aws

Guida per l'utente

Amazon EventBridge

Amazon EventBridge: Guida per l'utente

Table of Contents

Che cos'è Amazon EventBridge?

EventBridge è un servizio serverless che utilizza eventi per connettere tra loro i componenti delle applicazioni, semplificando la creazione di applicazioni scalabili basate su eventi. L'architettura basata su eventi è uno stile di creazione di sistemi software ad accoppiamento debole che interagiscono emettendo e rispondendo a eventi. L'architettura basata su eventi può aiutarti a potenziare l'agilità e creare applicazioni affidabili e scalabili.

Utilizza EventBridge per instradare eventi da origini come applicazioni, servizi AWS e software di terze parti sviluppati internamente ad applicazioni consumer in tutta l'organizzazione. EventBridge offre modi semplici e coerenti per importare, filtrare, trasformare e distribuire eventi in modo da poter creare applicazioni rapidamente.

Nel seguente video viene fornita una breve introduzione alle funzionalità di Amazon EventBridge:

EventBridge include due modi per elaborare gli eventi: router di eventi e pipe.

- I router di eventi sono router che ricevono eventi e li distribuiscono a nessuna o a più destinazioni. I router di eventi sono ideali per instradare eventi da un gran numero di origini a un gran numero di destinazioni, con la possibilità di trasformare gli eventi prima della distribuzione a una destinazione.

  Nel seguente video viene fornita una panoramica di alto livello dei router di eventi:

- EventBridge Pipes è destinato alle integrazioni point-to-point; ogni pipe riceve eventi da un'unica origine per l'elaborazione e la distribuzione a un'unica destinazione. Le pipe includono anche il supporto per trasformazioni avanzate e l'arricchimento degli eventi prima della distribuzione a una destinazione.

Le pipe e i router di eventi vengono spesso utilizzati insieme. Un caso d'uso comune consiste nel creare una pipe con un router di eventi come destinazione; la pipe invia gli eventi al router di eventi, che quindi invia tali eventi a più destinazioni. Ad esempio, potresti creare una pipe con un flusso DynamoDB per un'origine e un router di eventi come destinazione. La pipe riceve eventi dal flusso DynamoDB e li invia al router di eventi, che quindi li invia a più destinazioni in base alle regole che hai specificato nel router di eventi.

EventBridge è l'evoluzione di Eventi Amazon CloudWatch

EventBridge era precedentemente chiamato Amazon CloudWatch Events. Il router di eventi predefinito e le regole create in CloudWatch Events sono visualizzati anche nella console EventBridge. EventBridge utilizza la stessa API di CloudWatch Events, quindi il codice utilizzato dall'API di CloudWatch Events rimane lo stesso.

EventBridge si basa sulle funzionalità di CloudWatch Events, tra cui eventi partner, registro di schemi ed EventBridge Pipes. Le nuove funzionalità aggiunte a EventBridge non sono state aggiunte a CloudWatch Events. Per ulteriori informazioni, consulta ???.

Tutte le funzionalità comuni presenti in CloudWatch Events sono presenti anche in EventBridge, tra cui:

- ???
- ???
- ???
- ???

Le funzionalità di EventBridge che si basano sulle funzionalità relative agli eventi e che le ampliano includono:

- ???
- ???
- ???
- ???

EventBridge Configurazione e prerequisiti di Amazon

Per utilizzare Amazon EventBridge, è necessario un AWS account. Il tuo account ti consente di utilizzare servizi come Amazon EC2 per generare eventi che puoi vedere nella EventBridge console. Puoi anche installare e configurare AWS Command Line Interface (AWS CLI) per utilizzare un'interfaccia a riga di comando per visualizzare gli eventi.

Argomenti

Registrati per un Account AWS

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

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
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, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegnate l'accesso amministrativo a un utente e utilizzate solo l'utente root per eseguire attività che richiedono l'accesso da parte dell'utente root.

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita 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 AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

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

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
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, consulta Configurare l'accesso utente con le impostazioni predefinite IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

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

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso ad altri 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, consulta Creare un set di autorizzazioni nella Guida per l'utente.AWS IAM Identity Center
2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

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

Accedi alla EventBridge console Amazon

Per accedere alla EventBridge console Amazon

- Accedi AWS Management Console e apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.

Credenziali dell'account

Sebbene sia possibile utilizzare le credenziali dell'utente root per accedere EventBridge, consigliamo di utilizzare invece un account AWS Identity and Access Management (IAM). Se utilizzi un account IAM per accedere EventBridge, devi disporre delle seguenti autorizzazioni.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "events:*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:events:*:*:*"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
```

```
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "events.amazonaws.com"
        }
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta Autenticazione.

Configura il AWS Command Line Interface

È possibile utilizzare il AWS CLI per eseguire EventBridge operazioni.

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

Endpoint regionali

È necessario abilitare gli endpoint regionali predefiniti da utilizzare EventBridge. Per ulteriori informazioni, consulta Attivazione e disattivazione AWS STS in una AWS regione nella Guida per l'utente IAM.

Guida introduttiva ad Amazon EventBridge

La base di EventBridge è creare regole che indirizzino gli eventi verso un obiettivo. In questa sezione, crei una regola di base. Per tutorial su scenari e destinazioni specifici, consulta Tutorial di Amazon EventBridge.

Crea una regola in Amazon EventBridge

Per creare una regola per gli eventi, specifichi un'azione da intraprendere quando EventBridge riceve un evento che corrisponde allo schema di eventi nella regola. Quando un evento corrisponde, EventBridge invia l'evento al target specificato e attiva l'azione definita nella regola.

Quando un AWS servizio del tuo AWS account emette un evento, passa sempre al bus eventi predefinito per il tuo account. Per scrivere una regola che abbini gli eventi AWS dei servizi del tuo account, devi associarla al bus eventi predefinito.

Per creare una regola per un AWS servizio

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola.

   Una regola non può avere lo stesso nome di un'altra regola nella stessa regione e sullo stesso router di eventi.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se la regola deve cercare eventi corrispondenti provenienti dal tuo account, seleziona Bus di eventi predefiniti di AWS . Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Seleziona Successivo.
8. Per Origine evento, scegli Servizi AWS .
9. (Facoltativo) In Eventi di esempio, scegli il tipo di evento.

10. In Modello di eventi, procedi come segue:
    - Per utilizzare un modello per creare un modello di eventi, scegli Modulo del modello di eventi e scegli Origine evento e Tipo di evento. Se scegli Tutti gli eventi come tipo di evento, tutti gli eventi emessi da questo AWS servizio corrisponderanno alla regola.

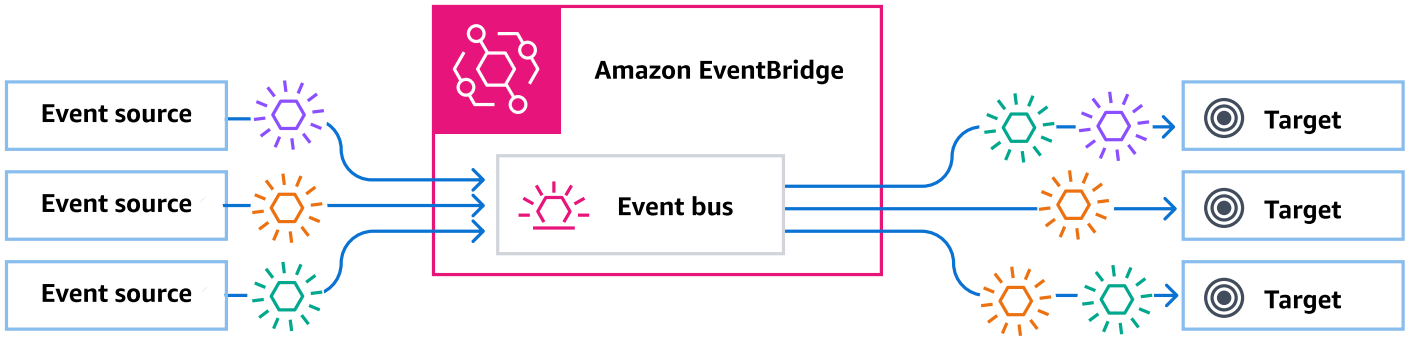
      Per personalizzare il modello, scegli Custom pattern (JSON editor) (Modello personalizzato [editor JSON]) e apporta le modifiche.
    - Per utilizzare un modello di eventi personalizzato, scegli Custom pattern (JSON editor) (Modello personalizzato [editor JSON]) e crea il tuo modello di evento.
11. Seleziona Successivo.
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. Per Seleziona un obiettivo, scegli il AWS servizio a cui desideri inviare le informazioni quando EventBridge rileva un evento che corrisponde allo schema dell'evento.
14. I campi visualizzati variano a seconda del servizio scelto. Se necessario, inserisci le informazioni specifiche per questo tipo di destinazione.
15. Per molti tipi di target, EventBridge sono necessarie le autorizzazioni per inviare eventi alla destinazione. In questi casi, EventBridge può creare il ruolo IAM necessario per l'esecuzione della regola. Esegui una di queste operazioni:
    - Per creare un ruolo IAM automaticamente, seleziona Crea un nuovo ruolo per questa risorsa specifica.
    - Per utilizzare un ruolo IAM creato in precedenza, seleziona Utilizza un ruolo esistente e seleziona il ruolo esistente dal menu a discesa.
16. (Facoltativo) Per Additional settings (Impostazioni aggiuntive), procedi come segue:
    a. Per Maximum age of event (Età massima dell'evento), immetti un valore compreso tra un minuto (00:01) e 24 ore (24:00).
    b. Per Tentativi, specifica un numero compreso tra 0 e 185.
    c. Per la coda di lettere non scritte, scegli se utilizzare una coda Amazon SQS standard come coda di lettere non scritte. EventBridge invia gli eventi che corrispondono a questa regola alla coda di lettere non scritte se non vengono consegnati correttamente alla destinazione. Esegui una di queste operazioni:
       - Scegli Nessuna per non utilizzare una coda DLQ.
       - Scegli Seleziona una coda Amazon SQS nell'account AWS corrente da utilizzare come coda DLQ, quindi seleziona la coda da utilizzare dal menu a discesa.

- Scegli Seleziona una coda Amazon SQS in un altro AWS account come coda di lettere non scritte, quindi inserisci l'ARN della coda da utilizzare. È necessario allegare una policy basata sulle risorse alla coda che conceda l'autorizzazione a inviarle messaggi. EventBridge Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.

17. (Facoltativo) Scegli Aggiungi destinazione per aggiungere un'altra destinazione per questa regola.
18. Seleziona Successivo.
19. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta la pagina EventBridge Etichette Amazon.
20. Seleziona Next (Successivo).
21. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Amazon EventBridge Event Bus

Un router di eventi è un router che riceve eventi e li invia a nessuna o a più destinazioni. I router di eventi sono ideali per instradare eventi da un gran numero di origini a un gran numero di destinazioni, con la possibilità di trasformare gli eventi prima della distribuzione a una destinazione.



Le regole associate al router di eventi valutano gli eventi man mano che arrivano. Ogni regola verifica se un evento corrisponde al modello della regola. Se l'evento corrisponde, EventBridge invia l'evento

Una regola viene associata a un router di eventi specifico, quindi la regola si applica solo agli eventi ricevuti da quel router di eventi.

> **Note**
>
> È inoltre possibile elaborare gli eventi utilizzando EventBridge Pipes. EventBridge Pipes è destinato point-to-point alle integrazioni; ogni pipe riceve eventi da un'unica fonte per l'elaborazione e la consegna a un'unica destinazione. Le pipe includono anche il supporto per trasformazioni avanzate e l'arricchimento degli eventi prima della distribuzione a una destinazione. Per ulteriori informazioni, consulta ???.

Argomenti

- Autorizzazioni per router di eventi di Amazon EventBridge
- Generazione di un modello di AWS CloudFormation da un router di eventi di Amazon EventBridge

Funzionamento dei router di eventi

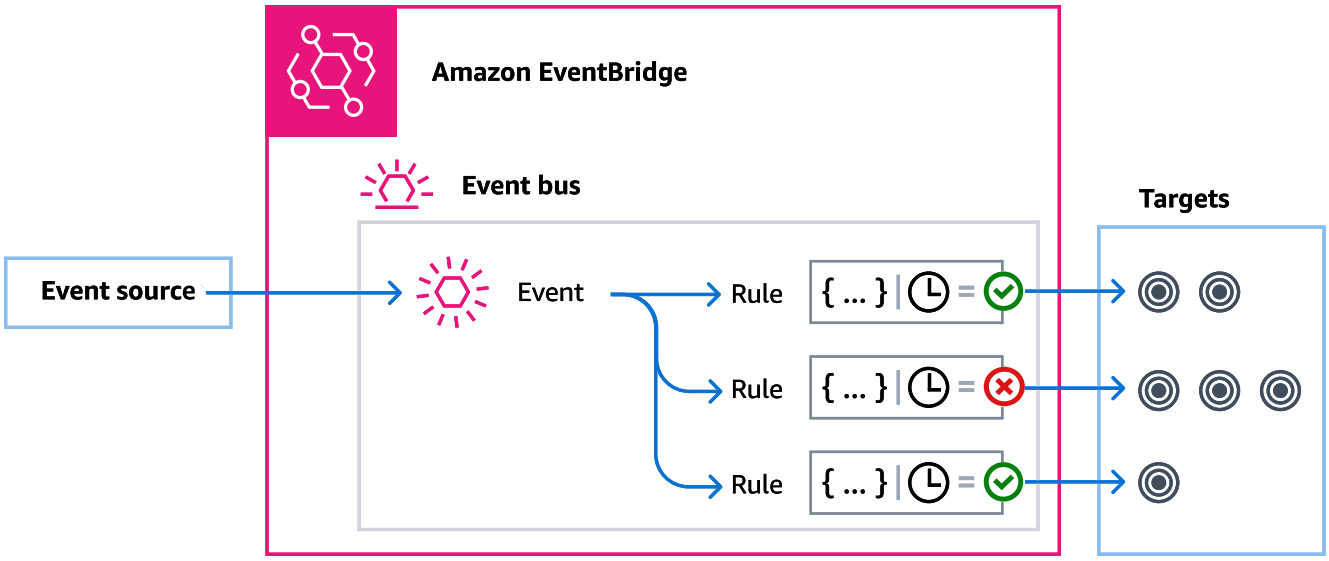
I router di eventi ti consentono di instradare gli eventi da più origini a molteplici destinazioni.

Di seguito è descritto come funziona un router di eventi:

1. Un'origine di eventi, che può essere un AWS servizio, un'applicazione personalizzata o un provider SaaS, invia un evento a un bus di eventi.
2. EventBridge quindi valuta l'evento in base a ciascuna regola definita per quel bus di eventi.

   Per ogni evento che corrisponde a una regola, EventBridge invia l'evento alle destinazioni specificate per quella regola. Facoltativamente, come parte della regola, puoi anche EventBridge specificare come trasformare l'evento prima di inviarlo alle destinazioni.

   Un evento può corrispondere a più regole e ogni regola può specificare fino a cinque destinazioni (Un evento potrebbe non corrispondere a nessuna regola, nel qual caso non EventBridge interviene.)



Consideriamo un esempio di utilizzo del bus di eventi EventBridge predefinito, che riceve automaticamente gli eventi dai AWS servizi:

1. Crei una regola nel router di eventi predefinito per l'evento `EC2 Instance State-change Notification`:
   - Specifichi che la regola corrisponde a eventi in cui lo `state` di un'istanza Amazon EC2 è cambiato in `running`.

     Lo fai specificando il codice JSON che definisce gli attributi e i valori a cui un evento deve corrispondere per attivare la regola. Ciò è denominato modello di eventi.

     ```
     {
       "source": ["aws.ec2"],
       "detail-type": ["EC2 Instance State-change Notification"],
         "detail": {
           "state": ["running"]
       }
     }
     ```
   - Specifichi che la destinazione della regola è una determinata funzione Lambda.
2. Ogni volta che lo stato di un'istanza Amazon EC2 cambia, Amazon EC2 (l'origine dell'evento) invia automaticamente quell'evento al router di eventi predefinito.
3. EventBridge valuta tutti gli eventi inviati al bus eventi predefinito rispetto alla regola che hai creato.

   Se l'evento corrisponde alla tua regola (ovvero se si tratta di un'istanza Amazon EC2 che cambia stato in`running`), EventBridge invia l'evento alla destinazione specificata. In questo caso, si tratta della funzione Lambda.

Il video seguente descrive cosa sono i router di eventi e a cosa servono: What are event buses

Il video seguente illustra i differenti router di eventi e quando utilizzarli: The differences between event buses

Concetti di Amazon EventBridge Event Bus

Di seguito viene fornita una descrizione più dettagliata dei componenti principali di un'architettura basata su router di eventi.

Bus di eventi

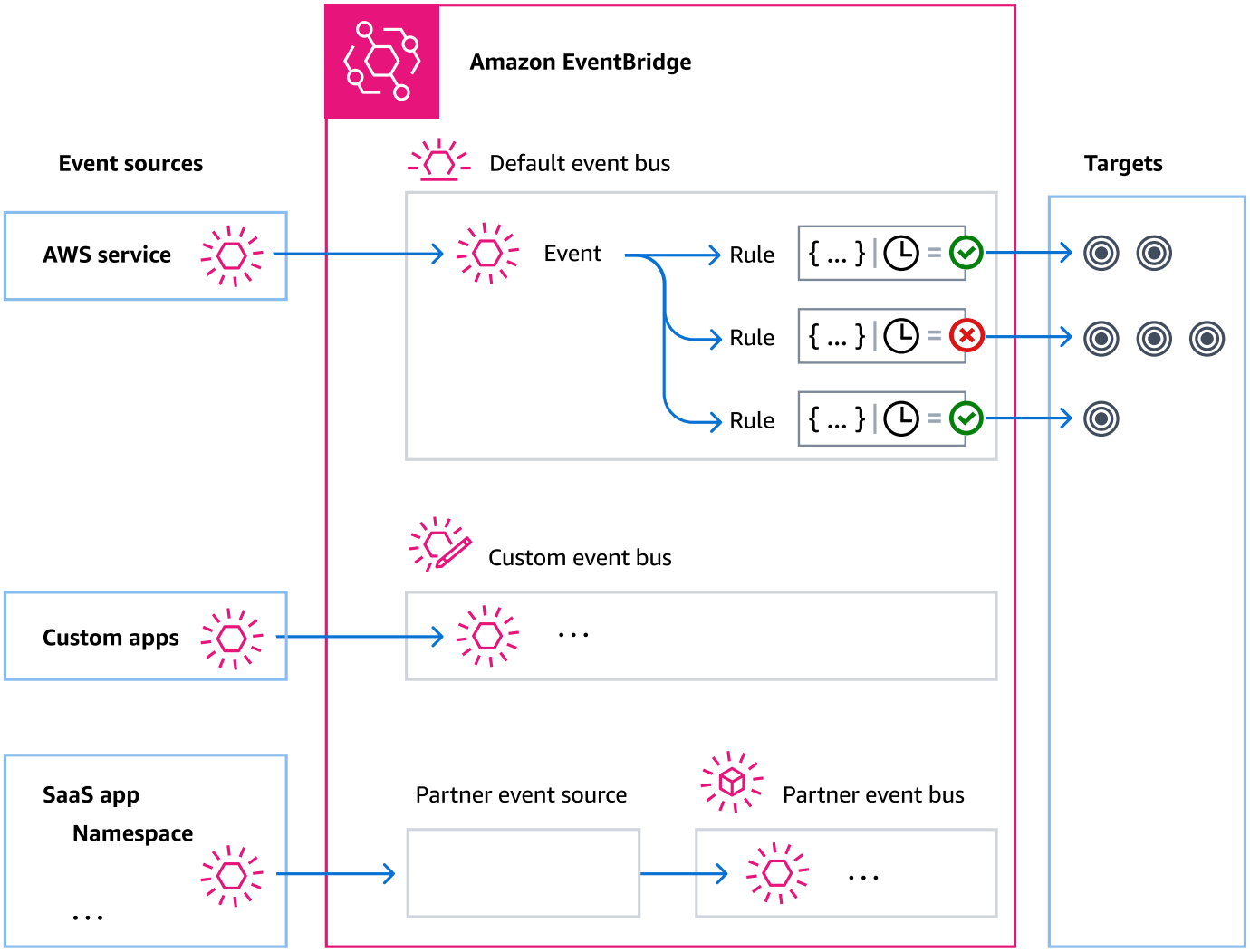
Un router di eventi è un router che riceve eventi e li invia a nessuna o a più destinazioni. Un router di eventi viene utilizzato per instradare eventi da un gran numero di origini a un gran numero di destinazioni, con la possibilità di trasformare gli eventi prima della distribuzione a una destinazione.

Il tuo account include un bus di eventi predefinito che riceve automaticamente gli eventi dai AWS servizi. Puoi anche:

- Creare router di eventi aggiuntivi, denominati route di eventi personalizzati, e specificare quali eventi devono ricevere.
- Creare router di eventi partner, che ricevono eventi da partner SaaS.

I casi d'uso più comuni per i router di eventi includono:

- Utilizzo di un router di eventi come agente tra diversi carichi di lavoro, servizi o sistemi.
- Utilizzo di più router di eventi nelle applicazioni per suddividere il traffico degli eventi. Ad esempio, creando un router per elaborare eventi contenenti informazioni di identificazione personale (PII) e un altro router di eventi che non esegue tali elaborazioni.
- L'aggregazione di eventi mediante l'invio di eventi da più router di eventi a un router di eventi centralizzato. Questo router centralizzato può essere nello stesso account degli altri router, ma anche in un account o in una Regione differente.



Eventi

Nella sua forma più semplice, un EventBridge evento è un oggetto JSON inviato a un bus o pipe di eventi.

Nel contesto dell'architettura basata su eventi (EDA), un evento rappresenta spesso un indicatore di un cambiamento in una risorsa o in un ambiente.

Per ulteriori informazioni, consulta ???.

Origini eventi

EventBridge può ricevere eventi da fonti di eventi, tra cui:

- AWS servizi
- applicazioni personalizzate;
- Partner Software as a Service (SaaS)

Regolamento

Una regola riceve gli eventi in entrata e li invia come appropriato alle destinazioni per l'elaborazione. Puoi specificare in che modo ogni regola richiama le proprie destinazioni in base a:

- Un modello di eventi, che contiene uno o più filtri per la corrispondenza con gli eventi. I modelli di eventi possono includere filtri per trovare corrispondenze con:
  - Metadati dell'evento: dati relativi all'evento, come l'origine dell'evento o l'account o la Regione in cui ha avuto origine l'evento.
  - Dati sull'evento: le proprietà dell'evento stesso. Queste proprietà variano in base all'evento.
  - Contenuto dell'evento: i valori effettivi delle proprietà dei dati dell'evento.
- Una pianificazione per richiamare le destinazioni a intervalli regolari.

  È possibile specificare una regola pianificata all'interno EventBridge o utilizzando EventBridge Scheduler.

  > **Note**
  >
  > EventBridge offre Amazon EventBridge Scheduler, uno strumento di pianificazione senza server che ti consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato. EventBridge Scheduler è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole EventBridge pianificate, con un set più ampio di operazioni e servizi API mirati. AWS
  > Ti consigliamo di utilizzare EventBridge Scheduler per richiamare gli obiettivi in base a una pianificazione. Per ulteriori informazioni, consulta ???.

Ogni regola è definita per uno specifico router di eventi e si applica solo agli eventi in quel router di eventi.

Una singola regola può inviare un evento a un massimo di cinque destinazioni.

Per impostazione predefinita è possibile configurare fino a 300 regole per router di eventi. Questa quota può essere aumentata fino a migliaia di regole nella console Service Quotas. Poiché il limite delle regole si applica a ciascun router, se hai bisogno di ancora più regole, puoi creare altri router di eventi personalizzati nel tuo account.

Puoi personalizzare il modo in cui gli eventi vengono ricevuti nel tuo account creando router di eventi con autorizzazioni diverse per servizi diversi.

Per personalizzare la struttura o la data di un evento prima di EventBridge passarlo a una destinazione, utilizzate il trasformatore di input per modificare le informazioni prima che arrivino alla destinazione.

Per ulteriori informazioni, consulta ???.

Destinazioni

Un target è una risorsa o un endpoint a cui EventBridge invia un evento quando l'evento corrisponde al modello di evento definito per una regola.

Una destinazione può ricevere più eventi da più router di eventi.

Per ulteriori informazioni, consulta ???.

Funzionalità avanzate per router di eventi

EventBridge include le seguenti funzionalità per aiutarvi a sviluppare, gestire e utilizzare i bus di eventi.

Utilizzo di destinazioni API per abilitare chiamate REST API tra servizi

EventBridge Le destinazioni API sono endpoint HTTP che è possibile impostare come destinazione di una regola, nello stesso modo in cui si inviano i dati degli eventi a un AWS servizio o a una risorsa. Con le destinazioni API, puoi utilizzare le chiamate API per instradare eventi tra servizi AWS , applicazioni SaaS integrate e le tue applicazioni esterne a AWS. Quando crei una destinazione API, specifichi una connessione per la destinazione. Ogni connessione include i dettagli sul tipo di autorizzazione e i parametri da utilizzare per l'autorizzazione con l'endpoint di destinazione API.

Archiviazione e riproduzione di eventi per favorire lo sviluppo e il ripristino di emergenza

È possibile archiviare o salvare gli eventi e riprodurli in un secondo momento dall'archivio. L'archiviazione è utile per:

- Testare un'applicazione perché si dispone di un archivio di eventi da utilizzare anziché dover attendere nuovi eventi.
- Idratare un nuovo servizio quando è online per la prima volta.
- Aggiungere maggiore durabilità alle applicazioni basate su eventi.

Utilizzo del registro di schemi per iniziare a creare rapidamente modelli di eventi

Quando si creano applicazioni serverless che utilizzano EventBridge, può essere utile conoscere la struttura degli eventi tipici senza dover generare l'evento. La struttura degli eventi è descritta in schemi, disponibili per tutti gli eventi generati dai AWS servizi di. EventBridge

Per gli eventi che non provengono dai AWS servizi, puoi:

- Creare o caricare schemi personalizzati.
- Usa Schema Discovery per creare EventBridge automaticamente schemi per gli eventi inviati al bus degli eventi.

Quando disponi di uno schema per un evento, puoi scaricare le associazioni di codice per i linguaggi di programmazione più diffusi.

Gestione delle risorse e dell'accesso con policy

Per organizzare AWS le risorse o tenere traccia dei costi EventBridge, puoi assegnare un'etichetta o un tag personalizzato alle AWS risorse. Utilizzando politiche basate su tag, puoi controllare ciò che le risorse possono e non possono fare all'interno. EventBridge

Oltre alle politiche basate su tag, EventBridge supporta politiche basate sull'identità e sulle risorse per controllare l'accesso. EventBridge Utilizza le policy basate su identità per controllare le autorizzazioni di un gruppo, ruolo o utente. Utilizza policy basate su risorse per assegnare autorizzazioni specifiche a ciascuna risorsa, ad esempio una funzione Lambda o un argomento di Amazon SNS.

Creazione di un bus di EventBridge eventi Amazon

Puoi creare un router di eventi personalizzato per ricevere eventi dalle tue applicazioni. Queste applicazioni possono anche inviare eventi al router di eventi predefinito. Quando crei un router di eventi, puoi associare una policy basata su risorse per concedere autorizzazioni ad altri account di modo che altri account possano inviare eventi al router di eventi nell'account corrente.

Il video seguente descrive come creare router di eventi: Creating an event bus

Per creare un router di eventi personalizzato

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Scegliere Create event bus (Crea bus di eventi).
4. Immettere un nome per il nuovo bus di eventi.
5. Scegli il KMS key formato EventBridge da utilizzare per crittografare i dati degli eventi memorizzati sul bus degli eventi.

> **Note**
>
> Gli archivi e l'individuazione degli schemi non sono supportati per i bus di eventi crittografati utilizzando un chiave gestita dal cliente. Per abilitare gli archivi o il rilevamento dello schema su un bus di eventi, scegli di utilizzare un Chiave di proprietà di AWS. Per ulteriori informazioni, consulta ???.

- Scegli Usa Chiave di proprietà di AWS per EventBridge crittografare i dati utilizzando un Chiave di proprietà di AWS.

  Si Chiave di proprietà di AWS tratta di un account KMS key che EventBridge possiede e gestisce per l'utilizzo in più AWS account. In generale, a meno che non sia necessario verificare o controllare la chiave di crittografia che protegge le risorse, an Chiave di proprietà di AWS è una buona scelta.

  Questa è l'impostazione predefinita.
- Scegliete Usa chiave gestita dal cliente EventBridge per cifrare i dati utilizzando chiave gestita dal cliente quello che avete specificato o creato.

  Chiavi gestite dal cliente sono KMS keys nel tuo AWS account che crei, possiedi e gestisci. Hai il pieno controllo su questi KMS keys.

  a. Specificane uno esistente chiave gestita dal cliente o scegli Crea un nuovo KMS key.

     EventBridge visualizza lo stato della chiave e tutti gli alias chiave che sono stati associati al valore specificato chiave gestita dal cliente.

b. Scegli la coda Amazon SQS da utilizzare come coda di lettere morte (DLQ) per questo bus di eventi, se disponibile.

   EventBridge invia gli eventi che non sono stati crittografati correttamente nel DLQ, se configurato, in modo da poterli elaborare in un secondo momento.

6. Configura le funzionalità opzionali del bus di eventi:

- Specificate una politica basata sulle risorse effettuando una delle seguenti operazioni:
  - Immetti la policy che include le autorizzazioni da concedere per il router di eventi. Puoi incollare una policy da un'altra origine o inserire il codice JSON per la policy. È possibile utilizzare una delle politiche di esempio e modificarla per il proprio ambiente.
  - Per utilizzare un modello per la policy, scegli Carica modello. Modifica la policy come necessario per il tuo ambiente, ad esempio aggiungendo ulteriori azioni che il principale nella policy sarà autorizzato a utilizzare.

  Per ulteriori informazioni sulla concessione delle autorizzazioni a un bus di eventi tramite politiche basate sulle risorse, consulta. ???
- Abilitare un archivio (opzionale)

  Puoi creare un archivio di eventi in modo da poterli riprodurre facilmente in un secondo momento. Ad esempio, è possibile che tu abbia la necessità di riprodurre gli eventi per correggere gli errori o per convalidare nuove funzionalità nell'applicazione. Per ulteriori informazioni, consulta ???

  a. In Archivi, scegli Abilitato.

  b. Specificate un nome e una descrizione per l'archivio.

  > **Note**
  >
  > Gli archivi e l'individuazione dello schema non sono supportati per i bus di eventi crittografati utilizzando un chiave gestita dal cliente. Per abilitare gli archivi o il rilevamento dello schema su un bus di eventi, scegli di utilizzare un Chiave di proprietà di AWS. Per ulteriori informazioni, consulta ???.
- Abilita l'individuazione dello schema (opzionale)

Abilita l'individuazione degli schemi per dedurre EventBridge automaticamente gli schemi direttamente dagli eventi in esecuzione su questo bus di eventi. Per ulteriori informazioni, consulta ???

a. In Scoperta dello schema, scegli Abilitato.

> **Note**
>
> Gli archivi e l'individuazione dello schema non sono supportati per i bus di eventi crittografati utilizzando un chiave gestita dal cliente. Per abilitare gli archivi o il rilevamento dello schema su un bus di eventi, scegli di utilizzare un Chiave di proprietà di AWS. Per ulteriori informazioni, consulta ???.

- Specificare i tag (opzionale)

  Un tag è un'etichetta di attributo personalizzata che si assegna a una AWS risorsa. Usa i tag per identificare e organizzare AWS le tue risorse. Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Per ulteriori informazioni, consulta ???

  a. In Tags (Tag), seleziona Add new tag (Aggiungi nuovo tag).

  b. Specificate una chiave e, facoltativamente, un valore per il nuovo tag.

7. Scegli Create (Crea).

Aggiornamento di un bus di EventBridge eventi Amazon

È possibile aggiornare la configurazione dei bus di eventi dopo averli creati. Ciò include il bus di eventi predefinito, che EventBridge viene creato automaticamente nel tuo account.

Aggiornamento del codice KMS key utilizzato per la crittografia

> **Note**
>
> Gli archivi e il rilevamento dello schema non sono supportati per i bus di eventi crittografati utilizzando un chiave gestita dal cliente. Per abilitare gli archivi o il rilevamento dello schema su un bus di eventi, scegli di utilizzare un Chiave di proprietà di AWS. Per ulteriori informazioni, consulta ???.

Per modificare l'impostazione KMS key utilizzata per la crittografia inattiva su un bus di eventi utilizzando la EventBridge console

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Scegli l'event bus che desideri aggiornare.
4. Nella pagina dei dettagli del bus degli eventi, scegli la scheda Crittografia.
5. Scegliete la EventBridge modalità da utilizzare KMS key per crittografare i dati degli eventi memorizzati sul bus degli eventi:
   - Scegli Usa Chiave di proprietà di AWS per EventBridge crittografare i dati utilizzando un. Chiave di proprietà di AWS

     Si Chiave di proprietà di AWS tratta di un account KMS key che EventBridge possiede e gestisce per l'utilizzo in più AWS account. In generale, a meno che non sia necessario verificare o controllare la chiave di crittografia che protegge le risorse, an Chiave di proprietà di AWS è una buona scelta.

     Questa è l'impostazione predefinita.
   - Scegliete Usa chiave gestita dal cliente EventBridge per crittografare i dati utilizzando chiave gestita dal cliente quello che avete specificato o creato.

     Chiavi gestite dal cliente sono KMS keys nel tuo AWS account che crei, possiedi e gestisci. Hai il pieno controllo su questi KMS keys.

     a. Specificane uno esistente chiave gestita dal cliente o scegli Crea un nuovo KMS key.

        EventBridge visualizza lo stato della chiave e tutti gli alias chiave che sono stati associati al valore specificato chiave gestita dal cliente.

     b. Scegli la coda Amazon SQS da utilizzare come coda di lettere morte (DLQ) per questo bus di eventi, se disponibile.

        EventBridge invia gli eventi che non sono stati crittografati correttamente nel DLQ, se configurato, in modo da poterli elaborare in un secondo momento.

Aggiornamento delle autorizzazioni su un bus di eventi

Puoi concedere autorizzazioni aggiuntive a un router di eventi allegandovi una policy basata su risorse. Per istruzioni dettagliate sull'aggiornamento delle autorizzazioni fornite a un bus di eventi, vedere Gestione delle autorizzazioni del bus di eventi.

Aggiungere o rimuovere archivi sui bus degli eventi

Un archivio consente di acquisire gli eventi in modo da poterli riprodurre facilmente in un secondo momento. Ad esempio, è possibile che tu abbia la necessità di riprodurre gli eventi per correggere gli errori o per convalidare nuove funzionalità nell'applicazione. Per ulteriori informazioni, consulta EventBridge archivia e riproduci.

> **Note**
>
> Gli archivi e l'individuazione dello schema non sono supportati per i bus di eventi crittografati utilizzando un chiave gestita dal cliente. Per abilitare gli archivi o il rilevamento dello schema su un bus di eventi, scegli di utilizzare un Chiave di proprietà di AWS. Per ulteriori informazioni, consulta ???.

Per aggiungere o rimuovere un archivio da un bus di eventi utilizzando la EventBridge console

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Scegli l'event bus che desideri aggiornare.
4. Nella pagina dei dettagli del bus degli eventi, scegli la scheda Archivi.
5. Esegui una di queste operazioni:
   - Per aggiungere un archivio:
     a. Scegli Crea archivio.
     b. Specificare gli attributi per l'archivio.
     c. Seleziona Successivo.
     d. Scegliete lo schema di eventi da applicare agli eventi per l'archivio.
     e. Scegli Crea archivio.
   - Per eliminare un archivio:

a. Per il tag che desideri rimuovere, scegli Elimina.
b. Inserisci il nome dell'archivio e scegli Elimina.

L'archivio viene eliminato definitivamente. Questa operazione non può essere annullata.

Per creare o eliminare un archivio per un bus di eventi utilizzando AWS CLI

- Per creare un archivio, usa create-archive.

  Per eliminare definitivamente un archivio, usa delete-archive.

Avvio o arresto del rilevamento dello schema sui bus degli eventi

Per ulteriori informazioni sull'individuazione degli schemi, vedere EventBridge schemi.

> **Note**
>
> Gli archivi e l'individuazione degli schemi non sono supportati per i bus di eventi crittografati utilizzando un chiave gestita dal cliente. Per abilitare gli archivi o il rilevamento dello schema su un bus di eventi, scegli di utilizzare un Chiave di proprietà di AWS. Per ulteriori informazioni, consulta ???.

Per avviare o interrompere l'individuazione dello schema su un bus di eventi utilizzando la EventBridge console

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Scegli l'event bus che desideri aggiornare.
4. Esegui una di queste operazioni:
   - Per avviare l'individuazione dello schema, scegli Avvia scoperta.
   - Per interrompere l'individuazione dello schema, scegli Elimina scoperta.

Per avviare o interrompere l'individuazione dello schema su un bus di eventi utilizzando il AWS CLI

- Per avviare l'individuazione dello schema, usa create-discoverer.

Per interrompere l'individuazione dello schema, usa delete-discoverer.

Aggiungere o rimuovere tag sui bus degli eventi

Un tag è un'etichetta di attributo personalizzata che l'utente o AWS assegna a una AWS risorsa. Usa i tag per identificare e organizzare AWS le tue risorse. Per ulteriori informazioni, consulta i EventBridge tag.

Per aggiungere o rimuovere tag da un bus di eventi utilizzando la EventBridge console

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Scegli l'event bus che desideri aggiornare.
4. Nella pagina dei dettagli del bus degli eventi, scegli la scheda Tag, quindi scegli Gestisci tag.
5. Esegui una di queste operazioni:
   - Per aggiungere un tag:
     a. Scegli Aggiungi nuovo tag.
     b. Specificate la chiave e il valore per il tag
     c. Scegli Aggiorna.
   - Per rimuovere un tag:
     a. Per il tag che desideri rimuovere, scegli Rimuovi.
     b. Scegli Aggiorna.

Per aggiungere o rimuovere tag da un bus di eventi utilizzando il AWS CLI

- Per aggiungere tag, usa tag-resource.

  Per rimuovere i tag, usa untag-resource.

Aggiornamento del bus degli eventi predefinito utilizzando AWS CloudFormation

AWS CloudFormation consente di configurare e gestire AWS le risorse tra account e regioni in modo centralizzato e ripetibile trattando l'infrastruttura come codice. CloudFormation lo fa consentendoti di creare modelli che definiscono le risorse che desideri fornire e gestire.

Poiché EventBridge inserisce automaticamente il bus degli eventi predefinito nel tuo account, non puoi crearlo utilizzando un CloudFormation modello, come faresti normalmente per qualsiasi risorsa che desideri includere in uno CloudFormation stack. Per includere il bus degli eventi predefinito in uno CloudFormation stack, devi prima importarlo in uno stack. Dopo aver importato il bus degli eventi predefinito in uno stack, potete aggiornare le proprietà del bus degli eventi come desiderate.

Per importare una risorsa esistente in uno CloudFormation stack nuovo o esistente, sono necessarie le seguenti informazioni:

- Un identificatore univoco per la risorsa da importare.

  Per i bus di eventi predefiniti, l'identificatore è `Name` e quindi il valore dell'identificatore è. `default`
- Un modello che descrive accuratamente le proprietà correnti della risorsa esistente.

  Il frammento di modello riportato di seguito contiene una `AWS::Events::EventBus` risorsa che descrive le proprietà correnti di un bus di eventi predefinito. In questo esempio, il bus degli eventi è stato configurato per utilizzare a chiave gestita dal cliente e DLQ per la crittografia a riposo.

  Inoltre, la `AWS::Events::EventBus` risorsa che descrive il bus di eventi predefinito da importare dovrebbe includere una `DeletionPolicy` proprietà impostata su. `Retain`

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Default event bus import example",
    "Resources": {
        "defaultEventBus": {
            "Type" : "AWS::Events::EventBus",
            "DeletionPolicy": "Retain",
            "Properties" : {
                "Name" : "default",
                "KmsKeyIdentifier" : "KmsKeyArn",
                "DeadLetterConfig" : {
                    "Arn" : "DLQ_ARN"
```

```
                }
            }
        }
    }
}
```

Per ulteriori informazioni, consulta la sezione CloudFormation Gestione delle risorse esistenti nella Guida CloudFormation per l'utente.

Eliminazione di un bus di EventBridge eventi Amazon

È possibile eliminare un bus di eventi personalizzato o partner. Non è possibile eliminare il bus di eventi predefinito. L'eliminazione di un bus di eventi elimina le regole associate a quel bus di eventi.

Per eliminare un bus di eventi utilizzando la console EventBridge

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Scegli il bus dell'evento che desideri eliminare.
4. Esegui una di queste operazioni:
   - Scegli Elimina.
   - Scegli il nome del bus dell'evento.

     Nella pagina dei dettagli del bus dell'evento, scegli Elimina.

Autorizzazioni per router di eventi di Amazon EventBridge

Il router di eventi predefinito del tuo account AWS consente solo eventi provenienti da un account. Puoi concedere autorizzazioni aggiuntive a un router di eventi allegandovi una policy basata su risorse. Con una policy basata su risorse, puoi consentire chiamate API `PutEvents`, `PutRule` e `PutTargets` da un altro account. Puoi anche utilizzare le condizioni IAM nella policy per concedere autorizzazioni a un'organizzazione, applicare tag o filtrare gli eventi relativi solo a una regola o un account specifico. Puoi impostare una policy basata su risorse per un router di eventi al momento della creazione o successivamente.

Le API di EventBridge che accettano un parametro `Name` del router di eventi come `PutRule`, `PutTargets`, `DeleteRule`, `RemoveTargets`, `DisableRule` e `EnableRule` accettano anche

l'ARN del router di eventi. Utilizza questi parametri per fare riferimento a router di eventi multi-account o multiregionali tramite le API. Ad esempio, puoi chiamare `PutRule` per creare una regola in un router di eventi in un account diverso senza dover assumere un ruolo.

Puoi allegare le policy di esempio in questo argomento a un ruolo IAM per concedere l'autorizzazione a inviare eventi a un account o a una Regione differenti. Utilizza i ruoli IAM per impostare le policy di controllo dell'organizzazione e i limiti su chi può inviare eventi dal tuo account ad altri account. Consigliamo di utilizzare sempre i ruoli IAM quando la destinazione di una regola è un router di eventi. Puoi associare i ruoli IAM utilizzando chiamate `PutTarget`. Per informazioni sulla creazione di una regola per inviare eventi a un account o a una Regione differente, consulta Invio e ricezione di EventBridge eventi Amazon tra AWS account.

Argomenti

Gestione delle autorizzazioni di un router di eventi

Per modificare le autorizzazioni di un router di eventi esistente, utilizza la procedura seguente. Per informazioni su come utilizzare AWS CloudFormation per creare una policy del router di eventi, consulta AWS::Events::EventBusPolicy.

Per gestire le autorizzazioni per un router di eventi esistente

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Router di eventi.
3. In Nome, scegli il nome del router di eventi per cui gestire le autorizzazioni.

   Se una policy basata su risorse è associata al router di eventi, viene visualizzata.
4. Scegli Gestisci le autorizzazioni, quindi esegui una delle seguenti operazioni:

- Immetti la policy che include le autorizzazioni da concedere per il router di eventi. Puoi incollare una policy da un'altra origine o immettere il codice JSON per la policy.
- Per utilizzare un modello per la policy, scegli Carica modello. Modifica la policy come necessario per il tuo ambiente e aggiungi altre azioni che il principale nella policy è autorizzato a utilizzare.

5. Scegli Aggiorna.

Il modello fornisce esempi di istruzioni di policy che puoi personalizzare per il tuo account e il tuo ambiente. Il modello non è una policy valida. Puoi modificare il modello in base al tuo caso d'uso oppure copiare una delle policy di esempio e personalizzarla.

Il modello carica policy che includono un esempio di come concedere le autorizzazioni a un account per utilizzare l'azione `PutEvents`, come concedere autorizzazioni a un'organizzazione e come concedere autorizzazioni all'account per gestire le regole nell'account. Puoi personalizzare il modello per il tuo account specifico e quindi eliminare le altre sezioni dal modello. Altri esempi di policy sono descritti più avanti in questo argomento.

Se tenti di aggiornare le autorizzazioni per il router ma la policy contiene un errore, un messaggio di errore indica il problema specifico nella policy.

```
 ### Choose which sections to include in the policy to match your use case. ###
 ### Be sure to remove all lines that start with ###, including the ### at the end of
the line. ###

 ### The policy must include the following: ###

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

     ### To grant permissions for an account to use the PutEvents action, include the
following, otherwise delete this section: ###

     {

       "Sid": "AllowAccountToPutEvents",
       "Effect": "Allow",
       "Principal": {
         "AWS": "<ACCOUNT_ID>"
```

```
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default"
    },

    ### Include the following section to grant permissions to all members of your AWS
Organizations to use the PutEvents action ###

    {
      "Sid": "AllowAllAccountsFromOrganizationToPutEvents",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "o-yourOrgID"
        }
      }
    },

    ### Include the following section to grant permissions to the account to manage
the rules created in the account ###

    {
      "Sid": "AllowAccountToManageRulesTheyCreated",
      "Effect": "Allow",
      "Principal": {
        "AWS": "<ACCOUNT_ID>"
      },
      "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events:DeleteRule",
        "events:RemoveTargets",
        "events:DisableRule",
        "events:EnableRule",
        "events:TagResource",
        "events:UntagResource",
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "events:ListTagsForResource"],
      "Resource": "arn:aws:events:us-east-1:123456789012:rule/default",
      "Condition": {
```

```
          "StringEqualsIfExists": {
            "events:creatorAccount": "<ACCOUNT_ID>"
          }
        }
    }]
  }
```

Policy di esempio: invio di eventi al router predefinito in un account diverso

La seguente policy di esempio concede all'account 111122223333 l'autorizzazione per pubblicare eventi sul router di eventi predefinito nell'account 123456789012.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
        "Sid": "sid1",
        "Effect": "Allow",
        "Principal": {"AWS":"arn:aws:iam::111112222333:root"},
        "Action": "events:PutEvents",
        "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default"
        }
    ]
  }
```

Policy di esempio: invio di eventi a un router personalizzato in un account diverso

La seguente policy di esempio concede all'account 111122223333 l'autorizzazione per pubblicare eventi su `central-event-bus` nell'account in 123456789012, ma solo per gli eventi con un valore di origine impostato su `com.exampleCorp.webStore` e un `detail-type` impostato su `newOrderCreated`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WebStoreCrossAccountPublish",
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
```

```
      ],
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/central-event-bus",
      "Condition": {
        "StringEquals": {
          "events:detail-type": "newOrderCreated",
          "events:source": "com.exampleCorp.webStore"
        }
      }
    }
  ]
}
```

Policy di esempio: invio di eventi a un router di eventi nello stesso account

La seguente policy di esempio associata a un router di eventi denominato `CustomBus1` consente al router di eventi di ricevere eventi dallo stesso account e dalla stessa Regione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:123456789:event-bus/CustomBus1"
      ]
    }
  ]
}
```

Policy di esempio: invio di eventi allo stesso account e limitazione degli aggiornamenti

La seguente policy di esempio concede all'account 123456789012 l'autorizzazione per creare, eliminare, aggiornare, disabilitare e abilitare regole e aggiungere o rimuovere destinazioni. Limita queste regole che corrispondono agli eventi con un'origine di `com.exampleCorp.webStore` e

utilizza `"events:creatorAccount": "${aws:PrincipalAccount}"` per garantire che solo l'account 123456789012 possa modificare tali regole e destinazioni una volta creati.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "InvoiceProcessingRuleCreation",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": [
        "events:PutRule",
        "events:DeleteRule",
        "events:DescribeRule",
        "events:DisableRule",
        "events:EnableRule",
        "events:PutTargets",
        "events:RemoveTargets"
      ],
      "Resource": "arn:aws:events:us-east-1:123456789012:rule/central-event-bus/*",
      "Condition": {
        "StringEqualsIfExists": {
          "events:creatorAccount": "${aws:PrincipalAccount}",
          "events:source": "com.exampleCorp.webStore"
        }
      }
    }
  ]
}
```

Policy di esempio: invia eventi solo da una regola specifica al router in una Regione diversa

La policy di esempio seguente concede all'account 111122223333 l'autorizzazione per inviare eventi che corrispondono a una regola denominata `SendToUSE1AnotherAccount` nelle Regioni Medio Oriente (Bahrein) e Stati Uniti occidentali (Oregon) a un router di eventi denominato `CrossRegionBus` nella Regione Stati Uniti orientali (Virginia settentrionale) nell'account 123456789012. La policy di esempio viene aggiunta al router di eventi denominato `CrossRegionBus` nell'account 123456789012. La policy consente gli eventi solo se corrispondono

a una regola specificata per il router di eventi nell'account 111122223333. L'istruzione `Condition` limita gli eventi ai soli eventi che corrispondono alle regole con l'ARN della regola specificata.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSpecificRulesAsCrossRegionSource",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:events:us-west-2:111112222333:rule/CrossRegionBus/
SendToUSE1AnotherAccount",
            "arn:aws:events:me-south-1:111112222333:rule/CrossRegionBus/
SendToUSE1AnotherAccount"
          ]
        }
      }
    }
  ]
}
```

Policy di esempio: invia eventi solo da una Regione specifica a un'altra Regione

La policy di esempio seguente concede all'account 111122223333 l'autorizzazione per inviare tutti gli eventi generati nelle Regioni Medio Oriente (Bahrein) e Stati Uniti occidentali (Oregon) a un router di eventi denominato `CrossRegionBus` nell'account 123456789012 nella Regione Stati Uniti orientali (Virginia settentrionale). L'account 111122223333 non è autorizzato a inviare eventi generati in qualsiasi altra Regione.

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

```
      "Sid": "AllowCrossRegionEventsFromUSWest2AndMESouth1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:events:us-west-2:*:*",
            "arn:aws:events:me-south-1:*:*"
          ]
        }
      }
    }
  ]
}
```

Policy di esempio: nega l'invio di eventi da specifiche Regioni

La seguente policy di esempio associata a un router di eventi denominato `CrossRegionBus` nell'account 123456789012 autorizza il router di eventi a ricevere eventi dall'account 111122223333, ma non eventi generati nella Regione Stati Uniti occidentali (Oregon).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1AllowAnyEventsFromAccount111112222333",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus"
    },
    {
      "Sid": "2DenyAllCrossRegionUSWest2Events",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
```

```
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:events:us-west-2:*:*"
          ]
        }
      }
    }
  ]
}
```

Generazione di un modello di AWS CloudFormation da un router di eventi di Amazon EventBridge

AWS CloudFormation consente di configurare e gestire le risorse AWS tra account e regioni in modo centralizzato e ripetibile trattando l'infrastruttura come codice. CloudFormation consente di creare modelli che definiscono le risorse da fornire e gestire.

EventBridge ti consente di generare modelli dai router di eventi esistenti nel tuo account, per aiutarti a iniziare subito a sviluppare modelli di CloudFormation. Inoltre, EventBridge ti offre la possibilità di includere le regole associate a quel router di eventi nel tuo modello. Puoi quindi utilizzare quei modelli come base per creare stack di risorse mediante la gestione con CloudFormation.

Per ulteriori informazioni su CloudFormation, consulta la Guida per l'utente di AWS CloudFormation.

> **Note**
>
> EventBridge non include regole gestite nel modello generato.

Poi anche generare un modello da una o più regole contenute in un router di eventi selezionato.

Per generare un modello di CloudFormation da un router di eventi

1. Apri la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Scegli il router di eventi dal quale generare un modello di CloudFormation.

4. Dal menu Azioni, scegli Modello CloudFormation, quindi scegli il formato in cui EventBridge deve generare il modello, ovvero JSON o YAML.

   EventBridge visualizza il modello, generato nel formato selezionato. Per impostazione predefinita, tutte le regole associate al router di eventi sono incluse nel modello.

   - Per generare il modello senza includere regole, deseleziona Includi regole su questo EventBus.
5. EventBridge ti offre la possibilità di scaricare il file di modello o di copiare il modello negli appunti.

   - Per scaricare il file di modello, scegli Scarica.
   - Per copiare il modello negli appunti, scegli Copia.
6. Per uscire dal modello, scegli Annulla.

Dopo aver personalizzato il modello di AWS CloudFormation come necessario per il tuo caso d'uso, puoi utilizzarlo per creare stack in CloudFormation.

Considerazioni sull'utilizzo di modelli CloudFormation generati da Amazon EventBridge

Prendi in considerazione i seguenti fattori quando utilizzi un modello CloudFormation generato da un router di eventi:

- EventBridge non include alcuna password nel modello generato.

  Puoi modificare il modello per includere parametri del modello che consentono agli utenti di specificare password o altre informazioni riservate quando utilizzano il modello per creare o aggiornare uno stack CloudFormation.

  Inoltre, gli utenti possono utilizzare Secrets Manager per creare un segreto nella Regione desiderata e quindi modificare il modello generato per utilizzare parametri dinamici.
- Le destinazioni nel modello generato rimangono esattamente come specificate nel router di eventi originale. Se il modello non viene modificato in modo appropriato prima di utilizzarlo per creare stack in altre Regioni, è possibile che si abbiano problemi in più Regioni.

  Inoltre, il modello generato non creerà automaticamente destinazioni a valle.

EventBridge Eventi Amazon

Un evento indica una modifica in un ambiente come un ambiente AWS , un'applicazione o un servizio partner SaaS oppure uno dei tuoi servizi o applicazioni. Di seguito sono riportati alcuni esempi di eventi:

- Amazon EC2 genera un evento quando lo stato di un'istanza cambia da In attesa a In esecuzione.
- Dimensionamento automatico Amazon EC2 genera eventi quando avvia o termina le istanze.
- AWS CloudTrail pubblica eventi quando effettui chiamate API.

Puoi anche impostare gli eventi pianificati generati periodicamente.

Per un elenco dei servizi che generano eventi, inclusi eventi di esempio di ogni servizio, consulta Eventi derivanti dai servizi AWS e segui i collegamenti nella tabella.

Gli eventi sono rappresentati come oggetti JSON e hanno tutti una struttura simile e gli stessi campi di primo livello.

I contenuti del campo di primo livello detail (dettaglio) sono diversi in base a quale servizio ha generato l'evento e all'evento stesso. La combinazione dei campi source (origine) e detail-type (tipo di dettaglio) serve a identificare i campi e i valori individuati nel campo detail (dettaglio). Per esempi di eventi generati dai AWS servizi, vediEventi derivanti dai servizi AWS.

Argomenti

- Riferimento per la struttura degli eventi
- Aggiungere EventBridge eventi Amazon con PutEvents
- Eventi derivanti dai servizi AWS
- Ricezione di eventi da un partner SaaS con Amazon EventBridge
- Debug della distribuzione di eventi

Il video seguente fornisce informazioni di base sugli eventi: What is an event

Il video seguente illustra come arrivano gli eventi EventBridge: Da dove provengono gli eventi

Riferimento per la struttura degli eventi

I seguenti campi vengono visualizzati in tutti gli eventi inviati a un bus di eventi e comprendono i metadati dell'evento:

```
{
  "???": "0",
  "???": "UUID",
  "???": "event name",
  "???": "event source",
  "???": "ARN",
  "???": "timestamp",
  "???": "region",
  "???": [
    "ARN"
  ],
  "???": {
    JSON object
  }
}
```

version

Per impostazione predefinita, questo valore è impostato su 0 (zero) in tutti gli eventi.

id

Un UUID versione 4 generato per ogni evento. Puoi utilizzare `id` per tracciare eventi mentre si spostano attraverso le regole verso le destinazioni.

detail-type (tipo di dettaglio)

Identifica, in combinazione con il campo source (origine), i campi e i valori visualizzati nel campo detail (dettaglio).

Gli eventi che vengono consegnati da CloudTrail hanno `AWS API Call via CloudTrail` come valore per. `detail-type`

source

Identifica il servizio che ha generato l'evento. Tutti gli eventi che provengono dai servizi AWS iniziano con "aws". Gli eventi generati dal cliente possono qui presentare qualsiasi valore, purché non inizi con "aws". Consigliamo l'uso di stringhe di nomi di domino inverse che utilizzano lo stile di nomi dei pacchetti di Java.

Per trovare il valore corretto `source` per un AWS servizio, consulta La tabella delle chiavi di condizione, seleziona un servizio dall'elenco e cerca il prefisso del servizio. Ad esempio, il `source` valore per Amazon CloudFront è`aws.cloudfront`.

account

Il numero di 12 cifre che identifica un AWS account.

time

Il timestamp dell'evento, che può essere specificato dal servizio che origina l'evento. Se l'evento si estende per un intervallo di tempo, il servizio potrebbe segnalare l'orario di inizio, pertanto questo valore potrebbe essere antecedente all'orario di ricezione dell'evento.

Regione

Identifica la AWS regione da cui ha avuto origine l'evento.

risorse

Un array JSON che contiene gli ARN che identificano le risorse coinvolte nell'evento. Il servizio che genera l'evento determina se includere questi ARN. Ad esempio, le modifiche dello stato delle istanze Amazon EC2 includono gli ARN delle istanze Amazon EC2, gli eventi Auto Scaling includono gli ARN delle istanze e dei gruppi Auto Scaling, ma la chiamate API con AWS CloudTrail non includono gli ARN di risorsa.

detail (dettaglio)

Un oggetto JSON contenente informazioni sull'evento. Il servizio che genera l'evento determina il contenuto di questo campo. Può essere `"{}"`.

AWS Gli eventi di chiamata API hanno oggetti di dettaglio con circa 50 campi annidati a diversi livelli di profondità.

> **Note**
>
> PutEventsaccetta dati in formato JSON. Per il tipo di dati numero JSON (intero), i vincoli sono: un valore minimo di -9.223.372.036.854.854.775.808 e un valore massimo di 9.223.372.036.854.854.854.775.807.

Example Esempio: notifica sulla modifica dello stato dell'istanza di Amazon EC2

Il seguente evento in Amazon EventBridge indica la chiusura di un'istanza Amazon EC2.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/i-1234567890abcdef0"
  ],
  "detail": {
    "instance-id": " i-1234567890abcdef0",
    "state": "terminated"
  }
}
```

Informazioni minime necessarie per un evento personalizzato valido

Quando crei eventi personalizzati, questi devono includere i seguenti campi:

- `detail`
- `detail-type`
- `source`

```
{
  "detail-type": "event name",
  "source": "event source",
  "detail": {
  }
}
```

Aggiungere EventBridge eventi Amazon con **`PutEvents`**

L'`PutEvents`azione invia più eventi EventBridge in un'unica richiesta. Per ulteriori informazioni, consulta PutEventsAmazon EventBridge API Reference e put-events in AWS CLI Command Reference.

Ogni richiesta `PutEvents` può supportare un numero limitato di voci. Per ulteriori informazioni, consulta Quote di Amazon EventBridge. L'operazione `PutEvents` tenta di elaborare tutte le voci secondo l'ordine naturale della richiesta. Dopo la chiamata`PutEvents`, EventBridge assegna a ogni evento un ID univoco.

Argomenti

- Gestione degli errori con PutEvents
- Invio di eventi tramite AWS CLI
- Calcolo delle dimensioni di iscrizione agli EventBridge PutEvents eventi Amazon

Il codice Java di esempio seguente invia due eventi identici a EventBridge.

AWS SDK for Java Version 2.x

```
EventBridgeClient eventBridgeClient =
    EventBridgeClient.builder().build();

PutEventsRequestEntry requestEntry = PutEventsRequestEntry.builder()
    .resources("resource1", "resource2")
    .source("com.mycompany.myapp")
    .detailType("myDetailType")
    .detail("{ \"key1\": \"value1\", \"key2\": \"value2\" }")
    .build();

List <
PutEventsRequestEntry > requestEntries = new ArrayList <
PutEventsRequestEntry > ();
requestEntries.add(requestEntry);

PutEventsRequest eventsRequest = PutEventsRequest.builder()
    .entries(requestEntries)
    .build();

PutEventsResponse result = eventBridgeClient.putEvents(eventsRequest);

for (PutEventsResultEntry resultEntry: result.entries()) {
    if (resultEntry.eventId() != null) {
        System.out.println("Event Id: " + resultEntry.eventId());
    } else {
        System.out.println("PutEvents failed with Error Code: " +
 resultEntry.errorCode());
```

```
    }
}
```

AWS SDK for Java Version 1.0

```
EventBridgeClient eventBridgeClient =
    EventBridgeClient.builder().build();

PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");

PutEventsRequest request = new PutEventsRequest()
        .withEntries(requestEntry, requestEntry);

PutEventsResult result = awsEventsClient.putEvents(request);

for (PutEventsResultEntry resultEntry : result.getEntries()) {
    if (resultEntry.getEventId() != null) {
        System.out.println("Event Id: " + resultEntry.getEventId());
    } else {
        System.out.println("Injection failed with Error Code: " +
 resultEntry.getErrorCode());
    }
}
```

Dopo aver eseguito questo codice, il risultato `PutEvents` include un array di voci di risposta. Ogni voce nell'array di risposte corrisponde a una voce nella matrice di richieste secondo l'ordine dall'inizio alla fine della richiesta e della risposta. La matrice di risposta `Entries` include sempre lo stesso numero di voci della matrice di richiesta.

Gestione degli errori con **`PutEvents`**

Per impostazione predefinita, se una singola immissione all'interno di una richiesta ha esito negativo, EventBridge continua a elaborare le altre voci della richiesta. Un array `Entries` di risposte può includere sia le voci riuscite che quelle non riuscite. È necessario rilevare le voci non riuscite e includerle in una chiamata successiva.

Le voci di risultati senza errori includono un valore `Id`, mentre le voci di risultati con errori includono i valori `ErrorCode` e `ErrorMessage`. `ErrorCode` descrive il tipo di errore. `ErrorMessage` fornisce ulteriori informazioni sull'errore. L'esempio seguente ha tre voci di risultati per una richiesta `PutEvents`. La seconda voce non ha esito positivo.

```
{
    "FailedEntryCount": 1,
    "Entries": [
        {
            "EventId": "11710aed-b79e-4468-a20b-bb3c0c3b4860"
        },
        {   "ErrorCode": "InternalFailure",
            "ErrorMessage": "Internal Service Failure"
        },
        {
            "EventId": "d804d26a-88db-4b66-9eaf-9a11c708ae82"
        }
    ]
}
```

> **Note**
>
> Se si utilizza `PutEvents` per pubblicare un evento su un bus di eventi che non esiste, EventBridge event matching non troverà una regola corrispondente e eliminerà l'evento. Sebbene EventBridge invierà una `200` risposta, non fallirà la richiesta né includerà l'evento nel `FailedEntryCount` valore della risposta alla richiesta.

Le voci non riuscite possono essere incluse nelle richieste `PutEvents` successive. In primo luogo, per determinare se vi sono voci non riuscite nella richiesta, verifica il parametro `FailedRecordCount` in `PutEventsResult`. Se è diverso da zero, puoi aggiungere ogni `Entry` che ha un `ErrorCode` non nullo a una richiesta successiva. L'esempio seguente mostra un semplice gestore di errori.

```
PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");
```

```
List<PutEventsRequestEntry> putEventsRequestEntryList = new ArrayList<>();
for (int i = 0; i < 3; i++) {
    putEventsRequestEntryList.add(requestEntry);
}

PutEventsRequest putEventsRequest = new PutEventsRequest();
putEventsRequest.withEntries(putEventsRequestEntryList);
PutEventsResult putEventsResult = awsEventsClient.putEvents(putEventsRequest);

while (putEventsResult.getFailedEntryCount() > 0) {
    final List<PutEventsRequestEntry> failedEntriesList = new ArrayList<>();
    final List<PutEventsResultEntry> PutEventsResultEntryList =
 putEventsResult.getEntries();
    for (int i = 0; i < PutEventsResultEntryList.size(); i++) {
        final PutEventsRequestEntry putEventsRequestEntry =
 putEventsRequestEntryList.get(i);
        final PutEventsResultEntry putEventsResultEntry =
 PutEventsResultEntryList.get(i);
        if (putEventsResultEntry.getErrorCode() != null) {
            failedEntriesList.add(putEventsRequestEntry);
        }
    }
    putEventsRequestEntryList = failedEntriesList;
    putEventsRequest.setEntries(putEventsRequestEntryList);
    putEventsResult = awsEventsClient.putEvents(putEventsRequest);
    }
```

Invio di eventi tramite AWS CLI

È possibile utilizzare il AWS CLI per inviare eventi personalizzati in EventBridge modo che possano essere elaborati. L'esempio seguente inserisce un evento personalizzato in EventBridge:

```
aws events put-events \
--entries '[{"Time": "2016-01-14T01:02:03Z", "Source": "com.mycompany.myapp",
 "Resources": ["resource1", "resource2"], "DetailType": "myDetailType", "Detail":
 "{ \"key1\": \"value1\", \"key2\": \"value2\" }"}]'
```

Puoi anche creare un file JSON contenente eventi personalizzati.

```
[
  {
```

```
    "Time": "2016-01-14T01:02:03Z",
    "Source": "com.mycompany.myapp",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }"
  }
]
```

Quindi, per utilizzare il AWS CLI per leggere le voci di questo file e inviare eventi, al prompt dei comandi digitate:

```
aws events put-events --entries file://entries.json
```

Calcolo delle dimensioni di iscrizione agli EventBridge PutEvents eventi Amazon

Puoi inviare eventi personalizzati a EventBridge utilizzando l'`PutEvents`azione. È possibile raggruppare in batch più voci di eventi in una richiesta per una maggiore efficacia. La dimensione totale delle voci deve essere inferiore a 256 KB. È possibile calcolare la dimensione delle voci prima dell'invio degli eventi.

> **Note**
>
> Il limite della dimensione viene imposto sulla voce. Anche se la voce è inferiore al limite di dimensione, l'evento in EventBridge è sempre maggiore della dimensione della voce a causa dei caratteri e delle chiavi necessari della rappresentazione JSON dell'evento. Per ulteriori informazioni, consulta EventBridge Eventi Amazon.

EventBridge calcola la `PutEventsRequestEntry` dimensione come segue:

- Se specificato, il parametro `Time` è di 14 byte.
- I parametri `Source` e `DetailType` sono il numero di byte per i relativi propri moduli con codifica UTF-8.
- Se specificato, il parametro `Detail` è il numero di byte del relativo modulo con codifica UTF-8.
- Se specificato, il parametro `Resources` è il numero di byte dei relativi moduli con codifica UTF-8.

Il seguente codice Java di esempio calcola le dimensioni di un determinato oggetto `PutEventsRequestEntry`.

```
int getSize(PutEventsRequestEntry entry) {
    int size = 0;
    if (entry.getTime() != null) {
        size += 14;
    }
    size += entry.getSource().getBytes(StandardCharsets.UTF_8).length;
    size += entry.getDetailType().getBytes(StandardCharsets.UTF_8).length;
    if (entry.getDetail() != null) {
        size += entry.getDetail().getBytes(StandardCharsets.UTF_8).length;
    }
```

```
    if (entry.getResources() != null) {
        for (String resource : entry.getResources()) {
            if (resource != null) {
                size += resource.getBytes(StandardCharsets.UTF_8).length;
            }
        }
    }
    return size;
}
```

> **Note**
>
> Se la dimensione della voce è superiore a 256 KB, consigliamo di caricare l'evento in un bucket di Amazon S3 e di includere `Object URL` nella voce `PutEvents`.

Eventi derivanti dai servizi AWS

Molti AWS servizi generano eventi che EventBridge ricevono. Quando un AWS servizio del tuo account emette un evento, questo passa al bus eventi predefinito del tuo account.

Erogazione di eventi tramite i AWS servizi

Ogni AWS servizio che genera eventi li invia come massimo impegno o EventBridge come tentativo di consegna duraturo.

- Per «consegna con la massima diligenza» si intende che il servizio tenta di inviare tutti gli eventi a EventBridge, ma in alcuni rari casi un evento potrebbe non essere consegnato.
- Per consegna durevole si intende che il servizio tenterà di fornire gli eventi con successo EventBridge almeno una volta.

  EventBridge accetterà tutti gli eventi validi in condizioni normali. Nei casi in cui gli eventi non possano essere consegnati a causa di un'interruzione del EventBridge servizio, verranno riprovati in un secondo momento dal AWS servizio per un massimo di 24 ore.

Una volta consegnato un evento EventBridge, lo EventBridge confronta con le regole e quindi segue la politica di riprova e l'eventuale coda di lettere non scritte specificata per le destinazioni dell'evento.

Per un elenco dei AWS servizi che generano eventi, consulta. ???

Accesso agli eventi AWS di servizio tramite AWS CloudTrail

AWS CloudTrail è un servizio che registra automaticamente eventi come le chiamate AWS API. È possibile creare EventBridge regole che utilizzano le informazioni di CloudTrail. Per ulteriori informazioni su CloudTrail, vedi Cos'è AWS CloudTrail? .

Tutti gli eventi che vengono consegnati da CloudTrail hanno `AWS API Call via CloudTrail` come valore per`detail-type`.

Per registrare eventi con un `detail-type` valore di`AWS API Call via CloudTrail`, è necessario un CloudTrail percorso con registrazione abilitata.

Quando si utilizza CloudTrail con Amazon S3, è necessario configurare la registrazione degli eventi relativi CloudTrail ai dati. Per ulteriori informazioni, consulta Abilitazione della registrazione CloudTrail degli eventi per i bucket e gli oggetti S3.

Alcune occorrenze nei AWS servizi possono essere segnalate EventBridge sia dal servizio stesso che da. CloudTrail Ad esempio, una chiamata all'API Amazon EC2 che avvia o interrompe un'istanza genera EventBridge eventi oltre a eventi. CloudTrail

CloudTrail supporta sia i chiamanti API che i proprietari di risorse per ricevere eventi nei loro bucket Amazon S3 creando percorsi e distribuisce eventi ai chiamanti API tramite. EventBridge I proprietari di risorse, oltre ai chiamanti API, possono monitorare le chiamate API tra account tramite. EventBridge CloudTraill'integrazione con EventBridge fornisce un modo conveniente per impostare flussi di lavoro automatizzati basati su regole in risposta agli eventi.

Non è possibile utilizzare AWS gli eventi di chiamata dell'API Put*Events di dimensioni superiori a 256 KB come modelli di eventi perché la dimensione massima di qualsiasi richiesta Put*Events è 256 KB. Per ulteriori informazioni sulle chiamate API che puoi utilizzare, consulta Servizi e integrazioni CloudTrail supportati.

Ricezione di eventi di gestione in sola lettura dai servizi AWS

È possibile impostare regole sul bus degli eventi predefinito o personalizzato per ricevere eventi di gestione in sola lettura dai servizi tramite. AWS CloudTrail Gli eventi di gestione forniscono visibilità sulle operazioni di gestione eseguite sulle risorse dell'account. AWS Queste operazioni sono definite anche operazioni del piano di controllo (control-plane). Per ulteriori informazioni, consulta Registrazione degli eventi di gestione nella Guida per l'utente di CloudTrail .

Per ogni regola nei router di eventi predefiniti o personalizzati, puoi impostare lo stato della regola per controllare i tipi di eventi da ricevere:

- Disattiva la regola in modo che gli eventi EventBridge non corrispondano alla regola.
- Abilita la regola in modo che gli eventi EventBridge corrispondano alla regola, ad eccezione degli eventi di AWS gestione in sola lettura forniti tramite. CloudTrail
- Abilita la regola in modo che tutti gli eventi EventBridge corrispondano alla regola, inclusi gli eventi di gestione in sola lettura forniti tramite. CloudTrail

Gli event bus partner non ricevono AWS eventi.

Alcuni aspetti da considerare quando si decide se ricevere eventi di gestione in sola lettura:

- Alcuni eventi di gestione di sola lettura, come AWS Key Management Service `GetKeyPolicy` and, o IAM `GetPolicy` and `GetRole` events`DescribeKey`, si verificano a un volume molto più elevato rispetto ai tipici eventi di modifica.
- È possibile che tu stia già ricevendo eventi di gestione in sola lettura, se tali eventi non iniziano con `Describe`, `Get` o `List`. Ad esempio, gli eventi delle seguenti AWS STS API sono eventi di modifica, anche se iniziano con il verbo: `Get`
  - `GetFederationToken`
  - `GetSessionToken`

  Per un elenco degli eventi di gestione in sola lettura che non rispettano la convenzione di denominazione, o di `List` denominazione `DescribeGet`, per servizi, vedere. AWS ???

Per creare una regola che riceva eventi di gestione in sola lettura utilizzando la CLI AWS

- Utilizza il comando `put-rule` per creare o aggiornare la regola e i parametri per:
  - Specificare che la regola appartiene al router di eventi predefinito o a uno specifico router di eventi personalizzato
  - Impostare lo stato della regola su `ENABLED_WITH_ALL_CLOUDTRAIL_MANAGEMENT_EVENTS`

```
aws events put-rule --name "ruleForManagementEvents" --event-bus-name
"default" --state "ENABLED_WITH_ALL_CLOUDTRAIL_MANAGEMENT_EVENTS"
```

> **Note**
>
> L'abilitazione di una regola per gli eventi di CloudWatch gestione è supportata solo tramite la AWS CLI e i AWS CloudFormation modelli.

Example

Nell'esempio seguente viene illustrato come cercare corrispondenze con specifici eventi. La best practice consiste nel definire una regola dedicata per la corrispondenza con eventi specifici, per garantire chiarezza e facilità di modifica.

In questo caso, la regola dedicata corrisponde all'evento di `AssumeRole` gestione di AWS Security Token Service.

```
{
    "source" : [ "aws.sts" ],
    "detail-type": ["AWS API Call via CloudTrail"],
    "detail" : {
        "eventName" : ["AssumeRole"]
    }
}
```

AWS servizi che generano eventi

La tabella seguente mostra AWS i servizi che generano eventi. Scegli il nome del servizio per visualizzare ulteriori informazioni sulla EventBridge collaborazione tra quel servizio e quello.

Ogni AWS servizio che genera eventi li invia come massimo impegno o EventBridge come tentativo di consegna duraturo. Per ulteriori informazioni, consulta ???.

Questa tabella include una rappresentazione dei AWS servizi a cui inviano eventi EventBridge, ma non include tutti i servizi. Per i servizi non elencati che inviano eventi a cui si inviano eventi EventBridge, si presuppone una consegna con il massimo impegno.

Servizio	Tipo di tentativo
Alexa for Business	Migliore
AWS Account Management	Migliore

Servizio	Tipo di tentativo
Amazon API Gateway	Migliore
AWS AppConfig	Migliore
Amazon AppFlow	Migliore
Application Auto Scaling	Migliore
AWS Application Cost Profiler	Migliore
AWS Application Migration Service	Migliore
Amazon Athena	Migliore
AWS Backup	Migliore
AWS Batch	Durevole
Amazon Braket	Durevole
AWS Certificate Manager	Migliore
Amazon Chime	Migliore
Directory del cloud Amazon	Migliore
AWS CloudFormation	Durevole
Amazon CloudFront	Migliore
AWS CloudHSM	Migliore
Amazon CloudSearch	Migliore
AWS CloudShell	Migliore
Eventi di AWS CloudTrail	Migliore
Amazon CloudWatch	Durevole

Servizio	Tipo di tentativo
Informazioni approfondite sulle CloudWatch applicazioni Amazon	Migliore
Amazon CloudWatch Internet Monitor	Migliore
CloudWatch Registri Amazon	Migliore
Amazon CloudWatch Synthetics	Migliore
AWS CodeArtifact	Durevole
AWS CodeBuild	Migliore
AWS CodeCommit	Migliore
AWS CodeDeploy	Migliore
Amazon CodeGuru Profiler	Migliore
AWS CodePipeline	Migliore
AWS CodeStar	Migliore
CodeConnections	Migliore
Amazon Cognito Identity	Migliore
Pool di utenti Amazon Cognito	Migliore
Amazon Cognito Sync	Migliore
AWS Config	Migliore
Amazon Connect	Migliore
Amazon Connect Voice ID	Migliore
AWS Control Tower	Migliore
AWS Database Migration Service	Migliore

Servizio	Tipo di tentativo
AWS Data Exchange	Migliore
Amazon Data Lifecycle Manager	Migliore
AWS Data Pipeline	Migliore
AWS DataSync	Migliore
AWS Device Farm	Migliore
Amazon DevOps Guru	Migliore
AWS Direct Connect	Migliore
AWS Directory Service	Migliore
Amazon DynamoDB	Migliore
AWS Elastic Beanstalk	Migliore
Amazon Elastic Block Store	Migliore
Modifiche del volume Amazon Elastic Block Store	Migliore
Amazon ElastiCache	Migliore
Amazon Elastic Compute Cloud (Amazon EC2)	Migliore
Dimensionamento automatico Amazon EC2	Migliore
Amazon EC2 Fleet	Migliore
Interruzione dell'istanza spot Amazon EC2	Migliore
Amazon Elastic Container Registry	Migliore
Amazon Elastic Container Service	Durevole
AWS Elastic Disaster Recovery	Migliore

Servizio	Tipo di tentativo
Amazon Elastic File System	Migliore
Amazon Elastic Kubernetes Service	Migliore
Sistema di bilanciamento del carico elastico	Migliore
Amazon Elastic MapReduce	Migliore
Amazon Elastic Transcoder	Migliore
AWS Elemental MediaConnect	Migliore
AWS Elemental MediaConvert	Durevole
AWS Elemental MediaLive	Migliore
AWS Elemental MediaPackage	Migliore
AWS Elemental MediaStore	Durevole
Amazon EMR	Migliore
Amazon EMR su EKS	Migliore
Amazon EMR Serverless	Migliore
Regole EventBridge pianificate di Amazon	Durevole
EventBridge Schemi Amazon	Migliore
AWS Fault Injection Service	Migliore
Previsione	Migliore
Amazon GameLift	Migliore
AWS Glue	Migliore
AWS Glue DataBrew	Migliore

Servizio	Tipo di tentativo
AWS Ground Station	Migliore
Amazon GuardDuty	Migliore
AWS Health	Migliore
AWS HealthLake	Durevole
AWS Identity and Access Management (IAM)	Migliore
IAM Access Analyzer	Migliore
Amazon Inspector Classic	Migliore
Amazon Inspector	Migliore
AWS IoT	Migliore
AWS IoT Analytics	Durevole
AWS IoT Greengrass V1	Migliore
AWS IoT Greengrass V2	Migliore
Amazon Interactive Video Service	Migliore
Amazon Kinesis	Migliore
Amazon Data Firehose	Migliore
AWS Key Management Service Eliminazione CMK	Durevole
AWS Key Management Service Rotazione CMK	Migliore
AWS Key Management Service scadenza del materiale chiave importato	Migliore
AWS Lambda	Migliore

Servizio	Tipo di tentativo
Servizio di posizione Amazon	Durevole
Amazon Machine Learning	Migliore
Amazon Macie	Migliore
Blockchain gestita da Amazon	Migliore
AWS Managed Services	Migliore
AWS Management Console Accedi	Migliore
AWS Marketplace della misurazione	Migliore
AWS Migration Hub	Migliore
AWS Migration Hub Refactor Spaces	Migliore
AWS Monitoraggio	Migliore
AWS Network Manager	Migliore
OpenSearch Servizio Amazon	Migliore
AWS OpsWorks	Durevole
AWS OpsWorks CM	Migliore
AWS Organizations	Migliore
Amazon Polly	Migliore
AWS Private Certificate Authority	Migliore
AWS Proton	Migliore
Amazon QLDB	Durevole
Amazon QuickSight	Migliore

Servizio	Tipo di tentativo
Amazon RDS	Migliore
AWS Cestino di riciclaggio	Migliore
Amazon Redshift	Durevole
API dati di Amazon Redshift	Migliore
Amazon Redshift Serverless	Migliore
AWS Resource Access Manager	Migliore
AWS Resource Groups	Migliore
AWS Resource Groups Tagging API	Migliore
Amazon Route 53	Migliore
Preparazione al ripristino di Amazon Route 53	Migliore
Amazon SageMaker	Migliore
Savings Plans	Migliore
AWS Secrets Manager	Migliore
AWS Security Hub	Durevole
AWS Security Token Service	Migliore
AWS Server Migration Service	Migliore
AWS Service Catalog	Migliore
AWS Signer	Durevole
Amazon Simple Email Service	Migliore
Amazon Simple Storage Service (Amazon S3)	Durevole

Servizio	Tipo di tentativo
Amazon S3 Glacier	Migliore
Amazon S3 su Outposts	Migliore
Amazon Simple Queue Service	Migliore
Amazon Simple Notification Service	Migliore
Amazon Simple Workflow Service	Migliore
AWS Step Functions	Migliore
AWS Storage Gateway	Durevole
AWS Support	Migliore
AWS Systems Manager	Migliore
Amazon Transcribe	Migliore
AWS Transfer Family	Migliore
AWS Transit Gateway	Migliore
Amazon Translate	Durevole
AWS Trusted Advisor	Migliore
AWS WAF	Migliore
AWS WAF Regionale	Migliore
AWS Well-Architected Tool	Migliore
Amazon WorkDocs	Migliore
Amazon WorkSpaces	Migliore
AWS X-Ray	Migliore

Eventi di gestione generati dai AWS servizi

In generale, le API che generano eventi di gestione (o in sola lettura) iniziano con i verbi `Describe`, `Get` o `List`. La tabella seguente elenca AWS i servizi e gli eventi di gestione da essi generati che non seguono questa convenzione di denominazione. Per ulteriori informazioni sugli eventi di gestione, consulta ???.

Eventi di gestione che non iniziano con **`Describe`**, **`Get`** o **`List`**

Nella tabella seguente sono elencati AWS i servizi e gli eventi di gestione da essi generati che non seguono le convenzioni di denominazione tipiche che iniziano con`Describe`, `Get` o. `List`

Servizio	Nome evento	Tipo di evento
Alexa for Business	ResolveRoom	Chiamata API
Alexa for Business	SearchAddressBooks	Chiamata API
Alexa for Business	SearchContacts	Chiamata API
Alexa for Business	SearchDevices	Chiamata API
Alexa for Business	SearchProfiles	Chiamata API
Alexa for Business	SearchRooms	Chiamata API
Alexa for Business	SearchSkillGroups	Chiamata API
Alexa for Business	SearchUsers	Chiamata API
Sistema di analisi degli accessi AWS IAM	ValidatePolicy	Chiamata API
AWS AdSpace Camere pulite	BatchGetSchema	Chiamata API
AWS Amplify Generatore di interfacce utente	ExportComponents	Chiamata API
AWS Amplify Generatore di interfacce utente	ExportForms	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS Amplify Generatore di interfacce utente	ExportThemes	Chiamata API
OpenSearch Servizio Amazon	BatchGetCollection	Chiamata API
Amazon API Gateway	ExportApi	Chiamata API
AWS AppConfig	ValidateConfiguration	Chiamata API
Amazon AppFlow	RetrieveConnectorData	Chiamata API
Informazioni approfondite sulle CloudWatch applicazioni Amazon	UpdateApplicationDashboardConfiguration	Chiamata API
Amazon Athena	BatchGetNamedQuery	Chiamata API
Amazon Athena	BatchGetPreparedStatement	Chiamata API
Amazon Athena	BatchGetQueryExecution	Chiamata API
Amazon Athena	CheckQueryCompatibility	Chiamata API
Amazon Athena	ExportNotebook	Chiamata API
AWS Auto Scaling	AreScalableTargetsRegistered	Chiamata API
AWS Auto Scaling	Test	Chiamata API
Marketplace AWS	SearchAgreements	Chiamata API
AWS Backup	CreateLegalHold	Chiamata API
AWS Backup	ExportBackupPlanTemplate	Chiamata API
AWS Backup gateway	TestHypervisorConfiguration	Chiamata API
AWS Billing and Cost Management	AWSPaymentInstrumentGateway.Ottieni	Azione da console

Servizio	Nome evento	Tipo di evento
AWS Billing and Cost Management	AWSPaymentPortalService.DescribeMakePaymentPage	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.DescribePaymentsDashboard	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetAccountPreferences	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetAdvancePaySummary	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetAsoBulkDownload	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetBillingContactAddress	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetDocuments	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetEligiblePaymentInstruments	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetEntitiesByIds	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetFundingDocuments	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetKybcValidationStatus	Azione da console

Servizio	Nome evento	Tipo di evento
AWS Billing and Cost Management	AWSPaymentPortalService.GetOneTimePasswordStatus	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentHistory	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileByArn	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileCurrencies	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfiles	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileServiceProviders	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentsDue	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetRemittanceInformation	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTaxInvoiceMetadata	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTermsAndConditionsForProgramGroup	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTransactionsHistory	Azione da console

Servizio	Nome evento	Tipo di evento
AWS Billing and Cost Management	AWSPaymentPortalService.GetUnappliedFunds	Azione da console
AWS Billing and Cost Management	AWSPaymentPortalService.GetUnpaidInvoices	Azione da console
AWS Billing and Cost Management	AWSPaymentPreferenceGateway.Ottenere	Azione da console
AWS Billing and Cost Management	CancelBulkDownload	Azione da console
AWS Billing and Cost Management	DownloadCommercialInvoice	Azione da console
AWS Billing and Cost Management	DownloadCsv	Azione da console
AWS Billing and Cost Management	DownloadDoc	Azione da console
AWS Billing and Cost Management	Scarica CSV ForBillingPeriod	Azione da console
AWS Billing and Cost Management	DownloadPaymentHistory	Azione da console
AWS Billing and Cost Management	DownloadRegistrationDocument	Azione da console
AWS Billing and Cost Management	DownloadTaxInvoice	Azione da console
AWS Billing and Cost Management	FindBankRedirectPaymentInstruments	Azione da console
AWS Billing and Cost Management	Trova CSV ForBillingPeriod	Azione da console

Servizio	Nome evento	Tipo di evento
AWS Billing and Cost Management	ValidateReportDestination	Azione da console
AWS Billing and Cost Management	VerifyChinaPaymentEligibility	Azione da console
Amazon Braket	SearchCompilations	Chiamata API
Amazon Braket	SearchDevices	Chiamata API
Amazon Braket	SearchQuantumTasks	Chiamata API
Amazon Connect Cases	BatchGetField	Chiamata API
Amazon Connect Cases	SearchCases	Chiamata API
Amazon Connect Cases	SearchRelatedItems	Chiamata API
Amazon Chime	RetrieveDataExports	Chiamata API
Amazon Chime	SearchChannels	Chiamata API
Identità di SDK di Amazon Chime	DeleteProfile	Evento del servizio
Identità di SDK di Amazon Chime	DeleteWorkTalkAccount	Evento del servizio
AWS Camere pulite	BatchGetSchema	Chiamata API
Directory del cloud Amazon	BatchRead	Chiamata API
Directory del cloud Amazon	LookupPolicy	Chiamata API
AWS CloudFormation	DetectStackDrift	Chiamata API
AWS CloudFormation	DetectStackResourceDrift	Chiamata API
AWS CloudFormation	DetectStackSetDrift	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS CloudFormation	EstimateTemplateCost	Chiamata API
AWS CloudFormation	ValidateTemplate	Chiamata API
AWS CloudShell	RedeemCode	Chiamata API
AWS CloudTrail	LookupEvents	Chiamata API
AWS CodeArtifact	ReadFromRepository	Chiamata API
AWS CodeArtifact	SearchPackages	Chiamata API
AWS CodeArtifact	VerifyResourcesExistForTagris	Chiamata API
AWS CodeBuild	BatchGetBuildBatches	Chiamata API
AWS CodeBuild	BatchGetBuilds	Chiamata API
AWS CodeBuild	BatchGetProjects	Chiamata API
AWS CodeBuild	BatchGetReportGroups	Chiamata API
AWS CodeBuild	BatchGetReports	Chiamata API
AWS CodeBuild	BatchPutCodeCoverages	Chiamata API
AWS CodeBuild	BatchPutTestCases	Chiamata API
AWS CodeBuild	RequestBadge	Evento del servizio
AWS CodeCommit	BatchDescribeMergeConflicts	Chiamata API
AWS CodeCommit	BatchGetCommits	Chiamata API
AWS CodeCommit	BatchGetPullRequests	Chiamata API
AWS CodeCommit	BatchGetRepositories	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS CodeCommit	EvaluatePullRequestApprovalRules	Chiamata API
AWS CodeCommit	GitPull	Chiamata API
AWS CodeDeploy	BatchGetApplicationRevisions	Chiamata API
AWS CodeDeploy	BatchGetApplications	Chiamata API
AWS CodeDeploy	BatchGetDeploymentGroups	Chiamata API
AWS CodeDeploy	BatchGetDeploymentInstances	Chiamata API
AWS CodeDeploy	BatchGetDeployments	Chiamata API
AWS CodeDeploy	BatchGetDeploymentTargets	Chiamata API
AWS CodeDeploy	BatchGetOnPremisesInstances	Chiamata API
Amazon CodeGuru Profiler	BatchGetFrameMetricData	Chiamata API
Amazon CodeGuru Profiler	SubmitFeedback	Chiamata API
AWS CodePipeline	PollForJobs	Chiamata API
AWS CodePipeline	PollForThirdPartyJobs	Chiamata API
CodeConnections	StartAppRegistrationHandshake	Chiamata API
CodeConnections	Inizia a AuthHandshake	Chiamata API
CodeConnections	ValidateHostWebhook	Chiamata API
Amazon CodeWhisperer	CreateCodeScan	Chiamata API
Amazon CodeWhisperer	CreateProfile	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon CodeWhisperer	CreateUploadUrl	Chiamata API
Amazon CodeWhisperer	GenerateRecommendations	Chiamata API
Amazon CodeWhisperer	UpdateProfile	Chiamata API
Amazon Cognito Identity	LookupDeveloperIdentity	Chiamata API
Pool di utenti Amazon Cognito	AdminGetDevice	Chiamata API
Pool di utenti Amazon Cognito	AdminGetUser	Chiamata API
Pool di utenti Amazon Cognito	AdminListDevices	Chiamata API
Pool di utenti Amazon Cognito	AdminListGroupsForUser	Chiamata API
Pool di utenti Amazon Cognito	AdminListUserAuthEvents	Chiamata API
Pool di utenti Amazon Cognito	Beta_Authorize_GET	Evento del servizio
Pool di utenti Amazon Cognito	Confirm_GET	Evento del servizio
Pool di utenti Amazon Cognito	ConfirmForgotPassw ord_OTTIENI	Evento del servizio
Pool di utenti Amazon Cognito	Error_GET	Evento del servizio
Pool di utenti Amazon Cognito	ForgotPassword_OTTIENI	Evento del servizio
Pool di utenti Amazon Cognito	IntrospectToken	Chiamata API
Pool di utenti Amazon Cognito	Login_Error_POST	Evento del servizio
Pool di utenti Amazon Cognito	Login_GET	Evento del servizio
Pool di utenti Amazon Cognito	Mfa_GET	Evento del servizio
Pool di utenti Amazon Cognito	MfaOption_OTTIENI	Evento del servizio
Pool di utenti Amazon Cognito	ResetPassword_OTTIENI	Evento del servizio

Servizio	Nome evento	Tipo di evento
Pool di utenti Amazon Cognito	Signup_GET	Evento del servizio
Pool di utenti Amazon Cognito	UserInfo_OTTIENI	Evento del servizio
Pool di utenti Amazon Cognito	UserInfo_PUBBLICA	Evento del servizio
Amazon Cognito Sync	BulkPublish	Chiamata API
Amazon Comprehend	BatchContainsPiiEntities	Chiamata API
Amazon Comprehend	BatchDetectDominantLanguage	Chiamata API
Amazon Comprehend	BatchDetectEntities	Chiamata API
Amazon Comprehend	BatchDetectKeyPhrases	Chiamata API
Amazon Comprehend	BatchDetectPiiEntities	Chiamata API
Amazon Comprehend	BatchDetectSentiment	Chiamata API
Amazon Comprehend	BatchDetectSyntax	Chiamata API
Amazon Comprehend	BatchDetectTargetedSentiment	Chiamata API
Amazon Comprehend	ClassifyDocument	Chiamata API
Amazon Comprehend	ContainsPiiEntities	Chiamata API
Amazon Comprehend	DetectDominantLanguage	Chiamata API
Amazon Comprehend	DetectEntities	Chiamata API
Amazon Comprehend	DetectKeyPhrases	Chiamata API
Amazon Comprehend	DetectPiiEntities	Chiamata API
Amazon Comprehend	DetectSentiment	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon Comprehend	DetectSyntax	Chiamata API
Amazon Comprehend	DetectTargetedSentiment	Chiamata API
Amazon Comprehend	DetectToxicContent	Chiamata API
AWS Compute Optimizer	ExportAutoScalingGroupRecommendations	Chiamata API
AWS Compute Optimizer	Esporta EBS VolumeRecommendations	Chiamata API
AWS Compute Optimizer	Esporta C InstanceRecommendations	Chiamata API
AWS Compute Optimizer	Esporta ECS ServiceRecommendations	Chiamata API
AWS Compute Optimizer	ExportLambdaFunctionRecommendations	Chiamata API
AWS Compute Optimizer	Esporta RDS InstanceRecommendations	Chiamata API
AWS Config	BatchGetAggregateResourceConfig	Chiamata API
AWS Config	BatchGetResourceConfig	Chiamata API
AWS Config	SelectAggregateResourceConfig	Chiamata API
AWS Config	SelectResourceConfig	Chiamata API
Amazon Connect	AdminGetEmergencyAccessToken	Chiamata API
Amazon Connect	SearchQueues	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon Connect	SearchRoutingProfiles	Chiamata API
Amazon Connect	SearchSecurityProfiles	Chiamata API
Amazon Connect	SearchUsers	Chiamata API
AWS Glue DataBrew	SendProjectSessionAction	Chiamata API
AWS Data Pipeline	EvaluateExpression	Chiamata API
AWS Data Pipeline	QueryObjects	Chiamata API
AWS Data Pipeline	ValidatePipelineDefinition	Chiamata API
AWS DataSync	VerifyResourcesExistForTagris	Chiamata API
AWS DeepLens	BatchGetDevice	Chiamata API
AWS DeepLens	BatchGetModel	Chiamata API
AWS DeepLens	BatchGetProject	Chiamata API
AWS DeepLens	CreateDeviceCertificates	Chiamata API
AWS DeepRacer	AdminGetAccountConfig	Chiamata API
AWS DeepRacer	AdminListAssociatedUsers	Chiamata API
AWS DeepRacer	TestRewardFunction	Chiamata API
AWS DeepRacer	VerifyResourcesExistForTagris	Chiamata API
Amazon Detective	BatchGetGraphMemberDatasources	Chiamata API
Amazon Detective	BatchGetMembershipDatasources	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon Detective	SearchGraph	Chiamata API
Amazon DevOps Guru	SearchInsights	Chiamata API
Amazon DevOps Guru	SearchOrganizationInsights	Chiamata API
AWS Database Migration Service	BatchStartRecommendations	Chiamata API
AWS Database Migration Service	ModifyRecommendation	Chiamata API
AWS Database Migration Service	StartRecommendations	Chiamata API
AWS Database Migration Service	VerifyResourcesExistForTagris	Chiamata API
AWS Directory Service	VerifyTrust	Chiamata API
Amazon Elastic Compute Cloud	ConfirmProductInstance	Chiamata API
Amazon Elastic Compute Cloud	ReportInstanceStatus	Chiamata API
Amazon Elastic Container Registry	BatchCheckLayerAvailability	Chiamata API
Amazon Elastic Container Registry	BatchGetImage	Chiamata API
Amazon Elastic Container Registry	BatchGetImageReferrer	Chiamata API
Amazon Elastic Container Registry	BatchGetRepositoryScanningConfiguration	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon Elastic Container Registry	DryRunEvent	Evento del servizio
Amazon Elastic Container Registry	PolicyExecutionEvent	Evento del servizio
Amazon Elastic Container Registry Public	BatchCheckLayerAvailability	Chiamata API
Amazon Elastic Container Service	DiscoverPollEndpoint	Chiamata API
Amazon Elastic Container Service	FindSubfleetRoute	Chiamata API
Amazon Elastic Container Service	ValidateResources	Chiamata API
Amazon Elastic Container Service	VerifyTaskSetsExist	Chiamata API
Amazon Elastic Kubernetes Service	AccessKubernetesApi	Chiamata API
AWS Elastic Beanstalk	CheckDNSAvailability	Chiamata API
AWS Elastic Beanstalk	RequestEnvironmentInfo	Chiamata API
AWS Elastic Beanstalk	RetrieveEnvironmentInfo	Chiamata API
AWS Elastic Beanstalk	ValidateConfigurationSettings	Chiamata API
Amazon Elastic File System	NewClientConnection	Evento del servizio
Amazon Elastic File System	UpdateClientConnection	Evento del servizio
Amazon Elastic Transcoder	ReadJob	Chiamata API
Amazon Elastic Transcoder	ReadPipeline	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon Elastic Transcoder	ReadPreset	Chiamata API
Amazon EventBridge	TestEventPattern	Chiamata API
Amazon EventBridge	TestScheduleExpression	Chiamata API
Amazon FinSpace API	BatchListCatalogNodesByDataset	Chiamata API
Amazon FinSpace API	BatchListNodesByDataset	Chiamata API
Amazon FinSpace API	BatchValidateAccess	Chiamata API
Amazon FinSpace API	CreateAuditRecordsQuery	Chiamata API
Amazon FinSpace API	SearchDatasets	Chiamata API
Amazon FinSpace API	SearchDatasetsV	Chiamata API
Amazon FinSpace API	ValidateIdToken	Chiamata API
AWS Firewall Manager	DisassociateAdminAccount	Chiamata API
Amazon Forecast	InvokeForecastEndpoint	Chiamata API
Amazon Forecast	QueryFeature	Chiamata API
Amazon Forecast	QueryForecast	Chiamata API
Amazon Forecast	QueryWhatIfForecast	Chiamata API
Amazon Forecast	VerifyResourcesExistForTagris	Chiamata API
Amazon Fraud Detector	BatchGetVariable	Chiamata API
Amazon Fraud Detector	VerifyResourcesExistForTagris	Chiamata API
FreeRTOS	VerifyEmailAddress	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon GameLift	RequestUploadCredentials	Chiamata API
Amazon GameLift	ResolveAlias	Chiamata API
Amazon GameLift	SearchGameSessions	Chiamata API
Amazon GameLift	ValidateMatchmakingRuleSet	Chiamata API
Amazon GameSparks	ExportSnapshot	Chiamata API
Servizio di posizione Amazon	BatchGetDevicePosition	Chiamata API
Servizio di posizione Amazon	CalculateRoute	Chiamata API
Servizio di posizione Amazon	CalculateRouteMatrix	Chiamata API
Servizio di posizione Amazon	SearchPlaceIndexForPosition	Chiamata API
Servizio di posizione Amazon	SearchPlaceIndexForSuggestions	Chiamata API
Servizio di posizione Amazon	SearchPlaceIndexForText	Chiamata API
Amazon S3 Glacier	InitiateJob	Chiamata API
AWS Glue	BatchGetBlueprints	Chiamata API
AWS Glue	BatchGetColumnStatisticsForTable	Chiamata API
AWS Glue	BatchGetCrawlers	Chiamata API
AWS Glue	BatchGetCustomEntityTypes	Chiamata API
AWS Glue	BatchGetDataQualityResult	Chiamata API
AWS Glue	BatchGetDevEndpoints	Chiamata API
AWS Glue	BatchGetJobs	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS Glue	BatchGetMLTransform	Chiamata API
AWS Glue	BatchGetPartition	Chiamata API
AWS Glue	BatchGetTriggers	Chiamata API
AWS Glue	BatchGetWorkflows	Chiamata API
AWS Glue	QueryJobRuns	Chiamata API
AWS Glue	QueryJobRunsAggregated	Chiamata API
AWS Glue	QueryJobs	Chiamata API
AWS Glue	QuerySchemaVersion Metadata	Chiamata API
AWS Glue	SearchTables	Chiamata API
AWS HealthLake	ReadResource	Chiamata API
AWS HealthLake	SearchWithGet	Chiamata API
AWS HealthLake	SearchWithPost	Chiamata API
AWS Identity and Access Management	GenerateCredentialReport	Chiamata API
AWS Identity and Access Management	GenerateOrganizationsAccess Report	Chiamata API
AWS Identity and Access Management	GenerateServiceLas tAccessedDetails	Chiamata API
AWS Identity and Access Management	SimulateCustomPolicy	Chiamata API
AWS Identity and Access Management	SimulatePrincipalPolicy	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS Archivio di identità	IsMemberInGroups	Chiamata API
AWS Autenticazione di Identity Store	BatchGetSession	Chiamata API
Amazon Inspector Classic	PreviewAgents	Chiamata API
Amazon Inspector Classic	BatchGetAccountStatus	Chiamata API
Amazon Inspector Classic	BatchGetFreeTrialInfo	Chiamata API
Amazon Inspector Classic	BatchGetMember	Chiamata API
Fatturazione AWS	ValidateDocumentDeliveryS3 LocationInfo	Chiamata API
AWS IoT	SearchIndex	Chiamata API
AWS IoT	TestAuthorization	Chiamata API
AWS IoT	TestInvokeAuthorizer	Chiamata API
AWS IoT	ValidateSecurityProfileBeha viors	Chiamata API
AWS IoT Analytics	SampleChannelData	Chiamata API
AWS IoT SiteWise	GatewaysVerifyResourcesExis tForTagrisInternal	Chiamata API
AWS IoT Things Graph	SearchEntities	Chiamata API
AWS IoT Things Graph	SearchFlowExecutions	Chiamata API
AWS IoT Things Graph	SearchFlowTemplates	Chiamata API
AWS IoT Things Graph	SearchSystemInstances	Chiamata API
AWS IoT Things Graph	SearchSystemTemplates	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS IoT Things Graph	SearchThings	Chiamata API
AWS IoT TwinMaker	ExecuteQuery	Chiamata API
AWS IoT Wireless	CreateNetworkAnalyzerConfiguration	Chiamata API
AWS IoT Wireless	DeleteNetworkAnalyzerConfiguration	Chiamata API
AWS IoT Wireless	DeregisterWirelessDevice	Chiamata API
Amazon Interactive Video Service	BatchGetChannel	Chiamata API
Amazon Interactive Video Service	BatchGetStreamKey	Chiamata API
Amazon Kendra	BatchGetDocumentStatus	Chiamata API
Amazon Kendra	Query	Chiamata API
Servizio gestito da Amazon per Apache Flink	DiscoverInputSchema	Chiamata API
AWS Key Management Service	Decrypt	Chiamata API
AWS Key Management Service	Crittografa	Chiamata API
AWS Key Management Service	GenerateDataKey	Chiamata API
AWS Key Management Service	GenerateDataKeyPair	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS Key Management Service	GenerateDataKeyPairWithoutPlaintext	Chiamata API
AWS Key Management Service	GenerateDataKeyWithoutPlaintext	Chiamata API
AWS Key Management Service	GenerateMac	Chiamata API
AWS Key Management Service	GenerateRandom	Chiamata API
AWS Key Management Service	ReEncrypt	Chiamata API
AWS Key Management Service	Sign	Chiamata API
AWS Key Management Service	Verifica	Chiamata API
AWS Key Management Service	VerifyMac	Chiamata API
AWS Lake Formation	SearchDatabasesByTag LF	Chiamata API
AWS Lake Formation	SearchTablesByTag LF	Chiamata API
AWS Lake Formation	StartQueryPlanning	Chiamata API
Amazon Lex	BatchCreateCustomVocabularyItem	Chiamata API
Amazon Lex	BatchDeleteCustomVocabularyItem	Chiamata API
Amazon Lex	BatchUpdateCustomVocabularyItem	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon Lex	DeleteCustomVocabulary	Chiamata API
Amazon Lex	SearchAssociatedTranscripts	Chiamata API
Amazon Lightsail	Crea GUI SessionAccessDetails	Chiamata API
Amazon Lightsail	DownloadDefaultKeyPair	Chiamata API
Amazon Lightsail	IsVpcPeered	Chiamata API
CloudWatch Registri Amazon	FilterLogEvents	Chiamata API
Amazon Macie	BatchGetCustomDataIdentifiers	Chiamata API
Amazon Macie	UpdateFindingsFilter	Chiamata API
AWS Elemental MediaConnect	ManagedDescribeFlow	Chiamata API
AWS Elemental MediaConnect	PrivateDescribeFlowMeta	Chiamata API
AWS Application Migration Service	OperationalDescribeJobLogItems	Chiamata API
AWS Application Migration Service	OperationalDescribeJobs	Chiamata API
AWS Application Migration Service	OperationalDescribeReplicationConfigurationTemplates	Chiamata API
AWS Application Migration Service	OperationalDescribeSourceServer	Chiamata API
AWS Application Migration Service	OperationalGetLaunchConfiguration	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS Application Migration Service	OperationalListSourceServers	Chiamata API
AWS Application Migration Service	VerifyClientRoleForMgn	Chiamata API
AWS HealthOmics	VerifyResourceExists	Chiamata API
AWS HealthOmics	VerifyResourcesExistForTagris	Chiamata API
Amazon Polly	SynthesizeLongSpeech	Chiamata API
Amazon Polly	SynthesizeSpeech	Chiamata API
Amazon Polly	SynthesizeSpeechGet	Chiamata API
AWS servizio che fornisce reti private gestite	Ping	Chiamata API
AWS Proton	DeleteEnvironmentTemplateVersion	Chiamata API
AWS Proton	DeleteServiceTemplateVersion	Chiamata API
Amazon QLDB	ShowCatalog	Chiamata API
Amazon QuickSight	GenerateEmbedUrlForAnonymousUser	Chiamata API
Amazon QuickSight	GenerateEmbedUrlForRegisteredUser	Chiamata API
Amazon QuickSight	QueryDatabase	Evento del servizio
Amazon QuickSight	SearchAnalyses	Chiamata API
Amazon QuickSight	SearchDashboards	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon QuickSight	SearchDataSets	Chiamata API
Amazon QuickSight	SearchDataSources	Chiamata API
Amazon QuickSight	SearchFolders	Chiamata API
Amazon QuickSight	SearchGroups	Chiamata API
Amazon QuickSight	SearchUsers	Chiamata API
Amazon Relational Database Service	DownloadCompleteDB LogFile	Chiamata API
Amazon Relational Database Service	Scarica DB LogFilePortion	Chiamata API
Amazon Rekognition	CompareFaces	Chiamata API
Amazon Rekognition	DetectCustomLabels	Chiamata API
Amazon Rekognition	DetectFaces	Chiamata API
Amazon Rekognition	DetectLabels	Chiamata API
Amazon Rekognition	DetectModerationLabels	Chiamata API
Amazon Rekognition	DetectProtectiveEquipment	Chiamata API
Amazon Rekognition	DetectText	Chiamata API
Amazon Rekognition	RecognizeCelebrities	Chiamata API
Amazon Rekognition	SearchFaces	Chiamata API
Amazon Rekognition	SearchFacesByImage	Chiamata API
Amazon Rekognition	SearchUsers	Chiamata API
Amazon Rekognition	SearchUsersByImage	Chiamata API

Servizio	Nome evento	Tipo di evento
Esploratore di risorse AWS	BatchGetView	Chiamata API
Esploratore di risorse AWS	Cerca	Chiamata API
AWS Resource Groups	SearchResources	Chiamata API
AWS Resource Groups	ValidateResourceSharing	Chiamata API
AWS RoboMaker	BatchDescribeSimulationJob	Chiamata API
Amazon Route 53	TestDNSAnswer	Chiamata API
Domini Amazon Route 53	checkAvailabilities	Chiamata API
Domini Amazon Route 53	CheckDomainAvailability	Chiamata API
Domini Amazon Route 53	checkDomainTransferability	Chiamata API
Domini Amazon Route 53	CheckDomainTransferability	Chiamata API
Domini Amazon Route 53	isEmailReachable	Chiamata API
Domini Amazon Route 53	searchDomains	Chiamata API
Domini Amazon Route 53	sendVerificationMessage	Chiamata API
Domini Amazon Route 53	ViewBilling	Chiamata API
Domini Amazon Route 53	viewBilling	Chiamata API
Amazon CloudWatch RUM	BatchGetRumMetricDefinitions	Chiamata API
Amazon Simple Storage Service	echo	Chiamata API
Amazon Simple Storage Service	GenerateInventory	Evento del servizio
Amazon SageMaker	BatchDescribeModelPackage	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon SageMaker	DeleteModelCard	Chiamata API
Amazon SageMaker	QueryLineage	Chiamata API
Amazon SageMaker	RenderUiTemplate	Chiamata API
Amazon SageMaker	Cerca	Chiamata API
EventBridge Schemi Amazon	ExportSchema	Chiamata API
EventBridge Schemi Amazon	SearchSchemas	Chiamata API
Amazon SimpleDB	DomainMetadata	Chiamata API
AWS Secrets Manager	ValidateResourcePolicy	Chiamata API
AWS Service Catalog	ScanProvisionedProducts	Chiamata API
AWS Service Catalog	SearchProducts	Chiamata API
AWS Service Catalog	SearchProductsAsAdmin	Chiamata API
AWS Service Catalog	SearchProvisionedProducts	Chiamata API
Amazon SES	BatchGetMetricData	Chiamata API
Amazon SES	TestRenderEmailTemplate	Chiamata API
Amazon SES	TestRenderTemplate	Chiamata API
Amazon Simple Notification Service	CheckIfPhoneNumber IsOptedOut	Chiamata API
AWS SQL Workbench	BatchGetNotebookCell	Chiamata API
AWS SQL Workbench	ExportNotebook	Chiamata API
Amazon EC2 Systems Manager	ExecuteApi	Chiamata API

Servizio	Nome evento	Tipo di evento
AWS Systems Manager Incident Manager	DeleteContactChannel	Chiamata API
AWS IAM Identity Center	IsMemberInGroup	Chiamata API
AWS IAM Identity Center	SearchGroups	Chiamata API
AWS IAM Identity Center	SearchUsers	Chiamata API
AWS STS	AssumeRole	Chiamata API
AWS STS	AssumeRoleWithSAML	Chiamata API
AWS STS	AssumeRoleWithWebIdentity	Chiamata API
AWS STS	DecodeAuthorizationMessage	Chiamata API
AWS Impostazioni fiscali	BatchGetTaxExemptions	Chiamata API
AWS WAFV2	CheckCapacity	Chiamata API
AWS WAFV2	GenerateMobileSdkReleaseUrl	Chiamata API
AWS Well-Architected Tool	ExportLens	Chiamata API
AWS Well-Architected Tool	TagResource	Chiamata API
AWS Well-Architected Tool	UntagResource	Chiamata API
AWS Well-Architected Tool	UpdateGlobalSettings	Chiamata API
Amazon Connect Wisdom	QueryAssistant	Chiamata API
Amazon Connect Wisdom	SearchContent	Chiamata API
Amazon Connect Wisdom	SearchSessions	Chiamata API
Amazon WorkDocs	AbortDocumentVersionUpload	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon WorkDocs	AddUsersToGroup	Chiamata API
Amazon WorkDocs	BatchGetUsers	Chiamata API
Amazon WorkDocs	CheckAlias	Chiamata API
Amazon WorkDocs	CompleteDocumentVersionUpload	Chiamata API
Amazon WorkDocs	CreateAnnotation	Chiamata API
Amazon WorkDocs	CreateComment	Chiamata API
Amazon WorkDocs	CreateFeedbackRequest	Chiamata API
Amazon WorkDocs	CreateFolder	Chiamata API
Amazon WorkDocs	CreateGroup	Chiamata API
Amazon WorkDocs	CreateShare	Chiamata API
Amazon WorkDocs	CreateUser	Chiamata API
Amazon WorkDocs	DeleteAnnotation	Chiamata API
Amazon WorkDocs	DeleteComment	Chiamata API
Amazon WorkDocs	DeleteDocument	Chiamata API
Amazon WorkDocs	DeleteFeedbackRequest	Chiamata API
Amazon WorkDocs	DeleteFolder	Chiamata API
Amazon WorkDocs	DeleteFolderContents	Chiamata API
Amazon WorkDocs	DeleteGroup	Chiamata API
Amazon WorkDocs	DeleteOrganizationShare	Chiamata API
Amazon WorkDocs	DeleteUser	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon WorkDocs	DownloadDocumentVersion	Chiamata API
Amazon WorkDocs	DownloadDocumentVersionUnderlays	Chiamata API
Amazon WorkDocs	InitiateDocumentVersionUpload	Chiamata API
Amazon WorkDocs	LogoutUser	Chiamata API
Amazon WorkDocs	PaginatedOrganizationActivity	Chiamata API
Amazon WorkDocs	PublishAnnotations	Chiamata API
Amazon WorkDocs	PublishComments	Chiamata API
Amazon WorkDocs	RestoreDocument	Chiamata API
Amazon WorkDocs	RestoreFolder	Chiamata API
Amazon WorkDocs	SearchGroups	Chiamata API
Amazon WorkDocs	SearchOrganizationUsers	Chiamata API
Amazon WorkDocs	TransferUserResources	Chiamata API
Amazon WorkDocs	UpdateAnnotation	Chiamata API
Amazon WorkDocs	UpdateComment	Chiamata API
Amazon WorkDocs	UpdateDocument	Chiamata API
Amazon WorkDocs	UpdateDocumentVersion	Chiamata API
Amazon WorkDocs	UpdateFolder	Chiamata API
Amazon WorkDocs	UpdateGroup	Chiamata API
Amazon WorkDocs	UpdateOrganization	Chiamata API

Servizio	Nome evento	Tipo di evento
Amazon WorkDocs	UpdateUser	Chiamata API
Amazon WorkMail	AssumeImpersonationRole	Chiamata API
Amazon WorkMail	QueryDnsRecords	Chiamata API
Amazon WorkMail	SearchMembers	Chiamata API
Amazon WorkMail	TestAvailabilityConfiguration	Chiamata API
Amazon WorkMail	TestInboundMailFlowRules	Chiamata API
Amazon WorkMail	TestOutboundMailFlowRules	Chiamata API

EventBridge riferimento ai dettagli degli eventi

EventBridge emette di per sé i seguenti eventi. Questi eventi vengono inviati automaticamente al bus degli eventi predefinito come con qualsiasi altro AWS servizio.

Per le definizioni dei campi di metadati inclusi in tutti gli eventi, vederethe section called “Riferimento per la struttura degli eventi”.

Argomenti

- Evento programmato
- Schema creato
- Versione dello schema creata

Evento programmato

Di seguito sono riportati i campi relativi ai dettagli dell'`Scheduled Event`evento.

I `detail-type` campi `source` e sono inclusi perché contengono valori specifici per EventBridge gli eventi. Per le definizioni degli altri campi di metadati inclusi in tutti gli eventi, vederethe section called “Riferimento per la struttura degli eventi”.

```
{
  . . .,
```

```
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  . . .,
  "detail": {}
}
```

`detail-type`

Identifica il tipo di evento.

Per questo evento, questo valore è`Scheduled Event`.

Campo obbligatorio: sì

`source`

Identifica il servizio che ha generato l'evento. Per EventBridge gli eventi, questo valore è`aws.events`.

Campo obbligatorio: sì

`detail`

Un oggetto JSON contenente informazioni sull'evento. Il servizio che genera l'evento determina il contenuto di questo campo.

Campo obbligatorio: sì

Non ci sono campi obbligatori in questo oggetto per `Scheduled Event` gli eventi.

Example Esempio di evento programmato

```
{
  "version": "0",
  "id": "89d1a02d-5ec7-412e-82f5-13505f849b41",
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  "account": "123456789012",
  "time": "2016-12-30T18:44:49Z",
  "region": "us-east-1",
  "resources": ["arn:aws:events:us-east-1:123456789012:rule/SampleRule"],
  "detail": {}
}
```

Schema creato

Di seguito sono riportati i campi di dettaglio dell'`Schema Created`evento.

Quando viene creato uno schema, EventBridge invia `Schema Created` sia un evento che un `Schema Version Created` evento.

I `detail-type` campi `source` e sono inclusi perché contengono valori specifici per EventBridge gli eventi. Per le definizioni degli altri campi di metadati inclusi in tutti gli eventi, vederethe section called "Riferimento per la struttura degli eventi".

```
{
  . . .,
  "detail-type": "Schema Created",
  "source": "aws.schemas",
  . . .,
  "detail": {
    "SchemaName" : "String",
    "SchemaType" : "String",
    "RegistryName" : "String",
    "CreationDate" : "DateTime",
    "Version" : "Number"
  }
}
```

`detail-type`

Identifica il tipo di evento.

Per questo evento, questo valore è`Schema Created`.

Campo obbligatorio: sì

`source`

Identifica il servizio che ha generato l'evento. Per EventBridge gli eventi, questo valore è`aws.schemas`.

Campo obbligatorio: sì

`detail`

Un oggetto JSON contenente informazioni sull'evento. Il servizio che genera l'evento determina il contenuto di questo campo.

Campo obbligatorio: sì

Per questo evento, questi dati includono:

`SchemaName`

Il nome dello schema.

Campo obbligatorio: sì

`SchemaType`

Il tipo di schema.

Valori validi: `OpenApi3` | `JSONSchemaDraft4`

Campo obbligatorio: sì

`RegistryName`

Il nome del registro che contiene lo schema.

Campo obbligatorio: sì

`CreationDate`

La data di creazione dello schema.

Campo obbligatorio: sì

`Version`

La versione dello schema.

Per `Schema Created` gli eventi, questo valore sarà sempre`1`.

Campo obbligatorio: sì

Example Esempio: evento Schema Created

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Schema Created",
  "source": "aws.schemas",
  "account": "123456789012",
```

```
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "resources": ["arn:aws:schemas:us-east-1::schema/myRegistry/mySchema"],
  "detail": {
    "SchemaName": "mySchema",
    "SchemaType": "OpenApi3",
    "RegistryName": "myRegistry",
    "CreationDate": "2019-11-29T20:08:55Z",
    "Version": "1"
  }
}
```

Versione dello schema creata

Di seguito sono riportati i campi di dettaglio dell'`Schema Version Created`evento.

Quando viene creato uno schema, EventBridge invia `Schema Created` sia un evento che un `Schema Version Created` evento.

I `detail-type` campi `source` e sono inclusi perché contengono valori specifici per EventBridge gli eventi. Per le definizioni degli altri campi di metadati inclusi in tutti gli eventi, vederethe section called “Riferimento per la struttura degli eventi”.

```
{
  . . .,
  "detail-type": "Schema Version Created",
  "source": "aws.schemas",
  . . .,
  "detail": {
    "SchemaName" : "String",
    "SchemaType" : "String",
    "RegistryName" : "String",
    "CreationDate" : "DateTime",
    "Version" : "Number"
  }
}
```

`detail-type`

Identifica il tipo di evento.

Per questo evento, questo valore è`Schema Version Created`.

Campo obbligatorio: sì

`source`

Identifica il servizio che ha generato l'evento. Per EventBridge gli eventi, questo valore è`aws.schemas`.

Campo obbligatorio: sì

`detail`

Un oggetto JSON contenente informazioni sull'evento. Il servizio che genera l'evento determina il contenuto di questo campo.

Campo obbligatorio: sì

Per questo evento, questi dati includono:

`SchemaName`

Il nome dello schema.

Campo obbligatorio: sì

`SchemaType`

Il tipo di schema.

Valori validi: `OpenApi3` | `JSONSchemaDraft4`

Campo obbligatorio: sì

`RegistryName`

Il nome del registro che contiene lo schema.

Campo obbligatorio: sì

`CreationDate`

La data di creazione della versione dello schema.

Campo obbligatorio: sì

`Version`

La versione dello schema.

Campo obbligatorio: sì

Example Esempio di evento Schema Version Created

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Schema Version Created",
  "source": "aws.schemas",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "resources": ["arn:aws:schemas:us-east-1::schema/myRegistry/mySchema"],
  "detail": {
    "SchemaName": "mySchema",
    "SchemaType": "OpenApi3",
    "RegistryName": "myRegistry",
    "CreationDate": "2019-11-29T20:08:55Z",
    "Version": "5"
  }
}
```

Ricezione di eventi da un partner SaaS con Amazon EventBridge

Per poter ricevere eventi da applicazioni e servizi partner SaaS, devi disporre di un'origine eventi partner del partner. Puoi quindi creare un router di eventi partner e associarlo all'origine eventi partner corrispondente.

Il video seguente illustra le integrazioni SaaS con EventBridge: partner Software as a service (SaaS)

Argomenti

- Integrazioni di partner SaaS supportate
- Configurazione di Amazon EventBridge per ricevere eventi da un'integrazione SaaS
- Creazione di una regola che corrisponde a eventi di partner SaaS
- Ricezione di eventi utilizzando gli URL delle funzioni AWS Lambda
- Ricezione di eventi da Salesforce

Integrazioni di partner SaaS supportate

EventBridge supporta le seguenti integrazioni di partner SaaS:

- Adobe
- Auth0
- Blitline
- BUIDLHub
- Buildkite
- CleverTap
- Datadog
- Epsagon
- Freshworks
- Genesys
- GS2
- Karte
- Kloudless
- Mackerel
- MongoDB
- New Relic
- OneLogin
- Opsgenie
- PagerDuty
- Payshield
- SaaSus Platform
- SailPoint
- Saviynt
- Segment
- Shopify
- SignalFx
- Site24x7
- Stax
- Stripe
- SugarCRM

- SugarCRM
- Symantec
- Thundra
- TriggerMesh
- Whispir
- Zendesk
- API partner rivenditori Amazon

Le origini eventi partner sono disponibili nelle seguenti Regioni.

Codice	Nome
us-east-1	Stati Uniti orientali (Virginia settentrionale)
us-east-2	Stati Uniti orientali (Ohio)
us-west-1	Stati Uniti occidentali (California settentrionale)
us-west-2	US West (Oregon)
ca-central-1	Canada (Centrale)
eu-central-1	Europa (Francoforte)
eu-central-2	Europa (Zurigo)
eu-west-1	Europa (Irlanda)
eu-west-2	Europe (London)
eu-west-3	Europe (Paris)
eu-north-1	Europe (Stockholm)
eu-south-1	Europa (Milano)
eu-south-2	Europa (Spagna)
af-south-1	Africa (Città del Capo)

Codice	Nome
ap-south-1	Asia Pacifico (Mumbai)
ap-south-2	Asia Pacific (Hyderabad)
ap-east-1	Asia Pacifico (Hong Kong)
ap-northeast-1	Asia Pacifico (Tokyo)
ap-northeast-2	Asia Pacifico (Seoul)
ap-northeast-3	Asia Pacifico (Osaka-Locale)
ap-southeast-1	Asia Pacifico (Singapore)
ap-southeast-2	Asia Pacifico (Sydney)
ap-southeast-3	Asia Pacifico (Giacarta)
ap-southeast-4	Asia Pacifico (Melbourne)
cn-north-1	Cina (Pechino)
cn-northwest-1	Cina (Ningxia)
me-central-1	Medio Oriente (Emirati Arabi Uniti)
me-south-1	Medio Oriente (Bahrein)
sa-east-1	Sud America (San Paolo)
il-central-1	Israele (Tel Aviv)

Configurazione di Amazon EventBridge per ricevere eventi da un'integrazione SaaS

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Partner event sources (Origini eventi partner).

3. Individua il partner desiderato e scegli Configura per tale partner.
4. Scegli Copia per copiare l'ID account negli appunti.
5. Nel riquadro di navigazione, scegliere Partner event sources (Origini eventi partner).
6. Vai al sito Web del partner e segui le istruzioni per creare un'origine eventi partner utilizzando l'ID del tuo account. L'origine eventi creata è disponibile solo per il tuo account.
7. Torna alla EventBridge console e scegli Partner event sources nel riquadro di navigazione.
8. Seleziona il pulsante accanto all'origine eventi partner e scegli Associa con bus di eventi.

   Lo stato dell'origine eventi cambia da `Pending` a `Active` e il nome del router di eventi viene aggiornato in modo che corrisponda al nome dell'origine eventi partner. Ora puoi iniziare a creare regole che corrispondono a eventi provenienti dall'origine eventi partner. Per ulteriori informazioni, consulta Creazione di una regola che corrisponde a eventi di partner SaaS.

   > **Note**
   >
   > Tutti gli eventi pubblicati da un partner su una fonte di eventi partner che non è stata associata a un router di eventi verranno immediatamente eliminati. Questi eventi non verranno mantenuti inalterati EventBridge.

Creazione di una regola che corrisponde a eventi di partner SaaS

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola.

   Una regola non può avere lo stesso nome di un'altra regola nella stessa regione e sullo stesso router di eventi.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se la regola deve cercare eventi corrispondenti provenienti dal tuo account, seleziona Bus di eventi predefiniti di AWS . Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).

7. Seleziona Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. (Facoltativo) In Eventi di esempio, scegli il tipo di evento.
10. In Modello di eventi, immetti un modello di eventi JSON.
11. Seleziona Successivo.
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. Per Seleziona una destinazione, scegli il AWS servizio a cui desideri inviare informazioni quando EventBridge rileva un evento che corrisponde allo schema dell'evento.
14. I campi visualizzati variano a seconda del servizio scelto. Se necessario, inserisci le informazioni specifiche per questo tipo di destinazione.
15. Per molti tipi di target, EventBridge sono necessarie le autorizzazioni per inviare eventi alla destinazione. In questi casi, EventBridge può creare il ruolo IAM necessario per l'esecuzione della regola. Esegui una di queste operazioni:

    - Per creare un ruolo IAM automaticamente, seleziona Crea un nuovo ruolo per questa risorsa specifica.
    - Per utilizzare un ruolo IAM creato in precedenza, seleziona Utilizza un ruolo esistente e seleziona il ruolo esistente dal menu a discesa.

16. (Facoltativo) Per Additional settings (Impostazioni aggiuntive), procedi come segue:

    a. Per Maximum age of event (Età massima dell'evento), immetti un valore compreso tra un minuto (00:01) e 24 ore (24:00).

    b. Per Tentativi, specifica un numero compreso tra 0 e 185.

    c. Per la coda di lettere non scritte, scegli se utilizzare una coda Amazon SQS standard come coda di lettere non scritte. EventBridge invia gli eventi che corrispondono a questa regola alla coda di lettere non scritte se non vengono consegnati correttamente alla destinazione. Esegui una di queste operazioni:

        - Scegli Nessuna per non utilizzare una coda DLQ.
        - Scegli Seleziona una coda Amazon SQS nell'account AWS corrente da utilizzare come coda DLQ, quindi seleziona la coda da utilizzare dal menu a discesa.
        - Scegli Seleziona una coda Amazon SQS in un altro AWS account come coda di lettere non scritte, quindi inserisci l'ARN della coda da utilizzare. È necessario allegare una policy basata sulle risorse alla coda che conceda l'autorizzazione a inviarle messaggi.

EventBridge Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.

17. (Facoltativo) Scegli Aggiungi destinazione per aggiungere un'altra destinazione per questa regola.
18. Seleziona Successivo.
19. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta la pagina EventBridge Etichette Amazon.
20. Seleziona Next (Successivo).
21. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Ricezione di eventi utilizzando gli URL delle funzioni AWS Lambda

> **Note**
>
> Affinché l'Inbound Webhook sia accessibile ai nostri partner, stiamo creando un Open Lambda nel tuo AWS account che è protetto a livello di applicazione Lambda verificando la firma di autenticazione inviata dal partner terzo. Esamina questa configurazione con il tuo team di sicurezza. Per ulteriori informazioni, consulta Modello di sicurezza e autenticazione per gli URL della funzione Lambda.

Il tuo bus di EventBridge eventi Amazon può utilizzare un URL di AWS Lambda funzione creato da un AWS CloudFormation modello per ricevere eventi dai provider SaaS supportati. Con gli URL delle funzioni, i dati dell'evento vengono inviati a una funzione Lambda. La funzione converte quindi questi dati in un evento che può essere acquisito EventBridge e inviato a un bus di eventi per l'elaborazione. Una volta che l'evento è in un router di eventi, è possibile utilizzare le regole per filtrare gli eventi, applicare eventuali trasformazioni di input configurate e quindi instradarlo alla destinazione corretta.

> **Note**
>
> La creazione di URL di funzioni Lambda aumenterà i costi mensili. Per ulteriori informazioni, consultare Prezzi di AWS Lambda.

Per configurare una connessione EventBridge, devi prima selezionare il provider SaaS con cui desideri configurare una connessione. Quindi, fornisci un segreto di firma che hai creato con quel provider e seleziona il bus degli EventBridge eventi a cui inviare gli eventi. Infine, usi un AWS CloudFormation modello e crei le risorse necessarie per completare la connessione.

Attualmente è possibile utilizzare i seguenti provider SaaS con l'utilizzo degli URL della funzione EventBridge Lambda:

- GitHub
- Twilio

Argomenti

- Configurazione di una connessione a GitHub

- Fase 1: Creare lo stack AWS CloudFormation
- Passaggio 2: creare un webhook GitHub
- Configurazione di una connessione a Twilio
- Aggiornamento del segreto o del token di autenticazione del webhook
- Aggiornamento della funzione Lambda
- Tipi di eventi disponibili
- Quote, codici di errore e nuovi tentativi di distribuzione

Configurazione di una connessione a GitHub

Fase 1: Creare lo stack AWS CloudFormation

Innanzitutto, usa la EventBridge console Amazon per creare uno CloudFormation stack:

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione scegli Avviamenti rapidi.
3. In Webhook in entrata tramite gli URL delle funzioni Lambda, scegli Inizia.
4. In GitHub, scegli Configura.
5. In Passaggio 1: selezionare un router di eventi, seleziona un router di eventi dall'elenco a discesa. Questo router di eventi riceve i dati dall'URL della funzione Lambda fornito a GitHub. Puoi anche creare un router di eventi selezionando Nuovo bus di eventi.
6. Nella fase 2: Configurazione tramite CloudFormation, scegli Nuovo GitHub webhook.
7. Seleziona Riconosco che il webhook in entrata che creo sarà accessibile pubblicamente. e scegli Conferma.
8. Immettere un nome per lo stack.
9. In Parametri, verifica che sia elencato il router di eventi corretto, quindi specifica un token sicuro per GitHubWebhookSecret. Per ulteriori informazioni sulla creazione di un token sicuro, consulta Setting your secret token nella documentazione GitHub.
10. In Funzionalità e trasformazioni, seleziona le seguenti opzioni:

    - Riconosco che ciò AWS CloudFormation potrebbe creare risorse IAM.
    - Riconosco che AWS CloudFormation potrebbe creare risorse IAM con nomi personalizzati.
    - Riconosco che AWS CloudFormation potrebbe richiedere la seguente funzionalità: **`CAPABILITY_AUTO_EXPAND`**

11. Seleziona Crea stack.

Passaggio 2: creare un webhook GitHub

A questo punto, devi creare il webhook in GitHub. Per completare questo passaggio sono necessari sia il token sicuro che l'URL della funzione Lambda creato nel passaggio 2. Per ulteriori informazioni, consulta Creating webhooks nella documentazione GitHub.

Configurazione di una connessione a Twilio

Passaggio 1: trovare il token di autenticazione Twilio

Per configurare una connessione tra Twilio e EventBridge, configura prima la connessione Twilio con il token di autenticazione, o segreto, per il tuo Twilio account. Per ulteriori informazioni, consulta Auth Tokens e How To Change Them nella documentazione Twilio.

Passaggio 2: crea lo stack AWS CloudFormation

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Avviamenti rapidi.
3. In Webhook in entrata tramite gli URL delle funzioni Lambda, scegli Inizia.
4. In Twilio, scegli Configura.
5. In Passaggio 1: selezionare un router di eventi, seleziona un router di eventi dall'elenco a discesa. Questo router di eventi riceve i dati dall'URL della funzione Lambda fornito a Twilio. Puoi anche creare un router di eventi selezionando Nuovo bus di eventi.
6. Nella fase 2: Configurazione tramite CloudFormation, scegli Nuovo Twilio webhook.
7. Seleziona Riconosco che il webhook in entrata che creo sarà accessibile pubblicamente. e scegli Conferma.
8. Immettere un nome per lo stack.
9. In Parametri, verifica che sia elencato il router di eventi corretto, quindi immetti TwilioWebhookSecret creato in Passaggio 1.
10. In Funzionalità e trasformazioni, seleziona le seguenti opzioni:
    - Riconosco che ciò AWS CloudFormation potrebbe creare risorse IAM.
    - Riconosco che AWS CloudFormation potrebbe creare risorse IAM con nomi personalizzati.

- Riconosco che AWS CloudFormation potrebbe richiedere la seguente funzionalità: CAPABILITY_AUTO_EXPAND

11. Seleziona Crea stack.

Passaggio 3: creare un webhook Twilio

Dopo aver impostato l'URL della funzione Lambda, devi fornirlo a Twilio in modo che i dati dell'evento possano essere inviati. Per ulteriori informazioni, consulta Configure your public URL with Twilio nella documentazione Twilio.

Aggiornamento del segreto o del token di autenticazione del webhook

Aggiornamento del segreto GitHub

> **Note**
>
> GitHub non supporta due segreti nello stesso momento. È possibile che si verifichino tempi di inattività delle risorse quando il GitHub segreto e il segreto nello stack non sono sincronizzati. AWS CloudFormation GitHubi messaggi inviati mentre i segreti non sono sincronizzati falliranno a causa di firme errate. Attendi che i CloudFormation segreti GitHub e i segreti siano sincronizzati, quindi riprova.

1. Crea un nuovo segreto GitHub. Per ulteriori informazioni, consulta Encryptes secrets nella documentazione GitHub.
2. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
3. Scegli Stack nel riquadro di navigazione.
4. Scegli lo stack per il webhook che include il segreto da aggiornare.
5. Scegli Aggiorna.
6. Assicurati che l'opzione Utilizza modello corrente sia selezionata e scegli Successivo.
7. In GitHubWebhookSecret, deseleziona Usa il valore esistente, inserisci il nuovo GitHub segreto creato nel passaggio 1 e scegli Avanti.
8. Seleziona Successivo.
9. Scegli Aggiorna stack.

La propagazione del segreto può richiedere fino a un'ora. Per ridurre questo periodo di inattività, puoi aggiornare il contesto di esecuzione Lambda.

Aggiornamento del segreto Twilio

> **Note**
>
> Twilio non supporta due segreti nello stesso momento. È possibile che si verifichino tempi di inattività delle risorse quando il Twilio segreto e il segreto nello AWS CloudFormation stack non sono sincronizzati. Twilioi messaggi inviati mentre i segreti non sono sincronizzati falliranno a causa di firme errate. Attendi che CloudFormation i segreti Twilio e i segreti siano sincronizzati, quindi riprova.

1. Crea un nuovo segreto Twilio. Per ulteriori informazioni, consulta Auth Tokens e How To Change Them nella documentazione Twilio.
2. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
3. Scegli Stack nel riquadro di navigazione.
4. Scegli lo stack per il webhook che include il segreto da aggiornare.
5. Scegli Aggiorna.
6. Assicurati che l'opzione Utilizza modello corrente sia selezionata e scegli Successivo.
7. In TwilioWebhookSecret, deseleziona Usa il valore esistente, inserisci il nuovo Twilio segreto creato nel passaggio 1 e scegli Avanti.
8. Seleziona Successivo.
9. Scegli Aggiorna stack.

La propagazione del segreto può richiedere fino a un'ora. Per ridurre questo periodo di inattività, puoi aggiornare il contesto di esecuzione Lambda.

Aggiornamento della funzione Lambda

La funzione Lambda creata dallo CloudFormation stack crea il webhook di base. Se desideri personalizzare la funzione Lambda per un caso d'uso specifico, come la registrazione personalizzata, usa la console per accedere alla funzione e poi usa la CloudFormation console Lambda per aggiornare il codice della funzione Lambda.

Aggiornamento della funzione Lambda

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudformation AWS CloudFormation .
2. Scegli Stack nel riquadro di navigazione.
3. Scegli lo stack del webhook che include la funzione Lambda da aggiornare.
4. Scegli la scheda Risorse.
5. Per aprire la funzione Lambda nella console Lambda, in ID fisico, scegli l'ID della funzione Lambda.

Ora che hai effettuato l'accesso alla funzione Lambda, utilizza la console Lambda per aggiornare il codice della funzione.

Aggiornamento della funzione Lambda

1. In Azioni, scegli Esporta funzione.
2. Scegli Scarica pacchetto di distribuzione e salva il file nel tuo computer.
3. Decomprimi il file .zip del pacchetto di implementazione, aggiorna il file `app.py` e comprimi il pacchetto di implementazione aggiornato, assicurandoti che siano inclusi tutti i file nel file .zip originale.
4. Nella console Lambda, scegli la scheda Codice.
5. In Code source (Origine codice), scegli Upload from (Carica da).
6. Scegli .zip file, quindi scegli Upload (Carica).
   - Nel selettore di file, seleziona il file aggiornato, scegli Apri, quindi scegli Salva.
7. In Azioni, scegli Pubblica nuova versione.

Tipi di eventi disponibili

I seguenti tipi di eventi sono attualmente supportati dai bus CloudFormation degli eventi:

- GitHub— Tutti i tipi di eventi sono supportati.
- Twilio: sono supportati webhook post-evento.

Quote, codici di errore e nuovi tentativi di distribuzione

Quote

Il numero di richieste in entrata al webhook è limitato dai servizi sottostanti. AWS La tabella seguente include le quote pertinenti.

Servizio	Quota
AWS Lambda	Impostazione predefinita: 10 esecuzioni simultanee Per ulteriori informazioni sulle quote, inclusa la richiesta di aumento delle stesse, consulta Quote di Lambda.
AWS Secrets Manager	Valore predefinito: 5.000 richieste al secondo Per ulteriori informazioni sulle quote, inclusa la richiesta di aumento delle stesse, consulta Quote di AWS Secrets Manager. **Note** Il numero di richieste al secondo viene ridotto al minimo utilizzando il client di caching Python di AWS Secrets Manager.
Amazon EventBridge	Dimensione massima di 256 KB di ingresso per PutEvents le azioni. EventBridge applica quote tariffarie basate sulla regione. Per ulteriori informazioni, consulta ???.

Codici di errore

Ogni AWS servizio restituisce codici di errore specifici quando si verificano errori. La tabella seguente include i codici di errore pertinenti.

Servizio	Codice di errore	Descrizione
AWS Lambda	429 «» TooManyRequestsException	La quota di esecuzioni simultanee è stata superata.
AWS Secrets Manager	500 "Errore interno del server"	La quota di richieste al secondo è stata superata.
Amazon EventBridge	500 "Errore interno del server"	La quota tariffaria è stata superata per la Regione.

Ridistribuzione degli eventi

In caso di errori, puoi riprovare a distribuire gli eventi interessati. Ogni provider SaaS ha procedure di ripetizione differenti.

GitHub

Utilizza l'API webhooks GitHub per verificare lo stato di distribuzione di ogni chiamata webhook e ridistribuire l'evento, se necessario. Per ulteriori informazioni, consulta la seguente documentazione GitHub:

- Organizzazione: Redeliver a delivery for an organization webhook
- Repository: Redeliver a delivery for a repository webhook
- App: Redeliver a delivery for an app webhook

Twilio

Gli utenti Twilio possono personalizzare le opzioni di ripetizione degli eventi utilizzando sostituzioni di connessioni. Per ulteriori informazioni, consulta Webhooks (HTTP callbacks): Connection Overrides nella documentazione Twilio.

Ricezione di eventi da Salesforce

Puoi usare Amazon EventBridge per ricevere eventi Salesforce nei seguenti modi:

- Utilizzando la funzione Salesforce's Event Bus Relay per ricevere eventi direttamente su un event bus EventBridge partner.
- Configurando un flusso in Amazon AppFlow che viene utilizzato Salesforce come fonte di dati. Amazon invia AppFlow quindi Salesforce gli eventi EventBridge utilizzando un bus di eventi partner.

Puoi inviare informazioni sugli eventi a Salesforce utilizzando destinazioni API. Una volta inviato a Salesforce, l'evento può essere elaborato da flussi o trigger Apex. Per ulteriori informazioni sulla configurazione di una destinazione API Salesforce, consulta ???.

Argomenti

- Ricezione di eventi da Salesforce mediante Event Bus Relay
- Ricezione di eventi Salesforce tramite Amazon AppFlow

Ricezione di eventi da Salesforce mediante Event Bus Relay

Passaggio 1: configura Salesforce Event Bus Relay e una fonte di eventi EventBridge partner

Quando crei una configurazione Event Relay suSalesforce, Salesforce crea una fonte di eventi partner nello stato EventBridge in sospeso.

Per configurare Event Bus Relay di Salesforce

1. Configura uno strumento REST API
2. (Facoltativo) Definisci un evento della piattaforma
3. Crea un canale per un evento della piattaforma personalizzato
4. Crea un membro del canale per associare l'evento della piattaforma personalizzato
5. Crea credenziali con nome
6. Crea una configurazione di inoltro di eventi

Passaggio 2: attiva l'origine degli eventi Salesforce partner nella EventBridge console e avvia l'inoltro dell'evento

1. Apri la pagina delle fonti degli eventi per i partner nella EventBridge console.
2. Seleziona l'origine di eventi partner Salesforce creata in Passaggio 1.
3. Scegli Associa con bus di eventi.
4. Convalida il nome del router di eventi partner.
5. Selezionare Associate (Associa).
6. Avvia l'inoltro degli eventi

Ora che hai impostato e avviato Event Bus Relay e configurato l'origine degli eventi partner, puoi creare una EventBridge regola che reagisce agli eventi per filtrare e inviare i dati a una destinazione.

Ricezione di eventi Salesforce tramite Amazon AppFlow

Amazon AppFlow incapsula gli eventi Salesforce in una busta di EventBridge eventi. L'esempio seguente mostra un Salesforce evento ricevuto da un bus di eventi EventBridge partner.


```
{
    "version": "0",
    "id": "5c42b99e-e005-43b3-c744-07990c50d2cc",
    "detail-type": "AccountChangeEvent",
    "source": "aws.partner/appflow.test/salesforce.com/364228160620/CustomSF-Source-
Final",
    "account": "000000000",
    "time": "2020-08-20T18:25:51Z",
    "region": "us-west-2",
    "resources": [],
    "detail": {
        "ChangeEventHeader": {
            "commitNumber": 248197218874,
            "commitUser": "0056g000003XW7AAAW",
            "sequenceNumber": 1,
            "entityName": "Account",
            "changeType": "UPDATE",
            "changedFields": [
                "LastModifiedDate",
                "Region__c"
            ],
            "changeOrigin": "com/salesforce/api/soap/49.0;client=SfdcInternalAPI/",
```

```
            "transactionKey": "000035af-b239-0581-9f14-461e4187de11",
            "commitTimestamp": 1597947935000,
            "recordIds": [
                "0016g00000MLhLeAAL"
            ]
        },
        "LastModifiedDate": "2020-08-20T18:25:35.000Z",
        "Region__c": "America"
    }
}
```

Fase 1: configura Amazon AppFlow per utilizzarlo Salesforce come fonte di eventi per i partner

Per inviare eventi a EventBridge, devi prima configurare Amazon AppFlow per utilizzarlo Salesforce come fonte di eventi partner.

1. Nella AppFlowconsole Amazon, scegli Create flow.
2. Nella sezione Dettagli flusso, in Nome flusso immetti un nome per il flusso.
3. (Facoltativo) Immetti un nome e una descrizione per il flusso, quindi scegli Successivo.
4. In Dettagli origine, scegli Salesforce dal menu a discesa Nome origine, quindi scegli Connetti per creare una nuova connessione.
5. Nella finestra di dialogo Connetti a Salesforce, scegli Produzione o Sandbox per l'ambiente Salesforce.
6. Nel campo Nome connessione, immetti un nome univoco per la connessione, quindi scegli Continua.
7. Nella finestra di dialogo Salesforce, procedi come segue:
    a. Immetti le credenziali di accesso Salesforce per accedere a Salesforce.
    b. Seleziona Salesforce gli eventi per i tipi di dati AppFlow da elaborare da Amazon.
8. Nel menu a discesa Scegli Salesforce evento, seleziona il tipo di evento a cui inviare. EventBridge
9. Per una destinazione, seleziona Amazon EventBridge.
10. Seleziona Crea una nuova origine di eventi partner.
11. (Facoltativo) Specifica un suffisso univoco per l'origine di eventi partner.
12. Scegli Genera origine di eventi partner.
13. Scegli un bucket Amazon S3 per archiviare file di payload di eventi di dimensioni superiori a 256 KB.

14. Nella sezione Trigger flusso, assicurati che sia selezionata l'opzione Esegui flusso con nuovo evento. Questa impostazione assicura che il flusso venga eseguito quando si verifica un nuovo evento Salesforce.
15. Seleziona Successivo.
16. Per la mappatura dei campi, seleziona Mappa direttamente tutti i campi. In alternativa, puoi selezionare i campi che ti interessano dall'elenco Nomi campi di origine.

    Per ulteriori informazioni sulla mappatura dei campi, consulta Mappatura di campi di dati.
17. Seleziona Successivo.
18. (Facoltativo) Configura i filtri per i campi di dati in Amazon AppFlow.
19. Seleziona Successivo.
20. Esamina le impostazioni e quindi scegli Crea flusso.

Con il flusso configurato, Amazon AppFlow crea una nuova fonte di eventi per i partner che devi quindi associare a un partner event bus nel tuo account.

Fase 2: Configurazione EventBridge per ricevere Salesforce eventi

Assicurati che il AppFlow flusso Amazon attivato dagli Salesforce eventi con EventBridge come destinazione sia configurato prima di seguire le istruzioni in questa sezione.

Per configurare la ricezione EventBridge di eventi Salesforce

1. Apri la pagina delle fonti degli eventi per i partner nella EventBridge console.
2. Seleziona l'origine di eventi partner Salesforce creata in Passaggio 1.
3. Scegli Associa con bus di eventi.
4. Convalida il nome del router di eventi partner.
5. Selezionare Associate (Associa).
6. Nella AppFlow console Amazon, apri il flusso che hai creato e scegli Attiva flusso.
7. Apri la pagina Regole nella EventBridge console.
8. Scegli Crea regola.
9. Immetti un nome univoco per il ruolo.
10. Nella sezione Definisci il modello, scegli Modello di eventi.
11. In Modello di corrispondenza degli eventi, seleziona Modello predefinito dal servizio.

12. Nella sezione Fornitore di servizi, seleziona Tutti gli eventi.
13. In Seleziona bus di eventi, scegli Bus di eventi personalizzato o dei partner.
14. Seleziona il bus di eventi che hai associato all'origine dell'evento AppFlow partner Amazon.
15. Per Select targets, scegli il AWS servizio che deve agire quando viene eseguita la regola. Una regola può avere fino a cinque destinazioni.
16. Scegli Crea.

Il servizio di destinazione riceve tutti gli eventi Salesforce configurati per il tuo account. Per filtrare gli eventi o inviare alcuni eventi a destinazioni diverse, puoi utilizzare il filtro basato su contenuto con modelli di eventi.

> **Note**
>
> Per eventi di dimensioni superiori a 256 KB, Amazon AppFlow non invia l'intero evento a EventBridge. Invece, Amazon AppFlow inserisce l'evento in un bucket S3 del tuo account, quindi invia un evento a EventBridge con un puntatore al bucket Amazon S3. Puoi utilizzare il puntatore per ottenere l'intero evento dal bucket.

Debug della distribuzione di eventi

I problemi di consegna degli eventi possono essere difficili da identificare e EventBridge offre alcuni modi per eseguire il debug e ripristinare gli errori di consegna degli eventi.

In che modo EventBridge riprova a fornire eventi

A volte un evento non viene distribuito correttamente alla destinazione specificata in una regola. Ciò può accadere, ad esempio:

- Se la risorsa di destinazione non è disponibile
- A causa delle condizioni della rete

Quando un evento non viene consegnato correttamente a una destinazione a causa di errori recuperabili, EventBridge riprova a inviare l'evento. A questo proposito, puoi impostare il periodo durante il quale effettua nuovi tentativi e il numero di tentativi nelle impostazioni Policy di ripetizione

della destinazione. Per impostazione predefinita, EventBridge riprova a inviare l'evento per 24 ore e fino a 185 volte con un backoff e un jitter esponenziali o un ritardo casuale.

Se un evento non viene consegnato dopo aver esaurito tutti i tentativi, l'evento viene eliminato e non viene più elaborato. EventBridge

Utilizzo di code di lettere morte per elaborare gli eventi non consegnati

Per evitare di perdere eventi dopo la mancata distribuzione a una destinazione, è possibile configurare una coda DLQ e inviarle tutti gli eventi non riusciti per un'elaborazione successiva.

EventBridge I DLQ sono code Amazon SQS standard che vengono utilizzate per archiviare eventi EventBridge che non è stato possibile consegnare correttamente a una destinazione. Quando crei una regola e aggiungi una destinazione, puoi scegliere se utilizzare o meno una coda DLQ. Quando configuri un coda DLQ, puoi mantenere tutti gli eventi che non sono stati distribuiti correttamente. È quindi possibile risolvere il problema che ha causato la mancata distribuzione dell'evento ed elaborare gli eventi in un secondo momento.

Quando configuri un DLQ per una destinazione di una regola, EventBridge invia gli eventi con chiamate non riuscite alla coda Amazon SQS selezionata.

Gli errori relativi agli eventi vengono gestiti in modi diversi. Alcuni eventi vengono abbandonati o inviati a una coda DLQ senza che venga effettuato alcun nuovo tentativo. Ad esempio, per gli errori derivanti dalla mancanza di autorizzazioni per una destinazione o da una risorsa di destinazione che non esiste più, tutti i nuovi tentativi falliscono fino a che non viene intrapresa un'azione per risolvere il problema alla base. Invece di riprovare, EventBridge invia questi eventi direttamente al DLQ, se disponibile.

Quando la consegna di un evento non riesce, EventBridge pubblica un evento su Amazon CloudWatch Metrics indicando che un obiettivo `invocation` non è riuscito. Se utilizzi un DLQ, vengono inviate metriche aggiuntive a, tra cui and. CloudWatch `InvocationsSentToDLQ` `InvocationsFailedToBeSentToDLQ`

È inoltre possibile specificare DLQ per i bus di eventi, se si utilizza per AWS KMS chiavi gestite dal cliente crittografare gli eventi inattivi. Per ulteriori informazioni, consulta ???.

Ogni messaggio nella tua coda DLQ includerà i seguenti attributi personalizzati:

- `RULE_ARN`
- `TARGET_ARN`

- ERROR_CODE

  Di seguito è riportato un esempio dei codici di errore che una coda DLQ può restituire:

  - CONNECTION_FAILURE
  - CROSS_ACCOUNT_INGESTION_FAILED
  - CROSS_REGION_INGESTION_FAILED
  - ERROR_FROM_TARGET
  - EVENTS_IN_BATCH_REQUEST_REJECTED
  - EVENTS_IN_BATCH_REQUEST_REJECTED
  - FAILED_TO_ASSUME_ROLE
  - INTERNAL_ERROR
  - INVALID_JSON
  - INVALID_PARAMETER
  - NO_PERMISSIONS
  - NO_RESOURCE
  - RESOURCE_ALREADY_EXISTS
  - RESOURCE_LIMIT_EXCEEDED
  - RESOURCE_MODIFICATION_COLLISION
  - SDK_CLIENT_ERROR
  - THIRD_ACCOUNT_HOP_DETECTED
  - THIRD_REGION_HOP_DETECTED
  - THROTTLING
  - TIMEOUT
  - TRANSIENT_ASSUME_ROLE
  - UNKNOWN
- ERROR_MESSAGE
- EXHAUSTED_RETRY_CONDITION

  Possono essere restituite le seguenti condizioni:

  - MaximumRetryAttempts
  - MaximumEventAgeInSeconds
- RETRY_ATTEMPTS

Il video seguente descrive le code DLQ: Using dead-letter queues (DLQs)

Argomenti

Considerazioni sull'utilizzo di una coda DLQ

Considerate quanto segue quando configurate un DLQ per. EventBridge

- Sono supportate solo le code standard. Non è possibile utilizzare una coda FIFO per un DLQ in. EventBridge
- EventBridge include i metadati degli eventi e gli attributi del messaggio, tra cui: il codice di errore, il messaggio di errore, la condizione Exhausted Retry, l'ARN della regola, i tentativi di tentativo e l'ARN di destinazione. È possibile utilizzare questi valori per identificare un evento e la causa dell'errore.
- Autorizzazioni per le code DLQ nello stesso account:
  - Se aggiungi un obiettivo a una regola utilizzando la console e scegli una coda Amazon SQS nello stesso account, alla coda viene allegata automaticamente una policy basata sulle risorse che concede EventBridge l'accesso alla coda.
  - Se utilizzi il `PutTargets` funzionamento dell' EventBridge API per aggiungere o aggiornare un obiettivo per una regola e scegli una coda Amazon SQS nello stesso account, devi concedere manualmente le autorizzazioni alla coda selezionata. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.
- Autorizzazioni per l'utilizzo delle code Amazon SQS da un altro account. AWS
  - Se crei una regola dalla console, non vengono visualizzate le code di altri account che puoi selezionare. È necessario fornire l'ARN della coda nell'altro account e quindi associare manualmente una policy basata su risorse per concedere l'autorizzazione alla coda. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.
  - Se crei una regola utilizzando l'API, devi associare manualmente una policy basata su risorse alle code SQS in un altro account utilizzato come coda DLQ. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.

- La coda Amazon SQS utilizzata deve trovarsi nella stessa Regione in cui crei la regola.

Concessione delle autorizzazioni per la coda DLQ

Per inviare correttamente gli eventi alla coda, EventBridge devi avere l'autorizzazione a farlo. Quando si specifica un DLQ utilizzando la EventBridge console, le autorizzazioni vengono aggiunte automaticamente. Questo include:

- Quando si configura un DLQ per un obiettivo di una regola.
- Quando configuri un DLQ per un bus di eventi in cui lo hai specificato, EventBridge usa un AWS KMS chiave gestita dal cliente per crittografare gli eventi inattivi.

  Per ulteriori informazioni, consulta ???.

Se specifichi un DLQ utilizzando l'API o utilizzi una coda che si trova in un AWS account diverso, devi creare manualmente una politica basata sulle risorse che conceda le autorizzazioni richieste e quindi collegarla alla coda.

Esempio di autorizzazioni Target per la coda a lettera morta

La seguente politica basata sulle risorse mostra come concedere le autorizzazioni necessarie per inviare messaggi di eventi EventBridge a una coda Amazon SQS. L'esempio di policy concede al EventBridge servizio le autorizzazioni per utilizzare l'`SendMessage`operazione per inviare messaggi a una coda denominata "DLQ». MyEvent La coda deve trovarsi nella regione us-west-2 nell'account 123456789012. AWS L'`Condition`istruzione consente solo le richieste che provengono da una regola denominata "MyTestRule" creata nella regione us-west-2 nell'account 123456789012. AWS

```
{
  "Sid": "Dead-letter queue permissions",
  "Effect": "Allow",
  "Principal": {
     "Service": "events.amazonaws.com"
  },
  "Action": "sqs:SendMessage",
  "Resource": "arn:aws:sqs:us-west-2:123456789012:MyEventDLQ",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:events:us-west-2:123456789012:rule/MyTestRule"
    }
  }
```

```
}
```

Esempio di autorizzazioni per la coda a lettere morte di Event Bus

La seguente politica basata sulle risorse dimostra come concedere le autorizzazioni richieste quando si specifica un DLQ per un bus di eventi. In questo caso, `aws:SourceArn` specifica l'ARN del bus degli eventi che invia gli eventi al DLQ. Anche in questo esempio, la coda deve trovarsi nella stessa regione del bus degli eventi.

```
{
  "Sid": "Dead-letter queue permissions",
  "Effect": "Allow",
  "Principal": {
     "Service": "events.amazonaws.com"
  },
  "Action": "sqs:SendMessage",
  "Resource": "arn:aws:sqs:region:account-id:queue-name",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-arn"
    }
  }
}
```

Per associare la policy alla coda, usa la console Amazon SQS, apri la coda, quindi scegli la policy di accesso e modifica la policy. Puoi anche utilizzare l' AWS CLI. Per ulteriori informazioni, consulta Autorizzazioni di Amazon SQS.

Come inviare nuovamente eventi da una coda DLQ

Puoi spostare i messaggi da una coda DQL in due modi:

- Evita di scrivere logica consumer di Amazon SQS: imposta la coda DQL come origine di evento sulla funzione Lambda per svuotare la coda DLQ.
- Scrivi la logica consumer di Amazon SQS: utilizza l'API Amazon SQS AWS , l'SDK AWS CLI o per scrivere una logica consumer personalizzata per il polling, l'elaborazione e l'eliminazione dei messaggi nel DLQ.

Modelli di EventBridge eventi Amazon

I modelli di eventi hanno la stessa struttura degli eventi a cui corrispondono. Le regole utilizzano modelli di eventi per selezionare eventi e inviarli alle destinazioni. Un modello di eventi può corrispondere o meno a un evento.

> **Important**
>
> Nel EventBridge, è possibile creare regole che possono portare ad higher-than-expected addebiti e limitazioni. Ad esempio, puoi creare inavvertitamente una regola che genera un ciclo infinito, in cui una regola viene attivata in modo ricorsivo senza fine. Ad esempio, hai creato una regola per rilevare eventuali modifiche alle liste di controllo degli accessi (ACL) in un bucket S3 e attivare un programma software che le imposti sullo stato desiderato. Se la regola non è scritta con attenzione, la successiva modifica alle ACL la riattiva, creando un loop infinito.
>
> Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta ??? e ???.

Il video seguente fornisce informazioni di base sui modelli di eventi: How to filter events

Argomenti

- Creazione di modelli di eventi
- Esempi di eventi e modelli di eventi
- Corrispondenza di valori nulli e stringhe vuote nei modelli di eventi di Amazon EventBridge
- Array nei modelli di EventBridge eventi di Amazon
- Filtraggio dei contenuti nei modelli di EventBridge eventi di Amazon
- Test di un pattern di eventi utilizzando la Sandbox EventBridge
- Le migliori pratiche per la definizione dei modelli di EventBridge eventi Amazon

L'evento seguente mostra un semplice AWS evento di Amazon EC2.

```
{
  "version": "0",
```

```
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/i-1234567890abcdef0"
  ],
  "detail": {
    "instance-id": "i-1234567890abcdef0",
    "state": "terminated"
  }
}
```

Il seguente modello di eventi elabora tutti gli eventi `instance-termination` di Amazon EC2.

```
{
  "source": ["aws.ec2"],
  "detail-type": ["EC2 Instance State-change Notification"],
  "detail": {
    "state": ["terminated"]
  }
}
```

Creazione di modelli di eventi

Per creare un modello di eventi, specifichi i campi di un evento a cui deve corrispondere il modello. Specifica solo i campi che utilizzi per la corrispondenza. Il precedente esempio di modello di evento fornisce solo valori per tre campi: i campi di primo livello `"source"` e e `"detail-type"` il `"state"` campo all'interno del campo `"detail"` oggetto. EventBridge ignora tutti gli altri campi dell'evento quando applica la regola.

Affinché un modello di eventi corrisponda a un evento, l'evento deve contenere tutti i nomi di campo elencati nel modello di eventi. I nomi di campo devono essere visualizzati nell'evento con la stessa struttura di nidificazione.

Quando si scrivono modelli di eventi per la corrispondenza con gli eventi, è possibile utilizzare l'API `TestEventPattern` o il comando CLI `test-event-pattern` per verificare che il modello corrisponda agli eventi corretti. Per ulteriori informazioni, vedere TestEventPattern.

Valori di eventi corrispondenti

In un modello di eventi, il valore per la corrispondenza si trova in un array JSON, racchiuso tra parentesi quadre ("[", "]") in modo da poter fornire più valori. Ad esempio, per abbinare gli eventi di Amazon EC2 oppure AWS Fargate, puoi utilizzare lo schema seguente, che corrisponde agli eventi in cui il valore del `"source"` campo è o`"aws.ec2"`. `"aws.fargate"`

```
{
    "source": ["aws.ec2", "aws.fargate"]
}
```

Considerazioni sulla creazione di modelli di eventi

Di seguito sono riportati alcuni aspetti da considerare nella creazione dei modelli di eventi:

- EventBridge ignora i campi dell'evento che non sono inclusi nel modello di evento. L'effetto è che esiste un carattere jolly `"*": "*"` per i campi che non compaiono nel modello di eventi.
- I valori che corrispondono ai modelli di eventi seguono le regole JSON. È possibile includere stringhe racchiuse tra virgolette ("), numeri e parole chiave `true`, `false`, e `null`.
- Per le stringhe, EventBridge utilizza la character-by-character corrispondenza esatta senza ripiegamento tra maiuscole e minuscole o qualsiasi altra normalizzazione delle stringhe.
- Per i numeri, utilizza la rappresentazione in formato stringa EventBridge . Ad esempio, 300, 300.0 e 3.0e2 non sono considerati uguali.
- Se vengono specificati più modelli per lo stesso campo JSON, utilizza EventBridge solo l'ultimo.
- Tieni presente che quando EventBridge compila i modelli di eventi da utilizzare, usa dot (.) come carattere di unione.

  Ciò significa che EventBridge tratterà i seguenti modelli di eventi come identici:

  ```
  ## has no dots in keys
  { "detail" : { "state": { "status": [ "running" ] } } }

  ## has dots in keys
  { "detail" : { "state.status": [ "running" ] } }
  ```

  Entrambi i modelli di eventi corrisponderanno quindi ai due eventi seguenti:

  ```
  ## has no dots in keys
  ```

```
{ "detail" : { "state": { "status": "running" } } }

## has dots in keys
{ "detail" : { "state.status": "running"  } }
```

> **Note**
>
> Questo descrive EventBridge il comportamento attuale e non dovrebbe essere considerato tale da non cambiare.

- I modelli di eventi contenenti campi duplicati non sono validi. Se un modello contiene campi duplicati, considera EventBridge solo il valore finale del campo.

  Ad esempio, i seguenti modelli di eventi corrisponderanno allo stesso evento:

  ```
  ## has duplicate keys
  {
    "source": ["aws.s3"],
    "source": ["aws.sns"],
    "detail-type": ["AWS API Call via CloudTrail"],
    "detail":  {
        "eventSource": ["s3.amazonaws.com"],
        "eventSource": ["sns.amazonaws.com"]
    }
  }

  ## has unique keys
  {
    "source": ["aws.sns"],
    "detail-type": ["AWS API Call via CloudTrail"],
    "detail": { "eventSource": ["sns.amazonaws.com"] }
  }
  ```

  E EventBridge tratta i due eventi seguenti come identici:

  ```
  ## has duplicate keys
  {
    "source": ["aws.s3"],
    "source": ["aws.sns"],
    "detail-type": ["AWS API Call via CloudTrail"],
    "detail":  [
  ```

```
    {
      "eventSource": ["s3.amazonaws.com"],
      "eventSource": ["sns.amazonaws.com"]
    }
  ]
}

## has unique keys
{
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": [
    { "eventSource": ["sns.amazonaws.com"] }
  ]
}
```

> **Note**
>
> Questo descrive EventBridge il comportamento attuale e non dovrebbe essere considerato tale da non cambiare.

Operazioni di confronto da utilizzare in modelli di eventi

Di seguito un riepilogo di tutti gli operatori di confronto disponibili in EventBridge.

Gli operatori di confronto funzionano solo su nodi foglia, a eccezione di `$or` e `anything-but`.

Confronto	Esempio	Sintassi delle regole
And	La posizione è "New York" e il giorno è "lunedì"	`"Location": [ "New York" ], "Day": ["Monday"]`
Tutto tranne	Lo stato è qualsiasi valore oltre a «inizializzazione».	`"state": [ { "anything-but": "initializing" } ]`
Qualsiasi cosa tranne (inizia con)	La regione non è negli Stati Uniti.	`"Region": [ { "anything-but": {"prefix": "us-" } } ]`

Confronto	Esempio	Sintassi delle regole
Tutto tranne (finisce con)	FileName non termina con un'estensione.png.	`"FileName": [ { "anything-but": { "suffix": ".png" } } ]`
Tutto tranne (ignora maiuscole e minuscole)	Lo stato è qualsiasi valore oltre a «inizializzazione» o qualsiasi altra variazione di maiuscolo/minuscolo, come «INITIALIZING».	`"state": : [{ "anything-but": { "equals-ignore-case": "initializing" }}]}`
Qualsiasi cosa, tranne usare un jolly	FileName non è un percorso di file che include. `/lib/`	`"FilePath" : [{ "anything-but": { "wildcard": "*/lib/*" }}]`
Begins with	La regione è negli Stati Uniti.	`"Region": [ {"prefix": "us-" } ]`
Inizia con (ignora maiuscole e minuscole)	Il nome del servizio inizia con le lettere «eventb», indipendentemente da maiuscole e minuscole.	`{"service" : [{ "prefix": { "equals-ignore-case": "eventb" }}]}`
Empty	LastName è vuoto.	`"LastName": [""]`
Equals	Il nome è "Alice"	`"Name": [ "Alice" ]`
Equals (ignora maiuscole e minuscole)	Il nome è "Alice"	`"Name": [ { "equals-ignore-case": "alice" } ]`
Ends with	FileName termina con un'estensione.png	`"FileName": [ { "suffix": ".png" } ]`
Termina con (ignora maiuscole)	Il nome del servizio termina con le lettere «tbridge» o con qualsiasi altra variante di maiuscola, ad esempio «TBRIDGE».	`{"service" : [{ "suffix": { "equals-ignore-case": "tBridge" }}]}`

Confronto	Esempio	Sintassi delle regole
Exists	ProductName esiste	`"ProductName": [ { "exists": true } ]`
Does not exist	ProductName non esiste	`"ProductName": [ { "exists": false } ]`
Not	Il tempo è qualsiasi tranne "piovoso"	`"Weather": [ { "anything-but": [ "Raining" ] } ]`
Null	UserID è nullo	`"UserID": [ null ]`
Numeric (uguale)	Il prezzo è 100	`"Price": [ { "numeric": [ "=", 100 ] } ]`
Numeric (intervallo)	Il prezzo è superiore a 10 e inferiore o uguale a 20	`"Price": [ { "numeric": [ ">", 10, "<=", 20 ] } ]`
Or	PaymentType è «Credito» o «Debito»	`"PaymentType": [ "Credit", "Debit"]`
Or (campi multipli)	La posizione è "New York" o il giorno è "lunedì".	`"$or": [ { "Location": [ "New York" ] }, { "Day": [ "Monday" ] } ]`
Carattere jolly	Qualsiasi file con estensione .png, situato nella cartella "dir"	`"FileName": [ { "wildcard": "dir/*.png" } ]`

Esempi di eventi e modelli di eventi

Puoi utilizzare tutti i tipi e i valori di dati JSON per trovare eventi corrispondenti. Di seguito sono riportati esempi di eventi e modelli di eventi corrispondenti.

Corrispondenza in base ai campi

È possibile trovare una corrispondenza in base al valore di un campo. Considera il seguente evento di Dimensionamento automatico Amazon EC2.

```
{
  "version": "0",
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventVersion": "",
    "responseElements": null
  }
}
```

Per l'evento precedente, puoi utilizzare il campo `"responseElements"` come criterio di corrispondenza.

```
{
  "source": ["aws.autoscaling"],
  "detail-type": ["EC2 Instance Launch Successful"],
  "detail": {
   "responseElements": [null]
  }
}
```

Corrispondenza in base ai valori

Considera il seguente evento Amazon Macie, che è troncato.

```
{
  "version": "0",
  "id": "0948ba87-d3b8-c6d4-f2da-732a1example",
  "detail-type": "Macie Finding",
  "source": "aws.macie",
  "account": "123456789012",
  "time": "2021-04-29T23:12:15Z",
  "region":"us-east-1",
  "resources": [

  ],
  "detail": {
```

```
    "schemaVersion": "1.0",
    "id": "64b917aa-3843-014c-91d8-937ffexample",
    "accountId": "123456789012",
    "partition": "aws",
    "region": "us-east-1",
    "type": "Policy:IAMUser/S3BucketEncryptionDisabled",
    "title": "Encryption is disabled for the S3 bucket",
    "description": "Encryption is disabled for the Amazon S3 bucket. The data in the
 bucket isn’t encrypted
        using server-side encryption.",
    "severity": {
        "score": 1,
        "description": "Low"
    },
    "createdAt": "2021-04-29T15:46:02Z",
    "updatedAt": "2021-04-29T23:12:15Z",
    "count": 2,
.
.
.
```

Il seguente modello di eventi corrisponde a qualsiasi evento con un punteggio di gravità pari a 1 e un conteggio pari a 2.

```
{
  "source": ["aws.macie"],
  "detail-type": ["Macie Finding"],
  "detail": {
    "severity": {
      "score": [1]
    },
    "count":[2]
  }
}
```

Corrispondenza di valori nulli e stringhe vuote nei modelli di eventi di Amazon EventBridge

> **Important**
>
> Nel EventBridge, è possibile creare regole che possono comportare higher-than-expected addebiti e limitazioni. Ad esempio, puoi creare inavvertitamente una regola che genera un ciclo infinito, in cui una regola viene attivata in modo ricorsivo senza fine. Ad esempio, hai creato una regola per rilevare eventuali modifiche alle liste di controllo degli accessi (ACL) in un bucket S3 e attivare un programma software che le imposti sullo stato desiderato. Se la regola non è scritta con attenzione, la successiva modifica alle ACL la riattiva, creando un loop infinito.
>
> Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta ??? e ???.

È possibile creare un modello di eventi che corrisponde a un campo evento con un valore null o in una stringa vuota. Analizza l'esempio seguente dell'evento .

Consulta le best practice per evitare addebiti e limitazioni superiori al previsto

```
{
  "version": "0",
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": [
   ],
  "detail": {
    "eventVersion": "",
    "responseElements": null
   }
}
```

Per trovare eventi corrispondenti in cui il valore di `eventVersion` è una stringa vuota, utilizza il seguente modello di eventi, che corrisponde all'evento precedente.

```
{
  "detail": {
     "eventVersion": [""]
  }
}
```

Per trovare eventi corrispondenti in cui il valore di `responseElements` è null, utilizza il seguente modello di eventi, che corrisponde all'evento precedente.

```
{
  "detail": {
     "responseElements": [null]
  }
}
```

> **Note**
>
> I valori Null e le stringhe vuote non sono intercambiabili nell'abbinamento dei modelli. Un modello di eventi che corrisponde a stringhe vuote non corrisponde ai valori `null`.

Array nei modelli di EventBridge eventi di Amazon

Il valore di ogni campo in un modello di eventi è un array contenente uno o più valori. Un modello di eventi corrisponde all'evento se uno qualsiasi dei valori nell'array corrisponde al valore nell'evento. Se il valore dell'evento è un array, il modello di eventi corrisponde se l'intersezione dell'array del modello di eventi e l'array dell'evento non è vuota.

> **Important**
>
> Nel EventBridge, è possibile creare regole che possono portare ad higher-than-expected addebiti e limitazioni. Ad esempio, puoi creare inavvertitamente una regola che genera un ciclo infinito, in cui una regola viene attivata in modo ricorsivo senza fine. Ad esempio, hai creato una regola per rilevare eventuali modifiche alle liste di controllo degli accessi (ACL) in un bucket S3 e attivare un programma software che le imposti sullo stato desiderato. Se la regola non è scritta con attenzione, la successiva modifica alle ACL la riattiva, creando un loop infinito.
>
> Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta ??? e ???.

Ad esempio, considera un modello di eventi che include il campo seguente.

```
"resources": [
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:111122223333:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:444455556666:instance/i-b188560f",
]
```

Il modello di esempio precedente corrisponde a un evento che include il campo seguente in quanto la prima voce nell'array del modello di eventi corrisponde alla seconda voce nell'array dell'evento.

```
"resources": [
   "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-
b893-d5978ed4a025:autoScalingGroupName/ASGTerminate",
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f"
]
```

Filtraggio dei contenuti nei modelli di EventBridge eventi di Amazon

Amazon EventBridge supporta il filtraggio dichiarativo dei contenuti utilizzando modelli di eventi. Grazie ai filtri di contenuti, puoi creare modelli di eventi complessi che corrispondono a eventi solo in condizioni molto specifiche. Ad esempio, puoi creare un modello di eventi che corrisponde a un evento quando:

- Un campo dell'evento rientra in un intervallo numerico specifico.
- L'evento proviene da un indirizzo IP specifico.
- Non esiste un campo specifico nell'evento JSON.

> **Important**
>
> Nel EventBridge, è possibile creare regole che possono portare ad higher-than-expected addebiti e limitazioni. Ad esempio, puoi creare inavvertitamente una regola che genera un ciclo infinito, in cui una regola viene attivata in modo ricorsivo senza fine. Ad esempio, hai creato una regola per rilevare eventuali modifiche alle liste di controllo degli accessi (ACL) in un bucket S3 e attivare un programma software che le imposti sullo stato desiderato. Se la regola non è scritta con attenzione, la successiva modifica alle ACL la riattiva, creando un loop infinito.
>
> Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta ??? e ???.

Tipi di filtro

- Corrispondenza in base al prefisso
- Corrispondenza in base al suffisso
- Corrispondenza anything-but
- Corrispondenza numerica
- Corrispondenza in base all'indirizzo IP
- Corrispondenza in base all'esistenza
- quals-ignore-caseCorrispondenza E
- Corrispondenza tramite caratteri jolly
- Esempio complesso con corrispondenza multipla

- Esempio complesso con corrispondenza $or

Corrispondenza in base al prefisso

Puoi trovare un evento corrispondente a seconda del prefisso di un valore nell'origine dell'evento. È possibile utilizzare la corrispondenza in base al prefisso per i valori delle stringhe.

Ad esempio, il seguente modello di eventi corrisponderebbe a qualsiasi evento in cui il campo `"time"` comincia con `"2017-10-02"`, come in `"time": "2017-10-02T18:43:48Z"`.

```
{
  "time": [ { "prefix": "2017-10-02" } ]
}
```

Corrispondenza dei prefissi ignorando le maiuscole

È inoltre possibile abbinare un valore di prefisso indipendentemente dalla maiuscola e minuscola dei caratteri con cui inizia un valore, utilizzando insieme a `equals-ignore-case` `prefix.`

Ad esempio, il seguente modello di evento corrisponderebbe a qualsiasi evento in cui il `service` campo inizia con la stringa di caratteri`EventB`, ma anche `EVENTBeventb`, o qualsiasi altra scrittura maiuscola di tali caratteri.

```
{
  "detail": {"service" : [{ "prefix": { "equals-ignore-case": "EventB" }}]}
}
```

Corrispondenza in base al suffisso

Puoi trovare un evento corrispondente a seconda del suffisso di un valore nell'origine dell'evento. È possibile utilizzare la corrispondenza in base al suffisso per i valori delle stringhe.

Ad esempio, il seguente modello di eventi corrisponderebbe a qualsiasi evento in cui il campo `"FileName"` termina con l'estensione di file `.png`.

```
{
  "FileName": [ { "suffix": ".png" } ]
}
```

Corrispondenza dei suffissi ignorando le maiuscole

È inoltre possibile abbinare un valore di suffisso indipendentemente dalla maiuscola e minuscola dei caratteri con cui termina un valore, utilizzando in combinazione con `equals-ignore-case suffix.`

Ad esempio, il seguente schema di eventi corrisponderebbe a qualsiasi evento in cui il `FileName` campo termina con la stringa di caratteri `.png`, ma anche `.PNG` a qualsiasi altra scrittura maiuscola di tali caratteri.

```
{
  "detail": {"FileName" : [{ "suffix": { "equals-ignore-case": ".png" }}]}
}
```

Corrispondenza anything-but

Tutto ciò che non corrisponde corrisponde a qualsiasi cosa ad eccezione di quanto specificato nella regola.

Puoi utilizzare la corrispondenza anything-but con stringhe e valori numerici, inclusi elenchi contenenti solo stringhe o solo numeri.

Il modello di eventi seguente mostra la corrispondenza anything-but con stringhe e numeri.

```
{
  "detail": {
    "state": [ { "anything-but": "initializing" } ]
  }
}

{
  "detail": {
    "x-limit": [ { "anything-but": 123 } ]
  }
}
```

Il modello di eventi seguente mostra la corrispondenza anything-but con un elenco di stringhe.

```
{
  "detail": {
```

```
    "state": [ { "anything-but": [ "stopped", "overloaded" ] } ]
  }
}
```

Il modello di eventi seguente mostra la corrispondenza anything-but con un elenco di numeri.

```
{
  "detail": {
    "x-limit": [ { "anything-but": [ 100, 200, 300 ] } ]
  }
}
```

Tutto tranne la corrispondenza ignorando le maiuscole e le minuscole

Puoi anche usarlo insieme `equals-ignore-case` a, per abbinare i valori delle stringhe indipendentemente dal `anything-but` maiuscolo e minuscolo dei caratteri.

Il seguente modello di eventi corrisponde ai `state` campi che non contengono la stringa «initializing», «INITIALIZING», «Initializing» o qualsiasi altra forma di maiuscolo di tali caratteri.

```
{
  "detail": {"state" : [{ "anything-but": { "equals-ignore-case": "initializing" }}]}
}
```

Puoi anche utilizzarlo insieme `equals-ignore-case` a `anything-but` per confrontare un elenco di valori:

```
{
  "detail": {"state" : [{ "anything-but": { "equals-ignore-case": ["initializing",
 "stopped"] }}]}
}
```

Tutto tranne la corrispondenza sui prefissi

È possibile utilizzare insieme `prefix` a `anything-but` per abbinare valori di stringa che non iniziano con il valore specificato. Ciò include valori singoli o un elenco di valori.

Il seguente schema di eventi mostra tutto tranne le corrispondenze che corrispondono a qualsiasi evento che non ha il prefisso `"init"` nel campo. `"state"`

```
{
  "detail": {
    "state": [ { "anything-but": { "prefix": "init" } } ]
  }
}
```

Il seguente schema di eventi mostra tutto tranne la corrispondenza utilizzata con un elenco di valori di prefisso. Questo modello di eventi corrisponde a qualsiasi evento che non ha né il prefisso né il campo`"init"`. `"stop"` `"state"`

```
{
"detail": {
  "state" : [{ "anything-but": { "prefix": ["init", "stop"] } } ] }
  }
}
```

Tutto tranne la corrispondenza sui suffissi

È possibile utilizzare insieme `suffix` a `anything-but` per abbinare valori di stringa che non terminano con il valore specificato. Ciò include valori singoli o un elenco di valori.

Il seguente modello di eventi corrisponde a tutti i valori del `FileName` campo che non terminano con`.txt`.

```
{
  "detail": {
    "FileName": [ { "anything-but": { "suffix": ".txt" } } ]
  }
}
```

Il seguente schema di eventi mostra tutto tranne la corrispondenza utilizzata con un elenco di valori di suffisso. Questo modello di eventi corrisponde a tutti i valori del `FileName` campo che non terminano con uno o. `.txt` `.rtf`

```
{
  "detail": {
    "FileName": [ { "anything-but": { "suffix": [".txt", ".rtf"] } } ]
  }
}
```

Tutto tranne la corrispondenza tramite caratteri jolly

È possibile utilizzare il carattere jolly (*) all'interno dei valori specificati per qualsiasi cosa tranne che per la corrispondenza. Ciò include valori singoli o un elenco di valori.

Il seguente modello di eventi corrisponde a tutti i valori del `FileName` campo che non lo contengono`/lib/`.

```
{
"detail": {
  "FilePath" : [{ "anything-but": { "wildcard": "*/lib/*" }}]
  }
}
```

Il seguente schema di eventi mostra tutto tranne la corrispondenza utilizzata con un elenco di valori che includono i caratteri jolly. Questo modello di evento corrisponde a tutti i valori del `FileName` campo che non contengono né l'uno né l'altro. `/lib/` `/bin/`

```
{
"detail": {
  "FilePath" : [{ "anything-but": { "wildcard": ["*/lib/*", "*/bin/*"] }}]
  }
}
```

Per ulteriori informazioni, consulta ???.

Corrispondenza numerica

La corrispondenza numerica funziona con valori che sono numeri JSON. È limitata a valori compresi tra -5.0e9 e +5.0e9 incluso, con 15 cifre di precisione (sei cifre a destra della virgola decimale).

Di seguito viene illustrata la corrispondenza numerica per un modello di eventi che corrisponde solo a eventi che sono veri per tutti i campi.

```
{
  "detail": {
    "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ],
    "d-count": [ { "numeric": [ "<", 10 ] } ],
    "x-limit": [ { "numeric": [ "=", 3.018e2 ] } ]
  }
```

```
}
```

Corrispondenza in base all'indirizzo IP

È possibile utilizzare la corrispondenza in base all'indirizzo IP per indirizzi IPv4 e IPv6. Il seguente modello di eventi mostra la corrispondenza in base all'indirizzo IP con indirizzi IP che iniziano con 10.0.0 e terminano con un numero compreso tra 0 e 255.

```
{
  "detail": {
    "sourceIPAddress": [ { "cidr": "10.0.0.0/24" } ]
  }
}
```

Corrispondenza in base all'esistenza

La corrispondenza in base all'esistenza è relativa alla presenza o all'assenza di un campo nel JSON dell'evento.

La corrispondenza in base all'esistenza funziona solo sui nodi foglia. Non funziona sui nodi intermedi.

Il seguente modello di eventi corrisponde a qualsiasi evento che abbia un campo `detail.state`.

```
{
  "detail": {
    "state": [ { "exists": true  } ]
  }
}
```

Il modello di eventi precedente corrisponde all'evento seguente.

```
{
  "version": "0",
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2015-11-11T21:29:54Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"],
```

```
  "detail": {
    "instance-id": "i-abcd1111",
    "state": "pending"
  }
}
```

Il modello di eventi precedente NON corrisponde all'evento seguente perché non ha un campo `detail.state`.

```
{
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1:123456789012:instance/i-02ebd4584a2ebd341" ],
  "detail": {
    "c-count" : {
       "c1" : 100
    }
  }
}
```

quals-ignore-caseCorrispondenza E

La quals-ignore-case corrispondenza E funziona sui valori di stringa indipendentemente dalle maiuscole e minuscole.

Il modello di eventi seguente corrisponde a qualsiasi evento che ha un campo `detail-type` che corrisponde alla stringa specificata, indipendentemente dall'uso di maiuscole e minuscole.

```
{
  "detail-type": [ { "equals-ignore-case": "ec2 instance state-change notification" } ]
}
```

Il modello di eventi precedente corrisponde all'evento seguente.

```
{
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1:123456789012:instance/i-02ebd4584a2ebd341" ],
  "detail": {
    "c-count" : {
       "c1" : 100
    }
  }
```

```
}
```

Corrispondenza tramite caratteri jolly

È possibile utilizzare il carattere jolly (*) per la corrispondenza con valori di stringa in modelli di eventi.

> **Note**
>
> Attualmente il carattere jolly è supportato solo nelle regole di router di eventi.

Considerazioni sull'uso dei caratteri jolly nei modelli di eventi:

- È possibile specificare un numero qualsiasi di caratteri jolly in un determinato valore di stringa; tuttavia, i caratteri jolly consecutivi non sono supportati.
- EventBridge supporta l'uso del carattere barra rovesciata (\) per specificare i caratteri letterali * e\ nei filtri jolly:
  - La stringa * rappresenta il carattere letterale *
  - La stringa \\ rappresenta il carattere letterale \

  L'utilizzo della barra rovesciata come carattere di escape per altri caratteri non è supportato.

Complessità dei caratteri jolly e dei modelli di eventi

Esiste un limite alla complessità di una regola che utilizza caratteri jolly. Se una regola è troppo complessa, EventBridge restituisce un `InvalidEventPatternException` quando tenta di creare la regola. Se la tua regola genera un errore di questo tipo, valuta la possibilità di utilizzare le istruzioni riportate di seguito per ridurre la complessità del modello di eventi:

- Riduci il numero di caratteri jolly utilizzati

  Utilizza caratteri jolly solo se è veramente necessario per la corrispondenza con molteplici valori possibili. Ad esempio, considera il seguente modello di eventi, in cui desideri trovare router di eventi corrispondenti nella stessa Regione:

```
{
"EventBusArn": [ { "wildcard": "*:*:*:*:*:event-bus/*" } ]
}
```

Nel caso precedente, molte delle sezioni dell'ARN si baseranno direttamente sulla Regione in cui risiedono i router di eventi. Quindi, se si utilizza la Regione `us-east-1`, un modello meno complesso che corrisponde comunque ai valori desiderati potrebbe essere come segue:

```
{
"EventBusArn": [ { "wildcard": "arn:aws:events:us-east-1:*:event-bus/*" } ]
}
```

- Riduci le sequenze di caratteri ripetute che si hanno dopo un carattere jolly

  La visualizzazione della stessa sequenza di caratteri più volte dopo l'uso di un carattere jolly aumenta la complessità dell'elaborazione del modello di eventi. Modifica il modello di eventi per ridurre al minimo le sequenze ripetute. Ad esempio, considera l'esempio seguente, che cerca la corrispondenza con il file `doc.txt` di qualsiasi utente:

  ```
  {
  "FileName": [ { "wildcard": "/Users/*/dir/dir/dir/dir/dir/doc.txt" } ]
  }
  ```

  Se si sapesse che il file `doc.txt` si troverebbe solo nel percorso specificato, si potrebbe ridurre la sequenza di caratteri ripetuta in questo modo:

  ```
  {
  "FileName": [ { "wildcard": "/Users/*/doc.txt" } ]
  }
  ```

Esempio complesso con corrispondenza multipla

Puoi combinare più regole di corrispondenza in un modello di evento più complesso. Ad esempio, il seguente modello di eventi combina `anything-but` e `numeric`.

```
{
  "time": [ { "prefix": "2017-10-02" } ],
  "detail": {
    "state": [ { "anything-but": "initializing" } ],
    "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ],
    "d-count": [ { "numeric": [ "<", 10 ] } ],
    "x-limit": [ { "anything-but": [ 100, 200, 300 ] } ]
  }
```

```
}
```

> **Note**
>
> Quando si creano modelli di eventi, se si include una chiave più di una volta, l'ultimo riferimento sarà quello utilizzato per valutare gli eventi. Ad esempio, per il seguente modello:
>
> ```
> {
>   "detail": {
>     "location": [ { "prefix": "us-" } ],
>     "location": [ { "anything-but": "us-east" } ]
>   }
> }
> ```
>
> solo `{ "anything-but": "us-east" }` verrà preso in considerazione nella valutazione di `location`.

Esempio complesso con corrispondenza **`$or`**

Puoi anche creare modelli di eventi complessi che verificano se i valori del campo any corrispondono in più campi. Utilizza `$or` per creare modello di eventi che corrisponde se uno qualsiasi dei valori di più campi corrisponde.

Nota che puoi includere altri tipi di filtri, come la corrispondenza numerica e array, nel modello per singoli campi nel tuo costrutto `$or`.

Il modello di eventi seguente corrisponde se viene soddisfatta una delle seguenti condizioni:

- Il campo `c-count` è maggiore di 0 o minore o uguale a 5.
- Il campo `d-count` è inferiore a 10.
- Il campo `x-limit` è uguale a 3.018e2.

```
{
  "detail": {
    "$or": [
      { "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ] },
      { "d-count": [ { "numeric": [ "<", 10 ] } ] },
```

```
      { "x-limit": [ { "numeric": [ "=", 3.018e2 ] } ] }
    ]
  }
}
```

> **Note**
>
> Le API che accettano un modello di eventi (come `PutRule`, `CreateArchive`, `UpdateArchive` e `TestEventPattern`) genereranno un `InvalidEventPatternException` se l'utilizzo di `$or` risulta in più di 1000 combinazioni di regole.
> Per determinare il numero di combinazioni di regole in un modello di eventi, moltiplica il numero totale di argomenti di ogni array `$or` del modello di eventi. Ad esempio, il modello precedente contiene un singolo array `$or` con tre argomenti, quindi anche il numero totale di combinazioni di regole è tre. Se hai aggiunto un altro array `$or` con due argomenti, le combinazioni di regole totali sarebbero quindi sei.

Test di un pattern di eventi utilizzando la Sandbox EventBridge

Le regole utilizzano modelli di eventi per selezionare eventi e inviarli alle destinazioni. I modelli di eventi hanno la stessa struttura degli eventi a cui corrispondono. Un modello di eventi può corrispondere o meno a un evento.

La definizione di un modello di eventi fa in genere parte del processo più ampio di creazione di una nuova regola o di modifica di una regola esistente. Utilizzando la Sandbox in EventBridge, tuttavia, è possibile definire rapidamente un pattern di eventi e utilizzare un evento di esempio per confermare che il pattern corrisponda agli eventi desiderati, senza dover creare o modificare una regola. Dopo aver testato il modello di evento, EventBridge avrai la possibilità di creare una nuova regola utilizzando quel modello di evento direttamente dalla sandbox.

Per ulteriori informazioni sui modelli di eventi, consulta ???.

> **Important**
>
> Inoltre EventBridge, è possibile creare regole che possono comportare higher-than-expected addebiti e limitazioni. Ad esempio, puoi creare inavvertitamente una regola che genera un ciclo infinito, in cui una regola viene attivata in modo ricorsivo senza fine. Ad esempio, hai

creato una regola per rilevare eventuali modifiche alle liste di controllo degli accessi (ACL) in un bucket S3 e attivare un programma software che le imposti sullo stato desiderato. Se la regola non è scritta con attenzione, la successiva modifica alle ACL la riattiva, creando un loop infinito.
Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta ??? e ???.

Per testare un pattern di eventi utilizzando la sandbox EventBridge

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Risorse per gli sviluppatori, quindi seleziona Sandbox e nella pagina Sandbox scegli la scheda Modello di eventi.
3. Per Event source, scegli AWS eventi o eventi EventBridge partner.
4. Nella sezione Eventi di esempio, scegli un Tipo evento di esempio in base al quale desideri testare il modello di eventi.

   Sono disponibili i seguenti tipi di evento di esempio:

   - AWS eventi: seleziona tra gli eventi emessi da Servizi AWS Supported.
   - EventBridge eventi partner: seleziona tra gli eventi emessi da servizi di terze parti che supportano EventBridge, come Salesforce.
   - Inserisci il mio: immetti il tuo evento in testo JSON.

     Puoi anche utilizzare un evento AWS o un evento partner come punto di partenza per creare il tuo evento personalizzato.

     1. Seleziona AWS eventi o eventi EventBridge partner.
     2. Nell'elenco a discesa Eventi di esempio, seleziona l'evento da utilizzare come riferimento per l'evento personalizzato.

        EventBridge visualizza l'evento di esempio.
     3. Seleziona Copia.
     4. Seleziona Inserisci il mio in Tipo di evento.
     5. Elimina la struttura degli eventi di esempio nel riquadro di modifica JSON e incolla l'evento AWS o il partner al suo posto.
     6. Modifica il testo JSON dell'evento per creare il tuo evento di esempio.

5. In Metodo di creazione, scegli un metodo di creazione. È possibile creare un modello di evento da uno EventBridge schema o modello oppure creare un modello di evento personalizzato.

   Existing schema

   Per utilizzare uno EventBridge schema esistente per creare il modello di eventi, effettuate le seguenti operazioni:

   1. Nella sezione Metodo di creazione, in Metodo, seleziona Utilizza schema.
   2. Nella sezione Modello di eventi, in Tipo di schema, seleziona Seleziona lo schema dal registro schemi.
   3. In Registro dello schema, scegli la casella a discesa e immetti il nome di un registro, ad esempio `aws.events`. Puoi anche selezionare un'opzione dall'elenco a discesa visualizzato.
   4. In Schema, scegli la casella a discesa e immetti il nome dello schema da utilizzare. Ad esempio, `aws.s3@0bjectDeleted`. Puoi anche selezionare un'opzione dall'elenco a discesa visualizzato.
   5. Nella sezione Modelli, scegli il pulsante Modifica accanto a qualsiasi attributo per visualizzarne le proprietà. Imposta i campi Relazione e Valore come necessario, quindi scegli Imposta per salvare l'attributo.

      > **Note**
      >
      > Per informazioni sulla definizione di un attributo, scegli l'icona Informazioni accanto al nome dell'attributo. Per informazioni di riferimento su come impostare le proprietà degli attributi nell'evento, apri la sezione Nota della finestra di dialogo delle proprietà degli attributi.
      > Per eliminare le proprietà di un attributo, scegli il pulsante Modifica accanto a quell'attributo, quindi scegli Cancella.

   6. Scegli Genera un modello di eventi in JSON per generare e convalidare il modello di eventi come testo JSON.
   7. Per testare l'evento di esempio in base al tuo modello di test, scegli Modello di test.

      EventBridge visualizza una finestra di messaggio che indica se l'evento di esempio corrisponde al modello di evento.

      È anche possibile scegliere una delle seguenti opzioni:

- Copia: copia il modello di eventi negli appunti del dispositivo.
- Abbellisci (Prettify): semplifica la lettura del testo JSON aggiungendo interruzioni di riga, tabulazioni e spazi.

Custom schema

Per scrivere uno schema personalizzato e convertirlo in un modello di eventi, procedi come segue:

1. Nella sezione Metodo di creazione, in Metodo, scegli Utilizza schema.
2. Nella sezione Modello di eventi, in Tipo di schema, scegli Inserisci schema.
3. Immetti lo schema nella casella di testo. Devi formattarlo come testo JSON valido.
4. Nella sezione Modelli, scegli il pulsante Modifica accanto a qualsiasi attributo per visualizzarne le proprietà. Imposta i campi Relazione e Valore come necessario, quindi scegli Imposta per salvare l'attributo.

   > **Note**
   >
   > Per informazioni sulla definizione di un attributo, scegli l'icona Informazioni accanto al nome dell'attributo. Per informazioni di riferimento su come impostare le proprietà degli attributi nell'evento, apri la sezione Nota della finestra di dialogo delle proprietà degli attributi.
   > Per eliminare le proprietà di un attributo, scegli il pulsante Modifica accanto a quell'attributo, quindi scegli Cancella.

5. Scegli Genera un modello di eventi in JSON per generare e convalidare il modello di eventi come testo JSON.
6. Per testare l'evento di esempio in base al tuo modello di test, scegli Modello di test.

   EventBridge visualizza una finestra di messaggio che indica se l'evento di esempio corrisponde al modello dell'evento.

   È anche possibile scegliere una delle seguenti opzioni:

   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Abbellisci (Prettify): semplifica la lettura del testo JSON aggiungendo interruzioni di riga, tabulazioni e spazi.

Event pattern

Per scrivere un modello di eventi personalizzato in formato JSON, procedi come segue:

1. Nella sezione Metodo di creazione, in Metodo, scegli Modello personalizzato (editor JSON).
2. In Modello di eventi, immetti il modello di eventi personalizzato in testo in formato JSON.
3. Per testare l'evento di esempio in base al tuo modello di test, scegli Modello di test.

   EventBridge visualizza una finestra di messaggio che indica se l'evento di esempio corrisponde al modello dell'evento.

   È anche possibile scegliere una delle seguenti opzioni:

   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Abbellisci (Prettify): semplifica la lettura del testo JSON aggiungendo interruzioni di riga, tabulazioni e spazi.
   - Modulo del modello di eventi: apre il modello di eventi in Generatore di modello. Se il pattern non può essere renderizzato in Pattern Builder così com'è, EventBridge avvisa l'utente prima di aprire Pattern Builder.

6. (Facoltativo) Per creare una regola con questo modello di eventi e assegnarla a un router di eventi specifico, scegli Creazione di una regola con modello.

   EventBridge ti porta alla Fase 1 di Create rule, che puoi usare per creare una regola e assegnarla al bus di eventi di tua scelta.

   Nota che la sezione Passaggio 2: creare un modello di eventi contiene le informazioni sul modello di eventi che hai già specificato e che puoi accettare o aggiornare.

   Per ulteriori informazioni su come creare regole, consulta ???.

Le migliori pratiche per la definizione dei modelli di EventBridge eventi Amazon

Di seguito sono riportate alcune best practice da prendere in considerazione quando si definiscono modelli di eventi nelle regole di router di eventi.

Evitare di scrivere loop infiniti

In EventBridge, è possibile creare regole che portano a loop infiniti, in cui una regola viene attivata ripetutamente. Ad esempio, una regola potrebbe rilevare che le ACL sono state modificate in un bucket S3 e attivare il software per ripristinare lo stato desiderato. Se la regola non è scritta con attenzione, la successiva modifica alle ACL la riattiva, creando un loop infinito.

Per evitare questi problemi, scrivi i modelli di eventi per le tue regole in modo che siano il più precisi possibile, affinché corrispondano solo agli eventi che desideri effettivamente inviare alla destinazione. Nell'esempio precedente, creeresti un modello di eventi per trovare eventi corrispondenti di modo che le azioni attivate non riattivino la stessa regola. Ad esempio, crea un modello di eventi nella regola per trovare eventi corrispondenti solo se lo stato delle liste di controllo degli accessi (ACL) non è corretto, anziché dopo qualsiasi modifica. Per ulteriori informazioni, consulta ??? e ???.

Un loop infinito può generare rapidamente costi più alti di quelli previsti. Può anche comportare limitazioni e ritardi nella distribuzione degli eventi. Puoi monitorare il limite superiore delle frequenze di invocazioni per essere avvisato in caso di picchi di volume imprevisti.

Utilizza il budgeting per ricevere avvisi quando gli addebiti superano il limite specificato. Per ulteriori informazioni, consulta Gestione dei costi con i budget.

Rendere i modelli di eventi il più precisi possibile

Più preciso è il modello di eventi, più è probabile che corrisponda solo agli eventi effettivamente desiderati e che eviti corrispondenze impreviste quando vengono aggiunti nuovi eventi a un'origine di eventi o gli eventi esistenti vengono aggiornati per includere nuove proprietà.

I modelli di eventi possono includere filtri per trovare corrispondenze con:

- Metadati relativi all'evento, ad esempio `source`. `detail-type` `account` oppure `region`.
- Dati relativi all'evento, ovvero i campi all'interno dell'oggetto `detail`.
- Contenuto dell'evento o valori effettivi dei campi all'interno dell'oggetto `detail`.

La maggior parte dei modelli è semplice, ad esempio specificando solo i filtri`source` e `detail-type`. Tuttavia, EventBridge i pattern includono la flessibilità di filtrare in base a qualsiasi chiave o valore dell'evento. Inoltre, puoi applicare filtri di contenuto come i filtri `prefix` e `suffix` per migliorare la precisione dei modelli. Per ulteriori informazioni, consulta ???.

Specificare l'origine dell'evento e il tipo di dettagli come filtri

Puoi ridurre la generazione di loop infiniti e la corrispondenza di eventi indesiderati rendendo più precisi i modelli di eventi mediante i campi di metadati `source` e `detail-type`.

Quando devi trovare la corrispondenza con valori specifici in due o più campi, utilizza l'operatore di confronto `$or` anziché elencare tutti i valori possibili in un unico array di valori.

Per gli eventi che vengono erogati tramite AWS CloudTrail, ti consigliamo di utilizzare il `eventName` campo come filtro.

Il seguente esempio di pattern di eventi corrisponde `CreateQueue` o `SetQueueAttributes` proviene dal servizio Amazon Simple Queue Service `CreateKey` o a `DisableKeyRotation` eventi del AWS Key Management Service servizio.

```
{
  "detail-type": ["AWS API Call via CloudTrail"],
  "$or": [{
      "source": [
        "aws.sqs"
        ],
      "detail": {
        "eventName": [
          "CreateQueue",
          "SetQueueAttributes"
        ]
      }
    },
    {
      "source": [
        "aws.kms"
        ],
      "detail": {
        "eventName": [
          "CreateKey",
          "DisableKeyRotation"
        ]
      }
    }
  ]
}
```

Specificare l'account e la Regione come filtri

L'inclusione dei campi `account` e `region` nel modello di eventi aiuta a limitare la corrispondenza di eventi in più account o regioni.

Specificare filtri basati sul contenuto

I filtri basati sul contenuto possono aiutare a migliorare la precisione dei modelli di eventi, mantenendo comunque al minimo la lunghezza del modello di eventi. Ad esempio, anziché elencare tutti i possibili valori numerici, può risultare più utile avere una corrispondenza basata su un intervallo numerico.

Per ulteriori informazioni, consulta ???.

Definire l'ambito dei modelli di eventi per tenere conto degli aggiornamenti delle origini di eventi

Quando crei modelli di eventi, devi considerare che gli schemi e i domini di eventi possono evolversi ed espandersi nel tempo. Anche in questo caso, rendere i modelli di eventi il più precisi possibile aiuta a limitare le corrispondenze impreviste se l'origine dell'evento cambia o si espande.

Ad esempio, supponi di cercare eventi corrispondenti di un nuovo microservizio che pubblica eventi relativi ai pagamenti. Inizialmente, il servizio utilizza il dominio `acme.payments` e pubblica un singolo evento, `Payment accepted`:

```
{
  "detail-type": "Payment accepted",
  "source": "acme.payments",
  "detail": {
    "type": "credit",
    "amount": "100",
    "date": "2023-06-10",
    "currency": "USD"
    }
  }
}
```

A questo punto, potresti creare un modello di eventi semplice che corrisponda agli eventi per i pagamenti accettati:

```
{ “source” : “acme.payments” }
```

Tuttavia, supponi che il servizio introduca in un secondo momento un nuovo evento per i pagamenti rifiutati:

```
{
  "detail-type": "Payment rejected",
  "source": "acme.payments",
  "detail": {
  }
}
```

In questo caso, il modello di eventi semplice che hai creato ora corrisponderà a entrambi gli eventi `Payment accepted` e `Payment rejected`. EventBridge indirizza entrambi i tipi di eventi verso la destinazione specificata per l'elaborazione, con possibili errori di elaborazione e costi di elaborazione aggiuntivi.

Per definire l'ambito del modello di eventi affinché sia relativo solo agli eventi `Payment accepted`, dovresti specificare, come minimo, `source` e `detail-type`:

```
{
  "detail-type": "Payment accepted",
  "source": "acme.payments"
  }
}
```

Nel modello di eventi puoi anche specificare l'account e la Regione, per limitare ulteriormente l'ambito quando eventi multi-account o multiregionali corrispondono a questa regola.

```
{
  "account": "012345678910",
  "source": "acme.payments",
  "region": "AWS-Region",
  "detail-type": "Payment accepted"
}
```

Convalidare i modelli di eventi

Per garantire che le regole corrispondano agli eventi desiderati, ti consigliamo vivamente di convalidare i modelli di eventi. Puoi convalidare i tuoi modelli di eventi utilizzando la EventBridge console o l'API:

- Nella EventBridge console, puoi creare e testare modelli di eventi come parte della creazione di una regola o separatamente utilizzando la Sandbox.
- Puoi testare i modelli degli eventi a livello di codice utilizzando l'azione. TestEventPattern

EventBridge Regole Amazon

Devi specificare EventBridge cosa fare con gli eventi distribuiti a ciascun bus di eventi. Per fare ciò, create delle regole. Una regola specifica quali eventi inviare a quali destinazioni per l'elaborazione. Una singola regola può inviare un evento a più destinazioni, che vengono eseguite in parallelo.

Puoi creare due tipi di regole:

- Regole che corrispondono ai dati degli eventi

  È possibile creare regole che corrispondano agli eventi in arrivo in base a criteri relativi ai dati degli eventi (denominati pattern di eventi). Un modello di eventi definisce la struttura dell'evento e i campi a cui una regola corrisponde. Se un evento corrisponde ai criteri definiti nel modello di evento, lo EventBridge invia alle destinazioni specificate.

  Per ulteriori informazioni, consulta ???.
- Regole che vengono eseguite secondo una pianificazione

  È inoltre possibile creare regole che inviano eventi alle destinazioni specificate a intervalli specifici. Ad esempio, per eseguire periodicamente una Lambda funzione, è possibile creare una regola da eseguire in base a una pianificazione.

> **Note**
>
> EventBridge offre Amazon EventBridge Scheduler, uno strumento di pianificazione senza server che ti consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato. EventBridge Scheduler è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole EventBridge pianificate, con un set più ampio di operazioni e servizi API mirati. AWS
> Ti consigliamo di utilizzare EventBridge Scheduler per richiamare gli obiettivi in base a una pianificazione. Per ulteriori informazioni, consulta ???.

Il video seguente fornisce informazioni di base sulle regole: What are rules

Regole EventBridge gestite da Amazon

Oltre alle regole create dall'utente, AWS i servizi possono creare e gestire EventBridge le regole AWS dell'account necessarie per determinate funzioni di tali servizi. Queste regole sono denominate regole gestite.

Quando un servizio crea una regola gestita, può anche creare una IAM politica che concede al servizio l'autorizzazione a creare la regola. Per consentire la creazione delle sole regole necessarie, l'ambito delle policy IAM create in questo modo può essere limitato con autorizzazioni a livello di risorsa.

Puoi eliminare le regole gestite utilizzando l'opzione Forza eliminazione, ma devi eliminarle solo se hai la certezza che non siano più necessarie all'altro servizio. In caso contrario, se elimini una regola gestita, le caratteristiche che si basano su di essa smettono di funzionare.

Creazione di regole Amazon EventBridge che reagiscono agli eventi

Per intraprendere azioni in relazione agli eventi ricevuti da Amazon EventBridge, puoi creare delle regole. Quando un evento corrisponde al modello di eventi definito nella regola, EventBridge invia l'evento alla destinazione specificata e attiva l'azione definita nella regola.

Nel video seguente viene illustrato come creare e verificare differenti tipi di regole: Learning about rules .

Utilizza la procedura seguente per creare una regola Amazon Eventbridge che risponde agli eventi.

Creazione di una regola che reagisce agli eventi

I passaggi seguenti forniscono procedure dettagliate su come creare una regola che EventBridge utilizza per trovare eventi corrispondenti tra quelli inviati al router di eventi specificato.

Passaggi

- Definizione della regola
- Creazione di un modello di eventi
- Selezionare le destinazioni
- Configurazione di tag e revisione della regola

Definizione della regola

Innanzitutto, immetti un nome e una descrizione per la regola in modo da identificarla. Devi inoltre definire il router di eventi in cui la regola cerca eventi corrispondenti a un modello di eventi.

Per definire i dettagli della regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Immetti un nome ed eventualmente una descrizione per la regola rispettivamente in Nome e Descrizione.

Una regola non può avere lo stesso nome di un'altra regola nella stessa Regione AWS e sullo stesso bus di eventi.

5. In Router di eventi, scegli il router di eventi da associare alla regola. Se la regola deve cercare eventi corrispondenti provenienti dal tuo account, seleziona Bus di eventi predefiniti di AWS. Quando un Servizio AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Successivo.

Creazione di un modello di eventi

Ora è necessario creare il modello di eventi. A tale scopo, specifica l'origine dell'evento, scegli la base per il modello di eventi e definisci gli attributi e i valori su cui basare la corrispondenza. Puoi anche generare il modello di eventi in JSON e verificarlo utilizzando un evento di esempio.

Per creare il modello di eventi

1. In Origine evento, scegli Eventi AWS o eventi partner EventBridge.
2. (Facoltativo) Nella sezione Eventi di esempio, in Tipo evento di esempio, scegli un tipo di evento di esempio in base al quale verificare il modello di eventi.

   Sono disponibili i seguenti tipi di evento di esempio:

   - Eventi AWS: seleziona uno degli eventi emessi dai Servizi AWS supportati.
   - Eventi partner EventBridge: seleziona uno degli eventi emessi da servizi di terze parti che supportano EventBridge, come Salesforce.
   - Inserisci il mio: immetti il tuo evento in testo JSON.

     Puoi anche utilizzare un evento partner o AWS come riferimento per la creazione di un evento personalizzato.

     1. Seleziona Eventi AWS o Eventi partner EventBridge.
     2. Nell'elenco a discesa Eventi di esempio, seleziona l'evento da utilizzare come riferimento per l'evento personalizzato.

        EventBridge visualizza l'evento di esempio.

3. Seleziona Copia.
4. Seleziona Inserisci il mio in Tipo di evento.
5. Nel riquadro di modifica JSON, elimina la struttura dell'evento di esempio e al suo posto incolla l'evento partner o AWS.
6. Modifica il testo JSON dell'evento per creare il tuo evento di esempio.

3. In Metodo di creazione, scegli un metodo di creazione. Puoi creare un modello di eventi da uno schema o un modello EventBridge oppure creare un modello di eventi personalizzato.

   Existing schema

   Per creare il modello di eventi utilizzando uno schema Eventbridge esistente, procedi come segue:

   1. Nella sezione Metodo di creazione, in Metodo, seleziona Utilizza schema.
   2. Nella sezione Modello di eventi, in Tipo di schema, seleziona Seleziona lo schema dal registro schemi.
   3. In Registro dello schema, scegli la casella a discesa e immetti il nome di un registro, ad esempio `aws.events`. Puoi anche selezionare un'opzione dall'elenco a discesa visualizzato.
   4. In Schema, scegli la casella a discesa e immetti il nome dello schema da utilizzare. Ad esempio, `aws.s3@0bjectDeleted`. Puoi anche selezionare un'opzione dall'elenco a discesa visualizzato.
   5. Nella sezione Modelli, scegli il pulsante Modifica accanto a qualsiasi attributo per visualizzarne le proprietà. Imposta i campi Relazione e Valore come necessario, quindi scegli Imposta per salvare l'attributo.

      > **Note**
      >
      > Per informazioni sulla definizione di un attributo, scegli l'icona Informazioni accanto al nome dell'attributo. Per informazioni di riferimento su come impostare le proprietà degli attributi nell'evento, apri la sezione Nota della finestra di dialogo delle proprietà degli attributi.
      > Per eliminare le proprietà di un attributo, scegli il pulsante Modifica accanto a quell'attributo, quindi scegli Cancella.

6. Scegli Genera un modello di eventi in JSON per generare e convalidare il modello di eventi come testo JSON.
7. (Facoltativo) Per testare l'evento di esempio in base al tuo modello di test, scegli Modello di test.

   EventBridge visualizza una finestra di messaggio che indica se l'evento di esempio corrisponde al modello di eventi.

   È anche possibile scegliere una delle seguenti opzioni:

   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Abbellisci (Prettify): semplifica la lettura del testo JSON aggiungendo interruzioni di riga, tabulazioni e spazi.

Custom schema

Per scrivere uno schema personalizzato e convertirlo in un modello di eventi, procedi come segue:

1. Nella sezione Metodo di creazione, in Metodo, scegli Utilizza schema.
2. Nella sezione Modello di eventi, in Tipo di schema, scegli Inserisci schema.
3. Immetti lo schema nella casella di testo. Devi formattarlo come testo JSON valido.
4. Nella sezione Modelli, scegli il pulsante Modifica accanto a qualsiasi attributo per visualizzarne le proprietà. Imposta i campi Relazione e Valore come necessario, quindi scegli Imposta per salvare l'attributo.

   > **Note**
   >
   > Per informazioni sulla definizione di un attributo, scegli l'icona Informazioni accanto al nome dell'attributo. Per informazioni di riferimento su come impostare le proprietà degli attributi nell'evento, apri la sezione Nota della finestra di dialogo delle proprietà degli attributi.
   > Per eliminare le proprietà di un attributo, scegli il pulsante Modifica accanto a quell'attributo, quindi scegli Cancella.

5. Scegli Genera un modello di eventi in JSON per generare e convalidare il modello di eventi come testo JSON.

6. (Facoltativo) Per testare l'evento di esempio in base al tuo modello di test, scegli Modello di test.

   EventBridge visualizza una finestra di messaggio che indica se l'evento di esempio corrisponde al modello di eventi.

   È anche possibile scegliere una delle seguenti opzioni:

   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Abbellisci (Prettify): semplifica la lettura del testo JSON aggiungendo interruzioni di riga, tabulazioni e spazi.

Event pattern

Per scrivere un modello di eventi personalizzato in formato JSON, procedi come segue:

1. Nella sezione Metodo di creazione, in Metodo, scegli Modello personalizzato (editor JSON).
2. In Modello di eventi, immetti il modello di eventi personalizzato in testo in formato JSON.
3. (Facoltativo) Per testare l'evento di esempio in base al tuo modello di test, scegli Modello di test.

   EventBridge visualizza una finestra di messaggio che indica se l'evento di esempio corrisponde al modello di eventi.

   È anche possibile scegliere una delle seguenti opzioni:

   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Abbellisci (Prettify): semplifica la lettura del testo JSON aggiungendo interruzioni di riga, tabulazioni e spazi.
   - Modulo del modello di eventi: apre il modello di eventi in Generatore di modello. Se non è possibile eseguire il rendering del modello in Generatore di modelli così com'è, EventBridge ti avvisa prima di aprire Generatore di modelli.

4. Scegli Successivo.

Selezionare le destinazioni

Scegli una o più destinazioni per ricevere eventi che corrispondono al modello specificato. Le destinazioni possono includere un router di eventi EventBridge, destinazioni API di EventBridge, inclusi partner SaaS come Salesforce, o un altro Servizio AWS.

Per selezionare le destinazioni

1. In Tipi di destinazione, scegli uno dei seguenti tipi di destinazione:

   Event bus

   Per selezionare un router di eventi EventBridge, seleziona Bus di eventi EventBridge, quindi procedi come segue:

   - Per utilizzare un router di eventi nella stessa Regione AWS della regola:
     1. Seleziona Bus eventi nello stesso account e nella stessa Regione.
     2. Per Bus di eventi come destinazione, scegli la casella a discesa e immetti il nome del router di eventi. Puoi anche selezionare il router di eventi dall'elenco a discesa.

        Per ulteriori informazioni, consulta ???.
   - Per utilizzare un router di eventi in una Regione AWS o account differente, segui questa regola:
     1. Seleziona Bus di eventi in un account diverso o in una Regione diversa.
     2. In Bus di eventi come destinazione, immetti l'ARN del router di eventi da utilizzare.

        Per ulteriori informazioni, consulta:

        - ???
        - ???

   API destination

   Per utilizzare una destinazione API di EventBridge, seleziona Destinazione API di EventBridge, quindi esegui una delle seguenti operazioni:

   - Per utilizzare una destinazione API esistente, seleziona Utilizza una destinazione API esistente. Quindi seleziona una destinazione API dall'elenco a discesa.

- Per creare una nuova destinazione API, seleziona Crea una nuova destinazione API. Fornisci quindi i seguenti dettagli per la destinazione:
  - Nome: immetti un nome per la destinazione.

    I nomi devono essere univoci nel tuo Account AWS. I nomi possono avere fino a 64 caratteri. I caratteri validi sono A-Z, a-z, 0-9 e . _ - (trattino).
  - (Facoltativo) Descrizione: immetti una descrizione per la destinazione.

    La descrizione può avere un massimo di 512 caratteri.
  - Endpoint di destinazione API: l'endpoint URL per la destinazione.

    L'URL dell'endpoint deve iniziare con **`https`**. Puoi includere il carattere jolly **`*`** come parametro di percorso. Puoi impostare i parametri di percorso dall'attributo `HttpParameters` della destinazione.
  - Metodo HTTP: seleziona il metodo HTTP utilizzato quando richiami l'endpoint.
  - (Facoltativo) Limite di velocità di invocazione al secondo: immetti il numero massimo di invocazioni accettate al secondo per la destinazione.

    Questo valore deve essere maggiore di zero. Per impostazione predefinita, è 300.
  - Connessione: scegli questa opzione per utilizzare una connessione nuova o esistente:
    - Per utilizzare una connessione esistente, seleziona Utilizza una connessione esistente e seleziona la connessione dall'elenco a discesa.
    - Per creare una nuova connessione per questa destinazione, seleziona Crea una nuova connessione, quindi definisci Nome, Tipo di destinazione e Tipo di autorizzazione della connessione. Eventualmente, puoi anche aggiungere una Descrizione per la connessione.

Per ulteriori informazioni, consulta ???.

Servizio AWS

Per utilizzare un Servizio AWS, seleziona Servizio AWS, quindi procedi come segue:

1. In Seleziona una destinazione, seleziona un Servizio AWS da utilizzare come destinazione. Fornisci le informazioni richieste per il servizio selezionato.

> **Note**
>
> I campi visualizzati variano a seconda del servizio selezionato. Per ulteriori informazioni sulle destinazioni disponibili, consulta Obiettivi disponibili nella EventBridge console.

2. Per molti tipi di destinazione, EventBridge necessita di autorizzazioni per l'invio degli eventi alla destinazione. In questi casi, EventBridge è in grado di creare il ruolo IAM necessario per l'esecuzione della regola.

   In Ruolo di esecuzione, esegui una delle seguenti operazioni:

   - Per creare un nuovo ruolo di esecuzione per questa regola:

     a. Seleziona Crea un nuovo ruolo per questa risorsa specifica.

     b. Immetti un nome per questo ruolo di esecuzione o utilizza il nome generato da EventBridge.

   - Per utilizzare un ruolo di esecuzione esistente per questa regola:

     a. Seleziona Utilizza un ruolo esistente.

     b. Immetti o seleziona il nome del ruolo di esecuzione da utilizzare dall'elenco a discesa.

3. (Facoltativo) In Impostazioni aggiuntive, specifica una delle impostazioni facoltative disponibili per il tipo di destinazione:

   Event bus

   (Facoltativo) In Coda DLQ scegli se utilizzare una coda Amazon SQS standard come coda DLQ. EventBridge invia eventi che corrispondono a questa regola alla coda DLQ se non vengono recapitati correttamente alla destinazione. Completa una delle seguenti operazioni:

   - Scegli Nessuna per non utilizzare una coda DLQ.
   - Scegli Seleziona una coda Amazon SQS nell'account AWS corrente da utilizzare come coda DLQ, quindi seleziona la coda da utilizzare dal menu a discesa.
   - Scegli Seleziona una coda Amazon SQS in un altro account AWS come coda DLQ e specifica l'ARN della coda da utilizzare. È necessario collegare alla coda una policy basata su risorse che concede l'autorizzazione EventBridge per l'invio di messaggi.

     Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.

API destination

1. (Facoltativo) In Configura l'input di destinazione, scegli come personalizzare il testo inviato alla destinazione per gli eventi corrispondenti. Scegli una delle seguenti opzioni:
   - Eventi corrispondenti: EventBridge invia l'intero evento di origine originale alla destinazione. Questa è l'impostazione predefinita.
   - Parte degli eventi corrispondenti: EventBridge invia solo la parte specificata dell'evento di origine originale alla destinazione.

     In Specifica la parte dell'evento corrispondente, specifica un percorso JSON che definisce la parte dell'evento che EventBridge deve inviare alla destinazione.
   - Costante (testo JSON): EventBridge invia solo il testo JSON specificato alla destinazione. Non viene inviata alcuna parte dell'evento di origine originale.

     In Specifica la costante in JSON, specifica il testo JSON che EventBridge deve inviare alla destinazione anziché all'evento.
   - Trasformatore di input: configura un trasformatore di input per personalizzare il testo che EventBridge deve inviare alla destinazione. Per ulteriori informazioni, consulta ???.

     a. Seleziona Configura il trasformatore di input.

     b. Configura il trasformatore di input seguendo la procedura descritta in ???.
2. (Facoltativo) In Policy di ripetizione, specifica in che modo EventBridge deve ritentare l'invio di un evento a una destinazione dopo che si è verificato un errore.
   - Durata massima dell'evento: immetti il periodo di tempo massimo (in ore, minuti e secondi) durante il quale EventBridge deve mantenere gli eventi non elaborati. Il valore predefinito è 24 ore.
   - Nuovi tentativi: immetti il numero massimo di volte in cui EventBridge deve riprovare a inviare un evento alla destinazione dopo che si è verificato un errore. L'impostazione predefinita è 185 volte.
3. (Facoltativo) In Coda DLQ scegli se utilizzare una coda Amazon SQS standard come coda DLQ. EventBridge invia eventi che corrispondono a questa regola alla coda DLQ se non vengono recapitati correttamente alla destinazione. Completa una delle seguenti operazioni:
   - Scegli Nessuna per non utilizzare una coda DLQ.

- Scegli Seleziona una coda Amazon SQS nell'account AWS corrente da utilizzare come coda DLQ, quindi seleziona la coda da utilizzare dal menu a discesa.
- Scegli Seleziona una coda Amazon SQS in un altro account AWS come coda DLQ e specifica l'ARN della coda da utilizzare. È necessario collegare alla coda una policy basata su risorse che concede l'autorizzazione EventBridge per l'invio di messaggi.

  Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.

AWS service

Tieni presente che EventBridge potrebbe non visualizzare tutti i seguenti campi per un determinato servizio AWS.

1. (Facoltativo) In Configura l'input di destinazione, scegli come personalizzare il testo inviato alla destinazione per gli eventi corrispondenti. Scegli una delle seguenti opzioni:
   - Eventi corrispondenti: EventBridge invia l'intero evento di origine originale alla destinazione. Questa è l'impostazione predefinita.
   - Parte degli eventi corrispondenti: EventBridge invia solo la parte specificata dell'evento di origine originale alla destinazione.

     In Specifica la parte dell'evento corrispondente, specifica un percorso JSON che definisce la parte dell'evento che EventBridge deve inviare alla destinazione.
   - Costante (testo JSON): EventBridge invia solo il testo JSON specificato alla destinazione. Non viene inviata alcuna parte dell'evento di origine originale.

     In Specifica la costante in JSON, specifica il testo JSON che EventBridge deve inviare alla destinazione anziché all'evento.
   - Trasformatore di input: configura un trasformatore di input per personalizzare il testo che EventBridge deve inviare alla destinazione. Per ulteriori informazioni, consulta ???.

     a. Seleziona Configura il trasformatore di input.

     b. Configura il trasformatore di input seguendo la procedura descritta in ???.
2. (Facoltativo) In Policy di ripetizione, specifica in che modo EventBridge deve ritentare l'invio di un evento a una destinazione dopo che si è verificato un errore.
   - Durata massima dell'evento: immetti il periodo di tempo massimo (in ore, minuti e secondi) durante il quale EventBridge deve mantenere gli eventi non elaborati. Il valore predefinito è 24 ore.

- Nuovi tentativi: immetti il numero massimo di volte in cui EventBridge deve riprovare a inviare un evento alla destinazione dopo che si è verificato un errore. L'impostazione predefinita è 185 volte.

3. (Facoltativo) In Coda DLQ scegli se utilizzare una coda Amazon SQS standard come coda DLQ. EventBridge invia eventi che corrispondono a questa regola alla coda DLQ se non vengono recapitati correttamente alla destinazione. Completa una delle seguenti operazioni:
   - Scegli Nessuna per non utilizzare una coda DLQ.
   - Scegli Seleziona una coda Amazon SQS nell'account AWS corrente da utilizzare come coda DLQ, quindi seleziona la coda da utilizzare dal menu a discesa.
   - Scegli Seleziona una coda Amazon SQS in un altro account AWS come coda DLQ e specifica l'ARN della coda da utilizzare. È necessario collegare alla coda una policy basata su risorse che concede l'autorizzazione EventBridge per l'invio di messaggi.

     Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.
4. (Facoltativo) Scegli Aggiungi un'altra destinazione per aggiungere un'altra destinazione per questa regola.
5. Scegli Successivo.

Tieni presente che EventBridge potrebbe non visualizzare tutti i seguenti campi per un determinato servizio AWS.

Configurazione di tag e revisione della regola

Infine, immetti i tag desiderati per la regola, quindi rivedi e crea la regola.

Per configurare i tag e rivedere e creare la regola

1. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta EventBridge Etichette Amazon.
2. Scegli Successivo.
3. Rivedi i dettagli della nuova regola. Per apportare modifiche a una qualsiasi sezione, scegli il pulsante Modifica accanto alla sezione in questione.

   Quando sei soddisfatto dei dettagli della regola, scegli Crea regola.

Utilizzo di Pianificatore Amazon EventBridge con Amazon EventBridge

Pianificatore Amazon EventBridge è un pianificatore serverless che consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato. Con Pianificatore EventBridge, puoi creare pianificazioni utilizzando espressioni Cron ed espressioni della frequenza per modelli ricorrenti oppure configurare invocazioni una tantum. Puoi configurare finestre temporali flessibili per la consegna, definire limiti per nuovi tentativi e impostare il tempo massimo di conservazione per invocazioni API non riuscite.

Pianificatore EventBridge è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole pianificate EventBridge, con un set più ampio di servizi AWS e operazioni API di destinazione. Ti consigliamo di utilizzare il Pianificatore EventBridge per richiamare le destinazioni su una pianificazione.

Argomenti

Configurare il ruolo di esecuzione

Quando crei una nuova pianificazione, il PIanificatore EventBridge deve disporre dell'autorizzazione per richiamare automaticamente l'operazione dell'API di destinazione. Concedi queste autorizzazioni al Pianificatore EventBridge utilizzando un ruolo di esecuzione. La policy di autorizzazione collegata al ruolo di esecuzione della pianificazione definisce le autorizzazioni necessarie. Tali autorizzazioni dipendono dall'API di destinazione che deve essere richiamata dal Pianificatore EventBridge.

Quando utilizzi la console del Pianificatore EventBridge per creare una pianificazione, come nella procedura seguente, il Pianificatore EventBridge configura automaticamente un ruolo di esecuzione in base alla destinazione selezionata. Per creare una pianificazione utilizzando uno degli SDK del Pianificatore EventBridge, AWS CLI o AWS CloudFormation, devi disporre di un ruolo di esecuzione esistente che conceda le autorizzazioni richieste dal Pianificatore EventBridge per richiamare una destinazione. Per ulteriori informazioni sull'impostazione manuale di un ruolo di esecuzione per la pianificazione, consulta Configurazione di un ruolo di esecuzione nella Guida per l'utente di Pianificatore EventBridge.

Creare una pianificazione.

Per creare una pianificazione utilizzando la console

1. Apri la console del Pianificatore Amazon EventBridge all'indirizzo https://console.aws.amazon.com/scheduler/home.
2. Nella pagina Pianificazioni, scegli Crea pianificazione.
3. Nella pagina Specifica i dettagli della pianificazione, nella sezione Nome e descrizione della pianificazione, effettua le seguenti operazioni:

   a. Per Nome pianificazione, inserisci un nome per la pianificazione. Ad esempio, **`MyTestSchedule`**.

   b. (Facoltativo) Per Descrizione, inserisci una descrizione per la pianificazione. Ad esempio, **`My first schedule`**.

   c. Per Gruppo di pianificazioni, scegli un gruppo di pianificazioni dall'elenco a discesa. Se non hai un gruppo, scegli predefinito. Per creare un gruppo di pianificazioni, scegli crea la tua pianificazione.

   I gruppi di pianificazione vengono utilizzati per aggiungere tag a gruppi di pianificazioni.

4. • Scegli le opzioni di pianificazione.

Ricorrenza	Esegui questa operazione…	
Pianificazione una tantum Una pianificazione unica richiama una destinazione solo una volta alla data e all'ora specificate.	Per Data e ora, effettua le seguenti operazioni: • Inserisci una data valida in formato `YYYY/MM/DD` . • Inserisci un timestamp in formato `hh:mm` 24 ore. • Per Fuso orario, scegli il fuso orario.	

Ricorrenza	Esegui questa operazione…	
Pianificazione ricorrente Una pianificazione ricorrente richiama una destinazione con una frequenza specificata utilizzando un'espressione cron o un'espressione rate.	a. Per Tipo di pianificazione, esegui una delle seguenti operazioni: • Per utilizzare un'espressione Cron per definire la pianificazione, scegli Pianificazione basata su cron e immetti l'espressione Cron. • Per utilizzare un'espressione di frequenza per definire la pianificazione, scegli Pianificazione basata su frequenza e inserisci l'espressione di frequenza. Per ulteriori informazioni sulle espressioni Cron e rate, consulta Tipi di pianificazione nel Pianificatore EventBridge nella Guida per l'utente di Pianificatore Amazon EventBridge. b. Per Finestra temporale flessibile, scegli Disattivata per disattivare l'opzione o scegli una delle finestre temporali	

Ricorrenza	Esegui questa operazione…	
	predefinite. Ad esempio, se scegli 15 minuti e imposti una pianificazione ricorrente per il richiamo della destinazione ogni ora, la pianificazione viene eseguita entro 15 minuti dall'inizio di ogni ora.	

5. (Facoltativo) Se hai scelto Pianificazione ricorrente nel passaggio precedente, nella sezione Intervallo di tempo effettua le seguenti operazioni:

   a. Per Fuso orario, scegli un fuso orario.

   b. Per Data e ora di inizio, inserisci una data valida in formato `YYYY/MM/DD`, quindi specifica un timestamp in formato `hh:mm` 24 ore.

   c. Per Data e ora di fine, inserisci una data valida in formato `YYYY/MM/DD`, quindi specifica un timestamp in formato `hh:mm` 24 ore.

6. Scegli Successivo.

7. Nella pagina Seleziona destinazione, scegli l'operazione API AWS richiamata da Pianificatore EventBridge:

   a. In API di destinazione, scegli Destinazioni basate su modelli.

   b. Scegli Amazon EventBridge PutEvents.

   c. In PutEvents, specifica quanto segue:

      - In Bus di eventi EventBridge, scegli il router di eventi dal menu a discesa. Ad esempio, **`default`**.

        Puoi anche creare un nuovo router di eventi nella console EventBridge scegliendo Crea un nuovo bus di eventi.

      - In Detail-type, immetti il tipo di dettaglio degli eventi per i quali intendi trovare una corrispondenza. Ad esempio, **`Object Created`**.

      - In Source, immetti il nome del servizio che è l'origine degli eventi.

Per gli eventi di servizi AWS, specifica il prefisso del servizio come origine. Non includere il prefisso `aws.`. Ad esempio, per gli eventi Amazon S3 immetti **`s3`**.

Per determinare il prefisso di un servizio, consulta La tabella delle chiavi di condizione nella Guida di riferimento per l'autorizzazione del servizio. Per ulteriori informazioni sui valori relativi a origine e tipo di dettaglio degli eventi, consulta ???.

- (Facoltativo) In Dettaglio, immetti un modello di eventi per filtrare ulteriormente gli eventi che Pianificatore EventBridge invia a EventBridge.

  Per ulteriori informazioni, consulta ???.

8. Scegli Successivo.
9. Nella pagina Settings (Impostazioni), eseguire le operazioni descritte di seguito.

   a. Per attivare la pianificazione, in Stato della pianificazione, attiva Abilita pianificazione.

   b. Per configurare una policy di ripetizione per la tua pianificazione, in Policy di ripetizione e coda DLQ (Dead-Letter Queue) effettua le seguenti operazioni:

      - Attiva/disattiva Riprova.
      - Per Età massima dell'evento, inserisci il numero massimo di ore e minuti per cui il Pianificatore EventBridge deve conservare un evento non elaborato.
      - La durata massima è 24 ore.
      - Per Numero massimo di tentativi, inserisci il numero massimo di volte che il Pianificatore EventBridge ritenta la pianificazione se la destinazione restituisce un errore.

        Il valore massimo è 185 tentativi.

      Con le policy per nuovi tentativi, se una pianificazione non riesce a richiamare la sua destinazione, il Pianificatore EventBridge esegue nuovamente la pianificazione. Se configurato, è necessario impostare il tempo di conservazione massimo e i nuovi tentativi per la pianificazione.

   c. Scegli la posizione in cui il Pianificatore EventBridge deve archiviare gli eventi non consegnati.

Opzione Dead-letter queue (DLQ)	Esegui questa operazione…
Non conservare	Scegliere None (Nessuno).
Memorizza l'evento nello stesso Account AWS in cui crei la pianificazione	a. Scegli Seleziona una coda Amazon SQS nel mio Account AWS come DLQ. b. Scegli il nome della risorsa Amazon (ARN) della coda di Amazon SQS.
Memorizza l'evento in un Account AWS diverso da quello in cui crei la pianificazione	a. Scegli Specifica una coda Amazon SQS in un altro Account AWS come DLQ. b. Inserisci il nome della risorsa Amazon (ARN) della coda di Amazon SQS.

d. Per utilizzare una chiave gestita dal cliente per crittografare l'input di destinazione, in Crittografia scegli Personalizza le impostazioni di crittografia (avanzate).

   Se scegli questa opzione, inserisci l'ARN di una chiave KMS esistente scegli Crea una AWS KMS key per accedere alla console AWS KMS. Per ulteriori informazioni sulla modalità con cui il Pianificatore EventBridge esegue la crittografia dei dati inattivi, consulta Crittografia a riposo nella Guida per l'utente di Pianificatore Amazon EventBridge.

e. Affinché il Pianificatore EventBridge crei automaticamente un nuovo ruolo di esecuzione, scegli Crea nuovo ruolo per questa pianificazione. Inserisci, quindi, un nome per Nome ruolo. Se scegli questa opzione, il Pianificatore EventBridge collega al ruolo le autorizzazioni necessarie per la destinazione basata sul modello.

10. Scegli Successivo.

11. Nella pagina Rivedi e crea pianificazione, rivedi i dettagli della pianificazione. In ogni sezione, scegli Modifica per tornare a tale passaggio e modificarne i dettagli.
12. Scegli Crea pianificazione.

    Puoi visualizzare un elenco delle pianificazioni nuove ed esistenti nella pagina Pianificazioni. Nella colonna Stato, accertati che la nuova pianificazione sia Abilitata.

Risorse correlate

Per ulteriori informazioni sul Pianificatore EventBridge, consulta:

- Guida per l'utente di Pianificatore EventBridge
- Riferimento all'API del Pianificatore EventBridge
- Prezzi del Pianificatore EventBridge

Creazione di una regola Amazon EventBridge eseguita in base a una pianificazione

Una regola può essere eseguita in risposta a un evento o a determinati intervalli di tempo. Ad esempio, per eseguire periodicamente una funzione AWS Lambda, puoi creare una regola eseguita in base a una pianificazione.

> **Note**
>
> EventBridge offre Pianificatore Amazon EventBridge, un pianficatore serverless che ti consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato. Pianificatore EventBridge è un servizio altamente personalizzabile che offre una migliore scalabilità rispetto alle regole pianificate EventBridge, con una gamma più ampia di operazioni API e servizi AWS di destinazione.
> Ti consigliamo di utilizzare il Pianificatore EventBridge per richiamare le destinazioni su una pianificazione. Per ulteriori informazioni, consulta ???.

In EventBridge, è possibile creare due tipi di regole pianificate:

- Regole che vengono eseguite a una frequenza regolare

EventBridge esegue queste regole a intervalli regolari, ad esempio ogni 20 minuti.

Per specificare la frequenza per una regola pianificata, devi definire un'espressione della frequenza.

- Regole che vengono eseguite in orari specifici

  EventBridge esegue queste regole in orari e date specifici, ad esempio, alle 8:00. PST il primo lunedì di ogni mese.

  Per specificare l'ora e le date di esecuzione di una regola pianificata, si definisce un'espressione Cron.

Le espressioni della frequenza sono più semplici da definire, mentre le espressioni Cron offrono un controllo dettagliato della pianificazione. Ad esempio, con un'espressione Cron, puoi definire una regola che viene eseguita a una determinata ora di un giorno specifico di ciascuna settimana o mese. Al contrario, le espressioni della frequenza eseguono una regola a intervalli regolari, ad esempio una volta all'ora o una volta al giorno.

Tutte gli eventi pianificati utilizzano il fuso orario UTC+0 e la precisione minima per le pianificazioni è un minuto.

> **Note**
>
> EventBridge non fornisce precisione a livello di secondo nelle espressioni di pianificazione. La risoluzione più alta che utilizza un'espressione Cron è un minuto. A causa della natura distribuita di EventBridge e dei servizi di destinazione, è possibile che vi sia un ritardo di vari secondi tra il momento in cui la regola pianificata viene attivata e il momento in cui il servizio di destinazione esegue la risorsa di destinazione.

Il video seguente fornisce una panoramica delle attività di pianificazione: Creating scheduled tasks with EventBridge

Argomenti

- Creazione di una regola eseguita in base a una pianificazione
- Riferimento alle espressioni Cron

- Riferimento alle espressioni della frequenza

Creazione di una regola eseguita in base a una pianificazione

I passaggi seguenti illustrano come creare una regola EventBridge che viene eseguita in base a una pianificazione regolare.

> **Note**
>
> Puoi creare regole pianificate solo utilizzando il router di eventi predefinito.

Passaggi

- Definizione della regola
- Definizione della pianificazione
- Selezionare le destinazioni
- Configurazione di tag e revisione della regola

Definizione della regola

Innanzitutto, immetti un nome e una descrizione per la regola in modo da identificarla.

Per definire i dettagli della regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Immetti un nome ed eventualmente una descrizione per la regola rispettivamente in Nome e Descrizione.

   Una regola non può avere lo stesso nome di un'altra regola nella stessa Regione AWS e sullo stesso bus di eventi.
5. In Router di eventi, scegli il router di eventi predefinito. Puoi creare regole pianificate solo utilizzando il router di eventi predefinito.
6. Affinché la regola abbia effetto non appena la crei, assicurati che l'opzione Abilita la regola sul bus di eventi selezionato sia abilitata.

7. Per Rule type (Tipo di regola), scegli Schedule (Pianifica).

   A questo punto, puoi scegliere di continuare a creare una regola che viene eseguita in base a una pianificazione oppure utilizzare Pianificatore Amazon EventBridge.

8. Scegli come vuoi continuare:

   - Utilizza Pianificatore EventBridge per creare la pianificazione

     > **Note**
     >
     > Pianificatore EventBridge è un pianificatore serverless che ti consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato. Offre funzionalità di pianificazione una tantum e ricorrenti che non dipendono da regole e router di eventi. Pianificatore EventBridge è un servizio altamente personalizzabile che offre una migliore scalabilità rispetto alle regole pianificate EventBridge, con una gamma più ampia di operazioni API e servizi AWS di destinazione.
     > Ti consigliamo di utilizzare il Pianificatore EventBridge per richiamare le destinazioni su una pianificazione. Per ulteriori informazioni, consulta Che cos'è Pianificatore Amazon EventBridge? nella Guida per l'utente di Pianificatore Amazon EventBridge.

     1. Seleziona Continua nel pianificatore EventBridge

        EventBridge apre la pagina Crea pianificazione nella console Pianificatore EventBridge.

     2. Crea la pianificazione nella console Pianificatore EventBridge.

   - Continua a utilizzare EventBridge per creare una regola pianificata per il router di eventi predefinito

     1. Seleziona Continua per creare una regola.

Definizione della pianificazione

Ora è necessario definire il modello di pianificazione.

Per definire il modello di pianificazione

1. In Modello di pianificazione, scegli se eseguire la pianificazione venga eseguita a un orario specifico o a una frequenza normale:

Specific time

1. Scegli Una pianificazione dettagliata che viene eseguita a un'ora specifica, ad esempio alle 8:00. PST il primo lunedì di ogni mese.
2. In Espressione cron, specifica i campi per definire l'espressione Cron che EventBridge deve utilizzare per determinare quando eseguire questa regola pianificata.

   Dopo aver specificato tutti i campi, EventBridge mostra le dieci date successive in cui EventBridge eseguirà la regola pianificata. Puoi scegliere se visualizzare tali date nel fuso orario UTC o nel Fuso orario locale.

   Per ulteriori informazioni sulla creazione di un'espressione Cron, consulta ???.

Regular rate

1. Scegli Una pianificazione che viene eseguita a una frequenza regolare, ad esempio ogni 10 minuti.
2. In Espressione rate, specifica i campi Valore e Unità per definire la frequenza alla quale EventBridge deve eseguire questa regola pianificata.

   Per ulteriori informazioni sulla creazione di un'espressione della frequenza, consulta ???.

2. Scegli Successivo.

Selezionare le destinazioni

Scegli una o più destinazioni per ricevere eventi che corrispondono al modello specificato. Le destinazioni possono includere un router di eventi EventBridge, destinazioni API di EventBridge, inclusi partner SaaS come Salesforce, o un altro Servizio AWS.

Per selezionare le destinazioni

1. In Tipi di destinazione, scegli uno dei seguenti tipi di destinazione:

   Event bus

   Per selezionare un router di eventi EventBridge, seleziona Bus di eventi EventBridge, quindi procedi come segue:

- Per utilizzare un router di eventi nella stessa Regione AWS della regola:
  1. Seleziona Bus eventi nello stesso account e nella stessa Regione.
  2. Per Bus di eventi come destinazione, scegli la casella a discesa e immetti il nome del router di eventi. Puoi anche selezionare il router di eventi dall'elenco a discesa.

     Per ulteriori informazioni, consulta ???.
- Per utilizzare un router di eventi in una Regione AWS o account differente, segui questa regola:
  1. Seleziona Bus di eventi in un account diverso o in una Regione diversa.
  2. In Bus di eventi come destinazione, immetti l'ARN del router di eventi da utilizzare.

     Per ulteriori informazioni, consulta:
     - ???
     - ???

API destination

Per utilizzare una destinazione API di EventBridge, seleziona Destinazione API di EventBridge, quindi esegui una delle seguenti operazioni:

- Per utilizzare una destinazione API esistente, seleziona Utilizza una destinazione API esistente. Quindi seleziona una destinazione API dall'elenco a discesa.
- Per creare una nuova destinazione API, seleziona Crea una nuova destinazione API. Fornisci quindi i seguenti dettagli per la destinazione:
  - Nome: immetti un nome per la destinazione.

    I nomi devono essere univoci nel tuo Account AWS. I nomi possono avere fino a 64 caratteri. I caratteri validi sono A-Z, a-z, 0-9 e . _ - (trattino).
  - (Facoltativo) Descrizione: immetti una descrizione per la destinazione.

    La descrizione può avere un massimo di 512 caratteri.
  - Endpoint di destinazione API: l'endpoint URL per la destinazione.

    L'URL dell'endpoint deve iniziare con **`https`**. Puoi includere il carattere jolly **`*`** come parametro di percorso. Puoi impostare i parametri di percorso dall'attributo `HttpParameters` della destinazione.

- Metodo HTTP: seleziona il metodo HTTP utilizzato quando richiami l'endpoint.
- (Facoltativo) Limite di velocità di invocazione al secondo: immetti il numero massimo di invocazioni accettate al secondo per la destinazione.

  Questo valore deve essere maggiore di zero. Per impostazione predefinita, è 300.
- Connessione: scegli questa opzione per utilizzare una connessione nuova o esistente:
  - Per utilizzare una connessione esistente, seleziona Utilizza una connessione esistente e seleziona la connessione dall'elenco a discesa.
  - Per creare una nuova connessione per questa destinazione, seleziona Crea una nuova connessione, quindi definisci Nome, Tipo di destinazione e Tipo di autorizzazione della connessione. Eventualmente, puoi anche aggiungere una Descrizione per la connessione.

Per ulteriori informazioni, consulta ???.

Servizio AWS

Per utilizzare un Servizio AWS, seleziona Servizio AWS, quindi procedi come segue:

1. In Seleziona una destinazione, seleziona un Servizio AWS da utilizzare come destinazione. Fornisci le informazioni richieste per il servizio selezionato.

   > **Note**
   >
   > I campi visualizzati variano a seconda del servizio selezionato. Per ulteriori informazioni sulle destinazioni disponibili, consulta Obiettivi disponibili nella EventBridge console.

2. Per molti tipi di destinazione, EventBridge necessita di autorizzazioni per l'invio degli eventi alla destinazione. In questi casi, EventBridge è in grado di creare il ruolo IAM necessario per l'esecuzione della regola.

   In Ruolo di esecuzione, esegui una delle seguenti operazioni:

   - Per creare un nuovo ruolo di esecuzione per questa regola:

     a. Seleziona Crea un nuovo ruolo per questa risorsa specifica.

     b. Immetti un nome per questo ruolo di esecuzione o utilizza il nome generato da EventBridge.

- Per utilizzare un ruolo di esecuzione esistente per questa regola:

  a. Seleziona Utilizza un ruolo esistente.

  b. Immetti o seleziona il nome del ruolo di esecuzione da utilizzare dall'elenco a discesa.

3. (Facoltativo) In Impostazioni aggiuntive, specifica una delle impostazioni facoltative disponibili per il tipo di destinazione:

   Event bus

   (Facoltativo) In Coda DLQ scegli se utilizzare una coda Amazon SQS standard come coda DLQ. EventBridge invia eventi che corrispondono a questa regola alla coda DLQ se non vengono recapitati correttamente alla destinazione. Completa una delle seguenti operazioni:

   - Scegli Nessuna per non utilizzare una coda DLQ.
   - Scegli Seleziona una coda Amazon SQS nell'account AWS corrente da utilizzare come coda DLQ, quindi seleziona la coda da utilizzare dal menu a discesa.
   - Scegli Seleziona una coda Amazon SQS in un altro account AWS come coda DLQ e specifica l'ARN della coda da utilizzare. È necessario collegare alla coda una policy basata su risorse che concede l'autorizzazione EventBridge per l'invio di messaggi.

     Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.

   API destination

   1. (Facoltativo) In Configura l'input di destinazione, scegli come personalizzare il testo inviato alla destinazione per gli eventi corrispondenti. Scegli una delle seguenti opzioni:
      - Eventi corrispondenti: EventBridge invia l'intero evento di origine originale alla destinazione. Questa è l'impostazione predefinita.
      - Parte degli eventi corrispondenti: EventBridge invia solo la parte specificata dell'evento di origine originale alla destinazione.

        In Specifica la parte dell'evento corrispondente, specifica un percorso JSON che definisce la parte dell'evento che EventBridge deve inviare alla destinazione.
      - Costante (testo JSON): EventBridge invia solo il testo JSON specificato alla destinazione. Non viene inviata alcuna parte dell'evento di origine originale.

In Specifica la costante in JSON, specifica il testo JSON che EventBridge deve inviare alla destinazione anziché all'evento.

- Trasformatore di input: configura un trasformatore di input per personalizzare il testo che EventBridge deve inviare alla destinazione. Per ulteriori informazioni, consulta ???.

  a. Seleziona Configura il trasformatore di input.

  b. Configura il trasformatore di input seguendo la procedura descritta in ???.

2. (Facoltativo) In Policy di ripetizione, specifica in che modo EventBridge deve ritentare l'invio di un evento a una destinazione dopo che si è verificato un errore.
   - Durata massima dell'evento: immetti il periodo di tempo massimo (in ore, minuti e secondi) durante il quale EventBridge deve mantenere gli eventi non elaborati. Il valore predefinito è 24 ore.
   - Nuovi tentativi: immetti il numero massimo di volte in cui EventBridge deve riprovare a inviare un evento alla destinazione dopo che si è verificato un errore. L'impostazione predefinita è 185 volte.
3. (Facoltativo) In Coda DLQ scegli se utilizzare una coda Amazon SQS standard come coda DLQ. EventBridge invia eventi che corrispondono a questa regola alla coda DLQ se non vengono recapitati correttamente alla destinazione. Completa una delle seguenti operazioni:
   - Scegli Nessuna per non utilizzare una coda DLQ.
   - Scegli Seleziona una coda Amazon SQS nell'account AWS corrente da utilizzare come coda DLQ, quindi seleziona la coda da utilizzare dal menu a discesa.
   - Scegli Seleziona una coda Amazon SQS in un altro account AWS come coda DLQ e specifica l'ARN della coda da utilizzare. È necessario collegare alla coda una policy basata su risorse che concede l'autorizzazione EventBridge per l'invio di messaggi.

     Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.

AWS service

Tieni presente che EventBridge potrebbe non visualizzare tutti i seguenti campi per un determinato servizio AWS.

1. (Facoltativo) In Configura l'input di destinazione, scegli come personalizzare il testo inviato alla destinazione per gli eventi corrispondenti. Scegli una delle seguenti opzioni:

- Eventi corrispondenti: EventBridge invia l'intero evento di origine originale alla destinazione. Questa è l'impostazione predefinita.
- Parte degli eventi corrispondenti: EventBridge invia solo la parte specificata dell'evento di origine originale alla destinazione.

  In Specifica la parte dell'evento corrispondente, specifica un percorso JSON che definisce la parte dell'evento che EventBridge deve inviare alla destinazione.
- Costante (testo JSON): EventBridge invia solo il testo JSON specificato alla destinazione. Non viene inviata alcuna parte dell'evento di origine originale.

  In Specifica la costante in JSON, specifica il testo JSON che EventBridge deve inviare alla destinazione anziché all'evento.
- Trasformatore di input: configura un trasformatore di input per personalizzare il testo che EventBridge deve inviare alla destinazione. Per ulteriori informazioni, consulta ???.

  a. Seleziona Configura il trasformatore di input.

  b. Configura il trasformatore di input seguendo la procedura descritta in ???.

2. (Facoltativo) In Policy di ripetizione, specifica in che modo EventBridge deve ritentare l'invio di un evento a una destinazione dopo che si è verificato un errore.
   - Durata massima dell'evento: immetti il periodo di tempo massimo (in ore, minuti e secondi) durante il quale EventBridge deve mantenere gli eventi non elaborati. Il valore predefinito è 24 ore.
   - Nuovi tentativi: immetti il numero massimo di volte in cui EventBridge deve riprovare a inviare un evento alla destinazione dopo che si è verificato un errore. L'impostazione predefinita è 185 volte.
3. (Facoltativo) In Coda DLQ scegli se utilizzare una coda Amazon SQS standard come coda DLQ. EventBridge invia eventi che corrispondono a questa regola alla coda DLQ se non vengono recapitati correttamente alla destinazione. Completa una delle seguenti operazioni:
   - Scegli Nessuna per non utilizzare una coda DLQ.
   - Scegli Seleziona una coda Amazon SQS nell'account AWS corrente da utilizzare come coda DLQ, quindi seleziona la coda da utilizzare dal menu a discesa.
   - Scegli Seleziona una coda Amazon SQS in un altro account AWS come coda DLQ e specifica l'ARN della coda da utilizzare. È necessario collegare alla coda una policy basata su risorse che concede l'autorizzazione EventBridge per l'invio di messaggi.

Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.

4. (Facoltativo) Scegli Aggiungi un'altra destinazione per aggiungere un'altra destinazione per questa regola.
5. Scegli Successivo.

Configurazione di tag e revisione della regola

Infine, immetti i tag desiderati per la regola, quindi rivedi e crea la regola.

Per configurare i tag e rivedere e creare la regola

1. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta EventBridge Etichette Amazon.
2. Scegli Successivo.
3. Rivedi i dettagli della nuova regola. Per apportare modifiche a una qualsiasi sezione, scegli il pulsante Modifica accanto alla sezione in questione.

   Quando sei soddisfatto dei dettagli della regola, scegli Crea regola.

Riferimento alle espressioni Cron

Le espressioni Cron hanno sei campi obbligatori separati da uno spazio vuoto.

Sintassi

```
cron(fields)
```

Campo	Valori	Caratteri jolly
Minuti	0-59	, - * /
Ore	0-23	, - * /
Day-of-month (Giorno del mese)	1-31	, - * ? / L W
Mese	1-12 o JAN-DEC	, - * /

Campo	Valori	Caratteri jolly
Day-of-week (Giorno della settimana)	1-7 o SUN-SAT	, - * ? L #
Anno	1970-2199	, - * /

Caratteri jolly

- Il carattere jolly , (virgola) include valori aggiuntivi. Nel campo Month (Mese), JAN,FEB,MAR (GEN,FEB,MAR) include gennaio, febbraio e marzo.
- Il carattere jolly - (trattino) specifica gli intervalli. Nel campo Day (Giorno), 1-15 include i primi 15 giorni del mese specificato.
- Il carattere jolly * (asterisco) include tutti i valori nel campo. Nel campo Hours (Ore), * include ogni ora. Non è possibile utilizzare il carattere * nei campi Day-of-month (Giorno del mese) e Day-of-week (Giorno della settimana). Se viene utilizzato in uno di tali campi, è necessario utilizzare ? nell'altro.
- Il carattere jolly / (barra) specifica gli incrementi. Nel campo Minutes (Minuti), puoi inserire 1/10 per specificare ogni decimo minuto, a partire dal primo minuto dell'ora (ad esempio, l'11°, il 21° e il 31° minuto e così via).
- Il carattere jolly ? (punto interrogativo) specifica qualsiasi valore. Nel campo Day-of-month (Giorno del mese) puoi immettere 7 se un qualsiasi giorno della settimana è accettabile, puoi immettere ? nel campo Day-of-week (Giorno della settimana).
- Il carattere jolly L nel campo Day-of-month (Giorno del mese) o Day-of-week (Giorno della settimana) specifica l'ultimo giorno del mese o della settimana.
- Il carattere jolly **W** nel campo Day-of-month (Giorno del mese) specifica un giorno feriale. Nel campo Day-of-month (Giorno del mese), **3W** specifica il giorno più vicino al terzo giorno feriale del mese.
- Il carattere jolly # nel campo Day-of-week (Giorno della settimana) specifica una determinata istanza del giorno della settimana specificato in un mese. Ad esempio, 3#2 sarebbe il secondo martedì del mese: il 3 fa riferimento a martedì perché è il terzo giorno di ogni settimana e il 2 fa riferimento al secondo giorno di questo tipo in un mese.

> **Note**
>
> Se utilizzi un carattere "#", puoi definire una sola espressione nel campo Day-of-week (Giorno della settimana). Ad esempio, `"3#1,6#3"` non è valido perché viene interpretato come due espressioni.

Limitazioni

- Non puoi specificare i campi Day-of-month (Giorno del mese) e Day-of-week (Giorno della settimana) nella stessa espressione Cron. Se specifichi un valore o * (asterisco) in uno dei campi, devi usare un carattere ? (punto interrogativo) nell'altro campo.
- Le espressioni Cron che indicano frequenze più rapide di 1 minuto non sono supportate.

Esempi

Quando crei una regola con pianificazione puoi utilizzare le seguenti stringhe Cron di esempio.

Minuti	Ore	Giorno del mese	Mese	Giorno della settimana	Anno	Significato
0	10	*	*	?	*	Esegui ogni giorno alle 10:00 (UTC+0)
15	12	*	*	?	*	Esegui ogni giorno alle 12:15 (UTC+0)
0	18	?	*	LUN-VEN	*	Esegui dal lunedì al venerdì

Minuti	Ore	Giorno del mese	Mese	Giorno della settimana	Anno	Significato
						alle 18:00 (UTC+0)
0	8	1	*	?	*	Esegui ogni primo giorno del mese alle 8:00 (UTC+0)
0/15	*	*	*	?	*	Esegui ogni 15 minuti
0/10	*	?	*	LUN-VEN	*	Esegui dal lunedì al venerdì ogni 10 minuti
0/5	8-17	?	*	LUN-VEN	*	Esegui dal lunedì al venerdì dalle 8:00 alle 17:55 (UTC+0) ogni 5 minuti

Minuti	Ore	Giorno del mese	Mese	Giorno della settimana	Anno	Significato
0/30	20-2	?	*	LUN-VEN	*	Esegui ogni 30 minuti dal lunedì al venerdì tra le 22:00 del giorno iniziale e le 2:00 del giorno successivo (UTC) Esegui dalle 00:00 alle 2:00 del lunedì mattina (UTC).

L'esempio seguente crea una regola che viene eseguita ogni giorno alle 12:00 UTC+0.

```
aws events put-rule --schedule-expression "cron(0 12 * * ? *)" --name MyRule1
```

L'esempio seguente crea una regola che viene eseguita ogni giorno alle 14:05 e alla 14:35 UTC+0.

```
aws events put-rule --schedule-expression "cron(5,35 14 * * ? *)" --name MyRule2
```

L'esempio successivo crea una regola che viene eseguita alle 10:15 UTC+0 l'ultimo venerdì di ogni mese dal 2019 al 2022.

```
aws events put-rule --schedule-expression "cron(15 10 ? * 6L 2019-2022)" --name MyRule3
```

Riferimento alle espressioni della frequenza

Un'espressione della frequenza inizia quando crei la regola di evento pianificata e successivamente la esegui in base a una pianificazione definita.

Le espressioni della frequenza hanno due campi obbligatori separati da uno spazio vuoto.

Sintassi

```
rate(value unit)
```

value

Un numero positivo.

unità

L'unità di tempo. Per i valori di 1, ad esempio `minute`, e i valori maggiori di 1, ad esempio `minutes`, sono necessarie unità diverse.

Valori validi: minuto | minuti | ora | ore | giorno | giorni

Limitazioni

Se il valore è uguale a 1, l'unità deve essere al singolare. Se il valore è superiore a 1, l'unità deve essere al plurale. Ad esempio, rate(1 ore) e rate(5 ora) non sono valide, ma rate(1 ora) e rate(5 ore) sono valide.

Esempi

Gli esempi seguenti mostrano come utilizzare le espressioni della frequenza con il comando `put-rule` dell'AWS CLI. Il primo esempio attiva la regola ogni minuto, quello successivo la attiva ogni cinque minuti, il terzo la attiva una volta all'ora e l'ultimo una volta al giorno.

```
aws events put-rule --schedule-expression "rate(1 minute)" --name MyRule2
```

```
aws events put-rule --schedule-expression "rate(5 minutes)" --name MyRule3
```

```
aws events put-rule --schedule-expression "rate(1 hour)" --name MyRule4
```

```
aws events put-rule --schedule-expression "rate(1 day)" --name MyRule5
```

Disabilitazione o eliminazione di una regola Amazon EventBridge

Per impedire a una regola di elaborare eventi o di essere eseguita in base a una pianificazione, puoi eliminare o disabilitare la regola. I passaggi seguenti illustrano come eliminare o disabilitare una regola EventBridge.

Per eliminare o disabilitare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).

   In Event bus (Bus di eventi), selezionare il bus di eventi associato alla regola.
3. Completa una delle seguenti operazioni:

   a. Per eliminare una regola, seleziona il pulsante accanto alla regola e seleziona Actions (Operazioni), Delete (Elimina), Delete (Elimina).

      Se la regola è una regola gestita, immetti il nome della regola per riconoscere che si tratta di una regola gestita e che l'eliminazione della stessa può causare l'arresto delle funzionalità nel servizio che ha creato la regola. Per continuare, digitare il nome della regola e scegliere Force delete (Forza eliminazione).

   b. Per disattivare temporaneamente una regola, selezionare il pulsante accanto alla regola, quindi selezionare Actions (Operazioni), Disable (Disattiva).

      Non è possibile disabilitare una regola gestita.

Best practice per la definizione di regole Amazon EventBridge

Di seguito sono riportate alcune best practice da prendere in considerazione quando crei regole per i tuoi router di eventi.

Impostazione di un'unica destinazione per ogni regola

Sebbene sia possibile specificare fino a cinque destinazioni per una determinata regola, la gestione delle regole è più semplice se si specifica una singola destinazione per ogni regola. Se più di una regola deve ricevere lo stesso set di eventi, consigliamo di duplicare la regola per distribuire gli stessi eventi a destinazioni diverse. Questo incapsulamento semplifica la gestione delle regole: se le

esigenze delle destinazioni degli eventi divergono nel tempo, puoi aggiornare ogni regola e il relativo modello di eventi indipendentemente dalle altre.

Impostazione delle autorizzazioni delle regole

È possibile consentire ai componenti o ai servizi delle applicazioni che utilizzano eventi di avere il controllo della gestione delle proprie regole. Un approccio architetturale comune adottato dai clienti consiste nell'isolare questi componenti o servizi delle applicazioni applicativi utilizzando account AWS separati. Per abilitare il flusso di eventi da un account all'altro, devi creare una regola in un router di eventi che instradi gli eventi a un router di eventi in un altro account. È possibile consentire ai team o a i servizi che utilizzano eventi di avere il controllo della gestione delle proprie regole. A tale scopo, devi specificare le autorizzazioni appropriate per i relativi account tramite policy basate su risorse. Ciò vale per account e Regioni.

Per ulteriori informazioni, consulta ???.

Per un esempio delle policy basate su risorse, consulta Multi-account design patterns with Amazon EventBridge su GitHub.

Monitoraggio delle prestazioni delle regole

Monitora le tue regole per assicurarti che funzionino come previsto:

- Il monitoraggio della metrica `TriggeredRules` per punti dati mancanti o anomalie può assisterti nel rilevare discrepanze per un editore che ha apportato una modifica sostanziale. Per ulteriori informazioni, consulta ???.
- Gli allarmi sulle anomalie o sul conteggio massimo previsto possono anche aiutare a rilevare quando una regola corrisponde a nuovi eventi. Ciò può accadere quando gli editori di eventi, inclusi servizi AWS e partner SaaS, introducono nuovi eventi abilitando nuovi casi d'uso e funzionalità. Quando questi nuovi eventi sono imprevisti e generano un volume superiore alla velocità di elaborazione della destinazione a valle, possono causare un backlog degli eventi.

  Tale elaborazione di eventi imprevisti può anche comportare addebiti di fatturazione indesiderati.

  Può inoltre attivare una limitazione delle regole quando l'account supera la quota aggregata relativa alle invocazioni di destinazione al secondo. EventBridge tenterà comunque di distribuire gli eventi che corrispondono a regole limitate ed effettuerà nuovi tentativi per un periodo fino a 24 ore o come descritto nella policy di ripetizione personalizzata della destinazione. Puoi rilevare le regole limitate e impostare allarmi per le stesse utilizzando la metrica `ThrottledRules`.

- Per i casi d'uso a bassa latenza, puoi anche monitorare la latenza utilizzando `IngestionToInvocationStartLatency`, che fornisce un'indicazione dell'integrità del tuo router di eventi. Qualsiasi periodo prolungato di latenza elevata superiore a 30 secondi può indicare un'interruzione del servizio o una limitazione delle regole.

Utilizzo di Amazon EventBridge e di modelli AWS Serverless Application Model

Puoi creare e testare manualmente le regole nella console EventBridge, il che può aiutarti nel processo di sviluppo durante il perfezionamento di modelli di eventi. Tuttavia, quando sei pronto per distribuire l'applicazione, è più semplice utilizzare un framework come AWS SAM per avviare tutte le risorse serverless in modo coerente.

Useremo questa applicazione di esempio per esaminare i modi in cui è possibile utilizzare i modelli AWS SAM per creare risorse EventBridge. Il file template.yaml in questo esempio è un modello AWS SAM che definisce quattro funzioni AWS Lambda e mostra due modi diversi di integrare le funzioni Lambda in EventBridge.

Per una procedura guidata di questa applicazione di esempio, consulta ???.

Esistono due approcci all'utilizzo di EventBridge e dei modelli AWS SAM. Per integrazioni semplici in cui una funzione Lambda viene richiamata da una regola, si consiglia l'approccio Modello combinato. Se utilizzi una logica di routing complessa o ti connetti a risorse esterne al modello AWS SAM, l'approccio Modello separato è la scelta migliore.

Approcci:

- Modello combinato
- Modello separato

Modello combinato

Il primo approccio utilizza la proprietà `Events` per configurare la regola EventBridge. Il codice di esempio seguente definisce un evento che richiama la funzione Lambda.

> **Note**
>
> Questo esempio crea automaticamente la regola nel router di eventi predefinito, presente in ogni account AWS. Per associare la regola a un router di eventi personalizzato, puoi aggiungere `EventBusName` al modello.

```
atmConsumerCase3Fn:
```

```
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: atmConsumer/
    Handler: handler.case3Handler
    Runtime: nodejs12.x
    Events:
      Trigger:
        Type: CloudWatchEvent
        Properties:
          Pattern:
            source:
              - custom.myATMapp
            detail-type:
              - transaction
            detail:
              result:
                - "anything-but": "approved"
```

Questo codice YAML equivale a un modello di eventi nella console EventBridge. In YAML, è sufficiente definire il modello di eventi e AWS SAM crea automaticamente un ruolo IAM con le autorizzazioni necessarie.

Modello separato

Nel secondo approccio alla definizione di una configurazione EventBridge in AWS SAM, le risorse vengono separate più chiaramente nel modello.

1. Innanzitutto, definisci la funzione Lambda:

   ```
   atmConsumerCase1Fn:
     Type: AWS::Serverless::Function
     Properties:
       CodeUri: atmConsumer/
       Handler: handler.case1Handler
       Runtime: nodejs12.x
   ```

2. Successivamente, definisci la regola utilizzando una risorsa `AWS::Events::Rule`. Le proprietà definiscono il modello di eventi e possono anche specificare le destinazioni. È possibile definire in modo esplicito molteplici destinazioni.

   ```
   EventRuleCase1:
     Type: AWS::Events::Rule
   ```

```
  Properties:
    Description: "Approved transactions"
    EventPattern:
      source:
        - "custom.myATMapp"
      detail-type:
        - transaction
      detail:
        result:
          - "approved"
    State: "ENABLED"
    Targets:
      -
        Arn:
          Fn::GetAtt:
            - "atmConsumerCase1Fn"
            - "Arn"
        Id: "atmConsumerTarget1"
```

3. Infine, definisci una risorsa `AWS::Lambda::Permission` che concede a EventBridge l'autorizzazione per richiamare la destinazione.

```
PermissionForEventsToInvokeLambda:
  Type: AWS::Lambda::Permission
  Properties:
    FunctionName:
      Ref: "atmConsumerCase1Fn"
    Action: "lambda:InvokeFunction"
    Principal: "events.amazonaws.com"
    SourceArn:
      Fn::GetAtt:
        - "EventRuleCase1"
        - "Arn"
```

Generazione di un modello AWS CloudFormation da regole Amazon EventBridge

AWS CloudFormation consente di configurare e gestire le risorse AWS tra account e regioni in modo centralizzato e ripetibile trattando l'infrastruttura come codice. CloudFormation consente di creare modelli che definiscono le risorse da fornire e gestire.

EventBridge ti consente di generare modelli da regole esistenti nel tuo account, per aiutarti a iniziare immediatamente lo sviluppo di modelli CloudFormation. Puoi selezionare una singola regola o più regole da includere nel modello. Puoi quindi utilizzare quei modelli come base per creare stack di risorse mediante la gestione con CloudFormation.

Per ulteriori informazioni su CloudFormation, consulta la Guida per l'utente di AWS CloudFormation.

> **Note**
>
> EventBridge non include regole gestite nel modello generato.

Puoi anche generare un modello da un router di eventi esistente, incluse le regole contenute nel router di eventi.

Per generare un modello AWS CloudFormation da una o più regole

1. Apri la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. In Seleziona bus di eventi, scegli il router di eventi che contiene le regole da includere nel modello.
4. In Regole, scegli le regole da includere nel modello AWS CloudFormation generato.

   Per una singola regola, puoi anche scegliere il nome della regola per visualizzare la pagina dei dettagli della regola.
5. Scegli Modello CloudFormation, quindi scegli il formato in cui EventBridge deve generare il modello, ovvero JSON o YAML.

   EventBridge visualizza il modello, generato nel formato selezionato.
6. EventBridge ti offre la possibilità di scaricare il file di modello o di copiare il modello negli appunti.
   - Per scaricare il file di modello, scegli Scarica.
   - Per copiare il modello negli appunti, scegli Copia.
7. Per uscire dal modello, scegli Annulla.

Dopo aver personalizzato il modello di AWS CloudFormation come necessario per il tuo caso d'uso, puoi utilizzarlo per creare stack in AWS CloudFormation.

Considerazioni sull'utilizzo di modelli CloudFormation generati da Amazon EventBridge

Prendi in considerazione i seguenti fattori quando utilizzi un modello CloudFormation generato da EventBridge:

- EventBridge non include alcuna password nel modello generato.

  Puoi modificare il modello per includere parametri del modello che consentono agli utenti di specificare password o altre informazioni riservate quando utilizzano il modello per creare o aggiornare uno stack CloudFormation.

  Inoltre, gli utenti possono utilizzare Secrets Manager per creare un segreto nella Regione desiderata e quindi modificare il modello generato per utilizzare parametri dinamici.
- Le destinazioni nel modello generato rimangono esattamente come specificate nel router di eventi originale. Se il modello non viene modificato in modo appropriato prima di utilizzarlo per creare stack in altre Regioni, è possibile che si abbiano problemi in più Regioni.

  Inoltre, il modello generato non creerà automaticamente destinazioni a valle.

EventBridge Obiettivi di Amazon

Un target è una risorsa o un endpoint che EventBridge invia un evento quando l'evento corrisponde al modello di evento definito per una regola. La regola elabora i dati dell'evento e invia le informazioni pertinenti alla destinazione. Per inviare i dati degli eventi a una destinazione, è EventBridge necessaria l'autorizzazione per accedere alla risorsa di destinazione. Puoi definire fino a cinque destinazioni per ciascuna regola.

Quando si aggiungono destinazioni a una regola e la regola viene eseguita subito dopo, le destinazioni nuove o aggiornate potrebbero non essere richiamate immediatamente. È necessario un breve periodo di tempo affinché vengano applicate le modifiche.

Il video seguente illustra le nozioni di base sulle destinazioni: What is a target

Obiettivi disponibili nella EventBridge console

È possibile configurare le seguenti destinazioni per gli eventi nella EventBridge console:

- Destinazione API
- API Gateway
- AWS AppSync;
- Coda di processi batch
- CloudWatch gruppo di log
- CodeBuild progetto
- CodePipeline
- Chiamata API `CreateSnapshot` di Amazon EBS
- EC2 Image Builder
- Chiamata API `RebootInstances` di EC2
- Chiamata API `StopInstances` di EC2
- Chiamata API `TerminateInstances` di EC2
- Attività ECS
- Bus di eventi in un altro account o Regione

- Bus di eventi nello stesso account e nella stessa Regione
- Flussi di distribuzione Firehose
- Workflow di Glue
- Piano di risposta dello strumento di gestione degli incidenti
- Modello di valutazione di Inspector
- Flusso di Kinesis
- Funzione Lambda (ASYNC)
- Query sull'API dati del cluster Amazon Redshift
- Query sull'API dati del gruppo di lavoro Amazon Redshift serverless
- SageMaker Gasdotto
- Argomento Amazon SNS

  EventBridge non supporta gli argomenti Amazon SNS FIFO (first in, first out).
- Coda Amazon SQS
- Macchine a stati di Step Functions (ASYNC)
- Systems Manager Automation
- Systems Manager OpsItem
- Run Command di Systems Manager

Parametri di destinazione

Alcune destinazioni non inviano le informazioni nel payload dell'evento alla destinazione, ma trattano l'evento come un trigger per richiamare un'API specifica. EventBridge utilizza i parametri Target per determinare cosa succede con quell'obiettivo. Questi sono i seguenti:

- Destinazioni API: i dati inviati a una destinazione API devono corrispondere alla struttura dell'API. È necessario utilizzare l'oggetto `InputTransformer` per assicurarsi che i dati siano strutturati correttamente. Se vuoi includere il payload dell'evento originale, fai riferimento a esso in `InputTransformer`.
- Gateway API: i dati inviati a Gateway API devono corrispondere alla struttura dell'API. È necessario utilizzare l'oggetto `InputTransformer` per assicurarsi che i dati siano strutturati correttamente. Se vuoi includere il payload dell'evento originale, fai riferimento a esso in `InputTransformer`.

- Amazon EC2 Image Builder
- `RedshiftDataParameters` (cluster delle API dati di Amazon Redshift)
- `SageMakerPipelineParameters`(Pipeline di creazione SageMaker di modelli Amazon Runtime)

> **Note**
>
> EventBridge non supporta tutta la sintassi di JSON Path e la valuta in fase di esecuzione. La sintassi supportata include:
>
> - notazione a punti (ad esempio, `$.detail`)
> - trattini
> - caratteri di sottolineatura
> - caratteri alfanumerici
> - indici array
> - caratteri jolly (*)

Parametri di percorso dinamici

Alcuni parametri di destinazione supportano la sintassi di percorso JSON dinamico facoltativa. Questa sintassi consente di specificare percorsi JSON anziché valori statici (ad esempio `$.detail.state`). L'intero valore deve essere un percorso JSON, non solo una parte di esso. Ad esempio, `RedshiftParameters.Sql` può essere `$.detail.state` ma non può essere `"SELECT * FROM $.detail.state"`. Questi percorsi vengono sostituiti dinamicamente al runtime con i dati del payload di eventi nel percorso specificato. I parametri di percorso dinamici non possono fare riferimento a valori nuovi o trasformati risultanti dalla trasformazione dell'input. La sintassi supportata per i percorsi JSON dei parametri dinamici è la stessa utilizzata per la trasformazione dell'input. Per ulteriori informazioni, consulta ???

La sintassi dinamica può essere utilizzata in tutti i campi stringhe non enum di questi parametri:

- `EcsParameters`
- `HttpParameters` (tranne le chiavi `HeaderParameters`)
- `RedshiftDataParameters`
- `SageMakerPipelineParameters`

Autorizzazioni

Per effettuare chiamate API sulle risorse di tua proprietà, è EventBridge necessaria l'autorizzazione appropriata. Per AWS Lambda le risorse Amazon SNS, EventBridge utilizza politiche basate sulle risorse. Per le istanze EC2, i flussi di dati Kinesis e le macchine a stati Step Functions, EventBridge utilizza i ruoli IAM specificati nel parametro in. `RoleARN` `PutTargets` Puoi richiamare un endpoint Gateway API con l'autorizzazione IAM configurata, ma il ruolo è facoltativo se non hai configurato l'autorizzazione. Per ulteriori informazioni, consulta Amazon EventBridge e AWS Identity and Access Management.

Se un altro account si trova nella stessa Regione e ti ha concesso l'autorizzazione, puoi inviare eventi a quell'account. Per ulteriori informazioni, consulta Invio e ricezione di EventBridge eventi Amazon tra AWS account.

Se la destinazione è crittografata, devi includere la sezione seguente nella policy della chiave KMS.

```
{
    "Sid": "Allow EventBridge to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": "events.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*"
}
```

EventBridge specifiche del bersaglio

AWS Batch code di lavoro

Alcuni parametri AWS Batch `submitJob` possono essere configurati tramite BatchParameters.

Altri possono essere specificati nel payload di eventi. Se il payload dell'evento (trasmesso o trasmesso InputTransformers) contiene le seguenti chiavi, queste vengono mappate in base ai parametri di `submitJob` richiesta:

- `ContainerOverrides: containerOverrides`

> **Note**
>
> Include solo comando, ambiente, memoria e vcpu

- `DependsOn: dependsOn`

> **Note**
>
> Include solo jobId

- `Parameters: parameters`

CloudWatch Gruppo di log

Se non si utilizza un oggetto InputTransformercon un obiettivo CloudWatch Logs, il payload dell'evento viene utilizzato come messaggio di registro e l'origine dell'evento come timestamp. Se si utilizza un InputTransformer, il modello deve essere:

```
{"timestamp":<timestamp>,"message":<message>}
```

EventBridge raggruppa in batch le voci inviate a un flusso di log; pertanto, EventBridge può inviare uno o più eventi a un flusso di log, a seconda del traffico.

CodeBuild progetto

Se si utilizza InputTransformersper modellare l'evento di input su un Target in modo che corrisponda alla CodeBuild StartBuildRequeststruttura, i parametri verranno mappati 1 a 1 e passati a. `codeBuild.StartBuild`

Processo di Amazon ECS

Se lo utilizzi InputTransformersper modellare l'evento di input su un Target in modo che corrisponda alla RunTask TaskOverridestruttura di Amazon ECS, i parametri verranno mappati 1 a 1 e passati a. `ecs.RunTask`

Piano di risposta dello strumento di gestione degli incidenti

Se l'evento corrispondente proviene da CloudWatch Alarms, i dettagli della modifica dello stato dell'allarme vengono inseriti nei dettagli del trigger della chiamata a Incident Manager. StartIncidentRequest

Configurazione di destinazioni

Scopri come configurare le impostazioni per gli obiettivi. EventBridge

Destinazioni:

- Destinazioni API
- EventBridge Obiettivi di Amazon per Amazon API Gateway
- AWS AppSync obiettivi per Amazon EventBridge
- Connessioni per destinazioni endpoint HTTP
- Invio e ricezione di EventBridge eventi Amazon tra AWS account
- Invio e ricezione di EventBridge eventi Amazon tra AWS regioni
- Invio e ricezione di EventBridge eventi Amazon tra bus di eventi nello stesso account e nella stessa regione

Destinazioni API

Le destinazioni EventBridge API di Amazon sono endpoint HTTP che puoi richiamare come destinazione di una regola, in modo simile a come richiami un AWS servizio o una risorsa come destinazione. Utilizzando le destinazioni API, è possibile instradare eventi tra AWS servizi, applicazioni SaaS (Software as a Service) integrate e applicazioni esterne AWS utilizzando chiamate API. Quando specificate una destinazione API come destinazione di una regola, EventBridge richiama l'endpoint HTTP per qualsiasi evento che corrisponde al modello di evento specificato nella regola e quindi fornisce le informazioni sull'evento con la richiesta. Con EventBridge, puoi utilizzare qualsiasi metodo HTTP tranne CONNECT e TRACE per la richiesta. I metodi HTTP più comuni da utilizzare sono PUT e POST. È inoltre possibile utilizzare trasformatori di input per personalizzare l'evento in base ai parametri di uno specifico endpoint HTTP. Per ulteriori informazioni, consulta Trasformazione degli EventBridge input di Amazon.

> **Note**
>
> Le destinazioni API non supportano destinazioni private, come gli endpoint VPC di interfaccia, incluse le API HTTPS private nei Virtual Private Clouds (VPC) che utilizzano Network e Application Load Balancer privati e endpoint VPC di interfaccia.
> Per ulteriori informazioni, consulta ???.

> **Important**
>
> EventBridge le richieste verso un endpoint di destinazione API devono avere un timeout di esecuzione client massimo di 5 secondi. Se l'endpoint di destinazione impiega più di 5 secondi per rispondere, la richiesta scade in EventBridge timeout. EventBridge i nuovi tentativi hanno determinato il timeout delle richieste fino ai valori massimi configurati nella politica di ripetizione dei tentativi. Per impostazione predefinita, i valori massimi sono 24 ore e 185 volte. Dopo l'esecuzione del numero massimo di tentativi, gli eventi vengono inviati alla coda DLQ se esistente. In caso contrario, l'evento viene abbandonato.

Il video seguente illustra l'uso della destinazione API: Using API destinations

In questo argomento:

- Creazione di una destinazione API
- Creazione di regole che inviano eventi a una destinazione API
- Ruolo collegato a un servizio per le destinazioni API
- Intestazioni incluse nelle richieste a destinazioni API
- Codici di errore delle destinazioni API
- Impatto della frequenza di invocazione sulla distribuzione degli eventi
- Invio di CloudEvents eventi a destinazioni API
- Partner di destinazione API

Creazione di una destinazione API

Ogni destinazione API richiede una connessione. Una connessione specifica il tipo di autorizzazione e le credenziali da utilizzare per l'autorizzazione con l'endpoint di destinazione API. Puoi scegliere una connessione esistente o creare una connessione quando crei una destinazione API. Per ulteriori informazioni, consulta ???

Per creare una destinazione API utilizzando la console EventBridge

1. Accedi AWS utilizzando un account con le autorizzazioni necessarie per gestire EventBridge e aprire la EventBridgeconsole.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scorri verso il basso fino alla tabella delle destinazioni API, quindi scegli Crea una destinazione API.
4. Nella pagina Crea una destinazione API, in Nome immetti un nome per la destinazione API. Puoi utilizzare fino a 64 caratteri maiuscoli o minuscoli, numeri, punti (.), trattini (-) o caratteri di sottolineatura (_).

   Il nome deve essere univoco per l'account nella Regione corrente.
5. In Descrizione, immetti una descrizione per la destinazione API.
6. Immetti un Endpoint di destinazione API per la destinazione API. L'endpoint di destinazione API è una destinazione endpoint di invocazione HTTP per gli eventi. Le informazioni di autorizzazione che includi nella connessione utilizzata per questa destinazione API vengono utilizzate per l'autorizzazione in base a questo endpoint. L'URL deve utilizzare HTTPS.
7. Immetti il Metodo HTTP da utilizzare per la connessione all'Endpoint di destinazione API.

8. (Facoltativo) In Limite di velocità di invocazione al secondo, immetti il numero massimo di invocazioni al secondo da inviare all'endpoint di destinazione API.

   Il limite di frequenza impostato può influire sulla modalità di EventBridge erogazione degli eventi. Per ulteriori informazioni, consulta Impatto della frequenza di invocazione sulla distribuzione degli eventi.
9. In Connessione, esegui una delle seguenti operazioni:
   - Scegli Utilizza una connessione esistente, quindi seleziona la connessione da utilizzare per questa destinazione API.
   - Scegli Crea una nuova connessione, quindi immetti i dettagli della connessione da creare. Per ulteriori informazioni, consulta Connessioni.
10. Scegli Crea.

Creazione di regole che inviano eventi a una destinazione API

Dopo aver creato una destinazione API, puoi selezionarla come destinazione di una regola. Per utilizzare una destinazione API come destinazione, devi fornire le autorizzazioni appropriate a un ruolo IAM. Per ulteriori informazioni, consulta ???

La selezione di una destinazione API come destinazione fa parte della creazione della regola.

Per creare una regola che invii eventi a una destinazione API utilizzando la console

1. Segui i passaggi nella procedura ???.
2. Nella ??? fase, quando viene richiesto di scegliere una destinazione API come tipo di destinazione:
   a. Seleziona la destinazione EventBridge dell'API.
   b. Esegui una di queste operazioni:
      - Scegli Usa una destinazione API esistente e seleziona una destinazione API esistente
      - Scegli Crea una nuova destinazione API e specifica l'impostazione necessaria per definire la tua nuova destinazione API.

        Per ulteriori informazioni sulla specificazione delle impostazioni richieste, consulta???.
   c. (Facoltativo): per specificare i parametri di intestazione per l'evento, in Parametri dell'intestazione scegliete Aggiungi parametro di intestazione.

Quindi, specificate la chiave e il valore per il parametro di intestazione.

d. (Facoltativo): per specificare i parametri della stringa di query per l'evento, in Parametri della stringa di query scegliete Aggiungi parametro della stringa di query.

   Quindi, specificate la chiave e il valore per il parametro della stringa di query.

3. Completa la creazione della regola seguendo i passaggi della procedura.

Ruolo collegato a un servizio per le destinazioni API

Quando crei una connessione per una destinazione API, al tuo account AWS ServiceRoleForAmazonEventBridgeApiDestinationsviene aggiunto un ruolo collegato al servizio denominato. EventBridge utilizza il ruolo collegato al servizio per creare e archiviare un segreto in Secrets Manager. Per concedere le autorizzazioni necessarie al ruolo collegato al servizio, associa la policy al ruolo EventBridge. AmazonEventBridgeApiDestinationsServiceRolePolicy La policy limita le autorizzazioni concesse solo a quelle necessarie affinché il ruolo interagisca con il segreto della connessione. Non sono incluse altre autorizzazioni e il ruolo può interagire solo con le connessioni presenti nell'account per la gestione del segreto.

La policy seguente è `AmazonEventBridgeApiDestinationsServiceRolePolicy`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:UpdateSecret",
                "secretsmanager:DescribeSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:events!connection/*"
        }
    ]
}
```

Per ulteriori informazioni sui ruoli collegati ai servizi, consulta Utilizzo di ruoli collegati ai servizi nella documentazione IAM.

Il ruolo `AmazonEventBridgeApiDestinationsServiceRolePolicy` collegato al servizio è supportato nelle seguenti regioni: AWS

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacific (Hong Kong)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Stoccolma)
- Sud America (San Paolo)
- Cina (Ningxia)
- Cina (Pechino)

Intestazioni incluse nelle richieste a destinazioni API

La sezione seguente descrive in dettaglio come EventBridge gestisce le intestazioni HTTP nelle richieste alle destinazioni API.

Intestazioni incluse nelle richieste a destinazioni API

Oltre alle intestazioni di autorizzazione definite per la connessione utilizzata per una destinazione API, EventBridge include le seguenti intestazioni in ogni richiesta.

Chiave intestazione	Valore intestazione
User-Agent	Amazon//EventBridgeApiDestinations
Content-Type	Se non viene specificato alcun valore Content-Type personalizzato, EventBridge include il seguente valore predefinito come Content-Type: application/json; charset=utf-8
Intervallo	bytes=0-1048575
Accept-Encoding	gzip,deflate
Connessione	close
Content-Length	Un'intestazione di entità che indica la dimensione del corpo dell'entità, in byte, inviata al destinatario.
Host	Un'intestazione di richiesta che specifica l'host e il numero di porta del server a cui viene inviata la richiesta.

Intestazioni che non possono essere sovrascritte nelle richieste verso destinazioni API

EventBridge non consente di sovrascrivere le seguenti intestazioni:

- User-Agent
- Intervallo

Le intestazioni vengono EventBridge rimosse dalle richieste alle destinazioni API

EventBridge rimuove le seguenti intestazioni per tutte le richieste di destinazione API:

- A-IM
- Accept-Charset
- Accept-Datetime
- Accept-Encoding
- Cache-Control
- Connessione
- Content-Encoding
- Content-Length
- Content-MD5
- Data
- Expect
- Forwarded
- Da
- Host
- sImpostazioni HTTP2
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Origin
- Pragma
- Proxy-Authorization
- Intervallo
- Referente
- TE
- Trailer

- Transfer-Encoding
- User-Agent
- Upgrade
- Via
- Attenzione

Codici di errore delle destinazioni API

Quando EventBridge tenta di inviare un evento a una destinazione API e si verifica un errore, EventBridge effettua le seguenti operazioni:

- Gli eventi associati ai codici di errore 409, 429 e 5xx vengono ritentati.
- Gli eventi associati ai codici di errore 1xx, 2xx, 3xx e 4xx (escluso 429) non vengono ritentati.

EventBridge Le destinazioni API leggono l'intestazione di risposta HTTP standard `Retry-After` per scoprire quanto tempo attendere prima di effettuare una richiesta di follow-up. EventBridge sceglie il valore più conservativo tra la politica di riprova definita e l'intestazione. `Retry-After` Se `Retry-After` il valore è negativo, EventBridge interrompe il nuovo tentativo di consegna per quell'evento.

Impatto della frequenza di invocazione sulla distribuzione degli eventi

Se imposti la frequenza di invocazione al secondo su un valore molto inferiore al numero di invocazioni generate, gli eventi potrebbero non essere distribuiti entro il tempo di ripetizione di 24 ore per gli eventi. Ad esempio, se imposti la frequenza di invocazione su 10 chiamate al secondo, ma vengono generati migliaia di eventi al secondo, avrai rapidamente un backlog di eventi da distribuire che supera le 24 ore. Per essere certo che nessun evento vada perso, imposta una coda DLQ a cui inviare gli eventi con invocazioni non riuscite in modo da poterli elaborare in un secondo momento. Per ulteriori informazioni, consulta Utilizzo di code di lettere morte per elaborare gli eventi non consegnati.

Invio di CloudEvents eventi a destinazioni API

CloudEvents è una specifica indipendente dal fornitore per la formattazione degli eventi, con l'obiettivo di fornire l'interoperabilità tra servizi, piattaforme e sistemi. È possibile utilizzarlo EventBridge per trasformare gli eventi AWS di servizio CloudEvents prima che vengano inviati a una destinazione, ad esempio una destinazione API.

> **Note**
>
> La procedura seguente spiega come trasformare gli eventi di origine in modalità CloudEventsstrutturata. Nella CloudEvents specifica, un messaggio in modalità strutturata è un messaggio in cui l'intero evento (attributi e dati) viene codificato nel payload dell'evento.

Per ulteriori informazioni sulle specifiche, consulta cloudevents.io CloudEvents .

Per trasformare AWS gli eventi nel formato utilizzando la console CloudEvents

Per trasformare gli eventi nel CloudEvents formato precedente alla consegna a una destinazione, iniziate creando una regola del bus degli eventi. Come parte della definizione della regola, utilizzate un trasformatore di input per disporre degli eventi di EventBridge trasformazione prima di inviarli alla destinazione specificata.

1. Segui i passaggi nella procedura ???.
2. Nella ??? fase, quando viene richiesto di scegliere una destinazione API come tipo di destinazione:
   a. Seleziona la destinazione EventBridge dell'API.
   b. Esegui una di queste operazioni:
      - Scegli Usa una destinazione API esistente e seleziona una destinazione API esistente
      - Scegli Crea una nuova destinazione API e specifica l'impostazione necessaria per definire la tua nuova destinazione API.

      Per ulteriori informazioni sulla specificazione delle impostazioni richieste, consulta???.
   c. Specificate i parametri di intestazione Content-Type necessari per gli eventi: CloudEvents
      - In Parametri di intestazione scegli Aggiungi parametro di intestazione.
      - Per chiave, specifica. `Content-Type`

      Per valore, specificare`application/cloudevents+json; charset=UTF-8`.
3. Specificate un ruolo di esecuzione per il vostro obiettivo.
4. Definisci un trasformatore di input per trasformare i dati dell'evento di origine nel CloudEvents formato:

a. In Impostazioni aggiuntive, per Configura l'input di destinazione, scegli Trasformatore di ingresso.

   Quindi scegli Configura trasformatore di ingresso.

b. In Target input transformer, specifica il percorso di input.

   Nel percorso di input riportato di seguito, l'attributo region è un attributo di estensione personalizzato del CloudEvents formato. In quanto tale, non è necessario per il rispetto delle CloudEvents specifiche.

   CloudEvents consente di utilizzare e creare attributi di estensione non definiti nella specifica di base. Per ulteriori informazioni, incluso un elenco di attributi di estensione noti, vedete Attributi di CloudEvents estensione nella documentazione delle CloudEvents specifiche su GitHub.

```
{
  "detail": "$.detail",
  "detail-type": "$.detail-type",
  "id": "$.id",
  "region": "$.region",
  "source": "$.source",
  "time": "$.time"
}
```

c. Per Template, inserite il modello per trasformare i dati dell'evento di origine nel CloudEvents formato.

   Nel modello seguente, non `region` è strettamente obbligatorio, poiché l'`region`attributo nel percorso di input è un attributo di estensione della CloudEvents specifica.

```
{
  "specversion":"1.0",
  "id":<id>,
  "source":<source>,
  "type":<detail-type>,
  "time":<time>,
  "region":<region>,
  "data":<detail>
}
```

5. Completa la creazione della regola seguendo i passaggi della procedura.

Partner di destinazione API

Utilizza le informazioni fornite dai seguenti AWS partner per configurare una destinazione e una connessione API per il loro servizio o applicazione.

Osservabilità nel cloud di Cisco

URL dell'endpoint di invocazione della destinazione API:

```
https://tenantName.observe.appdynamics.com/rest/awsevents/aws-eventbridge-integration/endpoint
```

Tipi di autorizzazione supportati:

Credenziali del client OAuth

I token OAuth vengono aggiornati quando viene restituita una risposta 401 o 407

Parametri di autorizzazione aggiuntivi necessari:

Cisco Client ID AppDynamics e Client Secret

Endpoint OAuth:

```
https://tenantName.observe.appdynamics.com/auth/tenantId/default/oauth2/token
```

I seguenti parametri della coppia chiave/valore OAuth:

Type	Chiave	Valore
Campo corporeo	grant_type	client_credentials
Header	Content-Type	applicazione/x-www-form-urlencoded; set di caratteri = utf-8

AppDynamics Documentazione Cisco:

AWS ingestione di eventi

Operazioni API di uso comune:

Non applicabile

Informazioni aggiuntive:

Scegliendo Cisco AppDynamics dal menu a discesa Partner destination vengono precompilate le informazioni OAuth necessarie, incluse le coppie chiave/valore dell'intestazione e del corpo necessarie per le chiamate API.

Per ulteriori informazioni, consulta l'inserimento degli eventi nella documentazione di Cisco.AWS AppDynamics

Confluent

URL dell'endpoint di invocazione della destinazione API:

In genere il seguente formato:

`https://random-id.region.aws.confluent.cloud:443/kafka/v3/clusters/cluster-id/topics/topic-name/records`

Per ulteriori informazioni, consulta Trova l'indirizzo dell'endpoint REST e l'ID del cluster nella documentazione di Confluent.

Tipi di autorizzazione supportati:

Base

Parametri di autorizzazione aggiuntivi necessari:

Non applicabile

Documentazione Confluent:

Produrre dischi

Proxy REST Confluent per Apache Kafka

Operazioni API di uso comune:

`POST`

Informazioni aggiuntive:

Per trasformare i dati dell'evento in un messaggio che l'endpoint può elaborare, crea un trasformatore di input di destinazione.

- Per generare un record senza specificare una chiave di partizionamento Kafka, utilizzate il seguente modello per il trasformatore di input. Non è richiesto alcun percorso di input.

```
{
  "value":{
      "type":"JSON",
      "data":aws.events.event.json
  },
}
```

- Per generare un record utilizzando un campo di dati di eventi come chiave di partizionamento Kafka, segui il percorso di input e l'esempio di modello di seguito. Questo esempio definisce il percorso di input per il `orderId` campo e quindi specifica quel campo come chiave di partizione.

  Innanzitutto, definisci il percorso di input per il campo di dati dell'evento:

```
{
  "orderId":"$.detail.orderId"
}
```

  Quindi, usa il modello di trasformatore di input per specificare il campo dati come chiave di partizione:

```
{
  "value":{
    "type":"JSON",
    "data":aws.events.event.json
  },
  "key":{
    "data":"<orderId>",
    "type":"STRING"
  }
}
```

Coralogix

URL dell'endpoint di invocazione della destinazione API

Per un elenco completo degli endpoint, consulta Coralogix API Reference.

Tipi di autorizzazione supportati

Chiave API

Parametri di autorizzazione aggiuntivi necessari

Intestazione `"x-amz-event-bridge-access-key"`, il valore è la chiave API Coralogix

Documentazione di Coralogix

EventBridgeAutenticazione Amazon

Operazioni API di uso comune

Stati Uniti: https://ingress.coralogix.us/aws/event-bridge

Singapore: https://ingress.coralogixsg.com/aws/event-bridge

Irlanda: https://ingress.coralogix.com/aws/event-bridge

Stoccolma: https://ingress.eu2.coralogix.com/aws/event-bridge

India: https://ingress.coralogix.in/aws/event-bridge

Informazioni aggiuntive

Gli eventi vengono archiviati come voci di log con `applicationName=[AWS Account]` e `subsystemName=[event.source]`.

Datadog

URL dell'endpoint di invocazione della destinazione API

Per un elenco completo degli endpoint, consulta Datadog API Reference.

Tipi di autorizzazione supportati

Chiave API

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di Datadog

Autenticazione

Operazioni API di uso comune

POST https://api.datadoghq.com/api/v1/events

POST https://http-intake.logs.datadoghq.com/v1/input

Informazioni aggiuntive

Gli URL degli endpoint variano a seconda della posizione dell'organizzazione Datadog. Per l'URL corretto per la tua organizzazione, consulta la documentazione.

Freshworks

URL dell'endpoint di invocazione della destinazione API

Per un elenco degli endpoint, consulta https://developers.freshworks.com/documentation/

Tipi di autorizzazione supportati

Base, Chiave API

Parametri di autorizzazione aggiuntivi necessari

Non applicabile

Documentazione di Freshworks

Autenticazione

Operazioni API di uso comune

https://developers.freshdesk.com/api/#create_ticket

https://developers.freshdesk.com/api/#update_ticket

https://developer.freshsales.io/api/#create_lead

https://developer.freshsales.io/api/#update_lead

Informazioni aggiuntive

Nessuno

MongoDB

URL dell'endpoint di invocazione della destinazione API

https://data.mongodb-api.com/app/*ID app*/endpoint/

Tipi di autorizzazione supportati

Chiave API

E-mail/password

Autenticazione JWT personalizzata

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di MongoDB

Atlas Data API

Endpoints

Custom HTTPS Endpoints

Autenticazione

Operazioni API di uso comune

Nessuno

Informazioni aggiuntive

Nessuno

New Relic

URL dell'endpoint di invocazione della destinazione API

Per ulteriori informazioni, consulta Our EU and US region data centers.

Eventi

Stati Uniti: https://insights-collector.newrelic.com/v1/accounts/*YOUR_NEW_RELIC_ACCOUNT_ID* / events

UE: https://insights-collector.eu01.nr-data.net/v1/accounts/*YOUR_NEW_RELIC_ACCOUNT_ID*/events

Metriche

STATI UNITI: https://metric-api.newrelic.com/metric/v1

UE: https://metric-api.eu.newrelic.com/metric/v1

Log

STATI UNITI: https://log-api.newrelic.com/log/v1

UE: https://log-api.eu.newrelic.com/log/v1

Tracce

STATI UNITI: https://trace-api.newrelic.com/trace/v1

UE: https://trace-api.eu.newrelic.com/trace/v1

Tipi di autorizzazione supportati

Chiave API

Documentazione di New Relic

Metric API

Event API

Log API

Trace API

Operazioni API di uso comune

Metric API

Event API

Log API

Trace API

Informazioni aggiuntive

- Metric API limits
- Event API limits
- Log API limits
- Trace API limits

Operata

URL dell'endpoint di invocazione della destinazione API:

- https://api.operata.io/v2/aws/events/contact-record

Tipi di autorizzazione supportati:

- Base

Parametri di autorizzazione aggiuntivi necessari:

- Nessuno

Documentazione di Operata:

- How do I create, view, change and revoke API Tokens?
- AWS Integrazione di Operata tramite Amazon EventBridge Scheduler Pipes

Operazioni API di uso comune:

- `POST https://api.operata.io/v2/aws/events/contact-record`

Informazioni aggiuntive:

- `username` è l'ID gruppo Operata e la password è il tuo token API.

Salesforce

URL dell'endpoint di invocazione della destinazione API

- Oggetto: https://myDomainName.my.salesforce.com/services/data/ VersionNumber /sobjects*//* *SobjectEndpoint*

Eventi della piattaforma personalizzati: https://myDomainName.my.salesforce.com/services/data/*VersionNumber /sobjects/ /* customPlatformEndpoint*

Per un elenco completo di endpoint, consulta Riferimento API di Salesforce

Tipi di autorizzazione supportati

Credenziali del client OAuth

I token OAUTH vengono aggiornati quando viene restituita una risposta 401 o 407.

Parametri di autorizzazione aggiuntivi necessari

SalesforceID client e segreto client dell'app connessa .

Uno dei seguenti endpoint di autorizzazione:

- Produzione: https *MyDomainName*://.my.salesforce.com. /services/oauth2/token
- Sandbox senza domini avanzati— https://-- .my. salesforce.com/services /oauth2/token *MyDomainName SandboxName*
- Sandbox con domini avanzati— https://*MyDomainName*-- *SandboxName*.sandbox.my.salesforce.com/services/oauth2/token

La seguente coppia chiave/valore:

Key (Chiave)	Value (Valore)
grant_type	client_credentials

Documentazione di Salesforce

REST API Developer Guide

Operazioni API di uso comune

Working with Object Metadata

Working with Records

Informazioni aggiuntive

Per un tutorial che spiega come utilizzare la EventBridge console per creare una connessione versoSalesforce, una destinazione API e una regola a cui indirizzare Salesforce le informazioni, consulta. ???

Slack

URL dell'endpoint di invocazione della destinazione API

Per un elenco di endpoint e altre risorse, consulta Using the Slack Web API

Tipi di autorizzazione supportati

OAuth 2.0

I token OAUTH vengono aggiornati quando viene restituita una risposta 401 o 407.

Quando crei un'applicazione Slack e la installi nel tuo workspace, verrà creato automaticamente un token portatore OAuth da utilizzare per autenticare le chiamate tramite la tua connessione della destinazione API.

Parametri di autorizzazione aggiuntivi necessari

Non applicabile

Documentazione di Slack

Basic app setup

Installing with OAuth

Retrieving messages

Invio di messaggi

Sending messages using Incoming Webhooks

Operazioni API di uso comune

https://slack.com/api/chat.postMessage

Informazioni aggiuntive

Quando si configura la EventBridge regola, è necessario evidenziare due configurazioni:

- Includi un parametro di intestazione che definisca il tipo di contenuto come "application/json;charset=utf-8".
- Utilizza un trasformatore di input per mappare l'evento di input all'output previsto per l'API Slack, in particolare assicurati che il payload inviato all'API Slack abbia coppie chiave/valore "canale" e "testo".

Shopify

URL dell'endpoint di invocazione della destinazione API

Per un elenco di endpoint e altre risorse e metodi, consulta Endpoints and requests

Tipi di autorizzazione supportati

OAuth, Chiave API

> **Note**
>
> I token OAUTH vengono aggiornati quando viene restituita una risposta 401 o 407.

Parametri di autorizzazione aggiuntivi necessari

Non applicabile

Documentazione di Shopify

Authentication and authorization overview

Operazioni API di uso comune

POST - /admin/api/2022-01/products.json

GET - admin/api/2022-01/products/{product_id}.json

PUT - admin/api/2022-01/products/{product_id}.json

DELETE - admin/api/2022-01/products/{product_id}.json

Informazioni aggiuntive

Create an app

Consegna EventBridge tramite Amazon webhook

Access tokens for custom apps in the Shopify admin

Product

Shopify Admin API

Splunk

URL dell'endpoint di invocazione della destinazione API

https://*SPLUNK_HEC_ENDPOINT*:*optional_port*/services/collector/raw

Tipi di autorizzazione supportati

Base, Chiave API

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di Splunk

Per entrambi i tipi di autorizzazione, è necessario un ID token HEC. Per ulteriori informazioni, consulta Set up and use HTTP Event Collector in Splunk Web.

Operazioni API di uso comune

POST https://*SPLUNK_HEC_ENDPOINT*:*optional_port*/services/collector/raw

Informazioni aggiuntive

Chiave API: quando si configura l'endpoint per EventBridge, il nome della chiave API è «Autorizzazione» e il valore è l'ID del token Splunk HEC.

Basic (nome utente/password): quando si configura l'endpoint per EventBridge, il nome utente è «Splunk» e la password è l'ID del token Splunk HEC.

Sumo Logic

URL dell'endpoint di invocazione della destinazione API

Gli URL degli endpoint HTTP Log e Metric Source saranno diversi per ogni utente. Per ulteriori informazioni, consulta HTTP Logs and Metrics Source.

Tipi di autorizzazione supportati

Sumo Logic non richiede l'autenticazione per le relative origini HTTP perché nell'URL è presente una chiave univoca. Per questo motivo, devi assicurarti di trattare l'URL come segreto.

Quando si configura la destinazione dell' EventBridge API, è richiesto un tipo di autorizzazione. Per soddisfare questo requisito, seleziona la chiave API e assegnale il nome di chiave "dummy-key" e un valore di chiave "dummy-value".

Parametri di autorizzazione aggiuntivi necessari

Non applicabile

Documentazione di Sumo Logic

Sumo Logicha già creato sorgenti ospitate per raccogliere log e metriche da molti AWS servizi e puoi utilizzare le informazioni sul loro sito Web per lavorare con tali fonti. Per ulteriori informazioni, consulta Amazon Web Services.

Se stai generando eventi personalizzati da un'applicazione e desideri inviarli Sumo Logic come log o metriche, utilizza EventBridge API Destinations e gli endpoint Sumo Logic HTTP Log e Metric Source.

- Per effettuare la registrazione e creare un'istanza Sumo Logic gratuita, consulta Start your free trial today.
- Per ulteriori informazioni sull'uso di Sumo Logic, consulta HTTP Logs and Metrics Source.

Operazioni API di uso comune

POST https://endpoint4.collection.us2.sumologic.com/receiver/v1/http/*UNIQUE_ID_PER_COLLECTOR*

Informazioni aggiuntive

Nessuno

TriggerMesh

URL dell'endpoint di invocazione della destinazione API

Utilizza le informazioni nell'argomento Event Source for HTTP per formulare l'URL dell'endpoint. L'URL di un endpoint include il nome dell'origine dell'evento e lo spazio dei nomi utente nel seguente formato:

https://*source-name*.*user-namespace*.cloud.triggermesh.io

Includi i parametri dell'autorizzazione Base nella richiesta all'endpoint.

Tipi di autorizzazione supportati

Base

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di TriggerMesh

Event Source for HTTP

Operazioni API di uso comune

Non applicabile

Informazioni aggiuntive

Nessuno

Zendesk

URL dell'endpoint di invocazione della destinazione API

https://developer.zendesk.com/rest_api/docs/support/tickets

Tipi di autorizzazione supportati

Base, Chiave API

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di Zendesk

Security and Authentication

Operazioni API di uso comune

POST https://*your_Zendesk_subdomain*/api/v2/tickets

Informazioni aggiuntive

Le richieste API vengono EventBridge conteggiate ai limiti delle API Zendesk. Per informazioni sui limiti di Zendesk per il tuo piano, consulta Usage limits.

Per proteggere meglio il tuo account e i tuoi dati, ti consigliamo di utilizzare una chiave API anziché l'autenticazione di base con credenziali di accesso.

EventBridge Obiettivi di Amazon per Amazon API Gateway

Per creare, pubblicare, gestire e monitorare le API puoi utilizzare Gateway Amazon API. Amazon EventBridge supporta l'invio di eventi a un endpoint API Gateway. Quando specifichi un endpoint

Gateway API come destinazione, ogni evento inviato alla destinazione viene mappato a una richiesta inviata all'endpoint.

> **Important**
>
> EventBridge supporta l'utilizzo di endpoint regionali e ottimizzati per API Gateway Edge come destinazioni. Gli endpoint privati non sono attualmente supportati. Per ulteriori informazioni sugli endpoint, consulta https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-api-endpoint-types.html.

Puoi utilizzare un destinazione di Gateway API per i seguenti casi d'uso:

- Per richiamare un'API specificata dal cliente ospitata in API Gateway in base AWS a eventi di terze parti.
- Per richiamare un endpoint periodicamente in base a una pianificazione.

Le informazioni sull'evento EventBridge JSON vengono inviate come corpo della richiesta HTTP all'endpoint. È possibile specificare gli altri attributi della richiesta nel campo `HttpParameters` della destinazione come segue:

- `PathParameterValues` elenca i valori che corrispondono in sequenza a qualsiasi variabile di percorso nell'ARN dell'endpoint, ad esempio `"arn:aws:execute-api:us-east-1:112233445566:myapi/dev/POST/pets/*/*"`.
- `QueryStringParameters`rappresenta i parametri della stringa di query che vengono EventBridge aggiunti all'endpoint richiamato.
- `HeaderParameters` definisce le intestazioni HTTP da aggiungere alla richiesta.

> **Note**
>
> Per motivi di sicurezza, le seguenti chiavi di intestazione HTTP non sono consentite:
>
> - Qualsiasi chiave con prefisso `X-Amz` o `X-Amzn`
> - `Authorization`
> - `Connection`
> - `Content-Encoding`

- `Content-Length`
- `Host`
- `Max-Forwards`
- `TE`
- `Transfer-Encoding`
- `Trailer`
- `Upgrade`
- `Via`
- `WWW-Authenticate`
- `X-Forwarded-For`

Parametri dinamici

Quando si richiama una destinazione di Gateway API, è possibile aggiungere dinamicamente dati agli eventi che vengono inviati alla destinazione. Per ulteriori informazioni, consulta the section called “Parametri di destinazione”.

Ripetizione di invocazioni

Come per tutti gli obiettivi, EventBridge riprova alcune chiamate non riuscite. Per API Gateway, EventBridge ritenta le risposte inviate con un codice di stato HTTP 5xx o 429 per un massimo di 24 ore con back off e jitter esponenziali. Successivamente, EventBridge pubblica una `FailedInvocations` metrica in Amazon. CloudWatch EventBridge non riprova altri errori HTTP 4xx.

Timeout

EventBridge regola Le richieste API Gateway devono avere un timeout di esecuzione client massimo di 5 secondi. Se API Gateway impiega più di 5 secondi per rispondere, EventBridge calcola il timeout della richiesta e riprova.

EventBridge Le richieste Pipes API Gateway hanno un timeout massimo di 29 secondi, il valore massimo dell'API Gateway.

AWS AppSync obiettivi per Amazon EventBridge

AWS AppSync consente agli sviluppatori di connettere le proprie applicazioni e servizi a dati ed eventi con API GraphQL e Pub/Sub sicure, serverless e ad alte prestazioni. Con AWS AppSync, puoi pubblicare aggiornamenti dei dati in tempo reale sulle tue applicazioni con mutazioni GraphQL. EventBridge supporta la chiamata di un'operazione di mutazione GraphQL valida per gli eventi corrispondenti. Quando specificate una mutazione AWS AppSync API come destinazione, AWS AppSync elabora l'evento tramite un'operazione di mutazione, che può quindi attivare le sottoscrizioni collegate alla mutazione.

> **Note**
>
> EventBridge supporta le API GraphQL AWS AppSync pubbliche. EventBridge attualmente non supporta le API AWS AppSync private.

È possibile utilizzare un target API AWS AppSync GraphQL per i seguenti casi d'uso:

- Per inviare, trasformare e archiviare i dati degli eventi nelle origini di dati configurate.
- Per inviare notifiche in tempo reale ai client applicativi connessi.

> **Note**
>
> AWS AppSync gli obiettivi supportano solo la chiamata alle API AWS AppSync GraphQL utilizzando il `AWS_IAM` tipo di autorizzazione.

Per ulteriori informazioni sulle API AWS AppSync GraphQL, consulta GraphQL e l' AWS AppSync architettura nella Developer Guide.AWS AppSync

Per specificare un AWS AppSync obiettivo per una EventBridge regola utilizzando la console

1. Crea o modifica la regola.
2. In Destinazione, specifica l'obiettivo scegliendo servizio AWS e poi AWS AppSync.
3. Specifica l'operazione di mutazione da analizzare ed eseguire, insieme al set di selezione.
   - Scegli l' AWS AppSync API, quindi la mutazione dell'API GraphQL da richiamare.

- In Configura parametri e set di selezione, scegli di creare un set di selezione utilizzando la mappatura chiave-valore o un trasformatore di input.

  Key-value mapping

  Per utilizzare la mappatura chiave-valore per creare il set di selezione:

  - Specifica le variabili per i parametri dell'API. Ogni variabile può essere un valore statico o un'espressione di percorso JSON dinamica per il payload dell'evento.
  - In Set di selezione, scegli le variabili che desideri includere nella risposta.

  Input transformer

  Per utilizzare un trasformatore di input per creare il set di selezione:

  - Specifica un percorso di input che definisca le variabili da utilizzare.
  - Specifica un modello di input per definire e formattare le informazioni che desideri trasmettere alla destinazione.

  Per ulteriori informazioni, consulta ???.

4. In Ruolo di esecuzione, scegli se creare un nuovo ruolo o utilizzarne uno esistente.
5. Completa la creazione o la modifica della regola.

Esempio: AWS AppSync obiettivi per Amazon EventBridge

Nel seguente esempio, spiegheremo come specificare un AWS AppSync obiettivo per una EventBridge regola, inclusa la definizione di una trasformazione di input per formattare gli eventi per la consegna.

Supponiamo di avere un'API AWS AppSync GraphQL`Ec2EventAPI`, definita dallo schema seguente:

```
type Event {
    id: ID!
    statusCode: String
    instanceId: String
}

type Mutation {
    pushEvent(id: ID!, statusCode: String!, instanceId: String): Event
}
```

```
type Query {
    listEvents: [Event]
}

type Subscription {
    subscribeToEvent(id: ID, statusCode: String, instanceId: String): Event
        @aws_subscribe(mutations: ["pushEvent"])
}
```

Le applicazioni client che utilizzano questa API possono sottoscrivere l'abbonamento `subscribeToEvent`, che viene attivato dalla mutazione `pushEvent`.

È possibile creare una EventBridge regola con un target che invia eventi all' AppSync API tramite la `pushEvent` mutazione. Quando viene richiamata la mutazione, qualsiasi client sottoscritto riceverà l'evento.

Per specificare questa API come destinazione di una EventBridge regola, procedi come segue:

1. Imposta l'Amazon Resource Name (ARN) della destinazione della regola sull'ARN dell'endpoint GraphQL dell'API `Ec2EventAPI`.
2. Specifica la mutazione GraphQL Operation come parametro di destinazione:

   ```
   mutation CreatePushEvent($id: ID!, $statusCode: String, $instanceId: String) {
     pushEvent(id: $input, statusCode: $statusCode, instanceId: $instanceId) {
       id
       statusCode
       instanceId
     }
   }
   ```

   Il set di selezione delle mutazioni deve includere tutti i campi a cui desideri iscriverti nel tuo abbonamento GraphQL.
3. Configura un trasformatore di input per specificare in che modo i dati degli eventi corrispondenti vengono utilizzati nella tua operazione.

   Supponiamo di aver selezionato l'evento di esempio "`EC2 Instance Launch Successful`":

   ```
   {
     "version": "0",
     "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
     "detail-type": "EC2 Instance Launch Successful",
   ```

```
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": ["arn:aws:autoscaling:us-
east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-b893-
d5978ed4a025:autoScalingGroupName/sampleLuanchSucASG", "arn:aws:ec2:us-
east-1:123456789012:instance/i-b188560f"],
  "detail": {
    "StatusCode": "InProgress",
    "AutoScalingGroupName": "sampleLuanchSucASG",
    "ActivityId": "9cabb81f-42de-417d-8aa7-ce16bf026590",
    "Details": {
      "Availability Zone": "us-east-1b",
      "Subnet ID": "subnet-95bfcebe"
    },
    "RequestId": "9cabb81f-42de-417d-8aa7-ce16bf026590",
    "EndTime": "2015-11-11T21:31:47.208Z",
    "EC2InstanceId": "i-b188560f",
    "StartTime": "2015-11-11T21:31:13.671Z",
    "Cause": "At 2015-11-11T21:31:10Z a user request created an AutoScalingGroup
 changing the desired capacity from 0 to 1.  At 2015-11-11T21:31:11Z an instance was
 started in response to a difference between desired and actual capacity, increasing
 the capacity from 0 to 1."
  }
}
```

È possibile definire le seguenti variabili da utilizzare nel modello, utilizzando il percorso di input del trasformatore di input di destinazione:

```
{
  "id": "$.id",
  "statusCode": "$.detail.StatusCode",
  "EC2InstanceId": "$.detail.EC2InstanceId"
}
```

Componi il modello di trasformatore di input per definire le variabili che EventBridge passano all'operazione di mutazione AWS AppSync . Il modello deve restituire un formato JSON. Dato il nostro percorso di input, puoi comporre il seguente modello:

```
{
  "id": <id>,
```

```
  "statusCode": <statusCode>,
  "instanceId": <EC2InstanceId>
}
```

Connessioni per destinazioni endpoint HTTP

Una connessione definisce il metodo di autorizzazione e le credenziali EventBridge da utilizzare per la connessione a un determinato endpoint HTTP. Quando si configurano le impostazioni di autorizzazione e si crea una connessione, viene creato un accesso segreto per AWS Secrets Manager archiviare in modo sicuro le informazioni di autorizzazione. È inoltre possibile aggiungere parametri aggiuntivi da includere nella connessione, in base alla destinazione dell'endpoint HTTP.

Usa connessioni con:

- Destinazioni API

  Quando crei una destinazione API, specifichi una connessione per la destinazione. Puoi scegliere una connessione esistente dal tuo account o creare una connessione quando crei una destinazione API.

Metodi di autorizzazione per le connessioni

EventBridge le connessioni supportano i seguenti metodi di autorizzazione:

- Base
- Chiave API

  Per l'autorizzazione Basic e API Key, EventBridge compila automaticamente le intestazioni di autorizzazione richieste.
- OAuth

  Per l'autorizzazione OAuth, scambia EventBridge anche l'ID cliente e il segreto con un token di accesso e quindi lo gestisce in modo sicuro.

  I token OAUTH vengono aggiornati quando viene restituita una risposta 401 o 407.

Quando crei una connessione, puoi anche includere i parametri header, body e query necessari per l'autorizzazione con un endpoint. È possibile utilizzare la stessa connessione per più di un endpoint HTTP se l'autorizzazione per l'endpoint è la stessa.

Quando si crea una connessione e si aggiungono parametri di autorizzazione, EventBridge crea un accesso segreto. AWS Secrets Manager Il costo di archiviazione e accesso relativo al segreto di Secrets Manager è incluso nell'addebito per l'utilizzo di una destinazione API. Per ulteriori informazioni sulle migliori pratiche per l'utilizzo dei segreti con le destinazioni API, consulta AWS::Events::ApiDestinationla Guida per CloudFormation l'utente.

> **Note**
>
> Per creare o aggiornare correttamente una connessione, devi utilizzare un account autorizzato a utilizzare Secrets Manager. L'autorizzazione necessaria è inclusa nella AmazonEventBridgeFullAccess politica. La stessa autorizzazione viene concessa al ruolo collegato al servizio creato nel tuo account per la connessione.

Creazione di connessioni per destinazioni endpoint HTTP

Per creare una connessione da utilizzare con gli endpoint HTTP utilizzando la console EventBridge

1. Accedi AWS utilizzando un account con le autorizzazioni necessarie per gestire EventBridge e aprire la EventBridge console.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scorri verso il basso fino alla tabella delle destinazioni API, quindi scegli la scheda Connessioni.
4. Scegli Crea connessione.
5. Nella pagina Crea connessione, immetti un nome per la connessione in Nome connessione.
6. Immetti una descrizione per la connessione in Descrizione.
7. In Tipo di autorizzazione, seleziona il tipo di autorizzazione da utilizzare per autorizzare le connessioni all'endpoint HTTP specificato per la destinazione API che utilizza questa connessione. Esegui una di queste operazioni:
   - Scegli Base (nome utente/password), quindi immetti il Nome utente e la Password da utilizzare per l'autorizzazione con l'endpoint HTTP.
   - Scegli le Credenziali client OAuth, quindi immetti Endpoint di autorizzazione, Metodo HTTP, ID client e Segreto client da utilizzare per l'autorizzazione con l'endpoint.

In Parametri Http OAuth, aggiungi eventuali parametri da includere per l'autorizzazione con l'endpoint di autorizzazione. Seleziona un Parametro dall'elenco a discesa, quindi immetti una Chiave e un Valore. Per includere un parametro aggiuntivo, scegli Aggiungi parametro.

In Parametri Http di chiamata, aggiungi eventuali parametri aggiuntivi da includere nella richiesta di autorizzazione. Per aggiungere un parametro, seleziona un Parametro dall'elenco a discesa, quindi immetti una Chiave e un Valore. Per includere un parametro aggiuntivo, scegli Aggiungi parametro.

- Scegli Chiave API, quindi immetti il Nome chiave API e il Valore associato da utilizzare per l'autorizzazione della chiave API.

  In Parametri Http di chiamata, aggiungi eventuali parametri aggiuntivi da includere nella richiesta di autorizzazione. Per aggiungere un parametro, seleziona un Parametro dall'elenco a discesa, quindi immetti una Chiave e un Valore. Per includere un parametro aggiuntivo, scegli Aggiungi parametro.

8. Scegli Crea.

Modifica delle connessioni tramite la console EventBridge

È possibile modificare le connessioni esistenti.

Per modificare una connessione utilizzando la EventBridge console

1. Accedi AWS utilizzando un account con le autorizzazioni necessarie per gestire EventBridge e aprire la EventBridge console.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scorri verso il basso fino alla tabella delle destinazioni API, quindi scegli la scheda Connessioni.
4. Nella tabella Connessioni, scegli la connessione da modificare.
5. Nella pagina Dettagli di connessione, scegli Modifica.
6. Aggiorna i valori per la connessione, quindi scegli Aggiorna.

Annullamento dell'autorizzazione delle connessioni tramite la console EventBridge

Quando si rimuove l'autorizzazione di una connessione, vengono rimossi tutti i parametri di autorizzazione. La rimozione dei parametri di autorizzazione rimuove il segreto dalla connessione, quindi è possibile riutilizzarlo senza dover creare una nuova connessione.

> **Note**
>
> È necessario aggiornare tutti gli endpoint HTTP che utilizzano la connessione non autorizzata per utilizzare una connessione diversa per inviare correttamente le richieste all'endpoint HTTP.

Per rimuovere l'autorizzazione di una connessione

1. Accedi AWS utilizzando un account con le autorizzazioni necessarie per gestire EventBridge e aprire la console. EventBridge
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scorri verso il basso fino alla tabella delle destinazioni API, quindi scegli la scheda Connessioni.
4. Nella tabella Connessioni, scegli la connessione.
5. Nella pagina Dettagli di connessione, scegli Rimuovi autorizzazione.
6. Nella finestra di dialogo Rimuovere l'autorizzazione della connessione?, immetti il nome della connessione, quindi scegli Rimuovi autorizzazione.

Lo stato della connessione diventa Rimozione dell'autorizzazione in corso fino al completamento del processo. Dopo la rimozione, lo stato diventa Autorizzazione rimossa. Ora puoi modificare la connessione per aggiungere nuovi parametri di autorizzazione.

Invio e ricezione di EventBridge eventi Amazon tra AWS account

Puoi configurare l'invio e EventBridge la ricezione di eventi tra i bus degli eventi negli AWS account. Quando EventBridge configuri l'invio o la ricezione di eventi tra account, puoi specificare quali AWS account possono inviare o ricevere eventi dal bus eventi del tuo account. Puoi anche consentire o negare eventi da regole specifiche associate al router di eventi o eventi provenienti da origine specifiche. Per ulteriori informazioni, consulta Semplificazione dell'accesso tra più account con le politiche delle risorse di Amazon EventBridge

> **Note**
>
> Se lo utilizzi AWS Organizations, puoi specificare un'organizzazione e concedere l'accesso a tutti gli account di quell'organizzazione. Inoltre, al router di eventi di invio devono essere

associati ruoli IAM quando si inviano eventi a un altro account. Per ulteriori informazioni, consulta Che cos'è AWS Organizations? nella Guida per l'utente di AWS Organizations .

> **Note**
>
> Se utilizzi un piano di risposta dello Strumento di gestione degli incidenti come destinazione, tutti i piani di risposta condivisi con il tuo account sono disponibili per impostazione predefinita.

È possibile inviare e ricevere eventi tra bus di eventi in AWS account all'interno della stessa regione in tutte le regioni e tra account in diverse regioni, purché la regione di destinazione sia una regione di destinazione interregionale supportata.

I passaggi EventBridge per configurare l'invio o la ricezione di eventi da un bus di eventi in un account diverso includono quanto segue:

- Nell'account del destinatario, modificate le autorizzazioni su un bus di eventi per consentire ad AWS account specifici, a un'organizzazione o a tutti gli AWS account di inviare eventi all'account del destinatario.
- Sull'account mittente, configura una o più regole che abbiano come target il bus di eventi dell'account ricevitore.

  Se l'account mittente eredita le autorizzazioni per inviare eventi da un' AWS organizzazione, l'account mittente deve inoltre avere un ruolo IAM con politiche che gli consentano di inviare eventi all'account del destinatario. Se si utilizza il AWS Management Console per creare la regola che si rivolge al bus degli eventi nell'account del destinatario, il ruolo viene creato automaticamente. Se si utilizza il AWS CLI, è necessario creare il ruolo manualmente.
- Sull'account ricevitore, configura una o più regole corrispondenti agli eventi provenienti dall'account mittente.

Gli eventi inviati da un account a un altro vengono addebitati all'account di invio come eventi personalizzati. All'account di ricezione non verrà addebitato alcun costo. Per ulteriori informazioni, consulta la pagina EventBridge dei prezzi di Amazon.

Se un account ricevitore configura una regola che invia gli eventi ricevuti da un account mittente a un terzo account, tali eventi non vengono inviati al terzo account.

Se hai tre bus di eventi nello stesso account e imposti una regola sul primo bus di eventi per inoltrare gli eventi dal secondo bus di eventi a un terzo bus eventi, tali eventi non vengono inviati al terzo bus eventi.

Il video seguente illustra il routing degli eventi tra account: Instradamento degli eventi ai bus di altri account AWS

Concedi le autorizzazioni per consentire eventi da altri account AWS

Per ricevere eventi da altri account o organizzazioni, devi innanzitutto modificare le autorizzazioni del router di eventi dove intendi ricevere gli eventi. Il bus degli eventi predefinito accetta eventi provenienti da AWS servizi, altri AWS account autorizzati e `PutEvents` chiamate. Le autorizzazioni per un router di eventi vengono concesse o negate utilizzando una policy basata su risorse associata al router di eventi. Nella politica, puoi concedere le autorizzazioni ad altri AWS account utilizzando l'ID account o a un' AWS organizzazione utilizzando l'ID dell'organizzazione. Per ulteriori informazioni sulle autorizzazioni dei router di eventi, incluse le policy di esempio, consulta Autorizzazioni per router di eventi di Amazon EventBridge.

> **Note**
>
> EventBridge ora richiede che tutti i nuovi target cross-account event bus aggiungano ruoli IAM. Ciò vale solo per le destinazioni di router di eventi create dopo il 2 marzo 2023. Le applicazioni create senza un ruolo IAM prima di tale data non sono interessate. Tuttavia, consigliamo di aggiungere ruoli IAM per concedere agli utenti l'accesso alle risorse in un altro account, in quanto ciò garantisce l'applicazione, mediante policy di controllo dei servizi, di determinare chi può inviare e ricevere eventi dagli account nell'organizzazione.

> **Important**
>
> Se scegli di ricevere eventi da tutti gli AWS account, fai attenzione a creare regole che corrispondano solo agli eventi da ricevere dagli altri. Per creare regole più sicure, assicurati che il modello di eventi per ciascuna regola contenga un campo `Account` con gli ID account di uno o più account da cui intendi ricevere eventi. Le regole che dispongono di un modello di eventi contenente un campo Account non corrispondono agli eventi inviati dagli account che non sono elencati nel campo `Account`. Per ulteriori informazioni, consulta EventBridge Eventi Amazon.

Regole per gli eventi tra AWS account

Se il tuo account è configurato per ricevere eventi dai bus degli eventi in altri AWS account, puoi scrivere regole che corrispondano a tali eventi. Imposta il modello di eventi della regola affinché corrisponda agli eventi che ricevi da route di eventi nell'altro account.

A meno che non venga specificato `account` nel modello di eventi di una regola, tutte le regole dell'account, nuove ed esistenti, corrispondenti a eventi ricevuti da router di eventi, vengono attivate sulla base di tali eventi. Se ricevi eventi da router di eventi in altro account e vuoi che una regola venga attivata solo su quel modello di eventi quando generata dal tuo account, devi aggiungere `account` e specificare l'ID del tuo account al modello di eventi della regola.

Se configuri il tuo AWS account in modo da accettare eventi dagli event bus in tutti gli AWS account, ti consigliamo vivamente di aggiungere delle regole `account` a tutte le EventBridge regole del tuo account. In questo modo si evita che le regole del tuo account si attivino sugli eventi di AWS account sconosciuti. Quando specifichi il campo `account` nella regola, puoi specificare l'ID account di più di un account AWS .

Per far sì che una regola si attivi su un evento corrispondente proveniente da qualsiasi bus di eventi AWS dell'account a cui hai concesso le autorizzazioni, non specificare* nel `account` campo della regola. In questo modo non corrisponderà a nessun evento, perché * non appare mai nel campo `account` di un evento. Al contrario, è sufficiente omettere il campo `account` dalla regola.

Creazione di regole che inviano eventi tra account AWS

Specificare un router di eventi in un altro account come destinazione fa parte della creazione della regola.

Per creare una regola che invii eventi a un altro AWS account utilizzando la console

1. Segui i passaggi nella procedura ???.
2. Nel passaggio ???, quando viene richiesto di scegliere un tipo di destinazione:
    a. Seleziona EventBridge Event Bus.
    b. Seleziona Bus di eventi in un account diverso o in una Regione diversa.
    c. In Bus di eventi come destinazione, immetti l'ARN del router di eventi da utilizzare.
3. Completa la creazione della regola seguendo i passaggi della procedura.

Invio e ricezione di EventBridge eventi Amazon tra AWS regioni

È possibile EventBridge configurare l'invio e la ricezione di eventi tra AWS regioni. Puoi anche consentire o negare eventi provenienti da Regioni specifiche, regole specifiche associate al router di eventi o eventi provenienti da origini specifiche. Per ulteriori informazioni, consulta Introduzione al routing di eventi tra regioni con Amazon EventBridge

Le seguenti Regioni sono Regioni di destinazione supportate:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Seoul)
- Asia Pacifico (Osaka-Locale)
- Asia Pacifico (Mumbai)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Singapore)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Sydney)
- Asia Pacifico (Melbourne)
- Canada (Centrale)
- Canada occidentale (Calgary)
- Europa (Francoforte)
- Europa (Spagna)
- Europa (Zurigo)
- Europa (Stoccolma)
- Europa (Milano)
- Europa (Irlanda)
- Europe (London)

- Europa (Parigi)
- Israele (Tel Aviv)
- Medio Oriente (Emirati Arabi Uniti)
- Medio Oriente (Bahrein)
- Sud America (San Paolo)

Il video seguente illustra il routing degli eventi tra regioni utilizzando https://console.aws.amazon.com/events/, AWS CloudFormation, e AWS Serverless Application Model: Routing degli eventi tra regioni

Creazione di regole per l'invio di eventi a una regione diversa AWS

La specificazione di un bus di eventi in un'altra AWS regione come destinazione fa parte della creazione della regola.

Per creare una regola che invii eventi a un altro AWS account utilizzando la console

1. Segui i passaggi nella procedura ???.
2. Nel passaggio ???, quando viene richiesto di scegliere un tipo di destinazione:
   a. Seleziona EventBridge Event Bus.
   b. Seleziona Bus di eventi in un account diverso o in una Regione diversa.
   c. In Bus di eventi come destinazione, immetti l'ARN del router di eventi da utilizzare.
3. Completa la creazione della regola seguendo i passaggi della procedura.

Invio e ricezione di EventBridge eventi Amazon tra bus di eventi nello stesso account e nella stessa regione

Puoi EventBridge configurare l'invio e la ricezione di eventi tra bus di eventi nello stesso AWS account e nella stessa regione.

Quando EventBridge configuri l'invio o la ricezione di eventi tra bus di eventi, utilizzi i ruoli IAM sul bus degli eventi del mittente per autorizzare il bus degli eventi del mittente a inviare eventi al bus degli eventi del destinatario. Utilizzi policy basate su risorse nel router di eventi ricevente per concedere a tale router l'autorizzazione a ricevere eventi dal router di eventi mittente. Puoi anche

consentire o negare eventi da determinati router di eventi, regole specifiche associate al router di eventi o eventi provenienti da origine specifiche. Per ulteriori informazioni sulle autorizzazioni dei router di eventi, incluse le policy di esempio, consulta Autorizzazioni per router di eventi di Amazon EventBridge.

I passaggi EventBridge per configurare l'invio o la ricezione di eventi tra i bus di eventi nel tuo account includono quanto segue:

- Per utilizzare un ruolo IAM esistente, devi assegnare le autorizzazioni del router di eventi mittente al router di eventi ricevente o le autorizzazioni del router di eventi ricevente al router di eventi mittente.
- Nel router di eventi mittente, configura una o più regole che abbiano il router di eventi come destinazione e crea un ruolo IAM. Per un esempio della policy da associare al ruolo, vedi ???.
- Nel router di eventi ricevente, modifica le autorizzazioni per consentire il passaggio degli eventi dall'altro router di eventi.
- Nell'evento ricevente, configura una o più regole corrispondenti agli eventi provenienti dal router di eventi mittente.

> **Note**
>
> EventBridge non può indirizzare gli eventi ricevuti da un bus eventi mittente a un terzo bus di eventi.

Gli eventi inviati da un router di eventi a un altro vengono addebitati come eventi personalizzati. Per ulteriori informazioni, consulta Prezzi di Amazon EventBridge.

Creazione di regole che inviano eventi a un bus di eventi diverso nello stesso AWS account e nella stessa regione

Per inviare eventi a un altro router di eventi, crei una regola con un router di eventi come destinazione. La specificazione di un bus di eventi nello stesso AWS account e nella stessa regione di destinazione fa parte della creazione della regola.

Per creare una regola che invii eventi a un bus di eventi diverso nello stesso AWS account e nella stessa regione utilizzando la console

1. Segui i passaggi nella procedura ???.

2. Nel passaggio ???, quando viene richiesto di scegliere un tipo di destinazione:

    a. Seleziona il bus degli EventBridge eventi.

    b. Seleziona Event bus nello stesso AWS account e nella stessa regione.

    c. In Bus di eventi come destinazione, seleziona un router di eventi dall'elenco a discesa.

3. Completa la creazione della regola seguendo i passaggi della procedura.

Trasformazione degli EventBridge input di Amazon

Puoi personalizzare il testo di un evento prima di EventBridge passare le informazioni al destinatario di una regola. Utilizzando il trasformatore di input nella console o nell'API, definisci le variabili che utilizzano il percorso JSON per fare riferimento ai valori nell'origine dell'evento originale. L'evento trasformato viene inviato a una destinazione anziché all'evento originale. Tuttavia, i parametri di percorso dinamici devono fare riferimento all'evento originale, non all'evento trasformato. Puoi definire fino a 100 variabili, assegnando a ciascuna un valore dall'input. Puoi quindi utilizzare queste variabili nel modello di input nel formato <*nome-variabile*>.

Per un tutorial sull'uso del trasformatore di input, consulta ???.

> **Note**
>
> EventBridge non supporta tutta la sintassi di JSON Path e la valuta in fase di esecuzione. La sintassi supportata include:
>
> - notazione a punti (ad esempio, `$.detail`)
> - trattini
> - caratteri di sottolineatura
> - caratteri alfanumerici
> - indici array
> - caratteri jolly (*)

In questo argomento:

Variabili predefinite

Esistono variabili predefinite che puoi utilizzare senza definire un percorso JSON. Queste variabili sono riservate e non puoi creare variabili con questi nomi:

- `aws.events.rule-arn`— L'Amazon Resource Name (ARN) della EventBridge regola.
- `aws.events.rule-name`— Il nome della EventBridge regola.
- `aws.events.event.ingestion-time`— L'ora in cui l'evento è stato ricevuto da EventBridge. Si tratta di un timestamp ISO 8601. Questa variabile è generata da EventBridge e non può essere sovrascritta.
- `aws.events.event`: il payload dell'evento originale in formato JSON (senza il campo `detail`). Può essere utilizzato solo come valore per un campo JSON, poiché il relativo contenuto non ha caratteri di escape.
- `aws.events.event.json`: il payload completo dell'evento originale in formato JSON. (con il campo `detail`). Può essere utilizzato solo come valore per un campo JSON, poiché il relativo contenuto non ha caratteri di escape.

Esempi di trasformazione di input

Di seguito è riportato un esempio di evento Amazon EC2.

```
{
  "version": "0",
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2015-11-11T21:29:54Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
  ],
  "detail": {
    "instance-id": "i-0123456789",
    "state": "RUNNING"
  }
}
```

Quando definisci una regola nella console, seleziona l'opzione Input Transformer (Trasformatore di input) in Configure input (Configura input). Questa opzione visualizza due caselle di testo: una per Input Path (Percorso di input) e una per Input Template (Modello di input).

Percorso di input viene utilizzato per definire le variabili. Utilizza il percorso JSON per fare riferimento a elementi nel tuo evento e archiviare tali valori in variabili. Ad esempio, puoi creare un Input Path (Percorso di input) per fare riferimento ai valori nell'evento di esempio immettendo quanto segue nella prima casella di testo. È inoltre possibile utilizzare parentesi e indici per ottenere elementi dagli array.

> **Note**
>
> EventBridge sostituisce i trasformatori di input in fase di esecuzione per garantire un output JSON valido. Per questo motivo, inserisci tra virgolette le variabili che fanno riferimento ai parametri di percorso JSON, ma non le variabili che si riferiscono a oggetti o array JSON.

```
{
  "timestamp" : "$.time",
  "instance" : "$.detail.instance-id",
  "state" : "$.detail.state",
  "resource" : "$.resources[0]"
}
```

In questo modo, si definiscono quattro variabili, `<timestamp>`, `<instance>`, `<state>` e `<resource>`. Puoi fare riferimento a queste variabili durante la creazione di Input Path (Percorso di input).

Input Template (Modello di input) è un modello per le informazioni che desideri passare alla destinazione. Puoi creare un modello che passa una stringa o un codice JSON alla destinazione. Utilizzando l'evento precedente e Input Path (Percorso di input), i seguenti esempi di Input Template (Modello di input) trasformeranno l'evento nell'output di esempio prima di indirizzarlo a una destinazione.

Descrizione	Modello	Output
Stringa semplice	`"instance <instance> is in <state>"`	`"instance i-0123456789 is in RUNNING"`

Descrizione	Modello	Output
Stringa con virgolette di escape	`"instance \"<instance> \" is in <state>"`	`"instance \"i-01234 56789\" is in RUNNING"` Nota che questo è il comportamento della console. EventBridge AWS CLI esegu e l'escape dei caratteri di barra e il risultato è `"instance "i-0123456789" is in RUNNING"`.
JSON semplice	`{ "instance" : <instance>, "state": <state> }`	`{ "instance" : "i-0123456789", "state": "RUNNING" }`
JSON con stringhe e variabili	`{ "instance" : <instance >, "state": "<state>", "instanceStatus": "instance \"<instance> \" is in <state>" }`	`{ "instance" : "i-012345 6789", "state": "RUNNING", "instanceStatus": "instance \"i-01234 56789\" is in RUNNING" }`
JSON con un mix di variabili e informazioni statiche	`{ "instance" : <instance>, "state": [ 9, <state>, true ], "Transformed" : "Yes" }`	`{ "instance" : "i-0123456789", "state": [ 9, "RUNNING", true ], "Transformed" : "Yes" }`

Descrizione	Modello	Output
Inclusione di variabili riservate in JSON	`{ "instance" : <instance>, "state": <state>, "ruleArn" : <aws.events.rule-arn>, "ruleName" : <aws.events.rule-name>, "originalEvent" : <aws.events.event.json> }`	`{ "instance" : "i-0123456789", "state": "RUNNING", "ruleArn" : "arn:aws:events:us-east-2:123456789012:rule/example", "ruleName" : "example", "originalEvent" : { ... // commented for brevity } }`
Inclusione di variabili riservate in una stringa	`"<aws.events.rule-name> triggered"`	`"example triggered"`
Gruppo di CloudWatch log Amazon	`{ "timestamp" : <timestamp>, "message": "instance \"<instance>\" is in <state>" }`	`{ "timestamp" : 2015-11-11T21:29:54Z, "message": "instance "i-0123456789" is in RUNNING }`

Trasformazione dell'input utilizzando l'API EventBridge

Per informazioni sull'utilizzo dell' EventBridge API per trasformare l'input, consulta Utilizzare Input Transformer per estrarre dati da un evento e immettere tali dati nella destinazione.

Trasformazione dell'input utilizzando AWS CloudFormation

Per informazioni sull'utilizzo AWS CloudFormation per trasformare l'input, vedere AWS::Events::Rule InputTransformer.

Problemi comuni con la trasformazione di input

Questi sono alcuni problemi comuni quando si trasforma l'input in EventBridge:

- Per le stringhe, le virgolette sono obbligatorie.
- Non vi è alcuna convalida durante la creazione del percorso JSON per il modello.
- Se specifichi una variabile per la corrispondenza con un percorso JSON che non esiste nell'evento, quella variabile non viene creata e non appare nell'output.
- Le proprietà JSON come `aws.events.event.json` possono essere utilizzate solo come valore di un campo JSON, non in linea in altre stringhe.
- EventBridge non sfugge ai valori estratti da Input Path, quando compila il modello di input per un target.
- Se un percorso JSON fa riferimento a un oggetto o a un array JSON, ma la variabile è referenziata in una stringa, EventBridge rimuove tutte le virgolette interne per garantire una stringa valida. Ad esempio, per una variabile a cui si `<detail>` punta`$.detail`, «Detail is<detail>" comporterebbe la EventBridge rimozione delle virgolette dall'oggetto.

  Pertanto, se come output vuoi un oggetto JSON basato su una singola variabile di percorso JSON, devi posizionarlo come chiave. In questo esempio, `{"detail": <detail>}`.
- Le virgolette non sono necessarie per le variabili che rappresentano stringhe. Sono consentite, ma aggiungono EventBridge automaticamente le virgolette ai valori delle variabili di stringa durante la trasformazione, per garantire che l'output della trasformazione sia JSON valido. EventBridge non aggiunge virgolette alle variabili che rappresentano oggetti o array JSON. Non aggiungere virgolette alle variabili che rappresentano oggetti o array JSON.

  Ad esempio, il seguente modello di input include variabili che rappresentano sia stringhe che oggetti JSON:

```
{
  "ruleArn" : <aws.events.rule-arn>,
  "ruleName" : <aws.events.rule-name>,
  "originalEvent" : <aws.events.event.json>
}
```

  Il risultato è JSON valido con un uso corretto delle virgolette:

```
{
```

```
  "ruleArn" : "arn:aws:events:us-east-2:123456789012:rule/example",
  "ruleName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

- Per l'output di testo (non JSON) come stringhe multilinea, raccogli ogni riga separata del modello di input tra virgolette doppie.

  Ad esempio, se stavi Amazon Inspector confrontando gli eventi di Finding con il seguente schema di eventi:

  ```
  {
    "detail": {
      "severity": ["HIGH"],
      "status": ["ACTIVE"]
    },
    "detail-type": ["Inspector2 Finding"],
    "source": ["inspector2"]
  }
  ```

  E utilizzando il seguente percorso di input:

  ```
  {
    "account": "$.detail.awsAccountId",
    "ami": "$.detail.resources[0].details.awsEc2Instance.imageId",
    "arn": "$.detail.findingArn",
    "description": "$.detail.description",
    "instance": "$.detail.resources[0].id",
    "platform": "$.detail.resources[0].details.awsEc2Instance.platform",
    "region": "$.detail.resources[0].region",
    "severity": "$.detail.severity",
    "time": "$.time",
    "title": "$.detail.title",
    "type": "$.detail.type"
  }
  ```

  È possibile utilizzare il modello di input seguente per generare un output di stringhe multilinea:

  ```
  "<severity> severity finding <title>"
  "Description: <description>"
  ```

```
"ARN: \"<arn>\""
"Type: <type>"
"AWS Account: <account>"
"Region: <region>"
"EC2 Instance: <instance>"
"Platform: <platform>"
"AMI: <ami>"
```

Configurazione di un trasformatore di input come parte della creazione di una regola

Come parte della creazione di una regola, è possibile specificare un trasformatore di input EventBridge da utilizzare per elaborare gli eventi di corrispondenza prima di inviarli alla destinazione specificata. È possibile configurare trasformatori di input per destinazioni che sono AWS servizi o destinazioni API.

Per creare un trasformatore di input di destinazione come parte di una regola

1. Segui i passaggi per creare una regola come descritto in ???.
2. In Passaggio 3: selezionare le destinazioni, espandi Impostazioni aggiuntive.
3. In Configura l'input di destinazione, scegli Trasformatore di input dall'elenco a discesa.

   Fai clic su Configura il trasformatore di input.

   EventBridge visualizza la finestra di dialogo Configura trasformatore di ingresso.
4. Nella sezione Evento di esempio, scegli un Tipo evento di esempio in base al quale desideri testare il modello di eventi. Puoi scegliere un AWS evento, un evento partner o inserire il tuo evento personalizzato.

   AWS events

   Seleziona uno degli eventi emessi dai Servizi AWS supportati.

   1. Seleziona Eventi AWS .
   2. In Eventi di esempio, scegli l' AWS evento desiderato. Gli eventi sono organizzati per AWS servizio.

      Quando si seleziona un evento, EventBridge compila l'evento di esempio.

Ad esempio, se scegliete S3 Object Created, EventBridge visualizza un esempio di evento S3 Object Created.

3. (Facoltativo) Puoi anche selezionare Copia per copiare l'evento di esempio negli appunti del dispositivo.

Partner events

Seleziona tra gli eventi emessi da servizi di terze parti che supportano EventBridge, come Salesforce.

1. Seleziona EventBridge gli eventi dei partner.
2. In Eventi di esempio, scegli l'evento partner desiderato. Gli eventi sono organizzati per partner.

   Quando si seleziona un evento, EventBridge compila l'evento di esempio.
3. (Facoltativo) Puoi anche selezionare Copia per copiare l'evento di esempio negli appunti del dispositivo.

Enter your own

Immetti il tuo evento in formato JSON.

1. Seleziona Inserisci il mio.
2. EventBridge compila l'evento di esempio con un modello di attributi di evento obbligatori.
3. Modifica e aggiungi all'evento di esempio come desiderato. L'evento di esempio deve essere JSON valido.
4. (Facoltativo) È anche possibile scegliere una delle seguenti opzioni:
   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Abbellisci (Prettify): semplifica la lettura del testo JSON aggiungendo interruzioni di riga, tabulazioni e spazi.

5. (Facoltativo) Espandi la sezione Esempi di percorsi di input, modelli e output per visualizzare esempi di:

   - Come vengono utilizzati i percorsi JSON per definire variabili che rappresentano i dati degli eventi

- Come possono essere utilizzate queste variabili in un modello di trasformatore di input
- L'output risultante che EventBridge viene inviato alla destinazione

Per esempi più dettagliati di trasformazioni di input, consulta ???.

6. Nella sezione Transformatore di input di destinazione, definisci le variabili che desideri utilizzare nel modello di input.

   Le variabili utilizzano il percorso JSON per fare riferimento ai valori nell'origine dell'evento originale. È quindi possibile fare riferimento a tali variabili nel modello di input per includere i dati dell'evento di origine originale nell'evento trasformato che EventBridge passa alla destinazione. Puoi definire fino a 100 variabili. Il trasformatore di input deve essere in formato JSON valido.

   Ad esempio, supponete di aver scelto l' AWS evento S3 Object Created come evento di esempio per questo trasformatore di input. Puoi quindi definire le seguenti variabili da utilizzare nel modello:

   ```
   {
     "requester": "$.detail.requester",
     "key": "$.detail.object.key",
     "bucket": "$.detail.bucket.name"
   }
   ```

   (Facoltativo) Puoi anche scegliere Copia per copiare il trasformatore di input negli appunti del tuo dispositivo.

7. Nella sezione Modello, componi il modello che desideri utilizzare per determinare cosa EventBridge passare al bersaglio.

   Puoi usare JSON, stringhe, informazioni statiche, variabili che hai definito e variabili riservate. Per esempi più dettagliati di trasformazioni di input, consulta ???.

   Ad esempio, supponiamo che hai definito le variabili nell'esempio precedente. È quindi possibile comporre il seguente modello, che fa riferimento a tali variabili, nonché a variabili riservate e ad informazioni statiche.

   ```
   {
       "message": "<requester> has created the object \"<key>\" in the bucket
    \"<bucket>\"",
       "RuleName": <aws.events.rule-name>,
   ```

```
    "ruleArn" : <aws.events.rule-arn>,
    "Transformed": "Yes"
}
```

(Facoltativo) Puoi anche scegliere Copia per copiare il modello negli appunti del tuo dispositivo.

8. Per testare il modello, seleziona Genera output.

   EventBridge elabora l'evento di esempio in base al modello di input e visualizza l'output trasformato generato in Output. Queste sono le informazioni che EventBridge verranno passate alla destinazione al posto dell'evento di origine originale.

   L'output generato per il modello di input di esempio descritto sopra sarebbe il come segue:

   ```
   {
       "message": "123456789012 has created the object "example-key" in the bucket
    "example-bucket"",
       "RuleName": rule-name,
       "ruleArn" : arn:aws:events:us-east-1:123456789012:rule/rule-name,
       "Transformed": "Yes"
   }
   ```

   (Facoltativo) Puoi anche scegliere Copia per copiare l'output generato negli appunti del tuo dispositivo.

9. Seleziona Conferma.
10. Segui gli altri passaggi per creare una regola come descritto in ???.

Test di un trasformatore di ingresso target utilizzando la Sandbox EventBridge

È possibile utilizzare trasformatori di input per personalizzare il testo di un evento prima di EventBridge passare le informazioni alla destinazione di una regola.

La configurazione di un trasformatore di input fa in genere parte del processo più ampio di specificazione di una destinazione durante la creazione di una nuova regola o la modifica di una regola esistente. Utilizzando Sandbox in EventBridge, tuttavia, è possibile configurare rapidamente un trasformatore di input e utilizzare un evento di esempio per confermare che si sta ottenendo l'output desiderato, senza dover creare o modificare una regola.

Per ulteriori informazioni sulle trasformazioni di input, consulta ???.

Per testare un trasformatore di input di destinazione

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. In Risorse per gli sviluppatori, scegli Sandbox e nella pagina Sandbox scegli la scheda Trasformatore di input di destinazione.
3. Nella sezione Evento di esempio, scegli un Tipo evento di esempio in base al quale desideri testare il modello di eventi. Puoi scegliere un AWS evento, un evento partner o partecipare al tuo evento personalizzato.

   AWS events

   Seleziona uno degli eventi emessi dai Servizi AWS supportati.

   1. Seleziona Eventi AWS .
   2. In Eventi di esempio, scegli l' AWS evento desiderato. Gli eventi sono organizzati per AWS servizio.

      Quando si seleziona un evento, EventBridge compila l'evento di esempio.

      Ad esempio, se scegliete S3 Object Created, EventBridge visualizza un esempio di evento S3 Object Created.
   3. (Facoltativo) Puoi anche selezionare Copia per copiare l'evento di esempio negli appunti del dispositivo.

   Partner events

   Seleziona tra gli eventi emessi da servizi di terze parti che supportano EventBridge, come Salesforce.

   1. Seleziona EventBridge gli eventi dei partner.
   2. In Eventi di esempio, scegli l'evento partner desiderato. Gli eventi sono organizzati per partner.

      Quando si seleziona un evento, EventBridge compila l'evento di esempio.
   3. (Facoltativo) Puoi anche selezionare Copia per copiare l'evento di esempio negli appunti del dispositivo.

Enter your own

Immetti il tuo evento in formato JSON.

1. Seleziona Inserisci il mio.
2. EventBridge compila l'evento di esempio con un modello di attributi di evento obbligatori.
3. Modifica e aggiungi all'evento di esempio come desiderato. L'evento di esempio deve essere JSON valido.
4. (Facoltativo) È anche possibile scegliere una delle seguenti opzioni:
   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Abbellisci (Prettify): semplifica la lettura del testo JSON aggiungendo interruzioni di riga, tabulazioni e spazi.

4. (Facoltativo) Espandi la sezione Esempi di percorsi di input, modelli e output per visualizzare esempi di:

   - Come vengono utilizzati i percorsi JSON per definire variabili che rappresentano i dati degli eventi
   - Come possono essere utilizzate queste variabili in un modello di trasformatore di input
   - L'output risultante che EventBridge viene inviato alla destinazione

   Per esempi più dettagliati di trasformazioni di input, consulta ???.

5. Nella sezione Transformatore di input di destinazione, definisci le variabili che desideri utilizzare nel modello di input.

   Le variabili utilizzano il percorso JSON per fare riferimento ai valori nell'origine dell'evento originale. È quindi possibile fare riferimento a tali variabili nel modello di input per includere i dati dell'evento di origine originale nell'evento trasformato che EventBridge passa alla destinazione. Puoi definire fino a 100 variabili. Il trasformatore di input deve essere in formato JSON valido.

   Ad esempio, supponete di aver scelto l' AWS evento S3 Object Created come evento di esempio per questo trasformatore di input. Puoi quindi definire le seguenti variabili da utilizzare nel modello:

```
{
  "requester": "$.detail.requester",
```

```
  "key": "$.detail.object.key",
  "bucket": "$.detail.bucket.name"
}
```

(Facoltativo) Puoi anche scegliere Copia per copiare il trasformatore di input negli appunti del tuo dispositivo.

6. Nella sezione Modello, componi il modello che desideri utilizzare per determinare cosa EventBridge passare al bersaglio.

   Puoi usare JSON, stringhe, informazioni statiche, variabili che hai definito e variabili riservate. Per esempi più dettagliati di trasformazioni di input, consulta ???.

   Ad esempio, supponiamo che hai definito le variabili nell'esempio precedente. È quindi possibile comporre il seguente modello, che fa riferimento a tali variabili, nonché a variabili riservate e ad informazioni statiche.

   ```
   {
       "message": "<requester> has created the object \"<key>\" in the bucket
    \"<bucket>\"",
       "RuleName": <aws.events.rule-name>,
       "ruleArn" : <aws.events.rule-arn>,
       "Transformed": "Yes"
   }
   ```

   (Facoltativo) Puoi anche scegliere Copia per copiare il modello negli appunti del tuo dispositivo.

7. Per testare il modello, seleziona Genera output.

   EventBridge elabora l'evento di esempio in base al modello di input e visualizza l'output trasformato generato in Output. Queste sono le informazioni che EventBridge verranno passate alla destinazione al posto dell'evento di origine originale.

   L'output generato per il modello di input di esempio descritto sopra sarebbe il come segue:

   ```
   {
       "message": "123456789012 has created the object "example-key" in the bucket
    "example-bucket"",
       "RuleName": rule-name,
       "ruleArn" : arn:aws:events:us-east-1:123456789012:rule/rule-name,
       "Transformed": "Yes"
   ```

```
}
```

(Facoltativo) Puoi anche scegliere Copia per copiare l'output generato negli appunti del tuo dispositivo.

Archiviazione e riproduzione di Amazon EventBridge

In EventBridge, puoi creare un archivio di eventi in modo da poterli riprodurre facilmente in un secondo momento. Ad esempio, è possibile che tu abbia la necessità di riprodurre gli eventi per correggere gli errori o per convalidare nuove funzionalità nell'applicazione.

> **Note**
>
> Potrebbe verificarsi un ritardo tra la pubblicazione di un evento su un router di eventi e l'arrivo dell'evento nell'archivio. Ti consigliamo di ritardare la riproduzione degli eventi archiviati di 10 minuti per assicurarti che tutti gli eventi vengano riprodotti.

Il video seguente illustra l'uso dell'archiviazione e della riproduzione: Creating archives and replays

Argomenti

- Archiviazione degli eventi Amazon EventBridge
- Riproduzione di eventi Amazon EventBridge archiviati

Archiviazione degli eventi Amazon EventBridge

Quando crei un archivio in EventBridge, puoi determinare quali eventi vengono inviati all'archivio specificando uno schema di evento. EventBridge invia all'archivio gli eventi che corrispondono al modello di evento. È inoltre possibile impostare il periodo di conservazione per archiviare eventi nell'archivio prima che vengano eliminati.

Per impostazione predefinita, EventBridge crittografa i dati degli eventi in un archivio utilizzando l'Advanced Encryption Standard (AES-256) a 256 bit con un CMK di AWS proprietà, che aiuta a proteggere i dati da accessi non autorizzati.

> **Note**
>
> I `SizeBytes` valori `EventCount` e dell'`DescribeArchive`operazione hanno un periodo di riconciliazione di 24 ore. Pertanto, eventuali eventi scaduti di recente o appena archiviati potrebbero non riflettersi immediatamente in questi valori.

Per creare un archivio per tutti gli eventi

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione a sinistra, scegli Archivi.
3. Scegli Crea archivio.
4. In Dettaglio dell'archivio, immetti un nome per l'archivio in Nome. Il nome deve essere univoco per l'account nella Regione selezionata.

   Non puoi modificare il nome dopo aver creato l'archivio.
5. (Facoltativo) Immetti una descrizione dell'archivio in Descrizione.
6. In Origine, seleziona il router di eventi che emette gli eventi da inviare all'archivio.
7. In Periodo di conservazione, effettua una delle seguenti operazioni:
   - Scegli Indefinita per mantenere gli eventi nell'archivio e non eliminarli mai.
   - Immetti il numero di giorni durante i quali mantenere gli eventi. Dopo il numero di giorni specificato, EventBridge elimina gli eventi dall'archivio.
8. Seleziona Successivo.
9. In Modello di eventi, scegli Nessun filtro di eventi.
10. Scegli Crea archivio.

Per creare un archivio con un modello di eventi

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione a sinistra, scegli Archivi.
3. Scegli Crea archivio.
4. In Dettaglio dell'archivio, immetti un nome per l'archivio in Nome. Il nome deve essere univoco per l'account nella Regione selezionata.

   Non puoi modificare il nome dopo aver creato l'archivio.
5. (Facoltativo) Immetti una descrizione dell'archivio in Descrizione.
6. In Origine, seleziona il router di eventi che emette gli eventi da inviare all'archivio.
7. In Periodo di conservazione, effettua una delle seguenti operazioni:
   - Scegli Indefinita* per mantenere gli eventi nell'archivio e non eliminarli mai.
   - Immetti il numero di giorni durante i quali mantenere gli eventi. Dopo il numero di giorni specificato, EventBridge elimina gli eventi dall'archivio.
8. Seleziona Successivo.
9. In Modello di eventi, scegli Filtraggio degli eventi in base alla corrispondenza del modello di eventi.
10. Esegui una di queste operazioni:
    - Seleziona Generatore di modelli, quindi scegli il fornitore di servizi in Fornitore di servizi. Se scegli AWS, seleziona anche il nome del servizio AWS e il tipo di evento da utilizzare nel modello.
    - Seleziona l'editor JSON per creare un modello manualmente. Puoi anche copiare il modello da una regola e incollarlo nell'editor JSON.
11. Scegli Crea archivio.

Per confermare che gli eventi siano stati inviati correttamente all'archivio, puoi utilizzare il `DescribeArchive`funzionamento dell' EventBridge API per vedere se `EventCount` riflette il numero di eventi nell'archivio. Se il valore è 0, non ci sono eventi nell'archivio.

Riproduzione di eventi Amazon EventBridge archiviati

Dopo aver creato un archivio, puoi riprodurre gli eventi dall'archivio. Ad esempio, se aggiorni un'applicazione con funzionalità aggiuntive, puoi riprodurre gli eventi storici per garantire la rielaborazione degli eventi allo scopo di mantenere l'applicazione coerente. Puoi anche utilizzare un archivio per riprodurre eventi per nuove funzionalità. Quando riproduci gli eventi, puoi specificare da quale archivio riprodurli, l'ora di inizio e quella di fine dell'evento da riprodurre, il router di eventi o una o più regole in base alle quali riprodurre gli eventi.

Gli eventi non vengono necessariamente riprodotti nello stesso ordine in cui sono stati aggiunti all'archivio. Una riproduzione elabora gli eventi da riprodurre in base all'ora dell'evento e li riproduce ad intervalli di un minuto. Se si specifica l'ora di inizio e l'ora di fine di un evento che copre un intervallo di tempo di 20 minuti, vengono riprodotti dapprima gli eventi del primo minuto di quell'intervallo. Quindi vengono riprodotti gli eventi del secondo minuto. Puoi utilizzare l'operazione `DescribeReplay` dell'API di EventBridge per determinare l'avanzamento di una riproduzione. `EventLastReplayedTime` restituisce il timestamp dell'ultimo evento riprodotto.

Gli eventi vengono riprodotti in base al limite di transazioni `PutEvents` al secondo per l'account AWS, ma separatamente da tale limite. È possibile richiedere un aumento del limite per PutEvents. Per ulteriori informazioni, consulta Quote di Amazon EventBridge.

> **Note**
>
> È possibile avere un massimo di 10 riproduzioni simultanee attive per account per Regione AWS.

Per avviare la riproduzione di un evento

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione a sinistra, scegli Riproduzioni.
3. Scegli Avvia nuova riproduzione.
4. Immetti un nome ed eventualmente una descrizione per la riproduzione rispettivamente in Nome e Descrizione.
5. In Origine, seleziona l'archivio da cui riprodurre gli eventi.
6. Per la destinazione, puoi riprodurre gli eventi solo nel router di eventi che li ha emessi.
7. In Specifica regole, esegui una delle operazioni descritte di seguito:

- Scegli Tutte le regole per riprodurre gli eventi in base a tutte le regole.
- Scegli Specifica regole, quindi seleziona la regola o le regole in base alle quali riprodurre gli eventi.

8. In Intervallo di tempo della riproduzione, specifica la Data, l'Ora e il Fuso orario per l'Ora di inizio e l'Ora di fine. Vengono riprodotti solo gli eventi che si sono verificati tra l'Ora di inizio e l'Ora di fine.
9. Scegli Avvia la riproduzione.

Quando gli eventi archiviati vengono riprodotti, lo stato della riproduzione è Completato.

Se avvii una riproduzione e poi desideri interromperla, puoi annullarla purché lo stato sia Avvio in corso o In esecuzione.

Per annullare una riproduzione

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione a sinistra, scegli Riproduzioni.
3. Scegli la riproduzione da annullare.
4. Seleziona Annulla.

EventBridge Tubi Amazon

Amazon EventBridge Pipes collega le sorgenti alle destinazioni. Le pipe sono destinate point-to-point alle integrazioni tra sorgenti e destinazioni supportate, con supporto per trasformazioni e arricchimenti avanzati. Riduce la necessità di conoscenze specialistiche e codice di integrazione durante lo sviluppo di architetture basate su eventi, favorendo la coerenza tra le applicazioni aziendali. Per configurare una pipe, si sceglie l'origine, si aggiungono filtri facoltativi, si definisce l'arricchimento facoltativo e si sceglie la destinazione per i dati dell'evento.

> **Note**
>
> Puoi anche instradare gli eventi utilizzando router di eventi. Gli Event Bus sono ideali per il many-to-many routing degli eventi tra servizi basati sugli eventi. Per ulteriori informazioni, consulta ???.

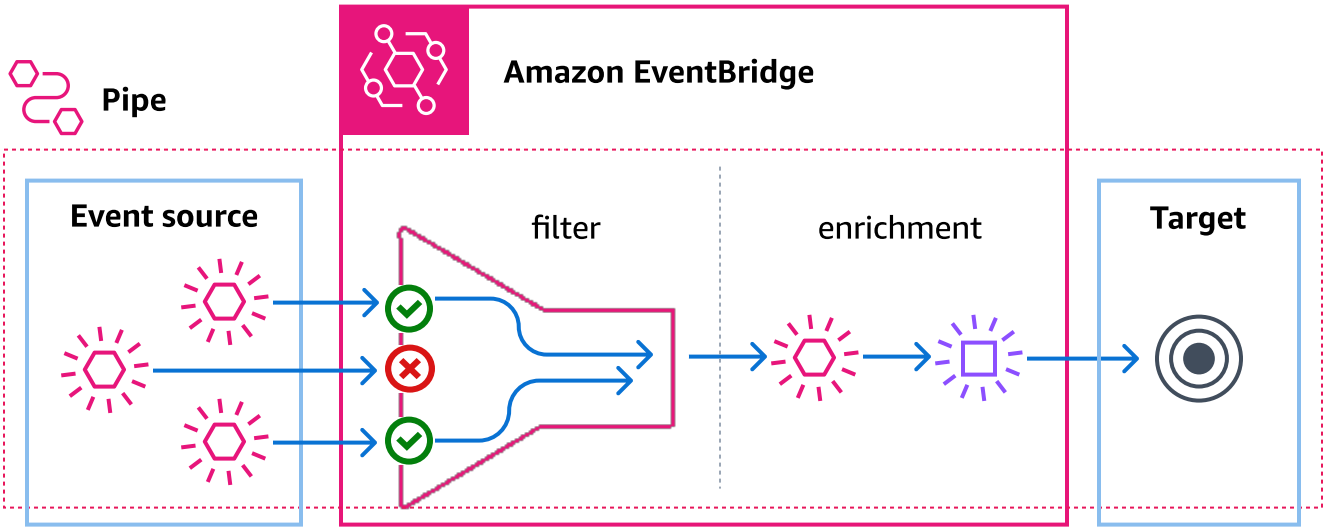
EventBridge Come funzionano i tubi

Ad alto livello, ecco come funziona EventBridge Pipes:

1. Crei una pipe nel tuo account. Questo include:
   - La selezione di una delle origini di eventi supportate da cui la pipe deve ricevere eventi.
   - Eventualmente, puoi configurare un filtro di modo che la pipe elabori solo un sottoinsieme degli eventi che riceve dall'origine.
   - Eventualmente, configurare un passaggio di arricchimento che migliori i dati dell'evento prima di inviarli alla destinazione.
   - Selezione di una delle destinazioni supportate a cui la pipe deve inviare gli eventi.
2. L'origine degli eventi inizia a inviare gli eventi alla pipe e la pipe elabora l'evento prima di inviarlo alla destinazione.
   - Se hai configurato un filtro, la pipe valuta l'evento e lo invia alla destinazione solo se corrisponde a quel filtro.

     Ti vengono addebitati solo gli eventi che corrispondono al filtro.
   - Se hai configurato un arricchimento, la pipe esegue quell'arricchimento sull'evento prima di inviarlo alla destinazione.

Se gli eventi sono in un batch, l'arricchimento mantiene l'ordine degli eventi nel batch.



Ad esempio, una pipe potrebbe essere utilizzata per creare un sistema di e-commerce. Supponiamo di avere un'API che contiene informazioni sui clienti, come gli indirizzi di spedizione.

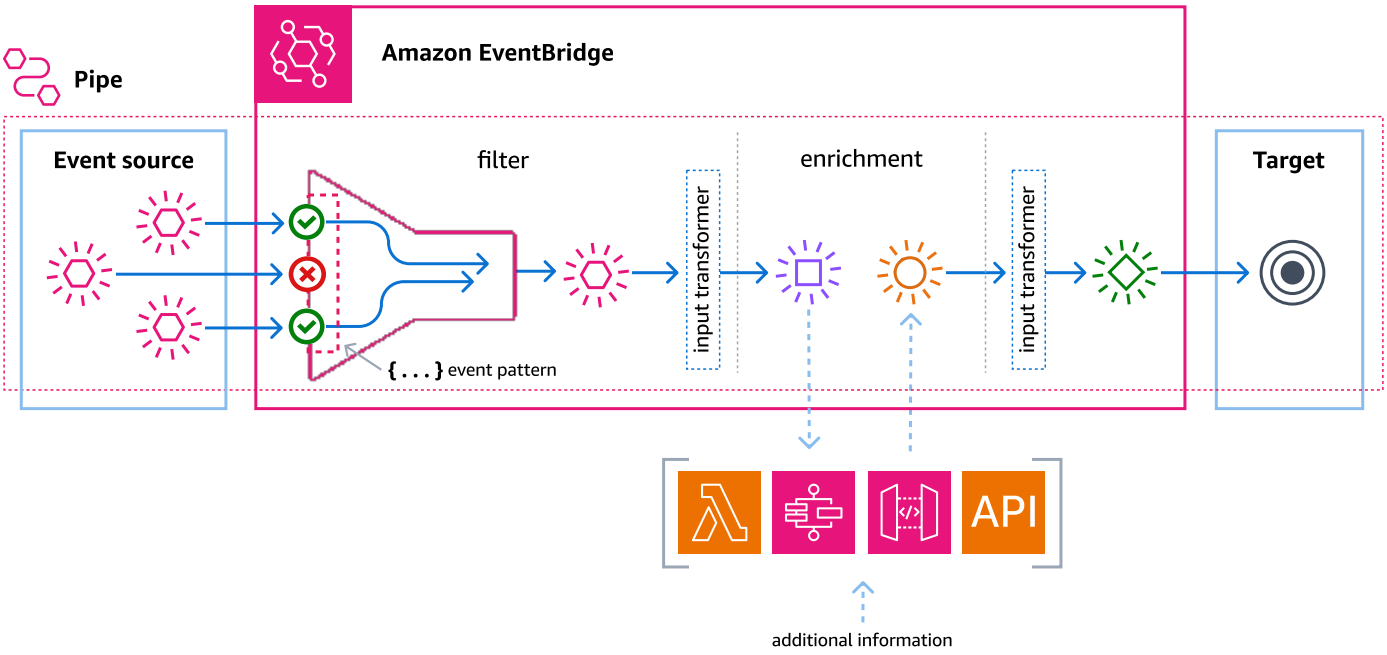
1. A questo proposito, crei una pipe con:
   - Una coda di messaggi ricevuti dall'ordine Amazon SQS come origine dell'evento.
   - Una destinazione EventBridge API come arricchimento
   - Una macchina a AWS Step Functions stati come obiettivo
2. Quindi, quando un messaggio ricevuto dell'ordine Amazon SQS appare nella coda, viene inviato alla pipe.
3. La pipe invia quindi i dati all' EventBridge API Destination Enrichment, che restituisce le informazioni sul cliente per quell'ordine.
4. Infine, la pipe invia i dati arricchiti alla macchina a AWS Step Functions stati, che elabora l'ordine.

EventBridge Concetti di tubi

Ecco uno sguardo più da vicino ai componenti di base di EventBridge Pipes.

Pipeline

Una pipe instrada gli eventi da un'unica origine a una singola destinazione. La pipe include anche la possibilità di filtrare eventi specifici e di eseguire arricchimenti sui dati degli eventi prima che vengano inviati alla destinazione.



Origine

EventBridge Pipes riceve i dati degli eventi da diverse fonti, applica filtri e arricchimenti opzionali a tali dati e li invia a una destinazione. Se un'origine impone l'ordine agli eventi inviati a EventBridge Pipes, tale ordine viene mantenuto durante l'intero processo verso la destinazione.

Per ulteriori informazioni sulle origini, consulta ???.

Filtri

Una pipe può filtrare gli eventi di una determinata origine e quindi elaborare solo un sottoinsieme di quegli eventi. Per configurare i filtri su una pipe, definisci un modello di eventi utilizzato dalla pipe per determinare quali eventi inviare alla destinazione.

Ti vengono addebitati solo gli eventi che corrispondono al filtro.

Per ulteriori informazioni, consulta ???.

Arricchimento

Con la fase di arricchimento di EventBridge Pipes, puoi migliorare i dati dall'origine prima di inviarli alla destinazione. Ad esempio, potresti ricevere eventi creati da ticket che non includono i dati completi del ticket. Con l'arricchimento, è possibile chiamare l'API `get-ticket` mediante una funzione Lambda per ottenere dettagli completi sul ticket. La pipe può quindi inviare tali informazioni a una destinazione.

Per ulteriori informazioni sull'arricchimento dei dati dell'evento, consulta ???.

Target

Dopo che i dati dell'evento sono stati filtrati e arricchiti, puoi specificare la pipe per inviarli a una destinazione specifica, ad esempio uno stream Amazon Kinesis o un gruppo di log CloudWatch Amazon. Per un elenco di destinazioni disponibili, consulta ???.

Puoi trasformare i dati dopo che sono stati migliorati e prima che vengano inviati dalla pipe alla destinazione. Per ulteriori informazioni, consulta ???.

Più pipe, ognuna con un'origine diversa, possono inviare eventi alla stessa destinazione.

È inoltre possibile utilizzare insieme pipe e router di eventi per inviare eventi a più destinazioni. Un caso d'uso comune consiste nel creare una pipe con un router di eventi come destinazione; la pipe invia gli eventi al router di eventi, che quindi invia tali eventi a più destinazioni. Ad esempio, potresti creare una pipe con un flusso DynamoDB per un'origine e un router di eventi come destinazione. La pipe riceve eventi dal flusso DynamoDB e li invia al router di eventi, che quindi li invia a più destinazioni in base alle regole che hai specificato nel router di eventi.

Autorizzazioni per Amazon EventBridge Pipes

Quando configuri una pipe, puoi utilizzare un ruolo di esecuzione esistente o fare in modo che EventBridge ne crei uno con le autorizzazioni necessarie. Le autorizzazioni richieste da EventBridge Pipes variano in base al tipo di origine e sono elencate di seguito. Se stai configurando il tuo ruolo di esecuzione, devi aggiungere tu stesso queste autorizzazioni.

> **Note**
>
> Se non sei sicuro delle autorizzazioni esatte necessarie per accedere all'origine, usa la console EventBridge Pipes per creare un nuovo ruolo, quindi esamina le azioni elencate nella policy.

Argomenti

Autorizzazioni del ruolo di esecuzione DynamoDB

Per i flussi DynamoDB, EventBridge Pipes richiede le seguenti autorizzazioni per gestire le risorse correlate al flusso di dati DynamoDB.

- `dynamodb:DescribeStream`
- `dynamodb:GetRecords`
- `dynamodb:GetShardIterator`
- `dynamodb:ListStreams`

Per inviare record di batch non riusciti alla coda DLQ delle pipe, il ruolo di esecuzione delle pipe necessita della seguente autorizzazione:

- `sqs:SendMessage`

Autorizzazioni del ruolo di esecuzione Kinesis

Per Kinesis, EventBridge Pipes richiede le seguenti autorizzazioni per gestire le risorse correlate al flusso di dati Kinesis.

- kinesis:DescribeStream
- kinesis:DescribeStreamSummary
- kinesis:GetRecords
- kinesis:GetShardIterator
- kinesis:ListShards
- kinesis:ListStreams
- kinesis:SubscribeToShard

Per inviare record di batch non riusciti alla coda DLQ delle pipe, il ruolo di esecuzione delle pipe necessita della seguente autorizzazione:

- sqs:SendMessage

Autorizzazioni del ruolo di esecuzione Amazon MQ

Per Amazon MQ, EventBridge Pipes richiede le seguenti autorizzazioni per gestire le risorse correlate al broker di messaggi Kinesis.

- mq:DescribeBroker
- secretsmanager:GetSecretValue
- ec2:CreateNetworkInterface
- ec2:DeleteNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- logs:CreateLogGroup
- logs:CreateLogStream

- logs:PutLogEvents

Autorizzazioni del ruolo di esecuzione Amazon MSK

Per Amazon MSK, EventBridge richiede le seguenti autorizzazioni per gestire le risorse correlate all'argomento Amazon MSK.

> **Note**
>
> Se utilizzi l'autenticazione basata su ruolo IAM, per il tuo ruolo di esecuzione saranno necessarie le autorizzazioni elencate in ??? oltre a quelle elencate di seguito.

- kafka:DescribeClusterV2
- kafka:GetBootstrapBrokers
- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeVpcs
- ec2:DeleteNetworkInterface
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- logs:CreateLogGroup
- logs:CreateLogStream
- logs:PutLogEvents

Autorizzazioni del ruolo di esecuzione Apache Kafka autogestite

Per Apache Kafka autogestito, EventBridge richiede le seguenti autorizzazioni per gestire le risorse correlate al flusso Apache Kafka autogestito.

Autorizzazioni richieste

Per creare e archiviare log in un gruppo di log in File di log Amazon CloudWatch, la pipe deve disporre delle seguenti autorizzazioni nel relativo ruolo di esecuzione:

- logs:CreateLogGroup
- logs:CreateLogStream
- logs:PutLogEvents

Autorizzazioni facoltative

La pipe potrebbe anche richiedere autorizzazioni per:

- Descrivere il segreto di Secrets Manager.
- Accedere alla chiave gestita dal cliente AWS Key Management Service (AWS KMS).
- Accedere ad Amazon VPC.

Secrets Manager e autorizzazioni AWS KMS

A seconda del tipo di controllo di accesso che stai configurando per i broker Apache Kafka, è possibile che per la tua pipe sia necessaria l'autorizzazione per accedere al segreto di Secrets Manager o per decrittare la chiave gestita dal cliente AWS KMS. Per accedere a queste risorse, il ruolo di esecuzione della funzione deve disporre delle seguenti autorizzazioni:

- secretsmanager:GetSecretValue
- kms:Decrypt

Autorizzazioni VPC

Se soltanto gli utenti in un VPC possono accedere al cluster Apache Kafka autogestito, la tua pipe deve disporre dell'autorizzazione per accedere alle risorse di Amazon VPC. Queste risorse includono la VPC, le sottoreti, i gruppi di sicurezza e le interfacce di rete. Per accedere a queste risorse, il ruolo di esecuzione della pipe deve disporre delle seguenti autorizzazioni:

- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeVpcs
- ec2:DeleteNetworkInterface
- ec2:DescribeSubnets

- `ec2:DescribeSecurityGroups`

Autorizzazioni del ruolo di esecuzione Amazon SQS

Per Amazon MSK, EventBridge richiede le seguenti autorizzazioni per gestire le risorse correlate alla coda Amazon SQS.

- `sqs:ReceiveMessage`
- `sqs:DeleteMessage`
- `sqs:GetQueueAttributes`

Autorizzazioni di arricchimento e destinazione

Per poter effettuare chiamate API alle risorse di tua proprietà, EventBridge Pipes necessita autorizzazioni appropriate. EventBridge Pipes utilizza il ruolo IAM specificato per la pipe per le chiamate di arricchimento e di destinazione mediante il principale IAM `pipes.amazonaws.com`.

Creare una EventBridge pipe Amazon

EventBridge Pipes ti consente di creare point-to-point integrazioni tra fonti e destinazioni, comprese trasformazioni e arricchimenti avanzati degli eventi. Per creare una EventBridge pipe, effettuate le seguenti operazioni:

1. ???
2. ???
3. ???
4. ???
5. ???

Per informazioni su come creare una pipe utilizzando la AWS CLI, consulta create-pipe nel CLI Command Reference.AWS

Specificare un'origine

Per iniziare, specifica l'origine da cui la pipe deve ricevere eventi.

Per specificare l''origine di una pipe utilizzando la console

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Pipe.
3. Scegli Crea pipe.
4. Immetti un nome per la pipe.
5. (Facoltativo) Aggiungi una descrizione per la pipe.
6. Nella scheda Costruisci pipe, in Origine, scegli il tipo di origine da specificare per la pipe e configura l'origine.

   Le proprietà di configurazione differiscono in base al tipo di origine che scegli:

   Confluent

   Per configurare uno stream Confluent Cloud come sorgente, utilizzando la console

   1. Per Source, scegli Confluent Cloud.
   2. In Server di bootstrap, immetti gli indirizzi della coppia `host:port` dei tuoi broker.
   3. In Nome dell'argomento, immetti il nome dell'argomento che la pipe leggerà.
   4. (Facoltativo) In VPC, scegli il VPC da utilizzare. In Sottoreti VPC, scegli le sottoreti. In Gruppi di sicurezza VPC, scegli i gruppi di sicurezza.
   5. Per l'autenticazione: opzionale, attiva Usa autenticazione ed esegui quanto segue:
      a. In Metodo di autenticazione, scegli il tipo di autenticazione.
      b. In Chiave segreta, scegli la chiave segreta.

      Per ulteriori informazioni, consulta l'autenticazione alle risorse di Confluent Cloud nella documentazione di Confluent.
   6. (Facoltativo) In Impostazioni aggiuntive (facoltativo), procedi come segue:
      a. In Posizione di partenza, scegli una delle seguenti opzioni:
         - Più recente: inizia a leggere il flusso con il record più recente nella partizione.
         - Orizzonte di taglio: inizia a leggere il flusso con l'ultimo record non tagliato nella partizione. Questo è il record meno recente nella partizione.
      b. In Dimensione del batch (facoltativo), immetti un numero massimo di record per ogni batch. Il valore predefinito è 100.

c. In Finestra batch (facoltativa), immetti un numero massimo di secondi per raccogliere i record prima di procedere.

DynamoDB

1. In Origine, scegli Dynamo DB.
2. In Flusso DynamoDB, scegli il flusso da utilizzare come origine.
3. In Posizione di partenza, scegli una delle seguenti opzioni:
   - Più recente: inizia a leggere il flusso con il record più recente nella partizione.
   - Orizzonte di taglio: inizia a leggere il flusso con l'ultimo record non tagliato nella partizione. Questo è il record meno recente nella partizione.
4. (Facoltativo) In Impostazioni aggiuntive (facoltativo), procedi come segue:
   a. In Dimensione del batch (facoltativo), immetti un numero massimo di record per ogni batch. Il valore predefinito è 100.
   b. In Finestra batch (facoltativa), immetti un numero massimo di secondi per raccogliere i record prima di procedere.
   c. In Batch simultanei per partizione (facoltativo), immetti il numero di batch della stessa partizione che possono essere letti contemporaneamente.
   d. In In caso di errore parziale dell'articolo di un batch, scegli quanto segue:
      - AUTOMATIC_BISECT: dimezza ogni batch e riprova ogni metà fino a quando tutti i record non risultano elaborati o rimane un messaggio di errore nel batch.

> **Note**
>
> Se non scegli AUTOMATIC_BISECT, è possibile che vengano restituiti specifici record non riusciti e solo per quelli viene effettuato un nuovo tentativo.

Kinesis

Per configurare un'origine Kinesis utilizzando la console

1. In Origine, scegli Kinesis.
2. In Flusso Kinesis, scegli il flusso da utilizzare come origine.
3. In Posizione di partenza, scegli una delle seguenti opzioni:

- Più recente: inizia a leggere il flusso con il record più recente nella partizione.
- Orizzonte di taglio: inizia a leggere il flusso con l'ultimo record non tagliato nella partizione. Questo è il record meno recente nella partizione.
- Al timestamp: inizia a leggere il flusso a partire dalla data specificata. In Timestamp, immetti una data e un'ora utilizzando il formato AAAA/MM/GG e hh:mm:ss.

4. (Facoltativo) In Impostazioni aggiuntive (facoltativo), procedi come segue:
   a. In Dimensione del batch (facoltativo), immetti un numero massimo di record per ogni batch. Il valore predefinito è 100.
   b. (Facoltativo) In Finestra batch (facoltativo), immetti un numero massimo di secondi per raccogliere i record prima di procedere.
   c. In Batch simultanei per partizione (facoltativo), immetti il numero di batch della stessa partizione che possono essere letti contemporaneamente.
   d. In In caso di errore parziale dell'articolo di un batch, scegli quanto segue:
      - AUTOMATIC_BISECT: dimezza ogni batch e riprova ogni metà fino a quando tutti i record non risultano elaborati o rimane un messaggio di errore nel batch.

> **Note**
>
> Se non scegli AUTOMATIC_BISECT, è possibile che vengano restituiti specifici record non riusciti e solo per quelli viene effettuato un nuovo tentativo.

Amazon MQ

Per configurare un'origine Amazon MQ utilizzando la console

1. In Origine, scegli Amazon MQ.
2. In Broker Amazon MQ, scegli il flusso da utilizzare come origine.
3. In Nome della coda, immetti il nome della coda che la pipe leggerà.
4. In Metodo di autenticazione, scegli BASIC_AUTH.
5. In Chiave segreta, scegli la chiave segreta.
6. (Facoltativo) In Impostazioni aggiuntive (facoltativo), procedi come segue:
   a. In Dimensione del batch (facoltativo), immetti un numero massimo di messaggi per ogni batch. Il valore predefinito è 100.

b. In Finestra batch (facoltativa), immetti un numero massimo di secondi per raccogliere i record prima di procedere.

Amazon MSK

Per configurare un'origine Amazon MSK utilizzando la console

1. In Origine, scegli Amazon MSK.
2. In Cluster Amazon MSK, scegli il cluster da utilizzare.
3. In Nome dell'argomento, immetti il nome dell'argomento che la pipe leggerà.
4. (Facoltativo) In ID del gruppo di consumatori (facoltativo), immetti l'ID del gruppo di consumer a cui la pipe deve aderire.
5. (Facoltativo) In Autenticazione (facoltativo), attiva Usa l'autenticazione ed esegui le seguenti operazioni:
   a. In Metodo di autenticazione, scegli il tipo che desideri.
   b. In Chiave segreta, scegli la chiave segreta.
6. (Facoltativo) In Impostazioni aggiuntive (facoltativo), procedi come segue:
   a. In Dimensione del batch (facoltativo), immetti un numero massimo di record per ogni batch. Il valore predefinito è 100.
   b. In Finestra batch (facoltativa), immetti un numero massimo di secondi per raccogliere i record prima di procedere.
   c. In Posizione di partenza, scegli una delle seguenti opzioni:
      - Più recente: inizia a leggere l'argomento con il record più recente nella partizione.
      - Orizzonte di taglio: inizia a leggere l'argomento con l'ultimo record non tagliato nella partizione. Questo è il record meno recente nella partizione.

> **Note**
>
> Orizzonte di taglio è simile a Meno recente per Apache Kafka.

Self managed Apache Kafka

Per configurare un'origine Apache Kafka autogestita utilizzando la console

1. In Origine, scegli Apache Kafka autogestita.
2. In Server di bootstrap, immetti gli indirizzi della coppia `host:port` dei tuoi broker.
3. In Nome dell'argomento, immetti il nome dell'argomento che la pipe leggerà.
4. (Facoltativo) In VPC, scegli il VPC da utilizzare. In Sottoreti VPC, scegli le sottoreti. In Gruppi di sicurezza VPC, scegli i gruppi di sicurezza.
5. (Facoltativo) In Autenticazione (facoltativo), attiva Usa l'autenticazione ed esegui le seguenti operazioni:
   a. In Metodo di autenticazione, scegli il tipo di autenticazione.
   b. In Chiave segreta, scegli la chiave segreta.
6. (Facoltativo) In Impostazioni aggiuntive (facoltativo), procedi come segue:
   a. In Posizione di partenza, scegli una delle seguenti opzioni:
      - Più recente: inizia a leggere il flusso con il record più recente nella partizione.
      - Orizzonte di taglio: inizia a leggere il flusso con l'ultimo record non tagliato nella partizione. Questo è il record meno recente nella partizione.
   b. In Dimensione del batch (facoltativo), immetti un numero massimo di record per ogni batch. Il valore predefinito è 100.
   c. In Finestra batch (facoltativa), immetti un numero massimo di secondi per raccogliere i record prima di procedere.

Amazon SQS

Per configurare un'origine Amazon SQS utilizzando la console

1. In Origine, scegli SQS.
2. In coda SQS, scegli la coda da utilizzare.
3. (Facoltativo) In Impostazioni aggiuntive (facoltativo), procedi come segue:
   a. In Dimensione del batch (facoltativo), immetti un numero massimo di record per ogni batch. Il valore predefinito è 100.

b. In Finestra batch (facoltativa), immetti un numero massimo di secondi per raccogliere i record prima di procedere.

Configurazione dei filtri di eventi (facoltativo)

Puoi aggiungere filtri alla tua pipe in modo da inviare solo un sottoinsieme di eventi dall'origine alla destinazione.

Per configurare i filtri utilizzando la console

1. Scegli Filtro.
2. In Evento di esempio (facoltativo), vedrai un evento di esempio che puoi usare per creare il tuo modello di eventi, oppure puoi immettere il tuo evento scegliendo Inserisci il mio.
3. In Modello di eventi, immetti il modello di eventi da utilizzare per filtrare gli eventi. Per ulteriori informazioni sulla creazione di filtri, consulta. ???

   Di seguito è riportato un esempio di modello di eventi che invia solo eventi con il valore Seattle nel campo City.

   ```
   {
     "data": {
       "City": ["Seattle"]
     }
   }
   ```

Ora che gli eventi vengono filtrati, puoi aggiungere un arricchimento facoltativo e un destinazione per la pipe.

Definizione dell'arricchimento degli eventi (facoltativo)

Puoi inviare i dati dell'evento per l'arricchimento a una funzione Lambda, una macchina a stati AWS Step Functions , Amazon API Gateway o una destinazione API.

Per selezionare l'arricchimento

1. Scegli Arricchimento.
2. In Dettagli, per Servizio, seleziona il servizio e le relative impostazioni da utilizzare per l'arricchimento.

Puoi anche trasformare i dati prima di inviarli per migliorarli.

(Facoltativo) Per definire il trasformatore di input

1. Scegli Trasformatore di input di arricchimento (facoltativo).
2. In Esempio di eventi/payload di eventi, scegli il tipo di evento di esempio.
3. In Trasformatore, immetti la sintassi del trasformatore, ad esempio `"Event happened at <$.detail.field>."` dove `<$.detail.field>` è un riferimento a un campo dell'evento di esempio. Puoi anche fare doppio clic su un campo dell'evento di esempio per aggiungerlo al trasformatore.
4. In Output, verifica che l'output sia come desiderato.

Ora che i dati sono stati filtrati e migliorati, devi definire una destinazione a cui inviare i dati dell'evento.

Configurazione di una destinazione

Per configurare una destinazione

1. Scegli Destinazione.
2. In Dettagli, per Servizio di destinazione, scegli la destinazione. I campi visualizzati variano a seconda della destinazione scelta. Immetti informazioni specifiche per questo tipo di destinazione, come necessario.

Puoi anche trasformare i dati prima di inviarli alla destinazione.

(Facoltativo) Per definire il trasformatore di input

1. Scegli Trasformatore di input di destinazione (facoltativo).
2. In Esempio di eventi/payload di eventi, scegli il tipo di evento di esempio.
3. In Trasformatore, immetti la sintassi del trasformatore, ad esempio `"Event happened at <$.detail.field>."` dove `<$.detail.field>` è un riferimento a un campo dell'evento di esempio. Puoi anche fare doppio clic su un campo dell'evento di esempio per aggiungerlo al trasformatore.
4. In Output, verifica che l'output sia come desiderato.

Ora che la pipe è configurata, assicurati che le relative impostazioni siano configurate correttamente.

Configurazione delle impostazioni della pipe

Una pipe è attiva per impostazione predefinita, ma è possibile disattivarla. È inoltre possibile specificare le autorizzazioni per le pipe, impostare la registrazione di log delle pipe e aggiungere tag.

Per configurare le impostazioni della pipe

1. Scegli la scheda Impostazioni delle pipe.
2. Per impostazione predefinita, le pipe appena create sono attive non appena vengono create. Se desideri creare una pipe inattiva, in Attivazione, per Attiva pipe, disattiva Attivo.
3. In Autorizzazioni, per Ruolo di esecuzione, effettua una delle seguenti operazioni:

    a. Per EventBridge creare un nuovo ruolo di esecuzione per questa pipe, scegli Crea un nuovo ruolo per questa risorsa specifica. In Nome ruolo, puoi eventualmente modificare il nome del ruolo.

    b. Per utilizzare il ruolo di esecuzione, scegli Utilizza un ruolo esistente. In Nome ruolo, scegli il ruolo.
4. (Facoltativo) Se avete specificato uno DynamoDB stream Kinesis or come sorgente pipe, potete configurare una politica di riprova e una coda di lettere dead-letter (DLQ).

    In Policy di ripetizione e coda DLQ (Dead-Letter Queue) (facoltativo), procedi come segue:

    In Policy di ripetizione, procedi come segue:

    a. Se desideri attivare le policy di ripetizione, attiva Riprova. Per impostazione predefinita, nelle pipe appena create non è attivata la policy di ripetizione.

    b. Per Maximum age of event (Età massima dell'evento), immetti un valore compreso tra un minuto (00:01) e 24 ore (24:00).

    c. Per Tentativi, specifica un numero compreso tra 0 e 185.

    d. Se desideri utilizzare una coda DLQ, attiva Coda DLQ, scegli il metodo e scegli la coda o l'argomento da utilizzare. Per impostazione predefinita, le pipe appena create non utilizzano una coda DLQ.
5. (Facoltativo) In Log (facoltativo), è possibile impostare il modo in cui EventBridge Pipes invia le informazioni sulla registrazione di log ai servizi supportati, incluso il modo in cui configurare tali log.

Per ulteriori informazioni sulla registrazione di log di record di pipe, consulta ???.

CloudWatch logs è selezionato come destinazione di log per impostazione predefinita, così come il livello di registro. `ERROR` Quindi, per impostazione predefinita, EventBridge Pipes crea un nuovo gruppo di CloudWatch log a cui invia i record di log contenenti il `ERROR` livello di dettaglio.

Per fare in modo che EventBridge Pipes invii i record di registro a una qualsiasi delle destinazioni di log supportate, effettuate le seguenti operazioni:

a. In Log (facoltativo), scegli le destinazioni a cui inviare i record di log.

b. Per Livello di registro, scegliete il livello di informazioni EventBridge da includere nei record di registro. Il livello di log `ERROR` è selezionato per impostazione predefinita.

   Per ulteriori informazioni, consulta ???.

c. Seleziona Includi dati di esecuzione se desideri includere EventBridge le informazioni sul payload degli eventi e le informazioni sulla richiesta e sulla risposta del servizio nei record di registro.

   Per ulteriori informazioni, consulta ???.

d. Configura ogni destinazione di log selezionata:

   Per CloudWatch Logs i log, in CloudWatch log procedi come segue:

   - Per CloudWatch il gruppo di log, scegli se EventBridge creare un nuovo gruppo di log oppure puoi selezionare un gruppo di log esistente o specificare l'ARN di un gruppo di log esistente.
   - Per i nuovi gruppi di log, modifica il nome del gruppo di log come desiderato.

   CloudWatch i log sono selezionati per impostazione predefinita.

   Per i log degli Firehose stream, in Firehose stream log, seleziona lo Firehose stream.

   Per Amazon S3 i log, in S3 logs procedi come segue:

   - Immetti il nome del bucket da utilizzare come destinazione dei log.
   - Inserisci l'ID dell' AWS account del proprietario del bucket.
   - Immetti il testo del prefisso da utilizzare quando EventBridge crea oggetti S3.

Per ulteriori informazioni, consulta Organizzazione degli oggetti utilizzando i prefissi nella Guida per l'utente di Amazon Simple Storage Service .

- Scegli come vuoi formattare EventBridge i record di registro S3:
  - `json`: JSON
  - `plain`: testo normale
  - `w3c`: formato di file di log W3C Extended

6. (Facoltativo) In Tag (facoltativo), scegli Aggiungi nuovo tag e immetti uno o più tag per la regola. Per ulteriori informazioni, consulta ???.
7. Scegli Crea pipe.

Convalida dei parametri di configurazione

Dopo aver creato una pipe, EventBridge convalida i seguenti parametri di configurazione:

- Ruolo IAM: poiché l'origine di una pipe non può essere modificata dopo la creazione della pipe, EventBridge verifica che il ruolo IAM fornito possa accedere all'origine.

> **Note**
>
> EventBridge non esegue la stessa convalida per gli arricchimenti o gli obiettivi perché possono essere aggiornati dopo la creazione della pipe.

- Batching: EventBridge verifica che la dimensione del batch dell'origine non superi la dimensione massima del batch della destinazione. In caso affermativo, EventBridge richiede una dimensione del batch inferiore. Inoltre, se una destinazione non supporta il batching, non è possibile configurare il batch in batch EventBridge per l'origine.
- Arricchimenti: EventBridge verifica che la dimensione del batch per API Gateway e gli arricchimenti delle destinazioni API sia 1 perché sono supportate solo le dimensioni dei batch pari a 1.

Avvio e arresto di una pipe

Per impostazione predefinita, una pipe è `Running` ed elabora eventi al momento della creazione.

Se crei una pipe con origini Amazon SQS, Kinesis o DynamoDB, l'operazione può richiedere in genere uno o due minuti.

Se crei una pipe con origini Amazon MSK o Amazon MQ oppure con origini Apache Kafka autogestite, l'operazione può richiedere fino a dieci minuti.

Per creare una pipe senza elaborare gli eventi utilizzando la console

- Disattiva l'impostazione Attiva pipe.

Per creare una pipe senza elaborare gli eventi a livello di codice

- Nella chiamata API, imposta `DesiredState` su `Stopped`.

Per avviare o arrestare una pipe esistente utilizzando la console

- Nella scheda Impostazioni Pipes, in Attivazione, per Attiva pipe, attiva o disattiva Attivo.

Per avviare o arrestare una pipe esistente a livello di codice

- Nella chiamata API, imposta il parametro `DesiredState` su `RUNNING` o `STOPPED`.

Può esserci un ritardo tra il momento in cui una pipe è `STOPPED` e il momento in cui non elabora più eventi:

- Per le origini Amazon SQS e di flussi, questo ritardo è in genere inferiore a due minuti.
- Per le origini Amazon MQ e Apache Kafka, questo ritardo può arrivare fino a quindici minuti.

Fonti Amazon EventBridge Pipes

EventBridge Pipes riceve i dati sugli eventi da diverse fonti, applica filtri e arricchimenti opzionali a tali dati e li invia a una destinazione.

Se una fonte impone un ordine agli eventi inviati a EventBridge Pipes, tale ordine viene mantenuto durante l'intero processo verso la destinazione.

I seguenti AWS servizi possono essere specificati come sorgenti per EventBridge Pipes:

- Flusso Amazon DynamoDB
- Flusso Amazon Kinesis

- Broker Amazon MQ
- Flusso Amazon MSK
- Coda Amazon SQS
- Stream Apache Kafka

  Quando specificate un flusso Apache Kafka come sorgente pipe, potete specificare uno stream Apache Kafka che gestite voi stessi o uno gestito da un provider di terze parti come:

  - Confluent Cloud
  - CloudKarafka
  - Redpanda

Flusso Amazon DynamoDB come origine

È possibile utilizzare EventBridge Pipes per ricevere record in un flusso DynamoDB. Puoi quindi eventualmente filtrare o migliorare questi record prima di inviarli a una delle destinazioni disponibili per l'elaborazione. Esistono impostazioni specifiche di flussi Amazon DynamoDB che puoi scegliere quando configuri una pipe. EventBridge Pipes mantiene l'ordine dei record dal flusso di dati durante l'invio dei dati alla destinazione.

> **Important**
>
> La disabilitazione di un flusso DynamoDB che è l'origine di una pipe fa sì che tale pipe diventi inutilizzabile, anche se successivamente si riabilita il flusso. Ciò avviene perché:
>
> - Non è possibile interrompere, avviare o aggiornare una pipe la cui origine è disabilitata.
> - Non è possibile aggiornare una pipe con una nuova origine dopo la creazione. Quando riabiliti un flusso DynamoDB, a quel flusso viene assegnato un nuovo nome della risorsa Amazon (ARN) e non è più associato alla tua pipe.
>
> Se riabiliti il flusso DynamoDB, dovrai creare una nuova pipe utilizzando il nuovo ARN del flusso.

Esempio di evento

L'evento di esempio seguente mostra le informazioni ricevute dalla pipe. È possibile utilizzare questo evento per creare e filtrare i modelli di eventi o per definire la trasformazione degli input. Non tutti i campi possono essere filtrati. Per ulteriori informazioni su quali campi è possibile filtrare, consulta ???.


```
[
  {
    "eventID": "1",
    "eventVersion": "1.0",
    "dynamodb": {
      "Keys": {
        "Id": {
          "N": "101"
        }
      },
      "NewImage": {
        "Message": {
          "S": "New item!"
        },
        "Id": {
          "N": "101"
        }
      },
      "StreamViewType": "NEW_AND_OLD_IMAGES",
      "SequenceNumber": "111",
      "SizeBytes": 26
    },
    "awsRegion": "us-west-2",
    "eventName": "INSERT",
    "eventSourceARN": "arn:aws:dynamodb:us-east-1:111122223333:table/EventSourceTable",
    "eventSource": "aws:dynamodb"
  },
  {
    "eventID": "2",
    "eventVersion": "1.0",
    "dynamodb": {
      "OldImage": {
        "Message": {
          "S": "New item!"
        },
        "Id": {
          "N": "101"
        }
```

```
      },
      "SequenceNumber": "222",
      "Keys": {
        "Id": {
          "N": "101"
        }
      },
      "SizeBytes": 59,
      "NewImage": {
        "Message": {
          "S": "This item has changed"
        },
        "Id": {
          "N": "101"
        }
      },
      "StreamViewType": "NEW_AND_OLD_IMAGES"
    },
    "awsRegion": "us-west-2",
    "eventName": "MODIFY",
    "eventSourceARN": "arn:aws:dynamodb:us-east-1:111122223333:table/EventSourceTable",
    "eventSource": "aws:dynamodb"
  }
]
```

Flussi di polling e batching

EventBridge esegue il polling delle partizioni presenti nel flusso DynamoDB per cercare i record a una velocità di base di quattro volte al secondo. Quando sono disponibili dei record, EventBridge elabora l'evento e attende il risultato. Se l'elaborazione ha esito positivo, EventBridge riprende a eseguire il polling fino a che non riceve più record.

Per impostazione predefinita, EventBridge richiama la pipe non appena i record sono disponibili. Se il batch che EventBridge legge dall'origine contiene un solo record, viene elaborato solo un evento. Per evitare di elaborare pochi record, puoi indicare alla pipe di memorizzare nel buffer i record per un massimo di cinque minuti configurando un periodo di batching. Prima di richiamare gli eventi, EventBridge continua a leggere i record dall'origine fino a quando non ha raccolto un batch completo, fino alla scadenza del periodo di batching o fino a quando il batch non ha raggiunto il limite del payload di 6 MB.

È anche possibile aumentare la simultaneità elaborando più batch da ogni partizione in parallelo. EventBridge può elaborare fino a 10 batch contemporaneamente in ogni partizione. Se aumenti il

numero di batch simultanei per partizione, EventBridge garantisce comunque l'ordine di elaborazione a livello di chiave di partizione.

Configura l'impostazione `ParallelizationFactor` per elaborare una partizione di un flusso di dati Kinesis o DynamoDB con più esecuzioni di pipe simultanee. È possibile specificare il numero di batch simultanei di cui EventBridge esegue il polling da una partizione tramite un fattore di parallelizzazione compreso tra 1 (predefinito) e 10. Ad esempio, se imposti `ParallelizationFactor` su 2, possono esserci al massimo 200 esecuzioni di pipe EventBridge simultanee per elaborare 100 partizioni di dati Kinesis. Ciò permette di aumentare la velocità effettiva di elaborazione quando il volume di dati non è stabile e `IteratorAge` è alto. Si noti che il fattore di parallelizzazione non funzionerà se si utilizza l'aggregazione Kinesis.

Posizioni di partenza di polling e flussi

Tieni presente che il polling di origine dei flussi durante la creazione e gli aggiornamenti della pipe alla fine è coerente.

- Durante la creazione della pipe, potrebbero essere necessari alcuni minuti per l'avvio degli eventi di polling dal flusso.
- Durante gli aggiornamenti della pipe per la configurazione del polling di origine, potrebbero essere necessari alcuni minuti per interrompere e riavviare gli eventi di polling dal flusso.

Ciò significa che se specifichi `LATEST` come posizione iniziale del flusso, la pipe potrebbe perdere degli eventi inviati durante la creazione o gli aggiornamenti della pipe. Per garantire che nessun evento venga perso, specifica la posizione iniziale del flusso come `TRIM_HORIZON`.

Segnalazione errori articoli batch

Quando EventBridge utilizza ed elabora i dati di streaming da un'origine, per impostazione predefinita imposta i checkpoint al numero di sequenza più alto di un batch solo quando il batch è riuscito completamente. Per evitare di rielaborare i messaggi correttamente elaborati in un batch non riuscito, puoi configurare l'arricchimento o la destinazione in modo da restituire un oggetto che indichi quali messaggi hanno avuto esito positivo e quali non. Questa operazione è nota come risposta batch parziale.

Per ulteriori informazioni, consulta ???.

Condizioni di batch riuscito e non riuscito

EventBridge considera un batch come riuscito completamente se viene restituito uno dei seguenti elementi:

- Un elenco `batchItemFailure` vuoto
- Un `batchItemFailure` elenco nullo
- Un vuoto `EventResponse`
- Un valore nullo `EventResponse`

EventBridge considera un batch come non riuscito completamente se viene restituito uno dei seguenti elementi:

- Una stringa `itemIdentifier` vuota
- Un valore nullo `itemIdentifier`
- Un `itemIdentifier` con un nome chiave errato

EventBridge esegue nuovi tentativi per gli errori in base alla strategia di ripetizione.

Flusso Amazon Kinesis come origine

Puoi utilizzare EventBridge Pipes per ricevere record in un flusso di dati Kinesis. Puoi eventualmente filtrare o migliorare questi record prima di inviarli a una delle destinazioni disponibili per l'elaborazione. Esistono impostazioni specifiche di Kinesis che puoi scegliere quando configuri la pipe. EventBridge Pipes mantiene l'ordine dei record dal flusso di dati durante l'invio dei dati alla destinazione.

Un flusso di dati Kinesis è un insieme di partizioni. Ogni partizione contiene una sequenza di record di dati. Un consumer è un'applicazione che elabora i dati da un flusso di dati Kinesis. È possibile mappare una pipe EventBridge a un consumer con velocità di elaborazione effettiva condivisa (iteratore standard) o a un consumer con velocità di elaborazione effettiva dedicata e fan-out avanzato.

Per gli iteratori standard, EventBridge esegue il polling di ogni partizione nel flusso Kinesis per i record utilizzando il protocollo HTTP. La pipe condivide la velocità di lettura effettiva con altri consumer della partizione.

Per ridurre al minimo la latenza e massimizzare la velocità di lettura effettiva, puoi creare un consumer di flussi di dati con fan-out avanzato. I consumer di flussi ottengono una connessione dedicata a ciascuna partizione che non ha alcun impatto su altre applicazioni che leggono dal flusso. La velocità di elaborazione effettiva dedicata può risultare utile se hai molte applicazioni che leggono gli stessi dati oppure se stai elaborando nuovamente un flusso con record di grandi dimensioni. Kinesis invia i record a EventBridge tramite HTTP/2. Per informazioni dettagliate su flussi di dati Kinesis, consulta Lettura dei dati dal flusso di dati Amazon Kinesis.

Esempio di evento

L'evento di esempio seguente mostra le informazioni ricevute dalla pipe. È possibile utilizzare questo evento per creare e filtrare i modelli di eventi o per definire la trasformazione degli input. Non tutti i campi possono essere filtrati. Per ulteriori informazioni su quali campi è possibile filtrare, consulta ???.


```
[
  {
    "kinesisSchemaVersion": "1.0",
    "partitionKey": "1",
    "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
    "data": "SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "approximateArrivalTimestamp": 1545084650.987
    "eventSource": "aws:kinesis",
    "eventVersion": "1.0",
    "eventID":
 "shardId-000000000006:49590338271490256608559692538361571095921575989136588898",
    "eventName": "aws:kinesis:record",
    "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
    "awsRegion": "us-east-2",
    "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
  },
  {
    "kinesisSchemaVersion": "1.0",
    "partitionKey": "1",
    "sequenceNumber": "49590338271490256608559692540925702759324208523137515618",
    "data": "VGhpcyBpcyBvbmx5IGEgdGVzdC4=",
    "approximateArrivalTimestamp": 1545084711.166
    "eventSource": "aws:kinesis",
    "eventVersion": "1.0",
    "eventID":
 "shardId-000000000006:49590338271490256608559692540925702759324208523137515618",
    "eventName": "aws:kinesis:record",
```

```
    "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
    "awsRegion": "us-east-2",
    "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
  }
]
```

Flussi di polling e batching

EventBridge esegue il polling di partizioni nel flusso Kinesis per record con una frequenza di base di quattro volte al secondo. Quando sono disponibili dei record, EventBridge elabora l'evento e attende il risultato. Se l'elaborazione ha esito positivo, EventBridge riprende a eseguire il polling fino a che non riceve più record.

Per impostazione predefinita, EventBridge richiama la pipe non appena i record sono disponibili. Se il batch che EventBridge legge dall'origine contiene un solo record, viene elaborato solo un evento. Per evitare di elaborare pochi record, puoi indicare alla pipe di memorizzare nel buffer i record per un massimo di cinque minuti configurando un periodo di batching. Prima di richiamare gli eventi, EventBridge continua a leggere i record dall'origine fino a quando non ha raccolto un batch completo, fino alla scadenza del periodo di batching o fino a quando il batch non ha raggiunto il limite del payload di 6 MB.

È anche possibile aumentare la simultaneità elaborando più batch da ogni partizione in parallelo. EventBridge può elaborare fino a 10 batch contemporaneamente in ogni partizione. Se aumenti il numero di batch simultanei per partizione, EventBridge garantisce comunque l'ordine di elaborazione a livello di chiave di partizione.

Configura l'impostazione `ParallelizationFactor` per elaborare una partizione di un flusso di dati Kinesis o DynamoDB con più esecuzioni di pipe simultanee. È possibile specificare il numero di batch simultanei di cui EventBridge esegue il polling da una partizione tramite un fattore di parallelizzazione compreso tra 1 (predefinito) e 10. Ad esempio, se imposti `ParallelizationFactor` su 2, possono esserci al massimo 200 esecuzioni di pipe EventBridge simultanee per elaborare 100 partizioni di dati Kinesis. Ciò permette di aumentare la velocità effettiva di elaborazione quando il volume di dati non è stabile e `IteratorAge` è alto. Si noti che il fattore di parallelizzazione non funzionerà se si utilizza l'aggregazione Kinesis.

Posizioni di partenza di polling e flussi

Tieni presente che il polling di origine dei flussi durante la creazione e gli aggiornamenti della pipe alla fine è coerente.

- Durante la creazione della pipe, potrebbero essere necessari alcuni minuti per l'avvio degli eventi di polling dal flusso.
- Durante gli aggiornamenti della pipe per la configurazione del polling di origine, potrebbero essere necessari alcuni minuti per interrompere e riavviare gli eventi di polling dal flusso.

Ciò significa che se specifichi `LATEST` come posizione iniziale del flusso, la pipe potrebbe perdere degli eventi inviati durante la creazione o gli aggiornamenti della pipe. Per assicurarti di non perdere alcun evento, specifica la posizione di partenza del flusso come `TRIM_HORIZON` o `AT_TIMESTAMP`.

Segnalazione errori articoli batch

Quando EventBridge utilizza ed elabora i dati di streaming da un'origine, per impostazione predefinita imposta i checkpoint al numero di sequenza più alto di un batch solo quando il batch è riuscito completamente. Per evitare di rielaborare i messaggi correttamente elaborati in un batch non riuscito, puoi configurare l'arricchimento o la destinazione in modo da restituire un oggetto che indichi quali messaggi hanno avuto esito positivo e quali non. Questa operazione è nota come risposta batch parziale.

Per ulteriori informazioni, consulta ???.

Condizioni di batch riuscito e non riuscito

EventBridge considera un batch come riuscito completamente se viene restituito uno dei seguenti elementi:

- Un elenco `batchItemFailure` vuoto
- Un `batchItemFailure` elenco nullo
- Un vuoto `EventResponse`
- Un valore nullo `EventResponse`

EventBridge considera un batch come non riuscito completamente se viene restituito uno dei seguenti elementi:

- Una stringa `itemIdentifier` vuota
- Un valore nullo `itemIdentifier`
- Un `itemIdentifier` con un nome chiave errato

EventBridge esegue nuovi tentativi per gli errori in base alla strategia di ripetizione.

Broker di messaggi Amazon MQ come origine

Puoi utilizzare EventBridge Pipes per ricevere record da un broker di messaggi Amazon MQ. Puoi eventualmente filtrare o migliorare questi record prima di inviarli a una delle destinazioni disponibili per l'elaborazione. Esistono impostazioni specifiche di Amazon MQ che puoi scegliere quando configuri una pipe. EventBridge Pipes mantiene l'ordine dei record dal broker di messaggi quando invia i dati alla destinazione.

Amazon MQ è un servizio gestito di broker dei messaggi per Apache ActiveMQ e RabbitMQ. Un broker di messaggi consente alle applicazioni e ai componenti software di comunicare utilizzando vari linguaggi di programmazione, sistemi operativi e protocolli di messaggistica formali con argomenti o code come destinazioni di eventi.

Amazon MQ può anche gestire automaticamente istanze di Amazon Elastic Compute Cloud (Amazon EC2) mediante l'installazione dei broker ActiveMQ o RabbitMQ. Dopo l'installazione, un broker fornisce diverse topologie di rete e altre esigenze di infrastruttura alle istanze.

L'origine Amazon MQ presenta le seguenti restrizioni di configurazione:

- Cross account: EventBridge non supporta l'elaborazione su più account. Non puoi utilizzarlo EventBridge per elaborare i record da un broker di messaggi Amazon MQ che si trova in un altro AWS account.
- Autenticazione: per ActiveMQ, è supportato solo SimpleAuthenticationPluginActiveMQ. Per RabbitMQ è supportato solo il meccanismo di autenticazione PLAIN. Per gestire le credenziali, usa AWS Secrets Manager. Per ulteriori informazioni sull'autenticazione di ActiveMQ, consulta Integrazione di broker ActiveMQ con LDAP nella Guida per gli sviluppatori di Amazon MQ.
- Quota di connessione: i broker hanno un numero massimo di connessioni consentite per ogni protocollo a livello di connessione. Questa quota si basa sul tipo di istanza del broker. Per ulteriori informazioni, consulta la sezione Broker di *Quote in Amazon MQ* nella Guida per gli sviluppatori di Amazon MQ.
- Connettività: puoi creare broker in un cloud privato virtuale (VPC) pubblico o privato. Per i VPC privati, la pipe deve accedere al VPC per poter ricevere messaggi.
- Destinazioni eventi: sono supportate solo le destinazioni di code. Tuttavia, puoi utilizzare un argomento virtuale, che si comporta come argomento internamente e come coda esternamente quando interagisce con le pipe. Per ulteriori informazioni, consulta Destinazioni virtuali sul sito Web di Apache ActiveMQ e Host virtuali sul sito Web di RabbitMQ.

- Topologia di rete: per ActiveMQ è supportato un solo broker a istanza singola o in standby per ogni pipe. Per RabbitMQ è supportata una sola implementazione di cluster o broker a istanza singola per ogni pipe. I broker a istanza singola richiedono un endpoint di failover. Per ulteriori informazioni su queste modalità di implementazione di broker, consulta Architettura del broker ActiveMQ e Architettura del broker RabbitMQ nella Guida per gli sviluppatori di Amazon MQ.
- Protocolli: i protocolli supportati dipendono dall'integrazione di Amazon MQ utilizzata.
  - Per le integrazioni ActiveMQ EventBridge , utilizza OpenWire il protocollo /Java Message Service (JMS) per consumare i messaggi. L'uso di messaggi non è supportato in nessun altro protocollo. EventBridge supporta solo le BytesMessageoperazioni TextMessagee all'interno del protocollo JMS. Per ulteriori informazioni sul OpenWire protocollo, vedere OpenWireil sito Web di Apache ActiveMQ.
  - Per le integrazioni RabbitMQ, EventBridge utilizza il protocollo AMQP 0-9-1 per consumare i messaggi. Non sono supportati altri protocolli per l'utilizzo dei messaggi. Per ulteriori informazioni sull'implementazione del protocollo AMQP 0-9-1 in RabbitMQ, consulta la Guida di riferimento completa di AMQP 0-9-1 sul sito web di RabbitMQ.

EventBridge supporta automaticamente le versioni più recenti di ActiveMQ e RabbitMQ supportate da Amazon MQ. Per le ultime versioni supportate, consulta le Note di rilascio di Amazon MQ nella Guida per gli sviluppatori di Amazon MQ.

> **Note**
>
> Per impostazione predefinita, Amazon MQ prevede un periodo di manutenzione settimanale per i broker. Durante tale periodo, i broker non sono disponibili. Per i broker senza standby, EventBridge non elaborerà i messaggi fino alla fine della finestra.

Eventi di esempio

L'evento di esempio seguente mostra le informazioni ricevute dalla pipe. È possibile utilizzare questo evento per creare e filtrare i modelli di eventi o per definire la trasformazione degli input. Non tutti i campi possono essere filtrati. Per ulteriori informazioni su quali campi è possibile filtrare, consulta ???.

ActiveMQ

```
[
```

```
  {
    "eventSource": "aws:amq",
    "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
    "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
    "messageType": "jms/text-message",
    "data": "QUJDOkFBQUE=",
    "connectionId": "myJMSCoID",
    "redelivered": false,
    "destination": {
      "physicalname": "testQueue"
    },
    "timestamp": 1598827811958,
    "brokerInTime": 1598827811958,
    "brokerOutTime": 1598827811959
  },
  {
    "eventSource": "aws:amq",
    "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
    "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
    "messageType": "jms/bytes-message",
    "data": "3DTOOW7crj51prgVLQaGQ82S48k=",
    "connectionId": "myJMSCoID1",
    "persistent": false,
    "destination": {
      "physicalname": "testQueue"
    },
    "timestamp": 1598827811958,
    "brokerInTime": 1598827811958,
    "brokerOutTime": 1598827811959
  }
]
```

RabbitMQ

```
[
  {
    "eventSource": "aws:rmq",
    "eventSourceArn": "arn:aws:mq:us-
west-2:111122223333:broker:pizzaBroker:b-9bcfa592-423a-4942-879d-eb284b418fc8",
```

```
    "eventSourceKey": "pizzaQueue::/",
    "basicProperties": {
      "contentType": "text/plain",
      "contentEncoding": null,
      "headers": {
        "header1": {
          "bytes": [
            118,
            97,
            108,
            117,
            101,
            49
          ]
        },
        "header2": {
          "bytes": [
            118,
            97,
            108,
            117,
            101,
            50
          ]
        },
        "numberInHeader": 10
      },
      "deliveryMode": 1,
      "priority": 34,
      "correlationId": null,
      "replyTo": null,
      "expiration": "60000",
      "messageId": null,
      "timestamp": "Jan 1, 1970, 12:33:41 AM",
      "type": null,
      "userId": "AIDACKCEVSQ6C2EXAMPLE",
      "appId": null,
      "clusterId": null,
      "bodySize": 80
    },
    "redelivered": false,
    "data": "eyJ0aW1lb3V0IjowLCJkYXRhIjoiQ1pybWYwR3c4T3Y0YnFMUXhENEUifQ=="
  }
```

```
]
```

Gruppo di consumer

Per interagire con Amazon MQ, EventBridge crea un gruppo di consumatori in grado di leggere i dati dei tuoi broker Amazon MQ. Il gruppo di consumer viene creato con lo stesso ID dell'UUID della pipe.

Per i sorgenti Amazon MQ, raggruppa EventBridge i record in batch e li invia alla tua funzione in un unico payload. Per controllare il comportamento, puoi configurare il periodo di batching e le dimensioni del batch. EventBridge estrae i messaggi fino a quando non si verifica una delle seguenti condizioni:

- I record elaborati raggiungono la dimensione di payload massima di 6 MB.
- Il periodo di batching scade.
- Il numero di record raggiunge la dimensione di batch massima.

EventBridge converte il batch in un unico payload e quindi richiama la funzione. I messaggi non sono persistenti né deserializzati. Il gruppo di consumer li recupera invece come BLOB di byte. Quindi li codifica in base64 in un payload JSON. Se la pipe restituisce un errore per uno qualsiasi dei messaggi di un batch, EventBridge riprova l'intero batch di messaggi fino a quando l'elaborazione non riesce o i messaggi scadono.

Configurazione della rete

Per impostazione predefinita, i broker Amazon MQ vengono creati con il flag `PubliclyAccessible` impostato su "false". Il broker riceve un indirizzo IP pubblico solo quando `PubliclyAccessible` è impostato su "true". Per l'accesso completo con la pipe, il broker deve utilizzare un endpoint pubblico o fornire l'accesso al VPC.

Se il tuo broker Amazon MQ non è accessibile al pubblico, EventBridge deve avere accesso alle risorse Amazon Virtual Private Cloud (Amazon VPC) associate al tuo broker.

- Per accedere al VPC dei tuoi broker Amazon MQ, EventBridge puoi utilizzare l'accesso a Internet in uscita per le sottoreti della tua fonte. Per le sottoreti pubbliche, deve essere un gateway NAT gestito. Per le sottoreti private può essere un gateway NAT o il proprio NAT. Assicurati che il NAT disponga di un indirizzo IP pubblico e possa connettersi a Internet.
- EventBridge Pipes supporta anche la distribuzione di eventi tramite AWS PrivateLink, che consente di inviare eventi da una fonte di eventi situata in un Amazon Virtual Private Cloud (Amazon VPC)

a una destinazione Pipes senza attraversare la rete Internet pubblica. È possibile utilizzare Pipes per eseguire il polling da Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogestito e Amazon MQ fonti che risiedono in una sottorete privata senza la necessità di implementare un gateway Internet, configurare regole firewall o configurare server proxy.

Per configurare un endpoint VPC, consulta Creare un endpoint VPC nella Guida per l'utente.AWS PrivateLink Per il nome del servizio, seleziona. `com.amazonaws.region.pipes-data`

È necessario configurare i gruppi di sicurezza Amazon VPC con le seguenti regole (come requisito minimo):

- Regole in entrata: consenti tutto il traffico sulla porta del broker Amazon MQ per i gruppi di sicurezza specificati per la tua origine.
- Regole in uscita: consenti tutto il traffico sulla porta 443 per tutte le destinazioni. Consenti tutto il traffico sulla porta del broker Amazon MQ per i gruppi di sicurezza specificati per la tua origine.

  Le porte del broker includono:

  - 9092 per testo non crittografato
  - 9094 per TLS
  - 9096 per SASL
  - 9098 per IAM

> **Note**
>
> La configurazione di Amazon VPC è individuabile tramite l'API di Amazon MQ. Non è necessario configurarlo durante l'installazione.

Argomento Streaming gestito da Amazon per Apache Kafka come origine

Puoi utilizzare EventBridge Pipes per ricevere record da un argomento Amazon Managed Streaming for Apache Kafka (Amazon MSK). Se lo desideri, puoi filtrare o migliorare questi record prima di inviarli a una delle destinazioni disponibili per l'elaborazione. Esistono impostazioni specifiche di Amazon MSK che puoi scegliere quando configuri una pipe. EventBridge Pipes mantiene l'ordine dei record dal broker di messaggi quando invia i dati alla destinazione.

Amazon MSK è un servizio completamente gestito che consente di creare ed eseguire applicazioni che utilizzano Apache Kafka per elaborare dati in streaming. Amazon MSK semplifica la configurazione, il dimensionamento e la gestione dei cluster che eseguono Kafka. Con Amazon MSK, puoi configurare la tua applicazione per più zone di disponibilità e per la sicurezza con AWS Identity and Access Management (IAM). Amazon MSK supporta più versioni open-source di Kafka.

Amazon MSK come sorgente funziona in modo simile all'utilizzo di Amazon Simple Queue Service (Amazon SQS) o Amazon Kinesis. EventBridgeesegue internamente il polling per individuare nuovi messaggi dall'origine e quindi richiama in modo sincrono la destinazione. EventBridge legge i messaggi in batch e li fornisce alla funzione come payload di eventi. La dimensione massima del batch è configurabile. (L'impostazione predefinita è 100 messaggi.)

Per i sorgenti basati su Apache Kafka, EventBridge supporta i parametri di controllo dell'elaborazione, come le finestre di batch e la dimensione del batch.

EventBridge legge i messaggi in sequenza per ogni partizione. Dopo aver EventBridge elaborato ogni batch, esegue il commit degli offset dei messaggi in quel batch. Se la destinazione della pipe restituisce un errore per uno qualsiasi dei messaggi di un batch, EventBridge riprova l'intero batch di messaggi fino alla riuscita dell'elaborazione o alla scadenza dei messaggi.

EventBridge invia il batch di messaggi nel caso in cui richiami la destinazione. Il payload evento contiene un array di messaggi. Ogni elemento dell'array contiene i dettagli dell'argomento e dell'identificatore della partizione Amazon MSK, insieme a una data/ora e a un messaggio con codifica base64.

Eventi di esempio

L'evento di esempio seguente mostra le informazioni ricevute dalla pipe. È possibile utilizzare questo evento per creare e filtrare i modelli di eventi o per definire la trasformazione degli input. Non tutti i campi possono essere filtrati. Per ulteriori informazioni su quali campi è possibile filtrare, consulta ???.

```
[
  {
    "eventSource": "aws:kafka",
    "eventSourceArn": "arn:aws:kafka:sa-east-1:123456789012:cluster/
vpc-2priv-2pub/751d2973-a626-431c-9d4e-d7975eb44dd7-2",
    "eventSourceKey": "mytopic-0",
    "topic": "mytopic",
    "partition": "0",
```

```
    "offset": 15,
    "timestamp": 1545084650987,
    "timestampType": "CREATE_TIME",
    "key":"abcDEFghiJKLmnoPQRstuVWXyz1234==",
    "value":"SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "headers": [
      {
        "headerKey": [
          104,
          101,
          97,
          100,
          101,
          114,
          86,
          97,
          108,
          117,
          101
        ]
      }
    ]
  }
]
```

Posizioni di partenza di polling e flussi

Tieni presente che il polling di origine dei flussi durante la creazione e gli aggiornamenti della pipe alla fine è coerente.

- Durante la creazione della pipe, potrebbero essere necessari alcuni minuti per l'avvio degli eventi di polling dal flusso.
- Durante gli aggiornamenti della pipe per la configurazione del polling di origine, potrebbero essere necessari alcuni minuti per interrompere e riavviare gli eventi di polling dal flusso.

Ciò significa che se specifichi `LATEST` come posizione iniziale del flusso, la pipe potrebbe perdere degli eventi inviati durante la creazione o gli aggiornamenti della pipe. Per garantire che nessun evento venga perso, specifica la posizione iniziale del flusso come `TRIM_HORIZON`.

Autenticazione cluster MSK

EventBridge necessita dell'autorizzazione per accedere al cluster Amazon MSK, recuperare record ed eseguire altre attività. Amazon MSK supporta diverse opzioni per il controllo dell'accesso del client al cluster MSK. Per ulteriori informazioni su quale metodo di autenticazione viene utilizzato, consulta ???.

Opzioni di accesso al cluster

- Accesso non autenticato
- Autenticazione SASL/SCRAM
- Autenticazione basata su ruoli IAM
- Autenticazione TLS reciproca
- Configurazione del segreto mTLS
- Come sceglie un broker bootstrap EventBridge

Accesso non autenticato

Consigliamo di utilizzare solo accessi non autenticati per lo sviluppo. L'accesso non autenticato funzionerà solo se l'autenticazione basata su ruoli IAM è disabilitata per il cluster.

Autenticazione SASL/SCRAM

Amazon MSK supporta l'autenticazione SASL/SCRAM (Simple Authentication and Security Layer/Salted Challenge Response Authentication Mechanism) con crittografia Transport Layer Security (TLS). Per connetterti EventBridge al cluster, memorizzi le credenziali di autenticazione (credenziali di accesso) in un luogo segreto. AWS Secrets Manager

Per ulteriori informazioni sull'uso di Secrets Manager, consulta Autenticazione nome utente e password con AWS Secrets Manager nella Guida per gli sviluppatori di Amazon Managed Streaming for Apache Kafka.

Amazon MSK non supporta l'autenticazione SASL/PLAIN.

Autenticazione basata su ruoli IAM

È possibile utilizzare IAM per autenticare l'identità dei client che si connettono al cluster MSK. Se l'autenticazione IAM è attiva sul cluster MSK e non si fornisce un segreto per l'autenticazione, per impostazione predefinita utilizza EventBridge automaticamente l'autenticazione IAM. Per creare e implementare policy IAM basate su utenti o ruoli, utilizza l'API o la console IAM. Per ulteriori

informazioni, consulta il controllo accessi IAM nella Guida per sviluppatori Amazon Managed Streaming for Apache Kafka.

Per consentire la connessione EventBridge al cluster MSK, leggere i record ed eseguire altre azioni richieste, aggiungi le seguenti autorizzazioni al ruolo di esecuzione delle tue pipe.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:DescribeGroup",
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeTopic",
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeClusterDynamicConfiguration"
            ],
            "Resource": [
                "arn:aws:kafka:region:account-id:cluster/cluster-name/cluster-uuid",
                "arn:aws:kafka:region:account-id:topic/cluster-name/cluster-uuid/topic-
name",
                "arn:aws:kafka:region:account-id:group/cluster-name/cluster-
uuid/consumer-group-id"
            ]
        }
    ]
}
```

È possibile assegnare queste autorizzazioni a un cluster, un argomento e un gruppo specifici. Per ulteriori informazioni, consulta le operazioni Kafka di Amazon MSK nella Guida per sviluppatori Amazon Managed Streaming for Apache Kafka.

Autenticazione TLS reciproca

MTLS (Mutual TLS) fornisce l'autenticazione bidirezionale tra client e server. Il client invia un certificato al server affinché il server verifichi il client e il server invia un certificato al client affinché il client verifichi il server.

Per Amazon MSK, EventBridge funge da client. È possibile configurare un certificato client (come segreto in Secrets Manager) per l'autenticazione EventBridge con i broker del cluster MSK. Il

certificato client deve essere firmato da un'autorità di certificazione (CA) presente nel trust store del server. Il cluster MSK invia un certificato server con cui EventBridge autenticare i broker. EventBridge Il certificato del server deve essere firmato da una CA presente nel AWS trust store.

Amazon MSK non supporta i certificati server autofirmati, poiché tutti i broker di Amazon MSK utilizzano certificati pubblici firmati dalle CA di Amazon Trust Services, che per impostazione predefinita si affidano ai trust. EventBridge

Per ulteriori informazioni, su mTLS per Amazon MSK, consulta Autenticazione TLS reciproca nella Guida per sviluppatori Amazon Managed Streaming for Apache Kafka.

Configurazione del segreto mTLS

Il segreto CLIENT_CERTIFICATE_TLS_AUTH richiede un campo certificato e un campo chiave privata. Per una chiave privata crittografata, il segreto richiede una password per chiave privata. Il certificato e la chiave privata devono essere in formato PEM.

> **Note**
>
> EventBridge supporta gli algoritmi di crittografia a chiave privata PBES1 (ma non PBES2).

Il campo certificato deve contenere un elenco di certificati, a partire dal certificato client, seguito da qualsiasi certificato intermedio, per finire con il certificato root. Ogni certificato deve iniziare su una nuova riga con la struttura seguente:

```
-----BEGIN CERTIFICATE-----
        <certificate contents>
-----END CERTIFICATE-----
```

Secrets Manager supporta segreti fino a 65.536 byte, che è uno spazio sufficiente per lunghe catene di certificati.

La chiave privata deve essere in formato PKCS #8, con la struttura seguente:

```
-----BEGIN PRIVATE KEY-----
         <private key contents>
-----END PRIVATE KEY-----
```

Per una chiave privata crittografata, utilizza la struttura seguente:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
          <private key contents>
-----END ENCRYPTED PRIVATE KEY-----
```

Nell'esempio seguente viene mostrato il contenuto di un segreto per l'autenticazione mTLS utilizzando una chiave privata crittografata. Per una chiave privata crittografata, includi una password per chiave privata nel segreto.

```
{
 "privateKeyPassword": "testpassword",
 "certificate": "-----BEGIN CERTIFICATE-----
MIIE5DCCAsygAwIBAgIRAPJdwaFaNRrytHBto0j5BA0wDQYJKoZIhvcNAQELBQAw
...
j0Lh4/+1HfgyE2KlmII36dg4IMzNjAFEBZiCRoPimO40s1cRqtFHXoal0QQbIlxk
cmUuiAii9R0=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFgjCCA2qgAwIBAgIQdjNZd6uFf9hbNC5RdfmHrzANBgkqhkiG9w0BAQsFADBb
...
rQoiowbbk5wXCheYSANQIfTZ6weQTgiCHCCbuuMKNVS95FkXm0vqVD/YpXKwA/no
c8PH3PSoAaRwMMgOSA2ALJvbRz8mpg==
-----END CERTIFICATE-----",
 "privateKey": "-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFKzBVBgkqhkiG9w0BBQ0wSDAnBgkqhkiG9w0BBQwwGgQUiAFcK5hT/X7Kjmgp
...
QrSekqF+kWzmB6nAfSzgO9IaoAaytLvNgGTckWeUkWn/V0Ck+LdGUXzAC4RxZnoQ
zp2mwJn2NYB7AZ7+imp0azDZb+8YG2aUCiyqb6PnnA==
-----END ENCRYPTED PRIVATE KEY-----"
}
```

Come sceglie un broker bootstrap EventBridge

EventBridge sceglie un broker di bootstrap in base ai metodi di autenticazione disponibili nel cluster e se fornisci un segreto per l'autenticazione. Se fornisci un segreto per MTL o SASL/SCRAM, EventBridge sceglie automaticamente quel metodo di autenticazione. Se non fornisci un segreto, EventBridge scegli il metodo di autenticazione più efficace attivo sul tuo cluster. Di seguito è riportato l'ordine di priorità in base al quale viene EventBridge selezionato un broker, dall'autenticazione più forte a quella più debole:

- mTLS (segreto fornito per mTLS)
- SASL/SCRAM (segreto fornito per SASL/SCRAM)

- IAM SASL (nessun segreto fornito e autenticazione IAM attiva)
- TLS non autenticato (nessun segreto fornito e autenticazione IAM non attiva)
- Testo semplice (nessun segreto fornito e autenticazione IAM e TLS non autenticato non attivi)

> **Note**
>
> Se non EventBridge riesce a connettersi al tipo di broker più sicuro, non tenta di connettersi a un tipo di broker diverso (più debole). Se desideri EventBridge scegliere un tipo di broker più debole, disattiva tutti i metodi di autenticazione più avanzati sul tuo cluster.

Configurazione della rete

EventBridge deve avere accesso alle risorse Amazon Virtual Private Cloud (Amazon VPC) associate al tuo cluster Amazon MSK.

- Per accedere al VPC del tuo cluster Amazon MSK, EventBridge puoi utilizzare l'accesso a Internet in uscita per le sottoreti della tua fonte. Per le sottoreti pubbliche, deve essere un gateway NAT gestito. Per le sottoreti private può essere un gateway NAT o il proprio NAT. Assicurati che il NAT disponga di un indirizzo IP pubblico e possa connettersi a Internet.
- EventBridge Pipes supporta anche la distribuzione di eventi tramite AWS PrivateLink, che consente di inviare eventi da una fonte di eventi situata in un Amazon Virtual Private Cloud (Amazon VPC) a una destinazione Pipes senza dover attraversare la rete Internet pubblica. È possibile utilizzare Pipes per eseguire il polling da Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogestito e Amazon MQ fonti che risiedono in una sottorete privata senza la necessità di implementare un gateway Internet, configurare regole firewall o configurare server proxy.

  Per configurare un endpoint VPC, consulta Creare un endpoint VPC nella Guida per l'utente.AWS PrivateLink Per il nome del servizio, seleziona. `com.amazonaws.region.pipes-data`

È necessario configurare i gruppi di sicurezza Amazon VPC con le seguenti regole (come requisito minimo):

- Regole in entrata: consenti tutto il traffico sulla porta del broker Amazon MSK per i gruppi di sicurezza specificati per la tua origine.

- Regole in uscita: consenti tutto il traffico sulla porta 443 per tutte le destinazioni. Consenti tutto il traffico sulla porta del broker Amazon MSK per i gruppi di sicurezza specificati per la tua origine.

  Le porte del broker includono:

  - 9092 per testo non crittografato
  - 9094 per TLS
  - 9096 per SASL
  - 9098 per IAM

> **Note**
>
> La configurazione di Amazon VPC è individuabile tramite l'API Amazon MSK. Non è necessario configurarlo durante l'installazione.

ID gruppo di consumer personalizzabile

Quando configuri Apache Kafka come origine, puoi specificare un ID gruppo di consumer. Questo ID gruppo di consumer è un identificatore esistente per il gruppo di consumer Apache Kafka a cui vuoi che la tua pipe aderisca. È possibile utilizzare questa funzionalità per migrare qualsiasi configurazione di elaborazione dei record di Apache Kafka in corso da altri consumatori a. EventBridge

Se specifichi l'ID gruppo di consumer e sono presenti altri poller attivi in quel gruppo di consumer, Apache Kafka distribuisce i messaggi a tutti i consumer. In altre parole, EventBridge non riceve tutti i messaggi relativi all'argomento Apache Kafka. Se desideri EventBridge gestire tutti i messaggi dell'argomento, disattiva tutti gli altri sondaggi in quel gruppo di consumatori.

Inoltre, se si specifica un ID di gruppo di consumatori e Apache Kafka trova un gruppo di consumatori esistente valido con lo stesso ID, EventBridge ignora il parametro relativo alla `StartingPosition` pipe. EventBridge Inizia invece a elaborare i record in base all'offset impegnato del gruppo di consumatori. Se si specifica un ID del gruppo di consumatori e Apache Kafka non riesce a trovare un gruppo di consumatori esistente, EventBridge configura l'origine con quello specificato. `StartingPosition`

L'ID gruppo di consumer che specifichi deve essere univoco tra tutte le origini eventi di Apache Kafka. Dopo aver creato una pipe con l'ID gruppo di consumer specificato, non sarà più possibile aggiornare questo valore.

Dimensionamento automatico dell'origine di Amazon MSK

Quando crei inizialmente una fonte Amazon MSK, EventBridge assegna un consumatore all'elaborazione di tutte le partizioni nell'argomento Apache Kafka. Ogni consumatore ha più processori in esecuzione in parallelo per gestire carichi di lavoro più elevati. Inoltre, aumenta o riduce EventBridge automaticamente il numero di consumatori, in base al carico di lavoro. Per preservare l'ordinamento dei messaggi in ogni partizione, il numero massimo di consumatori è un consumatore per ogni partizione dell'argomento.

A intervalli di un minuto, EventBridge valuta il ritardo di compensazione tra i consumatori di tutte le partizioni dell'argomento. Se il ritardo è troppo elevato, la partizione riceve i messaggi più velocemente di quanto possa elaborarli. EventBridge Se necessario, EventBridge aggiunge o rimuove utenti dall'argomento. Il processo di dimensionamento di aggiunta o rimozione dei consumatori avviene entro tre minuti dalla valutazione.

Se il target è sovraccarico, EventBridge riduce il numero di consumatori. Questa azione riduce il carico di lavoro sulla pipe riducendo il numero di messaggi che i consumer possono recuperare e inviare alla pipe.

Streaming di Apache Kafka come sorgente

Apache Kafka è una piattaforma di streaming di eventi open source che supporta carichi di lavoro come pipeline di dati e analisi dei dati di streaming. Puoi utilizzare Amazon Managed Streaming for Apache Kafka (Amazon MSK) o un cluster Apache Kafka autogestito. In AWS terminologia, un cluster autogestito si riferisce a qualsiasi cluster Apache Kafka non ospitato da. AWS Ciò include sia i cluster gestiti dall'utente, sia quelli ospitati da un provider di terze parti, ad esempio, o. Confluent CloudCloudKarafkaRedpanda

Per ulteriori informazioni su altre opzioni di AWS hosting per il tuo cluster, consulta le migliori pratiche per l'esecuzione di Apache Kafka AWS sul AWS blog Big Data.

Apache Kafka come sorgente funziona in modo simile all'utilizzo di Amazon Simple Queue Service (Amazon SQS) o Amazon Kinesis. EventBridgeesegue internamente il polling per individuare nuovi messaggi dall'origine e quindi richiama in modo sincrono la destinazione. EventBridge legge i messaggi in batch e li fornisce alla funzione come payload di eventi. La dimensione massima del batch è configurabile. (L'impostazione predefinita è 100 messaggi.)

Per i sorgenti basati su Apache Kafka, EventBridge supporta i parametri di controllo dell'elaborazione, come le finestre di batch e la dimensione del batch.

EventBridge invia il batch di messaggi nel parametro dell'evento quando richiama la pipe. Il payload evento contiene un array di messaggi. Ogni elemento dell'array contiene i dettagli dell'argomento Apache Kafka e dell'identificatore di partizione Apache Kafka, insieme a un timestamp e a un messaggio con codifica base64.

Eventi di esempio

L'evento di esempio seguente mostra le informazioni ricevute dalla pipe. È possibile utilizzare questo evento per creare e filtrare i modelli di eventi o per definire la trasformazione degli input. Non tutti i campi possono essere filtrati. Per ulteriori informazioni su quali campi è possibile filtrare, consulta ???.


```
[
  {
    "eventSource": "SelfManagedKafka",
    "bootstrapServers": "b-2.demo-cluster-1.a1bcde.c1.kafka.us-
east-1.amazonaws.com:9092,b-1.demo-cluster-1.a1bcde.c1.kafka.us-
east-1.amazonaws.com:9092",
    "eventSourceKey": "mytopic-0",
    "topic": "mytopic",
    "partition": 0,
    "offset": 15,
    "timestamp": 1545084650987,
    "timestampType": "CREATE_TIME",
    "key":"abcDEFghiJKLmnoPQRstuVWXyz1234==",
    "value":"SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "headers": [
      {
        "headerKey": [
          104,
          101,
          97,
          100,
          101,
          114,
          86,
          97,
          108,
          117,
          101
```

```
        ]
      }
    ]
  }
]
```

Autenticazione con il cluster Apache Kafka

EventBridge Pipes supporta diversi metodi di autenticazione con il cluster Apache Kafka autogestito. Assicurati di configurare il cluster Apache Kafka per utilizzare uno di questi metodi di autenticazione supportati. Per ulteriori informazioni sulla sicurezza con Apache Kafka, consulta la sezione Sicurezza della documentazione di Apache Kafka.

Accesso VPC

Se utilizzi un ambiente Apache Kafka autogestito in cui solo gli utenti Apache Kafka all'interno del tuo VPC hanno accesso ai tuoi broker Apache Kafka, devi configurare Amazon Virtual Private Cloud (Amazon VPC) nel sorgente Apache Kafka.

Autenticazione SASL/SCRAM

EventBridge Pipes supporta l'autenticazione SASL/SCRAM (Simple Authentication and Security Layer/Salted Challenge Response Authentication Mechanism) con crittografia Transport Layer Security (TLS). EventBridge Pipes invia le credenziali crittografate per l'autenticazione con il cluster. Per ulteriori informazioni sull'autenticazione SASL/SCRAM, consultare RFC 5802.

EventBridge Pipes supporta l'autenticazione SASL/PLAIN con crittografia TLS. Con l'autenticazione SASL/PLAIN, EventBridge Pipes invia le credenziali come testo non crittografato al server.

Per l'autenticazione SASL, è necessario archiviare le credenziali di accesso come segreto in AWS Secrets Manager.

Autenticazione TLS reciproca

MTLS (Mutual TLS) fornisce l'autenticazione bidirezionale tra client e server. Il client invia un certificato al server affinché il server verifichi il client e il server invia un certificato al client affinché il client verifichi il server.

In Apache Kafka autogestito, Pipes funge da client. EventBridge Configurate un certificato client (come segreto in Secrets Manager) per autenticare EventBridge Pipes con i vostri broker Apache Kafka. Il certificato client deve essere firmato da un'autorità di certificazione (CA) presente nel trust store del server.

Il cluster Apache Kafka invia un certificato server a Pipes per autenticare i broker Apache Kafka con EventBridge Pipes. EventBridge Il certificato del server può essere un certificato CA pubblico o un certificato CA/autofirmato privato. Il certificato CA pubblico deve essere firmato da una CA presente nel trust store di Pipes. EventBridge Per un certificato CA privato/autofirmato, si configura il certificato CA principale del server (come segreto in Secrets Manager). EventBridge Pipes utilizza il certificato root per verificare i broker Apache Kafka.

Per ulteriori informazioni su mTLS, consulta Introduzione all'autenticazione Mutual TLS per Amazon MSK come origine.

Configurazione del segreto del certificato client

Il segreto CLIENT_CERTIFICATE_TLS_AUTH richiede un campo certificato e un campo chiave privata. Per una chiave privata crittografata, il segreto richiede una password per chiave privata. Il certificato e la chiave privata devono essere in formato PEM.

> **Note**
>
> EventBridge Pipes supporta gli algoritmi di crittografia a chiave privata PBES1 (ma non PBES2).

Il campo certificato deve contenere un elenco di certificati, a partire dal certificato client, seguito da qualsiasi certificato intermedio, per finire con il certificato root. Ogni certificato deve iniziare su una nuova riga con la struttura seguente:

```
-----BEGIN CERTIFICATE-----
        <certificate contents>
-----END CERTIFICATE-----
```

Secrets Manager supporta segreti fino a 65.536 byte, che è uno spazio sufficiente per lunghe catene di certificati.

La chiave privata deve essere in formato PKCS #8, con la struttura seguente:

```
-----BEGIN PRIVATE KEY-----
         <private key contents>
-----END PRIVATE KEY-----
```

Per una chiave privata crittografata, utilizza la struttura seguente:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
          <private key contents>
-----END ENCRYPTED PRIVATE KEY-----
```

Nell'esempio seguente viene mostrato il contenuto di un segreto per l'autenticazione mTLS utilizzando una chiave privata crittografata. Per una chiave privata crittografata, includere la password per chiave privata nel segreto.

```
{
 "privateKeyPassword": "testpassword",
 "certificate": "-----BEGIN CERTIFICATE-----
MIIE5DCCAsygAwIBAgIRAPJdwaFaNRrytHBto0j5BA0wDQYJKoZIhvcNAQELBQAw
...
j0Lh4/+1HfgyE2KlmII36dg4IMzNjAFEBZiCRoPimO40s1cRqtFHXoal0QQbIlxk
cmUuiAii9R0=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFgjCCA2qgAwIBAgIQdjNZd6uFf9hbNC5RdfmHrzANBgkqhkiG9w0BAQsFADBb
...
rQoiowbbk5wXCheYSANQIfTZ6weQTgiCHCCbuuMKNVS95FkXm0vqVD/YpXKwA/no
c8PH3PSoAaRwMMgOSA2ALJvbRz8mpg==
-----END CERTIFICATE-----",
 "privateKey": "-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFKzBVBgkqhkiG9w0BBQ0wSDAnBgkqhkiG9w0BBQwwGgQUiAFcK5hT/X7Kjmgp
...
QrSekqF+kWzmB6nAfSzgO9IaoAaytLvNgGTckWeUkWn/V0Ck+LdGUXzAC4RxZnoQ
zp2mwJn2NYB7AZ7+imp0azDZb+8YG2aUCiyqb6PnnA==
-----END ENCRYPTED PRIVATE KEY-----"
}
```

Configurazione del segreto del certificato CA root del server

Questo segreto viene creato se i broker Apache Kafka utilizzano la crittografia TLS con certificati firmati da una CA privata. È possibile utilizzare la crittografia TLS per l'autenticazione VPC, SASL/SCRAM, SASL/PLAIN o mTLS.

Il segreto del certificato CA root del server richiede un campo che contiene il certificato CA root del broker Kafka in formato PEM. Il seguente esempio illustra la struttura del segreto.

```
{
     "certificate": "-----BEGIN CERTIFICATE-----
```

```
MIID7zCCAtegAwIBAgIBADANBgkqhkiG9w0BAQsFADCBmDELMAkGA1UEBhMCVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxJTAjBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluY4xOzA5BgNVBAMTMlN0YXJmaWVs
ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dG...
-----END CERTIFICATE-----"
```

Configurazione della rete

Se utilizzi un ambiente Apache Kafka autogestito che utilizza la connettività VPC privata, EventBridge devi avere accesso alle risorse Amazon Virtual Private Cloud (Amazon VPC) associate ai tuoi broker Apache Kafka.

- Per accedere al VPC del tuo cluster Apache Kafka, EventBridge puoi utilizzare l'accesso a Internet in uscita per le sottoreti della tua fonte. Per le sottoreti pubbliche, deve essere un gateway NAT gestito. Per le sottoreti private può essere un gateway NAT o il proprio NAT. Assicurati che il NAT disponga di un indirizzo IP pubblico e possa connettersi a Internet.
- EventBridge Pipes supporta anche la distribuzione di eventi tramite AWS PrivateLink, che consente di inviare eventi da una fonte di eventi situata in un Amazon Virtual Private Cloud (Amazon VPC) a una destinazione Pipes senza dover attraversare la rete Internet pubblica. È possibile utilizzare Pipes per eseguire il polling da Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogestito e Amazon MQ fonti che risiedono in una sottorete privata senza la necessità di implementare un gateway Internet, configurare regole firewall o configurare server proxy.

  Per configurare un endpoint VPC, consulta Creare un endpoint VPC nella Guida per l'utente.AWS PrivateLink Per il nome del servizio, seleziona. `com.amazonaws.region.pipes-data`

È necessario configurare i gruppi di sicurezza Amazon VPC con le seguenti regole (come requisito minimo):

- Regole in entrata: consenti tutto il traffico sulla porta del broker Apache Kafka per i gruppi di sicurezza specificati per la tua origine.
- Regole in uscita: consenti tutto il traffico sulla porta 443 per tutte le destinazioni. Consenti tutto il traffico sulla porta del broker Apache Kafka per i gruppi di sicurezza specificati per la tua origine.

  Le porte del broker includono:
  - 9092 per testo non crittografato
  - 9094 per TLS

- 9096 per SASL
- 9098 per IAM

Scalabilità automatica per i consumatori con sorgenti Apache Kafka

Quando crei inizialmente un sorgente Apache Kafka, EventBridge assegna un consumatore all'elaborazione di tutte le partizioni nell'argomento Kafka. Ogni consumatore ha più processori in esecuzione in parallelo per gestire carichi di lavoro più elevati. Inoltre, aumenta o riduce EventBridge automaticamente il numero di consumatori, in base al carico di lavoro. Per preservare l'ordinamento dei messaggi in ogni partizione, il numero massimo di consumatori è un consumatore per ogni partizione dell'argomento.

A intervalli di un minuto, EventBridge valuta il ritardo di compensazione tra i consumatori di tutte le partizioni dell'argomento. Se il ritardo è troppo elevato, la partizione riceve i messaggi più velocemente di quanto possa elaborarli. EventBridge Se necessario, EventBridge aggiunge o rimuove utenti dall'argomento. Il processo di dimensionamento di aggiunta o rimozione dei consumatori avviene entro tre minuti dalla valutazione.

Se il target è sovraccarico, EventBridge riduce il numero di consumatori. Questa operazione riduce il carico di lavoro sulla funzione riducendo il numero di messaggi che i consumer possono recuperare e inviare alla funzione.

Amazon Simple Queue Service come origine

Puoi utilizzare EventBridge Pipes per ricevere record da una coda Amazon SQS. Se lo desideri, puoi filtrare o migliorare questi record prima di inviarli a una destinazione disponibile per l'elaborazione.

Puoi utilizzare una pipe per elaborare i messaggi in una coda Amazon Simple Queue Service (Amazon SQS). EventBridge Le pipe supportano le code standard e le code FIFO (First-in, First-Out). Con Amazon SQS, è possibile scaricare le attività da un componente della applicazione inviandole a una coda ed elaborandole in modo asincrono.

EventBridge interroga la coda e richiama la pipe in modo sincrono con un evento che contiene messaggi in coda. EventBridge legge i messaggi in batch e richiama la pipe una volta per ogni batch. Quando la pipe elabora correttamente un batch, EventBridge elimina i relativi messaggi dalla coda.

Per impostazione predefinita, EventBridge esegue il polling simultaneo di un massimo di 10 messaggi nella coda e invia il batch alla pipe. Per evitare di richiamare la pipe con pochi record, è possibile

indicare all'origine dell'evento di memorizzare nel buffer i record per un massimo di cinque minuti configurando un periodo di batch. Prima di richiamare la pipe, EventBridge continua a eseguire il polling dei messaggi dalla coda standard di Amazon SQS fino a quando non si verifica una delle seguenti situazioni:

- Il periodo di batch scade.
- Viene raggiunta la quota della dimensione del payload di invocazione.
- Viene raggiunta la dimensione massima configurata del batch.

> **Note**
>
> Se utilizzi una finestra batch e la tua coda Amazon SQS contiene poco traffico, EventBridge potresti attendere fino a 20 secondi prima di richiamare la tua pipe. Lo stesso vale anche se imposti un periodo di batch inferiore a 20 secondi. Per le code FIFO, i record contengono attributi aggiuntivi correlati alla deduplicazione e al sequenziamento.

Quando EventBridge legge un batch, i messaggi rimangono in coda ma vengono nascosti per tutta la durata del timeout di visibilità della coda. Se la pipe elabora correttamente il batch, EventBridge elimina i messaggi dalla coda. Per impostazione predefinita, se la pipe rileva un errore durante l'elaborazione di un batch, tutti i messaggi in quel batch diventano nuovamente visibili nella coda. Per questo motivo, il codice della pipe deve riuscire a elaborare lo stesso messaggio più volte, senza effetti collaterali indesiderati. È possibile modificare questo comportamento di rielaborazione includendo, nella risposta della pipe, errori di elementi batch. Nell'esempio seguente viene illustrato un evento per un batch di due messaggi.

Eventi di esempio

L'evento di esempio seguente mostra le informazioni ricevute dalla pipe. È possibile utilizzare questo evento per creare e filtrare i modelli di eventi o per definire la trasformazione degli input. Non tutti i campi possono essere filtrati. Per ulteriori informazioni su quali campi è possibile filtrare, consulta ???.

Coda standard

```
[
  {
    "messageId": "059f36b4-87a3-44ab-83d2-661975830a7d",
```

```
    "receiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1545082649183",
      "SenderId": "AIDAIENQZJOLO23YVJ4VO",
      "ApproximateFirstReceiveTimestamp": "1545082649185"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:my-queue",
    "awsRegion": "us-east-2"
  },
  {
    "messageId": "2e1424d4-f796-459a-8184-9c92662be6da",
    "receiptHandle": "AQEBzWwaftRI0KuVm4tP+/7q1rGgNqicHq...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1545082650636",
      "SenderId": "AIDAIENQZJOLO23YVJ4VO",
      "ApproximateFirstReceiveTimestamp": "1545082650649"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:my-queue",
    "awsRegion": "us-east-2"
  }
]
```

Coda FIFO

```
[
  {
    "messageId": "11d6ee51-4cc7-4302-9e22-7cd8afdaadf5",
    "receiptHandle": "AQEBBX8nesZEXmkhsmZeyIE8iQAMig7qw...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1573251510774",
      "SequenceNumber": "18849496460467696128",
```

```
      "MessageGroupId": "1",
      "SenderId": "AIDAIO23YVJENQZJOL4VO",
      "MessageDeduplicationId": "1",
      "ApproximateFirstReceiveTimestamp": "1573251510774"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:fifo.fifo",
    "awsRegion": "us-east-2"
  }
]
```

Dimensionamento ed elaborazione

Per le code standard, EventBridge utilizza il polling lungo per interrogare una coda finché non diventa attiva. Quando i messaggi sono disponibili, EventBridge legge fino a cinque batch e li invia alla tua pipe. Se i messaggi sono ancora disponibili, EventBridge aumenta il numero di processi che leggono i batch fino a 300 istanze in più al minuto. Il numero massimo di batch che è possibile elaborare contemporaneamente con una pipe è 1.000.

Per le code FIFO, EventBridge invia i messaggi alla tua pipe nell'ordine in cui li riceve. Quando invii un messaggio a una coda FIFO, è necessario specificare un ID gruppo di messaggi. Amazon SQS semplifica il recapito di messaggi nello stesso gruppo a EventBridge, in ordine. EventBridge ordina i messaggi ricevuti in gruppi e invia solo un batch alla volta per gruppo. Se la pipe restituisce un errore, la pipe tenta tutti i tentativi sui messaggi interessati prima di EventBridge ricevere altri messaggi dallo stesso gruppo.

Configurazione di una coda da utilizzare con Pipes EventBridge

Crea una coda Amazon SQS da usare come origine eventi per la pipe.. Quindi configura la coda in modo che la pipe abbia il tempo di elaborare ogni batch di eventi e di riprovare in risposta EventBridge agli errori di throttling man mano che aumenta.

Per concedere alla pipe il tempo necessario per elaborare ogni batch di record, imposta il timeout visibilità della coda di origine su un tempo pari ad almeno sei volte il runtime combinato dei componenti di arricchimento e di destinazione della pipe. Il tempo supplementare consente di riprovare se la pipa viene EventBridge strozzata durante l'elaborazione di un batch precedente.

Se la pipe non dovesse riuscire a elaborare un messaggio più volte, Amazon SQS può inviarlo a una coda DLQ. Quando la pipe restituisce un errore, la EventBridge mantiene in coda. Dopo il timeout di

visibilità, EventBridge riceve nuovamente il messaggio. Per inviare messaggi a una seconda coda dopo un numero di ricevute, configurare una coda DLQ nella coda di origine.

> **Note**
>
> Assicurati di configurare la coda DLQ sulla coda di origine, non sulla pipe. La coda DLQ che configuri su una pipe viene utilizzata per la coda di chiamate asincrone della pipe, non per le code di origine.

Se la pipe restituisce un errore o non può essere richiamata perché la simultaneità è al massimo, l'elaborazione potrebbe avere esito positivo con ulteriori tentativi. Per offrire ai messaggi maggiori possibilità di essere elaborati prima di inviarli alla coda DLQ, imposta `maxReceiveCount` sulla policy di reindirizzamento della coda di origine su almeno 5.

Segnalazione errori articoli batch

Quando EventBridge utilizza ed elabora i dati in streaming da una fonte, per impostazione predefinita il checkpoint punta al numero di sequenza più alto di un batch, ma solo quando il batch ha esito positivo. Per evitare di rielaborare i messaggi correttamente elaborati in un batch non riuscito, puoi configurare l'arricchimento o la destinazione in modo da restituire un oggetto che indichi quali messaggi hanno avuto esito positivo e quali non. Questa operazione è nota come risposta batch parziale.

Per ulteriori informazioni, consulta ???.

Condizioni di successo e di errore

Se restituisci una delle seguenti condizioni, EventBridge considera un batch come un successo completo:

- Una `batchItemFailure` lista vuota
- Un `batchItemFailure` elenco nullo
- Un vuoto `EventResponse`
- Un valore nullo `EventResponse`

Se restituisci una delle seguenti condizioni, EventBridge considera un batch come un completo fallimento:

- Una stringa vuota `itemIdentifier`
- Un valore nullo `itemIdentifier`
- Un `itemIdentifier` con un nome chiave errato

EventBridge riprova gli errori in base alla strategia di ripetizione dei tentativi.

Filtraggio Amazon EventBridge Pipes

Con EventBridge Pipes, puoi filtrare gli eventi di una determinata fonte ed elaborarne solo un sottoinsieme. Questo filtraggio funziona allo stesso modo del filtraggio su un bus di EventBridge eventi o sulla mappatura della sorgente di eventi Lambda, utilizzando modelli di eventi. Per ulteriori informazioni sui modelli di eventi, consulta ???.

Un oggetto `FilterCriteria` criterio di filtro è una struttura costituita da un elenco di filtri (`Filters`). Ogni filtro è una struttura che definisce un modello di filtro (`Pattern`). Un `Pattern` è una rappresentazione di stringa di una regola di filtro JSON. L'aspetto di un oggetto `FilterCriteria` è simile a quanto illustrato nell'esempio seguente:

```
{
  "Filters": [
    {"Pattern": "{ \"Metadata1\": [ rule1 ], \"data\": { \"Data1\": [ rule2 ] }}"
    }
  ]
}
```

Per una maggiore chiarezza, ecco il valore del `Pattern` del filtro espanso in JSON semplice:

```
{
  "Metadata1": [ pattern1 ],
  "data": {"Data1": [ pattern2 ]}
}
```

Le parti principali di un oggetto `FilterCriteria` sono le proprietà di metadati e le proprietà di dati.

- Le proprietà di metadati sono i campi dell'oggetto evento. Nell'esempio, `FilterCriteria.Metadata1` si riferisce a una proprietà di metadati.
- Le proprietà di dati sono i campi dell'oggetto evento. Nell'esempio, `FilterCriteria.Data1` si riferisce a una proprietà di dati.

Ad esempio, supponiamo che il tuo flusso Kinesis contenga un evento come questo:

```
{
  "kinesisSchemaVersion": "1.0",
  "partitionKey": "1",
  "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
  "data": {"City": "Seattle",
    "State": "WA",
    "Temperature": "46",
    "Month": "December"
  },
  "approximateArrivalTimestamp": 1545084650.987
}
```

Quando l'evento attraversa la tua pipe, avrà il seguente aspetto con il campo `data` con codifica base64:

```
{
  "kinesisSchemaVersion": "1.0",
  "partitionKey": "1",
  "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
  "data": "SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
  "approximateArrivalTimestamp": 1545084650.987,
  "eventSource": "aws:kinesis",
  "eventVersion": "1.0",
  "eventID":
 "shardId-000000000006:49590338271490256608559692538361571095921575989136588898",
  "eventName": "aws:kinesis:record",
  "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
  "awsRegion": "us-east-2",
  "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
}
```

Le proprietà di metadati nell'evento Kinesis sono qualsiasi campo esterno all'oggetto`data`, ad esempio `partitionKey` o `sequenceNumber`.

Le proprietà di metadati dell'evento Kinesis sono i campi nell'oggetto `data`, ad esempio `City` o `Temperature`.

Quando applichi i filtri per trovare una corrispondenza con questo evento, puoi farlo sui campi decodificati. Ad esempio, per filtrare in base a `partitionKey` e `City` devi utilizzare il seguente filtro:

```
{
  "partitionKey": [
    "1"
  ],
  "data": {
    "City": [
      "Seattle"
    ]
  }
}
```

Quando crei filtri per eventi, EventBridge Pipes può accedere al contenuto degli eventi. Questo contenuto può avere caratteri di escape JSON, come il campo `body` di Amazon SQS, o la codifica base64, come il campo `data` di Kinesis. Se i dati sono JSON valido, i modelli di input o i percorsi JSON per i parametri di destinazione possono fare riferimento direttamente al contenuto. Ad esempio, se un'origine di evento Kinesis è JSON valido, puoi fare riferimento a una variabile utilizzando `<$.data.someKey>`.

Quando si creano modelli di eventi, è possibile filtrare in base ai campi inviati dall'API di origine e non ai campi aggiunti dall'operazione di polling. I seguenti campi non possono essere utilizzati nei modelli di eventi:

- `awsRegion`
- `eventSource`
- `eventSourceARN`
- `eventVersion`
- `eventID`
- `eventName`
- `invokeIdentityArn`
- `eventSourceKey`

Campi dati e messaggio

Ogni sorgente EventBridge Pipe contiene un campo che contiene il messaggio o i dati principali. Questi campi sono denominati campi messaggio o campi dati. Sono speciali perché possono avere caratteri di escape JSON o la codifica base64, ma quando sono JSON valido possono essere filtrati

con modelli JSON come se il corpo non avesse caratteri di escape. Il contenuto di questi campi può essere utilizzato senza problemi in trasformatori di input.

Filtro corretto dei messaggi Amazon SQS

Se un messaggio Amazon SQS non soddisfa i tuoi criteri di filtro, rimuove EventBridge automaticamente il messaggio dalla coda. Non è necessario eliminare manualmente questi messaggi in Amazon SQS.

Per Amazon SQS il messaggio `body` può essere qualsiasi stringa. Tuttavia, questo può essere problematico se il `FilterCriteria` si aspetta che `body` sia in un formato JSON valido. Anche lo scenario inverso è vero: se il messaggio in arrivo `body` è in un formato JSON valido ma i criteri di filtro si aspettano che `body` sia una stringa semplice, si ha un comportamento non previsto.

Per evitare questo problema, assicurati che il formato di `body` in `FilterCriteria` corrisponda al formato previsto di `body` nei messaggi ricevuti dalla coda. Prima di filtrare i messaggi, valuta EventBridge automaticamente il formato del messaggio in arrivo `body` e il modello di filtro per. `body` Se c'è una mancata corrispondenza, EventBridge elimina il messaggio. La tabella seguente riepiloga questa valutazione:

Formato **body** messaggio in arrivo	Formato **body** modello di filtro	Operazione risultante
Stringa normale	Stringa normale	EventBridge filtra in base ai tuoi criteri di filtro.
Stringa normale	Nessun modello di filtro per le proprietà dei dati	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
Stringa normale	JSON valido	EventBridge rilascia il messaggio.
JSON valido	Stringa normale	EventBridge rilascia il messaggio.
JSON valido	Nessun modello di filtro per le proprietà dei dati	EventBridge filtra (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.

Formato **body** messaggio in arrivo	Formato **body** modello di filtro	Operazione risultante
JSON valido	JSON valido	EventBridge filtri in base ai tuoi criteri di filtro.

Se non lo includi `body` come parte del tuo`FilterCriteria`, EventBridge salta questo controllo.

Filtraggio corretto dei messaggi Kinesis e Dynamo DB

Dopo che i criteri di filtro elaborano un record Kinesis o DynamoDB, l'iteratore di flussi ignora tale record. Se il registro non soddisfa i criteri di filtro, non è necessario eliminare manualmente il record dall'origine dell'evento. Dopo il periodo di conservazione, Kinesis e DynamoDB eliminano automaticamente questi vecchi record. Se vuoi che i record vengano eliminati prima, consulta Modifica del periodo di conservazione dei dati.

Per filtrare correttamente gli eventi dalle origini degli eventi di flusso, sia il campo dati che i criteri di filtro per il campo dati devono essere in formato JSON valido (per Kinesis, il campo dati è `data`, per Dynamo DB, il campo dati è `dynamodb`). Se uno dei due campi non è in un formato JSON valido, EventBridge elimina il messaggio o genera un'eccezione. La tabella seguente riepiloga il comportamento specifico:

Formato dei dati in entrata (**`data`** o **`dynamodb`**)	Formato del modello di filtro per le proprietà di dati	Operazione risultante
JSON valido	JSON valido	EventBridge filtra in base ai tuoi criteri di filtro.
JSON valido	Nessun modello di filtro per le proprietà dei dati	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
JSON valido	Non-JSON	EventBridge genera un'eccezione al momento della pipe o dell'aggiornamento. Il modello di filtro per le proprietà dei dati

Formato dei dati in entrata (**`data`** o **`dynamodb`**)	Formato del modello di filtro per le proprietà di dati	Operazione risultante
		deve essere in un formato JSON valido.
Non-JSON	JSON valido	EventBridge elimina il record.
Non-JSON	Nessun modello di filtro per le proprietà dei dati	EventBridge filtra (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
Non-JSON	Non-JSON	EventBridge genera un'eccezione al momento della creazione o dell'aggiornamento della pipe. Il modello di filtro per le proprietà dei dati deve essere in un formato JSON valido.

Filtro corretto dei messaggi di Streaming gestito da Amazon per Apache Kafka, Apache Kafka autogestito e Amazon MQ

Per le origini Amazon MQ, il campo del messaggio è `data`. Per le origini Apache Kafka (Amazon MSK e Apache Kafka autogestito), sono disponibili due campi di messaggio: `key` e `value`.

EventBridge elimina i messaggi che non corrispondono a tutti i campi inclusi nel filtro. Per Apache Kafka, esegue il EventBridge commit degli offset per i messaggi corrispondenti e non corrispondenti dopo aver richiamato correttamente la funzione. Per Amazon MQ, EventBridge riconosce i messaggi corrispondenti dopo aver richiamato correttamente la funzione e riconosce i messaggi non corrispondenti quando li filtra.

I messaggi Kafka e Amazon MQ devono essere stringhe codificate UTF-8, stringhe semplici o in formato JSON. Questo perché EventBridge decodifica gli array di byte Apache Kafka e Amazon MQ in UTF-8 prima di applicare i criteri di filtro. Se i tuoi messaggi utilizzano un'altra codifica, come UTF-16 o ASCII, o se il formato del messaggio non corrisponde al formato, elabora solo i filtri dei metadati. `FilterCriteria` EventBridge La tabella seguente riepiloga il comportamento specifico:

Formato del messaggio in arrivo (**`data`** o **`key`** e **`value`**)	Formato del modello di filtro per le proprietà di messaggi	Operazione risultante
Stringa normale	Stringa normale	EventBridge filtri in base ai tuoi criteri di filtro.
Stringa normale	Nessun modello di filtro per le proprietà dei dati	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
Stringa normale	JSON valido	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
JSON valido	Stringa normale	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
JSON valido	Nessun modello di filtro per le proprietà dei dati	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
JSON valido	JSON valido	EventBridge filtri in base ai tuoi criteri di filtro.
Stringa senza codifica UTF-8	JSON, stringa semplice o nessun modello	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.

Differenze tra Lambda ESM e Pipes EventBridge

Quando filtrano gli eventi, Lambda ESM EventBridge e Pipes funzionano generalmente allo stesso modo. La differenza principale è che il campo `eventSourceKey` non è presente nei payload ESM.

Arricchimento di eventi Amazon EventBridge Pipes

Con il passaggio relativo all'arricchimento di EventBridge Pipes, puoi migliorare i dati provenienti dall'origine prima di inviarli alla destinazione. Ad esempio, potresti ricevere eventi creati da ticket che

non includono i dati completi del ticket. Con l'arricchimento, è possibile chiamare l'API `get-ticket` mediante una funzione Lambda per ottenere dettagli completi sul ticket. Le pipe possono quindi inviare tali informazioni a una destinazione.

Puoi configurare i seguenti arricchimenti quando configuri una pipe in EventBridge:

- Destinazione API
- Gateway Amazon API
- Funzione Lambda
- Macchina a stati di Step Functions

> **Note**
>
> EventBridge Pipes supporta solo i flussi di lavoro Express come arricchimenti.

EventBridge richiama gli arricchimenti in modo sincrono perché deve attendere una risposta dall'arricchimento prima di richiamare la destinazione.

Le risposte di arricchimento sono limitate a una dimensione massima di 6 MB.

Puoi anche trasformare i dati ricevuti dall'origine prima di inviarli per migliorarli. Per ulteriori informazioni, consulta ???.

Filtrare eventi utilizzando l'arricchimento

EventBridge Pipes passa le risposte di arricchimento direttamente alla destinazione configurata. Ciò include le risposte degli array per le destinazioni che supportano i batch. Per ulteriori informazioni sul comportamento dei batch, consulta ???. È inoltre possibile utilizzare l'arricchimento come filtro e passare un numero di eventi inferiore a quello ricevuto dall'origine. Se non desideri richiamare la destinazione, restituisci una risposta vuota, ad esempio `""`, `{}` o `[]`.

> **Note**
>
> Se desideri richiamare la destinazione con un payload vuoto, restituisci un array con JSON vuoto `[{}]`.

Richiamo di arricchimenti

EventBridge richiama gli arricchimenti in modo sincrono (tipo di invocazione impostato su `REQUEST_RESPONSE`) perché deve attendere una risposta dall'arricchimento prima di richiamare la destinazione.

> **Note**
>
> Per le macchine a stati di Step Functions, EventBridge supporta solo i flussi di lavoro Express come arricchimenti, poiché possono essere richiamati in modo sincrono.

Obiettivi di Amazon EventBridge Pipes

Puoi inviare i dati presenti nella tua pipe a una destinazione specifica. Puoi configurare i seguenti obiettivi quando configuri una pipe in EventBridge:

- Destinazione API
- API Gateway
- Coda di processi batch
- CloudWatch gruppo di log
- Attività ECS
- Bus di eventi nello stesso account e nella stessa Regione
- Flussi di distribuzione Firehose
- Modello di valutazione di Inspector
- Flusso di Kinesis
- Funzione Lambda (SYNC o ASYNC)
- Query sull'API dati del cluster Redshift
- SageMaker Gasdotto
- Argomento Amazon SNS (argomenti FIFO SNS non supportati)
- Coda Amazon SQS
- Macchina a stati di Step Functions
  - Flussi di lavoro Express (SYNC o ASYNC)
  - Flussi di lavoro standard (ASYNC)

- Timestream per tavolo LiveAnalytics

Parametri di destinazione

Alcuni servizi di destinazione non inviano il payload dell'evento alla destinazione, ma trattano l'evento come un trigger per richiamare un'API specifica. EventBridge utilizza il `PipeTargetParameters`per specificare quali informazioni vengono inviate a quell'API. Questi sono i seguenti:

- Destinazioni API: i dati inviati a una destinazione API devono corrispondere alla struttura dell'API. È necessario utilizzare l'oggetto `InputTemplate` per assicurarsi che i dati siano strutturati correttamente. Se vuoi includere il payload dell'evento originale, fai riferimento a esso in `InputTemplate`.
- Gateway API: i dati inviati a Gateway API devono corrispondere alla struttura dell'API. È necessario utilizzare l'oggetto `InputTemplate` per assicurarsi che i dati siano strutturati correttamente. Se vuoi includere il payload dell'evento originale, fai riferimento a esso in `InputTemplate`.
- `PipeTargetRedshiftDataParameters` (cluster delle API dati di Amazon Redshift)
- `PipeTargetSageMakerPipelineParameters`(Pipeline di creazione SageMaker di modelli Amazon Runtime)
- `PipeTargetBatchJobParameters` (AWS Batch)

> **Note**
>
> EventBridge non supporta tutta la sintassi di JSON Path e la valuta in fase di esecuzione. La sintassi supportata include:
>
> - notazione a punti (ad esempio, `$.detail`)
> - trattini
> - caratteri di sottolineatura
> - caratteri alfanumerici
> - indici array
> - caratteri jolly (*)

Parametri di percorso dinamici

EventBridge I parametri di destinazione di Pipes supportano la sintassi del percorso JSON dinamico opzionale. È possibile utilizzare questa sintassi per specificare percorsi JSON anziché valori statici (ad esempio `$.detail.state`). L'intero valore deve essere un percorso JSON, non solo una parte di esso. Ad esempio, `RedshiftParameters.Sql` può essere `$.detail.state` ma non può essere `"SELECT * FROM $.detail.state"`. Questi percorsi vengono sostituiti dinamicamente al runtime con i dati del payload di eventi nel percorso specificato. I parametri di percorso dinamici non possono fare riferimento a valori nuovi o trasformati risultanti dalla trasformazione dell'input. La sintassi supportata per i percorsi JSON dei parametri dinamici è la stessa utilizzata per la trasformazione dell'input. Per ulteriori informazioni, consulta ???.

La sintassi dinamica può essere utilizzata su tutti i campi di tipo stringa non enum di tutti i parametri di arricchimento e di destinazione di EventBridge Pipes, ad eccezione di:

- `PipeTargetCloudWatchLogsParameters.LogStreamName`
- `PipeTargetEventBridgeEventBusParameters.EndpointId`
- `PipeEnrichmentHttpParameters.HeaderParameters`
- `PipeTargetHttpParameters.HeaderParameters`

Ad esempio, per impostare il target Kinesis `PartitionKey` di una pipe su una chiave personalizzata dal tuo evento di origine, imposta il. KinesisTargetParameter PartitionKeya:

- `"$.data.`*`someKey`*`"` per un'origine Kinesis
- `"$.body.`*`someKey`*`"` per un'origine Amazon SQS

Quindi, se il payload dell'evento è una stringa JSON valida, ad esempio`{"`*`someKey`*`":"`*`someValue`*`"}`, EventBridge estrae il valore dal percorso JSON e lo utilizza come parametro di destinazione. In questo esempio, EventBridge imposterebbe Kinesis su "`PartitionKey`*`someValue`*».

Autorizzazioni

Per effettuare chiamate API sulle risorse che possiedi, EventBridge Pipes necessita dell'autorizzazione appropriata. EventBridge PIPES utilizza il ruolo IAM specificato nella pipe per le chiamate di arricchimento e di destinazione utilizzando il principale `pipes.amazonaws.com` IAM.

Richiamo di destinazioni

EventBridge ha i seguenti modi per invocare un target:

- In modo sincrono (tipo di invocazione impostato su`REQUEST_RESPONSE`): EventBridge attende una risposta dal target prima di procedere.
- In modo asincrono (tipo di chiamata impostato su`FIRE_AND_FORGET`): non attende una risposta prima di procedere. EventBridge

Per impostazione predefinita, per le pipe con sorgenti ordinate, EventBridge richiama le destinazioni in modo sincrono perché è necessaria una risposta dalla destinazione prima di passare all'evento successivo.

Se un'origine non impone l'ordine, ad esempio una coda Amazon SQS standard, può richiamare una destinazione supportata in modo EventBridge sincrono o asincrono.

Con le funzioni Lambda e le macchine a stati Step Functions, puoi configurare il tipo di invocazione.

> **Note**
>
> Per le macchine a stati Step Functions, i Flussi di lavoro standard devono essere richiamati in modo asincrono.

EventBridge Specifiche del target Pipes

AWS Batch code di lavoro

Tutti i AWS Batch `submitJob` parametri sono configurati in modo esplicito con e`BatchParameters`, come tutti i parametri Pipe, possono essere dinamici utilizzando un percorso JSON verso il payload dell'evento in entrata.

CloudWatch Gruppo di log

Indipendentemente che si utilizzi o meno un trasformatore di input, il payload di eventi viene utilizzato come messaggio di log. Puoi impostare `Timestamp` (o `LogStreamName` esplicito della tua destinazione) tramite `CloudWatchLogsParameters` in `PipeTarget`. Come tutti i parametri pipe, questi parametri possono essere dinamici quando si utilizza un percorso JSON al payload di eventi in entrata.

Processo di Amazon ECS

Tutti i parametri `runTask` di Amazon ECS sono configurati esplicitamente tramite `EcsParameters`. Come tutti i parametri pipe, questi parametri possono essere dinamici quando si utilizza un percorso JSON al payload di eventi in entrata.

Funzioni Lambda e flussi di lavoro Step Functions

Lambda e Step Functions non dispongono di un'API batch. Per elaborare batch di eventi da un'origine pipe, il batch viene convertito in un array JSON e passato come input alla destinazione Lambda o Step Functions. Per ulteriori informazioni, consulta ???.

Timestream per tavolo LiveAnalytics

Le considerazioni da prendere in considerazione quando si specifica una LiveAnalytics tabella Timestream for come destinazione del tubo includono:

- Gli stream Apache Kafka (inclusi quelli provenienti da Amazon MSK fornitori terzi) non sono attualmente supportati come sorgenti pipe.
- Se hai specificato uno DynamoDB stream Kinesis or come sorgente pipe, devi specificare il numero di tentativi di nuovo tentativo.

  Per ulteriori informazioni, consulta ???.

Dosaggio e concorrenza di Amazon EventBridge Pipes

Comportamento di batching

EventBridge Pipes supporta il batching dall'origine e verso destinazioni che lo supportano. Inoltre, il batching per l'arricchimento è supportato per AWS Lambda e AWS Step Functions. Poiché servizi diversi supportano diversi livelli di batching, non è possibile configurare una pipe con di dimensioni di batch superiori a quelle supportate dalla destinazione. Ad esempio, le origini di flussi Amazon Kinesis supportano una dimensione di batch massima di 10.000 record, mentre Amazon Simple Queue Service supporta un massimo di 10 messaggi per batch come destinazione. Pertanto, la dimensione di batch massima configurata per l'origine di una pipe da un flusso Kinesis a una coda Amazon SQS è 10.

Se configuri una pipe con un arricchimento o una destinazione che non supporta il batching, non sarai in grado di attivare il batching per l'origine.

Quando il batching è attivato per l'origine, gli array di record JSON vengono passati tramite la pipe e quindi mappati all'API batch di un arricchimento o di una destinazione supportato. I trasformatori di input vengono applicati separatamente su ogni singolo record JSON nell'array, non sull'intero array. Per esempi di questi array, consulta ??? e seleziona un'origine specifica. Le pipe utilizzeranno l'API batch per l'arricchimento o la destinazione supportato anche se la dimensione del batch è 1. Se l'arricchimento o la destinazione non dispone di un'API batch ma riceve payload JSON completi, come Lambda e Step Functions, l'intero array JSON viene inviato in un'unica richiesta. La richiesta verrà inviata come array JSON anche se la dimensione del batch è 1.

Se una pipe è configurata per il batching all'origine e la destinazione supporta il batching, puoi restituire un array di elementi JSON dal tuo arricchimento. Questo array può includere un array più o meno lungo dell'origine originale. Tuttavia, se l'array è più grande della dimensione di batch supportata dalla destinazione, la pipe non richiamerà la destinazione.

Destinazioni batch supportate

Target	Dimensione massima batch
CloudWatch Registri	10.000
EventBridge bus per eventi	10
Flusso Firehose	500
Flusso di Kinesis	500
Funzione Lambda	definita dal cliente
Macchina a stati di Step Functions	definita dal cliente
Argomento Amazon SNS	10
Coda Amazon SQS	10

I seguenti arricchimenti e destinazioni ricevono il payload completo dell'evento batch per l'elaborazione e sono vincolati dalla dimensione totale del payload dell'evento, anziché dalla dimensione del batch:

- Macchina a stati di Step Functions (262.144 caratteri)

- Funzione Lambda (6 MB)

Errori batch parziali

Per Amazon SQS e sorgenti di flusso, come Kinesis e DynamoDB, Pipes supporta la gestione parziale degli errori in batch degli EventBridge errori di destinazione. Se la destinazione supporta il batching e solo una parte del batch riesce, riprova EventBridge automaticamente a eseguire il batch per il resto del payload. Per i contenuti più up-to-date arricchiti, questo nuovo tentativo avviene attraverso l'intera pipeline, inclusa la reinvocazione di qualsiasi arricchimento configurato.

La gestione degli errori di batch parziali dell'arricchimento non è supportata.

Per le destinazioni Lambda e Step Functions, puoi anche specificare un errore parziale restituendo un payload con struttura definita dalla destinazione. Ciò indica gli eventi per i quali deve essere effettuato un nuovo tentativo.

Esempio di struttura di payload con errore parziale

```
{
  "batchItemFailures": [
    {
      "itemIdentifier": "id2"
    },
    {
      "itemIdentifier": "id4"
    }
]
```

Nell'esempio, le due occorrenze di `itemIdentifier` corrispondono all'ID degli eventi gestiti dalla destinazione e provenienti dalla relativa origine originale. Per Amazon SQS, è `messageId`. Per Kinesis e DynamoDB, è `eventID`. EventBridge Affinché Pipes possa gestire in modo adeguato gli errori parziali dei batch provenienti dalle destinazioni, questi campi devono essere inclusi in qualsiasi payload dell'array restituito dall'arricchimento.

Capacità e comportamento di simultaneità

Ogni evento o batch di eventi ricevuto da una pipe e inviato a un arricchimento o a una destinazione viene considerato come un'esecuzione di pipe. Una pipe il cui stato è `STARTED` esegue continuamente il polling degli eventi dall'origine, aumentando o diminuendo in base al backlog disponibile e alle impostazioni di batching configurate.

Per informazioni sulle quote relative a esecuzioni simultanee di pipe e sul numero di pipe per account e Regione, consulta ???.

Per impostazione predefinita, una singola pipe verrà dimensionata al numero massimo di esecuzioni simultanee seguenti, a seconda dell'origine:

- DynamoDB: il numero massimo di esecuzioni simultanee è pari al valore di `ParallelizationFactor` configurato per la pipe moltiplicato per il numero di partizioni nel flusso.
- Apache Kafka: il numero massimo di esecuzioni simultanee è pari al numero di partizioni nell'argomento, ovvero 1000.
- Kinesis: il numero massimo di esecuzioni simultanee è pari al valore di `ParallelizationFactor` configurato per la pipe moltiplicato per il numero di partizioni nel flusso.
- Amazon MQ: 5
- Amazon SQS: 1.250

Se hai bisogno di limiti di polling massimo o di simultaneità più alti, contatta l'assistenza.

> **Note**
>
> I limiti di esecuzione sono considerati come limiti di sicurezza ottimali. Sebbene il polling possa scendere al di sotto di questi valori, una pipe o un account potrebbero superare questi valori consigliati.

Le esecuzioni delle pipe sono limitate a un massimo di 5 minuti, inclusa l'elaborazione di arricchimento e destinazione. Attualmente questo limite non può essere aumentato.

Le pipe con origini rigorosamente ordinate, come le code FIFO di Amazon SQS, Kinesis e DynamoDB Streams o gli argomenti Apache Kafka, sono ulteriormente limitate quanto a simultaneità dalla configurazione dell'origine, come il numero di ID gruppo di messaggi per le code FIFO o il numero di partizioni per le code Kinesis. Poiché l'ordinamento è strettamente garantito entro questi vincoli, una pipe con un'origine ordinata non può superare tali limiti di simultaneità.

Trasformazione di input di Amazon EventBridge Pipes

Amazon EventBridge Pipes supporta trasformatori di input facoltativi per il trasferimento dei dati all'arricchimento e alla destinazione. Puoi utilizzare i trasformatori di input per modificare il payload di input degli eventi JSON e soddisfare le esigenze del servizio di arricchimento o di destinazione. Di seguito è descritto come modificare l'evento di input in base al modello RESTful della tua API per Gateway Amazon API e le destinazioni API. I trasformatori di input sono modellati come parametro `InputTemplate`. Possono essere testo libero, un percorso JSON al payload di eventi o un oggetto JSON che include percorsi JSON in linea al payload di eventi. Per l'arricchimento, il payload di eventi proviene dall'origine. Per le destinazioni, il payload di eventi è ciò che viene restituito dall'arricchimento, se configurato nella pipe. Oltre ai dati specifici del servizio presenti nel payload di eventi, è possibile utilizzare variabili riservate in `InputTemplate` per fare riferimento ai dati per la pipe.

Per accedere agli elementi in un array, utilizza la notazione con parentesi quadre.

> **Note**
>
> EventBridge non supporta tutta la sintassi di percorso JSON e la valuta al runtime. La sintassi supportata include:
>
> - notazione a punti (ad esempio, `$.detail`)
> - trattini
> - caratteri di sottolineatura
> - caratteri alfanumerici
> - indici array
> - caratteri jolly (*)

Di seguito sono riportati alcuni parametri `InputTemplate` di esempio che fanno riferimento a un payload di eventi Amazon SQS:

Stringa statica

```
InputTemplate: "Hello, sender"
```

Percorso JSON

```
InputTemplate: <$.attributes.SenderId>
```

Stringa dinamica

```
InputTemplate: "Hello, <$.attributes.SenderId>"
```

JSON statico

```
InputTemplate: >
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3",
}
```

JSON dinamico

```
InputTemplate: >
{
  "key1": "value1"
  "key2": <$.body.key>,
  "d": <aws.pipes.event.ingestion-time>
}
```

Per accedere agli elementi in un array con la notazione tra parentesi quadre:

```
InputTemplate: >
{
  "key1": "value1"
  "key2": <$.body.Records[3]>,
  "d": <aws.pipes.event.ingestion-time>
}
```

> **Note**
>
> EventBridge sostituisce i trasformatori di input al runtime per garantire un output JSON valido. Per questo motivo, inserisci tra virgolette le variabili che fanno riferimento ai parametri di percorso JSON, ma non le variabili che si riferiscono a oggetti o array JSON.

Variabili riservate

I modelli di input possono utilizzare le seguenti variabili riservate:

- `<aws.pipes.pipe-arn>`: il nome della risorsa Amazon (ARN) della pipe.
- `<aws.pipes.pipe-name>`: il nome della pipe.
- `<aws.pipes.source-arn>`: l'ARN dell'origine dell'evento della pipe.
- `<aws.pipes.enrichment-arn>`: l'ARN di arricchimento della pipe.
- `<aws.pipes.target-arn>`: l'ARN della destinazione della pipe.
- `<aws.pipes.event.ingestion-time>`: l'ora alla quale il trasformatore di input ha ricevuto l'evento. Si tratta di un timestamp ISO 8601. Questa ora è diversa per il trasformatore di input di arricchimento e il trasformatore di input di destinazione, a seconda di quando l'arricchimento ha completato l'elaborazione dell'evento.
- `<aws.pipes.event>`: l'evento come ricevuto dal trasformatore di input.

  Per un trasformatore di input di arricchimento, si tratta dell'evento proveniente dall'origine. Contiene il payload originale dall'origine, nonché altri metadati specifici del servizio. Per esempi specifici del servizio, vedi gli argomenti in ???.

  Per un trasformatore di input di destinazione, si tratta dell'evento restituito dall'arricchimento, se configurato, senza metadati aggiuntivi. Pertanto, un payload restituito dall'arricchimento potrebbe non essere JSON. Se nessun arricchimento è configurato sulla pipe, questo è l'evento proveniente dall'origine con metadati.
- `<aws.pipes.event.json>`: uguale a `aws.pipes.event`, ma la variabile ha un valore solo se il payload originale, proveniente dall'origine o restituito dall'arricchimento, è JSON. Se la pipe ha un campo codificato, come il campo `body` di Amazon SQS o `data` di Kinesis, tali campi vengono decodificati e trasformati in JSON valido. Poiché non ha caratteri di escape, la variabile può essere utilizzata solo come valore per un campo JSON. Per ulteriori informazioni, consulta ???.

Esempi di trasformazione di input

Di seguito è riportato un esempio di evento Amazon EC2 che possiamo utilizzare come evento di esempio.

```
{
  "version": "0",
```

```
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2015-11-11T21:29:54Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
  ],
  "detail": {
    "instance-id": "i-0123456789",
    "state": "RUNNING"
  }
}
```

Utilizziamo il JSON seguente come trasformatore.

```
{
  "instance" : <$.detail.instance-id>,
  "state": <$.detail.state>,
  "pipeArn" : <aws.pipes.pipe-arn>,
  "pipeName" : <aws.pipes.pipe-name>,
  "originalEvent" : <aws.pipes.event.json>
}
```

Di seguito è riportato l'output risultante:

```
{
  "instance" : "i-0123456789",
  "state": "RUNNING",
  "pipeArn" : "arn:aws:pipe:us-east-1:123456789012:pipe/example",
  "pipeName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

Analisi implicita dei dati del corpo

I seguenti campi nel payload in entrata possono avere caratteri di escape JSON, come l'oggetto body Amazon SQS, o la codifica base64, come l'oggetto `data` Kinesis. Sia per i filtri che per la

trasformazione dell'input, EventBridge trasforma questi campi in JSON valido in modo da poter fare riferimento direttamente a valori secondari. Ad esempio, `<$.data.someKey>` per Kinesis.

Per fare in modo che la destinazione riceva il payload originale senza metadati aggiuntivi, utilizza un trasformatore di input con questi dati del corpo, specifici dell'origine. Ad esempio, `<$.body>` per Amazon SQS o `<$.data>` per Kinesis. Se il payload originale è una stringa JSON valida (ad esempio `{"key": "value"}`), l'uso del trasformatore di input con dati del corpo specifici dell'origine comporterà la rimozione delle virgolette nel payload di origine originale. Ad esempio, `{"key": "value"}` diventerà `"{key: value}"` quando distribuito alla destinazione. Se la destinazione richiede payload JSON validi (ad esempio, EventBridge Lambda o Step Functions), ciò causerà un errore di distribuzione. Per fare in modo che la destinazione riceva i dati dell'origine originale senza generare JSON non valido, racchiudi il trasformatore di input dei dati del corpo dell'origine in JSON. Ad esempio, `{"data": <$.data>}`.

L'analisi implicita del corpo può essere utilizzata anche per immettere dinamicamente i valori della maggior parte dei parametri di destinazione o di arricchimento delle pipe. Per ulteriori informazioni, consulta ???

> **Note**
>
> Se il payload originale è JSON valido, questo campo conterrà JSON senza caratteri di escape e codifica base64. Tuttavia, se il payload non è JSON valido, EventBridge utilizza la codifica base64 per i campi elencati di seguito, a eccezione di Amazon SQS.

- MQ attivo: `data`
- Kinesis: `data`
- Amazon MSK: `key` e `value`
- Rabbit MQ: `data`
- Apache Kafka autogestito: `key` e `value`
- Amazon SQS: `body`

Problemi comuni con la trasformazione di input

Di seguito sono riportati alcuni problemi comuni che si verificano durante la trasformazione di input in pipe EventBridge:

- Per le stringhe, le virgolette sono obbligatorie.
- Non vi è alcuna convalida durante la creazione del percorso JSON per il modello.
- Se specifichi una variabile per la corrispondenza con un percorso JSON che non esiste nell'evento, quella variabile non viene creata e non appare nell'output.
- Le proprietà JSON come `aws.pipes.event.json` possono essere utilizzate solo come valore di un campo JSON, non in linea in altre stringhe.
- EventBridge non utilizza caratteri di escape con i valori estratti da Percorso di input quando viene popolato il Modello di input per una destinazione.
- Se un percorso JSON fa riferimento a un oggetto o a un array JSON, ma alla variabile si fa riferimento in una stringa, EventBridge rimuove tutte le virgolette interne per garantire una stringa valida. Ad esempio, con "Body is <$.body>" EventBridge rimuove le virgolette dall'oggetto.

  Pertanto, se come output vuoi un oggetto JSON basato su una singola variabile di percorso JSON, devi posizionarlo come chiave. In questo esempio, `{"body": <$.body>}`.
- Le virgolette non sono necessarie per le variabili che rappresentano stringhe. Sono consentite, ma EventBridge Pipes aggiunge automaticamente le virgolette ai valori delle variabili di stringa durante la trasformazione, per garantire che l'output della trasformazione sia JSON valido. EventBridge Pipes non aggiunge virgolette alle variabili che rappresentano oggetti o array JSON. Non aggiungere virgolette alle variabili che rappresentano oggetti o array JSON.

  Ad esempio, il seguente modello di input include variabili che rappresentano sia stringhe che oggetti JSON:

```
{
  "pipeArn" : <aws.pipes.pipe-arn>,
  "pipeName" : <aws.pipes.pipe-name>,
  "originalEvent" : <aws.pipes.event.json>
}
```

  Il risultato è JSON valido con un uso corretto delle virgolette:

```
{
  "pipeArn" : "arn:aws:events:us-east-2:123456789012:pipe/example",
  "pipeName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
```

```
}
```

- Per gli arricchimenti o le destinazioni Lambda o Step Functions, i batch vengono distribuiti alla destinazione come array JSON, anche se la dimensione del batch è 1. Tuttavia, i trasformatori di input verranno comunque applicati a singoli record nell'array JSON e non all'intero array. Per ulteriori informazioni, consulta ???.

Registra Amazon EventBridge Pipes

EventBridge La registrazione di Pipes consente a EventBridge Pipes di inviare record che descrivono in dettaglio le prestazioni delle pipe ai servizi supportati. AWS Utilizza i log per ottenere informazioni dettagliate sulle prestazioni di esecuzione della tua pipe e per facilitare la risoluzione dei problemi e il debug.

È possibile selezionare i seguenti AWS servizi come destinazioni di registro a cui EventBridge Pipes invia i record:

- CloudWatch Registri

  EventBridge fornisce i record di registro al gruppo di CloudWatch log Logs specificato.

  Utilizza CloudWatch Logs per centralizzare i log di tutti i sistemi, le applicazioni e i AWS servizi che utilizzi, in un unico servizio altamente scalabile. Per ulteriori informazioni, consulta Working with log groups and log stream nella Amazon CloudWatch Logs User Guide.
- Stream log Firehose

  EventBridge invia i record di log a un flusso di distribuzione Firehose.

  Amazon Data Firehose è un servizio completamente gestito per la distribuzione di dati di streaming in tempo reale a destinazioni come determinati AWS servizi, nonché a qualsiasi endpoint HTTP personalizzato o endpoint HTTP di proprietà di provider di servizi terzi supportati. Per ulteriori informazioni, consulta Creazione di un flusso di distribuzione di Amazon Data Firehose nella Amazon Data Firehose User Guide.
- Log Amazon S3

  EventBridge fornisce i record di log come oggetti Amazon S3 al bucket specificato.

  Amazon S3 è un servizio di archiviazione di oggetti che offre scalabilità, disponibilità dei dati, sicurezza e prestazioni tra le migliori del settore. Per ulteriori informazioni, consulta Caricamento,

download e utilizzo di oggetti in Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Come funziona la registrazione di Amazon EventBridge Pipes

Un'esecuzione è un evento o batch di eventi ricevuto da una pipe verso un arricchimento e/o una destinazione. Se abilitato, EventBridge genera un record di registro per ogni fase di esecuzione eseguita durante l'elaborazione del batch di eventi. Le informazioni contenute nel record si applicano al batch di eventi, che si tratti di un singolo evento o di un massimo di 10.000 eventi.

È possibile configurare la dimensione del batch di eventi nell'origine e nella destinazione della pipe. Per ulteriori informazioni, consulta ???.

I dati dei record inviati a ciascuna destinazione di log sono gli stessi.

Se è configurata una destinazione Amazon CloudWatch Logs, i record di log consegnati a tutte le destinazioni hanno un limite di 256 kb. I campi verranno troncati come necessario.

Puoi personalizzare i record EventBridge inviati alle destinazioni di log selezionate nel modo seguente:

- È possibile specificare il livello di registro, che determina i passaggi di esecuzione per i quali EventBridge invia i record alle destinazioni di registro selezionate. Per ulteriori informazioni, consulta ???.
- È possibile specificare se EventBridge Pipes include i dati di esecuzione nei record per le fasi di esecuzione, laddove pertinenti. Questi dati includono:
  - Il payload del batch di eventi
  - La richiesta inviata al servizio di AWS arricchimento o di destinazione
  - La risposta restituita dal servizio di AWS arricchimento o di destinazione

  Per ulteriori informazioni, consulta ???.

Specificazione del livello di log di EventBridge Pipes

È possibile specificare i tipi di passaggi di esecuzione per i quali EventBridge invia i record alle destinazioni di registro selezionate.

Scegli tra i seguenti livelli di dettaglio da includere nei record di log. Il livello di log si applica a tutte le destinazioni di log specificate per la pipe. Ogni livello di log include le fasi di esecuzione dei livelli di log precedenti.

- OFF: EventBridge non invia alcun record a nessuna destinazione di registro specificata. Si tratta dell'impostazione di default.
- ERRORE — EventBridge invia tutti i record relativi agli errori generati durante l'esecuzione della pipe alle destinazioni di log specificate.
- INFO: EventBridge invia tutti i record relativi agli errori, oltre a selezionare altri passaggi eseguiti durante l'esecuzione della pipe alle destinazioni di log specificate.
- TRACE: EventBridge invia tutti i record generati durante qualsiasi fase dell'esecuzione della pipe alle destinazioni di log specificate.

Nella EventBridge console, CloudWatch i log sono selezionati come destinazione di log per impostazione predefinita, così come il livello di `ERROR` registro. Quindi, per impostazione predefinita, EventBridge Pipes crea un nuovo gruppo di CloudWatch log a cui invia i record di registro contenenti il `ERROR` livello di dettaglio. Non viene selezionato alcun valore predefinito quando si configurano i log a livello di codice.

La tabella seguente elenca le fasi di esecuzione incluse in ogni livello di log.

Fase	TRACE	INFO	ERRORE	OFF
Esecuzione non riuscita	x	x	x	
Esecuzione parzialmente non riuscita	x	x	x	
Esecuzione avviata	x	x		
Esecuzione riuscita	x	x		
Esecuzione limitata	x	x	x	
Timeout di esecuzione	x	x	x	
Invocazione arricchimento non riuscita	x	x	x	

Fase	TRACE	INFO	ERRORE	OFF
Invocazione arricchimento ignorata	x	x		
Invocazione arricchimento avviata	x			
Invocazione arricchimento riuscita	x			
Fase di arricchimento immessa	x	x		
Fase di arricchimento non riuscita	x	x	x	
Fase di arricchimento riuscita	x	x		
Trasformazione arricchimento non riuscita	x	x	x	
Trasformazione arricchimento avviata	x			
Trasformazione arricchimento riuscita	x			
Invocazione destinazione non riuscita	x	x	x	
Invocazione destinazione parzialmente non riuscita	x	x	x	
Invocazione destinazione ignorata	x			
Invocazione destinazione avviata	x			

Fase	TRACE	INFO	ERRORE	OFF
Invocazione destinazione riuscita	x			
Fase di destinazione immessa	x	x		
Fase di destinazione non riuscita	x	x	x	
Fase di destinazione parzialmente non riuscita	x	x	x	
Fase di destinazione ignorata	x			
Fase di destinazione riuscita	x	x		
Trasformazione destinazione non riuscita	x	x	x	
Trasformazione destinazione avviata	x			
Trasformazione destinazione riuscita	x			

Inclusione dei dati di esecuzione nei log di EventBridge Pipes

È possibile specificare EventBridge di includere i dati di esecuzione nei record generati. I dati di esecuzione includono i campi che rappresentano il payload dei batch di eventi, nonché la richiesta inviata e la risposta dell'arricchimento e della destinazione.

I dati di esecuzione sono utili per la risoluzione dei problemi e il debug. Il campo `payload` contiene il contenuto effettivo di ogni evento incluso nel batch e consente di correlare singoli eventi a un'esecuzione di pipe specifica.

Se scegli di includere i dati di esecuzione, questi vengono inclusi per tutte le destinazioni di log specificate per la pipe.

> **Important**
>
> Questi campi possono contenere informazioni riservate. EventBridge non tenta di oscurare il contenuto di questi campi durante la registrazione.

Quando include i dati di esecuzione, EventBridge aggiunge i seguenti campi ai record pertinenti:

- **`payload`**

  Rappresenta il contenuto del batch di eventi elaborato dalla pipe.

  EventBridge include il `payload` campo nei record generati nelle fasi in cui il contenuto del batch di eventi potrebbe essere stato aggiornato. Ciò include le seguenti fasi:

  - `EXECUTION_STARTED`
  - `ENRICHMENT_TRANSFORMATION_SUCCEEDED`
  - `ENRICHMENT_STAGE_SUCCEEDED`
  - `TARGET_TRANSFORMATION_SUCCEEDED`
  - `TARGET_STAGE_SUCCEEDED`
- **`awsRequest`**

  Rappresenta la richiesta inviata all'arricchimento o alla destinazione come stringa JSON. Per le richieste inviate a una destinazione API, rappresenta la richiesta HTTP inviata a quell'endpoint.

  EventBridge include il `awsRequest` campo nei record generati nelle fasi finali di arricchimento e targeting, ovvero dopo aver EventBridge eseguito o tentato di eseguire la richiesta relativa al servizio di arricchimento o di destinazione specificato. Ciò include le seguenti fasi:

  - `ENRICHMENT_INVOCATION_FAILED`
  - `ENRICHMENT_INVOCATION_SUCCEEDED`
  - `TARGET_INVOCATION_FAILED`
  - `TARGET_INVOCATION_PARTIALLY_FAILED`
  - `TARGET_INVOCATION_SUCCEEDED`
- **`awsResponse`**

  Rappresenta la risposta restituita dall'arricchimento o dalla destinazione, in formato JSON. Per le richieste inviate a una destinazione API, rappresenta la risposta HTTP restituita da quell'endpoint.

Ad esempioawsRequest, EventBridge include il awsResponse campo nei record generati nelle fasi finali di arricchimento e targeting, ovvero dopo aver EventBridge eseguito o tentato di eseguire una richiesta relativa al servizio di arricchimento o di destinazione specificato e aver ricevuto una risposta. Ciò include le seguenti fasi:

- ENRICHMENT_INVOCATION_FAILED
- ENRICHMENT_INVOCATION_SUCCEEDED
- TARGET_INVOCATION_FAILED
- TARGET_INVOCATION_PARTIALLY_FAILED
- TARGET_INVOCATION_SUCCEEDED

Per una descrizione delle fasi di esecuzione delle pipe, consulta ???.

Troncare i dati di esecuzione nei record di log di Pipes EventBridge

Se si sceglie di EventBridge includere i dati di esecuzione nei record di registro di una pipe, esiste la possibilità che un record superi il limite di dimensione di 256 KB. Per evitare ciò, tronca EventBridge automaticamente i campi dei dati di esecuzione, nell'ordine seguente. EventBridge tronca completamente ogni campo prima di passare al tronco del campo successivo. EventBridge tronca i dati del campo semplicemente rimuovendo i caratteri dalla fine della stringa di dati; non viene effettuato alcun tentativo di troncare in base all'importanza dei dati e il troncamento invaliderà la formattazione JSON.

- payload
- awsRequest
- awsResponse

Se tronca i campi nell'evento, il campo EventBridge include un elenco dei campi dati troncati. truncatedFields

Segnalazione degli errori nei registri di Pipes EventBridge

EventBridge include anche i dati di errore, ove disponibili, nelle fasi di esecuzione delle pipe che rappresentano gli stati di errore. Queste fasi includono:

- ExecutionThrottled

- `ExecutionTimeout`
- `ExecutionFailed`
- `ExecutionPartiallyFailed`
- `EnrichmentTransformationFailed`
- `EnrichmentInvocationFailed`
- `EnrichmentStageFailed`
- `TargetTransformationFailed`
- `TargetInvocationFailed`
- `TargetInvocationPartiallyFailed`
- `TargetStageFailed`
- `TargetStagePartiallyFailed`

EventBridge Fasi di esecuzione delle pipe

Comprendere il flusso delle fasi di esecuzione delle pipe può aiutarti nella risoluzione dei problemi o nel debug delle prestazioni della pipe utilizzando log.

Un'esecuzione di pipe è un evento o batch di eventi ricevuto da una pipe verso un arricchimento o una destinazione. Se abilitata, EventBridge genera un record di registro per ogni fase di esecuzione eseguita durante l'elaborazione del batch di eventi.

L'esecuzione contiene due fasi o un insieme di passaggi: arricchimento e destinazione. Ognuna di queste fasi comporta passaggi di trasformazione e invocazione.

I passaggi principali di una corretta esecuzione di una pipe seguono questo flusso:

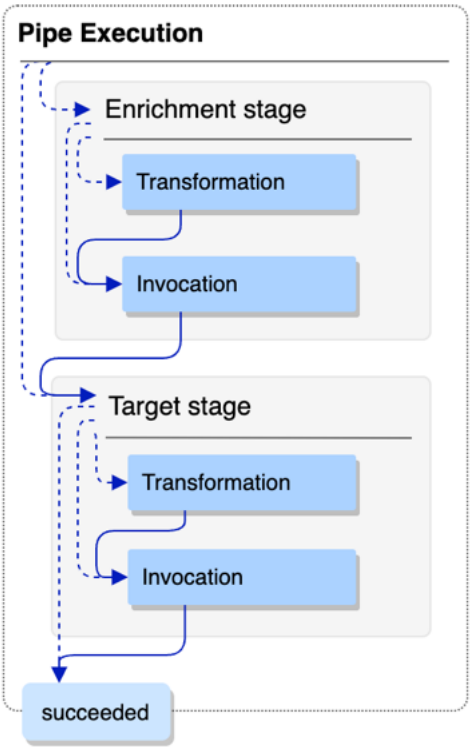
- Viene avviata l'esecuzione della pipe.
- L'esecuzione entra nella fase di arricchimento se è stato specificato un arricchimento per gli eventi. Se non hai specificato un arricchimento, l'esecuzione passa alla fase di destinazione.

  Nella fase di arricchimento, la pipe esegue qualsiasi trasformazione specificata, quindi richiama l'arricchimento.
- Nella fase di destinazione, la pipe esegue qualsiasi trasformazione specificata, quindi richiama la destinazione.

Se non hai specificato la trasformazione o la destinazione, l'esecuzione salta la fase di destinazione.

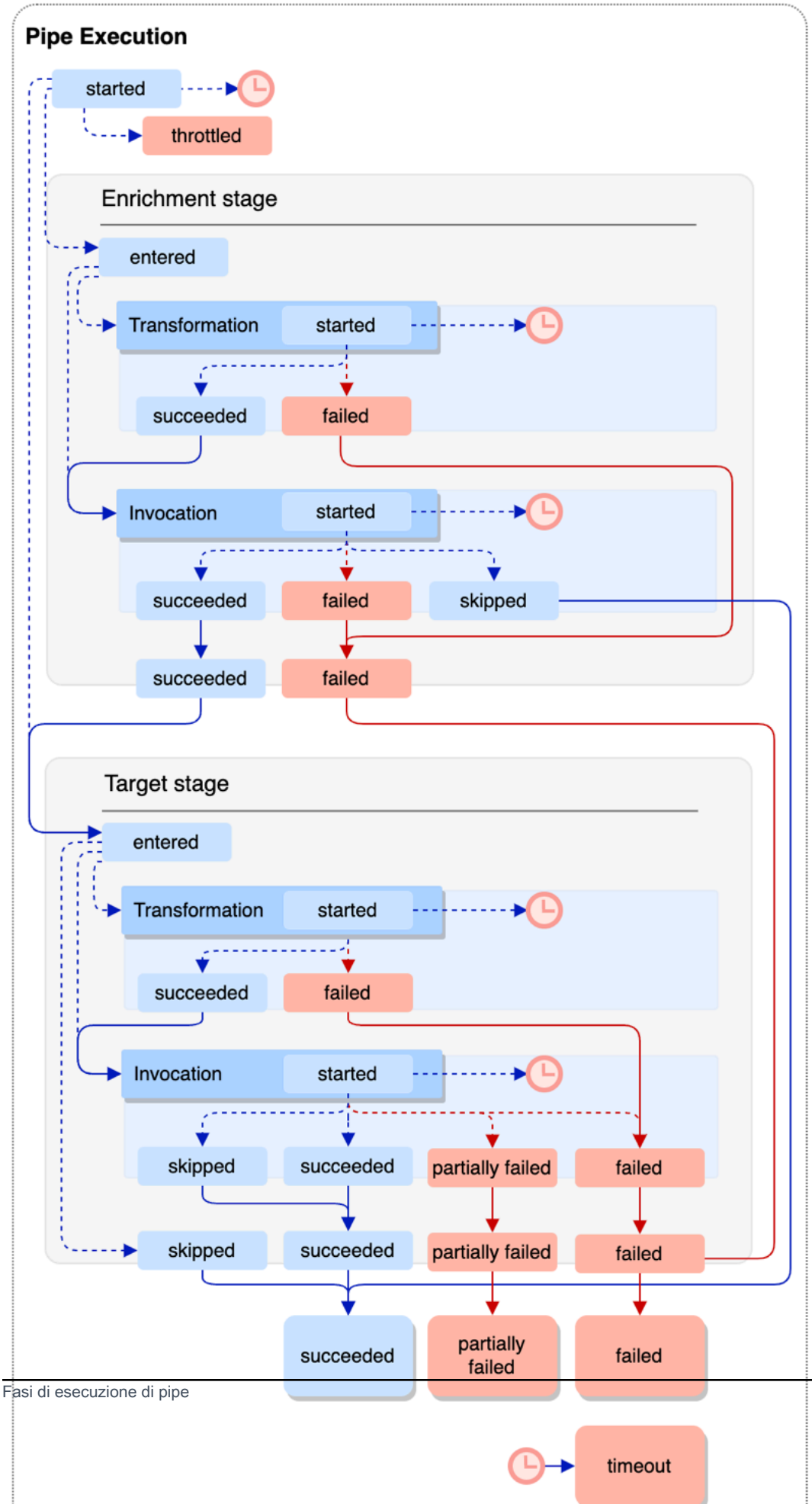
- L'esecuzione della pipe viene completata correttamente.

Il diagramma seguente illustra questo flusso. I percorsi divergenti sono formattati come linee tratteggiate.



Il diagramma seguente presenta una visualizzazione dettagliata del flusso di esecuzione delle pipe, con tutti i possibili passaggi di esecuzione rappresentati. Anche qui, i percorsi divergenti sono formattati come linee tratteggiate.

Per l'elenco completo dei passaggi di esecuzione delle pipe, consulta ???.


Pipe Execution
started
throttled
Enrichment stage
entered
Transformation
started
succeeded
failed
Invocation
started
succeeded
failed
skipped
succeeded
failed
Target stage
entered
Transformation
started
succeeded
failed
Invocation
started
skipped
succeeded
partially failed
failed
skipped
succeeded
partially failed
failed
succeeded
partially failed
failed
timeout

Nota che l'invocazione della destinazione può causare un errore parziale del batch. Per ulteriori informazioni, consulta ???.

EventBridge Riferimento allo schema del registro dei tubi

Il riferimento seguente descrive in dettaglio lo schema per i record di log di EventBridge Pipes.

Ogni record di log rappresenta un passaggio di esecuzione della pipe e può contenere fino a 10.000 eventi se l'origine e la destinazione della pipe sono state configurate per il batch.

Per ulteriori informazioni, consulta ???.

```
{
    "executionId": "guid",
    "timestamp": "date_time",
    "messageType": "execution_step",
    "resourceArn": "arn:aws:pipes:region:account:pipe/pipe-name",
    "logLevel": "TRACE | INFO | ERROR",
    "payload": "{}",
    "awsRequest": "{}"
    "awsResponse":"{}"
    "truncatedFields": ["awsRequest","awsResponse","payload"],
    "error": {
        "httpStatusCode": code,
        "message": "error_message",
        "details": "",
        "awsService": "service_name",
        "requestId": "service_request_id"
    }
}
```

executionId

L'ID dell'esecuzione della pipe.

Un'esecuzione di pipe è un evento o batch di eventi ricevuto da una pipe verso un arricchimento o una destinazione. Per ulteriori informazioni, consulta ???.

timestamp

La data e l'ora in cui il log eventi è stato emesso.

Unità: millisecondi

messageType

Il passaggio di esecuzione della pipe per la quale è stato generato il record.

Per ulteriori informazioni sui passaggi di esecuzione delle pipe, consulta ???.

resourceArn

Il nome della risorsa Amazon (ARN) per la pipe.

logLevel

Il livello di dettaglio specificato per il log della pipe.

Valori validi: `ERROR` | `INFO` | `TRACE`

Per ulteriori informazioni, consulta ???.

payload

Il contenuto del batch di eventi che viene elaborato dalla pipe.

EventBridge include questo campo solo se è stato specificato di includere i dati di esecuzione nei log di questa pipe. Per ulteriori informazioni, consulta ???

> **Important**
>
> Questi campi possono contenere informazioni riservate. EventBridge non tenta di oscurare il contenuto di questi campi durante la registrazione.

Per ulteriori informazioni, consulta ???.

awsRequest

La richiesta inviata all'arricchimento o alla destinazione, in formato JSON. Per le richieste inviate a una destinazione API, rappresenta la richiesta HTTP inviata a quell'endpoint.

EventBridge include questo campo solo se è stato specificato di includere i dati di esecuzione nei log di questa pipe. Per ulteriori informazioni, consulta ???

> **Important**
>
> Questi campi possono contenere informazioni riservate. EventBridge non tenta di oscurare il contenuto di questi campi durante la registrazione.

Per ulteriori informazioni, consulta ???.

awsResponse

La risposta restituita dall'arricchimento o dalla destinazione, in formato JSON. Per le richieste inviate a una destinazione API, rappresenta la risposta HTTP restituita da quell'endpoint e non la risposta restituita dal servizio Destinazione API.

EventBridge include questo campo solo se è stato specificato di includere i dati di esecuzione nei log di questa pipe. Per ulteriori informazioni, consulta ???

> **Important**
>
> Questi campi possono contenere informazioni riservate. EventBridge non tenta di oscurare il contenuto di questi campi durante la registrazione.

Per ulteriori informazioni, consulta ???.

truncatedFields

Un elenco di tutti i campi dei dati di esecuzione EventBridge è stato troncato per mantenere il record al di sotto del limite di 256 KB.

Se EventBridge non è stato necessario troncare nessuno dei campi dei dati di esecuzione, questo campo è presente ma. `null`

Per ulteriori informazioni, consulta ???.

error

Contiene informazioni per eventuali errori generati durante questo passaggio di esecuzione della pipe.

Se non è stato generato alcun errore durante questo passaggio di esecuzione della pipe, questo campo è presente ma `null`.

httpStatusCode

Il codice di stato HTTP restituito dal servizio chiamato.

message

Il messaggio di errore restituito dal servizio chiamato.

details

Qualsiasi informazione dettagliata sull'errore restituita dal servizio chiamato.

awsService

Il nome del servizio chiamato.

requestId

L'ID richiesta per questa richiesta dal servizio chiamato.

Registrazione e monitoraggio di Amazon EventBridge Pipes utilizzando AWS CloudTrail e Amazon CloudWatch Logs

Puoi registrare le chiamate e l'utilizzo di EventBridge Pipes CloudTrail e monitorare lo stato delle tue pipe utilizzando CloudWatch le metriche.

CloudWatch metriche

EventBridge Pipes invia i parametri ad Amazon CloudWatch ogni minuto per qualsiasi cosa, dalla limitazione delle esecuzioni di una pipe alla corretta invocazione di un bersaglio.

Parametro	Descrizione	Dimensioni	Unità
`Concurrency`	Il numero di esecuzioni simultanee di una pipe.	AwsAccountId	Nessuno
`Duration`	Periodo di tempo necessario per l'esecuzione della pipe.	PipeName	Millisecondi
`EventCount`	Il numero di eventi elaborati da una pipe.	PipeName	Nessuno
`EventSize`	La dimensione del payload dell'evento che ha richiamato la pipe.	PipeName	Byte
`ExecutionThrottled`	Il numero di esecuzioni di una pipe che sono state limitate.	AwsAccountId, PipeName	Nessuno

Parametro	Descrizione	Dimensioni	Unità
	Note Questo valore sarà `0` se nessuna esecuzione è stata limitata.		
`ExecutionTimeout`	Il numero di esecuzioni di una pipe per le quali si è verificato un timeout prima del completamento dell'esecuzione. **Note** Questo valore sarà `0` se non si è verificato il timeout di alcuna esecuzione.	PipeName	Nessuno
`ExecutionFailed`	Quante esecuzioni di una pipe non sono riuscite. **Note** Questo valore sarà `0` se nessuna esecuzione non riesce.	PipeName	Nessuno
`ExecutionPartiallyFailed`	Quante esecuzioni di una pipe non sono riuscite parzialmente. **Note** Questo valore sarà `0` se nessuna esecuzione non riesce.	PipeName	Nessuno
`EnrichmentStageDuration`	Il tempo necessario per il completamento della fase di arricchimento.	PipeName	Millisecondi

Parametro	Descrizione	Dimensioni	Unità
`EnrichmentStageFailed`	Quante esecuzioni di una fase di arricchimento di una pipe non sono riuscite. **Note** Questo valore sarà `0` se nessuna esecuzione non riesce.	PipeName	Nessuno
`Invocations`	Il numero totale di invocazioni.	AwsAccountId, PipeName	Nessuno
`TargetStageDuration`	Il tempo necessario per il completamento della fase di destinazione.	PipeName	Millisecondi
`TargetStageFailed`	Quante esecuzioni di una fase di destinazione di una pipe non sono riuscite. **Note** Questo valore sarà `0` se nessuna esecuzione non riesce.	PipeName	Nessuno
`TargetStagePartiallyFailed`	Quante esecuzioni di una fase di destinazione di una pipe non sono parzialmente riuscite. **Note** Questo valore sarà `0` se nessuna esecuzione della fase di destinazione non riesce.	PipeName	Nessuno

Parametro	Descrizione	Dimensioni	Unità
`TargetStageSkipped`	Quante esecuzioni della fase di destinazione di una pipe sono state ignorate (ad esempio, a causa dell'arricchimento che ha restituito un payload vuoto).	PipeName	Conteggio

CloudWatch Dimensioni per le metriche

CloudWatch le metriche hanno dimensioni, o attributi ordinabili, che sono elencati di seguito.

Dimensione	Descrizione
`AwsAccountId`	Filtra le metriche disponibili per ID account.
`PipeName`	Filtra le metriche disponibili per nome di pipe.

Gestione e risoluzione degli errori di Amazon EventBridge Pipes

Comportamento di ripetizione e gestione degli errori

EventBridge Pipes riprova automaticamente l'arricchimento e l'invocazione del target in caso di AWS errori ripetibili con il servizio di origine, i servizi di arricchimento o di destinazione oppure. EventBridge Tuttavia, se si verificano errori restituiti da implementazioni di arricchimento o destinazione del cliente, la velocità del polling della pipe diminuirà gradualmente. Per errori 4xx quasi continui (come problemi di autorizzazione con IAM o risorse mancanti), la pipe può essere disattivata automaticamente con un messaggio esplicativo in `StateReason`.

Errori di invocazione di pipe e comportamento di ripetizione

Quando richiami una pipe, possono verificarsi due tipi principali di errori: errori interni alle pipe ed errori di invocazione del cliente.

Errori interni alle pipe

Gli errori interni di Pipe sono errori derivanti da aspetti della chiamata gestiti dal servizio Pipes. EventBridge

Questi tipi di errori possono includere problemi come:

- Un errore di connessione HTTP durante il tentativo di richiamare il servizio di destinazione del cliente
- Un calo transitorio nella disponibilità del servizio di pipe.

In generale, EventBridge Pipes ripete gli errori interni un numero indefinito di volte e si interrompe solo alla scadenza del record nell'origine.

Per le pipe con una sorgente di flusso, EventBridge Pipes non conta i tentativi per errori interni rispetto al numero massimo di tentativi specificato nella politica di ripetizione per l'origine del flusso. Per le pipe con un'origine Amazon SQS, EventBridge Pipes non conta i tentativi per errori interni rispetto al conteggio massimo di ricezione per l'origine Amazon SQS.

Errori di invocazione del cliente

Gli errori di invocazione del cliente sono errori derivanti dalla configurazione o dal codice gestito dall'utente.

Questi tipi di errori possono includere problemi come:

- Autorizzazioni insufficienti sulla pipe per richiamare la destinazione.
- Un errore logico in un endpoint Gateway API o in una destinazione Lambda, Step Functions, API del cliente richiamata in modo sincrono.

Per gli errori di invocazione dei clienti, EventBridge Pipes esegue le seguenti operazioni:

- Per le pipe con una sorgente di flusso, EventBridge Pipes riprova fino ai tempi massimi di riprova configurati nella politica di riprova delle pipe o fino alla scadenza dell'età massima di registrazione, a seconda dell'evento che si verifica per primo.
- Per le pipe con una fonte Amazon SQS, EventBridge Pipes ripete un errore del cliente fino al numero massimo di ricezione nella coda di origine.
- Per le pipe con una fonte Apache Kafka o Amazon MQ, EventBridge ritenta gli errori del cliente allo stesso modo in cui ritenta gli errori interni.

Per le pipe con obiettivi di calcolo, è necessario richiamare la pipe in modo sincrono affinché EventBridge Pipes venga a conoscenza di eventuali errori di runtime generati dalla logica di calcolo

del cliente e riprovi a correggere tali errori. Le pipe non possono effettuare nuovi tentativi per gli errori generati dalla logica di un flusso di lavoro standard di Step Functions, poiché questa destinazione deve essere richiamata in modo asincrono.

Per Amazon SQS e sorgenti di flusso, come Kinesis e DynamoDB, Pipes supporta la gestione parziale degli errori in batch degli EventBridge errori di destinazione. Per ulteriori informazioni, consulta Errori batch parziali.

Comportamento DLQ delle pipe

Una pipe eredita il comportamento della coda DLQ dall'origine:

- Se la coda Amazon SQS di origine ha una coda DLQ configurata, i messaggi vengono distribuiti automaticamente in quella coda dopo il numero di tentativi specificato.
- Per le origini di flusso, come i flussi DynamoDB e Kinesis, puoi configurare una coda DLQ per gli eventi di pipe e instradamento. Le origini di flusso DynamoDB e Kinesis supportano le code Amazon SQS e gli argomenti Amazon SNS come destinazioni DLQ.

Se specifichi `DeadLetterConfig` per una pipe con un'origine Kinesis o DynamoDB, assicurati che la proprietà `MaximumRecordAgeInSeconds` della pipe sia inferiore a `MaximumRecordAge` dell'evento di origine. `MaximumRecordAgeInSeconds` controlla quando lo strumento per il polling delle pipe rinuncerà all'evento e lo distribuirà alla coda DLQ e `MaximumRecordAge` controlla per quanto tempo il messaggio sarà visibile nel flusso di origine prima che venga eliminato. Pertanto, imposta `MaximumRecordAgeInSeconds` su un valore inferiore all'origine `MaximumRecordAge` di modo che vi sia un intervallo di tempo adeguato tra il momento in cui l'evento viene inviato alla coda DLQ e quello in cui viene eliminato automaticamente dall'origine, in modo da determinare il motivo per cui l'evento è stato inviato alla coda DLQ.

Per le origini Amazon MQ, la coda DLQ può essere configurata direttamente nel broker di messaggi.

EventBridge Pipes non supporta DLQ FIFO (first-in first-out) per le sorgenti di streaming.

EventBridge Pipes non supporta DLQ per lo stream Amazon MSK e le sorgenti di flusso Apache Kafka gestite autonomamente.

Stati di errore delle pipe

La creazione, l'eliminazione e l'aggiornamento delle pipe sono operazioni asincrone che possono causare uno stato di errore. Allo stesso modo, una pipe può essere disabilitata automaticamente

a causa di errori. In tutti i casi, la proprietà `StateReason` della pipe fornisce informazioni utili a risolvere il problema.

Di seguito è riportato un elenco dei possibili valori di `StateReason`:

- Il flusso non è stato trovato. Per riprendere l'elaborazione, elimina la pipe e creane una nuova.
- Pipes non dispone delle autorizzazioni necessarie per eseguire le operazioni di coda (sqs:, sqs: e sqs:) ReceiveMessage DeleteMessage GetQueueAttributes
- Errore di connessione. Il VPC deve essere in grado di connettersi alle pipe. È possibile fornire l'accesso configurando un gateway NAT o un endpoint VPC a pipes-data. Per come configurare il gateway NAT o l'endpoint VPC su pipes-data, consulta la documentazione. AWS
- Al cluster MSK non sono associati gruppi di sicurezza

Una pipe può essere interrotta automaticamente con la proprietà `StateReason` aggiornata. Le ragioni possibili sono:

- Un flusso di lavoro standard di Step Functions configurato come arricchimento.
- Un flusso di lavoro standard Step Functions configurato come destinazione da richiamare in modo sincrono.

Errori di crittografia personalizzata

Se configuri una fonte per utilizzare una chiave di crittografia AWS KMS personalizzata (CMK), anziché una chiave AWS-managed, devi fornire esplicitamente l'autorizzazione di decrittografia del AWS KMS ruolo di esecuzione della tua pipe. A tale scopo, includi la seguente autorizzazione aggiuntiva nella policy CMK personalizzata:

```
  {
      "Sid": "Allow Pipes access",
      "Effect": "Allow",
      "Principal": {
          "AWS": "arn:aws:iam::01234567890:role/service-role/
Amazon_EventBridge_Pipe_DDBStreamSourcePipe_12345678"
      },
      "Action": "kms:Decrypt",
      "Resource": "*"
  }
```

Sostituisci il ruolo sopra con il ruolo di esecuzione della pipe.

Questo vale per tutte le sorgenti pipe con CMK, tra cui: AWS KMS

- Amazon DynamoDB Streams
- Amazon Kinesis Data Streams
- Amazon MQ
- MSK Amazon
- Amazon SQS

Tutorial: creazione di una pipe EventBridge che filtra gli eventi di origine

In questo tutorial, creerai una pipe che collega un'origine di flusso DynamoDB a una destinazione di coda Amazon SQS. Ciò include la selezione di un modello di eventi che la pipe deve utilizzare per filtrare gli eventi da distribuire alla coda. Quindi testerai la pipe per assicurarti che vengano distribuiti solo gli eventi desiderati.

Prerequisiti: creare l'origine e la destinazione

Prima di creare la pipe, devi creare l'origine e la destinazione a cui la pipe deve essere collegata. In questo caso, un flusso di dati Amazon DynamoDB che funge da origine della pipe e una coda Amazon SQS come destinazione della pipe.

Per semplificare questo passaggio, puoi utilizzare AWS CloudFormation per eseguire il provisioning delle risorse di origine e di destinazione. A questo proposito, creerai un modello CloudFormation che definisce le seguenti risorse:

- L'origine della pipe

  Una tabella Amazon DynamoDB, denominata `pipe-tutorial-source`, con un flusso abilitato per fornire un flusso ordinato di informazioni sulle modifiche apportate agli elementi nella tabella DynamoDB.
- La destinazione della pipe

  Una coda Amazon SQS, denominata `pipe-tutorial-target`, per ricevere il flusso di eventi DynamoDB dalla tua pipe.

Per creare il modello CloudFormation per il provisioning delle risorse della pipe

1. Copia il testo del modello JSON nella sezione ??? seguente.
2. Salva il modello come file JSON (ad esempio, `~/pipe-tutorial-resources.json`).

Successivamente, utilizza il file di modello che hai appena creato per eseguire il provisioning di uno stack CloudFormation.

> **Note**
>
> Una volta creato lo stack CloudFormation, ti verranno addebitate le risorse AWS di cui esegue il provisioning.

Provisioning dei requisiti del tutorial utilizzando l'AWS CLI

- Esegui il comando CLI seguente, dove `--template-body` specifica la posizione del file di modello:

```
aws cloudformation create-stack --stack-name pipe-tuturial-resources --template-
body file://~/pipe-tutorial-resources.json
```

Provisioning dei prerequisiti del tutorial utilizzando la console CloudFormation

1. Apri la console di AWS CloudFormation all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Scegli Stack, quindi seleziona Crea stack e scegli Con nuove risorse (standard).

   CloudFormation visualizza la procedura guidata Crea stack.
3. In Prerequisito - Prepara modello, lascia selezionato il valore predefinito, ovvero Il modello è pronto.
4. In Specifica modello, seleziona Carica un file di modello e quindi scegli il file e seleziona Successivo.
5. Configura lo stack e le risorse che fornisce:
   - In Nome stack, immetti `pipe-tuturial-resources`.
   - In Parametri, lascia i nomi predefiniti per la tabella DynamoDB e la coda Amazon SQS.

- Scegli Successivo.

6. Scegli Successivo, quindi scegli Invia.

   CloudFormation crea lo stack ed esegue il provisiong delle risorse definite nel modello.

Per ulteriori informazioni su CloudFormation, consulta Cos'è AWS CloudFormation? nella Guida per l'utente di AWS CloudFormation.

Passaggio 1: creare la pipe

Dopo aver eseguito il provisioning dell'origine e della destinazione della pipe, ora è possibile creare la pipe per connettere i due servizi.

Creazione della pipe utilizzando la console EventBridge

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Pipe.
3. Scegli Crea pipe.
4. In Nome, immetti un nome per la pipe `pipe-tutorial`.
5. Specifica l'origine del flusso di dati DynamoDB:

   a. In Dettagli, per Origine, seleziona Flusso di dati DynamoDB.

      EventBridge visualizza le impostazioni di configurazione dell'origine specifiche di DynamoDB.

   b. In Flusso DynamoDB, seleziona `pipe-tutorial-source`.

      Lascia Posizione di partenza impostata sul valore predefinito, `Latest`.

   c. Scegli Successivo.

6. Specifica e testa un modello di eventi per filtrare gli eventi:

   I filtri consenteno di determinare gli eventi che le pipe inviano all'arricchimento o alla destinazione. La pipe invia all'arricchimento o alla destinazione solo gli eventi che corrispondono al modello di eventi.

   Per ulteriori informazioni, consulta ???.

> **Note**
>
> Ti vengono fatturati solo gli eventi inviati all'arricchimento o alla destinazione.

a. In Evento di esempio (facoltativo), lascia selezionato Eventi AWS e assicurati che sia selezionato Evento di esempio di flusso DynamoDB 1.

   Questo è l'evento di esempio che utilizzerai per testare il nostro modello di eventi.

b. In Modello di eventi, immetti il seguente modello di eventi:

```
{
  "eventName": ["INSERT", "MODIFY"]
}
```

c. Scegli Modello di test.

   EventBridge visualizza un messaggio indicante che l'evento di esempio corrisponde al modello di eventi. Questo perché nell'evento il valore `eventName` è `INSERT`.

d. Scegli Successivo.

7. Scegli Successivo per non specificare un arricchimento.

   In questo esempio, non selezionerai un arricchimento. Gli arricchimenti ti consentono di selezionare un servizio per migliorare i dati dall'origine prima di inviarli alla destinazione. Per ulteriori dettagli, consulta ???.

8. Specifica la coda Amazon SQS come destinazione della pipe:

   a. In Dettagli, per Servizio di destinazione, seleziona Coda Amazon SQS.

   b. In Coda, seleziona `pipe-tutorial-target`.

   c. Lascia vuota la sezione Trasformatore di input di destinazione.

      Per ulteriori informazioni, consulta ???.

9. Seleziona Crea pipe.

   EventBridge crea la pipe e visualizza la pagina dei dettagli della pipe. La pipe è pronta quando il relativo stato è `Running`.

Passaggio 2: confermare gli eventi dei filtri della pipe

La pipe è configurata, ma non ha ancora ricevuto eventi dalla tabella.

Per testare la pipe, aggiornerai le voci nella tabella DynamoDB. Ogni aggiornamento genererà eventi che il flusso DynamoDB invia alla nostra pipe. Alcuni corrisponderanno al modello di eventi specificato, altri no. Puoi quindi esaminare la coda Amazon SQS per assicurarti che la pipe abbia distribuito solo gli eventi che corrispondono al nostro modello di eventi.

Aggiornamento degli elementi della tabella per generare eventi

1. Apri la console DynamoDB all'indirizzo https://console.aws.amazon.com/dynamodb/.
2. Nel riquadro di navigazione sinistro, seleziona Tabelle. Seleziona la tabella `pipe-tutorial-source`.

   DynamoDB visualizza la pagina dei dettagli della tabella per `pipe-tutorial-source`.
3. Seleziona Esplora elementi della tabella, quindi scegli Crea elemento.

   DynamoDB visualizza la pagina Crea elemento.
4. In Attributi, crea un nuovo elemento della tabella:

   a. In Album immetti `Album A`.
   b. In Artista, immetti `Artist A`.
   c. Scegli Crea elemento.
5. Aggiorna l'elemento della tabella:

   a. In Elementi restituiti, scegli Album A.
   b. Seleziona Aggiungi nuovo attributo, quindi seleziona Stringa.
   c. Immetti un nuovo valore di `Song`, con un valore di`Song A`.
   d. Seleziona Salva modifiche.
6. Elimina l'elemento della tabella:

   a. In Elementi restituiti, seleziona Album A.
   b. Nel menu Azioni, seleziona Elimina elementi.

Hai effettuato tre aggiornamenti all'elemento della tabella e ciò ha generato tre eventi per il flusso di dati DynamoDB:

- Un evento `INSERT` al momento della creazione dell'elemento.
- Un evento `MODIFY` quando hai aggiunto un attributo all'elemento.
- Un evento `REMOVE` quando hai eliminato l'elemento.

Tuttavia, il modello di eventi specificato per la pipe deve escludere, filtrandoli, tutti gli eventi che non sono eventi `INSERT` o `MODIFY`. Successivamente, conferma che la pipe abbia distribuito gli eventi previsti alla coda.

Conferma della distribuzione degli eventi previsti alla coda

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/sqs/.
2. Scegli la coda `pipe-tutorial-target`.

   Amazon SQS visualizza la pagina dei dettagli della coda.
3. Seleziona Invia e ricevi messaggi, quindi in Ricevi messaggi, scegli Polling per messaggi.

   La coda esegue il polling della pipe e quindi elenca gli eventi che riceve.
4. Scegli il nome dell'evento per vedere il JSON dell'evento che è stato distribuito.

Dovrebbero esserci due eventi nella coda: uno con `eventName` di `INSERT` e uno con `eventName` di `MODIFY`. Tuttavia, la pipe non ha distribuito l'evento per l'eliminazione dell'elemento della tabella, poiché quell'evento aveva `eventName` di `REMOVE`, che non corrispondeva al modello di eventi specificato nella pipe.

Fase 3: eliminazione delle risorse

Innanzitutto, elimina la pipe.

Eliminazione della pipe utilizzando la console EventBridge

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Pipe.
3. Selezionare la pipe `pipe-tutorial` e scegli Elimina.

Quindi, elimina lo stack CloudFormation, per evitare che ti venga addebitato l'utilizzo continuato delle risorse fornite nello stesso.

Eliminazione dei prerequisiti del tutorial utilizzando l'AWS CLI

- Esegui il comando della CLI seguente, dove `--stack-name` specifica il nome del tuo stack:

```
aws cloudformation delete-stack --stack-name pipe-tuturial-resources
```

Eliminazione dei prerequisiti del tutorial utilizzando la console AWS CloudFormation

1. Apri la console di AWS CloudFormation all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Nella pagina Stack, seleziona lo stack, quindi seleziona Elimina.
3. Seleziona Elimina per confermare l'azione.

Modello AWS CloudFormation per la generazione di prerequisiti

Utilizza il codice JSON riportato di seguito per creare un modello CloudFormation per il provisioning delle risorse di origine e destinazione necessarie per questo tutorial.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",

 "Description" : "Provisions resources to use with the EventBridge Pipes tutorial. You
 will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {
    "SourceTableName" : {
      "Type" : "String",
      "Default" : "pipe-tutorial-source",
      "Description" : "Specify the name of the table to provision as the pipe source,
 or accept the default."
    },
  "TargetQueueName" : {
    "Type" : "String",
    "Default" : "pipe-tutorial-target",
    "Description" : "Specify the name of the queue to provision as the pipe target, or
 accept the default."
  }
},
  "Resources": {
    "PipeTutorialSourceDynamoDBTable": {
```

```
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "AttributeDefinitions": [{
            "AttributeName": "Album",
            "AttributeType": "S"
          },
          {
            "AttributeName": "Artist",
            "AttributeType": "S"
          }

        ],
        "KeySchema": [{
            "AttributeName": "Album",
            "KeyType": "HASH"

          },
          {
            "AttributeName": "Artist",
            "KeyType": "RANGE"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 10,
          "WriteCapacityUnits": 10
        },
        "StreamSpecification": {
          "StreamViewType": "NEW_AND_OLD_IMAGES"
        },
        "TableName": { "Ref" : "SourceTableName" }
      }
    },
    "PipeTutorialTargetQueue": {
      "Type": "AWS::SQS::Queue",
      "Properties": {
        "QueueName": { "Ref" : "TargetQueueName" }
      }
    }
  }
}
```

Genera un AWS CloudFormation modello da EventBridge Pipes

AWS CloudFormation consente di configurare e gestire AWS le risorse tra account e regioni in modo centralizzato e ripetibile trattando l'infrastruttura come codice. CloudFormation lo fa consentendoti di creare modelli che definiscono le risorse che desideri fornire e gestire.

EventBridge ti consente di generare modelli a partire dalle pipe esistenti nel tuo account, come aiuto per iniziare subito a sviluppare modelli. CloudFormation Puoi selezionare un singolo pipe o più pipe da includere nel modello. È quindi possibile utilizzare questi modelli come base per creare pile di risorse da gestire. CloudFormation

Per ulteriori informazioni su CloudFormation, consulta la Guida per l' AWS CloudFormation utente.

Per i bus degli eventi, puoi generare CloudFormation modelli a partire dai bus degli eventi e dalle regole dei bus degli eventi.

Risorse incluse nei modelli EventBridge Pipe

Quando EventBridge genera il CloudFormation modello, crea una AWS::Pipes::Piperisorsa per ogni tubo selezionato. Inoltre, EventBridge include le seguenti risorse nelle condizioni descritte:

- AWS::Events::ApiDestination

  Se le tue pipe includono destinazioni API, come arricchimenti o come destinazioni, le EventBridge include nel CloudFormation modello come AWS::Events::ApiDestination risorse.

- AWS::Events::EventBus

  Se le tue pipe includono un bus di eventi come destinazione, lo EventBridge include nel CloudFormation modello come AWS::Events::EventBus risorsa.

- AWS::IAM::Role

  Se avete EventBridge creato un nuovo ruolo di esecuzione quando avete configurato la pipe, potete scegliere di EventBridge includere quel ruolo nel modello come AWS::IAM::Role risorsa. EventBridge non include i ruoli creati dall'utente. (In entrambi i casi, la `RoleArn` proprietà della AWS::Pipes::Pipe risorsa contiene l'ARN del ruolo.)

Considerazioni sull'utilizzo di CloudFormation modelli generati da Pipes EventBridge

Considerate i seguenti fattori quando utilizzate un CloudFormation modello generato da EventBridge:

- EventBridge non include alcuna password nel modello generato.

  È possibile modificare il modello per includere i parametri del modello che consentono agli utenti di specificare password o altre informazioni riservate quando lo utilizzano per creare o aggiornare uno CloudFormation stack.

  Inoltre, gli utenti possono utilizzare Secrets Manager per creare un segreto nella Regione desiderata e quindi modificare il modello generato per utilizzare parametri dinamici.
- Le destinazioni nel modello generato rimangono esattamente come specificate nel pipe originale. Se il modello non viene modificato in modo appropriato prima di utilizzarlo per creare stack in altre Regioni, è possibile che si abbiano problemi in più Regioni.

  Inoltre, il modello generato non creerà automaticamente destinazioni a valle.

Generazione di un CloudFormation modello da Pipes EventBridge

Per generare un CloudFormation modello da una o più pipe utilizzando la EventBridge console, effettuate le seguenti operazioni:

Per generare un CloudFormation modello da una o più pipe

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Pipe.
3. In Pipes, scegli una o più pipe che desideri includere nel CloudFormation modello generato.

   Per un singolo pipe, puoi anche scegliere il nome del pipe per visualizzare la pagina dei dettagli del pipe.
4. Scegliete CloudFormation Modello, quindi scegliete il formato in cui desiderate EventBridge generare il modello: JSON o YAML.

   EventBridge visualizza il modello, generato nel formato selezionato.

5. Se hai EventBridge creato un nuovo ruolo di esecuzione per una qualsiasi delle pipe selezionate e desideri EventBridge includere tali ruoli nel modello, scegli Includi IAM i ruoli creati dalla console per tuo conto.
6. EventBridge offre la possibilità di scaricare il file modello o di copiare il modello negli appunti.
   - Per scaricare il file di modello, scegli Scarica.
   - Per copiare il modello negli appunti, scegli Copia.
7. Per uscire dal modello, scegli Annulla.

Rendere le applicazioni tolleranti ai guasti a livello regionale con endpoint globali e replica degli eventi

Puoi migliorare la disponibilità della tua applicazione con gli endpoint EventBridge globali di Amazon. Gli endpoint globali aiutano a rendere l'applicazione tollerante ai guasti a livello regionale senza costi aggiuntivi. Innanzitutto, devi assegnare un controllo dell'integrità Amazon Route 53 all'endpoint. Quando viene avviato il failover, il controllo dell'integrità segnala uno stato "non integro". Entro pochi minuti dall'avvio del failover, tutti gli eventi personalizzati vengono instradati a un router di eventi nella Regione secondaria e vengono elaborati da tale router di eventi. Non appena il controllo dell'integrità segnala uno stato "integro", gli eventi vengono elaborati dal router di eventi nella Regione primaria.

Quando utilizzi endpoint globali, puoi abilitare la replica degli eventi. La replica degli eventi invia tutti gli eventi personalizzati ai router di eventi nelle Regioni primarie e secondarie utilizzando regole gestite.

> **Note**
>
> Se utilizzi router personalizzati, avrai bisogno di un router personalizzato in ogni Regione con lo stesso nome e nello stesso account affinché il failover funzioni correttamente.

Argomenti

Obiettivi del tempo di ripristino e del punto di ripristino

L'obiettivo del tempo di ripristino (RTO) è il tempo necessario alla Regione secondaria per iniziare a ricevere eventi dopo un guasto. Per quanto riguarda RTO, il periodo include il periodo di tempo

necessario per l'attivazione degli CloudWatch allarmi e l'aggiornamento degli stati dei controlli di integrità della Route 53. L'obiettivo del punto di ripristino (RPO) è la misura dei dati che rimarranno non elaborati in caso di guasto. Per quanto riguarda l'RPO, il periodo include gli eventi che non vengono replicati nella Regione secondaria e che rimangono bloccati nella Regione principale fino al ripristino del servizio o della Regione. Per quanto riguarda gli endpoint globali, se segui le nostre linee guida prescrittive per la configurazione degli allarmi, puoi prevedere che l'RTO e l'RPO siano compresi tra 360 e 420 secondi.

Replica di eventi

Gli eventi vengono elaborati nella Regione secondaria in modo asincrono. Ciò significa che non è garantito che gli eventi vengano elaborati contemporaneamente in entrambe le Regioni. Quando viene attivato il failover, gli eventi vengono elaborati dalla Regione secondaria e verranno elaborati dalla Regione primaria quando questa è disponibile. L'abilitazione della replica degli eventi comporterà un aumento dei costi mensili. Per ulteriori informazioni, consulta i EventBridgeprezzi di Amazon

Consigliamo di abilitare la replica degli eventi durante la configurazione degli endpoint globali per i seguenti motivi:

- La replica degli eventi consente di verificare la corretta configurazione degli endpoint globali. In questo modo, disporrai della copertura necessaria in caso di failover.
- La replica degli eventi è necessaria per il ripristino automatico da un evento di failover. Se non hai abilitato la replica degli eventi, dovrai reimpostare manualmente il controllo dell'integrità Route 53 su "integro" prima che gli eventi tornino nella Regione primaria.

Payload di evento replicato

Di seguito è riportato un esempio di payload di evento replicato:

> **Note**
>
> Per `region`, viene elencata la Regione da cui è stato replicato l'evento.

```
{
```

```
    "version": "0",
    "id": "a908baa3-65e5-ab77-367e-527c0e71bbc2",
    "detail-type": "Test",
    "source": "test.service.com",
    "account": "0123456789",
    "time": "1900-01-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:events:us-east-1:0123456789:endpoint/MyEndpoint"
    ],
    "detail": {
        "a": "b"
    }
}
```

Creazione di un endpoint globale

Completa i passaggi seguenti per configurare un endpoint globale:

1. Assicurati di disporre di regole e router di eventi corrispondenti sia nella Regione primaria che in quella secondaria.
2. Crea un controllo dell'integrità Route 53 per monitorare i tuoi router di eventi. Per ricevere assistenza nella creazione del controllo dell'integrità, scegli Nuovo controllo dell'integrità quando crei il tuo endpoint globale.
3. Crea l'endpoint globale.

Dopo aver configurato il controllo dell'integrità Route 53, puoi creare un endpoint globale.

Per creare un endpoint globale mediante la console

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Endpoint globali.
3. Scegliere Create Endpoint (Crea endpoint).
4. Immetti un nome e una descrizione per l'endpoint.
5. In Bus di eventi nella Regione primaria, scegli il router di eventi a cui desideri associare l'endpoint.
6. In Regione primaria, scegli la Regione verso cui indirizzare gli eventi in caso di failover.

> **Note**
>
> L'opzione Bus di eventi nella Regione secondaria viene compilato automaticamente e non è modificabile.

7. In Controllo dell'integrità Route 53 per attivazione del failover e ripristino, scegli il controllo dell'integrità che l'endpoint monitorerà. Se non disponi già di un controllo sanitario, scegli Nuovo controllo sanitario per aprire la AWS CloudFormation console e creare un controllo sanitario utilizzando un CloudFormation modello.

> **Note**
>
> In caso di dati mancanti, il controllo dell'integrità non verrà eseguito correttamente. Se devi inviare eventi solo a intermittenza, prendi in considerazione l'utilizzo di un programma più lungo MinimumEvaluationPeriodo considera i dati mancanti come «mancanti» anziché «violati».

8. (Facoltativo) In Replica degli eventi, procedi come segue:

   a. Seleziona Replica degli eventi abilitata.

   b. In Ruolo di esecuzione, scegli se creare un nuovo ruolo AWS Identity and Access Management o utilizzarne uno esistente. Esegui questa operazione:

      - Seleziona Create a new role for this specific resource (Crea un nuovo ruolo per questa risorsa specifica). Eventualmente, puoi aggiornare il campo Nome ruolo per creare un nuovo ruolo.
      - Scegli Utilizza un ruolo esistente. Quindi, in Ruolo di esecuzione, scegli il ruolo che intendi utilizzare.

9. Scegli Crea.

Per creare un endpoint globale utilizzando l'API

Per creare un endpoint globale utilizzando l' EventBridge API, consulta CreateEndpointAmazon EventBridge API Reference.

Per creare un endpoint globale utilizzando AWS CloudFormation

Per creare un endpoint globale utilizzando l' AWS CloudFormation API, consulta AWS::Events::Endpointsla Guida per l' AWS CloudFormation utente.

Lavorare con endpoint globali utilizzando un SDK AWS

> **Note**
>
> Il supporto per C++ sarà disponibile a breve.

Quando utilizzi un AWS SDK per lavorare con endpoint globali, tieni presente quanto segue:

- Dovrai avere la libreria AWS Common Runtime (CRT) installata per il tuo SDK specifico. Se non hai installato la libreria CRT, viene visualizzato un messaggio di eccezione che indica cosa deve essere installato. Per ulteriori informazioni, consulta gli argomenti seguenti:
  - Librerie AWS Common Runtime (CRT)
  - awslabs/ aws-crt-java
  - lastre in sega/ aws-crt-nodejs
  - lastre in sega/ aws-crt-python
- Dopo aver creato un endpoint globale, devi aggiungere `endpointId` e `EventBusName` a tutte le chiamate `PutEvents` che utilizzi.
- Gli endpoint globali supportano Signature Version 4A. Questa versione di SigV4 consente di firmare le richieste per più Regioni AWS. Ciò è utile nelle operazioni API che potrebbero comportare l'accesso ai dati da una tra più Regioni. Quando si utilizza l' AWS SDK, si forniscono le proprie credenziali e le richieste agli endpoint globali utilizzeranno la versione 4A della firma senza configurazioni aggiuntive. Per ulteriori informazioni su SigV4A, consulta Firma di richieste API di AWS in Riferimenti generali di AWS .

  Se richiedi credenziali temporanee all' AWS STS endpoint globale (sts.amazonaws.com), invia credenziali che, per impostazione predefinita, non supportano AWS STS SigV4A. Per ulteriori informazioni, consulta Managing in an Region nella Guida per l'utente. AWS STS AWS AWS Identity and Access Management

Regioni disponibili

Gli endpoint globali sono supportati nelle seguenti Regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- US West (Oregon)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Stoccolma)
- Asia Pacifico (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Sud America (San Paolo)

Best practice per l'utilizzo di endpoint globali di Amazon EventBridge

Le seguenti best practice sono consigliate per la configurazione degli endpoint globali.

Argomenti

Abilitazione della replica degli eventi

Ti consigliamo vivamente di attivare la replica ed elaborare gli eventi nella Regione secondaria assegnata al tuo endpoint globale. Ciò garantisce la corretta configurazione dell'applicazione nella Regione secondaria. Devi attivare la replica anche per garantire il ripristino automatico nella Regione primaria dopo che un problema è stato mitigato.

Gli ID evento possono cambiare nelle chiamate API, quindi la correlazione degli eventi nelle Regioni richiede un identificatore univoco e immutabile. I consumer devono inoltre essere progettati prendendo in considerazione l'idempotenza. In questo modo, se stai replicando eventi o riproducendoli da archivi, non vi sono effetti collaterali derivanti dall'elaborazione degli eventi in entrambe le Regioni.

Impedire la imitazione degli eventi

Per evitare che gli eventi vengano limitati, ti consigliamo di aggiornare i limiti relativi a `PutEvents` e alle destinazioni in modo che siano coerenti nelle Regioni.

Utilizzo delle metriche dell'abbonato nei controlli dell'integrità di Amazon Route 53

Evita di includere le metriche dell'abbonato nei controlli dell'integrità di Amazon Route 53. L'inclusione di queste metriche può provocare il failover nelle Regioni secondarie da parte dell'editore se un abbonato riscontra un problema nonostante tutti gli altri abbonati siano integri nella Regione primaria. Se uno dei tuoi abbonati non riesce a elaborare gli eventi nella Regione primaria, devi attivare la replica per assicurarti che il tuo abbonato nella Regione secondaria possa elaborare correttamente gli eventi.

Modello di AWS CloudFormation per configurare il controllo dell'integrità Route 53

Quando si utilizzano endpoint globali, è necessario effettuare un controllo dell'integrità Route 53 per monitorare lo stato delle Regioni. Il modello seguente definisce un allarme Amazon CloudWatch e lo utilizza per definire un controllo dell'integrità Route 53.

Argomenti

- Modello di AWS CloudFormation per definire un controllo dell'integrità Route 53
- Proprietà del modello di allarme CloudWatch

- Proprietà del modello del controllo dell'integrità Route 53

Modello di AWS CloudFormation per definire un controllo dell'integrità Route 53

Utilizza il seguente modello per definire il controllo dell'integrità Route 53.

```
Description: |-
  Global endpoints health check that will fail when the average Amazon EventBridge
  latency is above 30 seconds for a duration of 5 minutes. Note, missing data will
  cause the health check to fail, so if you only send events intermittently, consider
  changing the heath check to use a longer evaluation period or instead treat missing
  data as 'missing' instead of 'breaching'.

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: "Global endpoint health check alarm configuration"
        Parameters:
          - HealthCheckName
          - HighLatencyAlarmPeriod
          - MinimumEvaluationPeriod
          - MinimumThreshold
          - TreatMissingDataAs
    ParameterLabels:
      HealthCheckName:
        default: Health check name
      HighLatencyAlarmPeriod:
        default: High latency alarm period
      MinimumEvaluationPeriod:
        default: Minimum evaluation period
      MinimumThreshold:
        default: Minimum threshold
      TreatMissingDataAs:
        default: Treat missing data as

Parameters:
  HealthCheckName:
    Description: Name of the health check
    Type: String
    Default: LatencyFailuresHealthCheck
```

```
  HighLatencyAlarmPeriod:
    Description: The period, in seconds, over which the statistic is applied. Valid
 values are 10, 30, 60, and any multiple of 60.
    MinValue: 10
    Type: Number
    Default: 60
  MinimumEvaluationPeriod:
    Description: The number of periods over which data is compared to the specified
 threshold. You must have at least one evaluation period.
    MinValue: 1
    Type: Number
    Default: 5
  MinimumThreshold:
    Description: The value to compare with the specified statistic.
    Type: Number
    Default: 30000
  TreatMissingDataAs:
    Description: Sets how this alarm is to handle missing data points.
    Type: String
    AllowedValues:
      - breaching
      - notBreaching
      - ignore
      - missing
    Default: breaching

Mappings:
  "InsufficientDataMap":
    "missing":
      "HCConfig": "LastKnownStatus"
    "breaching":
      "HCConfig": "Unhealthy"

Resources:
  HighLatencyAlarm:
      Type: AWS::CloudWatch::Alarm
      Properties:
        AlarmDescription: High Latency in Amazon EventBridge
        MetricName: IngestionToInvocationStartLatency
        Namespace: AWS/Events
        Statistic: Average
        Period: !Ref HighLatencyAlarmPeriod
        EvaluationPeriods: !Ref MinimumEvaluationPeriod
        Threshold: !Ref MinimumThreshold
```

```
        ComparisonOperator: GreaterThanThreshold
        TreatMissingData: !Ref TreatMissingDataAs

  LatencyHealthCheck:
      Type: AWS::Route53::HealthCheck
      Properties:
        HealthCheckTags:
          - Key: Name
            Value: !Ref HealthCheckName
        HealthCheckConfig:
          Type: CLOUDWATCH_METRIC
          AlarmIdentifier:
            Name:
              Ref: HighLatencyAlarm
            Region: !Ref AWS::Region
          InsufficientDataHealthStatus: !FindInMap [InsufficientDataMap, !Ref
 TreatMissingDataAs, HCConfig]

Outputs:
  HealthCheckId:
    Description: The identifier that Amazon Route 53 assigned to the health check when
 you created it.
    Value: !GetAtt LatencyHealthCheck.HealthCheckId
```

Gli ID evento possono cambiare nelle chiamate API, quindi la correlazione degli eventi nelle Regioni richiede un identificatore univoco e immutabile. I consumer devono inoltre essere progettati prendendo in considerazione l'idempotenza. In questo modo, se stai replicando eventi o riproducendoli da archivi, non vi sono effetti collaterali derivanti dall'elaborazione degli eventi in entrambe le Regioni.

Proprietà del modello di allarme CloudWatch

> **Note**
>
> Per tutti i campi **`editable`**, prendi in considerazione la velocità di trasmissione effettiva al secondo. Se invii eventi solo a intermittenza, valuta la possibilità di modificare il controllo dell'integrità per utilizzare un periodo di valutazione più lungo o considera invece i dati mancanti come `missing` anziché `breaching`.

Le seguenti proprietà vengono utilizzate nella sezione degli allarmi CloudWatch del modello:

Metrica	Descrizione
`AlarmDescription`	La descrizione dell'allarme. Impostazione predefinita: **`High Latency in Amazon EventBridge`**
`MetricName`	Il nome del parametro associato all'allarme. È obbligatorio per un allarme basato su un parametro. Per un allarme basato su un'espressione matematica, puoi utilizzare invece `Metrics` e non puoi specificare `MetricName` . Impostazione predefinita: ingestionToInvocationStartLatency
`Namespace`	Lo spazio dei nomi del parametro associato all'allarme. È obbligatorio per un allarme basato su un parametro. Per un allarme basato su un'espressione matematica, non puoi specificare `Namespace` e devi invece utilizzare `Metrics`. Impostazione predefinita: `AWS/Events`
`Statistic`	Le statistiche del parametro associato all'allarme, diverse dai percentili. Impostazione predefinita: Media
`Period`	Il periodo, in secondi, durante il quale viene applicata la statistica. È obbligatorio per un allarme basato su un parametro. I valori validi sono 10, 30, 60 e qualsiasi multiplo di 60. Impostazione predefinita: **`60`**
`EvaluationPeriods`	Il numero di periodi in cui i dati vengono paragonati alla soglia specificata. Se si imposta un avviso che richiede la violazione di un numero di punti dati consecutivi per attivare l'avviso, questo valore specifica tale numero. Se si sta impostando un allarme «M da N», questo valore è N e `DatapointsToAlarm` è il valore M. Impostazione predefinita: **`5`**
`Threshold`	Il valore da confrontare con la statistica specificata.

Metrica	Descrizione
	Impostazione predefinita: **`30,000`**
`ComparisonOperator`	L'operazione aritmetica da utilizzare durante il confronto tra statistica e soglia specificate. Il valore statistico specificato viene usato come primo operando. Impostazione predefinita: `GreaterThanThreshold`
`TreatMissingData`	Imposta il modo in cui questo allarme dovrà gestire i punti di dati mancanti. Valori validi: `breaching` , `notBreaching` , `ignore` e `missing` Impostazione predefinita: `breaching`

Proprietà del modello del controllo dell'integrità Route 53

> **Note**
>
> Per tutti i campi **`editable`**, prendi in considerazione la velocità di trasmissione effettiva al secondo. Se invii eventi solo a intermittenza, valuta la possibilità di modificare il controllo dell'integrità per utilizzare un periodo di valutazione più lungo o considera invece i dati mancanti come `missing` anziché `breaching`.

Le seguenti proprietà sono utilizzate nella sezione relativa al controllo dell'integrità Route 53 del modello:

Metrica	Descrizione
`HealthCheckName`	.Il nome del controllo dell'integrità. Impostazione predefinita: **`LatencyFailuresHealthCheck`**
`InsufficientDataHealthStatus`	Quando CloudWatch ha dati insufficienti sulla metrica per determinare lo stato dell'allarme, lo stato che Amazon Route 53 deve assegnare al controllo dell'integrità:

Metrica	Descrizione
	Valori validi: • `Healthy`: Route 53 considera il controllo dello stato integro. • `Unhealthy` : Route 53 considera il controllo dello stato non integro. • `LastKnownStatus` : Route 53 utilizzerà lo stato del controllo dall'ulti ma volta in cui CloudWatch ha avuto dati sufficienti per determinare lo stato di allarme. Per i nuovi controlli dell'integrità che non hanno un ultimo stato noto, lo stato di default per il controllo dell'integrità è integro. Impostazione predefinita: Non integro **Note** Questo campo viene aggiornato in base all'input al campo `TreatMissingData` . Se `TreatingMissingData` è impostato su `Missing`, verrà aggiornato a `LastKnown Status` . Se `TreatingMissingData` è impostato su `Breaching` , verrà aggiornato a `Unhealthy` .

EventBridge Schemi Amazon

Uno schema definisce la struttura degli eventi a cui vengono inviati EventBridge. EventBridge fornisce schemi per tutti gli eventi generati dai AWS servizi. Puoi inoltre creare o caricare schemi personalizzati o dedurre schemi direttamente dagli eventi in un router di eventi. Quando disponi di uno schema per un evento, puoi scaricare le associazioni di codice per i linguaggi di programmazione più diffusi e accelerare la fase di sviluppo. È possibile utilizzare le associazioni di codice per gli schemi e gestire gli schemi dalla EventBridge console, utilizzando l'API o direttamente nell'IDE utilizzando i toolkit. AWS Per creare app serverless che utilizzano eventi, utilizza AWS Serverless Application Model.

> **Note**
>
> Quando si utilizza la funzionalità trasformatore di input, l'evento originale viene dedotto dall'individuazione dello schema, non l'evento trasformato inviato alla destinazione.

EventBridge supporta i formati OpenAPI 3 e JsonSchema Draft4.

Per AWS Toolkit for JetBrains e AWS Toolkit for VS Code, puoi sfogliare o cercare schemi e scaricare le associazioni di codice per gli schemi direttamente nel tuo IDE.

Il video seguente offre una panoramica degli schemi e dei registri di schemi: Using the Schema Registry

Argomenti

- Mascheramento del valore delle proprietà dell'API di registro di schemi
- Trovare uno EventBridge schema Amazon
- Registri EventBridge degli schemi Amazon
- Creazione di uno EventBridge schema Amazon
- Associazioni di EventBridge codice Amazon

Mascheramento del valore delle proprietà dell'API di registro di schemi

Alcuni valori di proprietà degli eventi utilizzate per creare un registro di schemi possono contenere informazioni riservate sui clienti. Per proteggere le informazioni del cliente, i valori verranno mascherati con asterischi (*). Poiché stiamo mascherando questi valori, EventBridge consigliamo di non creare applicazioni che dipendono esplicitamente dalle seguenti proprietà o dai relativi valori:

- CreateSchema— La `Content` proprietà del corpo `requestParameters`
- GetDiscoveredSchema— La `Events` proprietà del `requestParameters` corpo e la `Content` proprietà del `responseElements` corpo
- SearchSchemas— La `keywords` proprietà del `requestParameters`
- UpdateSchema— La `Content` proprietà di `requestParameters`

Trovare uno EventBridge schema Amazon

EventBridge include schemi per tutti i AWS servizi che generano eventi. Puoi trovare questi schemi nella EventBridge console oppure puoi trovarli utilizzando l'azione API. `SearchSchemas`

Per trovare schemi per i AWS servizi nella console EventBridge

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, seleziona Schemas (Schemi).
3. Nella pagina Schemi, seleziona Registro schemi eventi AWS .
   <result>

   Viene visualizzata la prima pagina degli schemi disponibili.
   </result>
4. Per trovare uno schema, in Cerca schemi di AWS eventi, inserisci un termine di ricerca.

   Una ricerca restituisce corrispondenze sia per il nome che per il contenuto degli schemi disponibili e visualizza le versioni dello schema che contengono corrispondenza.
5. Apri uno schema di eventi selezionando il nome dello schema.

Registri EventBridge degli schemi Amazon

I registri di schemi sono container di schemi. I registri di schemi raccolgono e organizzano gli schemi in gruppi logici. I registri di schemi predefiniti sono:

- Tutti gli schemi: tutti gli schemi dei registri degli AWS eventi, rilevati e degli schemi personalizzati.
- AWS registro degli schemi degli eventi: gli schemi incorporati.
- Registro schema individuato: gli schemi individuati con Individuazione schema.

Puoi inoltre creare registri personalizzati per organizzare gli schemi creati o caricati.

Per creare un registro di schemi personalizzato

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Schemi quindi Crea registro.
3. Nella pagina Dettagli del registro immetti un Nome.
4. (Facoltativo) Immetti una descrizione per il nuovo registro.
5. Scegli Crea.

Per creare uno schema personalizzato nel nuovo registro, seleziona Crea schemi personalizzati. Per aggiungere uno schema al registro, seleziona il registro quando crei un nuovo schema.

Per creare un registro mediante l'API, utilizza `CreateRegistry`. Per ulteriori informazioni, consulta Amazon EventBridge Schema Registry API Reference.

Per informazioni sull'utilizzo del registro degli EventBridge schemi tramite AWS CloudFormation, consulta EventSchemas Resource Type Reference in AWS CloudFormation.

Creazione di uno EventBridge schema Amazon

È possibile creare schemi utilizzando file JSON con la specifica OpenAPI o la specifica JsonSchema Draft4. Puoi creare o caricare i tuoi schemi utilizzando un modello o generando uno schema basato sul JSON di un evento. EventBridge Puoi anche dedurre lo schema da eventi in router di eventi. Per creare uno schema utilizzando l'API EventBridge Schema Registry, utilizza l'azione `CreateSchema`API.

Quando scegli tra i formati OpenAPI 3 e JsonSchema Draft4, tieni conto delle seguenti differenze:

- Il formato JsonSchema supporta parole chiave aggiuntive che non sono supportate in OpenAPI, ad esempio. `$schema, additionalItems`.
- Esistono piccole differenze nel modo in cui vengono gestite le parole chiave, ad esempio `type` e `format`.
- OpenAPI non supporta i collegamenti ipertestuali JsonSchema Hyper-Schema in documenti JSON.
- Gli strumenti per OpenAPI sono piuttosto relativi alla fase di compilazione, mentre gli strumenti per JsonSchema sono relativi alle operazioni di runtime, come strumenti client per la convalida dello schema.

Si consiglia di utilizzare il formato JsonSchema per implementare la convalida lato client in modo che gli eventi inviati siano conformi allo EventBridge schema. È possibile utilizzare JsonSchema per definire un contratto per documenti JSON validi e quindi utilizzare una convalida dello schema JSON prima di inviare gli eventi associati.

Dopo aver creato un nuovo schema, puoi scaricare le associazioni di codice utili per creare applicazioni per eventi con quello schema.

Argomenti

- Creazione di uno schema utilizzando un modello
- Modifica di un modello di schema direttamente nella console
- Creazione di uno schema per il JSON di un evento
- Crea uno schema da eventi in un router di eventi

Creazione di uno schema utilizzando un modello

È possibile creare uno schema da un modello o modificando un modello direttamente nella console. EventBridge Per scaricare il modello, devi scaricarlo dalla console. Modifica il modello in modo che lo schema corrisponda agli eventi. Quindi carica il nuovo modello tramite la console.

Per scaricare il modello di schema

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di spostamento, seleziona Schema registry (Registro degli schemi).
3. Nella sezione Getting started (Nozioni di base) in Schema template (Modello schema), scegli Download (Scarica).

In alternativa, puoi copiare il modello JSON dall'esempio di codice seguente.

```
{
    "openapi": "3.0.0",
    "info": {
      "version": "1.0.0",
      "title": "Event"
    },
    "paths": {},
    "components": {
      "schemas": {
        "Event": {
          "type": "object",
          "properties": {
            "ordinal": {
              "type": "number",
              "format": "int64"
            },
            "name": {
              "type": "string"
            },
            "price": {
              "type": "number",
              "format": "double"
            },
            "address": {
              "type": "string"
            },
```

```
            "comments": {
              "type": "array",
              "items": {
                "type": "string"
              }
            },
            "created_at": {
              "type": "string",
              "format": "date-time"
            }
          }
        }
      }
    }
  }
```

Per caricare un modello di schema

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Schemi quindi scegli Crea schema.
3. (Facoltativo) Seleziona o crea un registro di schemi.
4. In Dettagli dello schema, immetti un nome per lo schema.
5. (Facoltativo) Immetti una descrizione per lo schema.
6. In Tipo di schema, scegli OpenAPI 3.0 o JSON Schema Draft 4.
7. Nella scheda Crea, nella casella di testo, trascina il file dello schema nella casella di testo oppure incolla l'origine dello schema.
8. Seleziona Crea.

Modifica di un modello di schema direttamente nella console

Per modificare uno schema nella console

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Schemi quindi scegli Crea schema.
3. (Facoltativo) Seleziona o crea un registro di schemi.
4. In Dettagli dello schema, immetti un nome per lo schema.

5. In Tipo di schema, scegli OpenAPI 3.0 o JSON Schema Draft 4.
6. (Facoltativo) Puoi immettere una descrizione per lo schema da creare.
7. Nella scheda Crea, scegli Carica modello.
8. Nella casella di testo, modifica il modello in modo che lo schema corrisponda ai tuoi eventi.
9. Seleziona Crea.

Creazione di uno schema per il JSON di un evento

Se disponi del JSON di un evento, puoi creare automaticamente uno schema per quel tipo di evento.

Per creare uno schema basato sul JSON di un evento

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Schemi quindi scegli Crea schema.
3. (Facoltativo) Seleziona o crea un registro di schemi.
4. In Schema details (Dettagli schema) inserisci un nome per lo schema.
5. (Facoltativo) Puoi immettere una descrizione per lo schema creato.
6. In Tipo di schema, scegli OpenAPI 3.0.

   Non puoi usare JSONSchema quando crei uno schema dal JSON di un evento.
7. Seleziona Discover from JSON (Individua da JSON)
8. Nella casella di testo in JSON, incolla o trascina l'origine JSON di un evento.

   Ad esempio, puoi incollare il codice sorgente di questo AWS Step Functions evento per un'esecuzione non riuscita.

```
{
    "version": "0",
    "id": "315c1398-40ff-a850-213b-158f73e60175",
    "detail-type": "Step Functions Execution Status Change",
    "source": "aws.states",
    "account": "012345678912",
    "time": "2019-02-26T19:42:21Z",
    "region": "us-east-1",
    "resources": [
      "arn:aws:states:us-east-1:012345678912:execution:state-machine-
name:execution-name"
```

```
    ],
    "detail": {
        "executionArn": "arn:aws:states:us-east-1:012345678912:execution:state-
machine-name:execution-name",
        "stateMachineArn": "arn:aws:states:us-
east-1:012345678912:stateMachine:state-machine",
        "name": "execution-name",
        "status": "FAILED",
        "startDate": 1551225146847,
        "stopDate": 1551225151881,
        "input": "{}",
        "output": null
    }
}
```

9. Scegli Individua schema.
10. EventBridge genera uno schema OpenAPI per l'evento. Ad esempio, lo schema seguente viene generato per l'evento Step Functions precedente.

```
{
  "openapi": "3.0.0",
  "info": {
    "version": "1.0.0",
    "title": "StepFunctionsExecutionStatusChange"
  },
  "paths": {},
  "components": {
    "schemas": {
      "AWSEvent": {
        "type": "object",
        "required": ["detail-type", "resources", "detail", "id", "source", "time",
 "region", "version", "account"],
        "x-amazon-events-detail-type": "Step Functions Execution Status Change",
        "x-amazon-events-source": "aws.states",
        "properties": {
          "detail": {
            "$ref": "#/components/schemas/StepFunctionsExecutionStatusChange"
          },
          "account": {
            "type": "string"
          },
          "detail-type": {
            "type": "string"
```

```
          },
          "id": {
            "type": "string"
          },
          "region": {
            "type": "string"
          },
          "resources": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "source": {
            "type": "string"
          },
          "time": {
            "type": "string",
            "format": "date-time"
          },
          "version": {
            "type": "string"
          }
        }
      },
      "StepFunctionsExecutionStatusChange": {
        "type": "object",
        "required": ["output", "input", "executionArn", "name", "stateMachineArn",
"startDate", "stopDate", "status"],
        "properties": {
          "executionArn": {
            "type": "string"
          },
          "input": {
            "type": "string"
          },
          "name": {
            "type": "string"
          },
          "output": {},
          "startDate": {
            "type": "integer",
            "format": "int64"
          },
```

```
          "stateMachineArn": {
            "type": "string"
          },
          "status": {
            "type": "string"
          },
          "stopDate": {
            "type": "integer",
            "format": "int64"
          }
        }
      }
    }
  }
}
```

11. Una volta generato lo schema, scegli Crea.

Crea uno schema da eventi in un router di eventi

EventBridge può dedurre schemi scoprendo gli eventi. Per dedurre gli schemi, si attiva l'individuazione degli eventi in un router di eventi e ogni schema univoco viene aggiunto al registro di schemi, compresi quelli per eventi multi-account. Gli schemi scoperti da EventBridge vengono visualizzati nel registro degli schemi scoperti nella pagina Schemi.

Se il contenuto degli eventi sul bus degli eventi cambia, EventBridge crea nuove versioni dello schema correlato. EventBridge

> **Note**
>
> L'abilitazione dell'individuazione degli eventi in un router di eventi può comportare un costo. I primi cinque milioni di eventi elaborati ogni mese sono gratuiti.

> **Note**
>
> EventBridge per impostazione predefinita, deduce gli schemi dagli eventi tra account diversi, ma è possibile disabilitarlo aggiornando la proprietà. `cross-account` Per ulteriori informazioni, consulta Discoverers nello EventBridge Schema Registry API Reference.

Per abilitare l'individuazione dello schema in un router di eventi

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Esegui una di queste operazioni:
   - Per abilitare l'individuazione in un router di eventi predefinito, scegli Avvia individuazione.
   - Per abilitare l'individuazione in un router di eventi personalizzato, seleziona il pulsante di opzione per il router di eventi personalizzato e scegli Avvia individuazione.

Associazioni di EventBridge codice Amazon

Puoi generare associazioni di codice per schemi di eventi per accelerare lo sviluppo in Golang, Java, Python e. TypeScript Le associazioni di codice sono disponibili per eventi di servizi AWS , schemi che crei e per schemi che generati in base a eventi in un router di eventi. Puoi generare associazioni di codice per uno schema utilizzando la EventBridge console, l'API EventBridge Schema Registry o nel tuo IDE con un toolkit. AWS

Per generare associazioni di codice da uno schema EventBridge

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, seleziona Schemas (Schemi).
3. Trova uno schema per il quale desideri eseguire le associazioni di codice, cercando nei registri di schemi o cercando uno schema.
4. Seleziona il nome dello schema.
5. Nella pagina dei dettagli dello schema, nella sezione Versione, seleziona Scarica le associazioni del codice.
6. Nella pagina Download code bindings (Scarica associazioni di codice) selezionare la lingua delle associazioni di codice che si desidera scaricare.
7. Selezionare Download (Scarica).

   Potrebbero essere necessari alcuni secondi per l’avvio del download. Il file di download è un file zip di associazioni di codice per il linguaggio selezionato.

Servizi e strumenti correlati di Amazon EventBridge

Amazon EventBridge interagisce con altri strumenti e servizi AWS per elaborare eventi o richiamare una risorsa come destinazione di una regola. Per ulteriori informazioni sull'integrazione di EventBridge con altri strumenti e servizi AWS, consulta quanto segue:

Argomenti

Utilizzo di Amazon EventBridge con endpoint VPC di interfaccia

Se usi Amazon Virtual Private Cloud (Amazon VPC) per ospirare le risorse AWS, puoi stabilire una connessione privata tra il VPC ed EventBridge. Le risorse nel VPC possono utilizzare questa connessione per comunicare con EventBridge.

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 VPC a EventBridge, definisci un endpoint VPC di interfaccia per EventBridge. L'endpoint offre connettività scalabile e affidabile a EventBridge senza richiedere un gateway Internet, un'istanza Network Address Translation (NAT) 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 basati sulla tecnologia AWS PrivateLink, che consente la comunicazione privata tra servizi AWS utilizzando un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, consulta AWS PrivateLink ed endpoint VPC.

Quando utilizzi un endpoint VPC di interfaccia privata, gli eventi personalizzati che il tuo VPC invia a EventBridge utilizzano quell'endpoint. EventBridge invia quindi tali eventi ad altri servizi AWS in base alle regole e alle destinazioni che hai configurato. Una volta inviati gli eventi a un altro servizio, è possibile riceverli tramite l'endpoint pubblico o un endpoint VPC per quel servizio. Ad esempio, se crei una regola per inviare eventi a una coda Amazon SQS, puoi configurare un endpoint VPC di interfaccia per Amazon SQS per ricevere messaggi da quella coda nel tuo VPC senza utilizzare l'endpoint pubblico.

Disponibilità

EventBridge attualmente supporta endpoint VPC nelle seguenti Regioni:

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Mumbai)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Hong Kong)

- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Melbourne)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Seul)
- Asia Pacifico (Osaka-Locale)
- Canada (Centrale)
- Canada occidentale (Calgary)
- Cina (Pechino)
- China (Ningxia)
- Europa (Francoforte)
- Europa (Zurigo)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Spagna)
- Europa (Parigi)
- Europa (Stoccolma)
- Medio Oriente (Emirati Arabi Uniti)
- Medio Oriente (Bahrein)
- Sud America (San Paolo)
- Israele (Tel Aviv)
- AWS GovCloud (US-West)
- AWS GovCloud (US-East)

Creazione di un endpoint VPC per EventBridge

Per utilizzare EventBridge con il tuo VPC, crea un endpoint VPC di interfaccia per EventBridge e scegli com.amazonaws.***Region***.events come nome del servizio. Per ulteriori informazioni, consulta Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

Specifiche di EventBridge Pipes

Il supporto completo di EventBridge Pipes per gli endpoint VPC di interfaccia non è disponibile. Per utilizzare le seguenti origini in un VPC con EventBridge Pipes, consulta quanto segue:

- Configurazione di rete Amazon MSK
- Configurazione di rete Apache Kafka autogestita
- Configurazione di rete Amazon MQ

Integrazione di Amazon EventBridge con AWS X-Ray

È possibile utilizzare AWS X-Ray per tracciare gli eventi che passano attraverso EventBridge. EventBridge passa l'intestazione di traccia originale alla destinazione in modo che i servizi di destinazione possano tracciare, analizzare ed eseguire il debug.

EventBridge può passare un'intestazione di traccia per un evento solo se l'evento proviene da una richiesta `PutEvents` che ha passato il contesto di traccia. X-Ray non traccia gli eventi che provengono da partner, eventi pianificati o AWSservizi di terze parti e queste origini di eventi non vengono visualizzate nella mappa dei servizi X-Ray.

X-Ray convalida le intestazioni di traccia e quelle non valide vengono eliminate. Tuttavia, l'evento continua a essere elaborato.

> **Important**
>
> L'intestazione di traccia non è disponibile nell'evento che viene distribuito alla destinazione dell'invocazione.
>
> - Se disponi di un archivio di eventi, l'intestazione di traccia non è disponibile negli eventi archiviati. Se riproduci eventi archiviati, l'intestazione di traccia non è inclusa.
> - Se disponi di una coda DLQ, l'intestazione di traccia è inclusa nella richiesta `SendMessage` che invia l'evento alla coda DLQ. Se recuperi eventi (messaggi) dalla coda DLQ utilizzando `ReceiveMessage`, l'intestazione di traccia associata all'evento è inclusa nell'attributo del messaggio Amazon SQS, ma non è inclusa nel messaggio dell'evento.

Per informazioni su come un nodo evento EventBridge collega i servizi di origine e quelli di destinazione, consulta Visualizzazione dell'origine e delle destinazioni nella mappa dei servizi X-Ray nella Guida per gli sviluppatori di AWS X-Ray.

È possibile passare le seguenti informazioni sull'intestazione di traccia tramite EventBridge:

- Intestazione HTTP predefinita: l'SDK X-Ray popola automaticamente l'intestazione di traccia come intestazione HTTP `X-Amzn-Trace-Id` per tutte le destinazioni di invocazione. Per ulteriori informazioni sull'intestazione HTTP predefinita, consulta Intestazione di traccia nella Guida per gli sviluppatori di AWS X-Ray.
- Attributo di sistema **`TraceHeader`**: `TraceHeader` è un attributo putEventsRequestEntry riservato da EventBridge per inviare l'intestazione di traccia X-Ray a una destinazione. Se utilizzi anche

`PutEventsRequestEntry`, `PutEventsRequestEntry` sovrascrive l'intestazione di traccia HTTP.

> **Note**
>
> L'intestazione di traccia non viene conteggiato ai fini della dimensione dell'evento `PutEventsRequestEntry`. Per ulteriori informazioni, consulta Calcolo delle dimensioni di iscrizione agli EventBridge PutEvents eventi Amazon.

Il seguente video illustra l'uso congiunto di X-Ray ed EventBridge: Using AWS X-Ray for tracing

Utilizzo EventBridge con AWS Integrated Application Test Kit

Quando crei applicazioni composte da servizi serverless come Lambda EventBridge o Step Functions, molti dei componenti dell'architettura non possono essere distribuiti sul desktop, ma esistono solo nel cloud. AWS A differenza delle applicazioni distribuite localmente, questi tipi di applicazioni traggono vantaggio dalle strategie basate sul cloud per l'esecuzione di test automatici. AWS L'Integrated Application Test Kit (AWS IATK) consente di implementare alcune di queste strategie per le applicazioni.

AWS IATK è una libreria software che consente di scrivere test automatici per applicazioni basate su cloud.

EventBridge integrazione AWS con IATK

Puoi utilizzare EventBridge eventi e bus di eventi con AWS IATK per implementare i tuoi test automatici, tra cui:

Implementazione di test harness

Per scrivere test di integrazione per architetture basate su eventi, stabilisci i limiti logici suddividendo l'applicazione in sottosistemi. Una tecnica utile per testare i sottosistemi consiste nella creazione di test harness, ovvero risorse che crei appositamente per testare i sottosistemi.

Ad esempio, un test di integrazione può avviare un processo di sottosistema passandogli un evento di test di input. AWS IATK può creare per voi un test harness che ascolta gli eventi di

output. EventBridge (Sotto il cofano, l'harness è composto da una EventBridge regola che inoltra l'evento di output ad Amazon SQS.) Il test di integrazione esegue quindi una query sul test harness per esaminare l'output e determinare se l'esito del test è positivo o negativo.

Generazione di eventi fittizi

AWS IATK offre la possibilità di generare eventi fittizi da uno schema memorizzato nel registro degli schemi. EventBridge Ciò consente di generare un evento fittizio e richiamare qualsiasi consumer (come una funzione Lambda o una macchina a stati Step Functions) con l'evento generato.

Per ulteriori informazioni, vedere AWS Integrated Application Test Kit Overview su. GitHub

Inclusione EventBridge delle risorse Amazon negli AWS CloudFormation stack

AWS CloudFormation consente di configurare e gestire AWS le risorse tra account e regioni in modo centralizzato e ripetibile, trattando l'infrastruttura come codice. CloudFormation lo fa consentendoti di creare modelli che definiscono le risorse che desideri fornire e gestire. Queste risorse possono includere EventBridge artefatti come bus e regole degli eventi, pipe, schemi e pianificazioni, tra gli altri. Utilizza queste risorse per includere EventBridge funzionalità negli stack tecnologici tramite i quali esegui il provisioning e la gestione. CloudFormation

EventBridge Risorse Amazon disponibili in AWS CloudFormation

EventBridge fornisce risorse da utilizzare nei CloudFormation modelli nei seguenti namespace di risorse:

- AWS::Events

  Gli esempi di modelli includono:

  - Crea una destinazione API per PagerDuty
  - Creazione di una destinazione API per Slack
  - Crea una connessione con parametri di ApiKey autorizzazione
  - Creazione di una connessione con i parametri di autorizzazione OAuth
  - Creazione di un endpoint globale con la replica degli eventi
  - Policy di rifiuto utilizzando più principali e operazioni

- Concessione di un'autorizzazione a un'organizzazione utilizzando un router di eventi personalizzato
- Creazione di una regola tra Regioni
- Creare una regola che includa una coda DLQ per una destinazione
- Funzione Lambda da richiamare a intervalli regolari
- Richiamare la funzione Lambda in risposta a un evento
- Notifica a un argomento in risposta a una voce di log

- AWS::EventSchemi
- AWS::Pipes

  Gli esempi di modelli includono:

  - Crea una pipe con un filtro per gli eventi
- AWS::Scheduler

Generazione di definizioni di EventBridge risorse Amazon per i AWS CloudFormation modelli

Per aiutarti a iniziare subito a sviluppare CloudFormation modelli, la EventBridge console ti consente di creare CloudFormation modelli a partire dai bus di eventi, dalle regole e dalle pipe esistenti nel tuo account.

- ???
- ???
- ???

Gestione del bus degli eventi predefinito AWS CloudFormation

Poiché esegue automaticamente il EventBridge provisioning del bus degli eventi predefinito nel tuo account, non puoi crearlo utilizzando un CloudFormation modello, come faresti normalmente per qualsiasi risorsa che desideri includere in uno CloudFormation stack. Per includere il bus degli eventi predefinito in uno CloudFormation stack, devi prima importarlo in uno stack. Dopo aver importato il bus degli eventi predefinito in uno stack, potete aggiornare le proprietà del bus degli eventi come desiderate.

Per ulteriori informazioni, consulta ???

Gestione degli eventi AWS CloudFormation dello stack utilizzando EventBridge

Oltre a includere EventBridge risorse negli CloudFormation stack, puoi utilizzarle EventBridge per gestire gli eventi generati dagli CloudFormation stack stessi. CloudFormation invia eventi a EventBridge ogni volta che viene eseguita un'operazione di creazione, aggiornamento, eliminazione o rilevamento della deriva su uno stack. CloudFormation invia anche eventi a EventBridge per modificare lo stato dei set di stack e delle istanze di stack set. È possibile utilizzare EventBridge le regole per indirizzare gli eventi verso obiettivi definiti.

Per ulteriori informazioni, consulta Gestione CloudFormation degli eventi utilizzando EventBridge nella Guida AWS CloudFormation per l'utente.

Tutorial di Amazon EventBridge

EventBridge si integra con vari servizi AWS e partner SaaS. Questi tutorial sono progettati per aiutarti ad acquisire familiarità con le nozioni di base di EventBridge e a comprendere come integrarlo nella tua architettura serverless.

Tutorial:

- Tutorial introduttivi di Amazon EventBridge
- Tutorial di Amazon EventBridge per l'integrazione con altri servizi AWS
- Tutorial di Amazon EventBridge per l'integrazione con provider SaaS

Tutorial introduttivi di Amazon EventBridge

I seguenti tutorial ti aiutano a esplorare le funzionalità di EventBridge e come utilizzarle.

Tutorial:

- Archiviazione e riproduzione di eventi Amazon EventBridge
- Creazione di un'applicazione di esempio Amazon EventBridge
- Tutorial: download di associazioni di codice per eventi utilizzando il registro di schemi EventBridge
- Tutorial: utilizzo del trasformatore di input per personalizzare gli elementi che EventBridge passa alla destinazione di un evento

Archiviazione e riproduzione di eventi Amazon EventBridge

È possibile utilizzare EventBridge per instradare eventi a funzioni AWS Lambda specifiche utilizzando regole.

In questo tutorial, creerai una funzione da utilizzare come destinazione per la regola EventBridge mediante la console Lambda. Quindi, creerai un archivio e una regola per archiviare eventi di test utilizzando la console EventBridge. Una volta che in quell'archivio sono presenti eventi, li riprodurrai.

Passaggi:

- Passaggio 1: creare una funzione Lambda
- Passaggio 2: creare l'archivio
- Passaggio 3: creare una regola
- Passaggio 4: inviare eventi di test
- Passaggio 5: riprodurre gli eventi
- Passaggio 6: eliminare le risorse

Passaggio 1: creare una funzione Lambda

Innanzi tutto, crea una funzione Lambda per registrare gli eventi.

Per creare una funzione Lambda:

1. Apri la console AWS Lambda all'indirizzo https://console.aws.amazon.com/lambda/.
2. Selezionare Create function (Crea funzione).
3. Scegli Author from scratch (Crea da zero).
4. Digitare un nome e una descrizione per la funzione Lambda. Ad esempio, `LogScheduledEvent`.
5. Per le altre opzioni, mantieni il valore predefinito e scegli Crea funzione.
6. Nella scheda Codice della pagina della funzione, fai doppio clic su index.js.
7. Sostituisci il codice JavaScript esistente con il seguente:

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogScheduledEvent');
```

```
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Selezionare Deploy (Distribuisci).

Passaggio 2: creare l'archivio

A questo punto, devi creare l'archivio che conterrà tutti gli eventi di test.

Per creare un archivio

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Archivi.
3. Scegli Crea archivio.
4. Immetti un nome e una descrizione per l'archivio. Ad esempio, assegnagli il nome `ArchiveTest`.
5. Per le altre opzioni, mantieni il valore predefinito e scegli Successivo.
6. Scegli Crea archivio.

Passaggio 3: creare una regola

Crea una regola per archiviare gli eventi inviati al router di eventi.

Per creare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola. Ad esempio, assegnale il nome `ARTestRule`.

   Una regola non può avere lo stesso nome di un'altra regola nella stessa Regione e nello stesso router di eventi.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.

6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "detail-type": [
    "customerCreated"
  ]
}
```

10. Scegli Successivo.
11. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
12. In Seleziona una destinazione, scegli la funzione Lambda dall'elenco a discesa.
13. In Funzione, seleziona la funzione Lambda che hai creato nella sezione Passaggio 1: creare una funzione Lambda. In questo esempio, seleziona `LogScheduledEvent`.
14. Scegli Successivo.
15. Scegli Successivo.
16. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 4: inviare eventi di test

Ora che hai configurato l'archivio e la regola, invieremo eventi di test per assicurarci che l'archivio funzioni correttamente.

> **Note**
>
> È possibile che gli eventi non siano immediatamente disponibili nell'archivio.

Per inviare eventi di test (console)

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Nel riquadro Bus di eventi predefinito, scegli Azioni, Invia eventi.

4. Immetti un'origine per gli eventi. Ad esempio, `TestEvent`.
5. In Tipo di dettaglio, immetti `customerCreated`.
6. In dettagli dell'evento, immetti `{}`.
7. Scegli Invia.

Passaggio 5: riprodurre gli eventi

Una volta che gli eventi di test sono nell'archivio, puoi riprodurli.

Per riprodurre gli eventi archiviati (console)

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Riproduzioni.
3. Scegli Avvia nuova riproduzione.
4. Immetti un nome e una descrizione per la riproduzione. Ad esempio, assegnale il nome `ReplayTest`.
5. In Origine, seleziona l'archivio che hai creato nella sezione Passaggio 2: creare l'archivio.
6. In intervallo di tempo della riproduzione, procedi come segue.
    a. In Ora di inizio, seleziona la data in cui hai inviato gli eventi di test e un'ora prima dell'invio. Ad esempio `2021/08/11` e `08:00:00`.
    b. In Ora di fine, seleziona la data e l'ora correnti. Ad esempio `2021/08/11` e `09:15:00`.
7. Scegli Avvia la riproduzione.

Passaggio 6: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare la funzione Lambda

1. Aprire la pagina Funzioni della console Lambda.
2. Seleziona la funzione creata.
3. Scegliere Actions (Operazioni), Delete (Elimina).
4. Scegli Elimina.

Per eliminare l'archivio EventBridge

1. Apri la pagina Archivi nella console EventBridge.
2. Seleziona l'archivio creato.
3. Scegli Elimina.
4. Immetti il nome dell'archivio e scegli Elimina.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Creazione di un'applicazione di esempio Amazon EventBridge

È possibile utilizzare EventBridge per instradare eventi a funzioni Lambda specifiche con regole.

In questo tutorial, utilizzerai l'AWS CLI, Node.js e il codice nel repository GitHub per creare quanto segue:

- Una funzione AWS Lambda che produce eventi per transazioni bancarie ATM (Bancomat).
- Tre funzioni Lambda da utilizzare come destinazioni di una regola EventBridge.
- La regola che instrada gli eventi creati alla funzione a valle corretta in base a un modello di eventi.

Questo esempio utilizza modelli AWS SAM per definire le regole EventBridge. Per ulteriori informazioni sull'utilizzo di modelli AWS SAM con EventBridge, consulta ???.

Nel repository, la sottodirectory atmProducer contiene `handler.js`, che rappresenta il servizio ATM che produce eventi. Questo codice è un gestore Lambda scritto in Node.js che pubblica eventi su EventBridge tramite l'AWS SDK utilizzando questa riga di codice JavaScript.

```
const result = await eventbridge.putEvents(params).promise()
```

Questa directory contiene anche `events.js`, che elenca varie transazioni di test in un array Entries. Un singolo evento è definito in JavaScript come segue:

```
{
  // Event envelope fields
  Source: 'custom.myATMapp',
  EventBusName: 'default',
  DetailType: 'transaction',
  Time: new Date(),

  // Main event body
  Detail: JSON.stringify({
    action: 'withdrawal',
    location: 'MA-BOS-01',
    amount: 300,
    result: 'approved',
    transactionId: '123456',
    cardPresent: true,
    partnerBank: 'Example Bank',
    remainingFunds: 722.34
```

```
  })
}
```

La sezione Dettaglio dell'evento specifica gli attributi della transazione. Questi includono l'ubicazione dello sportello ATM, l'importo, la banca partner e il risultato della transazione.

Il file `handler.js` nella sottodirectory atmConsumer contiene tre funzioni:

```
exports.case1Handler = async (event) => {
  console.log('--- Approved transactions ---')
  console.log(JSON.stringify(event, null, 2))
}

exports.case2Handler = async (event) => {
  console.log('--- NY location transactions ---')
  console.log(JSON.stringify(event, null, 2))
}

exports.case3Handler = async (event) => {
  console.log('--- Unapproved transactions ---')
  console.log(JSON.stringify(event, null, 2))
}
```

Ogni funzione riceve eventi di transazione, che vengono registrati tramite le istruzioni `console.log` in File di log Amazon CloudWatch. Le funzioni consumer operano indipendentemente dal produttore e non conoscono l'origine degli eventi.

La logica di routing è contenuta nelle regole EventBridge distribuite dal modello AWS SAM dell'applicazione. Le regole valutano il flusso di eventi in entrata e instradano gli eventi corrispondenti alle funzioni Lambda di destinazione.

Le regole utilizzano modelli di eventi che sono oggetti JSON con la stessa struttura degli eventi a cui corrispondono. Di seguito è riportato il modello di eventi per una delle regole.

```
{
  "detail-type": ["transaction"],
  "source": ["custom.myATMapp"],
  "detail": {
    "location": [{
      "prefix": "NY-"
    }]
```

```
  }
}
```

Passaggi:

- Prerequisiti
- Passaggio 1: creare un'applicazione
- Passaggio 2: eseguire l'applicazione
- Passaggio 3: verificare i log e il funzionamento dell'applicazione
- Passaggio 4: eliminare le risorse

Prerequisiti

Per completare questo tutorial, avrai bisogno delle seguenti risorse:

- Un account AWS. Crea un account AWS se non ne hai già uno.
- AWS CLI installata. Per installare l'AWS CLI, consulta Installazione, aggiornamento e disinstallazione dell'AWS CLI versione 2.
- Node.js 12.x installato. Per installare Node.js, consulta Download.

Passaggio 1: creare un'applicazione

Per configurare l'applicazione di esempio, utilizzerai l'AWS CLI e Git per creare le risorse AWS di cui avrai bisogno.

Per creare l'applicazione

1. Esegui l'accesso a AWS.
2. Installa Git e installa l'AWS Serverless Application Model CLI sul tuo computer locale.
3. Crea una nuova directory, quindi accedi a quella directory in un terminale.
4. Alla riga di comando, immetti `git clone https://github.com/aws-samples/amazon-eventbridge-producer-consumer-example`.
5. Alla riga di comando esegui il comando seguente:

```
cd ./amazon-eventbridge-producer-consumer-example
sam deploy --guided
```

6. Nel terminale, procedi come segue:

   a. In **`Stack Name`**, immetti un nome per lo stack. Ad esempio, assegnagli il nome `Test`.
   b. In **`AWS Region`**, immetti la Regione. Ad esempio, `us-west-2`.
   c. In **`Confirm changes before deploy`**, immetti `Y`.
   d. In **`Allow SAM CLI IAM role creation`**, immetti `Y`.
   e. In **`Save arguments to configuration file`**, immetti `Y`.
   f. In **`SAM configuration file`**, immetti `samconfig.toml`.
   g. In **`SAM configuration environment`**, immetti `default`.

Passaggio 2: eseguire l'applicazione

Ora che hai configurato le risorse, utilizzerai la console per testare le funzioni.

Per eseguire l'applicazione

1. Apri la console Lambda nella stessa Regione in cui hai distribuito l'applicazione AWS SAM.
2. Esistono quattro funzioni Lambda con il prefisso atm-demo. Seleziona la funzione atmProducerFn, quindi scegli Azioni, Esegui test.
3. In Nome, immetti `Test`.
4. Scegli Test (Esegui test).

Passaggio 3: verificare i log e il funzionamento dell'applicazione

Ora che hai eseguito l'applicazione, utilizzerai la console per controllare File di log CloudWatch.

Per verificare i log

1. Apri la console CloudWatch nella stessa Regione in cui hai eseguito l'applicazione AWS SAM.
2. Scegli Log e quindi Gruppi di log.
3. Seleziona il gruppo di log contenente atmConsumerCase1. Vengono visualizzati due flussi che rappresentano le due transazioni approvate dall'ATM. Scegli un flusso di log per visualizzare l'output.
4. Torna all'elenco dei gruppi di log, quindi seleziona il gruppo di log contenente atmConsumerCase2. Vedrai due stream che rappresentano le due transazioni corrispondenti al filtro di ubicazione New York.

5. Torna all'elenco dei gruppi di log, quindi seleziona il gruppo di log contenente atmConsumerCase2. Apri il flusso per vedere le transazioni negate.

Passaggio 4: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Per eliminare la funzione Lambda

1. Aprire la pagina Funzioni della console Lambda.
2. Seleziona la funzione creata.
3. Scegliere Actions (Operazioni), Delete (Elimina).
4. Scegli Elimina.

Per eliminare il gruppo di log di File di log CloudWatch

1. Apri la console CloudWatch.
2. Scegli Log, Gruppi di log.
3. Seleziona il gruppo di log.
4. Scegli Actions (Azioni), Delete log group(s) (Elimina gruppo/i di log).
5. Scegli Elimina.

Tutorial: download di associazioni di codice per eventi utilizzando il registro di schemi EventBridge

È possibile generare associazioni di codice per schemi di eventi per accelerare lo sviluppo in Golang, Java, Python e TypeScript. Puoi ottenere associazioni di codice per servizi AWS esistenti, schemi creati e per schemi generati in base a eventi in un router di eventi. Puoi generare associazioni di codice per uno schema mediante uno dei seguenti elementi:

- Console EventBridge
- API registro di schemi EventBridge
- Il tuo ambiente di sviluppo integrato (IDE) con un kit di strumenti AWS

In questo tutorial verranno generate e scaricate associazioni di codice da uno schema EventBridge per gli eventi di un servizio AWS.

Per generare associazioni di codice da uno schema EventBridge

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, seleziona Schemas (Schemi).
3. Seleziona la scheda Registro schemi di eventi AWS.
4. Trova lo schema per il servizio AWS per il quale vuoi scaricare associazioni di codice, esplorando il registro di schemi o cercando uno schema.
5. Seleziona il nome dello schema.
6. Nella pagina dei dettagli dello schema, nella sezione Versione, seleziona Scarica le associazioni del codice.
7. Nella pagina Download code bindings (Scarica associazioni di codice) selezionare la lingua delle associazioni di codice che si desidera scaricare.
8. Selezionare Download (Scarica).

   Potrebbero essere necessari alcuni secondi per l'avvio del download. Il file di download sarà un file .zip di associazioni di codice per la lingua selezionata.
9. Decomprimi il file scaricato e aggiungilo al progetto.

   Il pacchetto scaricato contiene un file LEGGIMI che spiega come configurare le dipendenze del pacchetto in vari framework.

Utilizza queste associazioni di codice nel tuo codice per creare rapidamente applicazioni utilizzando questo evento EventBridge.

Tutorial: utilizzo del trasformatore di input per personalizzare gli elementi che EventBridge passa alla destinazione di un evento

Puoi utilizzare il trasformatore di input in EventBridge per personalizzare il testo di un evento prima di inviarlo alla destinazione di una regola.

A questo proposito, definisci percorsi JSON dall'evento e assegni i relativi output a variabili diverse. Puoi quindi utilizzare quelle variabili nel modello di input. I caratteri < and > non possono avere caratteri di escape. Per ulteriori informazioni, consulta Trasformazione degli EventBridge input di Amazon

> **Note**
>
> Se specifichi una variabile per abbinare un percorso JSON che non esiste nell'evento, quella variabile non viene creata e non appare nell'output.

In questo tutorial, crei una regola che corrisponde a un evento con `detail-type: "customerCreated"`. Il trasformatore di input mappa la variabile `type` al percorso JSON $.detail-type dall'evento. Quindi EventBridge inserisce la variabile nel modello di input "This event was <tipo>". Il risultato è il seguente messaggio di Amazon SNS.

```
"This event was of customerCreated type."
```

Passaggi:

- Passaggio 1: creare un argomento Amazon SNS
- Passaggio 2: creare una sottoscrizione Amazon SNS
- Passaggio 3: creare una regola
- Passaggio 4: inviare eventi di test
- Passaggio 5: verificare il corretto completamento del tutorial
- Passaggio 6: eliminare le risorse

Passaggio 1: creare un argomento Amazon SNS

Crea un argomento per ricevere gli eventi da EventBridge.

Per creare un argomento

1. Apri la console Amazon SNS all'indirizzo https://console.aws.amazon.com/sns/v3/home.
2. Nel pannello di navigazione, scegli Topics (Argomenti).
3. Scegli Create topic (Crea argomento).
4. Per Tipo, scegliere Standard.
5. Immetti **`eventbridge-IT-test`** come nome dell'argomento.
6. Scegli Create topic (Crea argomento).

Passaggio 2: creare una sottoscrizione Amazon SNS

Creazione di una sottoscrizione per ricevere e-mail con le informazioni trasformate.

Creazione di una sottoscrizione

1. Apri la console Amazon SNS all'indirizzo https://console.aws.amazon.com/sns/v3/home.
2. Nel riquadro di navigazione scegliere Subscriptions (Iscrizioni).
3. Scegliere Create Subscription (Crea iscrizione).
4. In ARN argomento, scegli l'argomento creato in Passaggio 1. Per questo tutorial, scegli eventbridge-IT-test.
5. Per Protocol, scegli Email.
6. Per Endpoint, immettere il proprio indirizzo e-mail.
7. Scegliere Create Subscription (Crea iscrizione).
8. Conferma la sottoscrizione scegliendo Conferma sottoscrizione nell'e-mail che ricevi dalle notifiche AWS.

Passaggio 3: creare una regola

Crea una regola per utilizzare il trasformatore di input per personalizzare le informazioni sullo stato dell'istanza inviate a una destinazione.

Per creare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).

3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola. Ad esempio, assegnale il nome `ARTestRule`.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "detail-type": [
    "customerCreated"
  ]
}
```

10. Scegli Successivo.
11. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
12. In Seleziona una destinazione, scegli l'argomento SNS dall'elenco a discesa.
13. In Argomento, seleziona l'argomento Amazon SNS che hai creato in Passaggio 1. Per questo tutorial, scegli eventbridge-IT-test.
14. In Impostazioni aggiuntive, procedi come segue:
    a. In Configura l'input di destinazione, scegli Trasformatore di input dall'elenco a discesa.
    b. Scegli Configura il trasformatore di input.
    c. In Eventi di esempio, immetti quanto segue:

```
{
  "detail-type": "customerCreated"
}
```

    d. In Trasformatore di input di destinazione, procedi come segue:
        i. In Percorso di input, immetti quanto segue:

```
{"detail-type":"$.detail-type"}
```

ii. In Modello di input, immetti quanto segue:

```
"This event was of <detail-type> type."
```

e. Scegli Conferma.

15. Scegli Successivo.
16. Scegli Successivo.
17. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 4: inviare eventi di test

Ora che hai impostato l'argomento SNS e la regola, invieremo eventi di test per assicurarci che la regola funzioni correttamente.

Per inviare eventi di test (console)

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegliere Event history (Cronologia eventi).
3. Nel riquadro Bus di eventi predefinito, scegli Azioni, Invia eventi.
4. Immetti un'origine per gli eventi. Ad esempio, `TestEvent`.
5. In Tipo di dettaglio, immetti `customerCreated`.
6. In dettagli dell'evento, immetti `{}`.
7. Scegli Invia.

Passaggio 5: verificare il corretto completamento del tutorial

Se ricevi un'e-mail dalle notifiche AWS che corrisponde all'output previsto, hai completato correttamente il tutorial.

Passaggio 6: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare l'argomento SNS

1. Apri la pagina Argomenti nella console SNS.
2. Seleziona l'argomento creato.
3. Scegli Elimina.
4. Specificare **`delete me`**.
5. Scegli Elimina.

Per eliminare la sottoscrizione SNS

1. Apri la pagina Sottoscrizioni della console SNS.
2. Seleziona la sottoscrizione creata.
3. Scegli Elimina.
4. Scegli Elimina.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Tutorial di Amazon EventBridge per l'integrazione con altri servizi AWS

Amazon EventBridge interagisce con altri servizi AWS per elaborare eventi o richiamare una risorsa AWS come destinazione di una regola. I tutorial seguenti mostrano come integrare EventBridge con altri servizi AWS.

Tutorial:

- Tutorial: registrazione di un gruppo con dimensionamento automatico tramite EventBridge
- Tutorial: Registra le chiamate AWS API utilizzando EventBridge
- Tutorial: registra lo stato di un'istanza Amazon EC2 utilizzando EventBridge
- Tutorial: registrazione di operazioni a livello di oggetto di Amazon S3 mediante EventBridge
- Tutorial: invio di eventi a uno stream Amazon Kinesis utilizzando EventBridge e lo schema aws.events
- Tutorial: pianificazione di snapshot Amazon EBS automatizzati mediante EventBridge
- Tutorial: invio di una notifica quando viene creato un oggetto Amazon S3
- Tutorial: pianificazione delle funzioni AWS Lambda mediante EventBridge

Tutorial: registrazione di un gruppo con dimensionamento automatico tramite EventBridge

Puoi eseguire una funzione AWS Lambda che registra eventi ogni volta che un gruppo con dimensionamento automatico avvia o termina un'istanza Amazon EC2 che indica se l'evento è stato eseguito correttamente.

Per informazioni su altri scenari che utilizzano gli eventi di Dimensionamento automatico Amazon EC2, consulta Utilizzo di EventBridge per gestire eventi di Dimensionamento automatico nella Guida per l'utente di Dimensionamento automatico Amazon EC2.

In questo tutorial, crei una funzione Lambda e una regola nella console EventBridge che richiama tale funzione quando un gruppo con dimensionamento automatico Amazon EC2 avvia o termina un'istanza.

Passaggi:

- Prerequisiti
- Passaggio 1: creare una funzione Lambda
- Passaggio 2: creare una regola
- Passaggio 3: testare la regola
- Passaggio 4: verificare il corretto completamento del tutorial
- Passaggio 5: eliminare le risorse

Prerequisiti

Per completare questo tutorial, avrai bisogno delle seguenti risorse:

- Un gruppo con dimensionamento automatico. Per ulteriori informazioni sulla creazione di un gruppo con dimensionamento automatico, consulta Creazione di un gruppo con dimensionamento automatico utilizzando una configurazione di avvio nella Guida per l'utente di Dimensionamento automatico Amazon EC2.

Passaggio 1: creare una funzione Lambda

Crea una funzione Lambda per la registrazione degli eventi di dimensionamento orizzontale e verticale per il gruppo Auto Scaling.

Per creare una funzione Lambda

1. Apri la console AWS Lambda all'indirizzo https://console.aws.amazon.com/lambda/.
2. Selezionare Create function (Crea funzione).
3. Scegli Author from scratch (Crea da zero).
4. Immetti un nome per la funzione Lambda. Ad esempio, `LogAutoScalingEvent`.
5. Per le altre opzioni, mantieni il valore predefinito e scegli Crea funzione.
6. Nella scheda Codice della pagina della funzione, fai doppio clic su index.js.
7. Sostituisci il codice esistente con il seguente.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogAutoScalingEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Selezionare Deploy (Distribuisci).

Passaggio 2: creare una regola

Crea una regola per eseguire la funzione Lambda creata nella sezione Passaggio 1. La regola viene eseguita quando il gruppo con dimensionamento automatico avvia o arresta un'istanza.

Per creare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola. Ad esempio, assegnale il nome `TestRule`.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).

7. Scegli Successivo.
8. Per Event source (Origine evento), scegli AWS services (Servizi).
9. Per Event pattern (Modello di eventi), procedi come segue:
    a. In Origine evento, seleziona Auto Scaling dall'elenco a discesa.
    b. In Tipo di evento, seleziona Avvia e termina istanza dall'elenco a discesa.
    c. Scegli Qualsiasi evento relativo all'istanza e Qualsiasi nome di gruppo.
10. Scegli Successivo.
11. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
12. In Seleziona una destinazione, scegli la funzione Lambda dall'elenco a discesa.
13. In Funzione, seleziona la funzione Lambda che hai creato nella sezione Passaggio 1: creare una funzione Lambda. In questo esempio, seleziona `LogAutoScalingEvent`.
14. Scegli Successivo.
15. Scegli Successivo.
16. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 3: testare la regola

Puoi testare la regola dimensionando manualmente un gruppo con dimensionamento automatico in modo che avvii un'istanza. Attendi alcuni minuti per l'evento di scalabilità orizzontale, quindi verifica che la funzione Lambda sia stata richiamata.

Per testare la regola tramite un gruppo Auto Scaling

1. Per aumentare le dimensioni del gruppo con dimensionamento automatico, procedi come segue:
    a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
    b. Nel riquadro di navigazione, selezionare Auto Scaling, Auto Scaling Groups (Gruppi Auto Scaling).
    c. Seleziona la casella di controllo accanto al gruppo Auto Scaling.
    d. Nella scheda Details (Dettagli), seleziona Edit (Modifica). In Desired (Desiderato), aumenta la capacità desiderata di una unità. Ad esempio, se il valore corrente è 2, immetti 3. La capacità desiderata deve essere minore o uguale alla dimensione massima del gruppo. Se il nuovo valore di Desired (Desiderato) è superiore a Max, devi aggiornare Max. Al termine, selezionare Save (Salva).

2. Per visualizzare l'output della funzione Lambda, procedi nel seguente modo:

   a. Aprire la console CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/.

   b. Nel riquadro di navigazione scegli Logs (Log).

   c. Seleziona il nome del gruppo di log per la funzione Lambda (`/aws/lambda/function-name`).

   d. Seleziona il nome del flusso di log per visualizzare i dati forniti dalla funzione per l'istanza avviata.

3. (Facoltativo) Al termine, puoi diminuire la capacità desiderata di una unità, in modo che il gruppo con dimensionamento automatico torni alle dimensioni precedenti.

Passaggio 4: verificare il corretto completamento del tutorial

Se vedi l'evento Lambda nei log CloudWatch, significa che hai completato correttamente questo tutorial. Se l'evento non è presente nei log CloudWatch, inizia la risoluzione dei problemi verificando che la regola sia stata creata correttamente e, se è il caso, verifica che il codice della funzione Lambda sia corretto.

Passaggio 5: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Per eliminare la funzione Lambda

1. Aprire la pagina Funzioni della console Lambda.
2. Seleziona la funzione creata.
3. Scegliere Actions (Operazioni), Delete (Elimina).
4. Scegli Elimina.

Tutorial: Registra le chiamate AWS API utilizzando EventBridge

Puoi utilizzare EventBridge le regole di Amazon per reagire alle chiamate API effettuate da un AWS servizio registrato da AWS CloudTrail.

In questo tutorial, crei un AWS CloudTrailtrail, una funzione Lambda e una regola nella EventBridge console. La regola richiama la funzione Lambda quando un'istanza Amazon EC2 viene interrotta.

Fasi:

- Fase 1: Creare un AWS CloudTrail percorso
- Passaggio 2: creare una funzione AWS Lambda
- Passaggio 3: creare una regola
- Passaggio 4: testare la regola
- Passaggio 5: verificare il corretto completamento del tutorial
- Fase 6: eliminare le risorse

Fase 1: Creare un AWS CloudTrail percorso

Se un trail è già configurato, vai al passaggio 2.

Per creare un trail

1. Apri la CloudTrail console all'indirizzo https://console.aws.amazon.com/cloudtrail/.
2. Scegliere Trails (Trail), Create trail (Crea trail).
3. In Trail name (Nome trail), digita un nome per il trail.
4. In Posizione archiviazione, in Crea un nuovo bucket S3, scegli Sì.
5. In Alias AWS KMS , digita un alias per la chiave KMS.
6. Seleziona Successivo.
7. Seleziona Successivo.
8. Scegliere Create trail (Creare trail).

Passaggio 2: creare una funzione AWS Lambda

Crea una funzione Lambda per registrare gli eventi di chiamate API.

Per creare una funzione Lambda

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Selezionare Create function (Crea funzione).
3. Scegli Author from scratch (Crea da zero).
4. Digitare un nome e una descrizione per la funzione Lambda. Ad esempio, denomina la funzione `LogEC2StopInstance`.
5. Per le altre opzioni, mantieni il valore predefinito e scegli Crea funzione.
6. Nella scheda Codice della pagina della funzione, fai doppio clic su index.js.
7. Sostituisci il codice esistente con il seguente.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2StopInstance');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Seleziona Deploy (Implementa).

Passaggio 3: creare una regola

Crea una regola per eseguire la funzione Lambda creata nel passaggio 2 ogni volta che arresti un'istanza Amazon EC2.

Per creare una regola

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Immettere un nome e una descrizione per la regola. Ad esempio, assegnale il nome `TestRule`.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.

6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Seleziona Successivo.
8. Per Origine evento, scegli Servizi AWS .
9. Per Event pattern (Modello di eventi), procedi come segue:
    a. In Origine evento, seleziona EC2 dall'elenco a discesa.
    b. Per Tipo di evento, seleziona AWS API Call via CloudTrail dall'elenco a discesa.
    c. Scegli Operazioni specifiche e immetti `StopInstances`.
10. Seleziona Successivo.
11. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
12. In Seleziona una destinazione, scegli la funzione Lambda dall'elenco a discesa.
13. In Funzione, seleziona la funzione Lambda che hai creato nella sezione Passaggio 1: creare una funzione Lambda. In questo esempio, seleziona `LogEC2StopInstance`.
14. Seleziona Successivo.
15. Seleziona Successivo.
16. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 4: testare la regola

Puoi testare la regola arrestando un'istanza Amazon EC2 tramite la console Amazon EC2. Attendi qualche minuto che l'istanza si interrompa, quindi controlla le AWS Lambda metriche sulla CloudWatch console per verificare che la funzione funzioni.

Test della regola arrestando un'istanza

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Avvia un'istanza. Per ulteriori informazioni, consulta Launch Your Instance nella Amazon EC2 User Guide.
3. Arrestare l'istanza. Per ulteriori informazioni, consulta Stop and Start Your Instance nella Amazon EC2 User Guide.
4. Per visualizzare l'output della funzione Lambda, procedi nel seguente modo:
    a. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
    b. Nel riquadro di navigazione scegli Logs (Log).

c. Seleziona il nome del gruppo di log per la funzione Lambda (`/aws/lambda/function-name`).

d. Selezionare il nome del flusso di log per visualizzare i dati forniti dalla funzione per l'istanza arrestata.

5. (Facoltativo) Al termine, terminare l'istanza arrestata. Per ulteriori informazioni, consulta Terminate Your Instance nella Amazon EC2 User Guide.

Passaggio 5: verificare il corretto completamento del tutorial

Se vedi l'evento Lambda nei CloudWatch log, significa che hai completato con successo questo tutorial. Se l'evento non è presente nei tuoi CloudWatch registri, inizia la risoluzione dei problemi verificando che la regola sia stata creata correttamente e, se la regola sembra corretta, verifica che il codice della tua funzione Lambda sia corretto.

Fase 6: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia conservarle. Eliminando AWS le risorse che non utilizzi più, eviti addebiti inutili sul tuo account. AWS

Per eliminare le EventBridge regole

1. Apri la pagina Regole della EventBridge console.
2. Seleziona la regola che hai creato.
3. Scegliere Delete (Elimina).
4. Scegliere Delete (Elimina).

Per eliminare la funzione Lambda

1. Aprire la pagina Funzioni della console Lambda.
2. Seleziona la funzione creata.
3. Scegliere Actions (Operazioni), Delete (Elimina).
4. Scegliere Delete (Elimina).

Per eliminare i CloudTrail percorsi

1. Apri la pagina Trails della CloudTrail console.

2. Seleziona il trail creato.
3. Scegliere Delete (Elimina).
4. Scegli Delete (Elimina).

Tutorial: registra lo stato di un'istanza Amazon EC2 utilizzando EventBridge

È possibile creare una funzione AWS Lambda che registri le modifiche dello stato per un'istanza Amazon EC2. Successivamente, puoi scegliere di creare una regola che esegua la funzione Lambda ogni volta che si verifica una transizione di stato o una transizione a uno o più stati di interesse. In questo tutorial, registrerai l'avvio di qualsiasi nuova istanza.

Fasi:

- Passaggio 1: creare una funzione AWS Lambda
- Fase 2: Creazione di una regola
- Fase 3: Test della regola
- Passaggio 4: verificare il corretto completamento del tutorial
- Passaggio 5: eliminare le risorse

Passaggio 1: creare una funzione AWS Lambda

Crea una funzione Lambda per registrare gli eventi di modifica dello stato. Quando crei la regola nella sezione Passaggio 2, specifichi questa funzione.

Per creare una funzione Lambda

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Selezionare Create function (Crea funzione).
3. Scegli Author from scratch (Crea da zero).
4. Digitare un nome e una descrizione per la funzione Lambda. Ad esempio, denomina la funzione `LogEC2InstanceStateChange`.
5. Per le altre opzioni, mantieni il valore predefinito e scegli Crea funzione.
6. Nella scheda Codice della pagina della funzione, fai doppio clic su index.js.
7. Sostituisci il codice esistente con il seguente.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2InstanceStateChange');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
```

```
};
```

8. Selezionare Deploy (Distribuisci).

Fase 2: Creazione di una regola

Crea una regola per eseguire la funzione Lambda creata nella sezione Passaggio 1. La regola viene eseguita quando avvii un'istanza Amazon EC2.

Per creare la EventBridge regola

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Immettere un nome e una descrizione per la regola. Ad esempio, assegnale il nome `TestRule`.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Seleziona Successivo.
8. Per Origine evento, scegli Servizi AWS .
9. Per Event pattern (Modello di eventi), procedi come segue:
   a. In Origine evento, seleziona EC2 dall'elenco a discesa.
   b. In Tipo di evento, scegli Notifica del cambio di stato istanza EC2 dall'elenco a discesa.
   c. Scegli Stati specifici e scegli In esecuzione dall'elenco a discesa.
   d. Scegli Qualsiasi istanza.
10. Seleziona Successivo.
11. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
12. In Seleziona una destinazione, scegli la funzione Lambda dall'elenco a discesa.
13. In Funzione, seleziona la funzione Lambda che hai creato nella sezione Passaggio 1: creare una funzione Lambda. In questo esempio, seleziona `LogEC2InstanceStateChange`.
14. Seleziona Successivo.

15. Seleziona Successivo.
16. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Fase 3: Test della regola

Puoi testare la regola arrestando un'istanza Amazon EC2 tramite la console Amazon EC2. Attendi qualche minuto che l'istanza si fermi, quindi controlla le AWS Lambda metriche sulla CloudWatch console per verificare che la funzione funzioni.

Test della regola arrestando un'istanza

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Avvia un'istanza. Per ulteriori informazioni, consulta Launch Your Instance nella Amazon EC2 User Guide.
3. Arrestare l'istanza. Per ulteriori informazioni, consulta Stop and Start Your Instance nella Amazon EC2 User Guide.
4. Per visualizzare l'output della funzione Lambda, procedi nel seguente modo:
    a. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
    b. Nel riquadro di navigazione scegli Logs (Log).
    c. Seleziona il nome del gruppo di log per la funzione Lambda (`/aws/lambda/`*`function-name`*).
    d. Selezionare il nome del flusso di log per visualizzare i dati forniti dalla funzione per l'istanza arrestata.
5. (Facoltativo) Al termine, terminare l'istanza arrestata. Per ulteriori informazioni, consulta Terminate Your Instance nella Amazon EC2 User Guide.

Passaggio 4: verificare il corretto completamento del tutorial

Se vedi l'evento Lambda nei CloudWatch log, significa che hai completato con successo questo tutorial. Se l'evento non è presente nei tuoi CloudWatch registri, inizia la risoluzione dei problemi verificando che la regola sia stata creata correttamente e, se la regola sembra corretta, verifica che il codice della tua funzione Lambda sia corretto.

Passaggio 5: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia conservarle. Eliminando AWS le risorse che non utilizzi più, eviti addebiti inutili sul tuo account. AWS

Per eliminare la/le EventBridge regola/e

1. Apri la pagina Regole della EventBridge console.
2. Seleziona la regola che hai creato.
3. Scegliere Delete (Elimina).
4. Scegliere Delete (Elimina).

Per eliminare la funzione Lambda

1. Aprire la pagina Funzioni della console Lambda.
2. Seleziona la funzione creata.
3. Scegliere Actions (Operazioni), Delete (Elimina).
4. Scegli Delete (Elimina).

Tutorial: registrazione di operazioni a livello di oggetto di Amazon S3 mediante EventBridge

Puoi registrare le operazioni API a livello di oggetto nei tuoi bucket Amazon S3. Affinché Amazon EventBridge possa trovare una corrispondeza con questi eventi, devi usare AWS CloudTrail per impostare e configurare un trail per la ricezione di questi eventi.

In questo tutorial, crei un trail CloudTrail, una funzione AWS Lambda e quindi una regola nella console EventBridge che richiama quella funzione in risposta a un evento di dati S3.

Passaggi:

- Passaggio 1: configurare il trail AWS CloudTrail
- Passaggio 2: creare una funzione AWS Lambda
- Passaggio 3: creare una regola
- Fase 4: test della regola
- Passaggio 5: verificare il corretto completamento del tutorial
- Passaggio 6: eliminare le risorse

Passaggio 1: configurare il trail AWS CloudTrail

Per registrare eventi di dati per un bucket S3 in AWS CloudTrail e EventBridge, devi dapprima creare un trail. Un trail acquisisce chiamate API e i relativi eventi nel tuo account e quindi distribuisce i file di log a un bucket S3 specificato. Puoi aggiornare un trail esistente oppure crearne uno.

Per ulteriori informazioni, consulta Eventi di dati nella Guida per l'utente di AWS CloudTrail.

Per creare un trail

1. Apri la console CloudTrail all'indirizzo https://console.aws.amazon.com/cloudtrail/.
2. Scegliere Trails (Trail), Create trail (Crea trail).
3. In Trail name (Nome trail), digita un nome per il trail.
4. In Posizione archiviazione, in Crea un nuovo bucket S3, scegli Sì.
5. In Alias AWS KMS, digita un alias per la chiave KMS.
6. Scegli Successivo.

7. In Tipo di evento, scegli Eventi di dati.
8. In Eventi di dati, esegui una delle operazioni descritte di seguito:
   - Per registrare gli eventi di dati per tutti gli oggetti Amazon S3 in un bucket, specifica un S3 Bucket e un prefisso vuoto. Quando si verifica un evento in un oggetto incluso in tale bucket, il trail elabora e registra l'evento.
   - Per registrare eventi di dati per oggetti Amazon S3 specifici, specifica un bucket S3 e il prefisso dell'oggetto. Quando si verifica un evento in un oggetto incluso in tale bucket e l'oggetto inizia con il prefisso specificato, il trail elabora e registra l'evento.
9. Per ciascuna risorsa, scegli se registrare gli eventi Lettura, Scrittura o entrambi.
10. Scegli Successivo.
11. Scegliere Create trail (Creare trail).

Passaggio 2: creare una funzione AWS Lambda

Crea una funzione Lambda per la registrazione di eventi di dati per gli S3 Bucket.

Per creare una funzione Lambda

1. Apri la console AWS Lambda all'indirizzo https://console.aws.amazon.com/lambda/.
2. Selezionare Create function (Crea funzione).
3. Scegli Author from scratch (Crea da zero).
4. Digitare un nome e una descrizione per la funzione Lambda. Ad esempio, `LogS3DataEvents`.
5. Per le altre opzioni, mantieni il valore predefinito e scegli Crea funzione.
6. Nella scheda Codice della pagina della funzione, fai doppio clic su index.js.
7. Sostituisci il codice esistente con il seguente.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogS3DataEvents');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Selezionare Deploy (Distribuisci).

Passaggio 3: creare una regola

Crea una regola per eseguire la funzione Lambda creata nella sezione Passaggio 2. Questa regola viene eseguita in risposta a un evento di dati Amazon S3.

Per creare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola. Ad esempio, assegnale il nome `TestRule`.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Successivo.
8. Per Event source (Origine evento), scegli AWS services (Servizi).
9. Per Event pattern (Modello di eventi), procedi come segue:
   a. In Origine evento, seleziona Simple Storage Service (S3) dall'elenco a discesa.
   b. In Tipo di evento, seleziona Chiamata API a livello di oggetto tramite CloudTrail dall'elenco a discesa.
   c. Scegliere Specific operation(s) (Operazioni specifiche), quindi PutObject.
   d. Per impostazione predefinita, la regola abbina gli eventi di dati per tutti i bucket nella Regione. Per abbinare eventi di dati per bucket specifici, selezionare Specify bucket(s) by name (Specifica bucket per nome), quindi specificare uno o più bucket.
10. Scegli Successivo.
11. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
12. In Seleziona una destinazione, scegli la funzione Lambda dall'elenco a discesa.
13. In Funzione, seleziona la funzione Lambda `LogS3DataEvents` che hai creato in Passaggio 1.
14. Scegli Successivo.
15. Scegli Successivo.

16. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Fase 4: test della regola

Per testare la regola, inserisci un oggetto nel bucket S3. Puoi verificare che la funzione Lambda sia stata invocata.

Per visualizzare i registri della funzione Lambda

1. Aprire la console CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegli Logs (Log).
3. Seleziona il nome del gruppo di log per la funzione Lambda (`/aws/lambda/`*`function-name`*).
4. Seleziona il nome del flusso di log per visualizzare i dati forniti dalla funzione per l'istanza avviata.

Puoi anche controllare i log CloudTrail nel bucket S3 specificato per il trail. Per ulteriori informazioni, consulta Recupero e visualizzazione dei file di log CloudTrail nella Guida per l'utente di AWS CloudTrail.

Passaggio 5: verificare il corretto completamento del tutorial

Se vedi l'evento Lambda nei log CloudWatch, significa che hai completato correttamente questo tutorial. Se l'evento non è presente nei log CloudWatch, inizia la risoluzione dei problemi verificando che la regola sia stata creata correttamente e, se è il caso, verifica che il codice della funzione Lambda sia corretto.

Passaggio 6: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Per eliminare la funzione Lambda

1. Aprire la pagina Funzioni della console Lambda.
2. Seleziona la funzione creata.
3. Scegliere Actions (Operazioni), Delete (Elimina).
4. Scegli Elimina.

Per eliminare il trail CloudTrail

1. Apri la pagina Trails (Percorsi) della console CloudTrail.
2. Seleziona il trail creato.
3. Scegli Elimina.
4. Scegli Elimina.

Tutorial: invio di eventi a uno stream Amazon Kinesis utilizzando EventBridge e lo schema `aws.events`

Puoi inviare eventi di chiamata AWS API in EventBridge un flusso Amazon Kinesis, creare applicazioni Kinesis Data Streams ed elaborare grandi quantità di dati. In questo tutorial, crei uno stream Kinesis e poi crei una regola nella EventBridge console che invia eventi a quel flusso quando un'istanza Amazon EC2 si interrompe.

Fasi:

Prerequisiti

In questo tutorial, utilizzerai quanto segue:

- Utilizzalo AWS CLI per lavorare con gli stream Kinesis.

  Per installare AWS CLI, consulta Installazione, aggiornamento e disinstallazione della AWS CLI versione 2.

> **Note**
>
> Questo tutorial utilizza AWS gli eventi e il registro `aws.events` dello schema integrato. È inoltre possibile creare una EventBridge regola basata sullo schema degli eventi personalizzati aggiungendoli manualmente a un registro degli schemi personalizzato o utilizzando l'individuazione dello schema.
> Per ulteriori informazioni sugli schemi, consulta ???. Per ulteriori informazioni sulla creazione di una regola utilizzando altre opzioni del modello di eventi, consulta ???.

Passaggio 1: creare un flusso Amazon Kinesis

Per creare uno stream, al prompt dei comandi, utilizzare il `create-stream` AWS CLI comando.

```
aws kinesis create-stream --stream-name test --shard-count 1
```

Quando lo stato del flusso è `ACTIVE`, il flusso è pronto. Per controllare lo stato del flusso, usa il comando `describe-stream`.

```
aws kinesis describe-stream --stream-name test
```

Fase 2: Creazione di una regola

Crea una regola per inviare eventi al flusso quando arresti un'istanza Amazon EC2.

Per creare una regola

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Immettere un nome e una descrizione per la regola. Ad esempio, assegnale il nome `TestRule`.
5. In Router di eventi, seleziona Predefinito.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Seleziona Successivo.
8. Per Event source, scegli AWS eventi o eventi EventBridge partner.
9. In Metodo di creazione, scegli Utilizza schema.
10. Per Event pattern (Modello di eventi), procedi come segue:

    a. In Tipo di schema, scegli Seleziona lo schema dal registro schemi.

    b. In Registro dello schema, scegli aws.events dall'elenco a discesa.

    c. Per Schema, scegli aws.ec2 @EC2 InstanceStateChangeNotification dall'elenco a discesa.

    EventBridge visualizza lo schema degli eventi in Modelli.

    EventBridge visualizza un asterisco rosso accanto a tutte le proprietà necessarie per l'evento, non per il modello di evento.

d. In Modelli, imposta le seguenti proprietà di filtro di eventi:

   i. Seleziona + Modifica accanto alla proprietà state.

      Lascia vuoto il campo Relazione. In Valore, specifica `running`. Scegli Imposta.

   ii. Seleziona + Modifica accanto alla proprietà source.

      Lascia vuoto il campo Relazione. In Valore, specifica `aws.ec2`. Scegli Imposta.

   iii. Seleziona + Modifica accanto alla proprietà detail-type.

      Lascia vuoto il campo Relazione. In Valore, specifica `EC2 Instance State-change Notification`. Scegli Imposta.

e. Per visualizzare il modello di eventi che hai creato, scegli Genera un modello di eventi in JSON

   EventBridge visualizza lo schema degli eventi in JSON:

```
{
  "detail": {
    "state": ["running"]
  },
  "detail-type": ["EC2 Instance State-change Notification"],
  "source": ["aws.ec2"]
}
```

11. Seleziona Successivo.
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. In Seleziona una destinazione, scegli Flusso Kinesis dall'elenco a discesa.
14. In Flusso, seleziona il flusso Kinesis che hai creato nella sezione Passaggio 1: creare un flusso Amazon Kinesis. In questo esempio, seleziona `test`.
15. In Ruolo di esecuzione, scegli Crea un nuovo ruolo per questa risorsa specifica.
16. Seleziona Successivo.
17. Seleziona Successivo.
18. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Fase 3: Test della regola

Per testare la regola, arresta un'istanza Amazon EC2. Attendi qualche minuto che l'istanza si fermi, quindi controlla le CloudWatch metriche per verificare che la funzione sia stata eseguita.

Test della regola arrestando un'istanza

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Avvia un'istanza. Per ulteriori informazioni, consulta Launch Your Instance nella Amazon EC2 User Guide.
3. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
4. Nel pannello di navigazione, scegli Regole.

   Scegliere il nome della regola creata, quindi scegliere Metrics for the rule (Parametri per la regola).
5. (Opzionale) Al completamento dell'operazione, terminare l'istanza. Per ulteriori informazioni, consulta Terminate Your Instance nella Amazon EC2 User Guide.

Passaggio 4: verificare l'invio dell'evento

Puoi usare il AWS CLI per ottenere il record dallo stream per verificare che l'evento sia stato inviato.

Per ottenere il record

1. Per iniziare a leggere dal tuo flusso Kinesis, al prompt dei comandi, utilizza il comando `get-shard-iterator`.

   ```
   aws kinesis get-shard-iterator --shard-id shardId-000000000000 --shard-iterator-type TRIM_HORIZON --stream-name test
   ```

   Di seguito è riportato un output di esempio.

   ```
   {
       "ShardIterator": "AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp+KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd+efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg="
   }
   ```

2. Per ottenere il record, utilizzare il comando `get-records` seguente. Utilizza l'iteratore di partizione dell'output nel passaggio precedente.

```
aws kinesis get-records --shard-
iterator AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp
+KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd
+efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg=
```

   Se il comando viene completato correttamente, richiede record dal flusso per lo shard specificato. Puoi ricevere zero o più record. Qualsiasi record restituito potrebbe non rappresentare tutti i record nel flusso. Se non si ricevono i dati previsti, continuare a chiamare `get-records`.
3. I record in Kinesis sono codificati in Base64. Utilizza un decodificatore Base64 per decodificare i dati in modo da poter verificare che si tratta dell'evento inviato al flusso in formato JSON.

Passaggio 5: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia conservarle. Eliminando AWS le risorse che non utilizzi più, eviti addebiti inutili sul tuo AWS account.

Per eliminare la/le EventBridge regola/e

1. Apri la pagina Regole della EventBridge console.
2. Seleziona la regola che hai creato.
3. Scegliere Delete (Elimina).
4. Scegliere Delete (Elimina).

Per eliminare il flusso Kinesis

1. Apri la pagina dei flussi di dati della console Kinesis.
2. Seleziona il flusso creato.
3. Scegli Operazioni > Elimina.
4. Immetti elimina nel campo e scegli Elimina.

Tutorial: pianificazione di snapshot Amazon EBS automatizzati mediante EventBridge

Puoi eseguire le regole EventBridge in base a una pianificazione. In questo tutorial, creerai uno snapshot di un volume Amazon Elastic Block Store (Amazon EBS) esistente in base a una pianificazione. Puoi scegliere una frequenza fissa per creare uno snapshot ogni pochi minuti oppure utilizzare un'espressione Cron per creare lo snapshot a un orario specifico del giorno.

> **Important**
>
> Per creare regole con destinazioni integrate, devi utilizzare la AWS Management Console.

Passaggi:

- Passaggio 1: creare la regola
- Passaggio 2: testare la regola
- Passaggio 3: verificare il corretto completamento del tutorial
- Passaggio 4: eliminare le risorse

Passaggio 1: creare la regola

Crea una regola che acquisisce snapshot su pianificazione. Puoi utilizzare un'espressione della frequenza o un'espressione Cron per specificare la pianificazione. Per ulteriori informazioni, consulta Creazione di una regola Amazon EventBridge eseguita in base a una pianificazione.

Per creare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola.

   Una regola non può avere lo stesso nome di un'altra regola nella stessa Regione e nello stesso router di eventi.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se desideri che questa regola venga attivata su eventi corrispondenti provenienti

dal tuo account, seleziona Bus eventi predefinito AWS. Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.

6. Per Rule type (Tipo di regola), scegli Schedule (Pianifica).
7. Scegli Successivo.
8. In Modello di pianificazione, scegli Una pianificazione che viene eseguita a una frequenza regolare, ad esempio ogni 10 minuti., immetti **5** e scegli Minuti nell'elenco a discesa.
9. Scegli Successivo.
10. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
11. In Seleziona una destinazione, scegli Crea snapshot EBS dall'elenco a discesa.
12. In ID volume, immetti l'ID volume del volume Amazon EBS.
13. In Ruolo di esecuzione, scegli Crea un nuovo ruolo per questa risorsa specifica.
14. Scegli Successivo.
15. Scegli Successivo.
16. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 2: testare la regola

Puoi verificare se la tua regola funziona visualizzando il primo snapshot acquisito.

Per testare la regola

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nel riquadro di navigazione, selezionare Elastic Block Store, Snapshots (Snapshot).
3. Verifica che il primo snapshot venga visualizzato nell'elenco.

Passaggio 3: verificare il corretto completamento del tutorial

Se vedi lo snapshot nell'elenco, significa che hai completato correttamente questo tutorial. Se lo snapshot non è nell'elenco, inizia la risoluzione dei problemi verificando che la regola sia stata creata correttamente.

Passaggio 4: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Tutorial: invio di una notifica quando viene creato un oggetto Amazon S3

Puoi inviare notifiche e-mail quando vengono creati oggetti Amazon Simple Storage Service (Amazon S3) utilizzando Amazon EventBridge e Amazon SNS. In questo tutorial, creerai un argomento e un abbonamento SNS. Quindi, creerai una regola nella console EventBridge che invia eventi a quell'argomento quando vengono ricevuti eventi `Object Created` Amazon S3.

Passaggi:

Prerequisiti

Per ricevere eventi Amazon S3 in EventBridge, devi abilitare EventBridge nella console Amazon S3. Questo tutorial presuppone che EventBridge sia abilitato. Per maggiori informazioni, consulta Abilitazione di Amazon EventBridge nella console S3.

Passaggio 1: creare un argomento Amazon SNS

Crea un argomento per ricevere gli eventi da EventBridge.

Per creare un argomento

1. Apri la console Amazon SNS all'indirizzo https://console.aws.amazon.com/sns/v3/home.
2. Nel pannello di navigazione, scegli Topics (Argomenti).
3. Scegli Create topic (Crea argomento).
4. Per Tipo, scegliere Standard.
5. Immetti **`eventbridge-test`** come nome dell'argomento.
6. Scegli Create topic (Crea argomento).

Passaggio 2: creare una sottoscrizione Amazon SNS

Crea una sottoscrizione per ricevere notifiche e-mail da Amazon S3 quando vengono ricevuti eventi in base all'argomento.

Creazione di una sottoscrizione

1. Apri la console Amazon SNS all'indirizzo https://console.aws.amazon.com/sns/v3/home.
2. Nel riquadro di navigazione scegliere Subscriptions (Iscrizioni).
3. Scegliere Create Subscription (Crea iscrizione).
4. In ARN argomento, scegli l'argomento creato in Passaggio 1. Per questo tutorial, scegli eventbridge-test.
5. Per Protocol, scegli Email.
6. Per Endpoint, immettere il proprio indirizzo e-mail.
7. Scegliere Create Subscription (Crea iscrizione).
8. Conferma la sottoscrizione scegliendo Conferma sottoscrizione nell'e-mail che ricevi dalle notifiche AWS.

Passaggio 3: creare una regola

Crea una regola per inviare eventi al tuo argomento quando viene creato un oggetto Amazon S3.

Per creare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola. Ad esempio, assegnale il nome `s3-test`.
5. In Router di eventi, seleziona Predefinito.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Successivo.
8. In Origine evento, scegli Eventi AWS o eventi partner EventBridge.
9. In Metodo di creazione scegli Utilizza modulo del modello.
10. Per Event pattern (Modello di eventi), procedi come segue:

a. In Origine evento, seleziona Servizi AWS dall'elenco a discesa.
b. In Servizio AWS, seleziona Simple Storage Service (S3) dall'elenco a discesa.
c. In Tipo di evento, scegli Notifica evento Amazon S3 dall'elenco a discesa.
d. Scegli Eventi specifici e quindi Oggetto creato dall'elenco a discesa.
e. Scegli Qualsiasi bucket.

11. Scegli Successivo.
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. In Seleziona una destinazione, scegli l'argomento SNS dall'elenco a discesa.
14. In Argomento, seleziona l'argomento Amazon SNS che hai creato nella sezione Passaggio 1: creare un argomento SNS. In questo esempio, seleziona `eventbridge-test`.
15. Scegli Successivo.
16. Scegli Successivo.
17. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 4: testare la regola

Per testare la regola, crea un oggetto Amazon S3 caricando un file in un bucket abilitato da EventBridge. Quindi, attendi qualche minuto e verifica se ricevi un'e-mail dalle notifiche AWS.

Passaggio 5: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare l'argomento SNS

1. Apri la pagina Argomenti nella console SNS.
2. Seleziona l'argomento creato.
3. Scegli Elimina.
4. Specificare **`delete me`**.
5. Scegli Elimina.

Per eliminare la sottoscrizione SNS

1. Apri la pagina Sottoscrizioni della console SNS.
2. Seleziona la sottoscrizione creata.
3. Scegli Elimina.
4. Scegli Elimina.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Tutorial: pianificazione delle funzioni AWS Lambda mediante EventBridge

Puoi configurare una regola per l'esecuzione di una funzione AWS Lambda in basa a una pianificazione. Questo tutorial mostra come utilizzare la AWS Management Console o AWS CLI per creare la regola. Se desideri utilizzare l'AWS CLI ma non l'hai ancora installata, consulta Installazione, aggiornamento e disinstallazione dell'AWS CLI versione 2.

Per le pianificazioni, EventBridge non fornisce precisione a livello di secondo nelle espressioni di pianificazione. La risoluzione più alta che utilizza un'espressione Cron è un minuto. A causa della natura distribuita di EventBridge e dei servizi di destinazione, è possibile che vi sia un ritardo di vari secondi tra il momento in cui la regola pianificata viene attivata e il momento in cui il servizio di destinazione esegue la risorsa di destinazione.

Passaggi:

- Passaggio 1: creare una funzione Lambda
- Passaggio 2: creare una regola
- Passaggio 3: verificare la regola
- Passaggio 4: verificare il corretto completamento del tutorial
- Passaggio 5: eliminare le risorse

Passaggio 1: creare una funzione Lambda

Crea una funzione Lambda per registrare gli eventi pianificati.

Per creare una funzione Lambda

1. Apri la console AWS Lambda all'indirizzo https://console.aws.amazon.com/lambda/.
2. Selezionare Create function (Crea funzione).
3. Scegli Author from scratch (Crea da zero).
4. Digitare un nome e una descrizione per la funzione Lambda. Ad esempio, `LogScheduledEvent`.
5. Per le altre opzioni, mantieni il valore predefinito e scegli Crea funzione.
6. Nella scheda Codice della pagina della funzione, fai doppio clic su index.js.
7. Sostituisci il codice esistente con il seguente.

```
'use strict';
```

```
exports.handler = (event, context, callback) => {
    console.log('LogScheduledEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Selezionare Deploy (Distribuisci).

Passaggio 2: creare una regola

Crea una regola per eseguire la funzione Lambda creata in Passaggio 1 in base a una pianificazione.

Per creare la regola puoi utilizzare la console o l'AWS CLI. Per utilizzare AWS CLI, devi dapprima concedere alla regola l'autorizzazione di richiamare la funzione Lambda. Puoi quindi creare la regola e aggiungere la funzione Lambda come target.

Per creare una regola (console)

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola.

   Una regola non può avere lo stesso nome di un'altra regola nella stessa Regione e nello stesso router di eventi.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se desideri che questa regola venga attivata su eventi corrispondenti provenienti dal tuo account, seleziona Bus eventi predefinito AWS. Quando un servizio di AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Schedule (Pianifica).
7. Scegli Successivo.
8. In Modello di pianificazione, scegli Una pianificazione che viene eseguita a una frequenza regolare, ad esempio ogni 10 minuti., immetti **5** e scegli Minuti nell'elenco a discesa.
9. Scegli Successivo.
10. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
11. In Seleziona una destinazione, scegli la funzione Lambda dall'elenco a discesa.

12. In Funzione, seleziona la funzione Lambda che hai creato nella sezione Passaggio 1: creare una funzione Lambda. In questo esempio, seleziona `LogScheduledEvent`.
13. Scegli Successivo.
14. Scegli Successivo.
15. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Per creare una regola (AWS CLI)

1. Per creare una regola che viene eseguita in base a una pianificazione, utilizza il comando `put-rule`.

```
aws events put-rule \
--name my-scheduled-rule \
--schedule-expression 'rate(5 minutes)'
```

Quando viene eseguita, questa regola crea un evento e quindi lo invia alle destinazioni. Di seguito è riportato un esempio di evento.

```
{
    "version": "0",
    "id": "53dc4d37-cffa-4f76-80c9-8b7d4a4d2eaa",
    "detail-type": "Scheduled Event",
    "source": "aws.events",
    "account": "123456789012",
    "time": "2015-10-08T16:53:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule"
    ],
    "detail": {}
}
```

2. Per concedere al principale del servizio EventBridge (`events.amazonaws.com`) l'autorizzazione per eseguire la regola, utilizza il comando `add-permission`.

```
aws lambda add-permission \
--function-name LogScheduledEvent \
--statement-id my-scheduled-event \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
```

```
--source-arn arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule
```

3. Creare il file `targets.json` con i seguenti contenuti.

```
[
  {
    "Id": "1",
    "Arn": "arn:aws:lambda:us-east-1:123456789012:function:LogScheduledEvent"
  }
]
```

4. Per aggiungere alla regola la funzione Lambda creata in Passaggio 1, utilizza il comando `put-targets`.

```
aws events put-targets --rule my-scheduled-rule --targets file://targets.json
```

Passaggio 3: verificare la regola

Attendi almeno cinque minuti dopo avere completato il passaggio 2 per verificare che la funzione Lambda è stata richiamata.

Visualizzazione dell'output della funzione Lambda

1. Aprire la console CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegli Logs (Log).
3. Seleziona il nome del gruppo di log per la funzione Lambda (`/aws/lambda/function-name`).
4. Seleziona il nome del flusso di log per visualizzare i dati forniti dalla funzione per l'istanza avviata.

Passaggio 4: verificare il corretto completamento del tutorial

Se vedi l'evento Lambda nei log CloudWatch, significa che hai completato correttamente questo tutorial. Se l'evento non è presente nei log CloudWatch, inizia la risoluzione dei problemi verificando che la regola sia stata creata correttamente e, se è il caso, verifica che il codice della funzione Lambda sia corretto.

Passaggio 5: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Per eliminare la funzione Lambda

1. Aprire la pagina Funzioni della console Lambda.
2. Seleziona la funzione creata.
3. Scegliere Actions (Operazioni), Delete (Elimina).
4. Scegli Elimina.

Tutorial di Amazon EventBridge per l'integrazione con provider SaaS

EventBridge può essere utilizzato direttamente con applicazioni e servizi dei partner SaaS per inviare e ricevere eventi. I tutorial seguenti mostrano come integrare EventBridge con partner SaaS.

Tutorial:

- Tutorial: creazione di una connessione a Datadog come destinazione API
- Tutorial: creazione di una connessione a Salesforce come destinazione API
- Tutorial: creazione di una connessione a Zendesk come destinazione API

Tutorial: creazione di una connessione a Datadog come destinazione API

Puoi utilizzare EventBridge per instradare eventi a servizi di terze parti, come Datadog.

In questo tutorial, utilizzerai la console EventBridge per creare una connessione a Datadog, una destinazione API che punta a Datadog e una regola per instradare eventi a Datadog.

Passaggi:

Prerequisiti

Per completare questo tutorial, avrai bisogno delle seguenti risorse:

- Un account Datadog.
- Una chiave API Datadog.
- Un bucket Amazon Simple Storage Service (Amazon S3) abilitato da EventBridge.

Passaggio 1: creare una connessione

Per inviare eventi a Datadog, devi prima stabilire una connessione all'API Datadog.

Per creare la connessione

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scegli la scheda Connessioni, quindi Crea connessione.
4. Immetti un nome e una descrizione per la connessione. Ad esempio, immetti **`Datadog`** come nome e **`Datadog API Connection`** come descrizione.
5. In Tipo di autorizzazione, scegli Chiave API.
6. In Nome chiave API, immetti **`DD-API-KEY`**.

7. In Valore, incolla la tua chiave API segreta Datadog.
8. Scegli Crea.

Passaggio 2: creare una destinazione API

Ora che hai creato la connessione, devi creare la destinazione API da utilizzare come destinazione della regola.

Per creare la destinazione API

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scegli Crea una destinazione API.
4. Immetti un nome e una descrizione per la destinazione API. In questo esempio, immetti **`DatadogAD`** come nome e **`Datadog API Destination`** come descrizione.
5. In Endpoint di destinazione API, immetti **`https://http-intake.logs.datadoghq.com/api/v2/logs`**.
6. In Metodo HTTP, scegli POST.
7. In Limite di velocità di invocazione, immetti **`300`**.
8. In Connessione, scegli Utilizza una connessione esistente e scegli la connessione `Datadog` che hai creato in Passaggio 1.
9. Scegli Crea.

Passaggio 3: creare una regola

Ora creerai una regola per inviare eventi a Datadog quando viene creato un oggetto Amazon S3.

Per creare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola. In questo esempio, immetti **`DatadogRule`** come nome e **`Rule to send events to Datadog for S3 object creation`** come descrizione.

5. Per Event bus (Bus di eventi), scegli default.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "source": ["aws.s3"]
}
```

10. Scegli Successivo.
11. In Tipi di destinazione, scegli Destinazione API di EventBridge.
12. In Destinazione API, scegli Utilizza una destinazione API esistente, quindi scegli la destinazione `DatadogAD` creata in Passaggio 2.
13. In Ruolo di esecuzione, scegli Crea un nuovo ruolo per questa risorsa specifica.
14. In Impostazioni aggiuntive, procedi come segue:
    a. In Configura l'input di destinazione, scegli Trasformatore di input dall'elenco a discesa.
    b. Scegli Configura il trasformatore di input.
    c. In Eventi di esempio, immetti quanto segue:

```
{
  "detail":[]
}
```

    d. In Trasformatore di input di destinazione, procedi come segue:
        i. In Percorso di input, immetti quanto segue:

```
{"detail":"$.detail"}
```

        ii. In Modello di input, immetti quanto segue:

```
{"message": <detail>}
```

    e. Scegli Conferma.
15. Scegli Successivo.

16. Scegli Successivo.
17. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 4: testare la regola

Per testare la regola, crea un oggetto Amazon S3 caricando un file in un bucket abilitato da EventBridge. L'oggetto creato verrà registrato nella console Datadog Logs.

Passaggio 5: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare la connessione EventBridge

1. Apri la pagina Destinazione API nella console Amazon EventBridge.
2. Scegli la scheda Connessioni.
3. Seleziona la connessione che hai creato.
4. Scegli Elimina.
5. Immetti il nome della connessione e scegli Elimina.

Per eliminare la destinazione API di EventBridge

1. Apri la pagina Destinazione API nella console Amazon EventBridge.
2. Seleziona la destinazione API che hai creato.
3. Scegli Elimina.
4. Immetti il nome della destinazione API e scegli Elimina.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Tutorial: creazione di una connessione a Salesforce come destinazione API

È possibile utilizzare EventBridge per indirizzare gli eventi a servizi di terze parti, ad esempio Salesforce.

In questo tutorial, utilizzerai la EventBridge console per creare una connessioneSalesforce, una destinazione API che punti e una regola a cui indirizzare gli eventiSalesforce. Salesforce

Fasi:

- Prerequisiti
- Passaggio 1: creare una connessione
- Passaggio 2: creare una destinazione API
- Passaggio 3: creare una regola
- Passaggio 4: testare la regola
- Passaggio 5: eliminare le risorse

Prerequisiti

Per completare questo tutorial, avrai bisogno delle seguenti risorse:

- Un account Salesforce.
- Un'app connessa Salesforce.
- Un token di sicurezza Salesforce.
- Un evento di piattaforma personalizzato Salesforce.
- Un EventBridge bucket Amazon Simple Storage Service (Amazon S3) abilitato per Amazon.

Passaggio 1: creare una connessione

Per inviare eventi a Salesforce, devi prima stabilire una connessione all'API Salesforce.

Per creare la connessione

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scegli la scheda Connessioni, quindi Crea connessione.

4. Immetti un nome e una descrizione per la connessione. Ad esempio, immetti **`Salesforce`** come nome e **`Salesforce API Connection`** come descrizione.
5. In Tipo di destinazione, scegli Partner e in Destinazioni partner, seleziona Salesforce dall'elenco a discesa.
6. In Endpoint di autorizzazione, immetti:
   - **`https://`*`MyDomainName`*`.my.salesforce.com./services/oauth2/token`** se utilizzi un'organizzazione di produzione
   - **`https://`*`MyDomainName`*`--`*`SandboxName`*`.my. salesforce.com/services /oauth2/token`** se utilizzi un ambiente di sperimentazione (sandbox) senza domini avanzati
   - **`https://`*`MyDomainName`*`-- `*`SandboxName`*`.sandbox.my.salesforce.com/services/oauth2/token`** se utilizzi un ambiente di sperimentazione (sandbox) con domini avanzati
7. In Metodo HTTP, scegli POST dall'elenco a discesa.
8. In ID client, immetti l'ID client dell'app Salesforce connessa.
9. In Segreto client, immetti il segreto client dell'app Salesforce connessa.
10. Per i parametri HTTP OAuth, inserisci la seguente coppia chiave/valore:

Key (Chiave)	Value (Valore)
grant_type	client_credentials

11. Scegli Crea.

Passaggio 2: creare una destinazione API

Ora che hai creato la connessione, devi creare la destinazione API da utilizzare come destinazione della regola.

Per creare la destinazione API

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scegli Crea una destinazione API.
4. Immetti un nome e una descrizione per la destinazione API. In questo esempio, immetti **`SalesforceAD`** come nome e **`Salesforce API Destination`** come descrizione.

5. In Endpoint di destinazione API, immetto **`https://MyDomainName.my.salesforce.com/services/data/v54.0/sobjects/MyEvent__e`** dove Myevent__e è l'evento della piattaforma a cui desideri inviare le informazioni.
6. In Metodo HTTP, scegli POST dall'elenco a discesa.
7. In Limite di velocità di invocazione, immetti **`300`**.
8. In Connessione, scegli Utilizza una connessione esistente e scegli la connessione `Salesforce` che hai creato in Passaggio 1.
9. Scegli Crea.

Passaggio 3: creare una regola

Ora creerai una regola per inviare eventi a Salesforce quando viene creato un oggetto Amazon S3.

Per creare una regola

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Immettere un nome e una descrizione per la regola. In questo esempio, immetti **`SalesforceRule`** come nome e **`Rule to send events to Salesforce for S3 object creation`** come descrizione.
5. Per Event bus (Bus di eventi), scegli default.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Seleziona Avanti.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "source": ["aws.s3"]
}
```

10. Seleziona Avanti.
11. Per i tipi di Target, scegli la destinazione EventBridge dell'API.

12. In Destinazione API, scegli Utilizza una destinazione API esistente, quindi scegli la destinazione `SalesforceAD` creata in Passaggio 2.
13. In Ruolo di esecuzione, scegli Crea un nuovo ruolo per questa risorsa specifica.
14. In Impostazioni aggiuntive, procedi come segue:

    a. In Configura l'input di destinazione, scegli Trasformatore di input dall'elenco a discesa.

    b. Scegli Configura il trasformatore di input.

    c. In Eventi di esempio, immetti quanto segue:

    ```
    {
      "detail":[]
    }
    ```

    d. In Trasformatore di input di destinazione, procedi come segue:

        i. In Percorso di input, immetti quanto segue:

        ```
        {"detail":"$.detail"}
        ```

        ii. In Modello di input, immetti quanto segue:

        ```
        {"message": <detail>}
        ```

    e. Scegli Conferma.
15. Seleziona Avanti.
16. Seleziona Avanti.
17. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 4: testare la regola

Per testare la tua regola, crea un oggetto Amazon S3 caricando un file in un bucket abilitato. EventBridge Le informazioni sull'oggetto creato verranno inviate all'evento della piattaforma Salesforce.

Passaggio 5: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia conservarle. Eliminando AWS le risorse che non utilizzi più, eviti addebiti inutili sul tuo account. AWS

Per eliminare le EventBridge connessioni

1. Apri la pagina di destinazione dell'API della EventBridge console.
2. Scegliere la scheda Connessioni.
3. Seleziona la connessione che hai creato.
4. Scegli Elimina.
5. Immetti il nome della connessione e scegli Elimina.

Per eliminare le destinazioni EventBridge API

1. Apri la pagina di destinazione dell'API della EventBridge console.
2. Seleziona la destinazione API che hai creato.
3. Scegli Elimina.
4. Immetti il nome della destinazione API e scegli Elimina.

Per eliminare le EventBridge regole

1. Apri la pagina Regole della EventBridge console.
2. Seleziona la regola che hai creato.
3. Scegliere Delete (Elimina).
4. Scegli Delete (Elimina).

Tutorial: creazione di una connessione a Zendesk come destinazione API

Puoi utilizzare EventBridge per instradare eventi a servizi di terze parti, ad esempio Zendesk.

In questo tutorial, utilizzerai la console EventBridge per creare una connessione a Zendesk, una destinazione API che punta a Zendesk e una regola per instradare eventi a Zendesk.

Passaggi:

Prerequisiti

Per completare questo tutorial, avrai bisogno delle seguenti risorse:

- Un account Zendesk.
- Un bucket Amazon Simple Storage Service (Amazon S3) abilitato da EventBridge.

Passaggio 1: creare una connessione

Per inviare eventi a Zendesk, devi prima stabilire una connessione all'API Zendesk.

Per creare la connessione

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scegli la scheda Connessioni, quindi Crea connessione.
4. Immetti un nome e una descrizione per la connessione. In questo esempio, immetti **`Zendesk`** come nome e **`Connection to Zendesk API`** come descrizione.
5. In Tipo di autorizzazione, scegli Base (nome utente/password).
6. In Nome utente, immetti il tuo nome utente Zendesk.
7. In Password, immetti la password Zendesk.

8. Scegli Crea.

Passaggio 2: creare una destinazione API

Ora che hai creato la connessione, creerai la destinazione API da utilizzare come destinazione della regola.

Per creare la destinazione API

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli Destinazioni API.
3. Scegli Crea una destinazione API.
4. Immetti un nome e una descrizione per la destinazione API. In questo esempio, immetti **`ZendeskAD`** come nome e **`Zendesk API destination`** come descrizione.
5. In Endpoint di destinazione API, immetti **`https://`*`your-subdomain`*`.zendesk.com/api/v2/tickets.json`**, dove *`your-subdomain`* è il sottodominio associato al tuo account Zendesk.
6. In Metodo HTTP, scegli POST.
7. In Limite di velocità di invocazione, immetti **`10`**.
8. In Connessione, scegli Utilizza una connessione esistente e scegli la connessione `Zendesk` che hai creato in Passaggio 1.
9. Scegli Crea.

Passaggio 3: creare una regola

Ora creerai una regola per inviare eventi a Zendesk quando viene creato un oggetto Amazon S3.

Per creare una regola

1. Aprire la console Amazon EventBridge all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegliere Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola. In questo esempio, immetti **`ZendeskRule`** come nome e **`Rule to send events to Zendesk when S3 objects are created`** come descrizione.

5. Per Event bus (Bus di eventi), scegli default.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "source": ["aws.s3"]
}
```

10. Scegli Successivo.
11. In Tipi di destinazione, scegli Destinazione API di EventBridge.
12. In Destinazione API, scegli Utilizza una destinazione API esistente, quindi scegli la destinazione `ZendeskAD` creata in Passaggio 2.
13. In Ruolo di esecuzione, scegli Crea un nuovo ruolo per questa risorsa specifica.
14. In Impostazioni aggiuntive, procedi come segue:
    a. In Configura l'input di destinazione, scegli Trasformatore di input dall'elenco a discesa.
    b. Scegli Configura il trasformatore di input.
    c. In Eventi di esempio, immetti quanto segue:

```
{
  "detail":[]
}
```

    d. In Trasformatore di input di destinazione, procedi come segue:
        i. In Percorso di input, immetti quanto segue:

```
{"detail":"$.detail"}
```

        ii. In Modello di input, immetti quanto segue:

```
{"message": <detail>}
```

    e. Scegli Conferma.
15. Scegli Successivo.

16. Scegli Successivo.
17. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 4: testare la regola

Per testare la regola, crea un oggetto Amazon S3 caricando un file in un bucket abilitato da EventBridge. Quando l'evento corrisponde alla regola, EventBridge chiamerà l'API di creazione di ticket Zendesk. Il nuovo ticket apparirà nella dashboard Zendesk.

Passaggio 5: eliminare le risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia mantenerle. Eliminando le risorse AWS non più utilizzate, puoi evitare addebiti inutili sul tuo account AWS.

Per eliminare la connessione EventBridge

1. Apri la pagina Destinazione API nella console Amazon EventBridge.
2. Scegli la scheda Connessioni.
3. Seleziona la connessione che hai creato.
4. Scegli Elimina.
5. Immetti il nome della connessione e scegli Elimina.

Per eliminare la destinazione API di EventBridge

1. Apri la pagina Destinazione API nella console Amazon EventBridge.
2. Seleziona la destinazione API che hai creato.
3. Scegli Elimina.
4. Immetti il nome della destinazione API e scegli Elimina.

Per eliminare la regola Amazon EventBridge

1. Apri la pagina Regole nella console Amazon EventBridge.
2. Seleziona la regola che hai creato.
3. Scegli Elimina.
4. Scegli Elimina.

Utilizzo EventBridge con un AWS SDK

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK for C++	AWS SDK for C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK for Go	AWS SDK for 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 EventBridge, vedereEsempi di codice per l' EventBridge utilizzo degli AWS SDK.

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.

Esempi di codice per l' EventBridge utilizzo degli AWS SDK

I seguenti esempi di codice mostrano come utilizzarlo EventBridge 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 operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Esempi cross-service: applicazioni di esempio che funzionano su più servizi Servizi AWS.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

Salve EventBridge

I seguenti esempi di codice mostrano come iniziare a utilizzare EventBridge.

.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 Amazon.EventBridge;
using Amazon.EventBridge.Model;

namespace EventBridgeActions;
```

```
public static class HelloEventBridge
{
    static async Task Main(string[] args)
    {
        var eventBridgeClient = new AmazonEventBridgeClient();

        Console.WriteLine($"Hello Amazon EventBridge! Following are some of your
 EventBuses:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first five event buses.
        var response = await eventBridgeClient.ListEventBusesAsync(
            new ListEventBusesRequest()
            {
                Limit = 5
            });

        foreach (var eventBus in response.EventBuses)
        {
            Console.WriteLine($"\tEventBus: {eventBus.Name}");
            Console.WriteLine($"\tArn: {eventBus.Arn}");
            Console.WriteLine($"\tPolicy: {eventBus.Policy}");
            Console.WriteLine();
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListEventBusessezione AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/**
 * 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 HelloEventBridge {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        EventBridgeClient eventBrClient = EventBridgeClient.builder()
                .region(region)
                .build();

        listBuses(eventBrClient);
        eventBrClient.close();
    }

    public static void listBuses(EventBridgeClient eventBrClient) {
        try {
            ListEventBusesRequest busesRequest = ListEventBusesRequest.builder()
                    .limit(10)
                    .build();

            ListEventBusesResponse response =
 eventBrClient.listEventBuses(busesRequest);
            List<EventBus> buses = response.eventBuses();
            for (EventBus bus : buses) {
                System.out.println("The name of the event bus is: " +
 bus.name());
                System.out.println("The ARN of the event bus is: " + bus.arn());
            }

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListEventBusessezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import aws.sdk.kotlin.services.eventbridge.EventBridgeClient
import aws.sdk.kotlin.services.eventbridge.model.ListEventBusesRequest
import aws.sdk.kotlin.services.eventbridge.model.ListEventBusesResponse

suspend fun main() {
    listBusesHello()
}

suspend fun listBusesHello() {
    val request = ListEventBusesRequest {
        limit = 10
    }

    EventBridgeClient { region = "us-west-2" }.use { eventBrClient ->
        val response: ListEventBusesResponse =
 eventBrClient.listEventBuses(request)
        response.eventBuses?.forEach { bus ->
            println("The name of the event bus is ${bus.name}")
            println("The ARN of the event bus is ${bus.arn}")
        }
    }
}
```

- Per i dettagli sull'API, ListEventBusesconsulta AWS SDK for Kotlin API reference.

Esempi di codice

Azioni per l'utilizzo degli SDK EventBridge AWS

I seguenti esempi di codice mostrano come eseguire EventBridge azioni individuali con gli AWS SDK. Questi estratti richiamano l' EventBridge API e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. Ogni esempio include un collegamento a GitHub, dove è possibile trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta Amazon EventBridge API Reference.

Esempi

- Utilizzo DeleteRule con un AWS SDK o una CLI
- Utilizzo DescribeRule con un AWS SDK o una CLI
- Utilizzo DisableRule con un AWS SDK o una CLI
- Utilizzo EnableRule con un AWS SDK o una CLI

- Utilizzo ListRuleNamesByTarget con un AWS SDK o una CLI
- Utilizzo ListRules con un AWS SDK o una CLI
- Utilizzo ListTargetsByRule con un AWS SDK o una CLI
- Utilizzo PutEvents con un AWS SDK o una CLI
- Utilizzo PutRule con un AWS SDK o una CLI
- Utilizzo PutTargets con un AWS SDK o una CLI
- Utilizzo RemoveTargets con un AWS SDK o una CLI

Utilizzo **`DeleteRule`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteRule`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su regole e destinazioni

.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.

Elimina una regola in base al nome della stessa.

```
/// <summary>
/// Delete an event rule by name.
/// </summary>
/// <param name="ruleName">The name of the event rule.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteRuleByName(string ruleName)
{
    var response = await _amazonEventBridge.DeleteRuleAsync(
        new DeleteRuleRequest()
```

```
            {
                Name = ruleName
            });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteRulesezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per eliminare una regola CloudWatch Events

Questo esempio elimina la regola denominata InstanceStateChanges EC2:

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- Per i dettagli sull'API, consulta AWS CLI Command DeleteRuleReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void deleteRuleByName(EventBridgeClient eventBrClient, String
ruleName) {
       DeleteRuleRequest ruleRequest = DeleteRuleRequest.builder()
               .name(ruleName)
               .build();

       eventBrClient.deleteRule(ruleRequest);
       System.out.println("Successfully deleted the rule");
```

```
    }
```

- Per i dettagli sull'API, consulta la DeleteRulesezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteRuleByName(ruleName: String?) {
    val ruleRequest = DeleteRuleRequest {
        name = ruleName
    }
    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.deleteRule(ruleRequest)
        println("Successfully deleted the rule")
    }
}
```

- Per i dettagli sull'API, DeleteRuleconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeRule`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeRule`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su regole e destinazioni

.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.

Ottieni lo stato di una regola utilizzando la descrizione della regola.

```
    /// <summary>
    /// Get the state for a rule by the rule name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="eventBusName">The optional name of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The state of the rule.</returns>
    public async Task<RuleState> GetRuleStateByRuleName(string ruleName, string?
eventBusName = null)
    {
        var ruleResponse = await _amazonEventBridge.DescribeRuleAsync(
            new DescribeRuleRequest()
            {
                Name = ruleName,
                EventBusName = eventBusName
            });
        return ruleResponse.State;
    }
```

- Per i dettagli sull'API, consulta la DescribeRulesezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per visualizzare informazioni su una regola CloudWatch Events

Questo esempio visualizza informazioni sulla regola denominata DailyLambdaFunction:

```
aws events describe-rule --name "DailyLambdaFunction"
```

- Per i dettagli sull'API, vedere DescribeRulein AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void checkRule(EventBridgeClient eventBrClient, String
eventRuleName) {
       try {
           DescribeRuleRequest ruleRequest = DescribeRuleRequest.builder()
                   .name(eventRuleName)
                   .build();

           DescribeRuleResponse response =
eventBrClient.describeRule(ruleRequest);
           System.out.println("The state of the rule is " +
response.stateAsString());

       } catch (EventBridgeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Per i dettagli sull'API, consulta la DescribeRulesezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun checkRule(eventRuleName: String?) {
    val ruleRequest = DescribeRuleRequest {
        name = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.describeRule(ruleRequest)
        println("The state of the rule is $response")
    }
}
```

- Per i dettagli sull'API, DescribeRuleconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DisableRule`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DisableRule`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su regole e destinazioni

.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.

Disabilita una regola in base al nome della stessa.

```
/// <summary>
/// Disable a particular rule on an event bus.
/// </summary
/// <param name="ruleName">The name of the rule.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DisableRuleByName(string ruleName)
{
    var ruleResponse = await _amazonEventBridge.DisableRuleAsync(
        new DisableRuleRequest()
        {
            Name = ruleName
        });
    return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
}
```

- Per i dettagli sull'API, consulta la DisableRulesezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per disabilitare una regola CloudWatch Events

Questo esempio disabilita la regola denominata DailyLambdaFunction. La regola non viene eliminata:

```
aws events disable-rule --name "DailyLambdaFunction"
```

- Per i dettagli sull'API, vedere DisableRulein AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Disabilita una regola utilizzando il nome della stessa.

```
   public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
       try {
           if (!isEnabled) {
               System.out.println("Disabling the rule: " + eventRuleName);
               DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();

               eventBrClient.disableRule(ruleRequest);
           } else {
               System.out.println("Enabling the rule: " + eventRuleName);
               EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();
               eventBrClient.enableRule(ruleRequest);
           }

       } catch (EventBridgeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Per i dettagli sull'API, consulta la DisableRulesezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}
```

- Per i dettagli sull'API, DisableRuleconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`EnableRule`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`EnableRule`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su regole e destinazioni

.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.

Abilita una regola in base al nome della stessa.

```
/// <summary>
/// Enable a particular rule on an event bus.
/// </summary>
/// <param name="ruleName">The name of the rule.</param>
/// <returns>True if successful.</returns>
public async Task<bool> EnableRuleByName(string ruleName)
{
    var ruleResponse = await _amazonEventBridge.EnableRuleAsync(
        new EnableRuleRequest()
        {
            Name = ruleName
        });
    return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
}
```

- Per i dettagli sull'API, consulta la EnableRulesezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per abilitare una regola CloudWatch Events

Questo esempio abilita la regola denominata DailyLambdaFunction, che era stata precedentemente disabilitata:

```
aws events enable-rule --name "DailyLambdaFunction"
```

- Per i dettagli sull'API, consulta EnableRule AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Abilita una regola utilizzando il nome della stessa.

```
   public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
       try {
           if (!isEnabled) {
               System.out.println("Disabling the rule: " + eventRuleName);
               DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();

               eventBrClient.disableRule(ruleRequest);
           } else {
               System.out.println("Enabling the rule: " + eventRuleName);
               EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();
               eventBrClient.enableRule(ruleRequest);
           }

       } catch (EventBridgeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
```

```
    }
```

- Per i dettagli sull'API, consulta la EnableRulesezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}
```

- Per i dettagli sull'API, EnableRuleconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListRuleNamesByTarget`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListRuleNamesByTarget`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su regole e destinazioni

.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.

Elenca tutti i nomi delle regole utilizzando la destinazione.

```
/// <summary>
/// List names of all rules matching a target.
/// </summary>
/// <param name="targetArn">The ARN of the target.</param>
/// <returns>The list of rule names.</returns>
public async Task<List<string>> ListAllRuleNamesByTarget(string targetArn)
{
    var results = new List<string>();
    var request = new ListRuleNamesByTargetRequest()
    {
        TargetArn = targetArn
    };
    ListRuleNamesByTargetResponse response;
    do
    {
```

```
            response = await
_amazonEventBridge.ListRuleNamesByTargetAsync(request);
            results.AddRange(response.RuleNames);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }
```

- Per i dettagli sull'API, consulta la ListRuleNamesByTargetsezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per visualizzare tutte le regole che hanno un obiettivo specificato

Questo esempio visualizza tutte le regole che hanno come destinazione la funzione Lambda denominata MyFunctionName "":

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-
east-1:123456789012:function:MyFunctionName"
```

- Per i dettagli sull'API, consulta ListRuleNamesByTarget AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca tutti i nomi delle regole utilizzando la destinazione.

```
   public static void listTargetRules(EventBridgeClient eventBrClient, String
topicArn) {
       ListRuleNamesByTargetRequest ruleNamesByTargetRequest =
ListRuleNamesByTargetRequest.builder()
               .targetArn(topicArn)
               .build();

       ListRuleNamesByTargetResponse response =
eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest);
       List<String> rules = response.ruleNames();
       for (String rule : rules) {
           System.out.println("The rule name is " + rule);
       }
   }
```

- Per i dettagli sull'API, consulta la ListRuleNamesByTargetsezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listTargetRules(topicArnVal: String?) {
    val ruleNamesByTargetRequest = ListRuleNamesByTargetRequest {
        targetArn = topicArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response =
eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest)
        response.ruleNames?.forEach { rule ->
            println("The rule name is $rule")
        }
    }
```

```
}
```

- Per i dettagli sull'API, ListRuleNamesByTargetconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListRules`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListRules`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su regole e destinazioni

.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.

Elenca tutte le regole per un router di eventi.

```
    /// <summary>
    /// List the rules on an event bus.
    /// </summary>
    /// <param name="eventBusArn">The optional ARN of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The list of rules.</returns>
    public async Task<List<Rule>> ListAllRulesForEventBus(string? eventBusArn =
null)
    {
```

```
        var results = new List<Rule>();
        var request = new ListRulesRequest()
        {
            EventBusName = eventBusArn
        };
        // Get all of the pages of rules.
        ListRulesResponse response;
        do
        {
            response = await _amazonEventBridge.ListRulesAsync(request);
            results.AddRange(response.Rules);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }
```

- Per i dettagli sull'API, consulta la ListRulessezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per visualizzare un elenco di tutte le regole CloudWatch degli eventi

Questo esempio visualizza tutte le regole CloudWatch Events della regione:

```
aws events list-rules
```

Per visualizzare un elenco di regole CloudWatch Events che iniziano con una determinata stringa.

Questo esempio visualizza tutte le regole CloudWatch Events nella regione il cui nome inizia con «Daily»:

```
aws events list-rules --name-prefix "Daily"
```

- Per i dettagli sull'API, consulta ListRules AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Abilita una regola utilizzando il nome della stessa.

```
    public static void listRules(EventBridgeClient eventBrClient) {
        try {
            ListRulesRequest rulesRequest = ListRulesRequest.builder()
                    .eventBusName("default")
                    .limit(10)
                    .build();

            ListRulesResponse response = eventBrClient.listRules(rulesRequest);
            List<Rule> rules = response.rules();
            for (Rule rule : rules) {
                System.out.println("The rule name is : " + rule.name());
                System.out.println("The rule description is : " +
rule.description());
                System.out.println("The rule state is : " +
rule.stateAsString());
            }

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta la ListRulessezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listRules() {
    val rulesRequest = ListRulesRequest {
        eventBusName = "default"
        limit = 10
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listRules(rulesRequest)
        response.rules?.forEach { rule ->
            println("The rule name is ${rule.name}")
            println("The rule ARN is ${rule.arn}")
        }
    }
}
```

- Per i dettagli sull'API, ListRulesconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListTargetsByRule`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListTargetsByRule`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su regole e destinazioni

.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.

Elenca tutte le destinazioni di una regola utilizzando il nome della stessa.

```
/// <summary>
/// List all of the targets matching a rule by name.
/// </summary>
/// <param name="ruleName">The name of the rule.</param>
/// <returns>The list of targets.</returns>
public async Task<List<Target>> ListAllTargetsOnRule(string ruleName)
{
    var results = new List<Target>();
    var request = new ListTargetsByRuleRequest()
    {
        Rule = ruleName
    };
    ListTargetsByRuleResponse response;
    do
    {
        response = await _amazonEventBridge.ListTargetsByRuleAsync(request);
        results.AddRange(response.Targets);
        request.NextToken = response.NextToken;

    } while (response.NextToken is not null);

    return results;
}
```

- Per i dettagli sull'API, consulta la ListTargetsByRulesezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per visualizzare tutti gli obiettivi di una regola CloudWatch Events

Questo esempio visualizza tutti gli obiettivi della regola denominata DailyLambdaFunction:

```
aws events list-targets-by-rule --rule  "DailyLambdaFunction"
```

- Per i dettagli sull'API, consulta ListTargetsByRule AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Elenca tutte le destinazioni di una regola utilizzando il nome della stessa.

```
   public static void listTargets(EventBridgeClient eventBrClient, String
ruleName) {
       ListTargetsByRuleRequest ruleRequest = ListTargetsByRuleRequest.builder()
               .rule(ruleName)
               .build();

       ListTargetsByRuleResponse res =
eventBrClient.listTargetsByRule(ruleRequest);
       List<Target> targetsList = res.targets();
       for (Target target: targetsList) {
           System.out.println("Target ARN: "+target.arn());
       }
   }
```

- Per i dettagli sull'API, consulta la ListTargetsByRulesezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listTargets(ruleName: String?) {
    val ruleRequest = ListTargetsByRuleRequest {
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(ruleRequest)
        response.targets?.forEach { target ->
            println("Target ARN: ${target.arn}")
        }
    }
}
```

- Per i dettagli sull'API, ListTargetsByRuleconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutEvents`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutEvents`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione e attivazione di una regola
- Nozioni di base su regole e destinazioni

.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.

Invia un evento che corrisponde a un modello personalizzato per una regola.

```
    /// <summary>
    /// Add an event to the event bus that includes an email, message, and time.
    /// </summary>
    /// <param name="email">The email to use in the event detail of the custom
event.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutCustomEmailEvent(string email)
    {
        var eventDetail = new
        {
            UserEmail = email,
            Message = "This event was generated by example code.",
            UtcTime = DateTime.UtcNow.ToString("g")
        };
        var response = await _amazonEventBridge.PutEventsAsync(
            new PutEventsRequest()
            {
                Entries = new List<PutEventsRequestEntry>()
                {
                    new PutEventsRequestEntry()
                    {
                        Source = "ExampleSource",
                        Detail = JsonSerializer.Serialize(eventDetail),
                        DetailType = "ExampleType"
                    }
                }
            });

        return response.FailedEntryCount == 0;
    }
```

- Per i dettagli sull'API, consulta la PutEventssezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/events/EventBridgeClient.h>
#include <aws/events/model/PutEventsRequest.h>
#include <aws/events/model/PutEventsResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Invia un evento.

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;

        Aws::CloudWatchEvents::Model::PutEventsRequestEntry event_entry;
        event_entry.SetDetail(MakeDetails(event_key, event_value));
        event_entry.SetDetailType("sampleSubmitted");
        event_entry.AddResources(resource_arn);
        event_entry.SetSource("aws-sdk-cpp-cloudwatch-example");

        Aws::CloudWatchEvents::Model::PutEventsRequest request;
        request.AddEntries(event_entry);

        auto outcome = cwe.PutEvents(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to post CloudWatch event: " <<
```

```
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully posted CloudWatch event" << std::endl;
        }
```

- Per i dettagli sull'API, consulta la PutEventssezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per inviare un evento personalizzato a CloudWatch Events

Questo esempio invia un evento personalizzato a CloudWatch Events. L'evento è contenuto nel file putevents.json:

```
aws events put-events --entries file://putevents.json
```

Visualizzare il contenuto del file putevents.json:

```
[
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
  },
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value3\", \"key2\": \"value4\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
   }
```

```
]
```

- Per i dettagli sull'API, consulta PutEvents AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void triggerCustomRule(EventBridgeClient eventBrClient, String
email) {
       String json = "{" +
               "\"UserEmail\": \"" + email + "\"," +
               "\"Message\": \"This event was generated by example code.\"," +
               "\"UtcTime\": \"Now.\"" +
               "}";

       PutEventsRequestEntry entry = PutEventsRequestEntry.builder()
               .source("ExampleSource")
               .detail(json)
               .detailType("ExampleType")
               .build();

       PutEventsRequest eventsRequest = PutEventsRequest.builder()
               .entries(entry)
               .build();

       eventBrClient.putEvents(eventsRequest);
   }
```

- Per i dettagli sull'API, consulta la PutEventssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import {
  EventBridgeClient,
  PutEventsCommand,
} from "@aws-sdk/client-eventbridge";

export const putEvents = async (
  source = "eventbridge.integration.test",
  detailType = "greeting",
  resources = [],
) => {
  const client = new EventBridgeClient({});

  const response = await client.send(
    new PutEventsCommand({
      Entries: [
        {
          Detail: JSON.stringify({ greeting: "Hello there." }),
          DetailType: detailType,
          Resources: resources,
          Source: source,
        },
      ],
    }),
  );

  console.log("PutEvents response:");
  console.log(response);
  // PutEvents response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
```

```
  //     requestId: '3d0df73d-dcea-4a23-ae0d-f5556a3ac109',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   Entries: [ { EventId: '51620841-5af4-6402-d9bc-b77734991eb5' } ],
  //   FailedEntryCount: 0
  // }

  return response;
};
```

- Per i dettagli sull'API, consulta la PutEventssezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatchEvents service object
var ebevents = new AWS.EventBridge({ apiVersion: "2015-10-07" });

var params = {
  Entries: [
    {
      Detail: '{ "key1": "value1", "key2": "value2" }',
      DetailType: "appRequestSubmitted",
      Resources: ["RESOURCE_ARN"],
      Source: "com.company.app",
    },
  ],
};
```

```
ebevents.putEvents(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.Entries);
  }
});
```

- Per i dettagli sull'API, consulta la PutEventssezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun triggerCustomRule(email: String) {
    val json = "{" +
        "\"UserEmail\": \"" + email + "\"," +
        "\"Message\": \"This event was generated by example code.\"" +
        "\"UtcTime\": \"Now.\"" +
        "}"

    val entry = PutEventsRequestEntry {
        source = "ExampleSource"
        detail = json
        detailType = "ExampleType"
    }

    val eventsRequest = PutEventsRequest {
        this.entries = listOf(entry)
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
```

```
        eventBrClient.putEvents(eventsRequest)
    }
}
```

- Per i dettagli sull'API, PutEventsconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutRule`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutRule`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Creazione e attivazione di una regola
- Nozioni di base su regole e destinazioni

.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.

Crea una regola che si attiva quando un oggetto viene aggiunto a un bucket di Amazon Simple Storage Service.

```
    /// <summary>
    /// Create a new event rule that triggers when an Amazon S3 object is created
in a bucket.
    /// </summary>
    /// <param name="roleArn">The ARN of the role.</param>
```

```
    /// <param name="ruleName">The name to give the rule.</param>
    /// <param name="bucketName">The name of the bucket to trigger the event.</
param>
    /// <returns>The ARN of the new rule.</returns>
    public async Task<string> PutS3UploadRule(string roleArn, string ruleName,
 string bucketName)
    {
        string eventPattern = "{" +
                               "\"source\": [\"aws.s3\"]," +
                                   "\"detail-type\": [\"Object Created\"]," +
                                   "\"detail\": {" +
                                       "\"bucket\": {" +
                                           "\"name\": [\"" + bucketName + "\"]"
 +
                                       "}" +
                                   "}" +
                               "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Example S3 upload rule for EventBridge",
                RoleArn = roleArn,
                EventPattern = eventPattern
            });

        return response.RuleArn;
    }
```

Crea una regola che utilizza un modello personalizzato.

```
    /// <summary>
    /// Update a rule to use a custom defined event pattern.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <returns>The ARN of the updated rule.</returns>
    public async Task<string> UpdateCustomEventPattern(string ruleName)
    {
        string customEventsPattern = "{" +
                                     "\"source\": [\"ExampleSource\"]," +
                                     "\"detail-type\": [\"ExampleType\"]" +
```

```
                                    "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Custom test rule",
                EventPattern = customEventsPattern
            });

        return response.RuleArn;
    }
```

- Per i dettagli sull'API, consulta la PutRulesezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/events/EventBridgeClient.h>
#include <aws/events/model/PutRuleRequest.h>
#include <aws/events/model/PutRuleResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Crea la regola.

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;
        Aws::CloudWatchEvents::Model::PutRuleRequest request;
        request.SetName(rule_name);
```

```
        request.SetRoleArn(role_arn);
        request.SetScheduleExpression("rate(5 minutes)");
        request.SetState(Aws::CloudWatchEvents::Model::RuleState::ENABLED);

        auto outcome = cwe.PutRule(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch events rule " <<
                rule_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch events rule " <<
                rule_name << " with resulting Arn " <<
                outcome.GetResult().GetRuleArn() << std::endl;
        }
```

- Per i dettagli sull'API, consulta la PutRulesezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per creare regole relative CloudWatch agli eventi

In questo esempio viene creata una regola attivata ogni giorno alle 9:00 UTC. Se usi put-targets per aggiungere una funzione Lambda come destinazione di questa regola, puoi eseguire la funzione Lambda ogni giorno all'ora specificata:

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9
 * * ? *)"
```

L'esempio seguente crea una regola che viene attivata quando lo stato di qualsiasi istanza EC2 nella regione cambia:

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"source
\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change Notification\"]}"
 --role-arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

In questo esempio viene creata una regola che si attiva quando un'istanza EC2 nella regione viene bloccata o terminata:

```
aws events put-rule --name "EC2InstanceStateChangeStopOrTerminate" --event-
pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change
 Notification\"],\"detail\":{\"state\":[\"stopped\",\"terminated\"]}}" --role-arn
 "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

- Per i dettagli sull'API, consulta PutRule AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea una regola pianificata.

```
   public static void createEBRule(EventBridgeClient eventBrClient, String
ruleName, String cronExpression) {
       try {
           PutRuleRequest ruleRequest = PutRuleRequest.builder()
                   .name(ruleName)
                   .eventBusName("default")
                   .scheduleExpression(cronExpression)
                   .state("ENABLED")
                   .description("A test rule that runs on a schedule created by
the Java API")
                   .build();

           PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
           System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

       } catch (EventBridgeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
```

```
        }
    }
```

Crea una regola che si attiva quando un oggetto viene aggiunto a un bucket di Amazon Simple Storage Service.

```
    // Create a new event rule that triggers when an Amazon S3 object is created
in
    // a bucket.
    public static void addEventRule(EventBridgeClient eventBrClient, String
roleArn, String bucketName,
            String eventRuleName) {
        String pattern = "{\n" +
                "  \"source\": [\"aws.s3\"],\n" +
                "  \"detail-type\": [\"Object Created\"],\n" +
                "  \"detail\": {\n" +
                "    \"bucket\": {\n" +
                "      \"name\": [\"" + bucketName + "\"]\n" +
                "    }\n" +
                "  }\n" +
                "}";

        try {
            PutRuleRequest ruleRequest = PutRuleRequest.builder()
                    .description("Created by using the AWS SDK for Java v2")
                    .name(eventRuleName)
                    .eventPattern(pattern)
                    .roleArn(roleArn)
                    .build();

            PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
            System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta la PutRulesezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import { EventBridgeClient, PutRuleCommand } from "@aws-sdk/client-eventbridge";

export const putRule = async (
  ruleName = "some-rule",
  source = "some-source",
) => {
  const client = new EventBridgeClient({});

  const response = await client.send(
    new PutRuleCommand({
      Name: ruleName,
      EventPattern: JSON.stringify({ source: [source] }),
      State: "ENABLED",
      EventBusName: "default",
    }),
  );

  console.log("PutRule response:");
  console.log(response);
  // PutRule response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: 'd7292ced-1544-421b-842f-596326bc7072',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   RuleArn: 'arn:aws:events:us-east-1:xxxxxxxxxxxx:rule/
EventBridgeTestRule-1696280037720'
```

```
  // }
  return response;
};
```

- Per i dettagli sull'API, consulta la PutRulesezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatchEvents service object
var ebevents = new AWS.EventBridge({ apiVersion: "2015-10-07" });

var params = {
  Name: "DEMO_EVENT",
  RoleArn: "IAM_ROLE_ARN",
  ScheduleExpression: "rate(5 minutes)",
  State: "ENABLED",
};

ebevents.putRule(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.RuleArn);
  }
});
```

- Per i dettagli sull'API, consulta la PutRulesezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea una regola pianificata.

```
suspend fun createScRule(ruleName: String?, cronExpression: String?) {
    val ruleRequest = PutRuleRequest {
        name = ruleName
        eventBusName = "default"
        scheduleExpression = cronExpression
        state = RuleState.Enabled
        description = "A test rule that runs on a schedule created by the Kotlin
 API"
    }

    EventBridgeClient { region = "us-west-2" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}
```

Crea una regola che si attiva quando un oggetto viene aggiunto a un bucket di Amazon Simple Storage Service.

```
// Create a new event rule that triggers when an Amazon S3 object is created in a
 bucket.
suspend fun addEventRule(roleArnVal: String?, bucketName: String, eventRuleName:
 String?) {
    val pattern = """{
        "source": ["aws.s3"],
        "detail-type": ["Object Created"],
        "detail": {
        "bucket": {
            "name": ["$bucketName"]
```

```
            }
        }
    }"""

    val ruleRequest = PutRuleRequest {
        description = "Created by using the AWS SDK for Kotlin"
        name = eventRuleName
        eventPattern = pattern
        roleArn = roleArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}
```

- Per i dettagli sull'API, PutRuleconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutTargets`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutTargets`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base su regole e destinazioni

.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.

Aggiungi un argomento Amazon SNS come destinazione per una regola.

```
    /// <summary>
    /// Add an Amazon SNS target topic to a rule.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <param name="targetArn">The ARN of the Amazon SNS target.</param>
    /// <param name="eventBusArn">The optional event bus name, uses default if
empty.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> AddSnsTargetToRule(string ruleName, string
targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        // Create the list of targets and add a new target.
        var targets = new List<Target>
        {
            new Target()
            {
                Arn = targetArn,
                Id = targetID
            }
        };

        // Add the targets to the rule.
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
```

```
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }

        return targetID;
    }
```

Aggiungi un trasformatore di input a una destinazione per una regola.

```
    /// <summary>
    /// Update an Amazon S3 object created rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
default event bus.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateS3UploadRuleTargetWithTransform(string
ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        var targets = new List<Target>
        {
            new Target()
            {
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputPathsMap = new Dictionary<string, string>()
                    {
                        {"bucket", "$.detail.bucket.name"},
                        {"time", "$.time"}
                    },
```

```
                    InputTemplate = "\"Notification: an object was uploaded to
bucket <bucket> at <time>.\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }
        return targetID;
    }
```

- Per i dettagli sull'API, consulta la PutTargetssezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/events/EventBridgeClient.h>
#include <aws/events/model/PutTargetsRequest.h>
```

```
#include <aws/events/model/PutTargetsResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Aggiungi la destinazione.

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;

        Aws::CloudWatchEvents::Model::Target target;
        target.SetArn(lambda_arn);
        target.SetId(target_id);

        Aws::CloudWatchEvents::Model::PutTargetsRequest request;
        request.SetRule(rule_name);
        request.AddTargets(target);

        auto putTargetsOutcome = cwe.PutTargets(request);
        if (!putTargetsOutcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch events target for rule "
                << rule_name << ": " <<
                putTargetsOutcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout <<
                "Successfully created CloudWatch events target for rule "
                << rule_name << std::endl;
        }
```

- Per i dettagli sull'API, consulta la PutTargetssezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per aggiungere obiettivi per le regole CloudWatch degli eventi

Nell'esempio seguente viene aggiunta una funzione Lambda come destinazione di una regola:

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="1","Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

Questo esempio imposta un flusso Amazon Kinesis come destinazione, in modo che gli eventi rilevati da questa regola vengano inoltrati allo stream:

```
aws events put-targets --rule EC2InstanceStateChanges --targets
 "Id"="1","Arn"="arn:aws:kinesis:us-east-1:123456789012:stream/
MyStream","RoleArn"="arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

Questo esempio imposta due flussi Amazon Kinesis come destinazione per una regola:

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="Target1","Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream1","RoleArn"="arn:aws:iam::379642911888:role/ MyRoleToAccessLambda"
  "Id"="Target2"," Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream2","RoleArn"="arn:aws:iam::379642911888:role/MyRoleToAccessLambda"
```

- Per i dettagli sull'API, consulta PutTargets AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Aggiungi un argomento Amazon SNS come destinazione per una regola.

```
    // Add a rule which triggers an SNS target when a file is uploaded to an S3
    // bucket.
    public static void addSnsEventRule(EventBridgeClient eventBrClient, String
 ruleName, String topicArn,
            String topicName, String eventRuleName, String bucketName) {
        String targetID = java.util.UUID.randomUUID().toString();
        Target myTarget = Target.builder()
```

```
                .id(targetID)
                .arn(topicArn)
                .build();

        List<Target> targets = new ArrayList<>();
        targets.add(myTarget);
        PutTargetsRequest request = PutTargetsRequest.builder()
                .eventBusName(null)
                .targets(targets)
                .rule(ruleName)
                .build();

        eventBrClient.putTargets(request);
        System.out.println("Added event rule " + eventRuleName + " with Amazon
SNS target " + topicName + " for bucket "
                + bucketName + ".");
    }
```

Aggiungi un trasformatore di input a una destinazione per una regola.

```
    public static void updateCustomRuleTargetWithTransform(EventBridgeClient
eventBrClient, String topicArn,
            String ruleName) {
        String targetId = java.util.UUID.randomUUID().toString();
        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: sample event was received.\"")
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
                    .targets(target)
                    .eventBusName(null)
                    .build();

            eventBrClient.putTargets(targetsRequest);
```

```
        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta la PutTargetssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import {
  EventBridgeClient,
  PutTargetsCommand,
} from "@aws-sdk/client-eventbridge";

export const putTarget = async (
  existingRuleName = "some-rule",
  targetArn = "arn:aws:lambda:us-east-1:000000000000:function:test-func",
  uniqueId = Date.now().toString(),
) => {
  const client = new EventBridgeClient({});
  const response = await client.send(
    new PutTargetsCommand({
      Rule: existingRuleName,
      Targets: [
        {
          Arn: targetArn,
          Id: uniqueId,
        },
      ],
    }),
```

```
  );

  console.log("PutTargets response:");
  console.log(response);
  // PutTargets response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: 'f5b23b9a-2c17-45c1-ad5c-f926c3692e3d',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   FailedEntries: [],
  //   FailedEntryCount: 0
  // }

  return response;
};
```

- Per i dettagli sull'API, consulta la PutTargetssezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatchEvents service object
var ebevents = new AWS.EventBridge({ apiVersion: "2015-10-07" });

var params = {
  Rule: "DEMO_EVENT",
```

```
  Targets: [
    {
      Arn: "LAMBDA_FUNCTION_ARN",
      Id: "myEventBridgeTarget",
    },
  ],
};

ebevents.putTargets(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per i dettagli sull'API, consulta la PutTargetssezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Add a rule that triggers an SNS target when a file is uploaded to an S3
 bucket.
suspend fun addSnsEventRule(ruleName: String?, topicArn: String?, topicName:
 String, eventRuleName: String, bucketName: String) {
    val targetID = UUID.randomUUID().toString()
    val myTarget = Target {
        id = targetID
        arn = topicArn
    }

    val targetsOb = mutableListOf<Target>()
```

```
    targetsOb.add(myTarget)

    val request = PutTargetsRequest {
        eventBusName = null
        targets = targetsOb
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(request)
        println("Added event rule $eventRuleName with Amazon SNS target
 $topicName for bucket $bucketName.")
    }
}
```

Aggiungi un trasformatore di input a una destinazione per una regola.

```
suspend fun updateCustomRuleTargetWithTransform(topicArn: String?, ruleName:
 String?) {
    val targetId = UUID.randomUUID().toString()

    val inputTransformerOb = InputTransformer {
        inputTemplate = "\"Notification: sample event was received.\""
    }

    val target = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransformerOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
        targets = listOf(target)
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(targetsRequest)
    }
}
```

- Per i dettagli sull'API, PutTargetsconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`RemoveTargets`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`RemoveTargets`.

.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.

Rimuovi tutte le destinazioni di una regola utilizzando il nome della stessa.

```
    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> RemoveAllTargetsFromRule(string ruleName)
    {
        var targetIds = new List<string>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse targetsResponse;
        do
        {
            targetsResponse = await
_amazonEventBridge.ListTargetsByRuleAsync(request);
            targetIds.AddRange(targetsResponse.Targets.Select(t => t.Id));
            request.NextToken = targetsResponse.NextToken;
```

```
        } while (targetsResponse.NextToken is not null);

        var removeResponse = await _amazonEventBridge.RemoveTargetsAsync(
            new RemoveTargetsRequest()
            {
                Rule = ruleName,
                Ids = targetIds
            });

        if (removeResponse.FailedEntryCount > 0)
        {
            removeResponse.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to remove target {e.TargetId}: {e.ErrorMessage},
code {e.ErrorCode}");
            });
        }

        return removeResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la RemoveTargetssezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per rimuovere una destinazione per un evento

Questo esempio rimuove lo stream Amazon Kinesis denominato MyStream 1 dall'obiettivo della regola. DailyLambdaFunction Quando DailyLambdaFunction è stato creato, questo flusso è stato impostato come destinazione con un ID Target1:

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- Per i dettagli sull'API, consulta AWS CLI Command RemoveTargetsReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Rimuovi tutte le destinazioni di una regola utilizzando il nome della stessa.

```
   public static void deleteTargetsFromRule(EventBridgeClient eventBrClient,
String eventRuleName) {
       // First, get all targets that will be deleted.
       ListTargetsByRuleRequest request = ListTargetsByRuleRequest.builder()
               .rule(eventRuleName)
               .build();

       ListTargetsByRuleResponse response =
eventBrClient.listTargetsByRule(request);
       List<Target> allTargets = response.targets();

       // Get all targets and delete them.
       for (Target myTarget : allTargets) {
           RemoveTargetsRequest removeTargetsRequest =
RemoveTargetsRequest.builder()
                   .rule(eventRuleName)
                   .ids(myTarget.id())
                   .build();

           eventBrClient.removeTargets(removeTargetsRequest);
           System.out.println("Successfully removed the target");
       }
   }
```

- Per i dettagli sull'API, consulta la RemoveTargetssezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteTargetsFromRule(eventRuleName: String?) {
    // First, get all targets that will be deleted.
    val request = ListTargetsByRuleRequest {
        rule = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(request)
        val allTargets = response.targets

        // Get all targets and delete them.
        if (allTargets != null) {
            for (myTarget in allTargets) {
                val removeTargetsRequest = RemoveTargetsRequest {
                    rule = eventRuleName
                    ids = listOf(myTarget.id.toString())
                }
                eventBrClient.removeTargets(removeTargetsRequest)
                println("Successfully removed the target")
            }
        }
    }
}
```

- Per i dettagli sull'API, RemoveTargetsconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per l' EventBridge utilizzo AWS degli SDK

I seguenti esempi di codice mostrano come implementare scenari comuni EventBridge con gli AWS SDK. Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno. EventBridge Ogni scenario include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice.

Esempi

- Crea e attiva una regola in Amazon EventBridge utilizzando un AWS SDK
- Inizia a definire EventBridge regole e obiettivi utilizzando un SDK AWS

Crea e attiva una regola in Amazon EventBridge utilizzando un AWS SDK

Il seguente esempio di codice mostra come creare e attivare una regola in Amazon EventBridge.

Ruby

SDK per Ruby

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Chiama le funzioni nell'ordine corretto.

```
require "aws-sdk-sns"
require "aws-sdk-iam"
require "aws-sdk-cloudwatchevents"
require "aws-sdk-ec2"
require "aws-sdk-cloudwatch"
require "aws-sdk-cloudwatchlogs"
require "securerandom"
```

Verifica se l'argomento Amazon Simple Notification Service (Amazon SNS) esiste tra quelli forniti a questa funzione.

```
# Checks whether the specified Amazon SNS
# topic exists among those provided to this function.
# This is a helper function that is called by the topic_exists? function.
#
# @param topics [Array] An array of Aws::SNS::Types::Topic objects.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   sns_client = Aws::SNS::Client.new(region: 'us-east-1')
#   response = sns_client.list_topics
#   if topic_found?(
#     response.topics,
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
#     puts 'Topic found.'
#   end

def topic_found?(topics, topic_arn)
  topics.each do |topic|
    return true if topic.topic_arn == topic_arn
  end
  return false
end
```

Verifica se l'argomento specificato esiste tra quelli disponibili per il chiamante in Amazon SNS.

```
# Checks whether the specified topic exists among those available to the
# caller in Amazon SNS.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   exit 1 unless topic_exists?(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def topic_exists?(sns_client, topic_arn)
  puts "Searching for topic with ARN '#{topic_arn}'..."
  response = sns_client.list_topics
  if response.topics.count.positive?
    if topic_found?(response.topics, topic_arn)
```

```
      puts "Topic found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.topics.count.positive?
        if topic_found?(response.topics, topic_arn)
          puts "Topic found."
          return true
        end
      end
    end
  end
  puts "Topic not found."
  return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end
```

Crea un argomento in Amazon SNS e quindi registrati con un indirizzo e-mail per ricevere notifiche su quell'argomento.

```
# Creates a topic in Amazon SNS
# and then subscribes an email address to receive notifications to that topic.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_name [String] The name of the topic to create.
# @param email_address [String] The email address of the recipient to notify.
# @return [String] The ARN of the topic that was created.
# @example
#   puts create_topic(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-topic',
#     'mary@example.com'
#   )
def create_topic(sns_client, topic_name, email_address)
  puts "Creating the topic named '#{topic_name}'..."
  topic_response = sns_client.create_topic(name: topic_name)
  puts "Topic created with ARN '#{topic_response.topic_arn}'."
  subscription_response = sns_client.subscribe(
    topic_arn: topic_response.topic_arn,
```

```
    protocol: "email",
    endpoint: email_address,
    return_subscription_arn: true
  )
  puts "Subscription created with ARN " \
    "'#{subscription_response.subscription_arn}'. Have the owner of the " \
    "email address '#{email_address}' check their inbox in a few minutes " \
    "and confirm the subscription to start receiving notification emails."
  return topic_response.topic_arn
rescue StandardError => e
  puts "Error creating or subscribing to topic: #{e.message}"
  return "Error"
end
```

Verifica se il ruolo specificato AWS Identity and Access Management (IAM) esiste tra quelli forniti a questa funzione.

```
# Checks whether the specified AWS Identity and Access Management (IAM)
# role exists among those provided to this function.
# This is a helper function that is called by the role_exists? function.
#
# @param roles [Array] An array of Aws::IAM::Role objects.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-east-1')
#   response = iam_client.list_roles
#   if role_found?(
#     response.roles,
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
#     puts 'Role found.'
#   end
def role_found?(roles, role_arn)
  roles.each do |role|
    return true if role.arn == role_arn
  end
  return false
end
```

Verifica se il ruolo specificato esiste tra quelli disponibili per il chiamante in IAM.

```
# Checks whether the specified role exists among those available to the
# caller in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   exit 1 unless role_exists?(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
def role_exists?(iam_client, role_arn)
  puts "Searching for role with ARN '#{role_arn}'..."
  response = iam_client.list_roles
  if response.roles.count.positive?
    if role_found?(response.roles, role_arn)
      puts "Role found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.roles.count.positive?
        if role_found?(response.roles, role_arn)
          puts "Role found."
          return true
        end
      end
    end
  end
  puts "Role not found."
  return false
rescue StandardError => e
  puts "Role not found: #{e.message}"
  return false
end
```

Crea un ruolo in IAM.

```
# Creates a role in AWS Identity and Access Management (IAM).
# This role is used by a rule in Amazon EventBridge to allow
# that rule to operate within the caller's account.
# This role is designed to be used specifically by this code example.
```

```
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to create.
# @return [String] The ARN of the role that was created.
# @example
#   puts create_role(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def create_role(iam_client, role_name)
  puts "Creating the role named '#{role_name}'..."
  response = iam_client.create_role(
    assume_role_policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "",
          'Effect': "Allow",
          'Principal': {
            'Service': "events.amazonaws.com"
          },
          'Action': "sts:AssumeRole"
        }
      ]
    }.to_json,
    path: "/",
    role_name: role_name
  )
  puts "Role created with ARN '#{response.role.arn}'."
  puts "Adding access policy to role..."
  iam_client.put_role_policy(
    policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "CloudWatchEventsFullAccess",
          'Effect': "Allow",
          'Resource': "*",
          'Action': "events:*"
        },
        {
          'Sid': "IAMPassRoleForCloudWatchEvents",
          'Effect': "Allow",
          'Resource': "arn:aws:iam::*:role/AWS_Events_Invoke_Targets",
```

```
          'Action': "iam:PassRole"
        }
      ]
    }.to_json,
    policy_name: "CloudWatchEventsPolicy",
    role_name: role_name
  )
  puts "Access policy added to role."
  return response.role.arn
rescue StandardError => e
  puts "Error creating role or adding policy to it: #{e.message}"
  puts "If the role was created, you must add the access policy " \
    "to the role yourself, or delete the role yourself and try again."
  return "Error"
end
```

Verifica se la EventBridge regola specificata esiste tra quelle fornite a questa funzione.

```
# Checks whether the specified Amazon EventBridge rule exists among
# those provided to this function.
# This is a helper function that is called by the rule_exists? function.
#
# @param rules [Array] An array of Aws::CloudWatchEvents::Types::Rule objects.
# @param rule_arn [String] The name of the rule to find.
# @return [Boolean] true if the name of the rule was found; otherwise, false.
# @example
#   cloudwatchevents_client = Aws::CloudWatch::Client.new(region: 'us-east-1')
#   response = cloudwatchevents_client.list_rules
#   if rule_found?(response.rules, 'aws-doc-sdk-examples-ec2-state-change')
#     puts 'Rule found.'
#   end
def rule_found?(rules, rule_name)
  rules.each do |rule|
    return true if rule.name == rule_name
  end
  return false
end
```

Verifica se la regola specificata esiste tra quelle disponibili per il chiamante in EventBridge.

```
# Checks whether the specified rule exists among those available to the
```

```
# caller in Amazon EventBridge.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to find.
# @return [Boolean] true if the rule name was found; otherwise, false.
# @example
#   exit 1 unless rule_exists?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1')
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def rule_exists?(cloudwatchevents_client, rule_name)
  puts "Searching for rule with name '#{rule_name}'..."
  response = cloudwatchevents_client.list_rules
  if response.rules.count.positive?
    if rule_found?(response.rules, rule_name)
      puts "Rule found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.rules.count.positive?
        if rule_found?(response.rules, rule_name)
          puts "Rule found."
          return true
        end
      end
    end
  end
  puts "Rule not found."
  return false
rescue StandardError => e
  puts "Rule not found: #{e.message}"
  return false
end
```

Crea una regola in EventBridge.

```
# Creates a rule in Amazon EventBridge.
# This rule is triggered whenever an available instance in
# Amazon EC2 changes to the specified state.
# This rule is designed to be used specifically by this code example.
```

```
#
# Prerequisites:
#
# - A role in AWS Identity and Access Management (IAM) that is designed
#   to be used specifically by this code example.
# - A topic in Amazon SNS.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to create.
# @param rule_description [String] Some description for this rule.
# @param instance_state [String] The state that available instances in
#   Amazon EC2 must change to, to
#   trigger this rule.
# @param role_arn [String] The Amazon Resource Name (ARN) of the IAM role.
# @param target_id [String] Some identifying string for the rule's target.
# @param topic_arn [String] The ARN of the Amazon SNS topic.
# @return [Boolean] true if the rule was created; otherwise, false.
# @example
#   exit 1 unless rule_created?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     'Triggers when any available EC2 instance starts.',
#     'running',
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change',
#     'sns-topic',
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def rule_created?(
  cloudwatchevents_client,
  rule_name,
  rule_description,
  instance_state,
  role_arn,
  target_id,
  topic_arn
)
  puts "Creating rule with name '#{rule_name}'..."
  put_rule_response = cloudwatchevents_client.put_rule(
    name: rule_name,
    description: rule_description,
    event_pattern: {
      'source': [
        "aws.ec2"
```

```
      ],
      'detail-type': [
        "EC2 Instance State-change Notification"
      ],
      'detail': {
        'state': [
          instance_state
        ]
      }
    }.to_json,
    state: "ENABLED",
    role_arn: role_arn
  )
  puts "Rule created with ARN '#{put_rule_response.rule_arn}'."

  put_targets_response = cloudwatchevents_client.put_targets(
    rule: rule_name,
    targets: [
      {
        id: target_id,
        arn: topic_arn
      }
    ]
  )
  if put_targets_response.key?(:failed_entry_count) &&
      put_targets_response.failed_entry_count > 0
    puts "Error(s) adding target to rule:"
    put_targets_response.failed_entries.each do |failure|
      puts failure.error_message
    end
    return false
  else
    return true
  end
rescue StandardError => e
  puts "Error creating rule or adding target to rule: #{e.message}"
  puts "If the rule was created, you must add the target " \
    "to the rule yourself, or delete the rule yourself and try again."
  return false
end
```

Verifica se il gruppo di log specificato esiste tra quelli disponibili per il chiamante in Amazon CloudWatch Logs.

```
# Checks to see whether the specified log group exists among those available
# to the caller in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to find.
# @return [Boolean] true if the log group name was found; otherwise, false.
# @example
#   exit 1 unless log_group_exists?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_exists?(cloudwatchlogs_client, log_group_name)
  puts "Searching for log group with name '#{log_group_name}'..."
  response = cloudwatchlogs_client.describe_log_groups(
    log_group_name_prefix: log_group_name
  )
  if response.log_groups.count.positive?
    response.log_groups.each do |log_group|
      if log_group.log_group_name == log_group_name
        puts "Log group found."
        return true
      end
    end
  end
  puts "Log group not found."
  return false
rescue StandardError => e
  puts "Log group not found: #{e.message}"
  return false
end
```

Crea un gruppo di log in CloudWatch Logs.

```
# Creates a log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to create.
```

```
# @return [Boolean] true if the log group name was created; otherwise, false.
# @example
#   exit 1 unless log_group_created?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_created?(cloudwatchlogs_client, log_group_name)
  puts "Attempting to create log group with the name '#{log_group_name}'..."
  cloudwatchlogs_client.create_log_group(log_group_name: log_group_name)
  puts "Log group created."
  return true
rescue StandardError => e
  puts "Error creating log group: #{e.message}"
  return false
end
```

Scrivi un evento in un flusso di log in CloudWatch Logs.

```
# Writes an event to a log stream in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
# - A log stream within the log group.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @param log_stream_name [String] The name of the log stream within
#   the log group.
# @param message [String] The message to write to the log stream.
# @param sequence_token [String] If available, the sequence token from the
#   message that was written immediately before this message. This sequence
#   token is returned by Amazon CloudWatch Logs whenever you programmatically
#   write a message to the log stream.
# @return [String] The sequence token that is returned by
#   Amazon CloudWatch Logs after successfully writing the message to the
#   log stream.
# @example
#   puts log_event(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
```

```
#     '2020/11/19/53f985be-199f-408e-9a45-fc242df41fEX',
#     "Instance 'i-033c48ef067af3dEX' restarted.",
#     '495426724868310740095796045676567882148068632824696073EX'
#   )
def log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  message,
  sequence_token
)
  puts "Attempting to log '#{message}' to log stream '#{log_stream_name}'..."
  event = {
    log_group_name: log_group_name,
    log_stream_name: log_stream_name,
    log_events: [
      {
        timestamp: (Time.now.utc.to_f.round(3) * 1_000).to_i,
        message: message
      }
    ]
  }
  unless sequence_token.empty?
    event[:sequence_token] = sequence_token
  end

  response = cloudwatchlogs_client.put_log_events(event)
  puts "Message logged."
  return response.next_sequence_token
rescue StandardError => e
  puts "Message not logged: #{e.message}"
end
```

Riavvia un'istanza Amazon Elastic Compute Cloud (Amazon EC2) e aggiunge informazioni sull'attività correlata a un flusso di log in Logs. CloudWatch

```
# Restarts an Amazon EC2 instance
# and adds information about the related activity to a log stream
# in Amazon CloudWatch Logs.
#
# Prerequisites:
```

```
#
# - The Amazon EC2 instance to restart.
# - The log group in Amazon CloudWatch Logs to add related activity
#   information to.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client]
#   An initialized Amazon CloudWatch Logs client.
# @param instance_id [String] The ID of the instance.
# @param log_group_name [String] The name of the log group.
# @return [Boolean] true if the instance was restarted and the information
#   was written to the log stream; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX',
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  log_stream_name = "#{Time.now.year}/#{Time.now.month}/#{Time.now.day}/" \
    "#{SecureRandom.uuid}"
  cloudwatchlogs_client.create_log_stream(
    log_group_name: log_group_name,
    log_stream_name: log_stream_name
  )
  sequence_token = ""

  puts "Attempting to stop the instance with the ID '#{instance_id}'. " \
    "This might take a few minutes..."
  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts "Instance stopped."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' stopped.",
    sequence_token
```

```
  )

  puts "Attempting to restart the instance. This might take a few minutes..."
  ec2_client.start_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
  puts "Instance restarted."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' restarted.",
    sequence_token
  )

  return true
rescue StandardError => e
  puts "Error creating log stream or stopping or restarting the instance: " \
    "#{e.message}"
  log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Error stopping or starting instance '#{instance_id}': #{e.message}",
    sequence_token
  )
  return false
end
```

Visualizza informazioni sull'attività per una regola in. EventBridge

```
# Displays information about activity for a rule in Amazon EventBridge.
#
# Prerequisites:
#
# - A rule in Amazon EventBridge.
#
# @param cloudwatch_client [Amazon::CloudWatch::Client] An initialized
#   Amazon CloudWatch client.
# @param rule_name [String] The name of the rule.
# @param start_time [Time] The timestamp that determines the first datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param end_time [Time] The timestamp that determines the last datapoint
```

```
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param period [Integer] The interval, in seconds, to check for activity.
# @example
#   display_rule_activity(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     Time.now - 600, # Start checking from 10 minutes ago.
#     Time.now, # Check up until now.
#     60 # Check every minute during those 10 minutes.
#   )
def display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)
  puts "Attempting to display rule activity..."
  response = cloudwatch_client.get_metric_statistics(
    namespace: "AWS/Events",
    metric_name: "Invocations",
    dimensions: [
      {
        name: "RuleName",
        value: rule_name
      }
    ],
    start_time: start_time,
    end_time: end_time,
    period: period,
    statistics: ["Sum"],
    unit: "Count"
  )

  if response.key?(:datapoints) && response.datapoints.count.positive?
    puts "The event rule '#{rule_name}' was triggered:"
    response.datapoints.each do |datapoint|
      puts "  #{datapoint.sum} time(s) at #{datapoint.timestamp}"
    end
  else
    puts "The event rule '#{rule_name}' was not triggered during the " \
      "specified time period."
  end
rescue StandardError => e
```

```
  puts "Error getting information about event rule activity: #{e.message}"
end
```

Visualizza le informazioni di registro per tutti i flussi di log in un gruppo di log CloudWatch Logs.

```
# Displays log information for all of the log streams in a log group in
# Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Amazon::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @example
#   display_log_data(
#     Amazon::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def display_log_data(cloudwatchlogs_client, log_group_name)
  puts "Attempting to display log stream data for the log group " \
    "named '#{log_group_name}'..."
  describe_log_streams_response = cloudwatchlogs_client.describe_log_streams(
    log_group_name: log_group_name,
    order_by: "LastEventTime",
    descending: true
  )
  if describe_log_streams_response.key?(:log_streams) &&
      describe_log_streams_response.log_streams.count.positive?
    describe_log_streams_response.log_streams.each do |log_stream|
      get_log_events_response = cloudwatchlogs_client.get_log_events(
        log_group_name: log_group_name,
        log_stream_name: log_stream.log_stream_name
      )
      puts "\nLog messages for '#{log_stream.log_stream_name}':"
      puts "-" * (log_stream.log_stream_name.length + 20)
      if get_log_events_response.key?(:events) &&
          get_log_events_response.events.count.positive?
        get_log_events_response.events.each do |event|
          puts event.message
```

```
        end
      else
        puts "No log messages for this log stream."
      end
    end
  end
rescue StandardError => e
  puts "Error getting information about the log streams or their messages: " \
    "#{e.message}"
end
```

Mostra al chiamante un promemoria affinché pulisca manualmente tutte AWS le risorse associate che non gli servono più.

```
# Displays a reminder to the caller to manually clean up any associated
# AWS resources that they no longer need.
#
# @param topic_name [String] The name of the Amazon SNS topic.
# @param role_name [String] The name of the IAM role.
# @param rule_name [String] The name of the Amazon EventBridge rule.
# @param log_group_name [String] The name of the Amazon CloudWatch Logs log
 group.
# @param instance_id [String] The ID of the Amazon EC2 instance.
# @example
#   manual_cleanup_notice(
#     'aws-doc-sdk-examples-topic',
#     'aws-doc-sdk-examples-cloudwatch-events-rule-role',
#     'aws-doc-sdk-examples-ec2-state-change',
#     'aws-doc-sdk-examples-cloudwatch-log',
#     'i-033c48ef067af3dEX'
#   )
def manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
  puts "-" * 10
  puts "Some of the following AWS resources might still exist in your account."
  puts "If you no longer want to use this code example, then to clean up"
  puts "your AWS account and avoid unexpected costs, you might want to"
  puts "manually delete any of the following resources if they exist:"
  puts "- The Amazon SNS topic named '#{topic_name}'."
  puts "- The IAM role named '#{role_name}'."
```

```
  puts "- The Amazon EventBridge rule named '#{rule_name}'."
  puts "- The Amazon CloudWatch Logs log group named '#{log_group_name}'."
  puts "- The Amazon EC2 instance with the ID '#{instance_id}'."
end

# Example usage:
def run_me
  # Properties for the Amazon SNS topic.
  topic_name = "aws-doc-sdk-examples-topic"
  email_address = "mary@example.com"
  # Properties for the IAM role.
  role_name = "aws-doc-sdk-examples-cloudwatch-events-rule-role"
  # Properties for the Amazon EventBridge rule.
  rule_name = "aws-doc-sdk-examples-ec2-state-change"
  rule_description = "Triggers when any available EC2 instance starts."
  instance_state = "running"
  target_id = "sns-topic"
  # Properties for the Amazon EC2 instance.
  instance_id = "i-033c48ef067af3dEX"
  # Properties for displaying the event rule's activity.
  start_time = Time.now - 600 # Go back over the past 10 minutes
                              # (10 minutes * 60 seconds = 600 seconds).
  end_time = Time.now
  period = 60 # Look back every 60 seconds over the past 10 minutes.
  # Properties for the Amazon CloudWatch Logs log group.
  log_group_name = "aws-doc-sdk-examples-cloudwatch-log"
  # AWS service clients for this code example.
  region = "us-east-1"
  sts_client = Aws::STS::Client.new(region: region)
  sns_client = Aws::SNS::Client.new(region: region)
  iam_client = Aws::IAM::Client.new(region: region)
  cloudwatchevents_client = Aws::CloudWatchEvents::Client.new(region: region)
  ec2_client = Aws::EC2::Client.new(region: region)
  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  cloudwatchlogs_client = Aws::CloudWatchLogs::Client.new(region: region)

  # Get the caller's account ID for use in forming
  # Amazon Resource Names (ARNs) that this code relies on later.
  account_id = sts_client.get_caller_identity.account

  # If the Amazon SNS topic doesn't exist, create it.
  topic_arn = "arn:aws:sns:#{region}:#{account_id}:#{topic_name}"
  unless topic_exists?(sns_client, topic_arn)
    topic_arn = create_topic(sns_client, topic_name, email_address)
```

```
  if topic_arn == "Error"
    puts "Could not create the Amazon SNS topic correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
    exit 1
  end
end

# If the IAM role doesn't exist, create it.
role_arn = "arn:aws:iam::#{account_id}:role/#{role_name}"
unless role_exists?(iam_client, role_arn)
  role_arn = create_role(iam_client, role_name)
  if role_arn == "Error"
    puts "Could not create the IAM role correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon EventBridge rule doesn't exist, create it.
unless rule_exists?(cloudwatchevents_client, rule_name)
  unless rule_created?(
    cloudwatchevents_client,
    rule_name,
    rule_description,
    instance_state,
    role_arn,
    target_id,
    topic_arn
  )
    puts "Could not create the Amazon EventBridge rule correctly. " \
      "Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon CloudWatch Logs log group doesn't exist, create it.
unless log_group_exists?(cloudwatchlogs_client, log_group_name)
  unless log_group_created?(cloudwatchlogs_client, log_group_name)
    puts "Could not create the Amazon CloudWatch Logs log group " \
```

```
      "correctly. Program stopped."
      manual_cleanup_notice(
        topic_name, role_name, rule_name, log_group_name, instance_id
      )
    end
  end

  # Restart the Amazon EC2 instance, which triggers the rule.
  unless instance_restarted?(
    ec2_client,
    cloudwatchlogs_client,
    instance_id,
    log_group_name
  )
    puts "Could not restart the instance to trigger the rule. " \
      "Continuing anyway to show information about the rule and logs..."
  end

  # Display how many times the rule was triggered over the past 10 minutes.
  display_rule_activity(
    cloudwatch_client,
    rule_name,
    start_time,
    end_time,
    period
  )

  # Display related log data in Amazon CloudWatch Logs.
  display_log_data(cloudwatchlogs_client, log_group_name)

  # Reminder the caller to clean up any AWS resources that are used
  # by this code example and are no longer needed.
  manual_cleanup_notice(
    topic_name, role_name, rule_name, log_group_name, instance_id
  )
end

run_me if $PROGRAM_NAME == __FILE__
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Ruby .
  - PutEvents

- PutRule

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Inizia a definire EventBridge regole e obiettivi utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Creare una regola e aggiungervi una destinazione.
- Abilitare e disabilitare regole.
- Elencare e aggiornare regole e destinazioni.
- Inviare eventi e quindi eliminare le risorse.

.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.

Esegui uno scenario interattivo al prompt dei comandi.

```
public class EventBridgeScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    This .NET example performs the following tasks with Amazon EventBridge:
    - Create a rule.
    - Add a target to a rule.
    - Enable and disable rules.
    - List rules and targets.
    - Update rules and targets.
```

```
    - Send events.
    - Delete the rule.
    */

    private static ILogger logger = null!;
    private static EventBridgeWrapper _eventBridgeWrapper = null!;
    private static IConfiguration _configuration = null!;

    private static IAmazonIdentityManagementService? _iamClient = null!;
    private static IAmazonSimpleNotificationService? _snsClient = null!;
    private static IAmazonS3 _s3Client = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for Amazon EventBridge.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonEventBridge>()
            .AddAWSService<IAmazonIdentityManagementService>()
            .AddAWSService<IAmazonS3>()
            .AddAWSService<IAmazonSimpleNotificationService>()
            .AddTransient<EventBridgeWrapper>()
            )
            .Build();

        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally, load local settings.
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<EventBridgeScenario>();

        ServicesSetup(host);

        string topicArn = "";
```

```
        string roleArn = "";

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Amazon EventBridge example scenario.");
        Console.WriteLine(new string('-', 80));

        try
        {
            roleArn = await CreateRole();

            await CreateBucketWithEventBridgeEvents();

            await AddEventRule(roleArn);

            await ListEventRules();

            topicArn = await CreateSnsTopic();

            var email = await SubscribeToSnsTopic(topicArn);

            await AddSnsTarget(topicArn);

            await ListTargets();

            await ListRulesForTarget(topicArn);

            await UploadS3File(_s3Client);

            await ChangeRuleState(false);

            await GetRuleState();

            await UpdateSnsEventRule(topicArn);

            await ChangeRuleState(true);

            await UploadS3File(_s3Client);

            await UpdateToCustomRule(topicArn);

            await TriggerCustomRule(email);

            await CleanupResources(topicArn);
        }
```

```
        catch (Exception ex)
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
            await CleanupResources(topicArn);
        }
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("The Amazon EventBridge example scenario is
complete.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Populate the services for use within the console application.
    /// </summary>
    /// <param name="host">The services host.</param>
    private static void ServicesSetup(IHost host)
    {
        _eventBridgeWrapper =
host.Services.GetRequiredService<EventBridgeWrapper>();
        _snsClient =
host.Services.GetRequiredService<IAmazonSimpleNotificationService>();
        _s3Client = host.Services.GetRequiredService<IAmazonS3>();
        _iamClient =
host.Services.GetRequiredService<IAmazonIdentityManagementService>();
    }

    /// <summary>
    /// Create a role to be used by EventBridge.
    /// </summary>
    /// <returns>The role Amazon Resource Name (ARN).</returns>
    public static async Task<string> CreateRole()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Creating a role to use with EventBridge and attaching
managed policy AmazonEventBridgeFullAccess.");
        Console.WriteLine(new string('-', 80));

        var roleName = _configuration["roleName"];

        var assumeRolePolicy = "{" +
                               "\"Version\": \"2012-10-17\"," +
                               "\"Statement\": [{" +
                               "\"Effect\": \"Allow\"," +
                               "\"Principal\": {" +
```

```
                                $"\"Service\": \"events.amazonaws.com\"" +
                                "}," +
                                "\"Action\": \"sts:AssumeRole\"" +
                                "}]" +
                                "}";

        var roleResult = await _iamClient!.CreateRoleAsync(
            new CreateRoleRequest()
            {
                AssumeRolePolicyDocument = assumeRolePolicy,
                Path = "/",
                RoleName = roleName
            });

        await _iamClient.AttachRolePolicyAsync(
            new AttachRolePolicyRequest()
            {
                PolicyArn = "arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess",
                RoleName = roleName
            });
        // Allow time for the role to be ready.
        Thread.Sleep(10000);
        return roleResult.Role.Arn;
    }

    /// <summary>
    /// Create an Amazon Simple Storage Service (Amazon S3) bucket with
 EventBridge events enabled.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateBucketWithEventBridgeEvents()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Creating an S3 bucket with EventBridge events
 enabled.");

        var testBucketName = _configuration["testBucketName"];

        var bucketExists = await
 Amazon.S3.Util.AmazonS3Util.DoesS3BucketExistV2Async(_s3Client,
            testBucketName);

        if (!bucketExists)
```

```
        {
            await _s3Client.PutBucketAsync(new PutBucketRequest()
            {
                BucketName = testBucketName,
                UseClientRegion = true
            });
        }

        await _s3Client.PutBucketNotificationAsync(new
PutBucketNotificationRequest()
        {
            BucketName = testBucketName,
            EventBridgeConfiguration = new EventBridgeConfiguration()
        });

        Console.WriteLine($"\tAdded bucket {testBucketName} with EventBridge
events enabled.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create and upload a file to an S3 bucket to trigger an event.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task UploadS3File(IAmazonS3 s3Client)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Uploading a file to the test bucket. This will trigger
a subscription email.");

        var testBucketName = _configuration["testBucketName"];

        var fileName = $"example_upload_{DateTime.UtcNow.Ticks}.txt";

        // Create the file if it does not already exist.
        if (!File.Exists(fileName))
        {
            await using StreamWriter sw = File.CreateText(fileName);
            await sw.WriteLineAsync(
                "This is a sample file for testing uploads.");
        }

        await s3Client.PutObjectAsync(new PutObjectRequest()
```

```
        {
            FilePath = fileName,
            BucketName = testBucketName
        });

        Console.WriteLine($"\tPress Enter to continue.");
        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create an Amazon Simple Notification Service (Amazon SNS) topic to use as
an EventBridge target.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task<string> CreateSnsTopic()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine(
            "Creating an Amazon Simple Notification Service (Amazon SNS) topic
for email subscriptions.");

        var topicName = _configuration["topicName"];

        string topicPolicy = "{" +
                             "\"Version\": \"2012-10-17\"," +
                             "\"Statement\": [{" +
                             "\"Sid\": \"EventBridgePublishTopic\"," +
                             "\"Effect\": \"Allow\"," +
                             "\"Principal\": {" +
                             $"\"Service\": \"events.amazonaws.com\"" +
                             "}," +
                             "\"Resource\": \"*\"," +
                             "\"Action\": \"sns:Publish\"" +
                             "}]" +
                             "}";

        var topicAttributes = new Dictionary<string, string>()
        {
            { "Policy", topicPolicy }
        };
```

```
        var topicResponse = await _snsClient!.CreateTopicAsync(new
CreateTopicRequest()
        {
            Name = topicName,
            Attributes = topicAttributes

        });

        Console.WriteLine($"\tAdded topic {topicName} for email subscriptions.");

        Console.WriteLine(new string('-', 80));

        return topicResponse.TopicArn;
    }

    /// <summary>
    /// Subscribe a user email to an SNS topic.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>The user's email.</returns>
    private static async Task<string> SubscribeToSnsTopic(string topicArn)
    {
        Console.WriteLine(new string('-', 80));


        string email = "";
        while (string.IsNullOrEmpty(email))
        {
            Console.WriteLine("Enter your email to subscribe to the Amazon SNS
topic:");
            email = Console.ReadLine()!;
        }

        var subscriptions = new List<string>();
        var paginatedSubscriptions =
_snsClient!.Paginators.ListSubscriptionsByTopic(
            new ListSubscriptionsByTopicRequest()
            {
                TopicArn = topicArn
            });

        // Get the entire list using the paginator.
        await foreach (var subscription in paginatedSubscriptions.Subscriptions)
        {
```

```
            subscriptions.Add(subscription.Endpoint);
        }

        if (subscriptions.Contains(email))
        {
            Console.WriteLine($"\tYour email is already subscribed.");
            Console.WriteLine(new string('-', 80));
            return email;
        }

        await _snsClient.SubscribeAsync(new SubscribeRequest()
        {
            TopicArn = topicArn,
            Protocol = "email",
            Endpoint = email
        });

        Console.WriteLine($"Use the link in the email you received to confirm
your subscription, then press Enter to continue.");

        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
        return email;
    }

    /// <summary>
    /// Add a rule which triggers when a file is uploaded to an S3 bucket.
    /// </summary>
    /// <param name="roleArn">The ARN of the role used by EventBridge.</param>
    /// <returns>Async task.</returns>
    private static async Task AddEventRule(string roleArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Creating an EventBridge event that sends an email when
an Amazon S3 object is created.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];

        await _eventBridgeWrapper.PutS3UploadRule(roleArn, eventRuleName,
testBucketName);
        Console.WriteLine($"\tAdded event rule {eventRuleName} for bucket
{testBucketName}.");
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add an SNS target to the rule.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>Async task.</returns>
    private static async Task AddSnsTarget(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Adding a target to the rule to that sends an email
when the rule is triggered.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];
        var topicName = _configuration["topicName"];
        await _eventBridgeWrapper.AddSnsTargetToRule(eventRuleName, topicArn);
        Console.WriteLine($"\tAdded event rule {eventRuleName} with Amazon SNS
target {topicName} for bucket {testBucketName}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List the event rules on the default event bus.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListEventRules()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Current event rules:");

        var rules = await _eventBridgeWrapper.ListAllRulesForEventBus();
        rules.ForEach(r => Console.WriteLine($"\tRule: {r.Name} Description:
{r.Description} State: {r.State}"));

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Update the event target to use a transform.
    /// </summary>
```

```
    /// <param name="topicArn">The SNS topic ARN target to update.</param>
    /// <returns>Async task.</returns>
    private static async Task UpdateSnsEventRule(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Let's update the event target with a transform.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];

        await
_eventBridgeWrapper.UpdateS3UploadRuleTargetWithTransform(eventRuleName,
topicArn);
        Console.WriteLine($"\tUpdated event rule {eventRuleName} with Amazon SNS
target {topicArn} for bucket {testBucketName}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Update the rule to use a custom event pattern.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task UpdateToCustomRule(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Updating the event pattern to be triggered by a custom
event instead.");

        var eventRuleName = _configuration["eventRuleName"];

        await _eventBridgeWrapper.UpdateCustomEventPattern(eventRuleName);

        Console.WriteLine($"\tUpdated event rule {eventRuleName} to custom
pattern.");
        await
_eventBridgeWrapper.UpdateCustomRuleTargetWithTransform(eventRuleName,
            topicArn);

        Console.WriteLine($"\tUpdated event target {topicArn}.");

        Console.WriteLine(new string('-', 80));
    }
```

```
    /// <summary>
    /// Send rule events for a custom rule using the user's email address.
    /// </summary>
    /// <param name="email">The email address to include.</param>
    /// <returns>Async task.</returns>
    private static async Task TriggerCustomRule(string email)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Sending an event to trigger the rule. This will
trigger a subscription email.");

        await _eventBridgeWrapper.PutCustomEmailEvent(email);

        Console.WriteLine($"\tEvents have been sent. Press Enter to continue.");
        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List all of the targets for a rule.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListTargets()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("List all of the targets for a particular rule.");

        var eventRuleName = _configuration["eventRuleName"];
        var targets = await
_eventBridgeWrapper.ListAllTargetsOnRule(eventRuleName);
        targets.ForEach(t => Console.WriteLine($"\tTarget: {t.Arn} Id: {t.Id}
Input: {t.Input}"));

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List all of the rules for a particular target.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>Async task.</returns>
    private static async Task ListRulesForTarget(string topicArn)
    {
```

```
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("List all of the rules for a particular target.");

        var rules = await _eventBridgeWrapper.ListAllRuleNamesByTarget(topicArn);
        rules.ForEach(r => Console.WriteLine($"\tRule: {r}"));

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Enable or disable a particular rule.
    /// </summary>
    /// <param name="isEnabled">True to enable the rule, otherwise false.</param>
    /// <returns>Async task.</returns>
    private static async Task ChangeRuleState(bool isEnabled)
    {
        Console.WriteLine(new string('-', 80));
        var eventRuleName = _configuration["eventRuleName"];

        if (!isEnabled)
        {
            Console.WriteLine($"Disabling the rule: {eventRuleName}");
            await _eventBridgeWrapper.DisableRuleByName(eventRuleName);
        }
        else
        {
            Console.WriteLine($"Enabling the rule: {eventRuleName}");
            await _eventBridgeWrapper.EnableRuleByName(eventRuleName);
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get the current state of the rule.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetRuleState()
    {
        Console.WriteLine(new string('-', 80));
        var eventRuleName = _configuration["eventRuleName"];

        var state = await
_eventBridgeWrapper.GetRuleStateByRuleName(eventRuleName);
```

```
        Console.WriteLine($"Rule {eventRuleName} is in current state {state}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Clean up the resources from the scenario.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic to clean up.</param>
    /// <returns>Async task.</returns>
    private static async Task CleanupResources(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"Clean up resources.");

        var eventRuleName = _configuration["eventRuleName"];
        if (GetYesNoResponse($"\tDelete all targets and event rule
{eventRuleName}? (y/n)"))
        {
            Console.WriteLine($"\tRemoving all targets from the event rule.");
            await _eventBridgeWrapper.RemoveAllTargetsFromRule(eventRuleName);

            Console.WriteLine($"\tDeleting event rule.");
            await _eventBridgeWrapper.DeleteRuleByName(eventRuleName);
        }

        var topicName = _configuration["topicName"];
        if (GetYesNoResponse($"\tDelete Amazon SNS subscription topic
{topicName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting topic.");
            await _snsClient!.DeleteTopicAsync(new DeleteTopicRequest()
            {
                TopicArn = topicArn
            });
        }

        var bucketName = _configuration["testBucketName"];
        if (GetYesNoResponse($"\tDelete Amazon S3 bucket {bucketName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting bucket.");
            // Delete all objects in the bucket.
            var deleteList = await _s3Client.ListObjectsV2Async(new
ListObjectsV2Request()
```

```
            {
                BucketName = bucketName
            });
            await _s3Client.DeleteObjectsAsync(new DeleteObjectsRequest()
            {
                BucketName = bucketName,
                Objects = deleteList.S3Objects
                    .Select(o => new KeyVersion { Key = o.Key }).ToList()
            });
            // Now delete the bucket.
            await _s3Client.DeleteBucketAsync(new DeleteBucketRequest()
            {
                BucketName = bucketName
            });
        }

        var roleName = _configuration["roleName"];
        if (GetYesNoResponse($"\tDelete role {roleName}? (y/n)"))
        {
            Console.WriteLine($"\tDetaching policy and deleting role.");

            await _iamClient!.DetachRolePolicyAsync(new DetachRolePolicyRequest()
            {
                RoleName = roleName,
                PolicyArn = "arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess",
            });

            await _iamClient!.DeleteRoleAsync(new DeleteRoleRequest()
            {
                RoleName = roleName
            });
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Helper method to get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
```

```
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null &&
                       ynResponse.Equals("y",
                           StringComparison.InvariantCultureIgnoreCase);
        return response;
    }
}
```

Crea una classe che racchiuda le operazioni. EventBridge

```
/// <summary>
/// Wrapper for Amazon EventBridge operations.
/// </summary>
public class EventBridgeWrapper
{
    private readonly IAmazonEventBridge _amazonEventBridge;
    private readonly ILogger<EventBridgeWrapper> _logger;

    /// <summary>
    /// Constructor for the EventBridge wrapper.
    /// </summary>
    /// <param name="amazonEventBridge">The injected EventBridge client.</param>
    /// <param name="logger">The injected logger for the wrapper.</param>
    public EventBridgeWrapper(IAmazonEventBridge amazonEventBridge,
 ILogger<EventBridgeWrapper> logger)

    {
        _amazonEventBridge = amazonEventBridge;
        _logger = logger;
    }

    /// <summary>
    /// Get the state for a rule by the rule name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="eventBusName">The optional name of the event bus. If empty,
 uses the default event bus.</param>
    /// <returns>The state of the rule.</returns>
```

```
    public async Task<RuleState> GetRuleStateByRuleName(string ruleName, string?
eventBusName = null)
    {
        var ruleResponse = await _amazonEventBridge.DescribeRuleAsync(
            new DescribeRuleRequest()
            {
                Name = ruleName,
                EventBusName = eventBusName
            });
        return ruleResponse.State;
    }

    /// <summary>
    /// Enable a particular rule on an event bus.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableRuleByName(string ruleName)
    {
        var ruleResponse = await _amazonEventBridge.EnableRuleAsync(
            new EnableRuleRequest()
            {
                Name = ruleName
            });
        return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Disable a particular rule on an event bus.
    /// </summary
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableRuleByName(string ruleName)
    {
        var ruleResponse = await _amazonEventBridge.DisableRuleAsync(
            new DisableRuleRequest()
            {
                Name = ruleName
            });
        return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// List the rules on an event bus.
```

```
    /// </summary>
    /// <param name="eventBusArn">The optional ARN of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The list of rules.</returns>
    public async Task<List<Rule>> ListAllRulesForEventBus(string? eventBusArn =
null)
    {
        var results = new List<Rule>();
        var request = new ListRulesRequest()
        {
            EventBusName = eventBusArn
        };
        // Get all of the pages of rules.
        ListRulesResponse response;
        do
        {
            response = await _amazonEventBridge.ListRulesAsync(request);
            results.AddRange(response.Rules);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// List all of the targets matching a rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>The list of targets.</returns>
    public async Task<List<Target>> ListAllTargetsOnRule(string ruleName)
    {
        var results = new List<Target>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse response;
        do
        {
            response = await _amazonEventBridge.ListTargetsByRuleAsync(request);
            results.AddRange(response.Targets);
            request.NextToken = response.NextToken;
```

```
        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// List names of all rules matching a target.
    /// </summary>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <returns>The list of rule names.</returns>
    public async Task<List<string>> ListAllRuleNamesByTarget(string targetArn)
    {
        var results = new List<string>();
        var request = new ListRuleNamesByTargetRequest()
        {
            TargetArn = targetArn
        };
        ListRuleNamesByTargetResponse response;
        do
        {
            response = await
 _amazonEventBridge.ListRuleNamesByTargetAsync(request);
            results.AddRange(response.RuleNames);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// Create a new event rule that triggers when an Amazon S3 object is created
 in a bucket.
    /// </summary>
    /// <param name="roleArn">The ARN of the role.</param>
    /// <param name="ruleName">The name to give the rule.</param>
    /// <param name="bucketName">The name of the bucket to trigger the event.</
param>
    /// <returns>The ARN of the new rule.</returns>
    public async Task<string> PutS3UploadRule(string roleArn, string ruleName,
 string bucketName)
    {
        string eventPattern = "{" +
                              "\"source\": [\"aws.s3\"]," +
```

```
                                    "\"detail-type\": [\"Object Created\"]," +
                                    "\"detail\": {" +
                                        "\"bucket\": {" +
                                            "\"name\": [\"" + bucketName + "\"]"
+
                                        "}" +
                                    "}" +
                                "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Example S3 upload rule for EventBridge",
                RoleArn = roleArn,
                EventPattern = eventPattern
            });

        return response.RuleArn;
    }

    /// <summary>
    /// Update an Amazon S3 object created rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
default event bus.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateS3UploadRuleTargetWithTransform(string
ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        var targets = new List<Target>
        {
            new Target()
            {
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputPathsMap = new Dictionary<string, string>()
                    {
```

```
                        {"bucket", "$.detail.bucket.name"},
                        {"time", "$.time"}
                    },
                    InputTemplate = "\"Notification: an object was uploaded to
bucket <bucket> at <time>.\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }
        return targetID;
    }

    /// <summary>
    /// Update a custom rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
default event bus.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateCustomRuleTargetWithTransform(string
ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        var targets = new List<Target>
        {
            new Target()
            {
```

```
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputTemplate = "\"Notification: sample event was received.
\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
 {e.ErrorCode}");
            });
        }
        return targetID;
    }

    /// <summary>
    /// Add an event to the event bus that includes an email, message, and time.
    /// </summary>
    /// <param name="email">The email to use in the event detail of the custom
 event.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutCustomEmailEvent(string email)
    {
        var eventDetail = new
        {
            UserEmail = email,
            Message = "This event was generated by example code.",
            UtcTime = DateTime.UtcNow.ToString("g")
        };
        var response = await _amazonEventBridge.PutEventsAsync(
            new PutEventsRequest()
```

```
            {
                Entries = new List<PutEventsRequestEntry>()
                {
                    new PutEventsRequestEntry()
                    {
                        Source = "ExampleSource",
                        Detail = JsonSerializer.Serialize(eventDetail),
                        DetailType = "ExampleType"
                    }
                }
            });

        return response.FailedEntryCount == 0;
    }

    /// <summary>
    /// Update a rule to use a custom defined event pattern.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <returns>The ARN of the updated rule.</returns>
    public async Task<string> UpdateCustomEventPattern(string ruleName)
    {
        string customEventsPattern = "{" +
                                     "\"source\": [\"ExampleSource\"]," +
                                     "\"detail-type\": [\"ExampleType\"]" +
                                     "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Custom test rule",
                EventPattern = customEventsPattern
            });

        return response.RuleArn;
    }

    /// <summary>
    /// Add an Amazon SNS target topic to a rule.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <param name="targetArn">The ARN of the Amazon SNS target.</param>
```

```
   /// <param name="eventBusArn">The optional event bus name, uses default if
empty.</param>
   /// <returns>The ID of the target.</returns>
   public async Task<string> AddSnsTargetToRule(string ruleName, string
targetArn, string? eventBusArn = null)
   {
       var targetID = Guid.NewGuid().ToString();

       // Create the list of targets and add a new target.
       var targets = new List<Target>
       {
           new Target()
           {
               Arn = targetArn,
               Id = targetID
           }
       };

       // Add the targets to the rule.
       var response = await _amazonEventBridge.PutTargetsAsync(
           new PutTargetsRequest()
           {
               EventBusName = eventBusArn,
               Rule = ruleName,
               Targets = targets,
           });

       if (response.FailedEntryCount > 0)
       {
           response.FailedEntries.ForEach(e =>
           {
               _logger.LogError(
                   $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
           });
       }

       return targetID;
   }

   /// <summary>
   /// Delete an event rule by name.
   /// </summary>
   /// <param name="ruleName">The name of the event rule.</param>
```

```
    /// <returns>True if successful.</returns>
    public async Task<bool> RemoveAllTargetsFromRule(string ruleName)
    {
        var targetIds = new List<string>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse targetsResponse;
        do
        {
            targetsResponse = await
_amazonEventBridge.ListTargetsByRuleAsync(request);
            targetIds.AddRange(targetsResponse.Targets.Select(t => t.Id));
            request.NextToken = targetsResponse.NextToken;

        } while (targetsResponse.NextToken is not null);

        var removeResponse = await _amazonEventBridge.RemoveTargetsAsync(
            new RemoveTargetsRequest()
            {
                Rule = ruleName,
                Ids = targetIds
            });

        if (removeResponse.FailedEntryCount > 0)
        {
            removeResponse.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to remove target {e.TargetId}: {e.ErrorMessage},
code {e.ErrorCode}");
            });
        }

        return removeResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteRuleByName(string ruleName)
```

```
    {
        var response = await _amazonEventBridge.DeleteRuleAsync(
            new DeleteRuleRequest()
            {
                Name = ruleName
            });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Java

SDK per Java 2.x

> **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.

```
/**
 * 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
 *
 * This Java code example performs the following tasks:
 *
 * This Java V2 example performs the following tasks with Amazon EventBridge:
 *
 * 1. Creates an AWS Identity and Access Management (IAM) role to use with
 * Amazon EventBridge.
 * 2. Amazon Simple Storage Service (Amazon S3) bucket with EventBridge events
 * enabled.
 * 3. Creates a rule that triggers when an object is uploaded to Amazon S3.
 * 4. Lists rules on the event bus.
 * 5. Creates a new Amazon Simple Notification Service (Amazon SNS) topic and
 * lets the user subscribe to it.
 * 6. Adds a target to the rule that sends an email to the specified topic.
 * 7. Creates an EventBridge event that sends an email when an Amazon S3 object
 * is created.
 * 8. Lists Targets.
 * 9. Lists the rules for the same target.
 * 10. Triggers the rule by uploading a file to the Amazon S3 bucket.
 * 11. Disables a specific rule.
 * 12. Checks and print the state of the rule.
 * 13. Adds a transform to the rule to change the text of the email.
 * 14. Enables a specific rule.
 * 15. Triggers the updated rule by uploading a file to the Amazon S3 bucket.
 * 16. Updates the rule to be a custom rule pattern.
 * 17. Sending an event to trigger the rule.
 * 18. Cleans up resources.
 *
 */
public class EventbridgeMVP {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws InterruptedException,
 IOException {
        final String usage = """

                Usage:
```

```
                    <roleName> <bucketName> <topicName> <eventRuleName>

                Where:
                    roleName - The name of the role to create.
                    bucketName - The Amazon Simple Storage Service (Amazon S3)
bucket name to create.
                    topicName - The name of the Amazon Simple Notification
Service (Amazon SNS) topic to create.
                    eventRuleName - The Amazon EventBridge rule name to create.
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String polJSON = "{" +
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                "\"Service\": \"events.amazonaws.com\"" +
                "}," +
                "\"Action\": \"sts:AssumeRole\"" +
                "}]" +
                "}";

        Scanner sc = new Scanner(System.in);
        String roleName = args[0];
        String bucketName = args[1];
        String topicName = args[2];
        String eventRuleName = args[3];

        Region region = Region.US_EAST_1;
        EventBridgeClient eventBrClient = EventBridgeClient.builder()
                .region(region)
                .build();

        S3Client s3Client = S3Client.builder()
                .region(region)
                .build();

        Region regionGl = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
```

```
                .region(regionGl)
                .build();

        SnsClient snsClient = SnsClient.builder()
                .region(region)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon EventBridge example
scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out
                .println("1. Create an AWS Identity and Access Management (IAM)
role to use with Amazon EventBridge.");
        String roleArn = createIAMRole(iam, roleName, polJSON);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Create an S3 bucket with EventBridge events
enabled.");
        if (checkBucket(s3Client, bucketName)) {
            System.out.println("Bucket " + bucketName + " already exists. Ending
this scenario.");
            System.exit(1);
        }

        createBucket(s3Client, bucketName);
        Thread.sleep(3000);
        setBucketNotification(s3Client, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Create a rule that triggers when an object is
uploaded to Amazon S3.");
        Thread.sleep(10000);
        addEventRule(eventBrClient, roleArn, bucketName, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. List rules on the event bus.");
        listRules(eventBrClient);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("5. Create a new SNS topic for testing and let the
user subscribe to the topic.");
        String topicArn = createSnsTopic(snsClient, topicName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Add a target to the rule that sends an email to
the specified topic.");
        System.out.println("Enter your email to subscribe to the Amazon SNS
topic:");
        String email = sc.nextLine();
        subEmail(snsClient, topicArn, email);
        System.out.println(
                "Use the link in the email you received to confirm your
subscription. Then, press Enter to continue.");
        sc.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create an EventBridge event that sends an email
when an Amazon S3 object is created.");
        addSnsEventRule(eventBrClient, eventRuleName, topicArn, topicName,
eventRuleName, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 8. List Targets.");
        listTargets(eventBrClient, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 9. List the rules for the same target.");
        listTargetRules(eventBrClient, topicArn);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 10. Trigger the rule by uploading a file to the S3
bucket.");
        System.out.println("Press Enter to continue.");
        sc.nextLine();
        uploadTextFiletoS3(s3Client, bucketName);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("11. Disable a specific rule.");
        changeRuleState(eventBrClient, eventRuleName, false);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Check and print the state of the rule.");
        checkRule(eventBrClient, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Add a transform to the rule to change the text of
the email.");
        updateSnsEventRule(eventBrClient, topicArn, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Enable a specific rule.");
        changeRuleState(eventBrClient, eventRuleName, true);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 15. Trigger the updated rule by uploading a file to
the S3 bucket.");
        System.out.println("Press Enter to continue.");
        sc.nextLine();
        uploadTextFiletoS3(s3Client, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 16. Update the rule to be a custom rule pattern.");
        updateToCustomRule(eventBrClient, eventRuleName);
        System.out.println("Updated event rule " + eventRuleName + " to use a
custom pattern.");
        updateCustomRuleTargetWithTransform(eventBrClient, topicArn,
eventRuleName);
        System.out.println("Updated event target " + topicArn + ".");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("17. Sending an event to trigger the rule. This will
trigger a subscription email.");
        triggerCustomRule(eventBrClient, email);
```

```
        System.out.println("Events have been sent. Press Enter to continue.");
        sc.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("18. Clean up resources.");
        System.out.println("Do you want to clean up resources (y/n)");
        String ans = sc.nextLine();
        if (ans.compareTo("y") == 0) {
            cleanupResources(eventBrClient, snsClient, s3Client, iam, topicArn,
eventRuleName, bucketName, roleName);
        } else {
            System.out.println("The resources will not be cleaned up. ");
        }
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Amazon EventBridge example scenario has
successfully completed.");
        System.out.println(DASHES);
    }

    public static void cleanupResources(EventBridgeClient eventBrClient,
SnsClient snsClient, S3Client s3Client,
            IamClient iam, String topicArn, String eventRuleName, String
bucketName, String roleName) {
        System.out.println("Removing all targets from the event rule.");
        deleteTargetsFromRule(eventBrClient, eventRuleName);
        deleteRuleByName(eventBrClient, eventRuleName);
        deleteSNSTopic(snsClient, topicArn);
        deleteS3Bucket(s3Client, bucketName);
        deleteRole(iam, roleName);
    }

    public static void deleteRole(IamClient iam, String roleName) {
        String policyArn = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess";
        DetachRolePolicyRequest policyRequest = DetachRolePolicyRequest.builder()
                .policyArn(policyArn)
                .roleName(roleName)
                .build();

        iam.detachRolePolicy(policyRequest);
        System.out.println("Successfully detached policy " + policyArn + " from
role " + roleName);
```

```
        // Delete the role.
        DeleteRoleRequest roleRequest = DeleteRoleRequest.builder()
                .roleName(roleName)
                .build();

        iam.deleteRole(roleRequest);
        System.out.println("*** Successfully deleted " + roleName);
    }

    public static void deleteS3Bucket(S3Client s3Client, String bucketName) {
        // Remove all the objects from the S3 bucket.
        ListObjectsRequest listObjects = ListObjectsRequest.builder()
                .bucket(bucketName)
                .build();

        ListObjectsResponse res = s3Client.listObjects(listObjects);
        List<S3Object> objects = res.contents();
        ArrayList<ObjectIdentifier> toDelete = new ArrayList<>();

        for (S3Object myValue : objects) {
            toDelete.add(ObjectIdentifier.builder()
                    .key(myValue.key())
                    .build());
        }

        DeleteObjectsRequest dor = DeleteObjectsRequest.builder()
                .bucket(bucketName)
                .delete(Delete.builder()
                        .objects(toDelete).build())
                .build();

        s3Client.deleteObjects(dor);

        // Delete the S3 bucket.
        DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
                .bucket(bucketName)
                .build();

        s3Client.deleteBucket(deleteBucketRequest);
        System.out.println("You have deleted the bucket and the objects");
    }

    // Delete the SNS topic.
```

```
   public static void deleteSNSTopic(SnsClient snsClient, String topicArn) {
       try {
           DeleteTopicRequest request = DeleteTopicRequest.builder()
                   .topicArn(topicArn)
                   .build();

           DeleteTopicResponse result = snsClient.deleteTopic(request);
           System.out.println("\n\nStatus was " +
result.sdkHttpResponse().statusCode());

       } catch (SnsException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   public static void deleteRuleByName(EventBridgeClient eventBrClient, String
ruleName) {
       DeleteRuleRequest ruleRequest = DeleteRuleRequest.builder()
               .name(ruleName)
               .build();

       eventBrClient.deleteRule(ruleRequest);
       System.out.println("Successfully deleted the rule");
   }

   public static void deleteTargetsFromRule(EventBridgeClient eventBrClient,
String eventRuleName) {
       // First, get all targets that will be deleted.
       ListTargetsByRuleRequest request = ListTargetsByRuleRequest.builder()
               .rule(eventRuleName)
               .build();

       ListTargetsByRuleResponse response =
eventBrClient.listTargetsByRule(request);
       List<Target> allTargets = response.targets();

       // Get all targets and delete them.
       for (Target myTarget : allTargets) {
           RemoveTargetsRequest removeTargetsRequest =
RemoveTargetsRequest.builder()
                   .rule(eventRuleName)
                   .ids(myTarget.id())
                   .build();
```

```
            eventBrClient.removeTargets(removeTargetsRequest);
            System.out.println("Successfully removed the target");
        }
    }

    public static void triggerCustomRule(EventBridgeClient eventBrClient, String
email) {
        String json = "{" +
                "\"UserEmail\": \"" + email + "\"," +
                "\"Message\": \"This event was generated by example code.\"," +
                "\"UtcTime\": \"Now.\"" +
                "}";

        PutEventsRequestEntry entry = PutEventsRequestEntry.builder()
                .source("ExampleSource")
                .detail(json)
                .detailType("ExampleType")
                .build();

        PutEventsRequest eventsRequest = PutEventsRequest.builder()
                .entries(entry)
                .build();

        eventBrClient.putEvents(eventsRequest);
    }

    public static void updateCustomRuleTargetWithTransform(EventBridgeClient
eventBrClient, String topicArn,
            String ruleName) {
        String targetId = java.util.UUID.randomUUID().toString();
        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: sample event was received.\"")
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
```

```
                    .targets(target)
                    .eventBusName(null)
                    .build();

            eventBrClient.putTargets(targetsRequest);
        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void updateToCustomRule(EventBridgeClient eventBrClient, String
ruleName) {
        String customEventsPattern = "{" +
                "\"source\": [\"ExampleSource\"]," +
                "\"detail-type\": [\"ExampleType\"]" +
                "}";

        PutRuleRequest request = PutRuleRequest.builder()
                .name(ruleName)
                .description("Custom test rule")
                .eventPattern(customEventsPattern)
                .build();

        eventBrClient.putRule(request);
    }

    // Update an Amazon S3 object created rule with a transform on the target.
    public static void updateSnsEventRule(EventBridgeClient eventBrClient, String
topicArn, String ruleName) {
        String targetId = java.util.UUID.randomUUID().toString();
        Map<String, String> myMap = new HashMap<>();
        myMap.put("bucket", "$.detail.bucket.name");
        myMap.put("time", "$.time");

        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: an object was uploaded to bucket
<bucket> at <time>.\"")
                .inputPathsMap(myMap)
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
```

```
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
                    .targets(target)
                    .eventBusName(null)
                    .build();

            eventBrClient.putTargets(targetsRequest);

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void checkRule(EventBridgeClient eventBrClient, String
eventRuleName) {
        try {
            DescribeRuleRequest ruleRequest = DescribeRuleRequest.builder()
                    .name(eventRuleName)
                    .build();

            DescribeRuleResponse response =
eventBrClient.describeRule(ruleRequest);
            System.out.println("The state of the rule is " +
response.stateAsString());

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
        try {
            if (!isEnabled) {
                System.out.println("Disabling the rule: " + eventRuleName);
                DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                        .name(eventRuleName)
                        .build();
```

```
                eventBrClient.disableRule(ruleRequest);
            } else {
                System.out.println("Enabling the rule: " + eventRuleName);
                EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                        .name(eventRuleName)
                        .build();
                eventBrClient.enableRule(ruleRequest);
            }

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Create and upload a file to an S3 bucket to trigger an event.
    public static void uploadTextFiletoS3(S3Client s3Client, String bucketName)
throws IOException {
        // Create a unique file name.
        String fileSuffix = new SimpleDateFormat("yyyyMMddHHmmss").format(new
Date());
        String fileName = "TextFile" + fileSuffix + ".txt";

        File myFile = new File(fileName);
        FileWriter fw = new FileWriter(myFile.getAbsoluteFile());
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write("This is a sample file for testing uploads.");
        bw.close();

        try {
            PutObjectRequest putOb = PutObjectRequest.builder()
                    .bucket(bucketName)
                    .key(fileName)
                    .build();

            s3Client.putObject(putOb, RequestBody.fromFile(myFile));

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

```
   public static void listTargetRules(EventBridgeClient eventBrClient, String
topicArn) {
       ListRuleNamesByTargetRequest ruleNamesByTargetRequest =
ListRuleNamesByTargetRequest.builder()
               .targetArn(topicArn)
               .build();

       ListRuleNamesByTargetResponse response =
eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest);
       List<String> rules = response.ruleNames();
       for (String rule : rules) {
           System.out.println("The rule name is " + rule);
       }
   }

   public static void listTargets(EventBridgeClient eventBrClient, String
ruleName) {
       ListTargetsByRuleRequest ruleRequest = ListTargetsByRuleRequest.builder()
               .rule(ruleName)
               .build();

       ListTargetsByRuleResponse res =
eventBrClient.listTargetsByRule(ruleRequest);
       List<Target> targetsList = res.targets();
       for (Target target: targetsList) {
           System.out.println("Target ARN: "+target.arn());
       }
   }

   // Add a rule which triggers an SNS target when a file is uploaded to an S3
   // bucket.
   public static void addSnsEventRule(EventBridgeClient eventBrClient, String
ruleName, String topicArn,
           String topicName, String eventRuleName, String bucketName) {
       String targetID = java.util.UUID.randomUUID().toString();
       Target myTarget = Target.builder()
               .id(targetID)
               .arn(topicArn)
               .build();

       List<Target> targets = new ArrayList<>();
       targets.add(myTarget);
       PutTargetsRequest request = PutTargetsRequest.builder()
               .eventBusName(null)
```

```
                .targets(targets)
                .rule(ruleName)
                .build();

        eventBrClient.putTargets(request);
        System.out.println("Added event rule " + eventRuleName + " with Amazon
SNS target " + topicName + " for bucket "
                + bucketName + ".");
    }

    public static void subEmail(SnsClient snsClient, String topicArn, String
email) {
        try {
            SubscribeRequest request = SubscribeRequest.builder()
                    .protocol("email")
                    .endpoint(email)
                    .returnSubscriptionArn(true)
                    .topicArn(topicArn)
                    .build();

            SubscribeResponse result = snsClient.subscribe(request);
            System.out.println("Subscription ARN: " + result.subscriptionArn() +
"\n\n Status is "
                    + result.sdkHttpResponse().statusCode());

        } catch (SnsException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void listRules(EventBridgeClient eventBrClient) {
        try {
            ListRulesRequest rulesRequest = ListRulesRequest.builder()
                    .eventBusName("default")
                    .limit(10)
                    .build();

            ListRulesResponse response = eventBrClient.listRules(rulesRequest);
            List<Rule> rules = response.rules();
            for (Rule rule : rules) {
                System.out.println("The rule name is : " + rule.name());
                System.out.println("The rule description is : " +
rule.description());
```

```
                System.out.println("The rule state is : " +
rule.stateAsString());
            }

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static String createSnsTopic(SnsClient snsClient, String topicName) {
        String topicPolicy = "{" +
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
                "\"Sid\": \"EventBridgePublishTopic\"," +
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                "\"Service\": \"events.amazonaws.com\"" +
                "}," +
                "\"Resource\": \"*\"," +
                "\"Action\": \"sns:Publish\"" +
                "}]" +
                "}";

        Map<String, String> topicAttributes = new HashMap<>();
        topicAttributes.put("Policy", topicPolicy);
        CreateTopicRequest topicRequest = CreateTopicRequest.builder()
                .name(topicName)
                .attributes(topicAttributes)
                .build();

        CreateTopicResponse response = snsClient.createTopic(topicRequest);
        System.out.println("Added topic " + topicName + " for email
subscriptions.");
        return response.topicArn();
    }

    // Create a new event rule that triggers when an Amazon S3 object is created
in
    // a bucket.
    public static void addEventRule(EventBridgeClient eventBrClient, String
roleArn, String bucketName,
            String eventRuleName) {
        String pattern = "{\n" +
```

```
                "  \"source\": [\"aws.s3\"],\n" +
                "  \"detail-type\": [\"Object Created\"],\n" +
                "  \"detail\": {\n" +
                "    \"bucket\": {\n" +
                "      \"name\": [\"" + bucketName + "\"]\n" +
                "    }\n" +
                "  }\n" +
                "}";

        try {
            PutRuleRequest ruleRequest = PutRuleRequest.builder()
                    .description("Created by using the AWS SDK for Java v2")
                    .name(eventRuleName)
                    .eventPattern(pattern)
                    .roleArn(roleArn)
                    .build();

            PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
            System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Determine if the S3 bucket exists.
    public static Boolean checkBucket(S3Client s3Client, String bucketName) {
        try {
            HeadBucketRequest headBucketRequest = HeadBucketRequest.builder()
                    .bucket(bucketName)
                    .build();

            s3Client.headBucket(headBucketRequest);
            return true;
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
        return false;
    }

    // Set the S3 bucket notification configuration.
```

```
   public static void setBucketNotification(S3Client s3Client, String
bucketName) {
       try {
           EventBridgeConfiguration eventBridgeConfiguration =
EventBridgeConfiguration.builder()
                   .build();

           NotificationConfiguration configuration =
NotificationConfiguration.builder()
                   .eventBridgeConfiguration(eventBridgeConfiguration)
                   .build();

           PutBucketNotificationConfigurationRequest configurationRequest =
PutBucketNotificationConfigurationRequest
                   .builder()
                   .bucket(bucketName)
                   .notificationConfiguration(configuration)
                   .skipDestinationValidation(true)
                   .build();

           s3Client.putBucketNotificationConfiguration(configurationRequest);
           System.out.println("Added bucket " + bucketName + " with EventBridge
events enabled.");

       } catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   public static void createBucket(S3Client s3Client, String bucketName) {
       try {
           S3Waiter s3Waiter = s3Client.waiter();
           CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
                   .bucket(bucketName)
                   .build();

           s3Client.createBucket(bucketRequest);
           HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
                   .bucket(bucketName)
                   .build();

           // Wait until the bucket is created and print out the response.
```

```
            WaiterResponse<HeadBucketResponse> waiterResponse =
 s3Waiter.waitUntilBucketExists(bucketRequestWait);
            waiterResponse.matched().response().ifPresent(System.out::println);
            System.out.println(bucketName + " is ready");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static String createIAMRole(IamClient iam, String rolename, String
 polJSON) {
        try {
            CreateRoleRequest request = CreateRoleRequest.builder()
                    .roleName(rolename)
                    .assumeRolePolicyDocument(polJSON)
                    .description("Created using the AWS SDK for Java")
                    .build();

            CreateRoleResponse response = iam.createRole(request);
            AttachRolePolicyRequest rolePolicyRequest =
 AttachRolePolicyRequest.builder()
                    .roleName(rolename)
                    .policyArn("arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess")
                    .build();

            iam.attachRolePolicy(rolePolicyRequest);
            return response.role().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - DeleteRule

- DescribeRule
- DisableRule
- EnableRule
- ListRuleNamesByTarget
- ListRules
- ListTargetsByRule
- PutEvents
- PutRule
- PutTargets

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/*
 Before running this Kotlin 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-kotlin/latest/developer-guide/setup.html

 This Kotlin example performs the following tasks with Amazon EventBridge:

 1. Creates an AWS Identity and Access Management (IAM) role to use with Amazon
 EventBridge.
 2. Creates an Amazon Simple Storage Service (Amazon S3) bucket with EventBridge
 events enabled.
 3. Creates a rule that triggers when an object is uploaded to Amazon S3.
 4. Lists rules on the event bus.
 5. Creates a new Amazon Simple Notification Service (Amazon SNS) topic and lets
 the user subscribe to it.
 6. Adds a target to the rule that sends an email to the specified topic.
```

```
 7. Creates an EventBridge event that sends an email when an Amazon S3 object is
 created.
 8. Lists targets.
 9. Lists the rules for the same target.
 10. Triggers the rule by uploading a file to the S3 bucket.
 11. Disables a specific rule.
 12. Checks and prints the state of the rule.
 13. Adds a transform to the rule to change the text of the email.
 14. Enables a specific rule.
 15. Triggers the updated rule by uploading a file to the S3 bucket.
 16. Updates the rule to a custom rule pattern.
 17. Sends an event to trigger the rule.
 18. Cleans up resources.
*/
val DASHES: String = String(CharArray(80)).replace("\u0000", "-")
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <roleName> <bucketName> <topicName> <eventRuleName>

    Where:
        roleName - The name of the role to create.
        bucketName - The Amazon Simple Storage Service (Amazon S3) bucket name to
 create.
        topicName - The name of the Amazon Simple Notification Service (Amazon
 SNS) topic to create.
        eventRuleName - The Amazon EventBridge rule name to create.
    """
    val polJSON = "{" +
        "\"Version\": \"2012-10-17\"," +
        "\"Statement\": [{" +
        "\"Effect\": \"Allow\"," +
        "\"Principal\": {" +
        "\"Service\": \"events.amazonaws.com\"" +
        "}," +
        "\"Action\": \"sts:AssumeRole\"" +
        "}]" +
        "}"

    if (args.size != 4) {
        println(usage)
        exitProcess(1)
    }
```

```
    val sc = Scanner(System.`in`)
    val roleName = args[0]
    val bucketName = args[1]
    val topicName = args[2]
    val eventRuleName = args[3]

    println(DASHES)
    println("Welcome to the Amazon EventBridge example scenario.")
    println(DASHES)

    println(DASHES)
    println("1. Create an AWS Identity and Access Management (IAM) role to use
with Amazon EventBridge.")
    val roleArn = createIAMRole(roleName, polJSON)
    println(DASHES)

    println(DASHES)
    println("2. Create an S3 bucket with EventBridge events enabled.")
    if (checkBucket(bucketName)) {
        println("$bucketName already exists. Ending this scenario.")
        exitProcess(1)
    }

    createBucket(bucketName)
    delay(3000)
    setBucketNotification(bucketName)
    println(DASHES)

    println(DASHES)
    println("3. Create a rule that triggers when an object is uploaded to Amazon
S3.")
    delay(10000)
    addEventRule(roleArn, bucketName, eventRuleName)
    println(DASHES)

    println(DASHES)
    println("4. List rules on the event bus.")
    listRules()
    println(DASHES)

    println(DASHES)
    println("5. Create a new SNS topic for testing and let the user subscribe to
the topic.")
    val topicArn = createSnsTopic(topicName)
```

```
    println(DASHES)

    println(DASHES)
    println("6. Add a target to the rule that sends an email to the specified
topic.")
    println("Enter your email to subscribe to the Amazon SNS topic:")
    val email = sc.nextLine()
    subEmail(topicArn, email)
    println("Use the link in the email you received to confirm your subscription.
Then press Enter to continue.")
    sc.nextLine()
    println(DASHES)

    println(DASHES)
    println("7. Create an EventBridge event that sends an email when an Amazon S3
object is created.")
    addSnsEventRule(eventRuleName, topicArn, topicName, eventRuleName,
bucketName)
    println(DASHES)

    println(DASHES)
    println("8. List targets.")
    listTargets(eventRuleName)
    println(DASHES)

    println(DASHES)
    println(" 9. List the rules for the same target.")
    listTargetRules(topicArn)
    println(DASHES)

    println(DASHES)
    println("10. Trigger the rule by uploading a file to the S3 bucket.")
    println("Press Enter to continue.")
    sc.nextLine()
    uploadTextFiletoS3(bucketName)
    println(DASHES)

    println(DASHES)
    println("11. Disable a specific rule.")
    changeRuleState(eventRuleName, false)
    println(DASHES)

    println(DASHES)
    println("12. Check and print the state of the rule.")
```

```
    checkRule(eventRuleName)
    println(DASHES)

    println(DASHES)
    println("13. Add a transform to the rule to change the text of the email.")
    updateSnsEventRule(topicArn, eventRuleName)
    println(DASHES)

    println(DASHES)
    println("14. Enable a specific rule.")
    changeRuleState(eventRuleName, true)
    println(DASHES)

    println(DASHES)
    println("15. Trigger the updated rule by uploading a file to the S3 bucket.")
    println("Press Enter to continue.")
    sc.nextLine()
    uploadTextFiletoS3(bucketName)
    println(DASHES)

    println(DASHES)
    println("16. Update the rule to a custom rule pattern.")
    updateToCustomRule(eventRuleName)
    println("Updated event rule $eventRuleName to use a custom pattern.")
    updateCustomRuleTargetWithTransform(topicArn, eventRuleName)
    println("Updated event target $topicArn.")
    println(DASHES)

    println(DASHES)
    println("17. Send an event to trigger the rule. This will trigger a
subscription email.")
    triggerCustomRule(email)
    println("Events have been sent. Press Enter to continue.")
    sc.nextLine()
    println(DASHES)

    println(DASHES)
    println("18. Clean up resources.")
    println("Do you want to clean up resources (y/n)")
    val ans = sc.nextLine()
    if (ans.compareTo("y") == 0) {
        cleanupResources(topicArn, eventRuleName, bucketName, roleName)
    } else {
        println("The resources will not be cleaned up. ")
```

```
    }
    println(DASHES)

    println(DASHES)
    println("The Amazon EventBridge example scenario has successfully
 completed.")
    println(DASHES)
}

suspend fun cleanupResources(topicArn: String?, eventRuleName: String?,
 bucketName: String?, roleName: String?) {
    println("Removing all targets from the event rule.")
    deleteTargetsFromRule(eventRuleName)
    deleteRuleByName(eventRuleName)
    deleteSNSTopic(topicArn)
    deleteS3Bucket(bucketName)
    deleteRole(roleName)
}

suspend fun deleteRole(roleNameVal: String?) {
    val policyArnVal = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess"
    val policyRequest = DetachRolePolicyRequest {
        policyArn = policyArnVal
        roleName = roleNameVal
    }
    IamClient { region = "us-east-1" }.use { iam ->
        iam.detachRolePolicy(policyRequest)
        println("Successfully detached policy $policyArnVal from role
 $roleNameVal")

        // Delete the role.
        val roleRequest = DeleteRoleRequest {
            roleName = roleNameVal
        }

        iam.deleteRole(roleRequest)
        println("*** Successfully deleted $roleNameVal")
    }
}

suspend fun deleteS3Bucket(bucketName: String?) {
    // Remove all the objects from the S3 bucket.
    val listObjects = ListObjectsRequest {
        bucket = bucketName
```

```
    }
    S3Client { region = "us-east-1" }.use { s3Client ->
        val res = s3Client.listObjects(listObjects)
        val myObjects = res.contents
        val toDelete = mutableListOf<ObjectIdentifier>()

        if (myObjects != null) {
            for (myValue in myObjects) {
                toDelete.add(
                    ObjectIdentifier {
                        key = myValue.key
                    }
                )
            }
        }

        val delOb = Delete {
            objects = toDelete
        }

        val dor = DeleteObjectsRequest {
            bucket = bucketName
            delete = delOb
        }
        s3Client.deleteObjects(dor)

        // Delete the S3 bucket.
        val deleteBucketRequest = DeleteBucketRequest {
            bucket = bucketName
        }
        s3Client.deleteBucket(deleteBucketRequest)
        println("You have deleted the bucket and the objects")
    }
}

// Delete the SNS topic.
suspend fun deleteSNSTopic(topicArnVal: String?) {
    val request = DeleteTopicRequest {
        topicArn = topicArnVal
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        snsClient.deleteTopic(request)
        println(" $topicArnVal was deleted.")
```

```
    }
}

suspend fun deleteRuleByName(ruleName: String?) {
    val ruleRequest = DeleteRuleRequest {
        name = ruleName
    }
    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.deleteRule(ruleRequest)
        println("Successfully deleted the rule")
    }
}

suspend fun deleteTargetsFromRule(eventRuleName: String?) {
    // First, get all targets that will be deleted.
    val request = ListTargetsByRuleRequest {
        rule = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(request)
        val allTargets = response.targets

        // Get all targets and delete them.
        if (allTargets != null) {
            for (myTarget in allTargets) {
                val removeTargetsRequest = RemoveTargetsRequest {
                    rule = eventRuleName
                    ids = listOf(myTarget.id.toString())
                }
                eventBrClient.removeTargets(removeTargetsRequest)
                println("Successfully removed the target")
            }
        }
    }
}

suspend fun triggerCustomRule(email: String) {
    val json = "{" +
        "\"UserEmail\": \"" + email + "\"," +
        "\"Message\": \"This event was generated by example code.\"" +
        "\"UtcTime\": \"Now.\"" +
        "}"
```

```
    val entry = PutEventsRequestEntry {
        source = "ExampleSource"
        detail = json
        detailType = "ExampleType"
    }

    val eventsRequest = PutEventsRequest {
        this.entries = listOf(entry)
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putEvents(eventsRequest)
    }
}

suspend fun updateCustomRuleTargetWithTransform(topicArn: String?, ruleName:
 String?) {
    val targetId = UUID.randomUUID().toString()

    val inputTransformerOb = InputTransformer {
        inputTemplate = "\"Notification: sample event was received.\""
    }

    val target = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransformerOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
        targets = listOf(target)
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(targetsRequest)
    }
}

suspend fun updateToCustomRule(ruleName: String?) {
    val customEventsPattern = "{" +
        "\"source\": [\"ExampleSource\"]," +
        "\"detail-type\": [\"ExampleType\"]" +
```

```
        "}"
    val request = PutRuleRequest {
        name = ruleName
        description = "Custom test rule"
        eventPattern = customEventsPattern
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putRule(request)
    }
}

// Update an Amazon S3 object created rule with a transform on the target.
suspend fun updateSnsEventRule(topicArn: String?, ruleName: String?) {
    val targetId = UUID.randomUUID().toString()
    val myMap = mutableMapOf<String, String>()
    myMap["bucket"] = "$.detail.bucket.name"
    myMap["time"] = "$.time"

    val inputTransOb = InputTransformer {
        inputTemplate = "\"Notification: an object was uploaded to bucket
 <bucket> at <time>.\""
        inputPathsMap = myMap
    }
    val targetOb = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
        targets = listOf(targetOb)
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(targetsRequest)
    }
}

suspend fun checkRule(eventRuleName: String?) {
    val ruleRequest = DescribeRuleRequest {
        name = eventRuleName
```

```
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.describeRule(ruleRequest)
        println("The state of the rule is $response")
    }
}

suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}

// Create and upload a file to an S3 bucket to trigger an event.
@Throws(IOException::class)
suspend fun uploadTextFiletoS3(bucketName: String?) {
    val fileSuffix = SimpleDateFormat("yyyyMMddHHmmss").format(Date())
    val fileName = "TextFile$fileSuffix.txt"
    val myFile = File(fileName)
    val fw = FileWriter(myFile.absoluteFile)
    val bw = BufferedWriter(fw)
    bw.write("This is a sample file for testing uploads.")
    bw.close()

    val putOb = PutObjectRequest {
        bucket = bucketName
        key = fileName
        body = myFile.asByteStream()
    }
```

```
    S3Client { region = "us-east-1" }.use { s3Client ->
        s3Client.putObject(putOb)
    }
}

suspend fun listTargetRules(topicArnVal: String?) {
    val ruleNamesByTargetRequest = ListRuleNamesByTargetRequest {
        targetArn = topicArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response =
 eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest)
        response.ruleNames?.forEach { rule ->
            println("The rule name is $rule")
        }
    }
}

suspend fun listTargets(ruleName: String?) {
    val ruleRequest = ListTargetsByRuleRequest {
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(ruleRequest)
        response.targets?.forEach { target ->
            println("Target ARN: ${target.arn}")
        }
    }
}

// Add a rule that triggers an SNS target when a file is uploaded to an S3
 bucket.
suspend fun addSnsEventRule(ruleName: String?, topicArn: String?, topicName:
 String, eventRuleName: String, bucketName: String) {
    val targetID = UUID.randomUUID().toString()
    val myTarget = Target {
        id = targetID
        arn = topicArn
    }

    val targetsOb = mutableListOf<Target>()
```

```
    targetsOb.add(myTarget)

    val request = PutTargetsRequest {
        eventBusName = null
        targets = targetsOb
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(request)
        println("Added event rule $eventRuleName with Amazon SNS target
 $topicName for bucket $bucketName.")
    }
}

suspend fun subEmail(topicArnVal: String?, email: String?) {
    val request = SubscribeRequest {
        protocol = "email"
        endpoint = email
        returnSubscriptionArn = true
        topicArn = topicArnVal
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        val result = snsClient.subscribe(request)
        println(" Subscription ARN: ${result.subscriptionArn}")
    }
}

suspend fun createSnsTopic(topicName: String): String? {
    val topicPolicy = "{" +
        "\"Version\": \"2012-10-17\"," +
        "\"Statement\": [{" +
        "\"Sid\": \"EventBridgePublishTopic\"," +
        "\"Effect\": \"Allow\"," +
        "\"Principal\": {" +
        "\"Service\": \"events.amazonaws.com\"" +
        "}," +
        "\"Resource\": \"*\"," +
        "\"Action\": \"sns:Publish\"" +
        "}]" +
        "}"

    val topicAttributes = mutableMapOf<String, String>()
```

```
    topicAttributes["Policy"] = topicPolicy

    val topicRequest = CreateTopicRequest {
        name = topicName
        attributes = topicAttributes
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        val response = snsClient.createTopic(topicRequest)
        println("Added topic $topicName for email subscriptions.")
        return response.topicArn
    }
}

suspend fun listRules() {
    val rulesRequest = ListRulesRequest {
        eventBusName = "default"
        limit = 10
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listRules(rulesRequest)
        response.rules?.forEach { rule ->
            println("The rule name is ${rule.name}")
            println("The rule ARN is ${rule.arn}")
        }
    }
}

// Create a new event rule that triggers when an Amazon S3 object is created in a
 bucket.
suspend fun addEventRule(roleArnVal: String?, bucketName: String, eventRuleName:
 String?) {
    val pattern = """{
        "source": ["aws.s3"],
        "detail-type": ["Object Created"],
        "detail": {
        "bucket": {
            "name": ["$bucketName"]
            }
        }
    }"""

    val ruleRequest = PutRuleRequest {
```

```
        description = "Created by using the AWS SDK for Kotlin"
        name = eventRuleName
        eventPattern = pattern
        roleArn = roleArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}

// Set the Amazon S3 bucket notification configuration.
suspend fun setBucketNotification(bucketName: String) {
    val eventBridgeConfig = EventBridgeConfiguration {
    }

    val configuration = NotificationConfiguration {
        eventBridgeConfiguration = eventBridgeConfig
    }

    val configurationRequest = PutBucketNotificationConfigurationRequest {
        bucket = bucketName
        notificationConfiguration = configuration
        skipDestinationValidation = true
    }

    S3Client { region = "us-east-1" }.use { s3Client ->
        s3Client.putBucketNotificationConfiguration(configurationRequest)
        println("Added bucket $bucketName with EventBridge events enabled.")
    }
}

// Create an S3 bucket using a waiter.
suspend fun createBucket(bucketName: String) {
    val request = CreateBucketRequest {
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
        s3.waitUntilBucketExists {
            bucket = bucketName
        }
```

```
        println("$bucketName is ready")
    }
}

suspend fun checkBucket(bucketName: String?): Boolean {
    try {
        // Determine if the S3 bucket exists.
        val headBucketRequest = HeadBucketRequest {
            bucket = bucketName
        }

        S3Client { region = "us-east-1" }.use { s3Client ->
            s3Client.headBucket(headBucketRequest)
            return true
        }
    } catch (e: S3Exception) {
        System.err.println(e.message)
    }
    return false
}

suspend fun createIAMRole(rolenameVal: String?, polJSON: String?): String? {
    val request = CreateRoleRequest {
        roleName = rolenameVal
        assumeRolePolicyDocument = polJSON
        description = "Created using the AWS SDK for Kotlin"
    }

    val rolePolicyRequest = AttachRolePolicyRequest {
        roleName = rolenameVal
        policyArn = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess"
    }

    IamClient { region = "us-east-1" }.use { iam ->
        val response = iam.createRole(request)
        iam.attachRolePolicy(rolePolicyRequest)
        return response.role?.arn
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Kotlin.

- DeleteRule
- DescribeRule
- DisableRule
- EnableRule
- ListRuleNamesByTarget
- ListRules
- ListTargetsByRule
- PutEvents
- PutRule
- PutTargets

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo EventBridge con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi multidisciplinari per EventBridge l'utilizzo degli SDK AWS

Le seguenti applicazioni di esempio utilizzano AWS gli SDK per combinarsi EventBridge con altri. Servizi AWS Ogni esempio include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire l'applicazione.

Esempi

- Utilizzo degli eventi pianificati per richiamare una funzione Lambda

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.

Java

SDK per Java 2.x

Mostra come creare un evento EventBridge pianificato da Amazon che richiami una AWS Lambda funzione. Configura EventBridge per utilizzare un'espressione cron per pianificare

quando viene richiamata la funzione Lambda. In questo esempio, viene creata una funzione Lambda utilizzando l'API di runtime Lambda Java. Questo esempio richiama diversi AWS servizi per eseguire un caso d'uso specifico. Questo esempio dimostra come creare un'app che invia un messaggio di testo via mobile ai tuoi dipendenti che si congratula con loro alla data dell'anniversario di un anno.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

JavaScript

SDK per JavaScript (v3)

Mostra come creare un evento EventBridge pianificato da Amazon che richiami una AWS Lambda funzione. Configura EventBridge per utilizzare un'espressione cron per pianificare quando viene richiamata la funzione Lambda. In questo esempio, crei una funzione Lambda utilizzando l'API JavaScript Lambda runtime. Questo esempio richiama diversi AWS servizi per eseguire un caso d'uso specifico. Questo esempio dimostra come creare un'app che invia un messaggio di testo via mobile ai tuoi dipendenti che si congratula con loro alla data dell'anniversario di un anno.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Questo esempio è anche disponibile nella Guida per lo sviluppatore di AWS SDK for JavaScript v3 .

Servizi utilizzati in questo esempio

- DynamoDB
- EventBridge
- Lambda

- Amazon SNS

Python

SDK per Python (Boto3)

Questo esempio mostra come registrare una AWS Lambda funzione come destinazione di un EventBridge evento Amazon pianificato. Il gestore Lambda scrive un messaggio intuitivo e i dati completi dell'evento su Amazon CloudWatch Logs per recuperarli in un secondo momento.

- Distribuzione di una funzione Lambda.
- Crea un evento EventBridge pianificato e rende la funzione Lambda la destinazione.
- Concede il permesso di EventBridge invocare la funzione Lambda.
- Stampa i dati più recenti dai CloudWatch registri per mostrare il risultato delle chiamate pianificate.
- Elimina tutte le risorse create durante la demo.

Questo esempio è visualizzato al meglio su. GitHub Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- CloudWatch Registri
- EventBridge
- Lambda

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Amazon EventBridge sicurezza

Amazon EventBridge usi AWS Identity and Access Management per controllare l'accesso ad altri AWS servizi e risorse. Per una panoramica sul funzionamento di IAM, vedi Panoramica della gestione degli accessi nella Guida per l'utente di IAM. Per una panoramica delle credenziali di sicurezza, consultare Credenziali di sicurezza AWS in Riferimenti generali di Amazon Web Services.

Argomenti

Protezione dei dati in Amazon EventBridge

Il modello di responsabilità AWS condivisa di si applica alla protezione dei dati in Amazon EventBridge. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or 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:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

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 lavori o Servizi AWS utilizzi la console, l'API EventBridge o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Crittografia dei dati per bus di EventBridge eventi

EventBridge fornisce sia la crittografia a riposo che la crittografia in transito per proteggere i dati degli eventi:

- Crittografia a riposo

  EventBridge si integra con AWS Key Management Service (KMS) per crittografare i dati degli eventi memorizzati sui bus degli eventi. Per impostazione predefinita, EventBridge utilizza an Chiave di proprietà di AWS per crittografare i dati degli eventi. È inoltre possibile specificare di EventBridge utilizzare invece un chiave gestita dal cliente per eventi personalizzati e per i partner.
- Crittografia in transito

  EventBridge crittografa i dati che passano tra EventBridge e altri servizi utilizzando Transport Layer Security (TLS). Per i bus di eventi, ciò include durante un evento a EventBridge cui viene inviato e quando EventBridge invia un evento a un obiettivo della regola.

Crittografia a riposo per gli event bus

EventBridge fornisce una crittografia trasparente lato server mediante l'integrazione con AWS Key Management Service (KMS). La crittografia predefinita dei dati a riposo aiuta a ridurre il sovraccarico operativo e la complessità associati alla protezione dei dati sensibili. Allo stesso tempo, consente di creare applicazioni sicure che soddisfano i rigorosi requisiti normativi e di conformità alla crittografia.

La EventBridge crittografia dei dati del bus degli eventi a riposo include:

- Dati relativi agli eventi AWS, agli eventi personalizzati e ai partner.

  Per gli event bus, i dati degli eventi includono tutti i campi contenuti nell'???elemento dell'evento.

  EventBridge non crittografa i metadati degli eventi. Per ulteriori informazioni sui metadati degli eventi, vedere. ???
- Modelli di eventi
- Trasformatori di ingresso

Per impostazione predefinita, EventBridge utilizza an Chiave di proprietà di AWS per crittografare i dati degli eventi. È inoltre possibile specificare di EventBridge utilizzare invece un chiave gestita dal cliente per eventi personalizzati e per i partner.

Considerazioni sulla sicurezza per la crittografia del bus degli eventi

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili nei seguenti campi, poiché non sono crittografate quando sono archiviate:

- Nomi dei bus degli eventi
- Nomi delle regole
- Risorse condivise come i tag

KMS key opzioni per la crittografia del bus degli eventi

EventBridge utilizza an Chiave di proprietà di AWS per crittografare gli eventi AWS di servizio memorizzati sui bus di eventi.

Per ogni bus di eventi, puoi scegliere il tipo di KMS key EventBridge utilizzo per crittografare gli eventi personalizzati e dei partner memorizzati su quel bus:

- Chiave di proprietà di AWS

  Per impostazione predefinita, EventBridge crittografa i dati utilizzando l'Advanced Encryption Standard (AES-256) a 256 bit con un Chiave di proprietà di AWS, che aiuta a proteggere i dati da accessi non autorizzati.

  Non è possibile visualizzarne, gestirne o utilizzarne o controllarne l'utilizzo Chiavi di proprietà di AWS. Tuttavia, non è necessario effettuare alcuna operazione o modificare programmi per proteggere le chiavi che eseguono la crittografia dei dati.

  In generale, a meno che non sia necessario verificare o controllare la chiave di crittografia che protegge le risorse, an Chiave di proprietà di AWS è una buona scelta. Chiavi di proprietà di AWS sono completamente gratuiti (senza canoni mensili o costi di utilizzo) e non influiscono sulle AWS KMS quote del tuo account. Non è necessario creare o mantenere la chiave o la relativa policy delle chiavi.

  Per ulteriori informazioni, consulta la pagina chiavi di proprietàAWS nella Guida per gli sviluppatori di AWS Key Management Service .
- Chiave gestita dal cliente

EventBridge supporta l'uso di un sistema simmetrico chiave gestita dal cliente creato, posseduto e gestito dall'utente. Poiché avete il pieno controllo di questo tipo di file KMS key, potete eseguire attività come:

- Stabilire e mantenere le policy delle chiavi
- Stabilire e mantenere le policy e le sovvenzioni IAM
- Abilitare e disabilitare le policy delle chiavi
- Ruotare i materiali crittografici delle chiavi
- Aggiungere tag
- Creare alias delle chiavi
- Pianificare l'eliminazione delle chiavi

Per ulteriori informazioni, consulta la sezione Chiavi gestite dal cliente nella Guida per gli sviluppatori di AWS Key Management Service .

EventBridge supporta chiavi multiregionali e l'accesso alle chiavi da più account.

Chiavi gestite dal cliente incorrere in un canone mensile. Per i dettagli, consulta AWS Key Management Service Prezzi e quote nella Guida per gli AWS Key Management Service sviluppatori.

> **Note**
>
> EventBridge non supporta le seguenti funzionalità sui bus di eventi crittografati con chiavi gestite dal cliente:
>
> - Archivi
> - Scoperta dello schema

Per ulteriori informazioni, consulta ???

Crittografia degli eventi con chiavi gestite dal cliente

È possibile specificare di EventBridge utilizzare AWS KMS chiave gestita dal cliente a per crittografare i dati (eventi personalizzati e partner) memorizzati su un bus di eventi, anziché utilizzare un Chiave di proprietà di AWS as come impostazione predefinita. È possibile specificare a chiave

gestita dal cliente quando si crea o si aggiorna un bus di eventi. È inoltre possibile aggiornare il bus di eventi predefinito chiave gestita dal cliente per utilizzarlo anche per eventi personalizzati e partner. Per ulteriori informazioni, consulta ???.

Se si specifica un chiave gestita dal cliente per un bus di eventi, è possibile specificare una coda di lettere morte (DLQ) per il bus degli eventi. EventBridge invia quindi a tale DLQ tutti gli eventi personalizzati o dei partner che generano errori di crittografia o decrittografia. Per ulteriori informazioni, consulta ???.

Specificare un chiave gestita dal cliente per la crittografia durante la creazione di un bus di eventi (utilizzando la console)

- Segui queste istruzioni:

  ???.

Specificare un chiave gestita dal cliente per la crittografia durante la creazione di un bus di eventi (utilizzando la CLI)

- Durante la chiamata`create-event-bus`, utilizzate l'`kms-key-identifier`opzione per specificare il form EventBridge da utilizzare chiave gestita dal cliente per la crittografia sul bus degli eventi.

  Facoltativamente, utilizzare `dead-letter-config` per specificare una coda di lettere morte (DLQ).

Aggiornamento di un bus di eventi per utilizzare un chiave gestita dal cliente per la crittografia (utilizzando la console)

- Segui queste istruzioni:

  ???.

Aggiornamento di un bus di eventi per utilizzare un chiave gestita dal cliente per la crittografia (utilizzando la CLI)

- Durante la chiamata`update-event-bus`, utilizzate l'`kms-key-identifier`opzione per specificare il modulo chiave gestita dal cliente EventBridge da utilizzare per la crittografia sul bus degli eventi.

Facoltativamente, utilizzare `dead-letter-config` per specificare una coda di lettere morte (DLQ).

Aggiornamento del bus di eventi predefinito per utilizzare un per la crittografia utilizzando chiave gestita dal clienteCloudFormation

Poiché EventBridge inserisce automaticamente il bus degli eventi predefinito nel tuo account, non puoi crearlo utilizzando un CloudFormation modello, come faresti normalmente per qualsiasi risorsa che desideri includere in uno CloudFormation stack. Per includere il bus degli eventi predefinito in uno CloudFormation stack, devi prima importarlo in uno stack. Dopo aver importato il bus degli eventi predefinito in uno stack, potete aggiornare le proprietà del bus degli eventi come desiderate.

- Segui queste istruzioni:

  ???.

Autorizzazione EventBridge all'uso di un chiave gestita dal cliente

Se utilizzi un codice chiave gestita dal cliente nel tuo account per proteggere il tuo EventBridge Event Bus, le relative politiche KMS key devono EventBridge autorizzare l'uso a tuo nome. Fornisci queste autorizzazioni in una politica chiave.

EventBridge non necessita di ulteriori autorizzazioni per utilizzare l'impostazione predefinita Chiave di proprietà di AWS per proteggere le EventBridge risorse del tuo AWS account.

EventBridge richiede le seguenti autorizzazioni su un chiavi gestite dal cliente:

- `kms:DescribeKey`

  EventBridge richiede questa autorizzazione per recuperare l' KMS key ARN per l'ID chiave fornito e per verificare che la chiave sia simmetrica.
- `kms:GenerateDataKey`

  EventBridge richiede questa autorizzazione per generare una chiave dati come chiave di crittografia per i dati dell'evento.
- `kms:Decrypt`

  EventBridge richiede questa autorizzazione per decrittografare la chiave dati crittografata e archiviata con i dati crittografati dell'evento.

EventBridge lo utilizza per la corrispondenza delle regole; gli utenti non hanno mai accesso ai dati.

Il seguente esempio di policy chiave fornisce le autorizzazioni richieste:

```
{
  "Sid": "Allow EventBridge to encrypt events",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ]
  "Resource": "*",
  "Condition": {
    "StringEquals": {
        "kms:EncryptionContext:aws:events:event-bus:arn":
 "arn:aws:events:region:account-id:event-bus/event-bus-arn",
        "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-name"
    }
  }
}
```

Sicurezza quando si utilizza chiavi gestite dal cliente per la crittografia del bus degli EventBridge eventi

Come procedura consigliata in materia di sicurezza `aws:SourceArn`aws:sourceAccount, aggiungi una chiave o una chiave di `kms:EncryptionContext:aws:events:event-bus:arn` condizione alla policy AWS KMS chiave. La chiave di condizione IAM globale aiuta a garantire che la chiave KMS venga EventBridge utilizzata solo per il bus o l'account specificato.

L'esempio seguente dimostra come seguire questa best practice nella propria IAM politica:

```
{
      "Sid": "Allow the use of key",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
```

```
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition" : {
        "StringEquals": {
          "aws:SourceAccount": "arn:aws:events:region:account-id",
          "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-name",
          "kms:EncryptionContext:aws:events:event-bus:arn":
"arn:aws:events:region:account-id:event-bus/event-bus-arn"
        }
      }
```

Gestione della chiavi gestite dal cliente crittografia del bus degli EventBridge eventi

Per garantire che conservi EventBridge sempre l'accesso al necessario chiave gestita dal cliente:

- Non eliminate un file chiave gestita dal cliente finché non siete sicuri che tutti gli eventi crittografati con esso siano stati elaborati.

  Quando eseguite una delle seguenti operazioni, conservate il materiale chiave precedente per assicurarvi di EventBridge poter continuare a utilizzarlo per eventi precedentemente crittografati:

  - Rotazione automatica delle chiavi
  - Rotazione manuale dei tasti
  - Aggiornamento di un alias chiave

  In generale, se state pensando di eliminare una AWS KMS chiave, disattivatela prima e impostate un CloudWatch allarme o un meccanismo simile per essere certi di non dover mai usare la chiave per decrittografare i dati crittografati.
- Non eliminate la politica della chiave che fornisce le autorizzazioni per EventBridge l'utilizzo della chiave.

Altre considerazioni includono:

- Specificare chiavi gestite dal cliente gli obiettivi delle regole, a seconda dei casi.

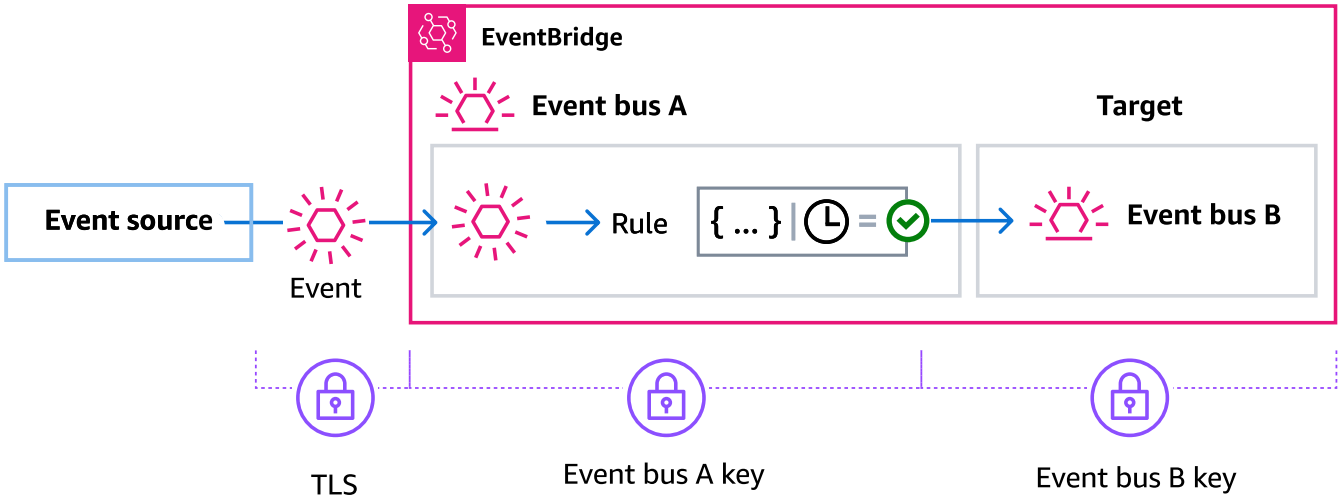
  Quando EventBridge invia un evento a una destinazione della regola, l'evento viene inviato utilizzando Transport Layer Security (TLS). Tuttavia, la crittografia applicata all'evento quando viene archiviato nella destinazione dipende dalla crittografia configurata sulla destinazione stessa.

Crittografia degli eventi quando un bus di eventi è l'obiettivo della regola

Quando un evento personalizzato o di un partner viene inviato a un bus di eventi, EventBridge crittografa l'evento in base alla configurazione della chiave KMS di crittografia a riposo per quel bus di eventi, predefinita Chiave di proprietà di AWS o chiave gestita dal cliente, se specificata. Se un evento corrisponde a una regola, EventBridge crittografa l'evento con la configurazione della chiave KMS per quel bus di eventi finché l'evento non viene inviato alla destinazione della regola, a meno che la destinazione della regola non sia un altro bus di eventi.

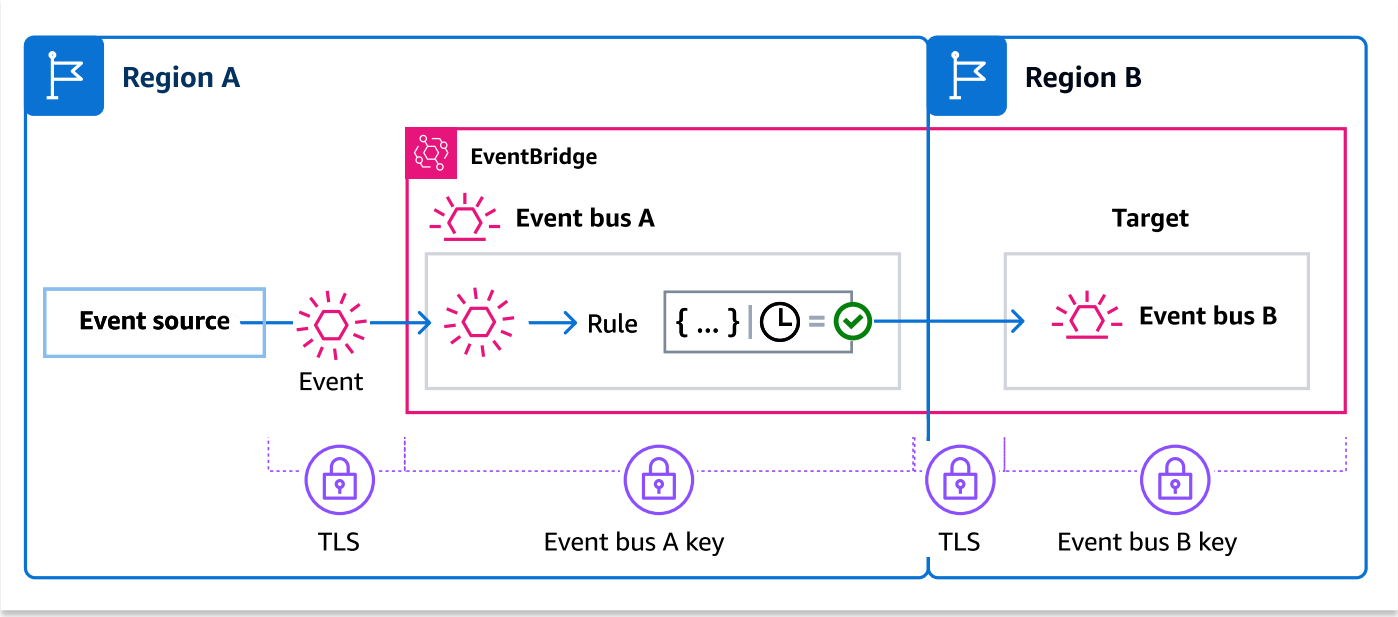
- Se l'obiettivo di una regola è un altro bus di eventi nella stessa AWS regione:

  Se il bus di eventi di destinazione ha un valore specificato chiave gestita dal cliente, EventBridge crittografa invece l'evento con il bus chiave gestita dal cliente di eventi di destinazione per la consegna.



- Se l'obiettivo di una regola è un altro bus di eventi in una AWS regione diversa:

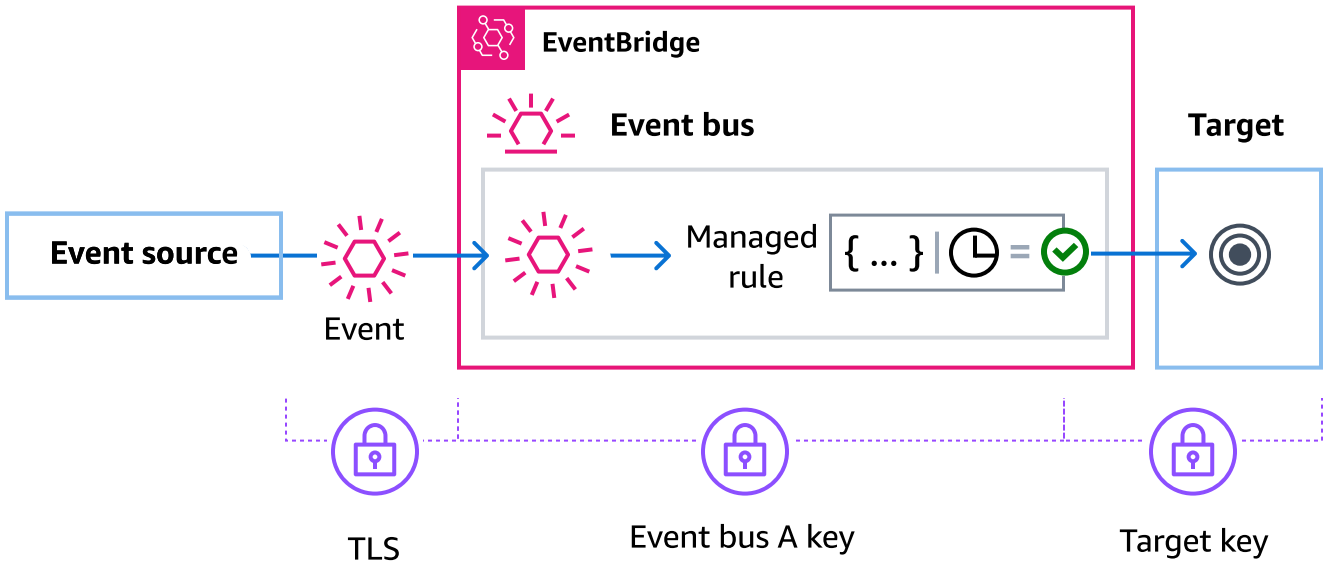
  EventBridge crittografa l'evento a riposo in base alla configurazione della chiave KMS sul primo bus di eventi. EventBridge utilizza TLS per inviare l'evento al secondo bus di eventi nella diversa regione, dove viene quindi crittografato in base alla configurazione della chiave KMS specificata per il bus eventi di destinazione.



Crittografia degli eventi per regole gestite

AWS i servizi possono creare e gestire le regole del bus degli eventi nell' AWS account necessarie per determinate funzioni di tali servizi. Come parte di una regola gestita, il AWS servizio può specificare che EventBridge utilizzare chiave gestita dal cliente quanto specificato per l'obiettivo della regola. Ciò offre la flessibilità necessaria per specificare quale chiave gestita dal cliente utilizzare in base all'obiettivo della regola.

In questi casi, una volta che un evento personalizzato o partner corrisponde alla regola gestita, EventBridge utilizza la destinazione chiave gestita dal cliente specificata dalla regola gestita per crittografare l'evento fino a quando non viene inviato alla destinazione della regola. Ciò avviene indipendentemente dal fatto che il bus degli eventi sia stato configurato per utilizzare il proprio chiave gestita dal cliente per la crittografia. Questo è il caso anche se la destinazione della regola gestita è un altro bus di eventi e tale bus di eventi dispone di un proprio bus di eventi chiave gestita dal cliente specifico per la crittografia. EventBridge continua a utilizzare la destinazione chiave gestita dal cliente specificata nella regola gestita fino a quando l'evento non viene inviato a una destinazione che non è un bus di eventi.



Nei casi in cui l'obiettivo della regola è un bus di eventi in un'altra regione, è necessario fornire una chiave multiregionale. Il bus degli eventi nella prima regione crittografa l'evento utilizzando chiave gestita dal cliente quanto specificato nella regola gestita. Quindi invia l'evento al bus degli eventi di destinazione nella seconda regione. Tale bus di eventi deve essere in grado di continuare a utilizzare il chiave gestita dal cliente fino a quando non invia l'evento alla sua destinazione.

EventBridge contesto di crittografia del bus degli eventi

Un contesto di crittografia è un set di coppie chiave-valore che contiene dati arbitrari non segreti. Quando includi un contesto di crittografia in una richiesta di crittografia dei dati, AWS KMS lega il contesto di crittografia ai dati crittografati, in modo che lo stesso contesto di crittografia sia necessario per decrittografare i dati.

È inoltre possibile utilizzare il contesto di crittografia come condizione per l'autorizzazione nelle politiche e nelle concessioni.

Per gli event bus, EventBridge utilizza lo stesso contesto di crittografia in tutte le operazioni AWS KMS crittografiche. Se si utilizza una chiave gestita dal cliente per proteggere le EventBridge risorse, è possibile utilizzare il contesto di crittografia per identificare l'utilizzo di tale chiave nei record e KMS key nei registri di controllo. Viene inoltre visualizzato nei log in testo chiaro, ad esempio AWS CloudTrail e Amazon CloudWatch Logs.

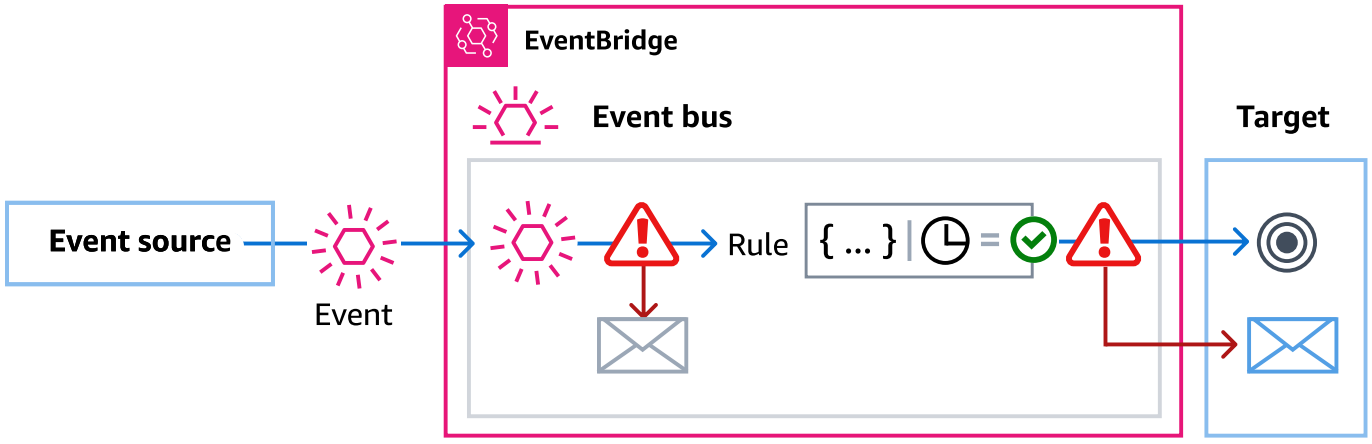
Nelle sue richieste a AWS KMS, EventBridge utilizza un contesto di crittografia con una singola coppia chiave-valore, che contiene l'ARN del bus di eventi:

```
"encryptionContext": {
    "kms:EncryptionContext:aws:events:event-bus:arn": "event-bus-arn"
}
```

Utilizzo di code di lettere morte per acquisire gli errori degli eventi crittografati

Se si configura chiave gestita dal cliente la crittografia su un bus di eventi, si consiglia di specificare una coda di lettere morte (DLQ) per quel bus di eventi. EventBridge invia eventi personalizzati e partner a questo DLQ se rileva un errore irreversibile durante l'elaborazione dell'evento sul bus degli eventi. Un errore non recuperabile è un errore in cui è necessaria l'azione dell'utente per risolvere il problema sottostante, ad esempio se quello specificato viene disabilitato o mancante. chiave gestita dal cliente

- Se si verifica un errore di crittografia o decrittografia non recuperabile durante EventBridge l'elaborazione dell'evento sul bus degli eventi, l'evento viene inviato al DLQ per il bus degli eventi, se specificato.
- Se si verifica un errore di crittografia o decrittografia non recuperabile durante EventBridge il tentativo di inviare l'evento a una destinazione, l'evento viene inviato al DLQ per la destinazione, se specificato.



Per ulteriori informazioni, incluse considerazioni sull'uso dei DLQ e istruzioni sull'impostazione delle autorizzazioni, vedere. ???

Decrittografia degli eventi nelle code con lettere non scritte EventBridge

Una volta risolto il problema di fondo che causa un errore non recuperabile, potete elaborare gli eventi inviati al bus degli eventi o ai DLQ di destinazione. Per gli eventi crittografati, è necessario prima decrittografare l'evento per elaborarlo.

L'esempio seguente mostra come decrittografare un evento inviato a un bus di eventi o a un DLQ di destinazione. EventBridge

```
// You will receive an encrypted event in the following json format.
        // ```
        //   {
        //     "version": "0",
        //     "id": "053afa53-cdd7-285b-e754-b0dfd0ac0bfb",  // New event id not the
 same as the original one
        //     "account": "123456789012",
        //     "time": "2020-02-10T10:22:00Z",
        //     "resources": [ ],
        //     "region": "us-east-1",
        //     "source": "aws.events",
        //     "detail-type": "Encrypted Events",
        //     "detail": {
        //       "event-bus-arn": "arn:aws:events:region:account:event-bus/bus-name",
        //       "rule-arn": "arn:aws:events:region:account:event-bus/bus-name/rule-
name",
        //       "kms-key-arn": "arn:aws:kms:region:account:key/key-arn",
        //       "encrypted-payload": "AgR4qiru/XNwTUyCgRHqP7rbbHn/
xpmVeVeRIAd12TDYYVwAawABABRhd3M6ZXZlbnRzOmV2ZW50LWJ1cwB
        //
 RYXJuOmF3czpldmVudHM6dXMtZWFzdC0xOjE0NjY4NjkwNDY3MzpldmVudC1idXMvY21rbXMtZ2EtY3Jvc3
        //
 MtYWNjb3VudC1zb3VyY2UtYnVzAAEAB2F3cy1rbXMAS2Fybjphd3M6a21zOnVzLWVhc3QtMToxNDY2ODY5"
        //     }
        //   }
        // ```

        // Construct an AwsCrypto object with the encryption algorithm
 `ALG_AES_256_GCM_HKDF_SHA512_COMMIT_KEY` which
        // is used by EventBridge for encryption operation. This object is an entry
 point for decryption operation.
        // It can later use decryptData(MasterKeyProvider, byte[]) method to decrypt
 data.
        final AwsCrypto crypto = AwsCrypto.builder()
```

```
 .withEncryptionAlgorithm(CryptoAlgorithm.ALG_AES_256_GCM_HKDF_SHA512_COMMIT_KEY)
                .build();

        // Construct AWS KMS master key provider with AWS KMS Client Supplier and AWS
 KMS Key ARN. The KMS Client Supplier can
        // implement a RegionalClientSupplier interface. The AWS KMS Key ARN can be
 fetched from kms-key-arn property in
        // encrypted event json detail.
        final KmsMasterKeyProvider kmsMasterKeyProvider =
 KmsMasterKeyProvider.builder()
                .customRegionalClientSupplier(...)
                .buildStrict(KMS_KEY_ARN);

        // The string of encrypted-payload is base64 encoded. Decode it into byte
 array, so it can be furthur
        // decrypted. The encrypted payload can be fetched from encrypted-payload field
 in encrypted event json detail.
        byte[] encryptedByteArray = Base64.getDecoder().decode(ENCRYPTED_PAYLOAD);

        // The decryption operation. It retrieves the encryption context and encrypted
 data key from the cipher
        // text headers, which is parsed from byte array encrypted data. Then it
 decrypts the data key, and
        // uses it to finally decrypt event payload. This encryption/decryption
 strategy is called envelope
        // encryption, https://docs.aws.amazon.com/kms/latest/developerguide/
concepts.html#enveloping
        final CryptoResult<byte[], KmsMasterKey> decryptResult =
 crypto.decryptData(kmsMasterKeyProvider, encryptedByteArray);

        final byte[] decryptedByteArray = decryptResult.getResult();

        // Decode the event json plaintext from byte array into string with UTF_8
 standard.
        String eventJson = new String(decryptedByteArray, StandardCharsets.UTF_8);
```

Policy basate su tag

In Amazon EventBridge, è possibile utilizzare policy basate su tag per controllare l'accesso alle risorse.

Ad esempio, è possibile limitare l'accesso alle risorse che includono un tag con la chiave `environment` e il valore `production`. La policy di esempio seguente nega a qualsiasi risorsa con tale tag di creare, eliminare o modificare tag, regole o router di eventi per le risorse contrassegnate con il tag `environment/production`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "events:PutRule",
                "events:DescribeRule",
                "events:DeleteRule",
                "events:CreateEventBus",
                "events:DescribeEventBus"
                "events:DeleteEventBus"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/environment": "production"}
            }
        }
    ]
}
```

Per ulteriori informazioni sul tagging, consulta:

- EventBridge Etichette Amazon
- Controllo degli accessi tramite tag IAM

Amazon EventBridge e AWS Identity and Access Management

Per accedere ad Amazon EventBridge, hai bisogno di credenziali da AWS utilizzare per autenticare le tue richieste. Queste credenziali devono disporre delle autorizzazioni per accedere alle risorse AWS, ad esempio il recupero di dati di eventi provenienti da altre risorse AWS. Le seguenti sezioni forniscono dettagli su come utilizzare AWS Identity and Access Management(IAM) e su come EventBridge proteggere le risorse controllando chi può accedervi.

Argomenti

Autenticazione

Puoi accedere ad AWS utilizzando uno dei seguenti tipi di identità:

- Account utente root di AWS: quando effettui la registrazione per AWS, fornisci un indirizzo e-mail e una password associata al tuo account. Si tratta delle tue credenziali root, che forniscono accesso completo a tutte le risorse AWS.

> **Important**
>
> Per motivi di sicurezza, consigliamo di utilizzare le credenziali root solo per creare un amministratore, ovvero un utente IAM con autorizzazioni complete per il tuo account. Potrai quindi utilizzare questo amministratore per creare altri utenti e ruoli con autorizzazioni limitate. Per ulteriori informazioni consulta Best practice IAM e l'argomento relativo alla creazione di un gruppo e un utente admin nella Guida per l'utente IAM.

- Utente IAM: un utente IAM è un'identità all'interno del tuo account che dispone di autorizzazioni specifiche, ad esempio l'autorizzazione a inviare i dati degli eventi a una destinazione in EventBridge. Puoi usare credenziali di accesso IAM per effettuare l'accesso a pagine Web AWS sicure, come la AWS Management Console, i forum di discussione AWS o il Centro di AWS Support.

  Inoltre, puoi generare le chiavi di accesso per ogni utente. Puoi utilizzare queste chiavi per accedere ai servizi AWS a livello di codice per firmare crittograficamente la richiesta tramite uno degli SDK o utilizzando l'AWS Command Line Interface (AWS CLI). Se non utilizzi gli strumenti AWS, devi firmare la richiesta tu stesso con Signature Version 4, un protocollo di autenticazione delle richieste API in entrata. Per ulteriori informazioni sulle richieste di autenticazione, consulta la pagina relativa al processo di firma Signature Version 4 nella Riferimenti generali di Amazon Web Services.

- Ruolo IAM: un ruolo IAM è un'altra identità IAM che puoi creare nell'account che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Utilizzando un ruolo IAM, puoi ottenere chiavi di accesso temporanee per accedere a servizi e risorse AWS. I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:
  - Accesso di utenti federati: anziché creare un utente, puoi utilizzare identità di AWS Directory Service, la tua directory di utente aziendale oppure un gestore dell'identità digitale. Questi sono noti come utenti federati. AWS assegna un ruolo a un utente federato quando l'utente richiede l'accesso tramite un gestore dell'identità. Per ulteriori informazioni sugli utenti federati, consulta la sezione relativa a utenti federati e ruoli nella Guida per l'utente di IAM.
  - Accesso multi-account: puoi utilizzare un ruolo IAM nel tuo account per concedere a un altro account l'autorizzazione per accedere alle risorse del tuo account. Per un esempio, consulta il tutorial sulla delega dell'accesso tra account AWS tramite ruoli IAM nella guida per l'utente IAM.
  - Accesso al servizio AWS: puoi utilizzare un ruolo IAM nel tuo account per concedere a un servizio AWS l'autorizzazione per accedere alle risorse del tuo account. Ad esempio, puoi creare un ruolo che consente ad Amazon Redshift di caricare i dati archiviati in un bucket Amazon S3 di un cluster Amazon Redshift. Per ulteriori informazioni, consulta Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS nella Guida per l'utente di IAM.
  - Applicazioni in esecuzione su Amazon EC2: per le applicazioni Amazon EC2 che richiedono l'accesso EventBridge a, puoi archiviare le chiavi di accesso nell'istanza EC2 oppure utilizzare un ruolo IAM per gestire le credenziali temporanee. Per assegnare un ruolo AWS a un'istanza EC2, crei un profilo dell'istanza che viene associato all'istanza. Un profilo dell'istanza contiene il ruolo e fornisce credenziali temporanee alle applicazioni in esecuzione nell'istanza EC2. Per ulteriori

informazioni, consulta la sezione relativa all'utilizzo di ruoli per le applicazioni su Amazon EC2 nella guida per l'utente IAM.

Controllo accessi

Per creare o accedere alle EventBridge risorse, sono necessarie credenziali e autorizzazioni valide. Ad esempio, per richiamare AWS Lambda, Amazon Simple Notification Service (Amazon SNS) e Amazon Simple Queue Service (Amazon SQS), devi disporre di autorizzazioni per tali servizi.

Gestione delle autorizzazioni di accesso alle risorse Amazon EventBridge

La gestione dell'accesso alle risorse EventBridge come regole o eventi avviene mediante policy basate su identità o basate su risorse.

Risorse EventBridge

Alle risorse e alle risorse secondarie EventBridge sono associati ARN univoci. Gli ARN sono utilizzati in EventBridge per creare modelli di eventi. Per ulteriori informazioni sugli ARN, consulta Amazon Resource Name (ARN) e spazi dei nomi dei servizi AWS nella Riferimenti generali di Amazon Web Services.

Per un elenco delle operazioni fornite da EventBridge per l'utilizzo delle risorse, consulta Riferimento alle autorizzazioni di Amazon EventBridge.

> **Note**
>
> La maggior parte dei servizi di AWS trattano i due punti (`:`) o la barra (`/`) come stesso carattere negli ARN. Tuttavia, EventBridge utilizza una corrispondenza esatta nei modelli di eventi e nelle regole. Utilizzare i caratteri ARN corretti durante la creazione di modelli di eventi, facendo in modo che corrispondano alla sintassi ARN nell'evento da far corrispondere.

Nella tabella riportata di seguito sono indicate le risorse in EventBridge.

Tipo di risorsa	Formato ARN
Archive (Archivia)	`arn:aws:events:` *`region`*`:`*`account`*`:archive/` *`archive-name`*
Riproduci di nuovo	`arn:aws:events:` *`region`*`:`*`account`*`:replay/`*`replay-name`*
Regola	`arn:aws:events:` *`region`*`:`*`account`*`:rule/[event-bus-name]/`*`rule-name`*
Router di eventi	`arn:aws:events:` *`region`*`:`*`account`*`:event-bus/` *`event-bus-name`*

Tipo di risorsa	Formato ARN
Tutte le risorse EventBridge	`arn:aws:events:*`
Tutte le risorse EventBridge di proprietà dell'account specificato nella Regione specifica ta	`arn:aws:events: region:account:*`

L'esempio seguente mostra come indicare una regola specifica (*myRule*) nell'istruzione utilizzando il relativo ARN.

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/myRule"
```

Per specificare tutte le regole appartenenti a un determinato account utilizzando il carattere jolly asterisco (*) come descritto di seguito.

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/*"
```

Per specificare tutte le risorse o se una determinata azione API non supporta gli ARN, utilizza il carattere jolly asterisco (*) nell'elemento `Resource` come descritto di seguito.

```
"Resource": "*"
```

Per specificare più risorse o `PutTargets` in una sola istruzione, separa i relativi ARN con una virgola come mostrato di seguito.

```
"Resource": ["arn1", "arn2"]
```

Proprietà delle risorse

Un account è proprietario delle risorse che include, indipendentemente da chi le crea. Il proprietario delle risorse è l'account dell'entità principale, l'utente root dell'account, un ruolo o un utente IAM che autentica la richiesta per creare la risorsa. Negli esempi seguenti viene illustrato il funzionamento:

- Se utilizzi le credenziali dell'utente root dell'account per creare una regola, il tuo account è il proprietario della risorsa EventBridge.
- Se crei un utente nell'account e concedi a tale utente le autorizzazioni per creare risorse EventBridge, l'utente può creare risorse EventBridge. Tuttavia, l'account AWS, cui appartiene l'utente, è il proprietario delle risorse EventBridge.
- Se crei un ruolo IAM nel tuo account con le autorizzazioni per creare risorse EventBridge, chi può assumere il ruolo può creare risorse EventBridge. Il tuo account, a cui appartiene il ruolo, è il proprietario delle risorse EventBridge.

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 descrive l'utilizzo di IAM nel contesto di EventBridge. 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 policy collegate a un'identità IAM vengono definite policy basate su identità (policy IAM), mentre quelle collegate a una risorsa vengono definite policy basate su risorse. In EventBridge, puoi utilizzare sia policy basate su identità (policy IAM) che policy basate su risorse.

Argomenti

- Policy basate su identità (policy IAM)
- Policy basate su risorse (policy IAM)

Policy basate su identità (policy IAM)

Puoi collegare le policy alle identità IAM. Ad esempio, puoi eseguire le operazioni seguenti:

- Associare una policy di autorizzazione a un utente oppure a un gruppo nell'account: per concedere a un utente l'autorizzazione per visualizzare le regole nella console Amazon CloudWatch, associa una policy di autorizzazione a un utente oppure a un gruppo a cui l'utente appartiene.
- Collega una policy di autorizzazione a un ruolo (assegnazione di autorizzazioni tra account): per concedere autorizzazioni tra più account, è possibile collegare una policy di autorizzazione basata su identità a un ruolo IAM. Ad esempio, l'amministratore nell'account A può creare un ruolo per concedere autorizzazioni multi-account a un account B oppure a un servizio AWS nel modo seguente:
  1. L'amministratore dell'account A crea un ruolo IAM e associa una policy di autorizzazione al ruolo che concede le autorizzazioni per le risorse nell'account A.
  2. L'amministratore dell'account A collega una policy di attendibilità al ruolo, identificando l'account B come principale per tale ruolo.
  3. L'amministratore dell'account B può quindi delegare le autorizzazioni per assumere il ruolo a qualsiasi utente nell'account B. In questo modo, gli utenti nell'account B possono creare risorse nell'account A o accedervi. Il principale nella policy di attendibilità può essere anche un principale del servizio AWS per concedere a un servizio AWS le autorizzazioni necessarie per assumere il ruolo.

  Per ulteriori informazioni sull'uso di IAM per delegare le autorizzazioni, consulta Access Management nella IAM User Guide (Guida per l'utente di IAM).

Puoi creare policy IAM specifiche per limitare le chiamate e le risorse a cui hanno accesso gli utenti nel tuo account e quindi associare tali policy agli utenti. Per ulteriori informazioni su come creare ruoli IAM ed esplorare le istruzioni di policy IAM di esempio per EventBridge, consulta Gestione delle autorizzazioni di accesso alle risorse Amazon EventBridge.

Policy basate su risorse (policy IAM)

Quando una regola viene eseguita in EventBridge, vengono richiamate tutte le destinazioni associate alla regola, il che significa richiamare le funzioni AWS Lambda, pubblicare sugli argomenti Amazon SNS o inoltrare l'evento ai flussi Amazon Kinesis. Per effettuare chiamate API sulle risorse di tua proprietà, EventBridge deve disporre delle autorizzazioni appropriate. Per le risorse Lambda, Amazon SNS e Amazon SQS, EventBridge usa le policy basate su risorse. Per i flussi Kinesis, EventBridge utilizza ruoli IAM.

Per ulteriori informazioni su come creare ruoli IAM ed esplorare istruzioni di esempio di policy basate su risorse per EventBridge, consulta Utilizzo di policy basate su risorse per Amazon EventBridge.

Specificare elementi delle policy: azioni, effetti e principali

Per ogni risorsa EventBridge, EventBridge definisce un set di operazioni API. Per concedere le autorizzazioni per queste operazioni API, EventBridge definisce un set di azioni che puoi specificare in una policy. Alcune operazioni API richiedono autorizzazioni per più azioni al fine di eseguire l'operazione API. Per ulteriori informazioni sulle risorse e sulle operazioni delle API, consulta Risorse EventBridge e Riferimento alle autorizzazioni di Amazon EventBridge.

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 Risorse EventBridge.
- Azione: utilizza parole chiave per identificare le operazioni sulle risorse da consentire o negare. Ad esempio, l'autorizzazione `events:Describe` concede all'utente le autorizzazioni per eseguire l'operazione `Describe`.
- Effetto: specifica allow o deny. Se non concedi esplicitamente (allow) l'accesso a una risorsa, l'accesso viene negato. È anche possibile negare esplicitamente l'accesso a una risorsa, per garantire che un utente non possa accedervi, anche se un'altra policy concede l'accesso.
- Principale - Nelle policy basate su identità (policy IAM), l'utente a cui la policy è collegata è il principale implicito. Per policy basate su risorse, specifichi l'utente, l'account, il servizio o un'altra entità che desideri riceva le autorizzazioni (si applica solo alle policy basate su risorse).

Per informazioni sulla sintassi delle policy IAM e le relative descrizioni, consulta Riferimento alla policy JSON IAM nella Guida per l'utente di IAM.

Per informazioni sulle azioni API di EventBridge e sulle risorse a cui si applicano, consulta Riferimento alle autorizzazioni di Amazon EventBridge.

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 definire le condizioni, si utilizzano chiavi di condizione. Sono disponibili chiavi di condizione AWS e chiavi specifiche di EventBridge che puoi usare come necessario. Per un elenco completo delle chiavi AWS, consulta Chiavi disponibili per le condizioni nella Guida per l'utente di IAM. Per un elenco

completo delle chiavi specifiche di EventBridge, consulta Utilizzo di condizioni di policy IAM per il controllo granulare degli accessi.

Utilizzo di politiche basate sull'identità (politiche IAM) per Amazon EventBridge

Le policy basate su identità sono policy di autorizzazione che vengono associate a identità IAM.

Argomenti

AWS politiche gestite per EventBridge

AWS affronta molti casi d'uso comuni fornendo policy IAM autonome create e amministrate da AWS. Le policy gestite, dette anche predefinite, concedono le autorizzazioni necessarie per casi d'uso comune, in modo da non dover determinare quali autorizzazioni sono necessarie. Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

Le seguenti politiche AWS gestite che puoi allegare agli utenti del tuo account sono specifiche per EventBridge:

- AmazonEventBridgeFullAccess— Garantisce l'accesso completo a EventBridge, inclusi EventBridge Pipes, EventBridge Schemas e EventBridge Scheduler.
- AmazonEventBridgeReadOnlyAccess— Garantisce l'accesso in sola lettura a EventBridge, inclusi EventBridge Pipes, Schemas e Scheduler. EventBridge EventBridge

AmazonEventBridgeFullAccess politica

La AmazonEventBridgeFullAccess politica concede le autorizzazioni per utilizzare tutte le EventBridge azioni, oltre alle seguenti autorizzazioni:

- `iam:CreateServiceLinkedRole`— EventBridge richiede questa autorizzazione per creare il ruolo di servizio nell'account per le destinazioni API. Questa autorizzazione concede solo le autorizzazioni del servizio IAM per creare un ruolo nel tuo account specificamente per le destinazioni API.
- `iam:PassRole`— EventBridge richiede questa autorizzazione per passare un ruolo di invocazione per EventBridge richiamare l'obiettivo di una regola.

- Autorizzazioni Secrets Manager: EventBridge richiede queste autorizzazioni per gestire i segreti nel tuo account quando fornisci credenziali tramite la risorsa di connessione per autorizzare le destinazioni API.

Il codice JSON seguente mostra la politica. AmazonEventBridgeFullAccess

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EventBridgeActions",
            "Effect": "Allow",
            "Action": [
                "events:*",
                "schemas:*",
                "scheduler:*",
                "pipes:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IAMCreateServiceLinkedRoleForApiDestinations",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
AmazonEventBridgeApiDestinationsServiceRolePolicy",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "apidestinations.events.amazonaws.com"
                }
            }
        },

        {
            "Sid": "SecretsManagerAccessForApiDestinations",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:UpdateSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue"
```

```
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:events!*"
        },
        {
            "Sid": "IAMPassRoleAccessForEventBridge",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "events.amazonaws.com"
                }
            }
        },
        {
            "Sid": "IAMPassRoleAccessForScheduler",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        },
        {
            "Sid": "IAMPassRoleAccessForPipes",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "pipes.amazonaws.com"
                }
            }
        }
    ]
}
```

> **Note**
>
> Le informazioni contenute in questa sezione si applicano anche alla policy `CloudWatchEventsFullAccess`. Tuttavia, si consiglia vivamente di utilizzare Amazon EventBridge anziché Amazon CloudWatch Events.

AmazonEventBridgeReadOnlyAccess politica

La AmazonEventBridgeReadOnlyAccess politica concede le autorizzazioni per utilizzare tutte le azioni di lettura EventBridge .

Il codice JSON seguente mostra la politica. AmazonEventBridgeReadOnlyAccess

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "events:DescribeRule",
                "events:DescribeEventBus",
                "events:DescribeEventSource",
                "events:ListEventBuses",
                "events:ListEventSources",
                "events:ListRuleNamesByTarget",
                "events:ListRules",
                "events:ListTargetsByRule",
                "events:TestEventPattern",
                "events:DescribeArchive",
                "events:ListArchives",
                "events:DescribeReplay",
                "events:ListReplays",
                "events:DescribeConnection",
                "events:ListConnections",
                "events:DescribeApiDestination",
                "events:ListApiDestinations",
                "events:DescribeEndpoint",
                "events:ListEndpoints",
                "schemas:DescribeCodeBinding",
                "schemas:DescribeDiscoverer",
                "schemas:DescribeRegistry",
```

```
                "schemas:DescribeSchema",
                "schemas:ExportSchema",
                "schemas:GetCodeBindingSource",
                "schemas:GetDiscoveredSchema",
                "schemas:GetResourcePolicy",
                "schemas:ListDiscoverers",
                "schemas:ListRegistries",
                "schemas:ListSchemas",
                "schemas:ListSchemaVersions",

                "schemas:ListTagsForResource",
                "schemas:SearchSchemas",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:ListSchedules",
                "scheduler:ListScheduleGroups",
                "scheduler:ListTagsForResource",
                "pipes:DescribePipe",
                "pipes:ListPipes",
                "pipes:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Le informazioni contenute in questa sezione si applicano anche alla policy `CloudWatchEventsReadOnlyAccess`. Tuttavia, si consiglia vivamente di utilizzare Amazon EventBridge anziché Amazon CloudWatch Events.

EventBridge Politiche gestite specifiche dello schema

Uno schema definisce la struttura degli eventi a cui vengono inviati. EventBridge EventBridge fornisce schemi per tutti gli eventi generati dai AWS servizi. Sono disponibili le seguenti politiche AWS gestite specifiche per EventBridge Schemas:

- AmazonEventBridgeSchemasServiceRolePolicy
- AmazonEventBridgeSchemasFullAccess

- AmazonEventBridgeSchemasReadOnlyAccess

EventBridge Politiche gestite specifiche per Scheduler

Amazon EventBridge Scheduler è uno strumento di pianificazione senza server che consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato. Per le policy AWS gestite specifiche di EventBridge Scheduler, consulta le politiche AWS gestite per Scheduler nella EventBridge Scheduler User Guide. EventBridge

EventBridge Politiche gestite specifiche per Pipes

Amazon EventBridge Pipes collega le sorgenti di eventi alle destinazioni. Pipes riduce la necessità di conoscenze specialistiche e codice di integrazione per lo sviluppo di architetture basate su eventi. Ciò aiuta a garantire la coerenza tra le applicazioni dell'azienda. Sono disponibili le seguenti politiche AWS gestite specifiche per EventBridge Pipes:

- AmazonEventBridgePipesFullAccess

  Fornisce accesso completo ad Amazon EventBridge Pipes.

  > **Note**
  >
  > Questa policy prevede`iam:PassRole`: EventBridge Pipes richiede questa autorizzazione per passare un ruolo di invocazione EventBridge per creare e avviare pipe.

- AmazonEventBridgePipesReadOnlyAccess

  Fornisce accesso in sola lettura ad Amazon EventBridge Pipes.

- AmazonEventBridgePipesOperatorAccess

  Fornisce l'accesso in sola lettura e all'operatore (ovvero la possibilità di interrompere e avviare Pipes) ad Amazon EventBridge Pipes.

Ruoli IAM per l'invio di eventi

Per inoltrare gli eventi agli obiettivi, è EventBridge necessario un ruolo IAM.

Per creare un ruolo IAM per l'invio di eventi a EventBridge

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

2. Per creare un ruolo IAM, segui i passaggi descritti in Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente IAM. e prendi nota di quanto segue:

   - In Nome ruolo, utilizza un nome univoco nel tuo account.
   - In Seleziona tipo di ruolo, scegli Ruoli di AWS servizio, quindi scegli Amazon EventBridge. Ciò concede EventBridge le autorizzazioni per assumere il ruolo.
   - In Allega politica, scegli. AmazonEventBridgeFullAccess

Puoi anche creare policy IAM personalizzate per consentire EventBridge autorizzazioni per azioni e risorse. Puoi associare queste policy personalizzate agli utenti o ai gruppi IAM che richiedono tali autorizzazioni. Per ulteriori informazioni sulle policy IAM, consulta Panoramica delle policy IAM nella Guida per l'utente di IAM. Per ulteriori informazioni sulla gestione e sulla creazione di policy IAM personalizzate, consulta Gestione di policy IAM nella Guida per l'utente di IAM.

Autorizzazioni necessarie per accedere EventBridge agli obiettivi utilizzando i ruoli IAM

EventBridge gli obiettivi in genere richiedono ruoli IAM che concedono l'autorizzazione EventBridge a richiamare l'obiettivo. Di seguito sono riportati alcuni esempi di vari AWS servizi e destinazioni. Per gli altri, usa la EventBridge console per creare una regola e creare un nuovo ruolo che verrà creato con una politica con autorizzazioni ben definite preconfigurate.

Amazon SQS, Amazon SNS, CloudWatch Lambda, Logs EventBridge e le destinazioni bus non utilizzano ruoli e le EventBridge autorizzazioni devono essere concesse tramite una politica delle risorse. Le destinazioni di Gateway API possono utilizzare policy basate su risorse o ruoli IAM.

Se la destinazione è una destinazione API, il ruolo specificato deve includere la policy seguente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "events:InvokeApiDestination" ],
            "Resource": [ "arn:aws:events:::api-destination/*" ]
        }
     ]
}
```

Se la destinazione è un flusso Kinesis, il ruolo utilizzato per inviare dati di eventi alla destinazione deve includere la policy seguente.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

Se la destinazione è il comando run di Systems Manager e specifichi uno o più valori `InstanceIds` per il comando, il ruolo specificato deve includere la policy seguente.

```
{
     "Version": "2012-10-17",
     "Statement": [
           {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ec2:region:accountId:instance/instanceIds",
                     "arn:aws:ssm:region:*:document/documentName"
                ]
           }
     ]
}
```

Se la destinazione è il comando run di Systems Manager e specifichi uno o più tag per il comando, il ruolo specificato deve includere la policy seguente.

```
{
     "Version": "2012-10-17",
     "Statement": [
           {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ec2:region:accountId:instance/*"
                ],
```

```
                "Condition": {
                    "StringEquals": {
                        "ec2:ResourceTag/*": [
                            "[[tagValues]]"
                        ]
                    }
                }
            },
            {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ssm:region:*:document/documentName"
                ]
            }
     ]
}
```

Se la destinazione è una macchina a AWS Step Functions stati, il ruolo specificato deve includere la seguente politica.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "states:StartExecution" ],
            "Resource": [ "arn:aws:states:*:*:stateMachine:*" ]
        }
     ]
}
```

Se la destinazione è un'attività Amazon ECS, il ruolo specificato deve includere la policy seguente.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ecs:RunTask"
        ],
        "Resource": [
            "arn:aws:ecs:*:account-id:task-definition/task-definition-name"
```

```
        ],
        "Condition": {
            "ArnLike": {
                "ecs:cluster": "arn:aws:ecs:*:account-id:cluster/cluster-name"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action":"iam:PassRole",
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringLike": {
                "iam:PassedToService": "ecs-tasks.amazonaws.com"
            }
        }
    }]
}
```

La seguente policy consente agli obiettivi integrati EventBridge di eseguire azioni Amazon EC2 per tuo conto. È necessario utilizzare il AWS Management Console per creare regole con obiettivi predefiniti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TargetInvocationAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:RebootInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "ec2:CreateSnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

La seguente politica consente di EventBridge inoltrare gli eventi agli stream Kinesis del tuo account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KinesisAccess",
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

Esempio di policy gestita dal cliente: utilizzo di tag per controllare l'accesso alle regole

L'esempio seguente mostra una politica utente che concede le autorizzazioni per le azioni. EventBridge Questa politica funziona quando utilizzi l' EventBridge API, gli AWS SDK o il. AWS CLI

Puoi concedere agli utenti l'accesso a EventBridge regole specifiche impedendo loro di accedere ad altre regole. A tale scopo, applichi tag a entrambi i set di regole e usi le policy IAM che fanno riferimento a tali tag. Per ulteriori informazioni sull'etichettatura EventBridge delle risorse, consulta EventBridge Etichette Amazon.

Puoi concedere una policy IAM a un utente per consentirgli di accedere unicamente alle regole con un determinato tag. Puoi scegliere a quali regole concedere l'accesso contrassegnandole con quel particolare tag. Ad esempio, la seguente policy garantisce a un utente l'accesso alle regole con il valore `Prod` per la chiave di tag `Stack`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "events:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Stack": "Prod"
                }
            }
```

```
        }
    ]
}
```

Per ulteriori informazioni sull'utilizzo di istruzioni di policy IAM, consulta Controllo dell'accesso tramite le policy nella Guida per l'utente di IAM.

EventBridge Aggiornamenti Amazon alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti alle politiche AWS gestite EventBridge 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 EventBridge documenti.

Modifica	Descrizione	Data
AmazonEventBridgeFullAccess— Politica aggiornata	AWS GovCloud (US) Regions solo La seguente autorizzazione non è inclusa, in quanto non viene utilizzata: • `iam:CreateServiceLinkedRole` autorizzazione per EventBridge Schema Registry	9 maggio 2024
AmazonEventBridgeSchemasFullAccess— Politica aggiornata	AWS GovCloud (US) Regions solo La seguente autorizzazione non è inclusa, in quanto non viene utilizzata: • `iam:CreateServiceLinkedRole` autorizzazione per EventBridge Schema Registry	9 maggio 2024

Modifica	Descrizione	Data
AmazonEventBridgePipesFullAccess— Aggiunta una nuova politica	EventBridge aggiunta una politica gestita per le autorizzazioni complete per l'utilizzo di EventBridge Pipes.	1 dicembre 2022
AmazonEventBridgePipesReadOnlyAccess— Aggiunta una nuova politica	EventBridge aggiunta una politica gestita per le autorizzazioni alla visualizzazione delle risorse informative di EventBridge Pipes.	1 dicembre 2022
AmazonEventBridgePipesOperatorAccess— Aggiunta una nuova politica	EventBridge è stata aggiunta una politica gestita per le autorizzazioni alla visualizzazione delle informazioni sui EventBridge tubi, nonché all'avvio e all'arresto delle pipe in esecuzione.	1 dicembre 2022
AmazonEventBridgeFullAccess: aggiornamento a una policy esistente	EventBridge ha aggiornato la politica per includere le autorizzazioni necessarie per l'utilizzo delle funzionalità di EventBridge Pipes.	1 dicembre 2022

Modifica	Descrizione	Data
AmazonEventBridgeReadOnlyAccess: aggiornamento a una policy esistente	EventBridge ha aggiunto i permessi necessari per visualizzare le risorse informative di EventBridge Pipes. Sono state aggiunte le seguenti azioni: • `pipes:DescribePipe` • `pipes:ListPipes` • `pipes:ListTagsForResource`	1 dicembre 2022
CloudWatchEventsReadOnlyAccess: aggiornamento a una policy esistente	Aggiornato per corrispondere AmazonEventBridgeReadOnlyAccess.	1 dicembre 2022
CloudWatchEventsFullAccess: aggiornamento a una policy esistente	Aggiornato per corrispondere AmazonEventBridgeFullAccess.	1 dicembre 2022

Modifica	Descrizione	Data
AmazonEventBridgeFullAccess: aggiornamento a una policy esistente	EventBridge ha aggiornato la politica per includere le autorizzazioni necessarie per l'utilizzo degli schemi e delle funzionalità di pianificazione. Sono state aggiunte le seguenti autorizzazioni: • EventBridge Azioni del registro dello schema • EventBridge Azioni dello scheduler • `iam:CreateServiceLinkedRole` autorizzazione per EventBridge Schema Registry • `iam:PassRole` autorizzazione per EventBridge Scheduler	10 novembre 2022

Modifica	Descrizione	Data
AmazonEventBridgeReadOnlyAccess: aggiornamento a una policy esistente	EventBridge ha aggiunto i permessi necessari per visualizzare le risorse informative dello schema e dello scheduler. Sono state aggiunte le seguenti azioni: • `schemas:DescribeCodeBinding` • `schemas:DescribeDiscoverer` • `schemas:DescribeRegistry` • `schemas:DescribeSchema` • `schemas:ExportSchema` • `schemas:GetCodeBindingSource` • `schemas:GetDiscoveredSchema` • `schemas:GetResourcePolicy` • `schemas:ListDiscoverers` • `schemas:ListRegistries` • `schemas:ListSchemas` • `schemas:ListSchemaVersions`	10 novembre 2022

Modifica	Descrizione	Data
	• `schemas:ListTagsForResource` • `schemas:SearchSchemas` • `scheduler:GetSchedule` • `scheduler:GetScheduleGroup` • `scheduler:ListSchedules` • `scheduler:ListScheduleGroups` • `scheduler:ListTagsForResource`	
AmazonEventBridgeReadOnlyAccess: aggiornamento a una policy esistente	EventBridge ha aggiunto le autorizzazioni necessarie per visualizzare le informazioni sugli endpoint. Sono state aggiunte le seguenti azioni: • `events:ListEndpoints` • `events:DescribeEndpoint`	7 aprile 2022

Modifica	Descrizione	Data
AmazonEventBridgeReadOnlyAccess: aggiornamento a una policy esistente	EventBridge ha aggiunto le autorizzazioni necessarie per visualizzare le informazioni sulla connessione e sulla destinazione dell'API. Sono state aggiunte le seguenti azioni: • `events:DescribeConnection` • `events:ListConnections` • `events:DescribeApiDestination` • `events:ListApiDestinations`	4 marzo 2021

Modifica	Descrizione	Data
AmazonEventBridgeFullAccess: aggiornamento a una policy esistente	EventBridge ha aggiornato la politica per includerla `iam:CreateServiceLinkedRole` e AWS Secrets Manager le autorizzazioni necessarie per l'utilizzo delle destinazioni API. Sono state aggiunte le seguenti azioni: • `secretsmanager:CreateSecret` • `secretsmanager:UpdateSecret` • `secretsmanager:DeleteSecret` • `secretsmanager:GetSecretValue` • `secretsmanager:PutSecretValue`	4 marzo 2021
EventBridge ha iniziato a tenere traccia delle modifiche	EventBridge ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	4 marzo 2021

Utilizzo di policy basate su risorse per Amazon EventBridge

Quando una regola viene eseguita in EventBridge, vengono richiamate tutte le destinazioni associate alla regola. Le regole possono richiamare funzioni AWS Lambda, pubblicare su argomenti Amazon SNS o inoltrare l'evento a flussi Kinesis. Per effettuare chiamate API alle risorse di tua proprietà, EventBridge necessita delle autorizzazioni appropriate. Per le risorse Lambda, Amazon SNS, Amazon SQS e File di log Amazon CloudWatch, EventBridge utilizza le policy basate su risorse. Per i flussi Kinesis, EventBridge utilizza le policy basate su identità.

L'AWS CLI ti consente di aggiungere autorizzazioni alle tue destinazioni. Per ulteriori informazioni su come installare e configurare l'AWS CLI, consulta Eseguire la configurazione con l'AWS Command Line Interface nella Guida per l'utente di AWS Command Line Interface.

Argomenti

Autorizzazioni di Gateway Amazon API

Per richiamare il tuo endpoint Gateway Amazon API utilizzando una regola EventBridge, aggiungi la seguente autorizzazione alla policy del tuo endpoint Gateway API.

```
{
  "Version": "2012-10-17",
  "Statement": [
       {
           "Effect": "Allow",
           "Principal": {
               "Service": "events.amazonaws.com"
           },
           "Action": "execute-api:Invoke",
           "Condition": {
               "ArnEquals": {
                   "aws:SourceArn": "arn:aws:events:region:account-id:rule/rule-name"
```

```
                }
            },
            "Resource": [
                "execute-api:/stage/GET/api"
            ]
        }
    ]
}
```

Autorizzazioni per File di log CloudWatch

Quando File di log CloudWatch è la destinazione di una regola, EventBridge crea flussi di log e File di log CloudWatch archivia il testo degli eventi come voci di log. Per consentire a EventBridge di creare il flusso di log e registrare gli eventi, File di log CloudWatch deve includere una policy basata su risorse che consenta a EventBridge di scrivere su File di log Amazon CloudWatch.

Se utilizzi AWS Management Console per aggiungere File di log CloudWatch come destinazione di una regola, la policy basata su risorse viene creata automaticamente. Se utilizzi AWS CLI per aggiungere la destinazione, devi creare questa policy se non esiste già.

L'esempio seguente consente a EventBridge di scrivere su tutti i gruppi di log i cui nomi iniziano con `/aws/events/`. Se utilizzi una policy di denominazione differente per questi tipi di log, modifica l'esempio di conseguenza.

```
{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": ["events.amazonaws.com", "delivery.logs.amazonaws.com"]
            },
            "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
            "Sid": "TrustEventsToStoreLogEvent"
        }
    ],
    "Version": "2012-10-17"
}
```

Per ulteriori informazioni, consulta PutResourcePolicy in Riferimento API di File di log Amazon CloudWatch.

Autorizzazioni di AWS Lambda

Per richiamare la funzione AWS Lambda utilizzando una regola EventBridge, aggiungi l'autorizzazione seguente alla policy della funzione Lambda.

```
{
  "Effect": "Allow",
  "Action": "lambda:InvokeFunction",
  "Resource": "arn:aws:lambda:region:account-id:function:function-name",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:events:region:account-id:rule/rule-name"
    }
  },
  "Sid": "InvokeLambdaFunction"
}
```

Per aggiungere le autorizzazioni precedenti che consentono a EventBridge di richiamare le funzioni Lambda tramite l'AWS CLI

- Al prompt dei comandi, immetti il comando seguente.

  ```
  aws lambda add-permission --statement-id "InvokeLambdaFunction" \
  --action "lambda:InvokeFunction" \
  --principal "events.amazonaws.com" \
  --function-name "arn:aws:lambda:region:account-id:function:function-name" \
  --source-arn "arn:aws:events:region:account-id:rule/rule-name"
  ```

Per ulteriori informazioni sull'impostazione di autorizzazioni che consentono a EventBridge di richiamare funzioni Lambda, consulta AddPermission e Utilizzo di Lambda con eventi pianificati nella Guida per gli sviluppatori di AWS Lambda.

Autorizzazioni di Amazon SNS

Per consentire a EventBridge di pubblicare su un argomento Amazon SNS, utilizza i comandi `aws sns get-topic-attributes` e `aws sns set-topic-attributes`.

> **Note**
>
> Non puoi utilizzare blocchi `Condition` nelle policy degli argomenti Amazon SNS per EventBridge.

Per aggiungere autorizzazioni che consentono a EventBridge di pubblicare argomenti SNS

1. Per elencare gli attributi di un argomento SNS, utilizza il comando seguente.

   ```
   aws sns get-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name"
   ```

   L'esempio seguente mostra il risultato di un nuovo argomento SNS.

   ```
   {
       "Attributes": {
           "SubscriptionsConfirmed": "0",
           "DisplayName": "",
           "SubscriptionsDeleted": "0",
           "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":
   {\"minDelayTarget\":20,\"maxDelayTarget\":20,\"numRetries\":3,\"numMaxDelayRetries
   \":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear
   \"},\"disableSubscriptionOverrides\":false}}",
           "Owner": "account-id",
           "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"__default_policy_ID\",
   \"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
   \":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes
   \",\"SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",
   \"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Publish\"],\"Resource
   \":\"arn:aws:sns:region:account-id:topic-name\",\"Condition\":{\"StringEquals\":
   {\"AWS:SourceOwner\":\"account-id\"}}}]}",
           "TopicArn": "arn:aws:sns:region:account-id:topic-name",
           "SubscriptionsPending": "0"
       }
   }
   ```

2. Utilizza un convertitore da JSON a stringa per convertire la seguente istruzione in una stringa.

```
{
  "Sid": "PublishEventsToMyTopic",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "arn:aws:sns:region:account-id:topic-name"
}
```

Dopo la conversione dell'istruzione in una stringa, la stringa dovrebbe risultare simile a quanto segue:

```
{\"Sid\":\"PublishEventsToMyTopic\",\"Effect\":\"Allow\",\"Principal\":
{\"Service\":\"events.amazonaws.com\"},\"Action\":\"sns:Publish\",\"Resource\":
\"arn:aws:sns:region:account-id:topic-name\"}
```

3. Aggiungi la stringa creata nel passaggio precedente alla raccolta `"Statement"` nell'attributo `"Policy"`.
4. Per impostare la nuova policy, utilizza il comando `aws sns set-topic-attributes`.

```
aws sns set-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name"
 \
--attribute-name Policy \
--attribute-value "{\"Version\":\"2012-10-17\",\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes
\",\"SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",
\"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Publish\"],\"Resource
\":\"arn:aws:sns:region:account-id:topic-name\",\"Condition\":{\"StringEquals
\":{\"AWS:SourceOwner\":\"account-id\"}}}, {\"Sid\":\"PublishEventsToMyTopic\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},\"Action
\":\"sns:Publish\",\"Resource\":\"arn:aws:sns:region:account-id:topic-name\"}]}"
```

Per ulteriori informazioni, vedi l'azione SetTopicAttributes in Riferimento API di Amazon Simple Notification Service.

Autorizzazioni di Amazon SQS

Per consentire a una regola EventBridge di richiamare una coda Amazon SQS, utilizza i comandi `aws sqs get-queue-attributes` e `aws sqs set-queue-attributes`.

Se la policy per la coda SQS è vuota, devi prima creare una policy e poi aggiungervi l'istruzione di autorizzazione. Una nuova coda SQS ha una policy vuota.

Se la coda SQS ha già una policy, devi copiare la policy originale e combinarla con una nuova istruzione per aggiungervi l'istruzione di autorizzazione.

Per aggiungere autorizzazioni che consentono alle regole EventBridge di richiamare una coda SQS

1. Per elencare gli attributi della coda SQS, Al prompt dei comandi, immetti il comando seguente.

```
aws sqs get-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attribute-names Policy
```

2. Aggiungi l'istruzione seguente.

```
{
      "Sid": "AWSEvents_custom-eventbus-ack-sqs-rule_dlq_sqs-rule-target",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:region:account-id:queue-name",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:events:region:account-id:rule/bus-name/rule-
name"
        }
      }
    }
```

3. Utilizza un convertitore da JSON a stringa per convertire l'istruzione precedente in una stringa. Dopo la conversione della policy in una stringa, la stringa dovrebbe risultare simile a quanto segue.

```
{\"Sid\": \"EventsToMyQueue\", \"Effect\": \"Allow\", \"Principal\": {\"Service
\": \"events.amazonaws.com\"}, \"Action\": \"sqs:SendMessage\", \"Resource\":
 \"arn:aws:sqs:region:account-id:queue-name\", \"Condition\": {\"ArnEquals\":
 {\"aws:SourceArn\": \"arn:aws:events:region:account-id:rule/rule-name\"}}
```

4. Crea un file denominato `set-queue-attributes.json`, con il seguente contenuto:

```
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"arn:aws:sqs:region:account-
id:queue-name/SQSDefaultPolicy\",\"Statement\":[{\"Sid\": \"EventsToMyQueue\",
 \"Effect\": \"Allow\", \"Principal\": {\"Service\": \"events.amazonaws.com\"},
 \"Action\": \"sqs:SendMessage\", \"Resource\": \"arn:aws:sqs:region:account-
id:queue-name\", \"Condition\": {\"ArnEquals\": {\"aws:SourceArn\":
 \"arn:aws:events:region:account-id:rule/rule-name\"}}}]}"
}
```

5. Imposta l'attributo della policy utilizzando il file `set-queue-attributes.json` appena creato come input, come mostrato nel comando seguente.

```
aws sqs set-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attributes file://set-queue-attributes.json
```

Per ulteriori informazioni, vedi Esempi di policy di Amazon SQS nella Guida per gli sviluppatori di Amazon Simple Queue Service.

Specifiche di EventBridge Pipes

EventBridge Pipes non supporta policy basate su risorse e non dispone di API che supportino le condizioni delle policy basate su risorse.

Prevenzione del confused deputy tra servizi

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. In AWS, la rappresentazione cross-service può comportare il problema confused deputy. 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.

Ti consigliamo di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` o `aws:SourceAccount` nelle policy basate su risorse per limitare le autorizzazioni che Amazon EventBridge fornisce a un altro servizio per la risorsa. Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

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 entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni.

Router di eventi

Per le destinazioni delle regole del router di eventi EventBridge, il valore di `aws:SourceArn` deve essere l'ARN della regola.

L'esempio seguente mostra il modo in cui puoi utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` in EventBridge per prevenire il problema confused deputy. Questo esempio è destinato all'uso in una policy di attendibilità dei ruoli, per un ruolo utilizzato da una regola EventBridge.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
    ],
```

```
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:events:*:123456789012:rule/myRule"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

EventBridge Pipes

Per EventBridge Pipes, il valore di `aws:SourceArn` deve essere l'ARN della pipe.

L'esempio seguente mostra il modo in cui puoi utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` in EventBridge per prevenire il problema confused deputy. Questo esempio è destinato all'uso in una policy di attendibilità dei ruoli, per un ruolo utilizzato da EventBridge Pipes.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:pipe:*:123456789012::pipe/example"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Policy basate su risorse per Amazon EventBridge Schemas

Il registro di schemi EventBridge supporta policy basate su risorse. Una policy basata su risorse è una policy associata a una risorsa anziché a un'identità IAM. Ad esempio, in Amazon Simple Storage Service (Amazon S3), una policy basata su risorse è associata a un bucket Amazon S3.

Per ulteriori informazioni su EventBridge Schemas e policy basate su risorse, vedi quanto segue.

- Riferimento REST API di Amazon EventBridge Schemas
- Policy basate su identità e policy basate su risorse nella Guida per l'utente di IAM

API supportate per policy basate su risorse

È possibile utilizzare le seguenti API con policy basate su risorse per il registro di schemi EventBridge.

- `DescribeRegistry`
- `UpdateRegistry`
- `DeleteRegistry`
- `ListSchemas`
- `SearchSchemas`
- `DescribeSchema`
- `CreateSchema`
- `DeleteSchema`
- `UpdateSchema`
- `ListSchemaVersions`
- `DeleteSchemaVersion`
- `DescribeCodeBinding`
- `GetCodeBindingSource`
- `PutCodeBinding`

Esempio di policy che concede tutte le azioni supportate a un account AWS

Per il registro di schemi EventBridge, è necessario associare sempre una policy basata su risorse a un registro. Per concedere l'accesso a uno schema, è necessario specificare l'ARN dello schema e l'ARN del registro nella policy.

Per concedere a un utente l'accesso a tutte le API disponibili per EventBridge Schemas, utilizza una policy simile alla seguente, sostituendo `"Principal"` con l'ID dell'account a cui intendi concedere l'accesso.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Test",
            "Effect": "Allow",
            "Action": [
                "schemas:*"
            ],
            "Principal": {
                "AWS": [
                    "109876543210"
                ]
            },
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
            ]
        }
    ]
}
```

Esempio di policy che concede azioni di sola lettura a un account AWS

L'esempio seguente concede l'accesso a un account solo per le API di sola lettura per EventBridge Schemas.

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

```
            "Effect": "Allow",
            "Action": [
                "schemas:DescribeRegistry",
                "schemas:ListSchemas",
                "schemas:SearchSchemas",
                "schemas:DescribeSchema",
                "schemas:ListSchemaVersions",
                "schemas:DescribeCodeBinding",
                "schemas:GetCodeBindingSource"
            ],
            "Principal": {
                "AWS": [
                    "109876543210"
                ]
            },
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
            ]
        }
    ]
}
```

Esempio di policy che concede tutte le azioni a un'organizzazione

È possibile utilizzare policy basate su risorse con il registro di schemi EventBridge per concedere l'accesso a un'organizzazione. Per ulteriori informazioni, consulta la Guida per l'utente di AWS Organizations. L'esempio seguente concede l'accesso al registro di schemi all'organizzazione con ID `o-a1b2c3d4e5`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Test",
            "Effect": "Allow",
            "Action": [
                "schemas:*"
            ],
            "Principal": "*",
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-a1b2c3d4e5"
                    ]
                }
            }
        }
    ]
}
```

Riferimento alle autorizzazioni di Amazon EventBridge

Per specificare un'azione in una policy EventBridge, utilizza il prefisso `events:` seguito dal nome dell'operazione API, come mostrato nel seguente esempio.

```
"Action": "events:PutRule"
```

Per specificare più operazioni in una singola istruzione, separarle con una virgola come mostrato di seguito.

```
"Action": ["events:action1", "events:action2"]
```

Per specificare più azioni, è possibile utilizzare i caratteri jolly. Ad esempio, puoi specificare tutte le azioni il cui nome inizia con la parola `"Put"` come segue.

```
"Action": "events:Put*"
```

Per specificare tutte le azioni API di EventBridge, utilizza il carattere jolly `*` come mostrato di seguito.

```
"Action": "events:*"
```

La tabella seguente elenca le operazioni API di EventBridge e le azioni corrispondenti che puoi specificare in una policy IAM.

Operazione API di EventBridge	Autorizzazioni richieste	Descrizione
DeleteRule	`events:DeleteRule`	Necessario per eliminare una regola.
DescribeEventBus	`events:DescribeEventBus`	Richiesto per elencare gli account che sono autorizzati a scrivere gli eventi sul bus dell'evento dell'account attuale.

Operazione API di EventBridge	Autorizzazioni richieste	Descrizione
DescribeRule	`events:DescribeRule`	Necessario per elencare i dettagli di una regola.
DisableRule	`events:DisableRule`	Necessario per disabilitare una regola.
EnableRule	`events:EnableRule`	Necessario per abilitare una regola.
ListRuleNamesByTarget	`events:ListRuleNamesByTarget`	Necessario per elencare le regole associate a un target.
ListRules	`events:ListRules`	Necessario per elencare tutte le regole nel tuo account.
ListTagsForResource	`events:ListTagsForResource`	Necessario per elencare tutti i tag associati a una risorsa EventBridge. Al momento, è possibile applicare tag solo alle regole.
ListTargetsByRule	`events:ListTargetsByRule`	Necessario per elencare tutti i target associati a una regola.
PutEvents	`events:PutEvents`	Necessario per aggiungere eventi personalizzati per i quali può essere trovata una corrispondenza alle regole.
PutPermission	`events:PutPermission`	Richiesto per autorizzare un altro account a scrivere eventi su un bus evento predefinito di questo account.
PutRule	`events:PutRule`	Necessario per creare o aggiornare una regola.

Operazione API di EventBridge	Autorizzazioni richieste	Descrizione
PutTargets	`events:PutTargets`	Necessario per aggiungere target a una regola.
RemovePermission	`events:RemovePermission`	Richiesto per revocare a un altro account le autorizzazioni per scrivere eventi su un bus evento predefinito di questo account.
RemoveTargets	`events:RemoveTargets`	Necessario per rimuovere un target da una regola.
TestEventPattern	`events:TestEventPattern`	Necessario per testare un modello di evento in un dato evento.

Utilizzo di condizioni di policy IAM per il controllo granulare degli accessi

Per concedere le autorizzazioni, utilizzi il linguaggio della policy IAM in un'istruzione di policy allo scopo di specificare le condizioni in base alle quale la policy deve essere applicata. Ad esempio, puoi avere una policy che viene applicata solo dopo una data specifica.

Una condizione in una policy è costituita da coppie chiave-valore. Le chiavi di condizione non fanno distinzione tra maiuscole e minuscole.

Se specifichi più condizioni o chiavi in una sola condizione, tutte le condizioni e le chiavi devono essere soddisfatte affinché EventBridge conceda l'autorizzazione. Se specifichi una sola condizione con più valori per una sola chiave, EventBridge concede l'autorizzazione se uno dei valori viene soddisfatto.

Puoi anche utilizzare segnaposto o variabili di policy quando specifichi le condizioni. Per ulteriori informazioni, consulta la pagina relativa alle variabili di policy nella Guida per l'utente di IAM. Per ulteriori informazioni su come specificare le condizioni in un linguaggio di policy IAM, consulta Condition nella Guida per l'utente di IAM.

Per impostazione predefinita, gli utenti e i ruoli IAM non possono accedere agli eventi nel tuo account. Per accedere agli eventi, un utente deve disporre dell'autorizzazione per l'azione API `PutRule`. Se un ruolo o un utente IAM dispone dell'autorizzazione per l'azione `events:PutRule`, può creare una regola corrispondente a determinati eventi. Tuttavia, affinché la regola sia utile, l'utente deve disporre anche delle autorizzazioni per l'azione `events:PutTargets` perché, se la regola non deve solo pubblicare una metrica CloudWatch, devi anche aggiungere una destinazione a una regola.

Puoi fornire una condizione nell'istruzione della policy di un ruolo o utente IAM che consente all'utente o al ruolo di creare una regola che corrisponde solo a un set specifico di origini e tipi di eventi. Per concedere l'accesso a origini e tipi di eventi specifici, utilizza le chiavi di condizione `events:source` e `events:detail-type`.

Analogamente, puoi fornire una condizione nell'istruzione della policy di un ruolo o utente IAM che consente all'utente o al ruolo di creare una regola che corrisponde solo a una specifica risorsa nei tuoi account. Per concedere l'accesso a una risorsa specifica, utilizza la chiave di condizione `events:TargetArn`.

L'esempio seguente è una policy che consente agli utenti di accedere a tutti gli eventi tranne gli eventi Amazon EC2 in EventBridge utilizzando un'istruzione di negazione sull'azione API `PutRule`.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyPutRuleForAllEC2Events",
            "Effect": "Deny",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

Chiavi di condizione EventBridge

La tabella seguente mostra le chiavi di condizione e le coppie chiave/valore che puoi utilizzare in una policy in EventBridge.

Chiave di condizione	Coppia chiave-valore	Tipi di valutazione
aws:SourceAccount	L'account in cui esiste la regola specificata da `aws:SourceArn` .	ID account, Null
aws:SourceArn	L'ARN della regola che invia l'evento.	ARN, Null
events:creatorAccount	`"events:creatorAccount":" creatorAccount "` Per *creatorAccount* , utilizza l'ID dell'account che ha creato la regola. Utilizza questa condizione per autorizzare le chiamate API sulle regole di un account specifico.	creatorAccount, Null
events:detail-type	`"events:detail-type":" detail-type "`	Detail-type, null

Chiave di condizione	Coppia chiave-valore	Tipi di valutazione
	Dove *detail-type* è la stringa letterale del campo dell'evento detail-type (tipo di dettaglio), ad esempio `"AWS API Call via CloudTrail"` e `"EC2 Instance State-change Notification"` .	
events: detail.eventTypeCode	`"events:detail.eventTypeCode":"` *eventTypeCode* `"` In *eventTypeCode* , utilizza la stringa letterale per il campo detail.eventTypeCode dell'evento, ad esempio `"AWS_ABUSE_DOS_REPORT"` .	eventTypeCode, Null
events: detail.service	`"events:detail.service":"` *service* `"` Per *service*, utilizza la stringa letterale per il campo detail.service dell'evento, ad esempio `"ABUSE"`.	service, Null

Chiave di condizione	Coppia chiave-valore	Tipi di valutazione
events: detail.userIdentity.principalId	`"events:detail.userIdentity.principalId":" principal-id "` Per *principal-id* , utilizza la stringa letterale per il campo detail.userIdentity.principalId dell'evento con detail-type `"AWS API Call via CloudTrail"` , ad esempio `"AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName."` .	Principal Id, null
events:eventBusInvocation	`"events:eventBusInvocation":" boolean"` Per *boolean*, usa true quando una regola invia un evento ad una destinazione che è un router di eventi in un altro account. Utilizza false quando viene utilizzata una chiamata API `PutEvents` .	eventBusInvocation, Null
events:ManagedBy	Utilizzata internamente dai servizi AWS. Per una regola creata automaticamente da un servizio AWS, il valore è il nome del principale del servizio che ha creato la regola	Non destinata all'uso nelle policy dei clienti.

Chiave di condizione	Coppia chiave-valore	Tipi di valutazione
events:source	`"events:source":" source "` Utilizza *source* per la stringa letterale del campo di origine dell'evento, ad esempio `"aws.ec2"` e `"aws.s3"`. Per visualizzare più valori possibili per *source*, consulta gli eventi di esempio in Eventi derivanti dai servizi AWS.	Source, null
events:TargetArn	`"events:TargetArn" :" target-arn "` Per *target-arn* , usa l'ARN della destinazione per la regola, ad esempio `"arn:aws:lambda:*:*:function:*"` .	ArrayOfARN, Null

Per alcuni esempi di istruzioni di policy per EventBridge, consulta Gestione delle autorizzazioni di accesso alle risorse Amazon EventBridge.

Argomenti

- Specifiche di EventBridge Pipes
- Esempio: utilizzo della condizione creatorAccount
- Esempio: utilizzo della condizione eventBusInvocation
- Esempio: limitazione dell'accesso a un'origine specifica
- Esempio: definizione di più origini che possono essere utilizzate individualmente in un modello di eventi
- Esempio: definizione di un'origine e di DetailType che possono essere utilizzati in un modello di eventi
- Esempio: accertarsi che l'origine sia definita nel modello di eventi

- Esempio: definizione di un elenco di origini consentite in un modello di eventi con più origini
- Esempio: limitazione dell'accesso PutRule mediante detail.service
- Esempio: limitazione dell'accesso PutRule mediante detail.eventTypeCode
- Esempio: garantire che siano consentiti solo gli eventi AWS CloudTrail per chiamate API provenienti da un determinato PrincipalId
- Esempio: limitazione dell'accesso alle destinazioni

Specifiche di EventBridge Pipes

EventBridge Pipes non supporta nessun'altra chiave di condizione di policy IAM.

Esempio: utilizzo della condizione **`creatorAccount`**

L'esempio seguente di istruzione di policy mostra come utilizzare la condizione `creatorAccount` in una policy per consentire la creazione di regole solo se l'account specificato come `creatorAccount` è l'account che ha creato la regola.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForOwnedRules",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "events:creatorAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
}
```

Esempio: utilizzo della condizione **`eventBusInvocation`**

`eventBusInvocation` indica se l'invocazione proviene da una destinazione multi-account o da una richiesta API `PutEvents`. Il valore è true quando l'invocazione risulta da una regola che include

una destinazione multi-account, ad esempio quando la destinazione è un router di eventi in un altro account. Il valore è false quando l'invocazione risulta da una richiesta API `PutEvents`. L'esempio seguente indica un'invocazione da una destinazione multi-account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountInvocationEventsOnly",
      "Effect": "Allow",
      "Action": "events:PutEvents",
      "Resource": "*",
      "Condition": {
        "BoolIfExists": {
          "events:eventBusInvocation": "true"
        }
      }
    }
  ]
}
```

Esempio: limitazione dell'accesso a un'origine specifica

Le seguenti policy di esempio possono essere associate a un utente IAM. La policy A consente all'azione API `PutRule` per tutti gli eventi, mentre la policy B consente `PutRule` solo se il modello di eventi della regola creata corrisponde agli eventi Amazon EC2.

Policy A: consente tutti gli eventi

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEvents",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*"
        }
    ]
    }
```

Policy B: consente solo eventi da Amazon EC2

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEC2Events",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

`EventPattern` è un argomento obbligatorio per `PutRule`. Pertanto, se l'utente con la policy B chiama `PutRule` con un modello di eventi come il seguente:

```
{
    "source": [ "aws.ec2" ]
}
```

La regola si crea perché la policy consente questa origine specifica, vale a dire `"aws.ec2"`. Tuttavia, se l'utente con la policy B chiama `PutRule` con un modello di eventi come il seguente, la creazione della regola viene negata perché la policy non consente questa origine specifica, ovvero `"aws.s3"`.

```
{
    "source": [ "aws.s3" ]
}
```

Essenzialmente, all'utente con la policy B viene consentita solo la creazione di una regola che corrisponderebbe agli eventi originati da Amazon EC2 e quindi gli è consentito l'accesso solo agli eventi da Amazon EC2.

Consulta la tabella riportata di seguito per un confronto tra la policy A e la policy B:

Modello di eventi	Consentito dalla policy A	Consentito dalla policy B
`{ "source": [ "aws.ec2" ] }`	Sì	Sì
`{ "source": [ "aws.ec2", "aws.s3" ] }`	Sì	No (l'origine aws.s3 non è consentita)
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sì	Sì
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sì	No (deve essere specificata l'origine)

Esempio: definizione di più origini che possono essere utilizzate individualmente in un modello di eventi

La seguente policy consente a un ruolo o utente IAM di creare una regola in cui l'origine in `EventPattern` è Amazon EC2 o Amazon ECS.

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

```
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsEC2OrECS",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": [ "aws.ec2", "aws.ecs" ]
                }
            }
        }
    ]
}
```

Nella tabella seguente sono riportati alcuni esempi di modelli di eventi consentiti o negati da questa policy.

Modello di eventi	Consentito dalla policy
`{ "source": [ "aws.ec2" ] }`	Sì
`{ "source": [ "aws.ecs" ] }`	Sì
`{ "source": [ "aws.s3" ] }`	No
`{ "source": [ "aws.ec2", "aws.ecs" ] }`	No
`{`	No

Modello di eventi	Consentito dalla policy

<pre>{
"detail-type": ["AWS API Call via CloudTrail"]
}</pre>	

Esempio: definizione di un'origine e di **`DetailType`** che possono essere utilizzati in un modello di eventi

La policy seguente consente solo gli eventi provenienti dall'origine `aws.ec2` con `DetailType` uguale a `EC2 instance state change notification`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid":
 "AllowPutRuleIfSourceIsEC2AndDetailTypeIsInstanceStateChangeNotification",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2",
                    "events:detail-type": "EC2 Instance State-change Notification"
                }
            }
        }
    ]
}
```

Nella tabella seguente sono riportati alcuni esempi di modelli di eventi consentiti o negati da questa policy.

Modello di eventi	Consentito dalla policy

<pre>{
"source": ["aws.ec2"]
}</pre>
	No

Modello di eventi	Consentito dalla policy
`{ "source": [ "aws.ecs" ] }`	No
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sì
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance Health Failed" ] }`	No
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	No

Esempio: accertarsi che l'origine sia definita nel modello di eventi

La policy seguente consente agli utenti di creare regole solo con `EventPatterns` che hanno il campo di origine. Con questa policy, un ruolo o un utente IAM non può creare una regola con un `EventPattern` che non fornisce un'origine specifica.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsSpecified",
            "Effect": "Allow",
            "Action": "events:PutRule",
```

```
            "Resource": "*",
            "Condition": {
                "Null": {
                    "events:source": "false"
                }
            }
        }
    ]
}
```

Nella tabella seguente sono riportati alcuni esempi di modelli di eventi consentiti o negati da questa policy.

Modello di eventi	Consentito dalla policy
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sì
`{ "source": [ "aws.ecs", "aws.ec2" ] }`	Sì
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	No

Esempio: definizione di un elenco di origini consentite in un modello di eventi con più origini

La policy seguente consente agli utenti di creare regole con `EventPatterns` che includono molteplici origini. Ogni origine nel modello di eventi deve essere un membro dell'elenco fornito nella

condizione. Quando utilizzi la condizione `ForAllValues`, assicurati che almeno uno degli elementi nell'elenco di condizioni sia definito.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsSpecifiedAndIsEitherS3OrEC2OrBoth",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "events:source": [ "aws.ec2", "aws.s3" ]
                },
                "Null": {
                    "events:source": "false"
                }
            }
        }
    ]
}
```

Nella tabella seguente sono riportati alcuni esempi di modelli di eventi consentiti o negati da questa policy.

Modello di eventi	Consentito dalla policy
`{ "source": [ "aws.ec2" ] }`	Sì
`{ "source": [ "aws.ec2", "aws.s3" ] }`	Sì
`{ "source": [ "aws.ec2", "aws.autoscaling" ]`	No

Modello di eventi	Consentito dalla policy
<pre>}</pre>	

<pre>{
    "detail-type": ["EC2
 Instance State-change Notificat
ion"]
}</pre>
	No

Esempio: limitazione dell'accesso **`PutRule`** mediante **`detail.service`**

Puoi limitare un ruolo o un utente IAM alla creazione di regole solo per eventi che hanno un determinato valore nel campo `events:details.service`. Il valore di `events:details.service` non deve essere necessariamente il nome di un servizio AWS.

Questa condizione di policy risulta utile quando si utilizzano eventi di AWS Health correlati alla sicurezza o a un uso illecito. Utilizzando questa condizione di policy, puoi limitare l'accesso a questi avvisi sensibili solo agli utenti a cui sono destinati.

Ad esempio, la seguente policy consente la creazione di regole solo per gli eventi in cui il valore di `events:details.service` è `ABUSE`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleEventsWithDetailServiceEC2",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail.service": "ABUSE"
                }
            }
        }
    ]
}
```

Esempio: limitazione dell'accesso **PutRule** mediante **detail.eventTypeCode**

Puoi limitare un ruolo o un utente IAM alla creazione di regole solo per eventi che hanno un determinato valore nel campo `events:details.eventTypeCode`. Questa condizione di policy risulta utile quando si utilizzano eventi di AWS Health correlati alla sicurezza o a un uso illecito. Utilizzando questa condizione di policy, puoi limitare l'accesso a questi avvisi sensibili solo agli utenti a cui sono destinati.

Ad esempio, la seguente policy consente la creazione di regole solo per gli eventi in cui il valore di `events:details.eventTypeCode` è `AWS_ABUSE_DOS_REPORT`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleEventsWithDetailServiceEC2",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail.eventTypeCode": "AWS_ABUSE_DOS_REPORT"
                }
            }
        }
    ]
}
```

Esempio: garantire che siano consentiti solo gli eventi AWS CloudTrail per chiamate API provenienti da un determinato **PrincipalId**

Tutti gli eventi AWS CloudTrail hanno il PrincipalId dell'utente che ha effettuato la chiamata API nel percorso `detail.userIdentity.principalId` di un evento. Mediante la chiave di condizione `events:detail.userIdentity.principalId`, puoi limitare l'accesso di utenti o ruoli IAM solo agli eventi CloudTrail che provengono da un account specifico.

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

```
            "Sid": "AllowPutRuleOnlyForCloudTrailEventsWhereUserIsASpecificIAMUser",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail-type": [ "AWS API Call via CloudTrail" ],
                    "events:detail.userIdentity.principalId":
 [ "AIDAJ45Q7YFFAREXAMPLE" ]
                }
            }
        }
    ]
}
```

Nella tabella seguente sono riportati alcuni esempi di modelli di eventi consentiti o negati da questa policy.

Modello di eventi	Consentito dalla policy
`{ "detail-type": [ "AWS API Call via CloudTrail" ] }`	No
`{ "detail-type": [ "AWS API Call via CloudTrail" ], "detail.userIdentity.principalId": [ "AIDAJ45Q7YFFAREXAMPLE" ] }`	Sì
`{ "detail-type": [ "AWS API Call via CloudTrail" ], "detail.userIdentity.principalId": [ "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName" ]`	No

Modello di eventi	Consentito dalla policy
}	

Esempio: limitazione dell'accesso alle destinazioni

Se un utente o ruolo IAM dispone di un'autorizzazione `events:PutTargets`, può aggiungere qualsiasi destinazione nello stesso account alle regole alle quali ha accesso. La seguente policy consente gli utenti di aggiungere destinazioni solo a una regola specifica: `MyRule` nell'account `123456789012`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRule",
            "Effect": "Allow",
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule"
        }
    ]
}
```

Per limitare i target che possono essere aggiunti alla regola, utilizza la chiave di condizione `events:TargetArn`. Puoi limitare le destinazioni alle sole funzioni Lambda, come nel seguente esempio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRuleAndOnlyLambdaFunctions",
            "Effect": "Allow",
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule",
            "Condition": {
                "ArnLike": {
                    "events:TargetArn": "arn:aws:lambda:*:*:function:*"
                }
            }
```

```
        }
    ]
}
```

Utilizzo di ruoli collegati ai servizi per EventBridge

Amazon EventBridge utilizza ruoli collegati al servizio AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente a EventBridge. I ruoli collegati ai servizi sono definiti automaticamente da EventBridge e includono tutte le autorizzazioni richieste dal servizio per eseguire chiamate agli altri servizi AWS per tuo conto.

Argomenti

- Utilizzo dei ruoli per la creazione di segreti per le destinazioni API
- Utilizzo dei ruoli per l'individuazione dello schema

Utilizzo dei ruoli per la creazione di segreti per le destinazioni API

Amazon EventBridge utilizza ruoli collegati al servizio AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente a EventBridge. I ruoli collegati ai servizi sono definiti automaticamente da EventBridge e includono tutte le autorizzazioni richieste dal servizio per eseguire chiamate agli altri servizi AWS per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di EventBridge perché non dovrai più aggiungere manualmente le autorizzazioni necessarie. EventBridge definisce le autorizzazioni dei relativi ruoli associati ai servizi e, salvo diversamente definito, EventBridge potrà assumere solo i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di EventBridge perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Yes (Sì) nella colonna Service-linked roles (Ruoli collegati ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per EventBridge

EventBridgeutilizza il ruolo collegato al servizio denominato AWSServiceRoleForAmazonEventBridgeApiDestinations— Consente l'accesso ai Secrets Manager Secrets creati da. EventBridge

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi AWSServiceRoleForAmazonEventBridgeApiDestinations considera attendibili i seguenti servizi:

- `apidestinations.events.amazonaws.com`

La politica di autorizzazione dei ruoli denominata AmazonEventBridgeApiDestinationsServiceRolePolicy consente di EventBridge completare le seguenti azioni sulle risorse specificate:

- Operazione: `create, describe, update and delete secrets; get and put secret values` su `secrets created for all connections by EventBridge`

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per EventBridge

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei una connessione nellaAWS Management Console, la o l'AWSAPIAWS CLI, EventBridge crea automaticamente il ruolo collegato al servizio.

> **Important**
>
> Questo ruolo collegato al servizio può apparire nell'account, se è stata completata un'operazione in un altro servizio che utilizza le caratteristiche supportate da questo ruolo. Se utilizzavi il EventBridge servizio prima dell'11 febbraio 2021, quando ha iniziato a supportare ruoli collegati al servizio, hai EventBridge creato il AWSServiceRoleForAmazonEventBridgeApiDestinationsruolo nel tuo account. Per ulteriori informazioni, consulta Un nuovo ruolo appare nel mio Account AWS.

Se elimini questo ruolo collegato ai servizi, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei una connessione, EventBridge crea nuovamente il ruolo collegato al servizio per te.

Modifica di un ruolo collegato ai servizi per EventBridge

EventBridge non consente di modificare il ruolo collegato ai servizi AWSServiceRoleForAmazonEventBridgeApiDestinations. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per EventBridge

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 ai servizi prima di poterlo eliminare manualmente.

Pulizia di un ruolo collegato ai servizi

Prima di utilizzare IAM; per eliminare un ruolo collegato al servizio, è necessario prima rimuovere qualsiasi risorsa utilizzata dal ruolo.

> **Note**
>
> Se il servizio EventBridge utilizza tale ruolo quando tenti di eliminare le risorse, è possibile che l'eliminazione non abbia esito positivo. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse di EventBridge utilizzate da AWSServiceRoleForAmazonEventBridgeApiDestinations (console)

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. In Integrazioni scegli le destinazioni API, quindi scegli la scheda Connessioni.
3. Scegli la connessione, quindi scegli Elimina.

Per eliminare le risorse di EventBridge utilizzate da AWSServiceRoleForAmazonEventBridgeApiDestinations (AWS CLI)

- Usa il seguente comando:`delete-connection`.

Per eliminare le risorse di EventBridge utilizzate da AWSServiceRoleForAmazonEventBridgeApiDestinations (API)

- Usa il seguente comando:`DeleteConnection`.

Eliminazione manuale del ruolo collegato ai servizi

Utilizzare la console IAM, AWS CLI, la AWS o l'API per eliminare i ruoli collegati ai servizi AWSServiceRoleForAmazonEventBridgeApiDestinations. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi EventBridge

EventBridge supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta AWSRegioni ed endpoint.

Utilizzo dei ruoli per l'individuazione dello schema

Amazon EventBridge utilizza ruoli collegati al servizio AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente a EventBridge. I ruoli collegati ai servizi sono definiti automaticamente da EventBridge e includono tutte le autorizzazioni richieste dal servizio per eseguire chiamate agli altri servizi AWS per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di EventBridge perché non dovrai più aggiungere manualmente le autorizzazioni necessarie. EventBridge definisce le autorizzazioni dei relativi ruoli associati ai servizi e, salvo diversamente definito, EventBridge potrà assumere solo i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di EventBridge perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Yes (Sì) nella colonna Service-linked roles (Ruoli

collegati ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per EventBridge

EventBridgeutilizza il ruolo collegato al servizio denominato AWSServiceRoleForSchemas— Concede le autorizzazioni alle regole gestite create dagli schemi.. Amazon EventBridge

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi AWSServiceRoleForSchemas considera attendibili i seguenti servizi:

- `schemas.amazonaws.com`

La politica di autorizzazione dei ruoli denominata AmazonEventBridgeSchemasServiceRolePolicyconsente di EventBridge completare le seguenti azioni sulle risorse specificate:

- Operazione: `put, enable, disable, and delete rules; put and remove targets; list targets per rule` su `all managed rules created by EventBridge`

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per EventBridge

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando esegui uno Schema Discovery nella AWS Management ConsoleAWS CLI, la o l'AWSAPI, EventBridge crea automaticamente il ruolo collegato al servizio.

> **Important**
>
> Questo ruolo collegato al servizio può apparire nell'account, se è stata completata un'operazione in un altro servizio che utilizza le caratteristiche supportate da questo ruolo. Se utilizzavi il EventBridge servizio prima del 27 novembre 2019, quando ha iniziato a supportare ruoli collegati al servizio, hai EventBridge creato il AWSServiceRoleForSchemasruolo nel tuo account. Per ulteriori informazioni, consulta Un nuovo ruolo appare nel mio Account AWS.

Se elimini questo ruolo collegato ai servizi, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando esegui uno Schema Discovery, EventBridge crea nuovamente il ruolo collegato al servizio per te.

Modifica di un ruolo collegato ai servizi per EventBridge

EventBridge non consente di modificare il ruolo collegato ai servizi AWSServiceRoleForSchemas. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per EventBridge

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 ai servizi prima di poterlo eliminare manualmente.

Pulizia di un ruolo collegato ai servizi

Prima di utilizzare IAM; per eliminare un ruolo collegato al servizio, è necessario prima rimuovere qualsiasi risorsa utilizzata dal ruolo.

> **Note**
>
> Se il servizio EventBridge utilizza tale ruolo quando tenti di eliminare le risorse, è possibile che l'eliminazione non abbia esito positivo. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse di EventBridge utilizzate da AWSServiceRoleForSchemas (console)

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. In Autobus scegli Event bus, quindi scegli un Event Bus.
3. Scegli Stop discovery.

Per eliminare le risorse di EventBridge utilizzate da AWSServiceRoleForSchemas (AWS CLI)

- Usa il seguente comando:`delete-discoverer`.

Per eliminare le risorse di EventBridge utilizzate da AWSServiceRoleForSchemas (API)

- Usa il seguente comando:`DeleteDiscoverer`.

Eliminazione manuale del ruolo collegato ai servizi

Utilizzare la console IAM, AWS CLI, la AWS o l'API per eliminare i ruoli collegati ai servizi AWSServiceRoleForSchemas. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi EventBridge

EventBridge supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta AWSRegioni ed endpoint.

Registrazione delle chiamate Amazon EventBridge API utilizzando AWS CloudTrail

Amazon EventBridge è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o un Servizio AWS. CloudTrail acquisisce tutte le chiamate API EventBridge come eventi. Le chiamate acquisite includono chiamate dalla EventBridge console e chiamate di codice alle operazioni EventBridge API. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare a quale richiesta è stata effettuata EventBridge, l'indirizzo IP da cui è stata effettuata la richiesta, 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 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 Account AWS quando crei l'account e hai automaticamente accesso alla cronologia degli CloudTrail eventi. La cronologia CloudTrail degli eventi fornisce un record visualizzabile, ricercabile, scaricabile e immutabile degli ultimi 90 giorni di eventi di gestione registrati in un. Regione AWSPer ulteriori informazioni, consulta Lavorare con la cronologia degli CloudTrail eventi nella Guida per l'utente.AWS CloudTrail Non sono CloudTrail previsti costi per la visualizzazione della cronologia degli eventi.

Per una registrazione continua degli eventi degli Account AWS ultimi 90 giorni, crea un trail o un data store di eventi CloudTrailLake.

CloudTrail sentieri

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Tutti i percorsi creati utilizzando il 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 in quanto consente di registrare l'intera attività del proprio account Regioni AWS . Se crei un percorso a regione singola, puoi visualizzare solo gli eventi registrati nel percorso. Regione

AWS Per ulteriori informazioni sui percorsi, consulta Creazione di un percorso per te Account AWS e Creazione di un percorso per un'organizzazione nella Guida per l'AWS CloudTrail 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 la pagina Prezzi.AWS CloudTrail Per informazioni sui prezzi di Amazon S3, consulta Prezzi di Amazon S3.

CloudTrail Archivi di dati sugli eventi di Lake

CloudTrail Lake ti consente di eseguire query basate su SQL sui tuoi eventi. CloudTrail Lake converte gli eventi esistenti in formato JSON basato su righe in formato Apache ORC. ORC è un formato di archiviazione a colonne ottimizzato per il recupero rapido dei dati. Gli eventi vengono aggregati in archivi di dati degli eventi, che sono raccolte di eventi immutabili basate sui criteri selezionati applicando i selettori di eventi avanzati. I selettori applicati a un archivio di dati degli eventi controllano quali eventi persistono e sono disponibili per l'esecuzione della query. Per ulteriori informazioni su CloudTrail Lake, consulta Working with AWS CloudTrail Lake nella Guida per l'utente.AWS CloudTrail

CloudTrail Gli archivi e le richieste di dati sugli eventi di Lake comportano dei costi. Quando crei un datastore di eventi, scegli l'opzione di prezzo da utilizzare per tale datastore. L'opzione di prezzo determina il costo per l'importazione e l'archiviazione degli eventi, nonché il periodo di conservazione predefinito e quello massimo per il datastore di eventi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione Prezzi.AWS CloudTrail

EventBridge eventi relativi ai dati in CloudTrail

Gli eventi di dati forniscono informazioni sulle operazioni delle risorse eseguite su o in una risorsa (ad esempio, lettura o scrittura su un oggetto Amazon S3). Queste operazioni sono definite anche operazioni del piano dei dati. Gli eventi di dati sono spesso attività che interessano volumi elevati di dati. Per impostazione predefinita, CloudTrail non registra gli eventi relativi ai dati. La cronologia CloudTrail degli eventi non registra gli eventi relativi ai dati.

Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione AWS CloudTrail Prezzi.

Puoi registrare gli eventi relativi ai dati per i tipi di EventBridge risorse utilizzando la CloudTrail console o AWS CLI le operazioni CloudTrail dell'API. Per ulteriori informazioni su come registrare gli eventi relativi ai dati, vedere Registrazione degli eventi relativi ai dati con AWS Management Console

e Registrazione degli eventi relativi ai dati con the AWS Command Line Interface nella Guida per l'AWS CloudTrail utente.

La tabella seguente elenca i tipi di EventBridge risorse per i quali è possibile registrare gli eventi relativi ai dati. La colonna Data event type (console) mostra il valore da scegliere dall'elenco Data event type (console) sulla CloudTrail console. La colonna del valore resources.type mostra il `resources.type` valore da specificare durante la configurazione dei selettori di eventi avanzati utilizzando le API o. AWS CLI CloudTrail La CloudTrail colonna Data API loggate mostra le chiamate API registrate per il tipo di risorsa. CloudTrail

Tipo di evento di dati (console)	valore resources.type	API di dati registrate su CloudTrail
Bus per eventi	`AWS::Events::Event Bus`	• DescribeEventBus
Regola del bus degli eventi	`AWS::Events::Rule`	• DeleteRule • DescribeRule • DisableRule • EnableRule • ListRuleNamesByTarget • ListRules • ListTargetsByRule • PutRule • PutTargets • RemoveTargets • TestEventPattern
Tubo	`AWS::Pipes::Pipe`	• CreatePipe • DeletePipe • DescribePipe • ListPipes • StartPipe • StopPipe

Tipo di evento di dati (console)	valore resources.type	API di dati registrate su CloudTrail
		• UpdatePipe

Puoi configurare selettori di eventi avanzati per filtrare `resources.ARN` i campi `eventName`, `readOnly`, e per registrare solo gli eventi che ritieni importanti. Per ulteriori informazioni su questi campi, consulta AdvancedFieldSelectorl'AWS CloudTrail API Reference.

EventBridge eventi di gestione in CloudTrail

Gli eventi di gestione forniscono informazioni sulle operazioni di gestione eseguite sulle risorse dell'azienda Account AWS. Queste operazioni sono definite anche operazioni del piano di controllo (control-plane). Per impostazione predefinita, CloudTrail registra gli eventi di gestione.

Amazon EventBridge registra tutte le operazioni EventBridge del piano di controllo come eventi di gestione. Per un elenco delle operazioni del piano di Amazon EventBridge controllo a cui si EventBridge effettua l'accesso CloudTrail, consulta l'Amazon EventBridge API Reference.

EventBridge esempi di eventi

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'operazione API richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi gli eventi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra un CloudTrail evento che dimostra l'`PutRule`operazione.


```
{
  "eventVersion":"1.03",
  "userIdentity":{
    "type":"Root",
    "principalId":"123456789012",
    "arn":"arn:aws:iam::123456789012:root",
    "accountId":"123456789012",
    "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
    "sessionContext":{
      "attributes":{
      "mfaAuthenticated":"false",
      "creationDate":"2015-11-17T23:56:15Z"
```

```
      }
    }
  },
  "eventTime":"2015-11-18T00:11:28Z",
  "eventSource":"events.amazonaws.com",
  "eventName":"PutRule",
  "awsRegion":"us-east-1",
  "sourceIPAddress":"AWS Internal",
  "userAgent":"AWS CloudWatch Console",
  "requestParameters":{
    "description":"",
    "name":"cttest2",
    "state":"ENABLED",
    "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-
change Notification\"]}",
    "scheduleExpression":""
  },
  "responseElements":{
    "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
  },
  "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
  "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
  "eventType":"AwsApiCall",
  "apiVersion":"2015-10-07",
  "recipientAccountId":"123456789012"
}
```

Per informazioni sul contenuto dei CloudTrail record, consultate il contenuto dei CloudTrail record nella Guida per l'AWS CloudTrail utente.

CloudTrail voci di registro relative alle azioni intraprese da EventBridge Pipes

EventBridge Pipes assume il ruolo IAM fornito durante la lettura di eventi da fonti, l'invocazione di arricchimenti o l'invocazione di obiettivi. Per le CloudTrail voci relative alle azioni intraprese nel tuo account su tutti gli arricchimenti, i target e le fonti Amazon SQS, Kinesis e DynamoDB, i campi e includeranno. `sourceIPAddress` `invokedBy` `pipes.amazonaws.com`

Elemento di CloudTrail log di esempio per tutti gli arricchimenti, i target e le fonti Amazon SQS, Kinesis e DynamoDB

```
{
```

```
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "...",
    "arn": "arn:aws:sts::111222333444:assumed-role/...",
    "accountId": "111222333444",
    "accessKeyId": "...",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "...",
        "arn": "...",
        "accountId": "111222333444",
        "userName": "userName"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-09-22T21:41:15Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "pipes.amazonaws.com"
  },
  "eventTime": ",,,",
  "eventName": "...",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "pipes.amazonaws.com",
  "userAgent": "pipes.amazonaws.com",
  "requestParameters": {
    ...
  },
  "responseElements": null,
  "requestID": "...",
  "eventID": "...",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "...",
  "eventCategory": "Management"
}
```

Per tutte le altre fonti, il `sourceIPAddress` campo delle voci di CloudTrail registro avrà un indirizzo IP dinamico e non dovrebbe essere utilizzato per alcuna integrazione o categorizzazione degli eventi. Inoltre, queste voci non avranno il campo `invokedBy`.

Esempio di voce di CloudTrail registro per tutte le altre fonti

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    ...
  },
  "eventTime": ",,,",
  "eventName": "...",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Python-httplib2/0.8 (gzip)",
}
```

Convalida della conformità in Amazon EventBridge

I revisori di terze parti come SOC, PCI, FedRAMP e HIPAA valutano la sicurezza e la conformità dei servizi AWS nell'ambito di molteplici programmi di conformità AWS.

Per un elenco dei servizi AWS che rientrano nell'ambito di programmi di conformità specifici, consulta Servizi AWS che rientrano nell'ambito del programma di conformità . Per informazioni generali, consulta Programmi di conformità AWS.

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Download dei report in AWS Artifact .

La responsabilità di conformità durante l'utilizzo di EventBridge è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle normative vigenti. AWS fornisce le seguenti risorse per scopi di conformità:

- Guide di configurazione rapida in materia di sicurezza e conformità: procedure e considerazioni relative all'architettura per l'implementazione di ambienti di base incentrati sulla sicurezza e sulla conformità in AWS.
- Whitepaper sulla progettazione di conformità e sicurezza HIPAA: in che modo le aziende possono utilizzare AWS per creare applicazioni conformi ai requisiti HIPAA.
- Risorse per la conformità AWS: una raccolta di cartelle di lavoro e guide.
- Valutazione delle risorse con regole nella Guida per gli sviluppatori di AWS Config: informazioni su come AWS Config valuta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida di settore e normative.
- AWS Security Hub: una visione completa dello stato di sicurezza in AWS che consente di verificare la conformità con standard industriali di sicurezza e best practice.

Resilienza di Amazon EventBridge

L'infrastruttura globale di AWS è basata su Regioni e zone di disponibilità AWS. AWS Le Regioni forniscono più zone di disponibilità fisicamente separate e isolate che sono connesse tramite reti altamente ridondanti, a bassa latenza e velocità di trasmissione effettiva elevata. Con le Zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su Regioni e zone di disponibilità AWS, consulta Infrastruttura globale di AWS.

Sicurezza dell'infrastruttura in Amazon EventBridge

In qualità di servizio gestito, Amazon EventBridge è protetto dalla sicurezza di rete globale di AWS. Per informazioni sui servizi di sicurezza AWS e su come AWS protegge l'infrastruttura, consulta la pagina Sicurezza del cloud AWS. Per progettare l'ambiente AWS utilizzando le best practice per la sicurezza dell'infrastruttura, consulta la pagina Protezione dell'infrastruttura nel Pilastro della sicurezza di AWS Well-Architected Framework.

Per accedere a EventBridge tramite la rete, utilizzi chiamate API pubblicate AWS. I clienti devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

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.

Puoi chiamare queste operazioni API da qualsiasi posizione di rete e puoi utilizzare policy di accesso basate su risorse in EventBridge, che possono includere limitazioni in base all'indirizzo IP di origine. È inoltre possibile utilizzare le policy EventBridge per controllare l'accesso da endpoint Amazon Virtual Private Cloud (Amazon VPC) o VPC specifici. Di fatto, ciò isola l'accesso di rete a una determinata risorsa EventBridge solo dal VPC specifico nella rete AWS.

Analisi della configurazione e delle vulnerabilità in Amazon EventBridge

Configurazione e controllo IT sono una responsabilità condivisa tra AWS e te, il nostro cliente. Per ulteriori informazioni, consulta il modello di responsabilità condivisa di AWS.

Monitoraggio di Amazon EventBridge

EventBridge invia ad Amazon CloudWatch ogni minuto parametri per qualsiasi cosa, dal numero di eventi corrispondenti al numero di volte in cui un target viene richiamato da una regola.

Il seguente video esamina il monitoraggio e il controllo del EventBridge comportamento tramite CloudWatch: Monitoraggio e controllo degli eventi

Argomenti

- EventBridge metriche
- Dimensioni per le metriche EventBridge

EventBridge metriche

Lo spazio dei nomi `AWS/Events` include i parametri descritti di seguito.

Per le metriche che utilizzano Count come unità, Sum e SampleCount tendono ad essere le statistiche più utili.

Le metriche che specificano solo la `RuleName` dimensione si riferiscono al bus eventi predefinito. Le metriche che specificano sia le `EventBusName` `RuleName` dimensioni che si riferiscono a un bus di eventi personalizzato.

Parametro	Descrizione	Dimensioni	Unità
`DeadLetterInvocations`	Il numero di volte in cui una destinazione di una regola non viene richiamata in risposta a un evento. Ciò comprende invocazioni che risulterebbero in una nuova attivazione della stessa regola, causando un loop infinito.	RuleName	Conteggio
`Events`	Il numero di eventi partner importati da. EventBridge	EventSourceName	Conteggio
`FailedInvocations`	Il numero di invocazioni non riuscite in modo definitivo. Non include le invocazioni ripetute	RuleName	Conteggio

Parametro	Descrizione	Dimensioni	Unità
	o riuscite dopo un nuovo tentativo. Non comprende nemmeno invocazioni non riuscite conteggiate in `DeadLetterInvocations` . **Note** EventBridge invia questa metrica solo a CloudWatch se è diversa da zero.		
`Invocations`	Il numero di volte in cui una destinazione viene richiamata da una regola in risposta a un evento. Include le invocazioni riuscite e non riuscite, ma non i tentativi limitati o ripetuti fino a un esito negativo definitivo. Non include `DeadLetterInvocations` . **Note** EventBridge invia questa metrica solo a CloudWatch se non è zero.	Nessuna, RuleName	Conteggio
`InvocationAttempts`	Numero di volte in cui si è EventBridge tentato di invocare un bersaglio.	Nessuno	Conteggio
`InvocationsCreated`	Il numero totale di invocazioni create in risposta a ciascun evento. Questa metrica viene spesso utilizzata per monitorare l'utilizzo del limite di accelerazione di Invocations nelle transazioni per secondo (quota di servizio). EventBridge	Nessuno	Conteggio

Parametro	Descrizione	Dimensioni	Unità
`InvocationsFailedToBeSentToDlq`	Il numero di invocazioni che non possono essere spostate a una coda DLQ. Errori nelle code DLQ possono verificarsi a causa di errori di autorizzazioni, risorse non disponibili o limiti di dimensione. **Note** EventBridge invia questa metrica solo a se è diversa da zero. CloudWatch	RuleName	Conteggio
`IngestionToInvocationCompleteLatency`	Il tempo impiegato dall'importazione dell'evento al completamento del primo tentativo di invocazione riuscito.	EventBusName, Nessuno, RuleName	Millisecondi
`IngestionToInvocationStartLatency`	Il tempo di elaborazione degli eventi, misurato dal momento in cui un evento viene inserito fino EventBridge alla prima invocazione di un bersaglio.	EventBusName, Nessuno, RuleName	Millisecondi
`InvocationsSentToDlq`	Il numero di invocazioni che vengono spostate in una coda DLQ. **Note** EventBridge invia questa metrica a solo CloudWatch se diversa da zero.	RuleName	Conteggio
`MatchedEvents`	Se EventSourceName è specificato EventBusName o, il numero di eventi che corrispondono a qualsiasi regola. Se RuleName specificato, il numero di eventi corrispondenti a una regola specifica.	EventBusName, EventSourceName, RuleName	Conteggio

Parametro	Descrizione	Dimensioni	Unità
`RetryInvocationAttempts`	Il numero di volte in cui è stata ripetuta l'invocazione della destinazione. **Note** EventBridge invia questa metrica solo a CloudWatch se è diversa da zero.	Nessuno	Conteggio
`SuccessfulInvocationAttempts`	Il numero di volte in cui la destinazione è stata richiamata senza errori.	Nessuno	Conteggio
`ThrottledRules`	Il numero di volte in cui l'esecuzione delle regole è stata limitata. Le invocazioni per tali regole potrebbero essere ritardate. Per ulteriori informazioni, consulta Limite di invocazioni in transazioni al secondo in ???.	EventBusName, Nessuno, RuleName	Conteggio
`TriggeredRules`	Il numero di regole che sono state eseguite e che corrispondono a qualsiasi evento. Questa metrica non verrà visualizzata CloudWatch finché non verrà attivata una regola.	EventBusName, Nessuno, RuleName	Conteggio

EventBridge PutEvents metriche

Lo spazio dei nomi `AWS/Events` include le seguenti metriche relative alle richieste API `PutEvents`.

Per le metriche che utilizzano Count come unità, Sum e SampleCount tendono ad essere le statistiche più utili.

Parametro	Descrizione	Dimensioni	Unità
`PutEventsApproximateCallCount`	Il numero approssimativo di richieste `PutEvents` ricevute.	Nessuno	Conteggio
`PutEventsApproximateFailedCount`	Il numero approssimativo di richieste `PutEvents` non riuscite.	Nessuno	Conteggio
`PutEventsApproximateSuccessCount`	Il numero di richieste `PutEvents` riuscite.	Nessuno	Conteggio
`PutEventsApproximateThrottledCount`	Il numero di richieste `PutEvents` rifiutate a causa della limitazione.	Nessuno	Conteggio
`PutEventsEntriesCount`	Il numero di voci di eventi contenute in una richiesta `PutEvents`.	Nessuno	Conteggio
`PutEventsFailedEntriesCount`	Il numero di voci di eventi contenute in una richiesta `PutEvents` che non è stata importata.	Nessuno	Conteggio
`PutEventsLatency`	Il tempo impiegato per la richiesta `PutEvents`.	Nessuno	Millisecondi
`PutEventsRequestSize`	La dimensione della richiesta `PutEvents`.	Nessuno	Byte

EventBridge PutPartnerEvents metriche

Lo spazio dei nomi `AWS/Events` include le seguenti metriche relative alle richieste API `PutPartnerEvents`.

> **Note**
>
> EventBridge include solo le metriche relative alle PutPartnerEventsrichieste negli account partner SaaS che inviano eventi. Per ulteriori informazioni, consulta ???

Per le metriche che utilizzano Count come unità, Sum e 2 SampleCount tendono ad essere le statistiche più utili.

Parametro	Descrizione	Dimensioni	Unità
`PutPartnerEventsApproximateCallCount`	Il numero approssimativo di richieste `PutPartnerEvents` ricevute.	Nessuno	Conteggio
`PutPartnerEventsApproximateFailedCount`	Il numero approssimativo di richieste `PutPartnerEvents` non riuscite.	Nessuno	Conteggio
`PutPartnerEventsApproximateThrottledCount`	Il numero di richieste `PutPartnerEvents` rifiutate a causa della limitazione.	Nessuno	Conteggio
`PutPartnerEventsApproximate`	Il numero di richieste `PutPartnerEvents` riuscite.	Nessuno	Conteggio

Parametro	Descrizione	Dimensioni	Unità
SuccessCount			
PutPartnerEventsEntriesCount	Il numero di voci di eventi contenute in una richiesta PutPartnerEvents .	Nessuno	Conteggio
PutPartnerEventsFailedEntriesCount	Il numero di voci di eventi contenute in una richiesta PutPartnerEvents che non è stata importata.	Nessuno	Conteggio
PutPartnerEventsLatency	Il tempo impiegato per la richiesta PutPartnerEvents .	Nessuno	Millisecondi

Dimensioni per le metriche EventBridge

EventBridge le metriche hanno dimensioni, o attributi ordinabili, che sono elencati di seguito.

Dimensione	Descrizione
EventBusName	Filtra le metriche disponibili per nome di router di eventi.
EventSourceName	Filtra le metriche disponibili per nome di origine di eventi partner.
RuleName	Filtra i parametri disponibili per nome regola.

Risoluzione dei problemi con Amazon EventBridge

Puoi utilizzare la procedura descritta in questa sezione per risolvere i problemi di Amazon. EventBridge

Argomenti

La mia regola è stata eseguita ma la funzione Lambda non è stata richiamata

Uno dei motivi per cui la funzione Lambda potrebbe non funzionare è che forse non disponi delle autorizzazioni appropriate.

Per verificare le autorizzazioni per la funzione Lambda

1. Utilizzando AWS CLI, esegui il comando seguente con la tua funzione e la tua AWS regione:

```
aws lambda get-policy --function-name MyFunction --region us-east-1
```

Vedrai il seguente output.

```
{
    "Policy": "{\"Version\":\"2012-10-17\",
    \"Statement\":[
        {\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:events:us-
east-1:123456789012:rule/MyRule\"}},
        \"Action\":\"lambda:InvokeFunction\",
        \"Resource\":\"arn:aws:lambda:us-east-1:123456789012:function:MyFunction\",
        \"Effect\":\"Allow\",
        \"Principal\":{\"Service\":\"events.amazonaws.com\"},
        \"Sid\":\"MyId\"}
    ],
    \"Id\":\"default\"}"
}
```

2. Se viene visualizzato il messaggio seguente.

```
A client error (ResourceNotFoundException) occurred when calling the GetPolicy
 operation: The resource you requested does not exist.
```

Oppure, se viene visualizzato l'output ma non riesci a individuare events.amazonaws.com come entità attendibile nella policy, esegui il comando seguente:

```
aws lambda add-permission \
--function-name MyFunction \
--statement-id MyId \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
```

3. Se l'output contiene un campo `SourceAccount`, devi rimuoverlo. Un'`SourceAccount`impostazione EventBridge impedisce di poter richiamare la funzione.

> **Note**
>
> Se il criterio non è corretto, puoi modificare la regola nella EventBridge console rimuovendola e quindi aggiungendola nuovamente alla regola. La EventBridge console imposta quindi le autorizzazioni corrette sulla destinazione.
> Se utilizzi un alias o una versione specifico di Lambda, aggiungi il parametro `--qualifier` nei comandi `aws lambda get-policy` e `aws lambda add-permission`, come mostrato nel comando seguente:
>
> ```
> aws lambda add-permission \
> --function-name MyFunction \
> --statement-id MyId \
> --action 'lambda:InvokeFunction' \
> --principal events.amazonaws.com \
> --source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
> --qualifier alias or version
> ```

Ho appena creato o modificato una regola ma non corrisponde a un evento di test

Quando apporti una modifica a una regola o alle relative destinazioni, gli eventi in entrata potrebbero non avviare o interrompere immediatamente la corrispondenza con regole nuove o aggiornate. È necessario un breve periodo di tempo affinché vengano applicate le modifiche.

Se gli eventi continuano a non corrispondere dopo un breve periodo di tempo, controlla le CloudWatch metriche `TriggeredRules` e verifica `FailedInvocations` la tua regola. `Invocations` Per ulteriori informazioni su questi parametri, consulta Monitoring Amazon EventBridge.

Se la regola è destinata a corrispondere a un evento di un AWS servizio, esegui una delle seguenti operazioni:

- Usa l'azione `TestEventPattern` per verificare che il modello di eventi della tua regola corrisponda a un evento di test. Per ulteriori informazioni, TestEventPatternconsulta Amazon EventBridge API Reference.
- Usa la Sandbox sulla EventBridge console.

La mia regola non è stata eseguita quando ho specificato `ScheduleExpression`

Assicurati di aver impostato la pianificazione per la regola nel fuso orario UTC+0. Se `ScheduleExpression` è corretta, segui la procedura descritta in Ho appena creato o modificato una regola ma non corrisponde a un evento di test.

La mia regola non è stata eseguita all'orario previsto

EventBridge esegue le regole entro un minuto dall'ora di inizio impostata. Il conteggio dell'orario di esecuzione viene avviato al momento della creazione della regola.

> **Note**
>
> Il tipo di distribuzione delle regole pianificate è `guaranteed` il che significa che gli eventi verranno attivati almeno una volta per ogni orario previsto.

Puoi utilizzare un'espressione Cron per richiamare le destinazioni a un orario specificato. Per creare una regola che viene eseguita ogni quattro ore al minuto 0, esegui una delle seguenti operazioni:

- Nella EventBridge console, si utilizza l'espressione `0 0/4 * * ? *` cron.
- Usando AWS CLI, si usa l'espressione`cron(0 0/4 * * ? *)`.

Ad esempio, per creare una regola denominata `TestRule` che viene eseguita ogni 4 ore utilizzando il AWS CLI, si utilizza il comando seguente.

```
aws events put-rule --name TestRule --schedule-expression 'cron(0 0/4 * * ? *)'
```

Per eseguire una regola ogni cinque minuti, utilizzi la seguente espressione Cron.

```
aws events put-rule --name TestRule --schedule-expression 'cron(0/5 * * * ? *)'
```

La risoluzione massima per una EventBridge regola che utilizza un'espressione cron è un minuto. La regola pianificata viene attivata entro tale minuto, ma non al secondo 0 preciso.

Poiché EventBridge tutti i servizi di destinazione sono distribuiti, può verificarsi un ritardo di diversi secondi tra l'esecuzione della regola pianificata e il momento in cui il servizio di destinazione esegue l'azione sulla risorsa di destinazione.

La mia regola corrisponde alle chiamate API di servizio AWS globali, ma non è stata eseguita

AWS servizi globali; come IAM e Amazon Route 53 sono disponibili solo nella regione Stati Uniti orientali (Virginia settentrionale), quindi gli eventi delle chiamate AWS API dei servizi globali sono disponibili solo in quella regione. Per ulteriori informazioni, consulta Eventi derivanti dai servizi AWS.

Il ruolo IAM associato alla mia regola viene ignorato durante l'esecuzione della regola

EventBridge utilizza i ruoli IAM solo per le regole che inviano eventi ai flussi Kinesis. Per le regole che richiamano funzioni Lambda o argomenti Amazon SNS, devi fornire autorizzazioni basate su risorse.

Assicurati che gli AWS STS endpoint regionali siano abilitati, in modo che EventBridge possano utilizzarli quando assumeranno il ruolo IAM che hai fornito. Per ulteriori informazioni, consulta Attivazione e disattivazione AWS STS in una AWS regione nella Guida per l'utente IAM.

La mia regola ha un modello di eventi che dovrebbe corrispondere a una risorsa, ma nessun evento corrisponde

La maggior parte dei servizi utilizza i due punti (:) o la barra (/) come lo stesso carattere in Amazon Resource Names (ARNs), ma EventBridge utilizza una corrispondenza esatta nei modelli e nelle regole degli eventi. AWS Assicurati di utilizzare i caratteri ARN corretti durante la creazione di modelli di eventi, facendo in modo che corrispondano alla sintassi ARN nell'evento per il quale cercare corrispondenze.

Alcuni eventi, come gli eventi di chiamata AWS API from CloudTrail, non hanno nulla nel campo delle risorse.

Si è verificato un ritardo nella distribuzione del mio evento alla destinazione

EventBridge tenta di inviare un evento a un obiettivo per un massimo di 24 ore, tranne negli scenari in cui la risorsa di destinazione è limitata. Il primo tentativo viene effettuato appena l'evento giunge nel flusso di eventi. Se il servizio di destinazione presenta problemi, riprogramma EventBridge automaticamente un'altra consegna. Se sono trascorse 24 ore dall'arrivo dell'evento, EventBridge interrompe il tentativo di consegna dell'evento e pubblica la metrica in. `FailedInvocations` CloudWatch Ti consigliamo di configurare una coda DLQ per archiviare gli eventi che non sono stati distribuiti correttamente a una destinazione. Per ulteriori informazioni, consulta Utilizzo di code di lettere morte per elaborare gli eventi non consegnati

Alcuni eventi non sono mai stati distribuiti nel target

Se l'obiettivo di una EventBridge regola è limitato per un periodo di tempo prolungato, EventBridge potrebbe non essere possibile ritentare la consegna. Ad esempio, se la destinazione non è predisposta per gestire il traffico degli eventi in entrata e il servizio di destinazione limita le richieste effettuate per tuo conto, potresti non EventBridge ritentare la consegna. EventBridge

La mia regola è stata eseguita più di una volta in risposta a un evento

In rari casi, la stessa regola può essere eseguita più di una volta per un unico evento o un orario pianificato, oppure la stessa destinazione può essere richiamata più di una volta per una determinata regola attivata.

Come evitare loop infiniti

In EventBridge, è possibile creare una regola che porti a cicli infiniti, in cui la regola viene eseguita ripetutamente. Se hai una regola che causa un loop infinito, riscrivila in modo che le azioni intraprese dalla regola non corrispondano alla stessa regola.

Ad esempio, una regola che rileva che gli ACL sono stati modificati in un bucket Amazon S3 e quindi esegue il software per modificarne lo stato causa un loop infinito. Un modo di risolvere questo inconveniente consiste nel riscrivere la regola in modo che corrisponda solo alle liste di controllo degli accessi (ACL) il cui stato non è corretto.

Un loop infinito può generare rapidamente costi più alti di quelli previsti. Ti consigliamo di utilizzare il budgeting, che avvisa quando i costi superano il limite indicato. Per ulteriori informazioni, consulta Gestione dei costi con i budget.

I miei eventi non vengono distribuiti alla coda di Amazon SQS target

Se la tua coda Amazon SQS è crittografata, devi creare una chiave KMS gestita dal cliente e includere la seguente sezione di autorizzazione nella policy della chiave KMS. Per ulteriori informazioni, vedere Configurazione delle autorizzazioni AWS KMS.

```
{
  "Sid": "Allow EventBridge to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

La regola viene eseguita ma non vedo messaggi pubblicati nell'argomento Amazon SNS

Scenario 1

Devi disporre dell'autorizzazione per pubblicare messaggi nel tuo argomento Amazon SNS. Usa il seguente comando usando AWS CLI, sostituendo us-east-1 con la tua regione e usando il tuo argomento ARN.

```
aws sns get-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic"
```

Per disporre dell'autorizzazione corretta, gli attributi della policy devono essere simili a quanto segue.

```
"{\"Version\":\"2012-10-17\",
\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",
\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"*\"},
\"Action\":[\"SNS:Subscribe\",
\"SNS:ListSubscriptionsByTopic\",
\"SNS:DeleteTopic\",
\"SNS:GetTopicAttributes\",
\"SNS:Publish\",
\"SNS:RemovePermission\",
\"SNS:AddPermission\",
\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\",
\"Condition\":{\"StringEquals\":{\"AWS:SourceOwner\":\"123456789012\"}}},{\"Sid\":
\"Allow_Publish_Events\",
\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}]}"
```

Se non vedi `events.amazonaws.com` con l'autorizzazione `Publish` nella tua policy, copia prima la policy corrente e aggiungi la seguente istruzione all'elenco delle istruzioni.

```
{\"Sid\":\"Allow_Publish_Events\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}
```

Quindi imposta gli attributi dell'argomento utilizzando AWS CLI, usa il seguente comando.

```
aws sns set-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic" --attribute-name Policy --attribute-
value NEW_POLICY_STRING
```

> **Note**
>
> Se il criterio non è corretto, puoi anche modificare la regola nella EventBridge console rimuovendola e quindi aggiungendola nuovamente alla regola. EventBridge imposta le autorizzazioni corrette sulla destinazione.

Scenario 2

Se l'argomento SNS è crittografato, devi includere la sezione seguente nella policy della chiave KMS.

```
{
  "Sid": "Allow EventBridge to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

Il mio argomento Amazon SNS dispone ancora delle autorizzazioni EventBridge anche dopo aver eliminato la regola associata all'argomento Amazon SNS

Quando crei una regola con Amazon SNS come destinazione, EventBridge aggiunge l'autorizzazione all'argomento Amazon SNS per tuo conto. Se elimini la regola poco dopo averla creata, EventBridge potresti non rimuovere l'autorizzazione dal tuo argomento Amazon SNS. In questo caso, puoi rimuovere l'autorizzazione dall'argomento utilizzando il comando `aws sns set-topic-attributes`. Per ulteriori informazioni sulle autorizzazioni basate su risorse per l'invio di eventi, consulta Utilizzo di policy basate su risorse per Amazon EventBridge.

Con EventBridge quali chiavi di condizione IAM posso usare?

EventBridge supporta le chiavi di condizione AWS-wide (vedi IAM e AWS STS condition context keys nella IAM User Guide), oltre alle chiavi elencate inUtilizzo di condizioni di policy IAM per il controllo granulare degli accessi.

Come posso sapere quando EventBridge le regole vengono violate?

Puoi utilizzare il seguente avviso per avvisarti quando EventBridge le tue regole vengono violate.

Creazione di un allarme di avviso dell'interruzione delle regole

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Scegli Crea allarme. Nel riquadro CloudWatch Metriche per categoria, scegli Metriche degli eventi.
3. Nell'elenco delle metriche, seleziona. FailedInvocations
4. Sopra il grafico, seleziona Statistic (Statistica), Sum (Somma).
5. In Period (Periodo), seleziona un valore, ad esempio 5 minutes (5 minuti). Seleziona Successivo.
6. In Soglia di allarme, per Nome, digita un nome univoco per l'allarme, ad esempio myFailedRules. In Descrizione, digita una descrizione dell'allarme, ad esempio Le regole non distribuiscono eventi a destinazioni.
7. In is (è), seleziona >= e 1. In for (per), immetti 10.
8. In Actions (Operazioni), in Whenever this alarm (Ogni volta che questo allarme), scegli State is ALARM (Lo stato è ALLARME).
9. In Invia notifica a, seleziona un argomento Amazon SNS esistente o creane uno. Per creare un nuovo argomento, seleziona New list (Nuovo elenco). Digita un nome per il nuovo argomento Amazon SNS, ad esempio:. myFailedRules
10. In Email list (Elenco e-mail), digita un elenco separato da virgola di indirizzi e-mail a cui inviare una notifica quando lo stato dell'allarme passa a ALARM.
11. Scegli Crea allarme.

Quote di Amazon EventBridge

Esistono quote per la maggior parte degli aspetti di EventBridge.

Argomenti

> **Note**
>
> Per un elenco delle quote per Painificatore EventBridge, consulta Quote per Pianificatore EventBridge nella Guida per l'utente di Pianificatore EventBridge.

Quote di EventBridge

EventBridge include le quote descritte di seguito.

La console Service Quotas fornisce informazioni sulle quote di EventBridge. Oltre a visualizzare le quote predefinite, è possibile utilizzare la console Service Quotas per richiedere aumenti di quota per le quote modificabili.

Nome	Predefinita	Adatta e	Descrizione
Destinazioni API	Ogni regione supportata: 3.000	Sì	Il numero massimo di destinazioni API per account per Regione.
Connessioni	Ogni regione supportata: 3.000	Sì	Il numero massimo di connessioni per account per Regione.

Nome	Predefinita	Adatta e	Descrizione
Limite di frequenza di CreateEndpoint nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di richieste al secondo per l'API CreateEnd point. Ulteriori richieste verranno sottopost e a limitazione (della larghezza di banda della rete).
Limite di frequenza di DeleteEndpoint nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di richieste al secondo per API DeleteEnd point. Ulteriori richieste verranno sottopost e a limitazione (della larghezza di banda della rete).
Endpoints	Ogni regione supportata: 100	Sì	Il numero massimo di endpoint per account per Regione.
Dimensione della policy per router di eventi	Ogni Regione supportata: 10.240	Sì	Dimensione massima della policy, espressa in caratteri. Le dimensioni della policy aumentano ogni volta che vengono concesse le credenzia li d'accesso a un altro account. Puoi verificar e la tua policy corrente e le relative dimension i utilizzando l'API DescribeEventBus.

Nome	Predefinita	Adatta e	Descrizione
Bus di eventi	Ogni regione supportata: 100	Sì	Numero massimo di router di eventi per account.
Dimensione del modello di eventi	Ogni Regione supportata: 2.048	Sì	Dimensione massima di un modello di eventi, espressa in caratteri.

Nome	Predefinita	Adatta e	Descrizione
Limite di invocazioni in transazioni al secondo	us-east-1: 18.750 al secondo us-east-2: 4.500 al secondo us-west-1: 2.250 al secondo us-east-2: 18.750 al secondo af-south-1: 750 per secondo ap-northeast-1: 2.250 al secondo ap-northeast-3: 750 al secondo ap-southeast-1: 2.250 al secondo ap-southeast-2: 2.250 al secondo ap-southeast-3: 750 al secondo eu-central-1: 4.500 al secondo eu-south-1: 750 al secondo	Sì	Un'invocazione è un evento che corrisponde a una regola e che viene inviata alle destinazioni delle regole. Una volta raggiunto il limite, le invocazioni vengono limitate, ovvero vengono comunque eseguite ma in ritardo.

Nome	Predefinita	Adatta e	Descrizione
	eu-west-1: 18.750 al secondo eu-west-2: 2.250 al secondo Ogni altra regione supportata: 1.100 al secondo		
Numero di regole	af-south-1: 100 eu-south-1: 100 Ogni altra Regione supportata: 300	Sì	Numero massimo di regole che un account può avere per bus evento

Nome	Predefinita	Adatta e	Descrizione
Limite di limitazione di PutEvents nelle transazioni al secondo	us-east-1: 10.000 al secondo us-east-2: 2.400 al secondo us-west-1: 1.200 al secondo us-east-2: 10.000 al secondo af-south-1: 750 per secondo ap-northeast-1: 1.200 al secondo ap-northeast-3: 400 al secondo ap-southeast-1: 1.200 al secondo ap-southeast-2: 1.200 al secondo ap-southeast-3: 400 al secondo eu-central-1: 2.400 al secondo eu-south-1: 400 al secondo	Sì	Il numero massimo di richieste al secondo per l'API PutEvents. Ulteriori richieste verranno sottoposte a limitazione (della larghezza di banda della rete).

Nome	Predefinita	Adatta e	Descrizione
	eu-west-1: 10.000 al secondo eu-west-2: 1.200 al secondo Ogni altra regione supportata: 600 al secondo		
Frequenza di richiami per destinazione API	Ogni regione supportata: 300 al secondo	Sì	Il numero massimo di invocazioni al secondo da inviare a ciascun endpoint di destinazione API per account per Regione. Una volta raggiunta la quota, le invocazioni future a quell'endpoint API saranno limitate. Le invocazioni verranno comunque eseguite, ma saranno in ritardo.
Obiettivi per regola	Ogni Regione supportata: 5	No	Il numero massimo di destinazioni che possono essere associate a una regola
Limite di limitazione nelle transazioni al secondo	Ogni regione supportata: 50 al secondo	Sì	Numero massimo di richieste al secondo per tutte le operazioni API EventBridge tranne PutEvents. Le richieste aggiuntive sono limitate

Nome	Predefinita	Adatta e	Descrizione
Limite di frequenza di UpdateEndpoint nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di richieste al secondo per API UpdateEnd point. Ulteriori richieste verranno sottopost e a limitazione (della larghezza di banda della rete).

Inoltre, EventBridge include le seguenti quote che non sono gestite tramite la console Service Quotas.

Nome	Predefinito	Descrizione
Bus di eventi	Ogni regione supportata: 100	Numero massimo di router di eventi per account.
Dimensione della policy per router di eventi	Ogni Regione supportat a: 10.240	Dimensione massima della policy, espressa in caratteri. Le dimensioni della policy aumentano ogni volta che vengono concesse le credenziali d'accesso a un altro account. Puoi controllare la policy corrente e le sue dimensioni utilizzando le API `DescribeEventBus` .
Dimensione del modello di eventi	Ogni Regione supportat a: 2048	Dimensione massima di un modello di eventi, espressa in caratteri. È regolabile fino a 4.096 caratteri. Se hai dei requisiti per il limite massimo più alto, contatta l'assistenza.
Regole contenenti caratteri jolly	Ogni Regione supportat	Numero massimo di regole, per router di eventi per account, che possono contenere filtri di

Nome	Predefinito	Descrizione
	a: 30 regole per router di eventi	eventi che includono caratteri jolly. Questa quota non può essere modificata. Per ulteriori informazioni sull'uso di caratteri jolly in modelli di eventi, consulta ???.
Livelli di rilevamento di schemi	Ogni Regione supportata: 255 livelli	Il numero massimo di livelli di rilevamento di schemi in cui verranno acquisiti eventi nidificati. Tutti gli eventi oltre i 255 livelli vengono ignorati.

Quote PutPartnerEvents per Regione

Se hai dei requisiti per limiti massimi più alti, contatta l'assistenza.

Regioni	Transazioni al secondo
• AWS GovCloud (US-West) • AWS GovCloud (US-East) • Stati Uniti orientali (Virginia settentrionale) • Stati Uniti orientali (Ohio) • Stati Uniti occidentali (California settentrionale) • Stati Uniti occidentali (Oregon) • Africa (Città del Capo) • Asia Pacific (Hong Kong) • Asia Pacific (Mumbai) • Asia Pacific (Osaka) • Asia Pacific (Seul) • Asia Pacifico (Singapore) • Asia Pacifico (Sydney)	Per impostazione predefinita PutPartnerEvents ha un limite flessibile di 1.400 richieste di capacità al secondo e 3.600 richieste burst al secondo in tutte le Regioni.

Regioni	Transazioni al secondo
• Asia Pacifico (Tokyo) • Canada (Centrale) • Europa (Francoforte) • Europa (Irlanda) • Europa (Londra) • Europa (Milano) • Europa (Parigi) • Europe (Stockholm) • Europa (Milano) • Sud America (San Paolo) • Cina (Ningxia) • Cina (Pechino)	

Quote del registro di schemi EventBridge

Il registro di schemi EventBridge include le quote seguenti.

La console Service Quotas fornisce informazioni sulle quote di EventBridge. Oltre a visualizzare le quote predefinite, è possibile utilizzare la console Service Quotas per richiedere aumenti di quota per le quote modificabili.

Nome	Predefinita	Adattabile	Descrizione
DiscoveredSchemas	Ogni Regione supportata: 200	Sì	Il numero massimo di schemi per un registro di schemi rilevato che è possibile creare nella Regione corrente

Nome	Predefinita	Adatta e	Descrizione
Discoverers	Ogni regione supportata: 10	Sì	Il numero massimo di rilevatori che puoi creare nella Regione corrente.
Registri	Ogni regione supportata: 10	Sì	Il numero massimo di registri che puoi creare nella Regione corrente.
SchemaVersions	Ogni regione supportata: 100	Sì	Il numero massimo di versioni per schema che puoi creare nella Regione corrente.
Schemi	Ogni regione supportata: 100	Sì	Il numero massimo di schemi per registro che puoi creare nella Regione corrente. (Ad eccezione del registro dello schema rilevato)

Quote di EventBridge Pipes

EventBridge Pipes include le quote descritte di segutio. Se hai dei requisiti per limiti massimi più alti, contatta l'assistenza.

Risorsa	Regioni	Limite predefinito
Esecuzioni di pipe simultanee per account	• AWS GovCloud (US-West) • AWS GovCloud (US-East) • Cina (Ningxia) • Cina (Pechino) • Asia Pacifico (Osaka-Lo cale)	1000

Risorsa	Regioni	Limite predefinito
	• Africa (Città del Capo) • Europa (Milano) • Stati Uniti orientali (Ohio) • Europa (Francoforte) • Stati Uniti occidentali (California settentrionale) • Europa (Londra) • Asia Pacifico (Sydney) • Asia Pacifico (Tokyo) • Asia Pacifico (Singapore) • Canada (Centrale) • Europa (Parigi) • Europa (Stoccolma) • Sud America (San Paolo) • Asia Pacifico (Seul) • Asia Pacifico (Mumbai) • Asia Pacifico (Hong Kong) • Medio Oriente (Bahrein) • Cina (Ningxia) • Cina (Pechino) • Asia Pacifico (Osaka-Lo cale) • Africa (Città del Capo) • Europa (Milano)	
Esecuzioni di pipe simultanee per account	• Stati Uniti orientali (Virginia settentrionale) • Stati Uniti occidentali (Oregon) • Europa (Irlanda)	3000

Risorsa	Regioni	Limite predefinito
Pipe per account	Tutti	1000

EventBridge Etichette Amazon

Un tag è un'etichetta di attributo personalizzata che tu o AWS assegnate a una AWS risorsa. In EventBridge, puoi assegnare tag ai bus di regole ed eventi. Ogni risorsa può avere un massimo di 50 tag.

Utilizzi i tag per identificare e organizzare AWS le tue risorse. Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, puoi assegnare lo stesso tag a una EventBridge regola che assegni a un'istanza EC2.

Un tag è costituito da due parti:

- Una chiave di tag, ad esempio, `CostCenter`, `Environment` o `Project`.
  - Le chiavi dei tag prevedono una distinzione tra lettere maiuscole e minuscole.
  - La lunghezza massima delle chiavi di tag è 128 caratteri Unicode in UTF-8.
  - Per ogni risorsa, la chiave di ciascun tag deve essere univoca.
  - I caratteri consentiti sono lettere, numeri, spazi rappresentabili in formato UTF-8, oltre ai seguenti caratteri: . : + = @ _ / - (trattino).
  - Il `aws:` prefisso è vietato per i tag perché è riservato all'uso. AWS Non è possibile modificare né eliminare le chiavi o i valori di tag con tale prefisso. I tag con questo prefisso non vengono conteggiati per il limite del numero di tag per risorsa.
- Un campo valore di tag facoltativo, ad esempio `111122223333` o `Production`.
  - La chiave di ogni tag può avere solo un valore.
  - I valori di tag fanno distinzione tra maiuscole e minuscole.
  - Non specificare il valore del tag equivale a utilizzare una stringa vuota.
  - Il valore massimo dei tag è 256 caratteri Unicode in UTF-8.
  - I caratteri consentiti sono lettere, numeri, spazi rappresentabili in formato UTF-8, oltre ai seguenti caratteri: . : + = @ _ / - (trattino).

> **Tip**
>
> Come best practice, è consigliabile definire una strategia per l'uso delle lettere maiuscole e minuscole nei tag e implementarla costantemente in tutti i tipi di risorse. Ad esempio, puoi

decidere se utilizzare `Costcenter`, `costcenter` o `CostCenter` e quindi utilizzare la stessa convenzione per tutti i tag.

Puoi utilizzare la EventBridge console, l' EventBridge API o AWS CLI per aggiungere, modificare o eliminare tag. Per ulteriori informazioni, consulta gli argomenti seguenti:

- TagResourcee ListTagsForResourcenell'Amazon EventBridge API Reference UntagResource
- tag-resource, untag-resource e nel Reference list-tags-for-resourceAWS CLI
- Utilizzo dell'editor di tag nella Guida per l'utente di Resource Groups

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti in ogni versione della Amazon EventBridge User Guide, a partire da luglio 2019. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data di rilascio
Politiche AWS gestite aggiornate.	AWS GovCloud (US) Regions solo `AmazonEventBridgeFullAccess` e `AmazonEventBridgeSchemasFullAccess` le politiche non includono `iam:CreateServiceLinkedRole`, in quanto non viene utilizzato. • the section called “Aggiornamenti alle policy”	9 maggio 2024
Genera AWS CloudFormation modelli da bus e regole di eventi.	Ora puoi generare AWS CloudFormation modelli a partire dai bus e dalle regole di EventBridge eventi Amazon esistenti. • Generazione di un modello di AWS CloudFormation da un router di eventi di Amazon EventBridge	18 novembre 2022
Ha lanciato la documentazione di EventBridge Pipes.	Ora puoi creare pipe per collegare origini a destinazioni, con filtri e arricchimento facoltativi. • Pipe	1 dicembre 2022
Genera AWS CloudFormation modelli da bus e regole di eventi.	Ora puoi generare AWS CloudFormation modelli a partire dai bus e dalle regole di EventBridge eventi Amazon esistenti. • Generazione di un modello di AWS CloudFormation da un router di eventi di Amazon EventBridge	18 novembre 2022

Modifica	Descrizione	Data di rilascio
È stata aggiunta la AmazonEventBridgePipesFullAccess politica.	Fornisce accesso completo ad Amazon EventBridge Pipes. • EventBridge Politiche gestite specifiche per Pipes	1 dicembre 2022
È stata aggiunta la AmazonEventBridgePipesReadOnlyAccess politica.	Fornisce accesso in sola lettura ad Amazon EventBridge Pipes. • EventBridge Politiche gestite specifiche per Pipes	1 dicembre 2022
È stata aggiunta la policy. AmazonEventBridgePipesOperatorAccess	Fornisce l'accesso in sola lettura e all'operatore (ovvero la possibilità di interrompere e avviare Pipes) ad Amazon EventBridge Pipes. • EventBridge Politiche gestite specifiche per Pipes	1 dicembre 2022
È stata aggiornata la politica. CloudWatchEventsFullAccess	Aggiornata per la corrispondenza ad `AmazonEventBridgeFullAccess` . • AmazonEventBridgeFullAccess politica	1 dicembre 2022
Aggiornata la CloudWatchEventsReadOnlyAccess politica.	Aggiornata per la corrispondenza ad `AmazonEventBridgeReadOnlyAccess` . • AmazonEventBridgeReadOnlyAccess politica	1 dicembre 2022

Modifica	Descrizione	Data di rilascio
Aggiornati i filtri dei contenuti in modelli di eventi.	Ora puoi utilizzare le opzioni di filtro `suffix`, `equals-ignore-case` e `$or` per creare modelli di eventi. • Filtraggio dei contenuti nei modelli di EventBridge eventi di Amazon	14 novembre 2022
Aggiornata la AmazonEventBridgeFullAccess politica.	Sono state aggiunte le autorizzazioni necessarie per utilizzare EventBridge Schema Registry and EventBridge Scheduler. • AmazonEventBridgeFullAccess politica	10 novembre 2022
È stata aggiornata la politica. AmazonEventBridgeReadOnlyAccess	È ora possibile visualizzare le informazioni sul registro degli EventBridge schemi e sull' EventBridge utilità di pianificazione. • AmazonEventBridgeReadOnlyAccess politica	10 novembre 2022
Aggiornati i filtri dei contenuti in modelli di eventi.	Ora puoi utilizzare le opzioni di filtro `suffix`, `equals-ignore-case` e `$or` per creare modelli di eventi. • Filtraggio dei contenuti nei modelli di EventBridge eventi di Amazon	14 novembre 2022
È stata aggiornata la AmazonEventBridgeFullAccess politica.	Sono state aggiunte le autorizzazioni necessarie per utilizzare EventBridge Schema Registry and EventBridge Scheduler. • AmazonEventBridgeFullAccess politica	10 novembre 2022

Modifica	Descrizione	Data di rilascio
È stata aggiornata la politica. AmazonEventBridgeReadOnlyAccess	È ora possibile visualizzare le informazioni sul registro degli EventBridge schemi e sull' EventBridge utilità di pianificazione. • AmazonEventBridgeReadOnlyAccess politica	10 novembre 2022
È stata aggiornata la AmazonEventBridgeReadOnlyAccess politica.	Ora puoi visualizzare le informazioni sugli endpoint. • AmazonEventBridgeReadOnlyAccess politica	7 aprile 2022
Aggiunto contenuto per endpoint globali.	Amazon EventBridge ora supporta l'utilizzo di endpoint globali per rendere la tua applicazione tollerante ai guasti regionali senza costi aggiuntivi. Per ulteriori informazioni, consulta quanto segue: • Rendere le applicazioni tolleranti ai guasti a livello regionale con endpoint globali e replica degli eventi • `CreateEndpoint`	7 aprile 2022
Aggiunto supporto per archivi e riproduzioni di eventi.	Amazon EventBridge ora supporta l'utilizzo di archivi per archiviare eventi e di replay di eventi per riprodurre gli eventi da un archivio. Per ulteriori informazioni, consulta quanto segue: • Archiviazione degli eventi Amazon EventBridge . • `CreateArchive` • `StartReplay`	5 novembre 2020

Modifica	Descrizione	Data di rilascio
Aggiunto supporto per le code DLQ e la policy di ripetizione per destinazioni.	Amazon EventBridge ora supporta l'utilizzo di code di lettere morte e la definizione di una politica di nuovi tentativi per gli obiettivi. Per ulteriori informazioni, consulta quanto segue: • Utilizzo di code di lettere morte per elaborare gli eventi non consegnati. • `PutTargets`	12 ottobre 2020
Aggiunto supporto per gli schemi in formato JsonSchema Draft4.	Amazon EventBridge ora supporta schemi in formato JsonSchema Draft 4. Ora puoi anche esportare schemi utilizzando l'API. EventBridge Per ulteriori informazioni, consulta quanto segue. • EventBridge Schemi Amazon • `Export`nello EventBridge Schema Registry API Reference.	28 settembre 2020
Politiche basate sulle risorse per lo Schema Registry EventBridge	Amazon EventBridge Schema Registry ora supporta politiche basate sulle risorse. Per ulteriori informazioni, consulta gli argomenti seguenti. • Policy basate su risorse per Amazon EventBridge Schemas • `Policy`nello EventBridge Schema Registry API Reference • RegistryPolicy Tipo di risorsa nella Guida AWS CloudFormation per l'utente	30 aprile 2020

Modifica	Descrizione	Data di rilascio
Tag per bus di eventi	Questa versione consente di creare e gestire tag per bus di eventi. Puoi aggiungere tag durante la creazione di un bus di eventi e aggiungere o gestire tag esistenti chiamando l'API correlata. Per ulteriori informazioni, consulta gli argomenti seguenti. • EventBridge Etichette Amazon • Policy basate su tag • `TagResource` • `UntagResource` • `ListTagsForResource`	24 febbraio 2020
Aumento delle quote di servizio	Amazon EventBridge ha aumentato le quote per le chiamate e per. `PutEvents` Le quote variano a seconda della Regione e possono essere aumentate se necessario.	11 febbraio 2020
Aggiunto un nuovo argomento sulla trasformazione dell'input di destinazione e aggiunto un collegamento agli eventi di Application Auto Scaling.	Documentazione migliorata sul trasformatore di input. • Trasformazione degli EventBridge input di Amazon • Utilizza il trasformatore di input per estrarre i dati da un evento e inserirli nella destinazione • Tutorial: utilizzo del trasformatore di input per personalizzare gli elementi che EventBridge passa alla destinazione di un evento Aggiunto un collegamento agli eventi di Application Auto Scaling. • Eventi di Application Auto Scaling e EventBridge • Eventi derivanti dai servizi AWS	20 dicembre 2019

Modifica	Descrizione	Data di rilascio
Filtraggio basato sul contenuto		19 dicembre 2019
Sono stati aggiunti collegamenti agli esempi di eventi di Amazon Augmented AI.	È stato aggiunto un collegamento all'argomento Amazon Augmented AI nella SageMaker Amazon Developer Guide che fornisce eventi di esempio per Amazon Augmented AI. Per ulteriori informazioni, consulta gli argomenti seguenti. • Utilizzo degli eventi di Amazon Augmented AI • Eventi derivanti dai servizi AWS	13 dicembre 2019
Aggiunti collegamenti agli esempi di eventi di Amazon Chime.	Aggiunto un collegamento all'argomento Amazon Chime che fornisce eventi di esempio per quel servizio. Per ulteriori informazioni, consulta gli argomenti seguenti. • Automazione di Amazon Chime con EventBridge • Eventi derivanti dai servizi AWS	12 dicembre 2019
EventBridge Schemi Amazon	Ora puoi gestire schemi e generare associazioni di codice per eventi in Amazon. EventBridge Per ulteriori informazioni, consulta gli argomenti seguenti. • EventBridge Schemi Amazon • EventBridge Riferimento all'API Schemas • EventSchemas Riferimento al tipo di risorsa in AWS CloudFormation	1 dicembre 2019
AWS CloudFormation supporto per Event Buses	AWS CloudFormation ora supporta la EventBus risorsa. Supporta anche il EventBusName parametro sia nelle risorse che nelle EventBusPolicy risorse Rule. Per ulteriori informazioni, consulta Amazon EventBridge Resource Type Reference.	7 ottobre 2019

Modifica	Descrizione	Data di rilascio
Nuovo servizio	Versione iniziale di Amazon EventBridge.	11 luglio 2019

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.