]
            }
        }
    },
    {
        "Action": "s3:PutObject" ,
        "Effect": "Allow",
        "Resource": "arn:aws:s3:::my-exported-logs/*",
        "Principal": { "Service": "logs.Region.amazonaws.com" },
        "Condition": {
          "StringEquals": {
              "s3:x-amz-acl": "bucket-owner-full-control",
              "aws:SourceAccount": [
                  "AccountId1",
                  "AccountId2",
                  ...
              ]
          },
          "ArnLike": {
                  "aws:SourceArn": [
                      "arn:aws:logs:Region:AccountId1:log-group:*",
                      "arn:aws:logs:Region:AccountId2:log-group:*",
                      ...
                  ]
          }
        }
    }
]
}
```

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.

(Opzionale) Passaggio 4: Esportazione in un bucket crittografato con SSE-KMS

Questo passaggio è necessario 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.

Esportazione di un bucket crittografato con SSE-KMS

1. Apri la console in /kms. AWS KMS https://console.aws.amazon.com
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra 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.

```
{
    "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 Fase 1: Creazione di un bucket S3 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 Encryption (Crittografia lato server), scegli Enable (Abilita).

19. In Tipo di crittografia, scegli Chiave AWS Key Management Service (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).

Passaggio 5: Creazione di un'attività di esportazione

In questa fase, verrà creata l'attività di esportazione per esportare 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 Fase 2: configurare le autorizzazioni di accesso.
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.

Esportazione in account diversi

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

Fase 1: creazione di un bucket Amazon S3

Ti consigliamo di utilizzare un bucket creato appositamente per i CloudWatch log. Tuttavia, se intendi utilizzare un bucket esistente, puoi passare alla fase 2.

> **Note**
>
> Il bucket S3 deve risiedere nella stessa regione dei dati di registro da esportare. CloudWatch I registri non supportano l'esportazione di dati in bucket S3 in una regione diversa.

Per creare un bucket 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 Create (Crea) .

Fase 2: configurare le autorizzazioni di accesso

Innanzitutto, devi creare una nuova policy IAM per consentire a CloudWatch Logs di avere l'`s3:PutObject`autorizzazione per il bucket Amazon S3 di destinazione nell'account di destinazione.

La policy che crei dipende dal fatto che il bucket di destinazione utilizzi la crittografia. AWS KMS

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 Create Policy (Crea policy).
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Se il bucket di destinazione non utilizza AWS KMS la crittografia, incolla la seguente politica nell'editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Se il bucket di destinazione utilizza la AWS KMS crittografia, incolla la seguente politica nell'editor.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-exported-logs/*"
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "ARN_OF_KMS_KEY"
    }
  ]
}
```

6. Scegli Next (Successivo).
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 l'attività di esportazione nella fase 5, sarà necessario essersi registrati con il ruolo IAM `AmazonS3ReadOnlyAccess`. Devi inoltre avere effettuato l'accesso con la policy IAM appena creata 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 AWS IAM Identity Center:

  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 AWS IAM Identity Center .
- 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.

Passaggio 3: Impostazione delle autorizzazioni su un bucket S3

Per impostazione predefinita, tutti i bucket e gli oggetti S3 sono privati. Solo il proprietario della risorsa, l' Account AWS 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.

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 di account IDs 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, scegliere il bucket creato nella fase 1.
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 `my-exported-logs` nel nome del bucket S3. Assicurati di specificare l'endpoint corretto della regione, come `us-west-1`, per Principale.

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

```
            "Action": "s3:GetBucketAcl",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                       ]
              }
            }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                      ]
              }
            }
        },
```

```
      {
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
          },
          "Action": "s3:PutObject",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "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.

(Opzionale) Passaggio 4: Esportazione in un bucket crittografato con SSE-KMS

Questo passaggio è necessario 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.

Esportazione di un bucket crittografato con SSE-KMS

1. Apri la console in /kms. AWS KMS https://console.aws.amazon.com
2. Per modificare il Regione AWS, usa il selettore della regione nell'angolo in alto a destra 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.

```
{
    "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": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

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 Fase 1: Creazione di un bucket S3 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 Encryption (Crittografia lato server), scegli Enable (Abilita).
19. In Tipo di crittografia, scegli Chiave AWS Key Management Service (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).

Passaggio 5: Creazione di un'attività di esportazione

In questa fase, verrà creata l'attività di esportazione per esportare 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 Fase 2: configurare le autorizzazioni di accesso.
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. `my-exported-logs` 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 DSSE-KMS non è supportata.

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

Esportazione nello stesso account

Se il bucket Amazon S3 si trova nello stesso account dei log che vengono esportati, segui le istruzioni in questa sezione.

Argomenti

Fase 1: Creazione di un bucket S3

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 S3 deve risiedere nella stessa regione dei dati di registro 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 my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

Di seguito è riportato un output di esempio.

```
{
    "Location": "/my-exported-logs"
}
```

Fase 2: configurare le autorizzazioni di accesso

Per creare l'attività di esportazione nella fase 5, è necessario essere registrati con il ruolo IAM `AmazonS3ReadOnlyAccess` e 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: AWS IAM Identity Center

  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 AWS IAM Identity Center .
- 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.

Passaggio 3: Impostazione delle autorizzazioni su un bucket S3

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 di account IDs 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 `my-exported-logs` con il nome del bucket S3 e `Principal` con l'endpoint della regione di esportazione dei dati di log, come `us-west-1`. Utilizza un editor di testo per creare questo file di policy. Non utilizzare la console IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
```

```
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                    ]
            }
          }
      }
    ]
}
```

2. Imposta la politica che hai appena aggiunto come politica di accesso sul tuo bucket utilizzando 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 my-exported-logs --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.

(Opzionale) Passaggio 4: Esportazione in un bucket crittografato con SSE-KMS

Questo passaggio è necessario 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.

Esportazione di 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.

```
{
    "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_id: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 *`bucket-name`* con il nome del bucket in cui state esportando i log.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

   Se il comando non restituisce un errore, il processo ha esito positivo.

Passaggio 5: Creazione di un'attività di esportazione

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 Fase 2: configurare le autorizzazioni di accesso.
2. Al prompt dei comandi, usa il seguente create-export-taskcomando 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 "my-exported-logs" --destination-prefix "export-
task-output"
```

Di seguito è riportato un output di esempio.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Esportazione in account diversi

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

- Fase 1: Creazione di un bucket S3
- Fase 2: configurare le autorizzazioni di accesso
- Passaggio 3: Impostazione delle autorizzazioni su un bucket S3
- (Opzionale) Passaggio 4: Esportazione in un bucket crittografato con SSE-KMS
- Passaggio 5: Creazione di un'attività di esportazione

Fase 1: Creazione di un bucket S3

Si consiglia di utilizzare un bucket creato appositamente per CloudWatch i registri. Tuttavia, se intendi utilizzare un bucket esistente, puoi passare alla fase 2.

> **Note**
>
> Il bucket S3 deve risiedere nella stessa regione dei dati di registro 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 my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

Di seguito è riportato un output di esempio.

```
{
    "Location": "/my-exported-logs"
}
```

Fase 2: configurare le autorizzazioni di accesso

Innanzitutto, devi creare una nuova policy IAM per consentire a CloudWatch Logs di avere l'`s3:PutObject`autorizzazione per il bucket Amazon S3 di destinazione.

Per creare l'attività di esportazione nella fase 5, sarà necessario essersi registrati con il ruolo IAM `AmazonS3ReadOnlyAccess` e altre autorizzazioni. È possibile creare una policy che contenga alcune di queste altre autorizzazioni necessarie.

La policy che crei 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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Se il bucket di destinazione utilizza AWS KMS la crittografia, create una policy con i seguenti contenuti.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

Per creare l'attività di esportazione nella fase 5, devi avere effettuato l'accesso con il ruolo IAM `AmazonS3ReadOnlyAccess`, la policy IAM appena creata 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 AWS IAM Identity Center:

  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 AWS IAM Identity Center .
- 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.

Passaggio 3: Impostazione delle autorizzazioni su un bucket S3

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 di account IDs 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 `my-exported-logs` con il nome del bucket S3 e `Principal` con l'endpoint della regione di esportazione dei dati di log, come `us-west-1`. Utilizza un editor di testo per creare questo file di policy. Non utilizzare la console IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
```

```
                ...
            ]
        },
        "ArnLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:Region:AccountId1:log-group:*",
                    "arn:aws:logs:Region:AccountId2:log-group:*",
                    ...
                 ]
        }
      }
    },
    {
        "Action": "s3:PutObject" ,
        "Effect": "Allow",
        "Resource": "arn:aws:s3:::my-exported-logs/*",
        "Principal": { "Service": "logs.Region.amazonaws.com" },
        "Condition": {
          "StringEquals": {
              "s3:x-amz-acl": "bucket-owner-full-control",
              "aws:SourceAccount": [
                  "AccountId1",
                  "AccountId2",
                  ...
              ]
          },
          "ArnLike": {
                  "aws:SourceArn": [
                      "arn:aws:logs:Region:AccountId1:log-group:*",
                      "arn:aws:logs:Region:AccountId2:log-group:*",
                      ...
                  ]
          }
        }
    },
    {
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
        },
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::my-exported-logs/*",
        "Condition": {
          "StringEquals": {
```

```
                "s3:x-amz-acl": "bucket-owner-full-control"
            }
        }
    }
  ]
}
```

2. Imposta la politica che hai appena aggiunto come politica di accesso sul tuo bucket utilizzando 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 my-exported-logs --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.

(Opzionale) Passaggio 4: Esportazione in un bucket crittografato con SSE-KMS

Questo passaggio è necessario 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.

Esportazione di 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.

```
{
    "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": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

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_id: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 *bucket-name* con il nome del bucket in cui state esportando i log.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

Se il comando non restituisce un errore, il processo ha esito positivo.

Passaggio 5: Creazione di un'attività di esportazione

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 Fase 2: configurare le autorizzazioni di accesso.
2. Al prompt dei comandi, usa il seguente create-export-taskcomando 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 "my-exported-logs" --destination-prefix "export-
task-output"
```

Di seguito è riportato un output di esempio.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Descrizione dei processi di esportazione

Dopo aver creato un'attività di esportazione, puoi ottenere lo stato corrente dell'attività.

Per descrivere le attività di esportazione utilizzando il AWS CLI

Al prompt dei comandi, utilizzare il describe-export-taskscomando seguente.

```
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": "my-exported-logs",
      "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
   }]
}
```

Annullamento di un processo di esportazione

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, utilizzate il seguente cancel-export-taskcomando:

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

È possibile utilizzare il describe-export-taskscomando per verificare che l'operazione sia stata annullata correttamente.

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 Elaborazione in tempo reale dei dati di log con le sottoscrizioni.

> **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. AWS Billing and Cost Management 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 VPC accesso, ma non è possibile modificare il tipo di accesso dopo la creazione del 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, fornisci il dominio ARN e l'endpoint.
6. Per Lambda IAM Execution Role, scegli il IAM ruolo che Lambda deve usare per eseguire le chiamate. OpenSearch

   Il IAM ruolo che scegli deve soddisfare questi requisiti:

   - Deve avere `lambda.amazonaws.com` nella relazione di trust.
   - Deve includere la policy seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:region:account-id:domain/target-domain-name/
*"
        }
    ]
}
```

- Se il dominio OpenSearch di servizio di destinazione utilizza VPC l'accesso, al ruolo deve essere associata la AWSLambdaVPCAccessExecutionRolepolicy. Questa politica gestita da Amazon concede l'accesso Lambda al clienteVPC, consentendo a Lambda di scrivere sull'endpoint in. OpenSearch VPC

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. Questo ti assicura di inviare solo i dati che ti interessano al tuo cluster. 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 SDKs

I seguenti esempi di codice mostrano come utilizzare CloudWatch Logs con un kit di sviluppo AWS software ()SDK.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto nei relativi scenari.

Gli scenari sono esempi di codice che mostrano come eseguire attività specifiche richiamando più funzioni all'interno di un servizio o combinandole con altre Servizi AWS.

Per un elenco completo di guide per AWS SDK sviluppatori ed esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Esempi di codice

Esempi di base per l'utilizzo CloudWatch dei log AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Amazon CloudWatch Logs con. AWS SDKs

Esempi

Azioni per CloudWatch l'utilizzo dei log AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole azioni di CloudWatch Logs con. AWS SDKs Ogni esempio include un collegamento a GitHub, dove è possibile trovare le istruzioni per la configurazione e l'esecuzione del codice.

Questi estratti si chiamano CloudWatch Logs API e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. È possibile visualizzare le azioni nel contesto in. Scenari per l'utilizzo CloudWatch dei log AWS SDKs

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta Amazon CloudWatch Logs API Reference.

Esempi

- Da utilizzare AssociateKmsKey con un AWS SDK
- Utilizzare CancelExportTask con un AWS SDK
- Utilizzare CreateExportTask con un AWS SDK
- Da utilizzare CreateLogGroup con un AWS SDK o CLI
- Da utilizzare CreateLogStream con un AWS SDK o CLI
- Da utilizzare DeleteLogGroup con un AWS SDK o CLI
- Utilizzare DeleteSubscriptionFilter con un AWS SDK
- Utilizzare DescribeExportTasks con un AWS SDK
- Utilizzare DescribeLogGroups con un AWS SDK o CLI
- Utilizzare DescribeSubscriptionFilters con un AWS SDK
- Utilizzare GetQueryResults con un AWS SDK
- Utilizzare PutSubscriptionFilter con un AWS SDK
- Utilizzare StartLiveTail con un AWS SDK
- Utilizzare StartQuery con un AWS SDK

Da utilizzare **`AssociateKmsKey`** con un AWS SDK

Il seguente esempio di codice mostra come utilizzare`AssociateKmsKey`.

.NET

AWS SDK for .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 API i dettagli, vedi AssociateKmsKey AWS SDK for .NETAPIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`CancelExportTask`** con un AWS SDK

Il seguente esempio di codice mostra come utilizzare`CancelExportTask`.

.NET

AWS SDK for .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 API i dettagli, vedi CancelExportTask AWS SDK for .NETAPIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`CreateExportTask`** con un AWS SDK

Il seguente esempio di codice mostra come utilizzare`CreateExportTask`.

.NET

AWS SDK for .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 API i dettagli, vedi CreateExportTask AWS SDK for .NETAPIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Da utilizzare **`CreateLogGroup`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateLogGroup`.

.NET

AWS SDK for .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 API i dettagli, vedi CreateLogGroup AWS SDK for .NETAPIReference.

CLI

AWS CLI

Il comando seguente crea un gruppo di log denominatomy-logs:

```
aws logs create-log-group --log-group-name my-logs
```

- Per API i dettagli, vedere CreateLogGroupin AWS CLI Command Reference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { 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 API i dettagli, vedi CreateLogGroup AWS SDK for JavaScriptAPIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Da utilizzare **`CreateLogStream`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`CreateLogStream`.

.NET

AWS SDK for .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 API i dettagli, vedi CreateLogStream AWS SDK for .NETAPIReference.

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 API i dettagli, vedere CreateLogStreamin AWS CLI Command Reference.

Per un elenco completo di guide per AWS SDK sviluppatori ed esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Da utilizzare **`DeleteLogGroup`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteLogGroup`.

.NET

AWS SDK for .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 API i dettagli, vedi DeleteLogGroup AWS SDK for .NETAPIReference.

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 API i dettagli, vedere DeleteLogGroupin AWS CLI Command Reference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { 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 API i dettagli, vedi DeleteLogGroup AWS SDK for JavaScriptAPIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`DeleteSubscriptionFilter`** con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`DeleteSubscriptionFilter`.

C++

SDKper 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 API i dettagli, vedi DeleteSubscriptionFilter AWS SDK for C++APIReference.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.services.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 API i dettagli, vedi DeleteSubscriptionFilter AWS SDK for Java 2.xAPIReference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { 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 API i dettagli, vedi DeleteSubscriptionFilter AWS SDK for JavaScriptAPIReference.

SDKper 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 for JavaScript.
- Per API i dettagli, vedi DeleteSubscriptionFilter AWS SDK for JavaScriptAPIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun deleteSubFilter(
    filter: String?,
    logGroup: String?,
) {
    val request =
        DeleteSubscriptionFilterRequest {
            filterName = filter
            logGroupName = logGroup
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { logs ->
        logs.deleteSubscriptionFilter(request)
        println("Successfully deleted CloudWatch logs subscription filter named
 $filter")
    }
}
```

- Per API i dettagli, vedi il riferimento DeleteSubscriptionFilter AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo dei CloudWatch log con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`DescribeExportTasks`** con un AWS SDK

Il seguente esempio di codice mostra come utilizzare`DescribeExportTasks`.

.NET

AWS SDK for .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 API i dettagli, vedi DescribeExportTasks AWS SDK for .NETAPIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`DescribeLogGroups`** con un AWS SDK o CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeLogGroups`.

.NET

AWS SDK for .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 API i dettagli, vedi DescribeLogGroups AWS SDK for .NETAPIReference.

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 API i dettagli, vedere DescribeLogGroupsin AWS CLI Command Reference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import {
  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 API i dettagli, vedi DescribeLogGroups AWS SDK for JavaScriptAPIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`DescribeSubscriptionFilters`** con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`DescribeSubscriptionFilters`.

C++

SDKper 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 API i dettagli, vedi DescribeSubscriptionFilters AWS SDK for C++APIReference.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.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 API i dettagli, vedi DescribeSubscriptionFilters AWS SDK for Java 2.xAPIReference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { 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 API i dettagli, vedi DescribeSubscriptionFilters AWS SDK for JavaScriptAPIReference.

SDKper 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 for JavaScript.
- Per API i dettagli, vedi DescribeSubscriptionFilters AWS SDK for JavaScriptAPIReference.

Kotlin

SDKper Kotlin

> **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.

```
suspend fun describeFilters(logGroup: String) {
    val request =
        DescribeSubscriptionFiltersRequest {
            logGroupName = logGroup
            limit = 1
        }

    CloudWatchLogsClient { 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 API i dettagli, vedi il riferimento DescribeSubscriptionFilters AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo dei CloudWatch log con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

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:

- Esegui una query di grandi dimensioni

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
```

- Per API i dettagli, vedi GetQueryResults AWS SDK for JavaScriptAPIReference.

Python

SDKper Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    def _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 API i dettagli, vedere GetQueryResultsPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo dei CloudWatch log con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`PutSubscriptionFilter`** con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`PutSubscriptionFilter`.

C++

SDKper 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 API i dettagli, vedi PutSubscriptionFilter AWS SDK for C++APIReference.

Java

SDKper Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.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 API i dettagli, vedi PutSubscriptionFilter AWS SDK for Java 2.xAPIReference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { 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 API i dettagli, vedi PutSubscriptionFilter AWS SDK for JavaScriptAPIReference.

SDKper 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 for JavaScript.

- Per API i dettagli, vedi PutSubscriptionFilter AWS SDK for JavaScriptAPIReference.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`StartLiveTail`** con un AWS SDK

I seguenti esempi di codice mostrano come utilizzare`StartLiveTail`.

.NET

AWS SDK for .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;
            }
```

Puoi gestire gli eventi della sessione di 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 API i dettagli, vedere StartLiveTailin AWS SDK for .NET APIReference.

Go

SDKper 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"
)
```

Gestisci 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)
```

Interrompi la sessione Live Tail dopo un certo 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 API i dettagli, vedi Reference StartLiveTail.AWS SDK per Go API

Java

SDKper 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;
```

Gestisci 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));
```

Interrompi la sessione Live Tail dopo un certo 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 API i dettagli, vedi Reference StartLiveTail.AWS SDK for Java 2.x API

JavaScript

SDKper JavaScript (v3)

Includere i file richiesti.

```
import { CloudWatchLogsClient, StartLiveTailCommand } from "@aws-sdk/client-
cloudwatch-logs";
```

Gestisci gli eventi della sessione di 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);
    }
```

Interrompi la sessione Live Tail dopo un certo 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 API i dettagli, vedi Reference StartLiveTail.AWS SDK for JavaScript API

Kotlin

SDKper 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 di 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 API i dettagli, consulta la sezione StartLiveTail AWSSDKdi riferimento di Kotlin API.

Python

SDKper Python (Boto3)

Includere i file richiesti.

```
import boto3
import time
from datetime import datetime
```

Avvia la sessione di 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 API i dettagli, vedere StartLiveTailPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo dei CloudWatch log con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

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:

- Esegui una query di grandi dimensioni

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/**
 * 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 API i dettagli, vedi StartQuery AWS SDK for JavaScriptAPIReference.

Python

SDKper Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    def 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 API i dettagli, vedere StartQueryPython (Boto3) Reference.AWS SDK API

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo dei CloudWatch log con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Scenari per l'utilizzo CloudWatch dei log AWS SDKs

I seguenti esempi di codice mostrano come implementare scenari comuni in CloudWatch Logs with. AWS SDKs 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, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice.

Gli scenari si basano su un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- Usa CloudWatch Logs per eseguire una query di grandi dimensioni
- Utilizzo degli eventi pianificati per richiamare una funzione Lambda

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.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Questo 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 divide le interrogazioni 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 API i dettagli, consultate i seguenti argomenti in AWS SDK for JavaScript API Reference.
  - GetQueryResults
  - StartQuery

Python

SDKper Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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 API i dettagli, consulta i seguenti argomenti in AWS SDKPython (Boto3) Reference. API
  - GetQueryResults
  - StartQuery

Per un elenco completo delle guide per gli AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo dei CloudWatch log con un SDK AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzo degli eventi pianificati per richiamare una funzione Lambda

I seguenti esempi di codice mostrano come creare una AWS Lambda funzione richiamata da un evento EventBridge pianificato di Amazon.

Python

SDKper 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
- EventBridge
- Lambda

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo dei CloudWatch log con un SDK AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

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

> **Note**
>
> Oltre alle seguenti informazioni sulla protezione generale dei dati in AWS, CloudWatch Logs consente inoltre 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 AWS modello di responsabilità condivisa modello 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 tutte le Cloud AWS. L'utente è responsabile del mantenimento del controllo sui contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile delle attività di configurazione e gestione della sicurezza per Servizi AWS che usi. Per ulteriori informazioni sulla privacy dei dati, consulta la sezione Privacy dei dati FAQ. Per informazioni sulla protezione dei dati in Europa, consulta la AWS Modello di responsabilità condivisa e post sul GDPR blog sul AWS Blog sulla sicurezza.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS credenziali e configura singoli utenti con AWS IAM Identity Center oppure AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Usa l'autenticazione a più fattori (MFA) con ogni account.
- UsaSSL/TLSper comunicare con AWS risorse. Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Configurazione API e registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi di acquisizione AWS attività, vedi Lavorare con i CloudTrail sentieri in AWS CloudTrail Guida per l'utente.
- Utilizzo AWS soluzioni di 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 sono necessari FIPS 140-3 moduli crittografici convalidati per l'accesso AWS tramite un'interfaccia a riga di comando o unAPI, utilizza un endpoint. FIPS Per ulteriori informazioni sugli FIPS endpoint disponibili, vedere Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo vivamente 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 si lavora con CloudWatch Logs o altro Servizi AWS utilizzando la console, API AWS CLI, oppure AWS SDKs. 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. Se fornisci un URL a un server esterno, ti consigliamo vivamente di non includere le informazioni sulle credenziali URL per convalidare la tua richiesta a quel server.

Crittografia a riposo

CloudWatch I registri proteggono i dati inattivi 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 (-) a 256 bit per crittografare i dati di registro inattivi. AES GCM

Se desideri gestire le chiavi utilizzate per crittografare e decrittografare i log, utilizza AWS KMS chiavi. 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 end-to-end la crittografia 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 relativi alle risorse cloud. Le seguenti sezioni 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 AWS IAM Identity Center:

  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 AWS IAM Identity Center .
- 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 a meno che tu non disponga delle autorizzazioni necessarie non puoi creare o accedere alle risorse di Logs. CloudWatch 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 registri. CloudWatch Consigliamo di leggere prima la panoramica.

- Panoramica della gestione delle autorizzazioni di accesso alle risorse Logs CloudWatch
- Utilizzo di politiche basate sull'identità (politiche IAM) per i registri CloudWatch
- CloudWatch Registra il riferimento alle autorizzazioni

Panoramica della gestione delle autorizzazioni di accesso alle risorse Logs CloudWatch

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in: AWS IAM Identity Center

  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 AWS IAM Identity Center .
- 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

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).

A queste risorse e sottorisorse sono associati Amazon Resource Names (ARNs) univoci, come illustrato nella tabella seguente.

Tipo di risorsa	Formato ARN
Gruppo di log	Si utilizzano entrambi i seguenti elementi. Il secondo, con la `:*` fine, è ciò che viene restituit

Tipo di risorsa	Formato ARN
	o dal comando `describe-log-groups` CLI e dall'DescribeLogGroupsAPI. arn:aws:logs: ::log-group: *region account-id log_group_name* arn:aws:logs: *region account-id* ::log-group:: * *log_group_name* Usa la prima versione, senza la parte finale, nelle seguenti situazioni: `:*` • Nel campo `logGroupIdentifier` di immissione in molti CloudWatch Logs APIs. • Sul `resourceArn` campo, nel tagging APIs • Nelle IAM politiche, quando si specifica no le autorizzazioni per TagResource, e UntagResource. ListTagsForResource Usa la seconda versione, con la fine `:*`, per fare riferimento all'ARN quando specifichi le autorizzazioni nelle policy IAM per tutte le altre azioni API.
Flusso di log	arn:aws:logs: :log-group ::log-stream: *region account-id log_group_name log-stream-name*
Destinazione	arn:aws:logs: *region* :destinazione: *account-id destination_name*

Per ulteriori informazioni su ARNs, consulta la IAM User Guide. ARNs Per informazioni sui CloudWatch log ARNs, consulta Amazon Resource Names (ARNs) in Riferimenti generali di Amazon Web Services. Per un esempio di politica che copre CloudWatch i log, consulta. Utilizzo di politiche basate sull'identità (politiche IAM) per i registri CloudWatch

CloudWatch Logs fornisce una serie di operazioni per utilizzare le risorse CloudWatch Logs. Per un elenco di operazioni disponibili, consulta la sezione CloudWatch Registra il riferimento alle autorizzazioni.

Informazioni sulla proprietà delle risorse

L' AWS account possiede le risorse create nell'account, indipendentemente da chi ha creato le risorse. 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 delle risorse. Negli esempi seguenti viene illustrato il funzionamento:

- Se utilizzi le credenziali dell'account root del tuo AWS account per creare un gruppo di log, quest'ultimo è il AWS 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 Logs. CloudWatch Tuttavia, l' AWS account a cui appartiene l'utente è proprietario delle risorse Logs. CloudWatch
- Se crei un ruolo IAM nel tuo AWS account con le autorizzazioni per creare risorse CloudWatch Logs, chiunque possa assumere il ruolo può creare CloudWatch risorse Logs. Il tuo AWS account, a cui appartiene il ruolo, possiede le CloudWatch risorse 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 allegate a una risorsa sono denominate politiche basate sulle risorse. CloudWatch Logs supporta politiche basate sull'identità e politiche basate sulle risorse per le destinazioni, che vengono

utilizzate per abilitare sottoscrizioni tra account. Per ulteriori informazioni, consulta Abbonamenti tra più account e più regioni.

Argomenti

- Autorizzazioni del gruppo di log e Contributor Insights
- Policy basate sulle risorse

Autorizzazioni del gruppo di log e Contributor Insights

Contributor Insights è una funzionalità CloudWatch che consente di analizzare i dati dei 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 dati ad alta cardinalità.

Quando concedi a un utente le `cloudwatch:GetInsightRuleReport` autorizzazioni `cloudwatch:PutInsightRule` and, quell'utente può creare una regola che valuta qualsiasi gruppo di log in CloudWatch Logs e quindi visualizzare i risultati. I risultati possono contenere dati dei collaboratori per tali gruppi di log. Assicurarsi di concedere queste autorizzazioni solo a utenti che devono essere in grado di visualizzare questi dati.

Policy basate sulle risorse

CloudWatch Logs supporta politiche basate sulle risorse per le destinazioni, che puoi utilizzare per abilitare le sottoscrizioni 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 delle risorse alla destinazione utilizzando l'API. PutDestinationPolicy L'esempio seguente consente a un altro AWS account con l'ID account 111122223333 di iscrivere i propri gruppi di log alla destinazione. `arn:aws:logs:us-east-1:123456789012:destination:testDestination`

```
{
  "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 un set di operazioni API. Per concedere le autorizzazioni per queste operazioni API, CloudWatch Logs definisce una serie di azioni che è possibile specificare in una politica. Alcune operazioni API possono richiedere le autorizzazioni per più di un'azione 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 Registra il riferimento alle autorizzazioni.

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, si specifica l'utente, l'account, il servizio o l'altra entità a cui si desidera ricevere le autorizzazioni (si applica solo alle politiche basate sulle risorse). CloudWatch Logs supporta 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 Registra il riferimento alle autorizzazioni

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 ogni 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 le chiavi di contesto delle condizioni AWS globali.

> **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.

Utilizzo di politiche basate sull'identità (politiche IAM) per i registri 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 consultare 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 della 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 registri CloudWatch
- Esempi di policy gestite dal cliente

Di seguito è riportato un esempio di policy delle autorizzazioni:

```
{
  "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 (*) alla fine del valore `Resource` indica che la dichiarazione concede l'autorizzazione per le operazioni `logs:CreateLogGroup`, `logs:CreateLogStream`, `logs:PutLogEvents` e `logs:DescribeLogStreams` su qualsiasi gruppo di log. Per limitare questa autorizzazione a uno specifico gruppo di log, sostituisci il carattere jolly (*) nell'ARN della risorsa con l'ARN del gruppo di log specifico. Per ulteriori informazioni sulle sezioni all'interno della dichiarazione di policy IAM, consulta Riferimento agli elementi di policy IAM nella Guida per l'utente di IAM. Per un elenco che mostra tutte le azioni di CloudWatch Logs, consulta. CloudWatch Registra il riferimento alle autorizzazioni

Autorizzazioni necessarie per utilizzare la console CloudWatch

Affinché un utente possa utilizzare CloudWatch Logs nella CloudWatch console, deve disporre di un set minimo di autorizzazioni che gli consentano di descrivere AWS le altre risorse del proprio

account. AWS Per utilizzare CloudWatch Logs nella CloudWatch console, è necessario disporre delle autorizzazioni dei seguenti servizi:

- CloudWatch
- CloudWatch Registri
- OpenSearch Servizio
- IAM
- Kinesis
- Lambda
- 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 continuare a utilizzare la CloudWatch console, allega anche la policy `CloudWatchReadOnlyAccess` gestita all'utente, come descritto in AWS politiche gestite (predefinite) per i registri CloudWatch .

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso l'API AWS CLI o verso l'API CloudWatch Logs.

Il set completo di autorizzazioni necessarie per lavorare con la CloudWatch console per un utente che non utilizza la console per gestire gli abbonamenti ai registri è:

- cloudwatch: GetMetricData
- orologio nuvoloso: 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:

- Sì: DescribeElasticsearchDomain
- Sì: ListDomainNames
- sono: AttachRolePolicy
- Io sono: CreateRole
- Io sono: GetPolicy
- Io sono: GetPolicyVersion
- Io sono: GetRole

- Io sono: ListAttachedRolePolicies
- Io sono: ListRoles
- cinesi: 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 registri CloudWatch

AWS affronta molti casi d'uso comuni fornendo politiche 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 dei CloudWatch log:

- CloudWatchLogsFullAccess— Garantisce l'accesso completo ai registri. CloudWatch
- CloudWatchLogsReadOnlyAccess— Garantisce l'accesso in sola lettura ai registri. CloudWatch

CloudWatchLogsFullAccess

La CloudWatchLogsFullAccesspolitica garantisce l'accesso completo ai registri. CloudWatch La politica include l'`cloudwatch:GenerateQuery`autorizzazione, in modo che gli utenti con questa politica possano generare una stringa di query di CloudWatch Logs Insights da un prompt in linguaggio naturale. Include le autorizzazioni Amazon OpenSearch Service e IAM per abilitare l'integrazione di CloudWatch Logs con OpenSearch Service per alcune funzionalità. I contenuti completi sono i seguenti:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchLogsFullAccess",
            "Effect": "Allow",
            "Action": [
                "logs:*",
                "cloudwatch:GenerateQuery",
                "opensearch:ApplicationAccessAll",
                "iam:ListRoles",
                "iam:ListUsers",
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "es:ListApplications"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLogsOpenSearchCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "opensearchservice.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsObservabilityCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/*/
AWSServiceRoleForAmazonOpenSearchServerless",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "observability.aoss.amazonaws.com"
```

```
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionRequestAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateCollection"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsApplicationRequestAccess",
            "Effect": "Allow",
            "Action": [
                "es:CreateApplication"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/OpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "OpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionResourceAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:DeleteCollection"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsApplicationResourceAccess",
            "Effect": "Allow",
            "Action": [
                "es:UpdateApplication",
                "es:GetApplication"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/OpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateSecurityPolicy",
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:DeleteSecurityPolicy",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityPolicy",
                "aoss:APIAccessAll"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "logs-collection-*"
                }
            }
```

```
        },
        {
            "Sid": "CloudWatchLogsIndexPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:GetAccessPolicy",
                "aoss:CreateLifecyclePolicy",
                "aoss:DeleteLifecyclePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:index": "logs-collection-*"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsStartDirectQueryAccess",
            "Effect": "Allow",
            "Action": [
                "opensearch:StartDirectQuery"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/logs_datasource_*"
        },
        {
            "Sid": "CloudWatchLogsDQSRequestQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddDirectQueryDataSource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
```

```
        {
            "Sid": "CloudWatchLogsDQSResourceQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:GetDirectQueryDataSource",
                "es:DeleteDirectQueryDataSource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsPassRoleAccess",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService":
"directquery.opensearchservice.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAossTagsAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:TagResource",
                "es:AddTags"
            ],
            "Resource": "arn:aws:aoss:*:*:collection/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
```

```
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsEsApplicationTagsAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddTags"
            ],
            "Resource": "arn:aws:opensearch:*:*:application/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/OpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "OpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsEsDataSourceTagsAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddTags"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        }
    ]
}
```

CloudWatchLogsReadOnlyAccess

La CloudWatchLogsReadOnlyAccesspolitica garantisce l'accesso in sola lettura ai registri. CloudWatch Include l'`cloudwatch:GenerateQuery`autorizzazione, in modo che gli utenti con questa politica possano generare una stringa di query di CloudWatch Logs Insights da un prompt in linguaggio naturale. I contenuti sono come indicato di seguito:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudWatchOpenSearchDashboardsFullAccess

La CloudWatchOpenSearchDashboardsFullAccesspolicy concede l'accesso per creare, gestire ed eliminare integrazioni con il OpenSearch Servizio e per creare dashboard di eliminazione e gestione dei log forniti in tali integrazioni. Per ulteriori informazioni, consulta Analizza con Amazon OpenSearch Service.

I contenuti sono come indicato di seguito:

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

```
            "Effect": "Allow",
            "Action": [
                "logs:ListIntegrations",
                "logs:GetIntegration",
                "logs:DeleteIntegration",
                "logs:PutIntegration",
                "logs:DescribeLogGroups",
                "opensearch:ApplicationAccessAll",
                "iam:ListRoles",
                "iam:ListUsers"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLogsOpensearchReadAPIs",
            "Effect": "Allow",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "es:ListApplications"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsOpensearchCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "opensearchservice.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
```

```
            "Sid": "CloudWatchLogsObservabilityCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
observability.aoss.amazonaws.com/AWSServiceRoleForAmazonOpenSearchServerless",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "observability.aoss.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionRequestAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateCollection"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsApplicationRequestAccess",
            "Effect": "Allow",
            "Action": [
                "es:CreateApplication"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/OpenSearchIntegration": [
```

```
                    "Dashboards"
                ]
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": "OpenSearchIntegration"
            }
        }
    },
    {
        "Sid": "CloudWatchLogsCollectionResourceAccess",
        "Effect": "Allow",
        "Action": [
            "aoss:DeleteCollection"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:CalledViaFirst": "logs.amazonaws.com",
                "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                    "Dashboards"
                ]
            }
        }
    },
    {
        "Sid": "CloudWatchLogsApplicationResourceAccess",
        "Effect": "Allow",
        "Action": [
            "es:UpdateApplication",
            "es:GetApplication"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:CalledViaFirst": "logs.amazonaws.com",
                "aws:ResourceTag/OpenSearchIntegration": [
                    "Dashboards"
                ]
            }
        }
    },
    {
        "Sid": "CloudWatchLogsCollectionPolicyAccess",
        "Effect": "Allow",
```

```
            "Action": [
                "aoss:CreateSecurityPolicy",
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:DeleteSecurityPolicy",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAPIAccessAll",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsIndexPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:GetAccessPolicy",
                "aoss:CreateLifecyclePolicy",
                "aoss:DeleteLifecyclePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:index": "cloudwatch-logs-*",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
```

```
                }
            }
        },
        {
            "Sid": "CloudWatchLogsDQSRequestQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsStartDirectQueryAccess",
            "Effect": "Allow",
            "Action": [
                "opensearch:StartDirectQuery",
                "opensearch:GetDirectQuery"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*"
        },
        {
            "Sid": "CloudWatchLogsDQSResourceQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:GetDirectQueryDataSource",
                "es:DeleteDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
```

```
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsPassRoleAccess",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService":
"directquery.opensearchservice.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAossTagsAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:TagResource",
                "es:AddTags"
            ],
            "Resource": "arn:aws:aoss:*:*:collection/*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsEsApplicationTagsAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddTags"
```

```
            ],
            "Resource": "arn:aws:opensearch:*:*:application/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/OpenSearchIntegration": [
                        "Dashboards"
                    ],
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "OpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsEsDataSourceTagsAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddTags"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ],
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        }
    ]
}
```

CloudWatchOpenSearchDashboardAccess

La CloudWatchOpenSearchDashboardAccesspolitica consente l'accesso alla visualizzazione dei dashboard dei registri venduti creati con analisi. Amazon OpenSearch Service Per ulteriori informazioni, consulta Analizza con Amazon OpenSearch Service.

> **Important**
>
> Oltre a concedere questa politica, per consentire a un ruolo o a un utente di visualizzare i dashboard dei log venduti, è necessario specificarli anche quando si crea l'integrazione con Service. OpenSearch Per ulteriori informazioni, consulta Fase 1: Creare l'integrazione con Service OpenSearch .

I contenuti di CloudWatchOpenSearchDashboardAccesssono i seguenti:

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "CloudWatchOpenSearchDashboardsIntegration",
            "Effect": "Allow",
            "Action": [
                "logs:ListIntegrations",
                "logs:GetIntegration",
                "logs:DescribeLogGroups",
                "opensearch:ApplicationAccessAll",
                "iam:ListRoles",
                "iam:ListUsers"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLogsOpensearchReadAPIs",
            "Effect": "Allow",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "es:ListApplications"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAPIAccessAll",
```

```
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsDQSCollectionPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aoss:collection": "cloudwatch-logs-*"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsApplicationResourceAccess",
            "Effect": "Allow",
            "Action": [
                "es:GetApplication"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/OpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsDQSResourceQueryAccess",
```

```
            "Effect": "Allow",
            "Action": [
                "es:GetDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsDirectQueryStatusAccess",
            "Effect": "Allow",
            "Action": [
                "opensearch:GetDirectQuery"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*"
        }
    ]
}
```

CloudWatchLogsCrossAccountSharingConfiguration

La CloudWatchLogsCrossAccountSharingConfigurationpolitica consente l'accesso alla creazione, alla gestione e alla visualizzazione dei collegamenti di Observability Access Manager per la condivisione delle risorse di CloudWatch Logs tra account. Per ulteriori informazioni, consulta CloudWatch osservabilità tra account.

I contenuti sono come indicato di seguito:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Link",
                "oam:ListLinks"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

CloudWatch Registra gli aggiornamenti delle politiche gestite AWS

Visualizza i dettagli sugli aggiornamenti alle politiche AWS gestite per CloudWatch Logs da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei documenti di CloudWatch Logs.

Modifica	Descrizione	Data
CloudWatchLogsFullAccess: aggiorna a una policy esistente.	CloudWatch Registra le autorizzazioni aggiunte a. CloudWatchLogsFullAccess Sono state aggiunte le autorizzazioni per Amazon	1 dicembre 2024

Modifica	Descrizione	Data
	OpenSearch Service e IAM, per consentire l'integrazione di CloudWatch Logs con OpenSearch Service per alcune funzionalità.	
CloudWatchOpenSearchDashboardsFullAccess— Nuova politica IAM.	CloudWatch Logs ha aggiunto una nuova policy IAM, CloudWatchOpenSearchDashboardsFullAccess.- Questa policy consente l'accesso per creare, gestire ed eliminare integrazioni con OpenSearch Service e per creare, gestire ed eliminare dashboard di log vendute in tali integrazioni. Per ulteriori informazioni, consulta Analizza con Amazon OpenSearch Service.	1 dicembre 2024
CloudWatchOpenSearchDashboardAccess— Nuova policy IAM.	CloudWatch Logs ha aggiunto una nuova policy IAM, CloudWatchOpenSearchDashboardAccess.- Questa policy consente l'accesso alla visualizzazione dei dashboard dei log forniti da. Amazon OpenSearch Service Per ulteriori informazioni, consulta Analizza con Amazon OpenSearch Service.	1 dicembre 2024

Modifica	Descrizione	Data
CloudWatchLogsFullAccess: aggiorna a una policy esistente.	CloudWatch Logs ha aggiunto un'autorizzazione a. CloudWatchLogsFullAccess L'`cloudwatch:GenerateQuery` autorizzazione è stata aggiunta, in modo che gli utenti con questo criterio possano generare una stringa di query di 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 questo criterio possano generare una stringa di query di CloudWatch Logs Insights da un prompt in linguaggio naturale.	27 novembre 2023

Modifica	Descrizione	Data
CloudWatchLogsReadOnlyAccess: aggiornamento a una policy esistente	CloudWatch Registra le autorizzazioni aggiunte a. CloudWatchLogsReadOnlyAccess Le `logs:StopLiveTail` autorizzazioni `logs:StartLiveTail` e sono state aggiunte in modo che gli utenti con questo criterio 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 politica che consente di gestire i link di osservabilità CloudWatch tra account che condividono i gruppi di log di Logs. CloudWatch Per ulteriori informazioni, consulta osservabilità tra account CloudWatch	27 novembre 2022

Modifica	Descrizione	Data
CloudWatchLogsReadOnlyAccess: aggiornamento a una policy esistente	CloudWatch Registra le autorizzazioni aggiunte a. CloudWatchLogsReadOnlyAccess Le `oam:ListAttachedLinks` autorizzazioni `oam:ListSinks` e sono state aggiunte in modo che gli utenti con questo criterio possano utilizzare la console per visualizzare i dati condivisi dagli account di origine in CloudWatch modo osservabile 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 autorizzazioni per varie CloudWatch azioni di Logs. Queste politiche funzionano quando si utilizza l'API CloudWatch Logs o il AWS SDKs. AWS CLI

Esempi

- Esempio 1: consenti l'accesso completo ai registri CloudWatch
- Esempio 2: consentire l'accesso in sola lettura ai registri CloudWatch
- Esempio 3: consentire l'accesso a un gruppo di log

Esempio 1: consenti l'accesso completo ai registri CloudWatch

La seguente politica consente a un utente di accedere a tutte le azioni di CloudWatch Logs.

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

Esempio 2: consentire l'accesso in sola lettura ai registri CloudWatch

AWS fornisce una CloudWatchLogsReadOnlyAccesspolitica che consente l'accesso in sola lettura ai dati dei registri. CloudWatch Questa policy include le seguenti autorizzazioni:

```
{
    "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

La policy seguente consente a un utente di leggere e scrivere eventi di log in un gruppo di log specificato.

> **Important**
>
> I caratteri `:*` alla fine del nome del gruppo di log sulla riga `Resource` sono necessari per indicare che la policy si applica a tutti i flussi di log in questo gruppo di log. Se ometti `:*`, la policy non verrà applicata.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "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`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

Le operazioni StopQuerye le StopLiveTailAPI non interagiscono con AWS le risorse nel senso tradizionale. Non restituiscono né inseriscono alcun dato, né modificano una risorsa in alcun modo. Funzionano invece 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.

```
{
    "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 Registra il riferimento alle autorizzazioni

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, è possibile 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 le chiavi di condizione AWS-wide nelle politiche di CloudWatch Logs per esprimere condizioni. Per un elenco completo delle chiavi AWS-wide, consulta AWS Global and IAM 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:CreateLogGrouplogs: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`

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
	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` 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 politica delle risorse CloudWatch di Logs.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
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.
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.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Necessario per visualizzare l'elenco delle definizioni delle query salvate in CloudWatch Logs Insights.
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 Logs. CloudWatch
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.
GetQueryResults	`logs:GetQueryResults` Necessario per recuperare i risultati delle interrogazioni di CloudWatch Logs Insights.
ListEntitiesForLogGroup (autorizzazione solo per console) CloudWatch	`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 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.
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.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
PutLogEvents	`logs:PutLogEvents` Necessaria per caricare un batch di eventi di log in un flusso di eventi.
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.
PutResourcePolicy	`logs:PutResourcePolicy` Necessario per creare una politica delle risorse CloudWatch di 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.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
StopQuery	`logs:StopQuery` Necessario per interrompere una query di 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 registri 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 ai log. CloudWatch I ruoli collegati ai servizi sono predefiniti da CloudWatch Logs e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio rende la configurazione di CloudWatch Logs più efficiente perché non è necessario aggiungere manualmente le autorizzazioni necessarie. CloudWatch Logs definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo Logs può assumere tali ruoli. CloudWatch 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 che hanno Sì nella colonna Ruolo collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni relative ai ruoli collegati al servizio per Logs 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 dai servizi AWS.

Il AWSServiceRoleForLogDeliveryservice-linked role si fida che i seguenti servizi assumano il ruolo:

- `logs.amazonaws.com`

La politica di autorizzazione dei ruoli consente a CloudWatch Logs di completare le seguenti azioni sulle risorse specificate:

- Azione: `firehose:PutRecord` e `firehose:PutRecordBatch` su tutti gli stream Firehose che hanno un tag con una `LogDeliveryEnabled` chiave con un valore di. `True` Questo tag viene collegato automaticamente a uno stream Firehose quando si crea un abbonamento per inviare i log a Firehose.

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 Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato al servizio per Logs CloudWatch

Non è necessario creare manualmente un ruolo collegato ai servizi. Quando configurate i log da inviare direttamente a uno stream Firehose nell'API, AWS CLI nella o AWS Management Console nell'API CloudWatch , Logs crea automaticamente AWS il ruolo collegato al servizio.

Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando configuri nuovamente i log per essere inviati direttamente 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 I registri non consentono di modificare AWSServiceRoleForLogDelivery, o qualsiasi 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 la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato al servizio per Logs CloudWatch

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il 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 al servizio 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 AWSServiceRoleForLogDeliveryRuolo collegato ai servizi

- Smetti di inviare i log direttamente agli stream di Firehose.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, l'o l' AWS API per eliminare il AWSServiceRoleForLogDeliveryruolo collegato al servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi.

Regioni supportate per i ruoli collegati al servizio CloudWatch Logs

CloudWatch Logs supporta l'utilizzo di ruoli collegati al servizio in tutte le AWS regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta CloudWatch Registra regioni ed endpoint.

Convalida della conformità per Amazon Logs CloudWatch

I revisori di terze parti valutano la sicurezza e la conformità di Amazon CloudWatch Logs nell'ambito di diversi programmi di AWS conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta AWS Services in Scope by Compliance ProgramAWS Program. Per informazioni generali, consulta Programmi di conformitàAWS.

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La tua responsabilità di conformità quando utilizzi Amazon CloudWatch Logs è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Security and Compliance Quick Start Guides (Guide Quick Start Sicurezza e compliance): queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Progettazione per la sicurezza e la conformità HIPAA su Amazon Web Services: questo white paper descrive come le aziende possono utilizzare AWS per creare applicazioni conformi allo standard HIPAA.
- AWS Risorse per la conformità Risorse perAWS : questa raccolta di cartelle di lavoro e guide potrebbe riguardare il tuo settore e la tua area geografica.
- Evaluating Resources with Rules nella AWS Config Developer Guide: AWS Configvaluta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente di verificare la conformità agli standard e alle best practice del settore della sicurezza. AWS

Resilienza in Amazon CloudWatch Logs

L'infrastruttura globale di AWS è basata su regioni e zone di disponibilità AWS. 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.

Per ulteriori informazioni sulle regioni e le zone di disponibilità AWS, consulta Infrastruttura globale di AWS.

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 le API chiamate AWS pubblicate per accedere ai CloudWatch registri attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS) come (Ephemeral Diffie-Hellman) o DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale. IAM In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

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.

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***.logs. Non è necessario modificare alcuna impostazione per Logs. CloudWatch Per ulteriori informazioni, consulta Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

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. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione, seleziona Endpoints (Endpoint).
3. Se non hai ancora creato l'endpoint for CloudWatch Logs, scegli Crea endpoint. Quindi seleziona com.amazonaws.***Region***.logs e scegli Create endpoint (Crea endpoint).
4. Seleziona l'endpoint com.amazonaws.***Region***.logs e scegli la scheda Policy nella parte 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 VPC o endpoint VPC 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.

Registrazione dei CloudWatch log API e delle operazioni della console AWS CloudTrail

Amazon CloudWatch Logs è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, un ruolo o un Servizio AWS. CloudTrail acquisisce le API chiamate per CloudWatch i registri come eventi. Le chiamate acquisite includono le chiamate dalla console CloudWatch Logs e le chiamate in codice alle operazioni Logs. CloudWatch API 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 di IAM Identity Center.
- 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 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 AWS. Per ulteriori informazioni, consulta Lavorare con la cronologia degli CloudTrail eventi nella AWS CloudTrail Guida per l'utente. Non ci sono CloudTrail costi per la visualizzazione della cronologia degli eventi.

Per una registrazione continua degli eventi nel tuo Account AWS negli ultimi 90 giorni, crea un trail o un archivio dati di eventi CloudTrailLake.

CloudTrail sentieri

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Tutti i percorsi creati utilizzando AWS Management Console sono multiregionali. È possibile creare un percorso a regione singola o multiregione utilizzando AWS CLI. La creazione di un percorso multiregionale è consigliata perché consente di registrare tutte le attività Regioni AWS nel tuo account. Se crei un percorso a regione singola, puoi visualizzare solo gli eventi registrati nel percorso Regione AWS.

Per ulteriori informazioni sui sentieri, consulta Creazione di un percorso per il Account AWSe Creazione di un percorso per un'organizzazione in AWS CloudTrail Guida per l'utente.

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 AWS CloudTrail Prezzi. Per informazioni sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

CloudTrail Archivi di dati sugli eventi sul lago

CloudTrail Lake ti consente di eseguire query SQL basate sui tuoi eventi. CloudTrail Lake converte gli eventi esistenti in JSON formato basato su righe in formato Apache. ORC ORCè un formato di archiviazione colonnare 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 Lago nel AWS CloudTrail Guida per l'utente.

CloudTrail Gli archivi di dati e le richieste di Lake Event 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 AWS CloudTrail Prezzi.

CloudWatch Logs supporta la registrazione delle seguenti azioni come eventi nei CloudTrail file di registro:

- CancelExportTask
- CreateExportTask
- CreateLogGroup
- CreateLogStream
- DeleteDestination
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteRetentionPolicy

- DeleteSubscriptionFilter
- PutDestination
- PutDestinationPolicy
- PutMetricFilter
- PutResourcePolicy
- PutRetentionPolicy
- PutSubscriptionFilter
- StartQuery
- StopQuery
- TestMetricFilter

Solo gli elementi di richiesta vengono registrati CloudTrail per queste CloudWatch azioni di registro: API

- DescribeDestinations
- DescribeExportTasks
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeResourcePolicies
- DescribeSubscriptionFilters
- FilterLogEvents
- GetLogEvents
- GetLogGroupFields
- GetLogRecord
- GetQueryResults

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali root o IAM utente.

- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata fatta da un altro AWS servizio.

Per ulteriori informazioni, consulta l'CloudTrail userIdentityelemento.

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"
}
```

Comprensione delle 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 stack ordinata delle API chiamate 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"
}
```

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 versione 2 (IMDSv2), devi utilizzare il nuovo agente unificato. CloudWatch Tuttavia, anche se non lo utilizziIMDSv2, ti consigliamo vivamente di utilizzare il nuovo agente unificato anziché l' CloudWatch agente Logs obsoleto. CloudWatch Per informazioni sul nuovo CloudWatch agente unificato, consulta Raccolta di metriche e log dall'istanza EC2 Amazon 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 automatico per inviare dati di log a CloudWatch Logs da istanze AmazonEC2. 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.pylogging-config-fileformatthon.org/2/library/logging.config.html#). 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 reader e publisher impostandolo su un valore predefinito pari a. WARNING 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 con gzip per inviare payload compressi ai registri. CloudWatch Ciò riduce l'CPUutilizzo, 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, A-Z, 0-9, '_' (trattino basso), '-' (trattino), '/' (barra) e '.' (punto).

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(). Anche l'offset del fuso orario (%z) è supportato anche se non è supportato fino a python 3.2, [+-] senza due punti (:). HHMM 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: UTC offset nella forma + o -. HHMM 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 L'impostazione predefinita èLOCAL, che viene utilizzata se il fuso orario non può essere dedotto in base a datetime_format.

file

Specificare i file di registro che si desidera inviare a Logs. CloudWatch File può puntare a un determinato file o più file (tramite 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 in 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 i proxy. HTTP

> **Note**
>
> HTTPi proxy sono supportati nella versione 1.3.8 o successiva del awslogs-agent-setup file.py.

Per utilizzare l'agente Logs con i proxy CloudWatch HTTP

1. Esegui una di queste 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 EC2 metadati Amazon sulle EC2 istanze, usa --no-proxy 169.254.169.254 (consigliato). Per ulteriori informazioni, consulta i metadati dell'istanza e i dati utente nella Amazon EC2 User Guide.

Nei valori per `http-proxy` e`https-proxy`, specifichi l'interoURL.

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
```

CloudWatch Compartimentazione dei file di configurazione dell'agente Logs

Se stai usando awslogs-agent-setup la 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/. CloudWatch All'avvio, l'agente Logs include tutte le configurazioni di flusso in questi file di configurazione aggiuntivi. Le proprietà di configurazione nella sezione [general] devono essere definite nel file di configurazione principale (/var/awslogs/etc/awslogs.conf) e vengono ignorate in tutti i file di configurazione aggiuntivi in /var/awslogs/etc/config/.

Se non disponi di una directory /var/awslogs/etc/config/ perché hai installato l'agente con rpm, puoi in alternativa utilizzare la directory /etc/awslogs/config/.

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 Agente Logs FAQ

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 in /var/log/syslog.log.1. Se /var/log/syslog.log.1 esiste già da una rotazione precedente, viene rinominato in /var/log/syslog.log.2.
- Troncamento del file di log originale in vigore dopo la creazione di una copia. Ad esempio, /var/log/syslog.log viene copiato in /var/log/syslog.log.1 e /var/log/syslog.log viene troncato. 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, /var/log/syslog.log.2014-01-01 rimane e viene creato /var/log/syslog.log.2014-01-02.

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 hai utilizzato uno script di installazione per installare l'agente CloudWatch Logs, puoi utilizzare /var/awslogs/bin/awslogs-version.sh per verificare quale versione dell'agente stai utilizzando. 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 controllare 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 log contiene un timestamp come "2014-01-02T13:13:01Z", allora 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 CloudWatch Logs scrive un avviso nel suo registro quando i dati vengono ignorati.

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'arresto dell'agente causa perdita di dati o duplicati?

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

L'agente CloudWatch Logs richiede le `PutLogEvents` operazioni `CreateLogGroupCreateLogStream`,`DescribeLogStreams`, e. Se utilizzi l'ultimo agente, `DescribeLogStreams` non è necessario. Osserva la policy IAM di esempio seguente.

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

```
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

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

Nella tua IAM politica, puoi limitare l'agente solo alle seguenti operazioni:`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`.

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	Descrizione
`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	Descrizione
`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 riutilizzabile (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
`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

Metrica	Descrizione
`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 CloudWatch incorporato, vedere Specificazione: formato metrico incorporato. 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
`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

Metrica	Descrizione
`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
`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

Metrica	Descrizione
`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
`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	Descrizione
`LogGroupName`	Il nome del gruppo di CloudWatch log Logs per il quale visualizz are le metriche.

Dimensione	Descrizione
`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 (?).

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	Descrizione
`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`

Parametro	Descrizione
`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
`TransformedLogEvents`	Il numero di eventi di registro trasformati. Unità: nessuna Statistica valida: Sum

Le seguenti dimensioni vengono utilizzate dalle metriche dei trasformatori.

Dimensione	Descrizione
`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`

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.

Metriche

Parametro	Descrizione
`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	Descrizione
`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`.
`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.

Risorsa	Quota predefinita
Politiche a livello di account	Una politica di filtro di abbonamento a livello di account per regione per account. Una politica di protezione dei dati a livello di account per regione per account. 20 politiche di indicizzazione dei campi a livello di account per account. I prefissi dei nomi dei gruppi di log a cui si applicano non possono sovrapporsi. Queste quote non possono essere modificate.
Rilevatori di anomalie	500 rilevatori di anomalie per account. È possibile richiedere un aumento della quota.
Dimensione batch	La dimensione massima di un batch è di 1.048.576 byte. Questa dimensione viene calcolata come la somma di tutti i messaggi di eventi in UTF-8, più 26 byte per ogni log eventi. Questa quota non può essere modificata.
Archiviazione di dati	Fino a 5 GB di spazio di archiviazione di dati gratis. Questa quota non può essere modificata.

Risorsa	Quota predefinita
CreateLogGroup	10 transazioni al secondo (TPS/account/Region), dopodiché le transazioni vengono limitate. È possibile richiedere un aumento della quota.
CreateLogStream	50 transazioni al secondo (TPS/account/Region), dopodiché le transazioni vengono limitate. È possibile richiedere un aumento della quota.
Identificatori di dati personalizzati	Ogni politica di protezione dei dati può includere fino a 10 identificatori di dati personalizzati. È possibile richiedere un aumento della quota. Ogni espressione regolare che definisce un identificatore di dati personalizzato può includere fino a 200 caratteri. Questa quota non può essere modificata.
DeleteLogGroup	10 transazioni al secondo (TPS/account/Region), dopodiché le transazioni vengono limitate. È possibile richiedere un aumento della quota.
DeleteLogStream	15 transazioni al secondo (TPS/account/Region), dopodiché le transazioni vengono limitate. È possibile richiedere un aumento della quota.
DescribeLogGroups	10 transazioni al secondo ()TPS/account/Region. È possibile richiedere un aumento della quota.
DescribeLogStreams	25 transazioni al secondo (TPS/account/Region). È possibile richiedere un aumento della quota.
Campi di log rilevati	CloudWatch Logs Insights è in grado di rilevare un massimo di 1000 campi di eventi di registro in un gruppo di log. Questa quota non può essere modificata. Per ulteriori informazioni, consulta Registri supportati e campi rilevati.

Risorsa	Quota predefinita
Campi di log estratti nei log JSON	CloudWatch Logs Insights può estrarre un massimo di 200 campi di eventi di registro da un registro JSON. Questa quota non può essere modificata. Per ulteriori informazioni, consulta Registri supportati e campi rilevati.
Attività di esportazione	Un'attività di esportazione (in esecuzione o in attesa) attiva alla volta, per ogni account. Questa quota non può essere modificata.
Indici di campo	Fino a 20 campi indicizzati per policy. Questa quota non può essere modificata.
FilterLogEvents	25 richieste al secondo negli Stati Uniti orientali (Virginia settentrionale). 5 richieste al secondo nelle seguenti regioni: • Asia Pacifico (Giacarta) • Asia Pacifico (Osaka-Locale) • Europa (Francoforte) • Canada occidentale (Calgary) • Israele (Tel Aviv) 10 richieste al secondo in altre regioni. Questa quota non può essere modificata.

Risorsa	Quota predefinita
GetLogEvents	30 richieste al secondo in Europa (Parigi). 10 richieste al secondo nelle seguenti regioni: • US West (Oregon) • Asia Pacifico (Giacarta) • Asia Pacifico (Osaka-Locale) • Canada occidentale (Calgary) • Europa (Irlanda) • Europa (Francoforte) • Israele (Tel Aviv) 25 richieste al secondo in tutte le altre regioni. Questa quota non può essere modificata. In caso di costante elaborazione di nuovi dati, è consiglia bile sottoscrivere un abbonamento. In caso di necessità dei dati storici, è consigliabile esportare i dati in Amazon S3.
Dati in ingresso	Fino a 5 GB di dati in ingresso gratis. Questa quota non può essere modificata.
Sessioni simultanee di Live Tail.	15 sessioni simultanee. È possibile richiedere un aumento della quota.
Live Tail: gruppi di log cercati in una sessione.	Massimo 10 gruppi di log scansionati in una sessione Live Tail. Questa quota non può essere modificata.
Dimensione del log eventi	256 KB (massimo). Questa quota non può essere modificata.

Risorsa	Quota predefinita
Gruppi di log	1.000.000 gruppi di log per account e per Regione. È possibile richiedere un aumento della quota. Non vi è alcuna quota per il numero di flussi di log che possono appartenere a un gruppo di log.
Filtri di parametri	100 per gruppo di log. Questa quota non può essere modificata.
Parametri del formato dei parametri incorporato	100 parametri per log eventi e 30 dimensioni per parametro. Per ulteriori informazioni sul formato metrico incorporato, consulta Specificazione: Embedded Metric Format nella Amazon CloudWatch User Guide.
PutLogEvents	La dimensione massima del batch di una PutLogEvents richiesta è di 1 MB. Questa dimensione viene calcolata come la somma di tutti i messaggi di eventi in UTF-8, più 26 byte per ogni log eventi. 5000 transazioni al secondo per account per regione È possibile richiedere un aumento della quota di limitazione al secondo utilizzando il servizio. Service Quotas
Timeout di esecuzione della query	Le query in CloudWatch Logs Insights scadono dopo 60 minuti. Questo limite di tempo non può essere modificato.
Gruppi di log di query	È possibile interrogare un massimo di 50 gruppi di log in una singola query di CloudWatch Logs Insights, quando si specificano i gruppi di log singolarmente. Questa quota non può essere modificata. Se si utilizzano i criteri dei gruppi di log per scegliere i gruppi di log in base ai prefissi dei nomi o si sceglie di interrogare «tutti i gruppi di log», è possibile interrogare fino a 10.000 gruppi di log in una singola query.

Risorsa	Quota predefinita
Query simultanee	Per i gruppi di log di classe Standard, un massimo di 30 interrogazioni simultanee di CloudWatch Logs Insights, incluse le query che sono state aggiunte ai dashboard. Questo massimo di 30 si applica al numero totale di query simultanee, indipendentemente dal linguaggio di query utilizzato. Solo 15 di queste query simultanee possono essere in OpenSearch Service PPL e/o Service SQL. OpenSearch Per i gruppi di log delle classi Infrequent Access, un massimo di 5 query simultanee di CloudWatch Logs Insights, incluse le query che sono state aggiunte ai dashboard. Queste quote non possono essere modificate.
Query generate dal linguaggio naturale	Fino a cinque richieste di interrogazione generate simultaneamente in linguaggio naturale.
Richiedi disponibilità	Le query create nella console sono disponibili per 30 giorni, tramite il comando Cronologia. Questo periodo di disponibilità non può essere modificato. Le definizioni delle query create utilizzando PutQueryDefinitionnon hanno scadenza.
Disponibilità dei risultati delle query	I risultati di una query sono disponibili per 7 giorni. Questo periodo di disponibilità non può essere modificato.
Risultati della query visualizzati nella console	Sulla console vengono visualizzate fino a 10.000 righe di risultati delle query.

Risorsa	Quota predefinita
Espressioni regolari	Fino a 5 modelli di filtro contenenti espressioni regolari per ogni gruppo di log durante la creazione di filtri di parametri o filtri di sottoscrizione. Questa quota non può essere modificata. Fino a 2 espressioni regolari per ogni modello di filtro durante la creazione di un modello di filtro delimitato o JSON per filtri di parametri e filtri di sottoscrizione o quando si filtrano log eventi.
Policy delle risorse	Fino a 10 politiche relative CloudWatch alle risorse di registro per regione per account. Questa quota non può essere modificata.
Query salvate	Puoi salvare fino a 1000 query di CloudWatch Logs Insights, per regione per account. Questa quota non può essere modificata.
Filtri di sottoscrizione	2 per gruppo di log. Questa quota non può essere modificata.
Transformers	Un trasformatore di log può avere un massimo di 5 processori di tipo parser. Può avere un massimo di 20 processori in totale. Ogni gruppo di log può avere un solo trasformatore a livello di gruppo di log. Ogni account può avere fino a 20 trasformatori a livello di account. Nessuno di questi trasformatori può essere applicato a prefissi di gruppi di log identici o sovrapposti. Nessuna di queste quote può essere modificata.

Gestione delle quote del CloudWatch servizio 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.

AWS Management Console

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 ogni versione della CloudWatch Logs User Guide, a partire da giugno 2018. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi iscriverti a un RSS feed.

Modifica	Descrizione	Data
CloudWatchOpenSearchDashboardAccesse CloudWatchOpenSearchDashboardsFullAccesssono nuove IAM politiche	CloudWatch Logs ha aggiunto due nuove IAM politiche CloudWatchOpenSearchDashboardsFullAccesse CloudWatchOpenSearchDashboardAccess. CloudWatchOpenSearchDashboardsFullAccessconcede l'autorizzazione a creare e gestire integrazioni con il Servizio. OpenSearch CloudWatchOpenSearchDashboardAccessconcede l'accesso alla visualizzazione dei dashboard dei registri venduti creati in queste integrazioni. Per ulteriori informazioni, consulta le dashboard di Vending log basate su Amazon OpenSearch Service.	1 dicembre 2024
CloudWatchLogsFullAccesspolitica aggiornata	CloudWatch Registra le autorizzazioni aggiunte per Amazon OpenSearch Service e IAM alla CloudWatchLogsFullAccesspolitica, per abilitare l'integrazione	1 dicembre 2024

	di CloudWatch Logs con il OpenSearch servizio per alcune funzionalità.	
CloudWatch Logs Insights aggiunge nuovi tipi di struttura alla sintassi della query	CloudWatch Logs Insights aggiunge il `unnest` comando e due JSON funzioni, che consentono di utilizzare le JSON stringhe come mappe ed elenchi. Per ulteriori informazioni, vedere Tipi di struttura.	21 novembre 2024
CloudWatch Logs supporta la trasformazione dei log durante l'ingestione dei log	Puoi creare trasformatori di log in grado di modificar e gli eventi di registro al momento dell'inserimento, aiutandoti a normalizzare i log in diversi formati e da diverse fonti in formati coerenti e ricchi di contesto. Per ulteriori informazioni, consulta Trasformare i log durante l'ingestione.	20 novembre 2024

CloudWatch Logs Insights aggiunge l'indicizzazione dei campi	CloudWatch Logs Insights ha aggiunto il supporto per l'indicizzazione dei campi dei log. Quando si utilizza quindi 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. Per ulteriori informazioni, vedere 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 Utilizzare il linguaggio naturale per generare e aggiornare le query di CloudWatch Logs Insights.	20 giugno 2024
CloudWatchLogsReadOnlyAccesspolitica aggiornata	CloudWatch Logs ha aggiunto l'`cloudwatch:GenerateQuery` autorizzazione a CloudWatchLogsReadOnlyAccess, in modo che gli utenti con questo criterio possano generare una stringa di query di CloudWatch Logs Insights da un prompt in linguaggio naturale.	26 novembre 2023

CloudWatchLogsFull Accesspolitica aggiornata	CloudWatch Logs ha aggiunto l'`cloudwatch:GenerateQuery` autorizzazione a CloudWatchLogsFullAccess, in modo che gli utenti con questo criterio possano generare una stringa di query di CloudWatch Logs Insights da un prompt in linguaggio naturale.	26 novembre 2023
CloudWatch Logs aggiunge l'analisi del pattern di log	CloudWatch Logs ora analizza i modelli negli eventi di registro ogni volta che si esegue una query di CloudWatch Logs Insights. Per ulteriori informazioni, vedere Pattern analysis.	26 novembre 2023
CloudWatch Logs aggiunge il rilevamento delle anomalie nei registri	È possibile creare un rilevatore di anomalie di registro 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 usare CloudWatch Logs Insights per confrontare le modifiche degli eventi di registro nel tempo. Per ulteriori informazioni, consulta Confronta (diff) con gli 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 poter disporre di un'opzione conveniente per i registri a cui si accede raramente e di un'opzion e completa per i log che richiedono il 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 Utilizzare il linguaggio naturale per generare e aggiornare le query di CloudWatch Logs Insights.	26 novembre 2023
CloudWatch Logs aggiunge il supporto alla sintassi del pattern di filtro delle 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 pattern di filtro nella Amazon CloudWatch Logs User Guide.	13 novembre 2023

CloudWatch Logs aggiunge la sintassi del pattern di filtro delle espressioni regolari, il supporto 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 del pattern di filtro nella Amazon CloudWatch Logs User Guide.	5 settembre 2023
CloudWatch Logs Insights aggiunge un comando pattern	Ora puoi utilizzare pattern nelle tue 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 dedup	È ora possibile utilizzare dedup nelle query di CloudWatch Logs Insights per rimuovere i risultati duplicati in base a valori specifici nei campi specificati. Per ulteriori informazioni, consulta dedup nella Amazon CloudWatch Logs User Guide.	20 giugno 2023

Policy di protezione dei dati a livello di account	Ora puoi impostare le policy di protezione dei dati a livello di account. Queste policy a livello di account possono verificare e mascherare le informazioni riservate dei log eventi in tutti i gruppi di log dell'account. Per ulteriori informazioni, consulta Aiutare a proteggere 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, quindi puoi 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 evidenziare 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 Registra le autorizzazioni aggiunte a. CloudWatchLogsRead OnlyAccess Le `logs:StopLiveTail` autorizzazioni `logs:StartLiveTail` e sono state aggiunte in modo che gli utenti con questo criterio possano utilizzar e 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	È possibile utilizzare CloudWatch Logs Insights per cercare e analizzare in modo interattivo i dati di registro. Per ulteriori informazioni, consulta Analizza i dati di log con CloudWatch Logs Insights nella Amazon CloudWatch Logs User Guide.	27 novembre 2018
Support per VPC endpoint Amazon	Ora puoi stabilire una connessione privata tra i tuoi 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	Descrizione	Data di rilascio
Endpoint VPC di interfaccia	In alcune regioni, puoi utilizzare un VPC endpoint di interfaccia per impedire che il traffico tra Amazon VPC e CloudWatch Logs esca dalla rete Amazon. Per ulteriori informazioni, consulta Utilizzo dei CloudWatch log con endpoint VPC di interfaccia.	7 marzo 2018
Registri delle interrogazioni di Route 53 DNS	È possibile utilizzare CloudWatch Logs per archiviare i registri relativi alle DNS query ricevute da Route 53. Per ulteriori informazioni, consulta Che cos'è Amazon CloudWatch Logs? la sezione Logging DNS Queries nella Amazon Route 53 Developer Guide.	7 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
Aggiunto AWS CloudTrail supporto per Amazon CloudWatch Logs e nuove metriche	È stato aggiunto il AWS CloudTrail supporto per Logs. CloudWatch Per ulteriori informazioni, consulta Registrazione dei CloudWatch log API e delle operazioni della console AWS CloudTrail.	10 marzo 2016

Modifica	Descrizione	Data di rilascio
CloudWatch 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 dei CloudWatch log in Amazon S3. Per ulteriori informazioni, consulta Esportazione di dati di log in Amazon S3.	7 dicembre 2015
Aggiunto il supporto per gli eventi AWS CloudTrai l registrati in Amazon CloudWatch Logs	Puoi creare allarmi CloudWatch e ricevere notifiche relative a particolari API attività acquisite da CloudTrail e utilizzare la notifica per eseguire la risoluzione dei problemi.	10 novembre 2014
Aggiunto il supporto per Amazon CloudWatch Logs	Puoi utilizzare Amazon CloudWatch Logs per monitorare, archiviare e accedere a sistemi, applicazioni e file di log personalizzati da istanze Amazon Elastic Compute Cloud EC2 (Amazon) o altre fonti. Puoi quindi recuperare i dati di log associati da CloudWatch Logs utilizzando la CloudWatch console Amazon, i comandi CloudWatch Logs in o Logs. AWS CLI CloudWatch SDK 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.