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 gli eventi per connettere tra loro i componenti dell'applicazione, semplificando la creazione di applicazioni scalabili basate sugli 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.

Viene utilizzato EventBridge per indirizzare gli eventi da fonti quali applicazioni, AWS servizi e software di terze parti sviluppati internamente alle applicazioni destinate ai consumatori all'interno dell'organizzazione. EventBridge offre modi semplici e coerenti per importare, filtrare, trasformare e fornire eventi in modo da poter creare applicazioni rapidamente.

Il video seguente fornisce una breve introduzione alle funzionalità di Amazon EventBridge:

EventBridge include due modi per elaborare gli eventi: bus 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 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.

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 Amazon CloudWatch Events

EventBridge in precedenza si chiamava Amazon CloudWatch Events. Il bus degli eventi predefinito e le regole che hai creato in CloudWatch Events vengono visualizzati anche nella EventBridge console. EventBridge utilizza gli stessi CloudWatch EventsAPI, quindi il codice che utilizza gli CloudWatch Events API rimane lo stesso.

EventBridge si basa sulle funzionalità di CloudWatch Events con funzionalità quali eventi per i partner, Schema Registry e EventBridge Pipes. Le nuove funzionalità aggiunte a non EventBridge vengono aggiunte agli CloudWatch eventi. Per ulteriori informazioni, consulta ???.

Sono presenti anche tutte le funzionalità a cui sei abituato in CloudWatch Events EventBridge, tra cui:

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

EventBridge le funzionalità che si basano e ampliano le funzionalità degli eventi 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

Iscriviti 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 https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

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

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

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. 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 la registrazione 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. Attiva l'autenticazione a più fattori (MFA) per il tuo utente root.

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

Crea un utente con accesso amministrativo

1. Abilita IAM Identity Center.

   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 i valori predefiniti IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

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

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

Assegna l'accesso a ulteriori utenti

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

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

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

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

Accedi alla EventBridge console Amazon

Per accedere alla EventBridge console Amazon

- Accedi a 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 IAM account 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, vedere 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 EventBridge Tutorial Amazon.

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.

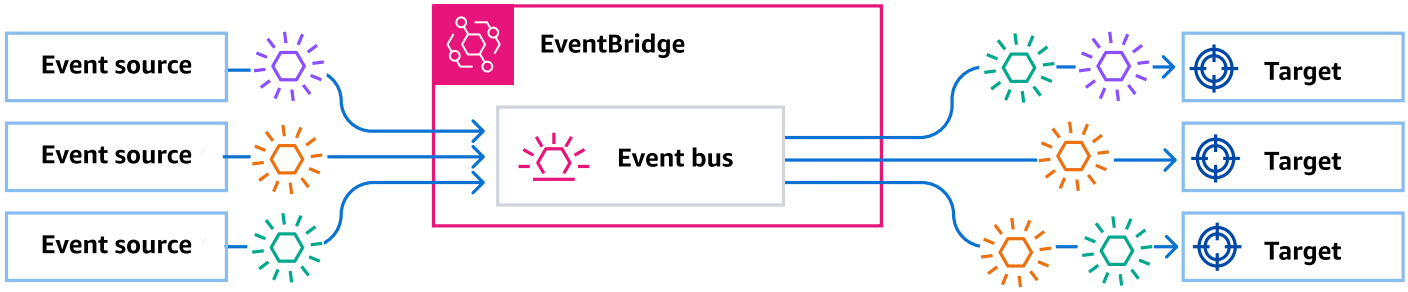
      Per personalizzare il modello, scegli Modello personalizzato (JSONeditor) e apporta le modifiche.
    - Per utilizzare un modello di evento personalizzato, scegli Modello personalizzato (JSONeditor) e crea il tuo modello di evento.
11. Seleziona Successivo.
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. Per Seleziona un obiettivo, scegli il AWS servizio a cui desideri inviare informazioni quando EventBridge rileva un evento che corrisponde al modello 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 IAM ruolo necessario per l'esecuzione della regola. Esegui una di queste operazioni:
    - Per creare un IAM ruolo automaticamente, scegli Crea un nuovo ruolo per questa risorsa specifica.
    - Per utilizzare un IAM ruolo creato in precedenza, scegli Usa ruolo esistente e seleziona il ruolo esistente dall'elenco 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 SQS coda Amazon 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 SQS coda Amazon nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dall'elenco a discesa.

- Scegli Seleziona una SQS coda Amazon in un altro AWS account come coda di lettere non scritte, quindi inserisci la ARN coda da utilizzare. È necessario allegare alla coda una politica basata sulle risorse che autorizza l'invio di 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 Etichettare le risorse in Amazon EventBridge.
20. Seleziona Next (Successivo).
21. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Autobus per eventi su Amazon EventBridge

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 ???.

Come funzionano gli Event Bus EventBridge

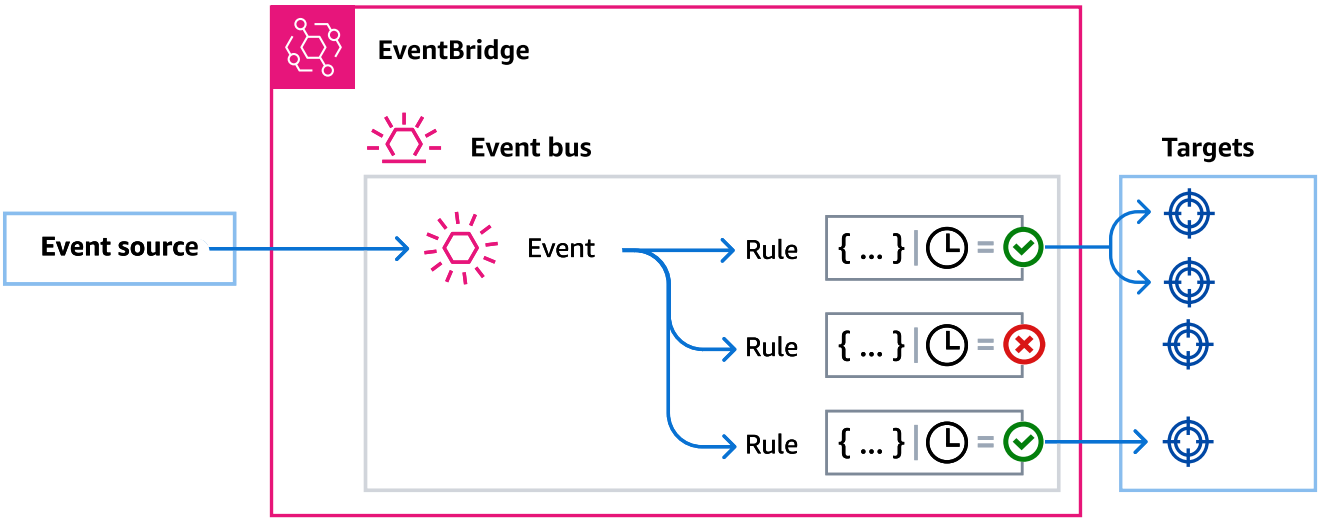
I router di eventi ti consentono di instradare gli eventi da più origini a molteplici destinazioni.

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 agli eventi in cui un'EC2istanza Amazon l'ha `state` modificata`running`.

     A tale scopo, specifica 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 un'EC2istanza Amazon cambia stato, Amazon EC2 (l'origine dell'evento) invia automaticamente l'evento al bus eventi predefinito.
3. EventBridge valuta tutti gli eventi inviati al bus di eventi predefinito rispetto alla regola che hai creato.

   Se l'evento corrisponde alla tua regola (ovvero se l'evento era un'EC2istanza Amazon 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 Event Bus in Amazon EventBridge

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

Bus di eventi

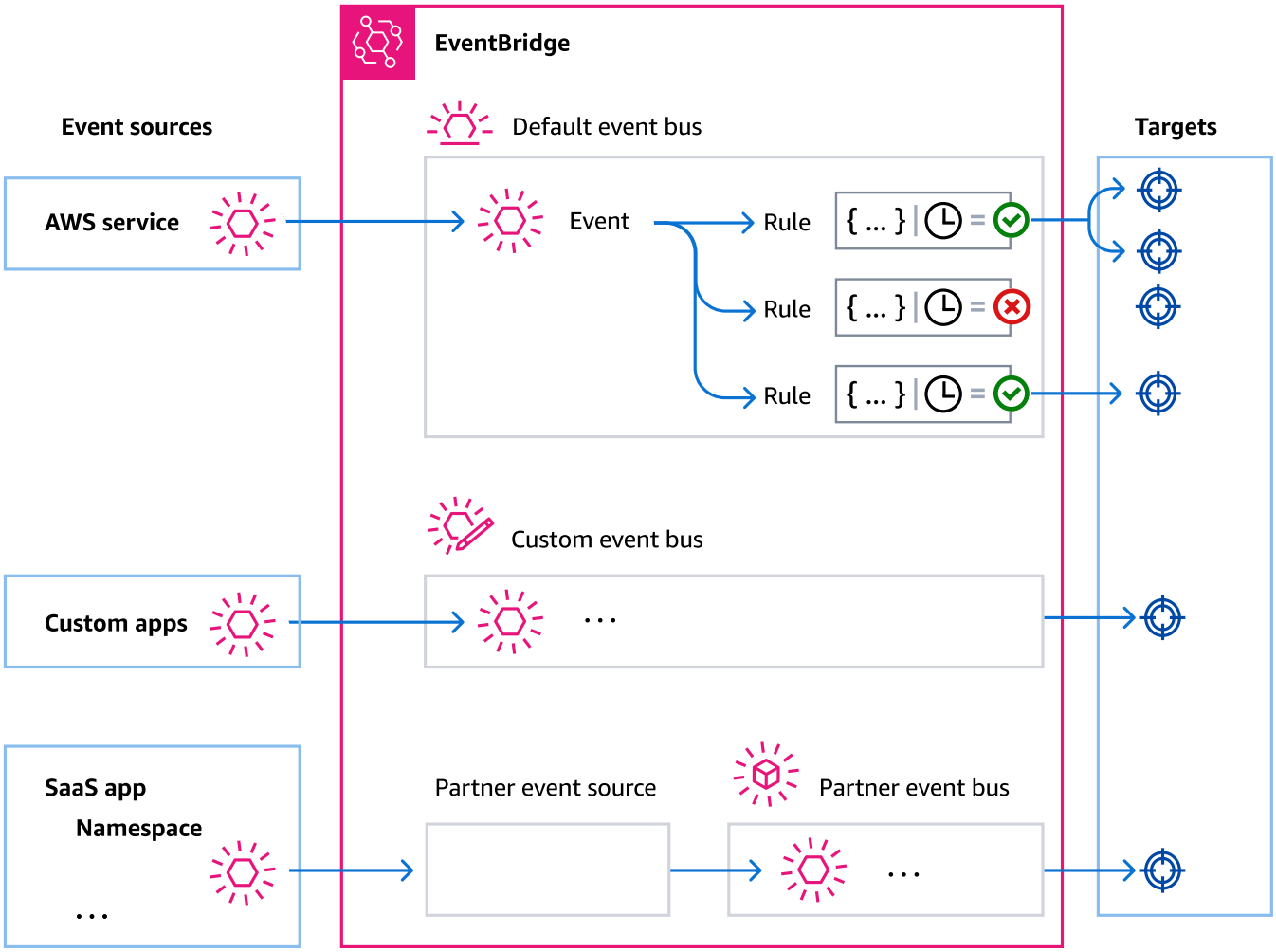
Un router di eventi è un router che riceve eventi e li invia a nessuna o a più destinazioni. Un router di eventi viene utilizzato per instradare eventi da un gran numero di origini a un gran numero di destinazioni, con la possibilità di trasformare gli eventi prima della distribuzione a una destinazione.

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 bus per elaborare eventi contenenti informazioni di identificazione personale (PII) e un altro bus per eventi che non lo fanno.
- 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 JSON oggetto inviato a un bus o pipe di eventi.

Nel contesto dell'architettura basata sugli 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**
  >
  > Sebbene sia possibile creare regole che vengono eseguite secondo una pianificazione, EventBridge ora offre un modo più flessibile e potente per creare, eseguire e gestire le

attività pianificate centralmente: Pianificatore EventBridge. Con Pianificatore EventBridge, puoi creare pianificazioni utilizzando le espressioni cron e rate per modelli ricorrenti o configurare chiamate una tantum. È possibile impostare finestre temporali flessibili per la consegna, definire limiti di tentativi e impostare il tempo massimo di conservazione per le chiamate non riuscite. API
Scheduler è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole pianificate, con una serie più ampia di operazioni e servizi mirati. API AWS Si consiglia di utilizzare 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 API delle destinazioni per abilitare REST API le chiamate tra i servizi

EventBridge APIIe destinazioni sono HTTP endpoint 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. Utilizzando API le destinazioni, puoi utilizzare API le chiamate per instradare eventi tra AWS servizi, applicazioni SaaS integrate e le tue applicazioni esterne. AWS Quando si crea una API destinazione, si specifica una connessione da utilizzare per tale destinazione. Ogni connessione include i dettagli sul tipo di autorizzazione e sui parametri da utilizzare per l'autorizzazione con l'endpoint di API destinazione.

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 lo 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 politiche basate sulle risorse per concedere autorizzazioni specifiche a ciascuna risorsa, ad esempio una funzione Lambda o un argomento Amazon. SNS

Creare un bus di eventi in Amazon EventBridge

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 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 KMS key opzioni.

- 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 a quella specificata chiave gestita dal cliente.

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

     EventBridge invia gli eventi che non sono stati crittografati correttamente aDLQ, se configurati, 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. È possibile incollare una politica da un'altra fonte o inserire la politica JSON per la politica. 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 KMS key opzioni.

- 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 KMS key opzioni.

- 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 eventi in Amazon EventBridge

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.

Per aggiornare un bus di eventi (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. Effettuare una o più delle seguenti operazioni:

   - Per aggiungere o rimuovere un archivio, seguite la procedura seguente:

     the section called “Aggiornamento degli archivi”
   - Per aggiungere o rimuovere tag, seguite la procedura seguente:

     the section called “Gestione dei tag dei bus degli eventi”
   - Per gestire le autorizzazioni del bus degli eventi, vedere la seguente procedura:

     the section called “Gestione delle autorizzazioni di un router di eventi”
   - Per modificare la AWS KMS chiave utilizzata per crittografare gli eventi, vedere la procedura seguente:

     the section called “Aggiornamento della AWS KMS chiave utilizzata su un bus di eventi”

Aggiornamento di un bus di eventi predefinito utilizzando AWS CloudFormation in EventBridge

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 un 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 eventi in Amazon EventBridge

È possibile eliminare un bus di eventi personalizzato o di un 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.

Generazione di un AWS CloudFormation modello da un bus di EventBridge eventi esistente

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 dai bus di eventi esistenti nel tuo account, come aiuto per iniziare subito a sviluppare modelli. CloudFormation Inoltre, EventBridge offre la possibilità di includere le regole associate a quel bus di eventi nel modello. È quindi possibile utilizzare questi modelli come base per creare pile di risorse da CloudFormation gestire.

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

> **Note**
>
> EventBridge non include regole gestite (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 CloudFormation modello da un bus 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. Scegli il bus degli eventi da cui desideri generare un CloudFormation modello.
4. Dal menu Azioni, scegliete CloudFormation Modello, quindi scegliete il formato in cui desiderate EventBridge generare il modello: JSONoppure 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 al riguardo EventBus.
5. EventBridge ti dà 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.
6. Per uscire dal modello, scegli Annulla.

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

Considerazioni sull'utilizzo di CloudFormation modelli generati da Amazon EventBridge

Considera i seguenti fattori quando utilizzi un CloudFormation modello generato da un bus di eventi:

- EventBridge non include alcuna password nel modello di generazione.

  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 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.

Autorizzazioni per gli event bus in Amazon EventBridge

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

EventBridge APIsche accettano un `Name` parametro del bus degli eventi come`PutRule`,`PutTargets`, `DeleteRule` `RemoveTargets`, `DisableRule`, e accettano `EnableRule` anche il bus degli ARN eventi. Utilizzate questi parametri per fare riferimento a bus di eventi tra account o regioni diverse tramite. APIs Ad esempio, puoi chiamare `PutRule` per creare una regola in un router di eventi in un account diverso senza dover assumere un ruolo.

È possibile allegare le politiche di esempio riportate in questo argomento a un IAM ruolo per concedere l'autorizzazione all'invio di eventi a un account o a una regione diversi. Utilizza IAM i ruoli per impostare le politiche di controllo dell'organizzazione e i limiti su chi può inviare eventi dal tuo account ad altri account. Ti consigliamo di utilizzare sempre IAM i ruoli quando l'obiettivo di una regola è un bus di eventi. È possibile allegare IAM ruoli utilizzando `PutTarget` le chiamate. Per informazioni sulla creazione di una regola per inviare eventi a un account o a una Regione differente, consulta Invio e ricezione di eventi tra AWS account in Amazon EventBridge.

Gestione delle autorizzazioni dei bus di eventi in Amazon EventBridge

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

Per gestire le autorizzazioni per un router di eventi esistente

1. Apri la EventBridge console Amazon 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 politica da un'altra fonte o inserire il campo JSON per la politica.
   - 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 inclusi più avanti in questa sezione.

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>"
          }
        }
    }]
  }
```

Politica di esempio: invio di eventi al bus predefinito in un altro account in Amazon EventBridge

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"
        }
    ]
  }
```

Politica di esempio: invio di eventi a un bus personalizzato in un altro account in Amazon EventBridge

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"
        }
      }
    }
  ]
}
```

Politica di esempio: invio di eventi a un bus di eventi nello stesso account in Amazon EventBridge

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"
      ]
    }
  ]
```

```
}
```

Esempio di politica: invio di eventi allo stesso account e limitazione degli aggiornamenti in Amazon EventBridge

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"
        }
      }
    }
  ]
}
```

Politica di esempio: invio di eventi da regole specifiche a un EventBridge bus Amazon 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'`Condition`istruzione limita gli eventi ai soli eventi che corrispondono alle regole con la regola specificata. ARN

```
{
  "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"
          ]
        }
      }
    }
  ]
}
```

Esempio di politica: invia eventi solo da una regione specifica a un'altra regione di Amazon EventBridge

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:*:*"
          ]
        }
      }
    }
  ]
}
```

Esempio di politica: nega l'invio di eventi da regioni specifiche di Amazon EventBridge

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:*:*"
          ]
        }
      }
    }
  ]
}
```

Eventi in Amazon EventBridge

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 passa da sospeso a in esecuzione.
- Amazon EC2 Auto Scaling genera eventi quando avvia o chiude 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 AWS relativi ai servizi in Amazon EventBridge e segui i collegamenti nella tabella.

Gli eventi sono rappresentati come JSON oggetti 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 AWS relativi ai servizi in Amazon EventBridge.

Argomenti

- Riferimento ai campi dei metadati degli eventi
- Invio di eventi con PutEvents Amazon EventBridge
- Eventi AWS relativi ai servizi in Amazon EventBridge
- In che modo EventBridge riprova a fornire eventi
- Utilizzo di code di lettere non recapitate per elaborare eventi non consegnati in EventBridge

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

Il video seguente illustra il modo in cui arrivano gli eventi EventBridge: Da dove provengono gli eventi

Riferimento ai campi dei metadati 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

Una versione 4 UUID generata 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 JSON array ARNs che contiene le risorse identificative coinvolte nell'evento. Il servizio che genera l'evento determina se includerleARNs. Ad esempio, le modifiche allo stato delle EC2 istanze Amazon includono l'EC2istanza AmazonARNs, gli eventi Auto Scaling ARNs includono sia le istanze che i gruppi di Auto Scaling, API ma le chiamate AWS CloudTrail con non includono la risorsa. ARNs

detail (dettaglio)

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

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

> **Note**
>
> PutEventsaccetta dati in JSON formato. Per il tipo di dati JSON numerico (intero), i vincoli sono: un valore minimo di -9.223.372.036.854.775.808 e un valore massimo di 9.223.372.036.854.775.807.

Example Esempio: notifica di modifica dello stato dell'EC2istanza Amazon

Il seguente evento in Amazon EventBridge indica la chiusura di un'EC2istanza Amazon.

```
{
  "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": {
  }
}
```

Invio di eventi con **`PutEvents`** Amazon EventBridge

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

Ogni richiesta `PutEvents` può supportare un numero limitato di voci. Per ulteriori informazioni, consulta EventBridge Quote Amazon. 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.

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\" }"}]'
```

È inoltre possibile creare un JSON file 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 immissione PutEvents degli eventi

Quando invii eventi personalizzati EventBridge utilizzando l'`PutEvents`azione, puoi raggruppare più eventi in un'unica richiesta di efficienza. Tuttavia, 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 dimensione della 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 per la JSON rappresentazione dell'evento. Per ulteriori informazioni, consulta Eventi in Amazon EventBridge.

EventBridge calcola la `PutEventsRequestEntry` dimensione come segue:

- Se specificato, il parametro `Time` è di 14 byte.
- I `DetailType` parametri `Source` and sono il numero di byte per i rispettivi formati codificati UTF -8.
- Se specificato, il `Detail` parametro è il numero di byte per il formato codificato UTF -8.
- Se specificato, ogni voce del `Resources` parametro è il numero di byte per i suoi formati codificati 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 AWS relativi ai servizi in Amazon EventBridge

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 il massimo impegno 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 AWS API le chiamate. 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 EC2 API chiamata Amazon che avvia o interrompe un'istanza genera EventBridge eventi oltre a eventi CloudTrail.

CloudTrail consente sia API ai chiamanti che ai proprietari di risorse di ricevere eventi nei loro bucket Amazon S3 creando percorsi e distribuisce eventi API ai chiamanti. EventBridge I proprietari delle risorse, oltre ai API chiamanti, possono monitorare le chiamate tra account tramite. API 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 API chiamata 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 API chiamate 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 dei 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 gli eventi AWS Key Management Service `GetKeyPolicy` and`DescribeKey`, or IAM `GetPolicy` and, si verificano a un volume molto più elevato rispetto ai tipici `GetRole` 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 seguenti AWS STS APIs 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`Describe`, o di `List` denominazione`Get`, per servizi, vedere. AWS ???

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

- 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'attivazione di una regola per gli eventi di CloudWatch gestione è supportata solo tramite i AWS CloudFormation modelli AWS CLI and.

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 `AssumeRole` di gestione di AWS Security Token Service.

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

AWS servizi che generano eventi in EventBridge

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 (AmazonEC2)	Migliore
Amazon EC2 Auto Scaling	Migliore
EC2Flotte Amazon	Migliore
Interruzione dell'istanza Amazon EC2 Spot	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 CMKcancel lazione	Durevole
AWS Key Management Service CMKrotazione	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 dei sistemi di misurazione	Migliore
AWS Migration Hub	Migliore
AWS Migration Hub Orchestratore	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

Servizio	Tipo di tentativo
Amazon QuickSight	Migliore
Amazon RDS	Migliore
AWS Cestino di riciclaggio	Migliore
Amazon Redshift	Durevole
Dati Amazon Redshift API	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

Servizio	Tipo di tentativo
Amazon Simple Storage Service (Amazon S3)	Durevole
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

Servizio	Tipo di tentativo
AWS X-Ray	Migliore

Eventi di gestione generati dai AWS servizi in EventBridge

In generale, APIs gli eventi che generano eventi di gestione (o di 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	APIchiamata
Alexa for Business	SearchAddressBooks	APIchiamata
Alexa for Business	SearchContacts	APIchiamata
Alexa for Business	SearchDevices	APIchiamata
Alexa for Business	SearchProfiles	APIchiamata
Alexa for Business	SearchRooms	APIchiamata
Alexa for Business	SearchSkillGroups	APIchiamata
Alexa for Business	SearchUsers	APIchiamata
Access Analyzer di IAM	ValidatePolicy	APIchiamata
AWS AdSpace Camere pulite	BatchGetSchema	APIchiamata
AWS Amplify Generatore di interfacce utente	ExportComponents	APIchiamata

Servizio	Nome evento	Tipo di evento
AWS Amplify Generatore di interfacce utente	ExportForms	APIchiamata
AWS Amplify Generatore di interfacce utente	ExportThemes	APIchiamata
OpenSearch Servizio Amazon	BatchGetCollection	APIchiamata
Amazon API Gateway	ExportApi	APIchiamata
AWS AppConfig	ValidateConfiguration	APIchiamata
Amazon AppFlow	RetrieveConnectorData	APIchiamata
Informazioni approfondite sulle CloudWatch applicazioni Amazon	UpdateApplicationDashboardConfiguration	APIchiamata
Amazon Athena	BatchGetNamedQuery	APIchiamata
Amazon Athena	BatchGetPreparedStatement	APIchiamata
Amazon Athena	BatchGetQueryExecution	APIchiamata
Amazon Athena	CheckQueryCompatibility	APIchiamata
Amazon Athena	ExportNotebook	APIchiamata
AWS Auto Scaling	AreScalableTargetsRegistered	APIchiamata
AWS Auto Scaling	Test	APIchiamata
Marketplace AWS	SearchAgreements	APIchiamata
AWS Backup	CreateLegalHold	APIchiamata
AWS Backup	ExportBackupPlanTemplate	APIchiamata
AWS Backup gateway	TestHypervisorConfiguration	APIchiamata

Servizio	Nome evento	Tipo di evento
AWS Billing and Cost Management	AWSPaymentInstrumentGateway.Ottenere	Azione da console
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

Servizio	Nome evento	Tipo di evento
AWS Billing and Cost Management	AWSPaymentPortalService.GetKybcValidationStatus	Azione da console
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

Servizio	Nome evento	Tipo di evento
AWS Billing and Cost Management	AWSPaymentPortalService.GetTransactionsHistory	Azione da console
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	D ownloadECSVFor BillingPeriod	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

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

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

Servizio	Nome evento	Tipo di evento
AWS CodeCommit	EvaluatePullRequestApprovalRules	APIchiamata
AWS CodeCommit	GitPull	APIchiamata
AWS CodeDeploy	BatchGetApplicationRevisions	APIchiamata
AWS CodeDeploy	BatchGetApplications	APIchiamata
AWS CodeDeploy	BatchGetDeploymentGroups	APIchiamata
AWS CodeDeploy	BatchGetDeploymentInstances	APIchiamata
AWS CodeDeploy	BatchGetDeployments	APIchiamata
AWS CodeDeploy	BatchGetDeploymentTargets	APIchiamata
AWS CodeDeploy	BatchGetOnPremisesInstances	APIchiamata
Amazon CodeGuru Profiler	BatchGetFrameMetricData	APIchiamata
Amazon CodeGuru Profiler	SubmitFeedback	APIchiamata
AWS CodePipeline	PollForJobs	APIchiamata
AWS CodePipeline	PollForThirdPartyJobs	APIchiamata
CodeConnections	StartAppRegistrationHandshake	APIchiamata
CodeConnections	Una tartOAuth stretta di mano	APIchiamata
CodeConnections	ValidateHostWebhook	APIchiamata
Amazon CodeWhisperer	CreateCodeScan	APIchiamata
Amazon CodeWhisperer	CreateProfile	APIchiamata

Servizio	Nome evento	Tipo di evento
Amazon CodeWhisperer	CreateUploadUrl	APIchiamata
Amazon CodeWhisperer	GenerateRecommendations	APIchiamata
Amazon CodeWhisperer	UpdateProfile	APIchiamata
Amazon Cognito Identity	LookupDeveloperIdentity	APIchiamata
Pool di utenti Amazon Cognito	AdminGetDevice	APIchiamata
Pool di utenti Amazon Cognito	AdminGetUser	APIchiamata
Pool di utenti Amazon Cognito	AdminListDevices	APIchiamata
Pool di utenti Amazon Cognito	AdminListGroupsForUser	APIchiamata
Pool di utenti Amazon Cognito	AdminListUserAuthEvents	APIchiamata
Pool di utenti Amazon Cognito	Beta_Authorize_ GET	Evento del servizio
Pool di utenti Amazon Cognito	Conferma_ GET	Evento del servizio
Pool di utenti Amazon Cognito	ConfirmForgotPassword_GET	Evento del servizio
Pool di utenti Amazon Cognito	Errore_ GET	Evento del servizio
Pool di utenti Amazon Cognito	ForgotPassword_GET	Evento del servizio
Pool di utenti Amazon Cognito	IntrospectToken	APIchiamata
Pool di utenti Amazon Cognito	Errore_di accesso_ POST	Evento del servizio
Pool di utenti Amazon Cognito	Accedi_ GET	Evento del servizio
Pool di utenti Amazon Cognito	Mfa_ GET	Evento del servizio
Pool di utenti Amazon Cognito	MfaOption_GET	Evento del servizio
Pool di utenti Amazon Cognito	ResetPassword_GET	Evento del servizio

Servizio	Nome evento	Tipo di evento
Pool di utenti Amazon Cognito	Registrati _ GET	Evento del servizio
Pool di utenti Amazon Cognito	UserInfo_GET	Evento del servizio
Pool di utenti Amazon Cognito	UserInfo_POST	Evento del servizio
Amazon Cognito Sync	BulkPublish	APIchiama
Amazon Comprehend	BatchContainsPiiEntities	APIchiamata
Amazon Comprehend	BatchDetectDominantLanguage	APIchiamata
Amazon Comprehend	BatchDetectEntities	APIchiamata
Amazon Comprehend	BatchDetectKeyPhrases	APIchiamata
Amazon Comprehend	BatchDetectPiiEntities	APIchiamata
Amazon Comprehend	BatchDetectSentiment	APIchiamata
Amazon Comprehend	BatchDetectSyntax	APIchiamata
Amazon Comprehend	BatchDetectTargetedSentiment	APIchiamata
Amazon Comprehend	ClassifyDocument	APIchiamata
Amazon Comprehend	ContainsPiiEntities	APIchiamata
Amazon Comprehend	DetectDominantLanguage	APIchiamata
Amazon Comprehend	DetectEntities	APIchiamata
Amazon Comprehend	DetectKeyPhrases	APIchiamata
Amazon Comprehend	DetectPiiEntities	APIchiamata
Amazon Comprehend	DetectSentiment	APIchiamata

Servizio	Nome evento	Tipo di evento
Amazon Comprehend	DetectSyntax	APIchiamata
Amazon Comprehend	DetectTargetedSentiment	APIchiamata
Amazon Comprehend	DetectToxicContent	APIchiamata
AWS Compute Optimizer	ExportAutoScalingGroupRecommendations	APIchiamata
AWS Compute Optimizer	xportEBSVolumeRaccomandazioni E	APIchiamata
AWS Compute Optimizer	xportECInstanceRaccomandazioni E	APIchiamata
AWS Compute Optimizer	xportECSServiceRaccomandazioni E	APIchiamata
AWS Compute Optimizer	ExportLambdaFunctionRecommendations	APIchiamata
AWS Compute Optimizer	xportRDSInstanceRaccomandazioni E	APIchiamata
AWS Config	BatchGetAggregateResourceConfig	APIchiamata
AWS Config	BatchGetResourceConfig	APIchiamata
AWS Config	SelectAggregateResourceConfig	APIchiamata
AWS Config	SelectResourceConfig	APIchiamata
Amazon Connect	AdminGetEmergencyAccessToken	APIchiamata
Amazon Connect	SearchQueues	APIchiamata

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

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

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	APIchiamata
Amazon Elastic Container Service	DiscoverPollEndpoint	APIchiamata
Amazon Elastic Container Service	FindSubfleetRoute	APIchiamata
Amazon Elastic Container Service	ValidateResources	APIchiamata
Amazon Elastic Container Service	VerifyTaskSetsExist	APIchiamata
Amazon Elastic Kubernetes Service	AccessKubernetesApi	APIchiamata
AWS Elastic Beanstalk	C heckDNSAvailability	APIchiamata
AWS Elastic Beanstalk	RequestEnvironmentInfo	APIchiamata
AWS Elastic Beanstalk	RetrieveEnvironmentInfo	APIchiamata
AWS Elastic Beanstalk	ValidateConfigurationSettings	APIchiamata
Amazon Elastic File System	NewClientConnection	Evento del servizio
Amazon Elastic File System	UpdateClientConnection	Evento del servizio
Amazon Elastic Transcoder	ReadJob	APIchiamata
Amazon Elastic Transcoder	ReadPipeline	APIchiamata

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

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

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

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

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

Servizio	Nome evento	Tipo di evento
AWS Key Management Service	GenerateDataKeyPairWithoutPlaintext	APIchiamata
AWS Key Management Service	GenerateDataKeyWithoutPlaintext	APIchiamata
AWS Key Management Service	GenerateMac	APIchiamata
AWS Key Management Service	GenerateRandom	APIchiamata
AWS Key Management Service	ReEncrypt	APIchiamata
AWS Key Management Service	Sign	APIchiamata
AWS Key Management Service	Verifica	APIchiamata
AWS Key Management Service	VerifyMac	APIchiamata
AWS Lake Formation	SearchDatabasesByLFTags	APIchiamata
AWS Lake Formation	SearchTablesByLFTags	APIchiamata
AWS Lake Formation	StartQueryPlanning	APIchiamata
Amazon Lex	BatchCreateCustomVocabularyItem	APIchiamata
Amazon Lex	BatchDeleteCustomVocabularyItem	APIchiamata
Amazon Lex	BatchUpdateCustomVocabularyItem	APIchiamata

Servizio	Nome evento	Tipo di evento
Amazon Lex	DeleteCustomVocabulary	APIchiamata
Amazon Lex	SearchAssociatedTranscripts	APIchiamata
Amazon Lightsail	C reateGUISession AccessDet ails	APIchiamata
Amazon Lightsail	DownloadDefaultKeyPair	APIchiamata
Amazon Lightsail	IsVpcPeered	APIchiamata
CloudWatch Registri Amazon	FilterLogEvents	APIchiamata
Amazon Macie	BatchGetCustomDataIdentifie rs	APIchiamata
Amazon Macie	UpdateFindingsFilter	APIchiamata
AWS Elemental MediaConn ect	ManagedDescribeFlow	APIchiamata
AWS Elemental MediaConn ect	PrivateDescribeFlowMeta	APIchiamata
AWS Application Migration Service	OperationalDescribeJobLogIt ems	APIchiamata
AWS Application Migration Service	OperationalDescribeJobs	APIchiamata
AWS Application Migration Service	OperationalDescribeReplicat ionConfigurationTemplates	APIchiamata
AWS Application Migration Service	OperationalDescribeSourceSe rver	APIchiamata
AWS Application Migration Service	OperationalGetLaunchConfigu ration	APIchiamata

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

Servizio	Nome evento	Tipo di evento
Amazon QuickSight	SearchDataSets	APIchiamata
Amazon QuickSight	SearchDataSources	APIchiamata
Amazon QuickSight	SearchFolders	APIchiamata
Amazon QuickSight	SearchGroups	APIchiamata
Amazon QuickSight	SearchUsers	APIchiamata
Amazon Relational Database Service	DownloadCompleteDBLogFile	APIchiamata
Amazon Relational Database Service	D ownloadDBLog FilePortion	APIchiamata
Amazon Rekognition	CompareFaces	APIchiamata
Amazon Rekognition	DetectCustomLabels	APIchiamata
Amazon Rekognition	DetectFaces	APIchiamata
Amazon Rekognition	DetectLabels	APIchiamata
Amazon Rekognition	DetectModerationLabels	APIchiamata
Amazon Rekognition	DetectProtectiveEquipment	APIchiamata
Amazon Rekognition	DetectText	APIchiamata
Amazon Rekognition	RecognizeCelebrities	APIchiamata
Amazon Rekognition	SearchFaces	APIchiamata
Amazon Rekognition	SearchFacesByImage	APIchiamata
Amazon Rekognition	SearchUsers	APIchiamata
Amazon Rekognition	SearchUsersByImage	APIchiamata

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

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

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

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

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

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

Riferimento dettagliato EventBridge sugli eventi Amazon

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 “Campi di metadati degli eventi”.

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 “Campi di metadati 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 JSON oggetto che contiene 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 "Campi di metadati 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 JSON oggetto che contiene 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 “Campi di metadati 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 JSON oggetto che contiene 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"
  }
}
```

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 recapitato 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

Per evitare di perdere eventi dopo che non sono stati consegnati a una destinazione, configura una coda di lettere morte (DLQ) per ricevere tutti gli eventi non riusciti. Per ulteriori informazioni, consulta the section called “Utilizzo di code DLQ”.

Utilizzo di code di lettere non recapitate per elaborare eventi non consegnati in EventBridge

Per evitare di perdere gli eventi dopo che non vengono recapitati a una destinazione, potete configurare una dead-letter queue (DLQ) e inviarvi tutti gli eventi non riusciti per elaborarli in un secondo momento.

EventBridge DLQssono SQS code standard di Amazon EventBridge utilizzate per archiviare eventi che non è stato possibile consegnare correttamente a una destinazione. Quando crei una regola e aggiungi un obiettivo, puoi scegliere se utilizzare o meno unDLQ. Quando configuri unDLQ, puoi conservare tutti gli eventi che non sono stati consegnati 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 un obiettivo di una regola, EventBridge invia gli eventi con chiamate non riuscite alla SQS coda Amazon selezionata.

Gli errori relativi agli eventi vengono gestiti in modi diversi. Alcuni eventi vengono eliminati o inviati a un DLQ senza alcun tentativo di nuovo tentativo. Ad esempio, per gli errori derivanti dalla mancanza di autorizzazioni per una destinazione o se si tratta di una risorsa di destinazione che non esiste più, non verrà effettuato alcun tentativo finché non verrà intrapresa un'azione per risolvere il problema sottostante. EventBridge invia questi eventi direttamente alla destinazioneDLQ, se ne hai specificata una.

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 unDLQ, vengono inviate metriche aggiuntive a CloudWatch Includer and. `InvocationsSentToDLQ` `InvocationsFailedToBeSentToDLQ`

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

Ogni tuo messaggio DLQ includerà i seguenti attributi personalizzati:

- `RULE_ARN`
- `TARGET_ARN`
- `ERROR_CODE`

  Di seguito è riportato un esempio dei codici di errore che è DLQ possibile 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 ripercorre le impostazioniDLQs: Utilizzo delle code di lettere morte () DLQs

Argomenti

- Considerazioni sull'utilizzo di una coda DLQ
- Concessione delle autorizzazioni per la coda DLQ

- Come inviare nuovamente eventi da una coda DLQ

Considerazioni sull'utilizzo di una coda DLQ

Considerate quanto segue quando configurate un DLQ for. EventBridge

- Sono supportate solo le code standard. Non puoi usare una FIFO coda per entrare. DLQ EventBridge
- EventBridge include i metadati degli eventi e gli attributi del messaggio, tra cui: il codice di errore, il messaggio di errore, la condizione di ripetizione esaurita, la regolaARN, i tentativi di ripetizione e la destinazione. ARN È possibile utilizzare questi valori per identificare un evento e la causa dell'errore.
- Autorizzazioni per DLQs lo stesso account:
  - Se aggiungi un obiettivo a una regola utilizzando la console e scegli una SQS coda Amazon nello stesso account, alla coda viene allegata automaticamente una policy basata sulle risorse che EventBridge concede l'accesso alla coda.
  - Se utilizzi l'`PutTargets`operazione di per EventBridge API aggiungere o aggiornare un obiettivo per una regola e scegli una SQS coda Amazon 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 di Amazon SQS Queues da un altro AWS account.
  - Se crei una regola dalla console, non vengono visualizzate le code di altri account che puoi selezionare. Devi fornire le informazioni ARN per la coda nell'altro account, quindi allegare manualmente una policy basata sulle risorse per concedere l'autorizzazione alla coda. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.
  - Se si crea una regola utilizzando ilAPI, è necessario allegare manualmente una politica basata sulle risorse alle code di un altro account utilizzato come SQS coda delle lettere morte. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.
- La SQS coda Amazon che usi deve trovarsi nella stessa regione in cui crei la regola.

Concessione delle autorizzazioni per la coda DLQ

Per inviare correttamente gli eventi alla coda, è EventBridge necessario disporre dell'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 a riposo.

  Per ulteriori informazioni, consulta ???.

Se specifichi un DLQ utilizzo di 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. API

Esempio di autorizzazioni Target per una coda di lettere non scritte

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 politica concede al EventBridge servizio le autorizzazioni per utilizzare l'`SendMessage`operazione per inviare messaggi a una coda denominata "». MyEvent DLQ 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 mostra come concedere le autorizzazioni richieste quando si specifica un per un bus di eventi. DLQ In questo caso, `aws:SourceArn` specifica il bus ARN degli eventi che invia gli eventi a. 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 allegare la politica alla coda, usa la SQS console Amazon, apri la coda, quindi scegli la politica di accesso e modifica la politica. Puoi anche utilizzare l' AWS CLI. Per ulteriori informazioni, consulta SQSAutorizzazioni Amazon.

Come inviare nuovamente eventi da una coda DLQ

Puoi spostare i messaggi da a DLQ in due modi:

- Evita di scrivere la logica SQS dei consumatori di Amazon: imposta la tua fonte di eventi DLQ come fonte di eventi sulla funzione Lambda per esaurire il tuo. DLQ
- Scrivi la logica di SQS consumo di Amazon: utilizza Amazon SQS API o per AWS CLI scrivere una logica di consumo personalizzata per il polling, l'elaborazione e l'eliminazione dei messaggi in. AWS SDK DLQ

Regole in Amazon EventBridge

È possibile 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.

È possibile creare due tipi di regole: regole che corrispondono ai dati degli eventi man mano che gli eventi vengono consegnati e regole che vengono eseguite secondo una pianificazione definita. Inoltre, alcuni AWS servizi possono creare e gestire regole anche nel tuo account.

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

> **Note**
>
> Sebbene sia possibile creare regole che vengono eseguite secondo una pianificazione, EventBridge ora offre un modo più flessibile e potente per creare, eseguire e gestire le attività pianificate centralmente: Pianificatore EventBridge. Con Pianificatore EventBridge, puoi creare pianificazioni utilizzando le espressioni cron e rate per modelli ricorrenti o configurare chiamate una tantum. È possibile impostare finestre temporali flessibili per la consegna, definire limiti di tentativi e impostare il tempo massimo di conservazione per le chiamate non riuscite. API
> Scheduler è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole pianificate, con una serie più ampia di operazioni e servizi mirati. API AWS Si consiglia di utilizzare Scheduler per richiamare gli obiettivi in base a una pianificazione.
> Per ulteriori informazioni, consulta ???.

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.

Per ulteriori informazioni, consulta ???.

Regole gestite dai AWS servizi

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. IAMle politiche create in questo modo hanno un ambito ristretto di autorizzazioni a livello di risorsa per consentire la creazione solo delle regole necessarie.

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.

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

Creazione di regole che reagiscono agli eventi in Amazon EventBridge

Per intervenire sugli eventi ricevuti da Amazon EventBridge, puoi creare regole. Quando un evento corrisponde al modello di evento definito nella regola, EventBridge invia l'evento al target specificato 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 seguente procedura per creare una EventBridge regola Amazon che risponda agli eventi.

I passaggi seguenti illustrano come creare una regola da EventBridge utilizzare per abbinare gli eventi man mano che vengono inviati al bus di eventi specificato.

Fasi

- 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. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli 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 nello stesso 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 AWS servizio utente emette un evento, questo passa sempre al bus degli eventi predefinito del tuo account.

6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Seleziona 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 lo schema degli eventi JSON e testarlo rispetto a un evento di esempio.

Per creare il modello di eventi

1. Per Event source, scegli AWS eventi o eventi EventBridge partner.
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:

   - AWS eventi: seleziona tra gli eventi emessi da AWS servizi Supported.
   - EventBridge eventi partner: seleziona tra gli eventi emessi da servizi di terze parti che supportano EventBridge, come Salesforce.
   - Inserisci il mio: inserisci il tuo evento nel 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. Eliminate la struttura degli eventi di esempio nel riquadro di JSON modifica e incollate l'evento AWS o l'evento partner al suo posto.

6. Modifica l'evento JSON per creare il tuo evento di esempio.

3. In Metodo di creazione, scegli un metodo di creazione. È possibile creare un modello di evento da EventBridge uno 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. Scegliete Genera modello di evento in JSON per generare e convalidare il modello di evento come JSON testo.
   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 dell'evento.

È anche possibile scegliere una delle seguenti opzioni:

- Copia: copia il modello di eventi negli appunti del dispositivo.
- Prettify: semplifica la lettura del JSON testo 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. È necessario formattare lo schema come testo valido. JSON
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 modello di evento in JSON per generare e convalidare il modello di evento come JSON testo.
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 dell'evento.

È anche possibile scegliere una delle seguenti opzioni:

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

Event pattern

Per scrivere un modello di evento personalizzato in JSON formato, effettuate le seguenti operazioni:

1. Nella sezione Metodo di creazione, per Metodo, scegliete Modello personalizzato (JSONeditor).
2. Per Schema di evento, inserisci il modello di evento personalizzato in un JSON testo formattato.
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 evento.

   È anche possibile scegliere una delle seguenti opzioni:

   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Prettify: semplifica la lettura del JSON testo 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 che Pattern Builder venga aperto.

4. Seleziona Successivo.

Selezionare le destinazioni

Scegli una o più destinazioni per ricevere eventi che corrispondono al modello specificato. Gli obiettivi possono includere un bus per EventBridge eventi, EventBridge API destinazioni, inclusi partner SaaS come Salesforce o altro. AWS servizio

Per selezionare le destinazioni

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

   Event bus

   Per selezionare un bus di EventBridge eventi, seleziona EventBridge Event bus, quindi procedi come segue:

   - Per utilizzare un bus di eventi nello stesso modo in cui viene applicata questa Regione AWS 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 bus di eventi in un account Regione AWS or diverso come segue:
     1. Seleziona Bus di eventi in un account diverso o in una Regione diversa.
     2. Per Event bus come destinazione, inserisci il ARN bus degli eventi che desideri utilizzare.

        Per ulteriori informazioni, consultare:

        - ???
        - ???

   API destination

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

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

- Per creare una nuova API destinazione, seleziona Crea una nuova API destinazione. 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.
  - APIendpoint di destinazione: l'URLendpoint per la destinazione.

    L'endpoint URL deve iniziare con. **`https`** Puoi includere il carattere jolly **`*`** come parametro di percorso. Puoi impostare i parametri di percorso dall'attributo `HttpParameters` della destinazione.
  - HTTPmetodo: seleziona il HTTP metodo 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 ???.

AWS servizio

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

1. In Seleziona una destinazione, seleziona un AWS servizio 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 Target del bus degli eventi disponibili nella EventBridge console.

2. Per molti tipi di oggetto, EventBridge sono necessarie le autorizzazioni per inviare eventi alla destinazione. In questi casi, EventBridge può creare il IAM ruolo 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. Immettete un nome per questo ruolo di esecuzione o utilizzate 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) Per la coda di lettere morte, scegli se utilizzare una SQS coda Amazon 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 SQS coda Amazon nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dall'elenco a discesa.
   - Scegli Seleziona una SQS coda Amazon in un altro AWS account come coda di lettere non scritte, quindi inserisci la ARN coda da utilizzare. È necessario allegare alla coda una politica basata sulle risorse che autorizza l'invio di messaggi. EventBridge

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. Seleziona 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: invia EventBridge solo la parte specificata dell'evento di origine originale alla destinazione.

     In Specificate la parte dell'evento corrispondente, specificate un JSON percorso che definisca la parte dell'evento che desiderate inviare EventBridge alla destinazione.
   - Constant (JSONtesto): EventBridge invia solo il JSON testo specificato alla destinazione. Non viene inviata alcuna parte dell'evento di origine originale.

     In Specificate la costante in JSON, specificate il JSON testo che desiderate inviare EventBridge alla destinazione anziché all'evento.
   - Trasformatore di ingresso: configura un trasformatore di input per personalizzare il testo che desideri EventBridge 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 Politica Riprova, specificate come EventBridge ripetere l'invio di un evento a una destinazione dopo che si è verificato un errore.
   - Età massima dell'evento: immettete il periodo di tempo massimo (in ore, minuti e secondi) per EventBridge conservare gli eventi non elaborati. Il valore predefinito è 24 ore.
   - Tentativi di nuovo tentativo: immettete il numero massimo di volte in cui è EventBridge necessario riprovare a inviare un evento alla destinazione dopo che si è verificato un errore. L'impostazione predefinita è 185 volte.
3. (Facoltativo) Per la coda di lettere morte, scegli se utilizzare una SQS coda Amazon 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 SQS coda Amazon nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dall'elenco a discesa.
- Scegli Seleziona una SQS coda Amazon in un altro AWS account come coda di lettere non scritte, quindi inserisci la ARN coda da utilizzare. È necessario allegare alla coda una politica basata sulle risorse che autorizza l'invio di messaggi. EventBridge

  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. Seleziona 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: invia EventBridge solo la parte specificata dell'evento di origine originale alla destinazione.

     In Specificate la parte dell'evento corrispondente, specificate un JSON percorso che definisca la parte dell'evento che desiderate inviare EventBridge alla destinazione.
   - Constant (JSONtesto): EventBridge invia solo il JSON testo specificato alla destinazione. Non viene inviata alcuna parte dell'evento di origine originale.

     In Specificate la costante in JSON, specificate il JSON testo che desiderate inviare EventBridge alla destinazione anziché all'evento.
   - Trasformatore di ingresso: configura un trasformatore di input per personalizzare il testo che desideri EventBridge 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 Politica Riprova, specificate come EventBridge ripetere l'invio di un evento a una destinazione dopo che si è verificato un errore.
   - Età massima dell'evento: immettete il periodo di tempo massimo (in ore, minuti e secondi) per EventBridge conservare gli eventi non elaborati. Il valore predefinito è 24 ore.
   - Tentativi di nuovo tentativo: immettete il numero massimo di volte in cui è EventBridge necessario riprovare a inviare un evento alla destinazione dopo che si è verificato un errore. L'impostazione predefinita è 185 volte.
3. (Facoltativo) Per la coda di lettere morte, scegli se utilizzare una SQS coda Amazon 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 SQS coda Amazon nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dall'elenco a discesa.
   - Scegli Seleziona una SQS coda Amazon in un altro AWS account come coda di lettere non scritte, quindi inserisci la ARN coda da utilizzare. È necessario allegare alla coda una politica basata sulle risorse che autorizza l'invio di messaggi. EventBridge

     Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la coda DLQ.
4. (Facoltativo) Scegli Aggiungi destinazione per aggiungere un’altra destinazione per questa regola.
5. Seleziona 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 la pagina Etichettare le risorse in Amazon EventBridge.
2. Seleziona Next (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.

Creazione di una regola che viene eseguita secondo una pianificazione in Amazon EventBridge

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**
>
> Sebbene sia possibile creare regole che vengono eseguite secondo una pianificazione, EventBridge ora offre un modo più flessibile e potente per creare, eseguire e gestire le attività pianificate centralmente: Pianificatore EventBridge. Con Pianificatore EventBridge, puoi creare pianificazioni utilizzando le espressioni cron e rate per modelli ricorrenti o configurare chiamate una tantum. È possibile impostare finestre temporali flessibili per la consegna, definire limiti di tentativi e impostare il tempo massimo di conservazione per le chiamate non riuscite. API
> Scheduler è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole pianificate, con una serie più ampia di operazioni e servizi mirati. API AWS Si consiglia di utilizzare Scheduler per richiamare gli obiettivi in base a una pianificazione.
> Per ulteriori informazioni, consulta ???.

In EventBridge, puoi 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. PSTIl 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.

Tutti gli eventi programmati utilizzano il fuso orario UTC +0 e la precisione minima per una pianificazione è di un minuto.

> **Note**
>
> EventBridge non fornisce una precisione di secondo livello nelle espressioni di pianificazione. La risoluzione più alta che utilizza un'espressione Cron è un minuto. A causa della natura distribuita dei servizi di destinazione EventBridge e dei servizi di destinazione, può verificarsi un ritardo di diversi secondi tra l'attivazione della regola pianificata e il momento in cui il servizio di destinazione esegue la risorsa di destinazione.

Il video seguente offre una panoramica delle attività di pianificazione: Creazione di attività pianificate con EventBridge

Argomenti

- Creazione di una regola eseguita in base a una pianificazione

Creazione di una regola eseguita in base a una pianificazione

I passaggi seguenti illustrano come creare una EventBridge regola che venga eseguita secondo una pianificazione regolare.

> **Note**
>
> Puoi creare regole pianificate solo utilizzando il router di eventi predefinito.

Fasi

- 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. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli 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 nello stesso 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 secondo una pianificazione o utilizzare Amazon EventBridge Scheduler.
8. Scegli come vuoi continuare:

   - Usa EventBridge Scheduler per creare la tua pianificazione

     > **Note**
     >
     > EventBridge Scheduler è uno strumento di pianificazione senza server che 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. EventBridge Scheduler è altamente personalizzabile e offre una

migliore scalabilità rispetto alle regole EventBridge pianificate, con una serie più ampia di operazioni e servizi mirati. API AWS
Si consiglia di utilizzare EventBridge Scheduler per richiamare gli obiettivi in base a una pianificazione. Per ulteriori informazioni, consulta Cos'è Amazon EventBridge Scheduler? nella Amazon EventBridge Scheduler User Guide.

1. Seleziona Continua in Scheduler EventBridge

   EventBridge apre la console EventBridge Scheduler alla pagina Crea pianificazione.
2. Crea la pianificazione nella console EventBridge Scheduler.

- Continua a utilizzare EventBridge per creare una regola pianificata per il bus degli 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. PSTIl primo lunedì di ogni mese.
   2. Per l'espressione Cron, specifica i campi per definire l'espressione cron da EventBridge utilizzare per determinare quando eseguire questa regola pianificata.

      Dopo aver specificato tutti i campi, EventBridge mostra le dieci date successive in cui EventBridge verrà eseguita questa regola pianificata. È possibile scegliere se visualizzare tali date nel UTCfuso 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. Per l'espressione della velocità, specifica i campi Valore e Unità per definire la velocità alla quale EventBridge eseguire questa regola pianificata.

   Per ulteriori informazioni sulla creazione di un'espressione della frequenza, consulta ???.

2. Seleziona Successivo.

Selezionare le destinazioni

Scegli una o più destinazioni per ricevere eventi che corrispondono al modello specificato. Gli obiettivi possono includere un bus per EventBridge eventi, EventBridge API destinazioni, inclusi partner SaaS come Salesforce o altro. AWS servizio

Per selezionare le destinazioni

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

   Event bus

   Per selezionare un bus di EventBridge eventi, seleziona EventBridge Event bus, quindi procedi come segue:

   - Per utilizzare un bus di eventi Regione AWS come questa 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 bus di eventi in un account Regione AWS or diverso come segue:
     1. Seleziona Bus di eventi in un account diverso o in una Regione diversa.
     2. Per Event bus come destinazione, inserisci il ARN bus degli eventi che desideri utilizzare.

        Per ulteriori informazioni, consultare:

- ???
- ???

API destination

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

- Per utilizzare una API destinazione esistente, seleziona Usa una API destinazione esistente. Quindi seleziona una API destinazione dall'elenco a discesa.
- Per creare una nuova API destinazione, seleziona Crea una nuova API destinazione. 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.
  - APIendpoint di destinazione: l'URLendpoint per la destinazione.

    L'endpoint URL deve iniziare con. **`https`** Puoi includere il carattere jolly **`*`** come parametro di percorso. Puoi impostare i parametri di percorso dall'attributo `HttpParameters` della destinazione.
  - HTTPmetodo: seleziona il HTTP metodo 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 ???.

AWS servizio

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

1. In Seleziona una destinazione, seleziona un AWS servizio 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 Target del bus degli eventi disponibili nella EventBridge console.

2. Per molti tipi di oggetto, EventBridge sono necessarie le autorizzazioni per inviare eventi alla destinazione. In questi casi, EventBridge può creare il IAM ruolo 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. Inserisci un nome per questo ruolo di esecuzione o usa 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) Per la coda di lettere morte, scegli se utilizzare una SQS coda Amazon 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 SQS coda Amazon nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dall'elenco a discesa.
- Scegli Seleziona una SQS coda Amazon in un altro AWS account come coda di lettere non scritte, quindi inserisci la ARN coda da utilizzare. È necessario allegare alla coda una politica basata sulle risorse che autorizza l'invio di messaggi. EventBridge

  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. Seleziona 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: invia EventBridge solo la parte specificata dell'evento di origine originale alla destinazione.

     In Specificate la parte dell'evento corrispondente, specificate un JSON percorso che definisca la parte dell'evento che desiderate inviare EventBridge alla destinazione.
   - Constant (JSONtesto): EventBridge invia solo il JSON testo specificato alla destinazione. Non viene inviata alcuna parte dell'evento di origine originale.

     In Specificate la costante in JSON, specificate il JSON testo che desiderate inviare EventBridge alla destinazione anziché all'evento.
   - Trasformatore di ingresso: configura un trasformatore di input per personalizzare il testo che desideri EventBridge 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 Politica Riprova, specificate come EventBridge ripetere l'invio di un evento a una destinazione dopo che si è verificato un errore.
   - Età massima dell'evento: immettete il periodo di tempo massimo (in ore, minuti e secondi) per EventBridge conservare gli eventi non elaborati. Il valore predefinito è 24 ore.

- Tentativi di nuovo tentativo: immettete il numero massimo di volte in cui è EventBridge necessario riprovare a inviare un evento alla destinazione dopo che si è verificato un errore. L'impostazione predefinita è 185 volte.

3. (Facoltativo) Per la coda di lettere morte, scegli se utilizzare una SQS coda Amazon 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 SQS coda Amazon nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dall'elenco a discesa.
   - Scegli Seleziona una SQS coda Amazon in un altro AWS account come coda di lettere non scritte, quindi inserisci la ARN coda da utilizzare. È necessario allegare alla coda una politica basata sulle risorse che autorizza l'invio di messaggi. EventBridge

     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. Seleziona 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: invia EventBridge solo la parte specificata dell'evento di origine originale alla destinazione.

     In Specificate la parte dell'evento corrispondente, specificate un JSON percorso che definisca la parte dell'evento che desiderate inviare EventBridge alla destinazione.
   - Constant (JSONtesto): EventBridge invia solo il JSON testo specificato alla destinazione. Non viene inviata alcuna parte dell'evento di origine originale.

     In Specificate la costante in JSON, specificate il JSON testo che desiderate inviare EventBridge alla destinazione anziché all'evento.

- Trasformatore di ingresso: configura un trasformatore di input per personalizzare il testo che desideri EventBridge 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 Politica Riprova, specificate come EventBridge ripetere l'invio di un evento a una destinazione dopo che si è verificato un errore.

   - Età massima dell'evento: immettete il periodo di tempo massimo (in ore, minuti e secondi) per EventBridge conservare gli eventi non elaborati. Il valore predefinito è 24 ore.
   - Tentativi di nuovo tentativo: immettete il numero massimo di volte in cui è EventBridge necessario riprovare a inviare un evento alla destinazione dopo che si è verificato un errore. L'impostazione predefinita è 185 volte.

3. (Facoltativo) Per la coda di lettere morte, scegli se utilizzare una SQS coda Amazon 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 SQS coda Amazon nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dall'elenco a discesa.
   - Scegli Seleziona una SQS coda Amazon in un altro AWS account come coda di lettere non scritte, quindi inserisci la ARN coda da utilizzare. È necessario allegare alla coda una politica basata sulle risorse che autorizza l'invio di messaggi. EventBridge

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

4. (Facoltativo) Scegli Aggiungi destinazione per aggiungere un’altra destinazione per questa regola.
5. Seleziona 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 la pagina Etichettare le risorse in Amazon EventBridge.
2. Seleziona Next (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 delle espressioni cron e rate per pianificare le regole in Amazon EventBridge

Quando si crea una regola pianificata in, EventBridge è possibile specificare uno schema di pianificazione che determina quando EventBridge viene eseguita la regola:

- Usa un'espressione cron per eseguire la regola in orari e date specifici.
- Usa un'espressione di frequenza per eseguire la regola a intervalli regolari.

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	, - * /
D ay-of-month	1-31	, - * ? / L W
Mese	1-12 o JAN - DEC	, - * /

Campo	Valori	Caratteri jolly
D ay-of-week	1-7 o SUN - SAT	, - * ? L #
Anno	1970-2199	, - * /

Caratteri jolly

- Il carattere jolly , (virgola) include valori aggiuntivi. Nel campo Mese,JAN,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 puoi usare* in entrambi i ay-of-week campi D ay-of-month e D. 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 ay-of-month campo D puoi inserire 7 e se qualsiasi giorno della settimana fosse accettabile, potresti inserire? nel ay-of-week campo D.
- Il carattere L nei ay-of-week campi D ay-of-month o D specifica l'ultimo giorno del mese o della settimana.
- Il carattere **`W`** jolly nel ay-of-month campo D specifica un giorno della settimana. Nel ay-of-month campo D, **`3W`** specifica il giorno della settimana più vicino al terzo giorno del mese.
- Il carattere jolly # nel ay-of-week campo D specifica una determinata istanza del giorno della settimana specificato all'interno di 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 si utilizza un carattere '#', è possibile definire solo un'espressione nel day-of-week campo. Ad esempio, `"3#1,6#3"` non è valido perché viene interpretato come due espressioni.

Limitazioni

- Non è possibile specificare i ay-of-week campi D ay-of-month e D 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	*	*	?	*	Corri alle 10:00 (UTC+0) ogni giorno
15	12	*	*	?	*	Corri alle 12:15 (UTC+0) ogni giorno
0	18	?	*	MON-FRI	*	Corri alle 18:00 (UTC+0) dal lunedì al venerdì
0	8	1	*	?	*	Esegui alle 8:00 (UTC+0) ogni primo giorno del mese

Minuti	Ore	Giorno del mese	Mese	Giorno della settimana	Anno	Significato
0/15	*	*	*	?	*	Esegui ogni 15 minuti
0/10	*	?	*	MON-FRI	*	Esegui dal lunedì al venerdì ogni 10 minuti
0/5	8-17	?	*	MON-FRI	*	Corri ogni 5 minuti dal lunedì al venerdì tra le 8:00 e le 17:55 (+0) UTC

Minuti	Ore	Giorno del mese	Mese	Giorno della settimana	Anno	Significato
0/30	20-2	?	*	MON-FRI	*	Corri ogni 30 minuti dal lunedì al venerdì dalle 22:00 del giorno di partenza alle 2:00 del giorno successivo () UTC Corri dalle 12:00 alle 2:00 del lunedì mattina (). UTC

L'esempio seguente crea una regola che viene eseguita ogni giorno alle 12:00 +0. UTC

```
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 alle 14:35 +0. UTC

```
aws events put-rule --schedule-expression "cron(5,35 14 * * ? *)" --name MyRule2
```

L'esempio seguente crea una regola che viene eseguita alle 10:15 UTC +0 l'ultimo venerdì di ogni mese negli anni dal 2019 al 2022.

```
aws events put-rule --schedule-expression "cron(15 10 ? * 6L 2019-2022)" --name MyRule3
```

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

Negli esempi seguenti viene illustrato come utilizzare le espressioni di frequenza con il AWS CLI `put-rule` comando. 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
```

Disattivazione o eliminazione di una regola in 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 EventBridge regola.

Per eliminare o disabilitare una regola

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.

   In Event bus (Bus di eventi), selezionare il bus di eventi associato alla regola.
3. Esegui una di queste 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.

Le migliori pratiche per la definizione delle regole in 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 architettonico comune adottato dai clienti consiste nell'isolare questi componenti o servizi applicativi utilizzando AWS account 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 di politiche relative alle risorse, consulta Modelli di progettazione multiaccount con Amazon EventBridge on 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 anche attivare una limitazione delle regole quando l'account supera la quota di servizio prevista aggregata al secondo. EventBridge cercherà comunque di fornire eventi conformi a regole limitate e riproverà fino a 24 ore o come descritto nella politica di riprova personalizzata dell'obiettivo. 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 AWS Serverless Application Model di modelli per distribuire risorse Amazon EventBridge

Puoi creare e testare le regole manualmente nella EventBridge console, il che può aiutarti nel processo di sviluppo mentre perfezioni i modelli di eventi. Tuttavia, quando è tutto pronto per distribuire l'applicazione, è più semplice utilizzare un framework come AWS SAM per avviare tutte le risorse serverless in modo coerente.

Useremo quest'applicazione di esempio per esaminare i modi in cui è possibile utilizzare i AWS SAM modelli per creare EventBridge risorse. Il file template.yaml in questo esempio è un AWS SAM modello che definisce quattro AWS Lambdafunzioni e mostra due modi diversi per integrare le funzioni Lambda. EventBridge

Per una procedura guidata di questa applicazione di esempio, consulta ???.

Esistono due approcci all'utilizzo e ai modelli. EventBridge AWS SAM Per integrazioni semplici in cui una funzione Lambda viene richiamata da una regola, si consiglia l'approccio Modello combinato. Se si utilizza una logica di routing complessa o ci si connette a risorse esterne al AWS SAM modello, l'approccio basato su modelli separati è la scelta migliore.

Approcci:

- Modello combinato
- Modello separato

Modello combinato

Il primo approccio utilizza la `Events` proprietà per configurare la EventBridge regola. Il codice di esempio seguente definisce un evento che richiama la funzione Lambda.

> **Note**
>
> Questo esempio crea automaticamente la regola sul bus di eventi predefinito, che esiste in ogni AWS account. 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 YAML codice è equivalente a un pattern di eventi nella EventBridge console. InYAML, è sufficiente definire il modello di evento e creare AWS SAM automaticamente un IAM ruolo con le autorizzazioni richieste.

Modello separato

Nel secondo approccio alla definizione di una EventBridge configurazione 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 più 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 `AWS::Lambda::Permission` risorsa che conceda il permesso di EventBridge invocare l'obiettivo.

```
PermissionForEventsToInvokeLambda:
  Type: AWS::Lambda::Permission
  Properties:
    FunctionName:
      Ref: "atmConsumerCase1Fn"
    Action: "lambda:InvokeFunction"
    Principal: "events.amazonaws.com"
    SourceArn:
      Fn::GetAtt:
        - "EventRuleCase1"
        - "Arn"
```

Generazione di un AWS CloudFormation modello da una EventBridge regola esistente

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 regole esistenti nel tuo account, come aiuto per iniziare subito a sviluppare modelli. CloudFormation Puoi selezionare una singola regola o più regole da includere nel modello. È quindi possibile utilizzare questi modelli come base per creare pile di risorse da gestire. CloudFormation

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

> **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 AWS CloudFormation modello da una o più regole

1. Apri la EventBridge console Amazon 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 che desideri includere nel AWS CloudFormation modello generato.

   Per una singola regola, puoi anche scegliere il nome della regola per visualizzare la pagina dei dettagli della regola.
5. Scegliete CloudFormation Modello, quindi scegliete il formato in cui desiderate EventBridge generare il modello: JSONoppure YAML.

   EventBridge visualizza il modello, generato nel formato selezionato.
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.

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

Considerazioni sull'utilizzo di CloudFormation modelli generati da Amazon EventBridge

Quando utilizzi un CloudFormation modello generato da EventBridge te, considera i seguenti fattori:

- 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 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.

Obiettivi degli Event Bus in Amazon EventBridge

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

Target del bus degli eventi disponibili nella EventBridge console

È possibile configurare le seguenti destinazioni per gli eventi nella EventBridge console:

- APIdestinazione
- APIGateway

- AWS AppSync
- Coda di processi batch
- CloudWatch gruppo di log
- CodeBuild progetto
- CodePipeline
- EBS`CreateSnapshot`APIChiamata Amazon
- EC2Image Builder
- EC2`RebootInstances`APIchiamata
- EC2`StopInstances`APIchiamata
- EC2`TerminateInstances`APIchiamata
- ECScompito
- 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
- Interrogazioni sui dati del cluster Amazon Redshift API
- Interrogazioni sui dati dei gruppi di lavoro Serverless di Amazon Redshift API
- SageMaker Pipeline
- SNSArgomento Amazon

  EventBridge non supporta gli argomenti Amazon SNS FIFO (first in, first out).
- SQSCoda Amazon
- Macchina a stati Step Functions (ASYNC)
- Systems Manager Automation
- Systems Manager OpsItem
- Run Command di Systems Manager

Parametri di destinazione

Alcune destinazioni non inviano le informazioni del payload dell'evento alla destinazione, ma trattano l'evento come un fattore scatenante per richiamarne uno specifico. API EventBridge utilizza i parametri Target per determinare cosa succede con quell'obiettivo. Questi sono i seguenti:

- APIdestinazioni (I dati inviati a una API destinazione devono corrispondere alla struttura diAPI. È necessario utilizzare l'`InputTransformer`oggetto per assicurarsi che i dati siano strutturati correttamente. Se desideri includere il payload dell'evento originale, fai riferimento a esso in `InputTransformer`.)
- APIGateway (I dati inviati a API Gateway devono corrispondere alla struttura di. API È necessario utilizzare l'`InputTransformer`oggetto per assicurarsi che i dati siano strutturati correttamente. Se desideri includere il payload dell'evento originale, fai riferimento a esso in `InputTransformer`.)
- Amazon EC2 Image Builder
- `RedshiftDataParameters`(Cluster di dati API 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 opzionale del JSON percorso dinamico. Questa sintassi consente di specificare JSON percorsi anziché valori statici (ad esempio`$.detail.state`). L'intero valore deve essere un JSON percorso, 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 JSON percorsi dinamici dei parametri 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 API chiamate sulle risorse di tua proprietà, è EventBridge necessaria l'autorizzazione appropriata. Per AWS Lambda e per SNS le risorse Amazon, EventBridge utilizza politiche basate sulle risorse. EC2Ad esempio, i flussi di dati Kinesis e le macchine a stati Step Functions IAM utilizzano EventBridge i ruoli specificati nel parametro in`RoleARN`. `PutTargets` Puoi richiamare un endpoint API Gateway con IAM l'autorizzazione 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 eventi tra AWS account in Amazon EventBridge.

Se la destinazione è crittografata, è necessario includere la sezione seguente nella politica delle KMS chiavi.

```
{
    "Sid": "Allow EventBridge to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": "events.amazonaws.com"
    },
```

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

AWS Batch le code di lavoro come obiettivi

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**
>
> Ciò include solo jobId

- `Parameters: parameters`

CloudWatch Registra i gruppi come obiettivi

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 progetti come obiettivi

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`

ECSLe attività di Amazon come obiettivi

Se lo utilizzi InputTransformersper modellare l'evento di input su un Target in modo che corrisponda alla ECS RunTask TaskOverridestruttura di Amazon, i parametri verranno mappati 1 a 1 e passati a. `ecs.RunTask`

I piani di risposta di Incident Manager come obiettivi

Se l'evento corrispondente proviene da CloudWatch Alarms, i dettagli della modifica dello stato dell'allarme vengono inseriti nei dettagli del trigger della StartIncidentRequest chiamata a Incident Manager.

Obiettivi di Amazon API Gateway per le regole in Amazon EventBridge

Puoi usare Amazon API Gateway per creare, pubblicare, gestire e monitorareAPIs. Amazon EventBridge supporta l'invio di eventi a un endpoint API Gateway. Quando specifichi un endpoint API Gateway come destinazione, ogni evento inviato al target corrisponde a una richiesta inviata all'endpoint.

> **Important**
>
> EventBridge supporta l'utilizzo di API endpoint regionali e ottimizzati per Gateway Edge come obiettivi. 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.

È possibile utilizzare un target API Gateway per i seguenti casi d'uso:

- Per richiamare un API server ospitato in API Gateway specificato dal cliente in base AWS a eventi di terze parti.
- Per richiamare un endpoint periodicamente in base a una pianificazione.

Le informazioni sull' EventBridge JSONevento vengono inviate come corpo della HTTP richiesta 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'endpointARN, 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 HTTP intestazioni da aggiungere alla richiesta.

> **Note**
>
> Per motivi di sicurezza, le seguenti chiavi di HTTP intestazione 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 API Gateway, 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 invocazioni non riuscite. Per API Gateway, EventBridge riprova le risposte inviate con un codice di HTTP stato 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 4xx. HTTP

Timeout

EventBridge regola Le richieste API Gateway devono avere un timeout di esecuzione del client massimo di 5 secondi. Se API Gateway impiega più di 5 secondi per rispondere, calcola il EventBridge timeout della richiesta e quindi riprova.

EventBridge Le richieste Pipes API Gateway hanno un timeout massimo di 29 secondi, il massimo del API Gateway.

AWS AppSync obiettivi per le regole in Amazon EventBridge

AWS AppSync consente agli sviluppatori di connettere le proprie applicazioni e servizi a dati ed eventi con GraphQL e Pub/Sub sicuri, serverless e ad alte prestazioni. APIs 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 AWS AppSync API mutazione come obiettivo, AWS AppSync elabora l'evento tramite un'operazione di mutazione, che può quindi attivare le sottoscrizioni collegate alla mutazione.

> **Note**
>
> EventBridge supporta AWS AppSync GraphQL APIs pubblico. EventBridge attualmente non supporta AWS AppSync PrivateAPIs.

È possibile utilizzare un API target 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 a AWS AppSync GraphQL APIs utilizzando il tipo di `AWS_IAM`autorizzazione.

Per ulteriori informazioni su AWS AppSync GraphQLAPIs, 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.

- Scegliete la mutazione GraphQL da richiamare AWS AppSync API, quindi la API mutazione GraphQL.
- 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:

  - Specificate le variabili per i parametri. API Ogni variabile può essere un valore statico o un'espressione di JSON percorso dinamica verso 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 AWS AppSync API GraphQL`Ec2EventAPI`, definito 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 lo utilizzano API possono sottoscrivere l'`subscribeToEvent`abbonamento, che viene attivato dalla mutazione. `pushEvent`

È possibile creare una EventBridge regola con un target che invia eventi a AppSync API tramite la `pushEvent` mutazione. Quando viene richiamata la mutazione, qualsiasi client sottoscritto riceverà l'evento.

Per specificarlo API come obiettivo di una EventBridge regola, procedi come segue:

1. Imposta l'Amazon Resource Name (ARN) della destinazione della regola sull'endpoint ARN GraphQL di. `Ec2EventAPI` API
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 da EventBridge passare all'operazione di mutazione. AWS AppSync Il modello deve restituire a. JSON Dato il nostro percorso di input, puoi comporre il seguente modello:

```
{
  "id": <id>,
  "statusCode": <statusCode>,
  "instanceId": <EC2InstanceId>
}
```

Invio e ricezione di eventi tra AWS account in Amazon EventBridge

È possibile EventBridge configurare l'invio e la ricezione di eventi tra bus di 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 bus degli eventi di invio devono essere associati IAM dei ruoli 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.

Puoi 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 disporre di politiche che gli IAM 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 gli event bus in Amazon EventBridge.

> **Note**
>
> EventBridge ora richiede che tutti i nuovi target del bus di eventi interaccount aggiungano IAM ruoli. Ciò vale solo per le destinazioni di router di eventi create dopo il 2 marzo 2023. Le applicazioni create senza un IAM ruolo prima di tale data non sono interessate. Tuttavia, consigliamo di aggiungere IAM ruoli per concedere agli utenti l'accesso alle risorse di un altro account, in quanto ciò garantisce che vengano applicati i limiti dell'organizzazione utilizzando le politiche di controllo dei servizi (SCPs) per determinare chi può inviare e ricevere eventi dagli account dell'organizzazione.

> **Important**
>
> Se scegli di ricevere eventi da tutti gli AWS account, fai attenzione a creare regole che corrispondano solo agli eventi che puoi ricevere dagli altri. Per creare regole più sicure, assicurati che lo schema di eventi per ogni regola contenga un `Account` campo con l'account IDs di uno o più account da cui 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 Eventi in Amazon EventBridge.

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 si specifica il `account` campo nella regola, è possibile specificare l'account IDs di più di un AWS account nel campo.

Per far sì che una regola si attivi su un evento corrispondente proveniente da qualsiasi bus di eventi AWS dell'account a cui sono state concesse le autorizzazioni, non specificate * 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. Per Event bus come destinazione, inserisci il ARN bus dell'evento che desideri utilizzare.
3. Completa la creazione della regola seguendo i passaggi della procedura.

Invio e ricezione di eventi tra AWS regioni in Amazon EventBridge

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

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

Disponibilità nelle regioni

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 () UAE
- Medio Oriente (Bahrein)
- Sud America (San Paolo)

Creazione di regole per l'invio di eventi in una AWS regione diversa

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. Per Event bus come destinazione, inserisci il ARN bus dell'evento che desideri utilizzare.
3. Completa la creazione della regola seguendo i passaggi della procedura.

Invio di eventi tra bus di eventi nello stesso account e nella stessa regione in Amazon EventBridge

È possibile EventBridge configurare l'invio e la ricezione di eventi tra bus di eventi nello stesso AWS account e nella stessa regione.

Quando si configura EventBridge l'invio o la ricezione di eventi tra bus di eventi, si utilizzano IAM i ruoli sul bus degli eventi del mittente per concedere al bus degli eventi del mittente l'autorizzazione 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 gli event bus in Amazon EventBridge.

I passaggi EventBridge per configurare l'invio o la ricezione di eventi tra bus di eventi nel tuo account includono quanto segue:

- Per utilizzare un IAM ruolo esistente, è necessario assegnare le autorizzazioni del bus degli eventi del mittente al bus degli eventi del ricevitore o le autorizzazioni del bus degli eventi del ricevitore al bus degli eventi del mittente.
- Sul bus degli eventi del mittente, impostate una o più regole con il bus degli eventi del destinatario come destinazione e create 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 di eventi del 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 la pagina EventBridgedei prezzi di Amazon.

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

È possibile personalizzare il testo di un evento prima di EventBridge passare le informazioni alla destinazione di una regola. Utilizzando il trasformatore di input nella console o ilAPI, si definiscono le variabili che utilizzano il JSON percorso per fare riferimento ai valori nella fonte 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. Quindi puoi usare quelle variabili nel modello di input come <*variable-name*>.

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 è possibile utilizzare senza definire un JSON percorso. 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 ISO timestamp 8601. Questa variabile è generata da EventBridge e non può essere sovrascritta.
- `aws.events.event`— Il payload originale dell'evento as JSON (senza il `detail` campo). Può essere utilizzato solo come valore per un JSON campo, poiché il suo contenuto non viene ignorato.
- `aws.events.event.json`— Il payload completo dell'evento originale come. JSON (con il `detail` campo). Può essere usato solo come valore per un JSON campo, poiché il suo contenuto non viene escluso.

Esempi di trasformazione di input

Di seguito è riportato un esempio di EC2 evento Amazon.

```
{
  "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. Usa JSON path per fare riferimento agli elementi del tuo evento e memorizza 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 ingresso in fase di esecuzione per garantire un output valido. JSON Per questo motivo, inserite tra virgolette le variabili che fanno riferimento ai parametri del JSON percorso, ma non le virgolette attorno alle variabili che si riferiscono a JSON oggetti o matrici.

```
{
  "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. È possibile creare un modello che passi una stringa o 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 EventBridge console. AWS CLI esegue l'escape dei caratteri di barra e il risultato è `"instance "i-012345 6789" is in RUNNING"` .
Semplice JSON	`{ "instance" : <instance>, "state": <state> }`	`{ "instance" : "i-0123456789", "state": "RUNNING" }`
JSONcon 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" }`
JSONcon 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 il EventBridge API

Per informazioni sull'utilizzo dell'input EventBridge API per trasformare, consulta Use 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 è prevista alcuna convalida durante la creazione del JSON percorso per il modello.
- Se specifichi una variabile in modo che corrisponda a un JSON percorso che non esiste nell'evento, quella variabile non viene creata e non verrà visualizzata nell'output.
- JSONproprietà come `aws.events.event.json` possono essere utilizzate solo come valore di un JSON campo, 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 JSON percorso fa riferimento a un JSON oggetto o a un array, ma la variabile è referenziata in una stringa, EventBridge rimuove tutte le virgolette interne per garantire una stringa valida. Ad esempio, per una variabile `<detail>` puntata`$.detail`, «Detail is<detail>" comporterebbe la EventBridge rimozione delle virgolette dall'oggetto.

  Pertanto, se si desidera generare un JSON oggetto basato su una singola variabile di JSON percorso, è necessario 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 validoJSON. EventBridge non aggiunge virgolette alle variabili che rappresentano JSON oggetti o matrici. Non aggiungete virgolette per le variabili che rappresentano JSON oggetti o matrici.

  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>
}
```

  Risulta valido JSON con la citazione corretta:

```
{
  "ruleArn" : "arn:aws:events:us-east-2:123456789012:rule/example",
```

```
  "ruleName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

- Per l'output (nonJSON) testuale 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 ingresso durante la creazione di una regola in EventBridge

Come parte della creazione di una regola, è possibile specificare un trasformatore di input EventBridge da utilizzare per elaborare gli eventi corrispondenti 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 AWS servizi 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

Inserisci il tuo evento nel JSON testo.

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 validoJSON.
4. (Facoltativo) È anche possibile scegliere una delle seguenti opzioni:
   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Prettify: semplifica la lettura del JSON testo 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 JSON i percorsi per definire le 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 JSON il percorso per fare riferimento ai valori nella fonte 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 validoJSON.

   Ad esempio, supponiamo 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 stringheJSON, 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 per i partner o partecipare al tuo evento personalizzato.

   AWS events

   Seleziona uno degli eventi emessi dai AWS servizi 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

Inserisci il tuo evento nel JSON testo.

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 validoJSON.
4. (Facoltativo) È anche possibile scegliere una delle seguenti opzioni:
   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Prettify: semplifica la lettura del JSON testo 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 JSON i percorsi per definire le 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 JSON il percorso per fare riferimento ai valori nella fonte 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 validoJSON.

   Ad esempio, supponiamo 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 stringheJSON, 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 eventi in 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

- Creazione di un archivio per gli eventi in Amazon EventBridge
- Riproduzione di eventi Amazon archiviati EventBridge
- Aggiungere o rimuovere archivi sui bus di EventBridge eventi Amazon

Creazione di un archivio per gli eventi in Amazon EventBridge

Quando si crea un archivio in EventBridge, è possibile 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 di AWS proprietà CMK, 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

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. Per Build event pattern (Crea modello di eventi), procedi come segue:
   - Per archiviare tutti gli eventi, scegli Nessun filtro degli eventi.

- Per archiviare eventi specifici, specificate uno schema di eventi:

  a. Scegliete Filtraggio degli eventi in base alla corrispondenza del modello di evento

  b. 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'JSONeditor per creare un pattern manualmente. Puoi anche copiare il motivo da una regola e incollarlo nell'JSONeditor.

10. Scegli Crea archivio.

Per confermare che gli eventi siano stati inviati correttamente all'archivio, puoi usare l'`DescribeArchive`operazione di 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 archiviati EventBridge

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. È possibile utilizzare l'`DescribeReplay`operazione di EventBridge API per determinare l'avanzamento di un replay. `EventLastReplayedTime`restituisce il timestamp dell'ultimo evento ripetuto.

Gli eventi vengono riprodotti in base, ma separatamente, al limite di `PutEvents` transazioni al secondo per l'account. AWS Puoi richiedere un aumento del limite per PutEvents. Per ulteriori informazioni, consulta Amazon EventBridge Quotas.

> **Note**
>
> È possibile avere un massimo di 10 riproduzioni simultanee attive per account per Regione AWS .

Per avviare la riproduzione di un evento

1. Apri la EventBridge console Amazon 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. Apri la EventBridge console Amazon 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.

Aggiungere o rimuovere archivi sui bus di EventBridge eventi Amazon

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 KMS key opzioni.

Per aggiungere o rimuovere un archivio su un bus di eventi (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 (AWS CLI)

- Per creare un archivio, usa create-archive.

  Per eliminare definitivamente un archivio, usa delete-archive.

Rendere le applicazioni tolleranti ai guasti regionali con endpoint globali in EventBridge

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.

Obiettivi del tempo di ripristino e del punto di ripristino

Il Recovery Time Objective (RTO) è il tempo impiegato dalla regione secondaria per iniziare a ricevere eventi dopo un errore. InfattiRTO, l'ora include il periodo di tempo necessario per l'attivazione degli CloudWatch allarmi e l'aggiornamento degli stati per i controlli di integrità della Route 53. Il Recovery Point Objective (RPO) è la misura dei dati che rimarranno non elaborati in caso di errore. Il tempoRPO, infatti, 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 aspettarti che la RTO durata sia di 360 secondi con un massimo di 420 secondi. RPO

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"
    }
}
```

Lavorare con endpoint globali utilizzando un AWS SDK

> **Note**
>
> Il supporto per C++ sarà disponibile a breve.

Quando usi an AWS SDK per lavorare con endpoint globali, tieni presente quanto segue:

- Dovrai avere la libreria AWS Common Runtime (CRT) installata per le tue esigenze specificheSDK. Se non l'hai CRT installata, riceverai un messaggio di eccezione che indica cosa deve essere installato. Per ulteriori informazioni, consulta gli argomenti seguenti:
  - AWS Librerie 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 API operazioni che potrebbero comportare l'accesso ai dati da una delle diverse regioni. Quando si utilizza AWS SDK, l'utente fornisce le proprie credenziali e le richieste agli endpoint globali utilizzeranno la versione 4A della firma senza configurazioni aggiuntive. Per ulteriori informazioni su SigV4a, consulta Firmare AWS API le richieste nella Guida generale.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)

Creazione di un endpoint globale in Amazon EventBridge

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 API

Per creare un endpoint globale utilizzando il EventBridge API, consulta CreateEndpointAmazon EventBridge API Reference.

Per creare un endpoint globale utilizzando AWS CloudFormation

Per creare un endpoint globale utilizzando AWS CloudFormation API, consulta AWS: :Events: :Endpoints nella Guida per l'utente. AWS CloudFormation

Le migliori pratiche per gli endpoint EventBridge globali di Amazon

Le seguenti best practice sono consigliate per la configurazione degli endpoint globali.

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.

L'evento IDs può cambiare tra API le chiamate, quindi la correlazione degli eventi tra le 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.

Configurazione del controllo dello stato della Route 53 per gli endpoint EventBridge globali

Quando si utilizzano endpoint globali, è necessario effettuare un controllo dell'integrità Route 53 per monitorare lo stato delle Regioni. Il seguente modello definisce un CloudWatch allarme Amazon e lo utilizza per definire un controllo dello stato di Route 53.

Argomenti

- AWS CloudFormation modello per definire un controllo dello stato della Route 53
- CloudWatch proprietà del modello di allarme
- Proprietà del modello del controllo dell'integrità Route 53

AWS CloudFormation modello per definire un controllo dello stato della 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
```

L'evento IDs può cambiare tra API le chiamate, quindi la correlazione degli eventi tra le 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.

CloudWatch proprietà del modello di allarme

> **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 relativa agli CloudWatch allarmi del modello:

Parametro	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` . Predefinito: 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.

Parametro	Descrizione
	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. 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:

Parametro	Descrizione
`HealthCheckName`	.Il nome del controllo dell'integrità. Impostazione predefinita: **`LatencyFailuresHealthCheck`**
`InsufficientDataHealthStatus`	CloudWatch When non dispone di dati sufficienti sulla metrica per determinare lo stato di allarme, lo stato che si desidera che Amazon Route 53 assegni al controllo dello stato di salute Valori validi: • `Healthy`: Route 53 considera il controllo dello stato integro. • `Unhealthy` : Route 53 considera il controllo dello stato non integro. • `LastKnownStatus` : Route 53 utilizza lo stato del controllo di integrità dell'ultima volta che CloudWatch aveva dati sufficienti per determinare lo stato dell'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 `LastKnownStatus` . Se `TreatingMissingData` è impostato su `Breaching` , verrà aggiornato a `Unhealthy` .

Modelli di EventBridge eventi Amazon

È probabile che non vogliate elaborare ogni singolo evento che viene inviato a un determinato event bus o pipe. Piuttosto, probabilmente vorrai selezionare un sottoinsieme di tutti gli eventi forniti, in base alla fonte dell'evento, al tipo di evento e/o agli attributi di tali eventi.

Per specificare quali eventi inviare a una destinazione, create uno schema di eventi. Un modello di evento definisce i dati EventBridge utilizzati per determinare se inviare l'evento alla destinazione. Se il modello dell'evento corrisponde all'evento, EventBridge invia l'evento alla destinazione. I modelli di eventi hanno la stessa struttura degli eventi a cui corrispondono. Un modello di eventi può corrispondere o meno a un evento.

Ad esempio, considera il seguente evento di AmazonEC2:

```
{
  "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 schema di eventi seleziona tutti gli EC2 `instance-termination` eventi Amazon. Il pattern di eventi esegue questa operazione specificando tre requisiti per soddisfare un evento:

1. L'origine dell'evento deve essere AmazonEC2.
2. L'evento deve essere una notifica di EC2 modifica dello stato di Amazon.
3. Lo stato dell'EC2istanza Amazon deve essere`terminated`.

```
{
```

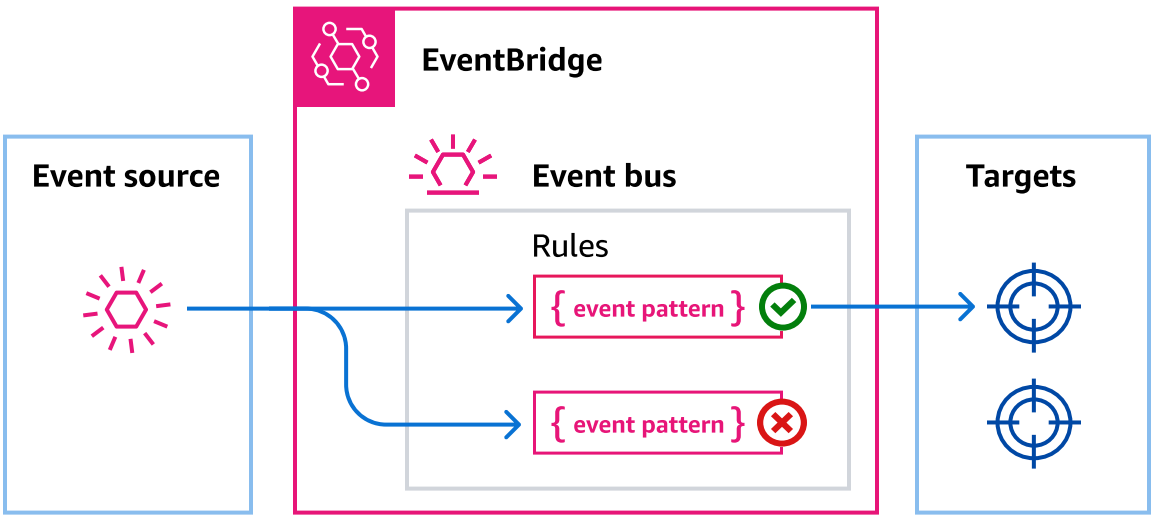
```
  "source": ["aws.ec2"],
  "detail-type": ["EC2 Instance State-change Notification"],
  "detail": {
    "state": ["terminated"]
  }
}
```

Nota che in questo esempio, il modello di evento include campi relativi all'evento-- `source` e `detail-type` --oltre a un campo del corpo dell'evento--. `state`

> **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. Supponiamo di ACLs aver creato una regola per rilevare le modifiche in un bucket Amazon S3 e di attivare un software per modificarle nello stato desiderato. Se la regola non viene scritta con cura, la successiva modifica alla regola la ACLs riattiva, creando un ciclo infinito.
> Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta Le migliori pratiche per le regole e Best practice.

Schemi di eventi per bus di eventi

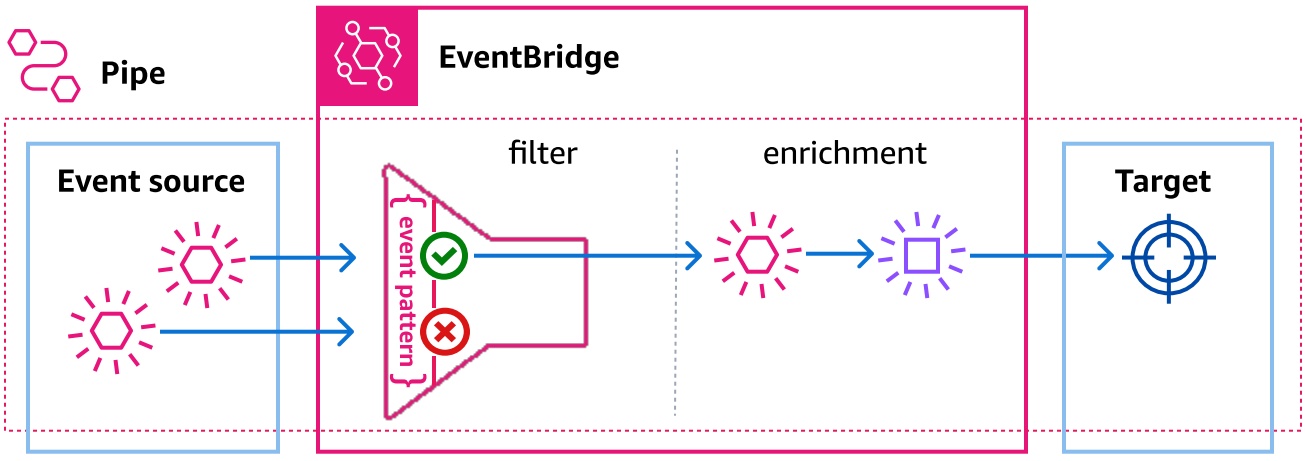
Per i bus di eventi, è possibile specificare uno schema di eventi per ogni regola creata per il bus. In questo modo, è possibile selezionare gli eventi da inviare a destinazioni specifiche. I modelli di eventi per gli event bus possono corrispondere all'origine dell'evento, ai metadati dell'evento e/o ai valori di dettaglio dell'evento.



Il video seguente illustra le nozioni di base dei modelli di eventi per i bus di eventi: Come filtrare gli eventi

Modelli di eventi per Pipes EventBridge

Per EventBridge Pipes, è possibile specificare modelli di eventi per filtrare gli eventi dalla sorgente pipe che si desidera recapitare alla destinazione del pipe. Poiché ogni pipe ha un'unica origine di eventi, i modelli di eventi per le pipe possono corrispondere ai metadati degli eventi e/o ai valori di dettaglio.



Non tutti i campi di eventi possono essere utilizzati per costruire modelli di eventi pipe. Per ulteriori informazioni, consulta Filtraggio.

Creazione di modelli di eventi in EventBridge

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.

Ad esempio, il seguente 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. EventBridgeignora tutti gli altri campi dell'evento quando applica la regola.

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

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 in modo che corrispondano agli eventi, è possibile utilizzare il `test-event-pattern` CLI comando `TestEventPattern` API o per verificare che il modello corrisponda agli eventi corretti. Per ulteriori informazioni, vedere TestEventPattern.

Valori di eventi corrispondenti

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

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

Per ulteriori informazioni, consulta Corrispondenza su più valori di campo.

Utilizzo degli operatori di confronto 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.
- Nell'evento non esiste un campo specifico. JSON

Per ulteriori informazioni, consulta Operatori di confronto.

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 a cui corrispondono i modelli di eventi seguono le JSON regole. 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 JSON campo, utilizza EventBridge solo l'ultimo.
- Tieni presente che quando EventBridge compila i modelli di eventi da utilizzare, usa il punto (.) 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.

Eventi corrispondenti ai valori dei campi degli eventi

Puoi utilizzare tutti i tipi e i valori di JSON dati per abbinare gli eventi. 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. Prendi in considerazione il seguente evento Amazon EC2 Auto Scaling.

```
{
  "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]
  }
}
```

Eventi di corrispondenza su valori nulli e stringhe vuote in Amazon EventBridge

> **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. Supponiamo di ACLs aver creato una regola per rilevare le modifiche in un bucket Amazon S3 e di attivare un software per modificarle nello stato desiderato. Se la regola non viene scritta con cura, la successiva modifica alla regola la ACLs riattiva, creando un ciclo infinito.
> Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta Le migliori pratiche per le regole e Best practice.

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`.

Corrispondenza su più valori per un campo evento in Amazon EventBridge

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. Supponiamo di ACLs aver creato una regola per rilevare le modifiche in un bucket Amazon S3 e di attivare un software per modificarle nello stato desiderato. Se la regola non viene scritta con cura, la successiva modifica alla regola la ACLs riattiva, creando un ciclo infinito.
> Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta Le migliori pratiche per le regole e Best practice.

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"
]
```

Operatori di confronto da utilizzare nei modelli di eventi in Amazon EventBridge

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	Supporto Event Bus	Supporto per tubi
And	La posizione è "New York" e il giorno è "lunedì"	`"Location": [ "New York" ], "Day": ["Monday"]`	Sì	Sì
Tutto tranne	Lo stato è qualsiasi valore oltre a «inizializzazione».	`"state": [ { "anything-but": "initializing" } ]`	Sì	Sì
Qualsiasi cosa tranne (inizia con)	La regione non è negli Stati Uniti.	`"Region": [ { "anything-but": {"prefix": "us-" } } ]`	Sì	No
Tutto tranne (finisce con)	FileName non termina con un'estensione.png.	`"FileName": [ { "anything-but": { "suffix": ".png" } } ]`	Sì	No
Tutto tranne (ignora maiuscole e minuscole)	Lo stato è qualsiasi valore oltre a «inizializzazione» o qualsiasi altra variazione di maiuscola, come "». INITIALIZING	`"state": : [{ "anything-but": { "equals-ignore-case": "initializing" }}]}`	Sì	No

Confronto	Esempio	Sintassi delle regole	Supporto Event Bus	Supporto per tubi
Tutto tranne usare un jolly	FileName non è un percorso di file che include. `/lib/`	`"FilePath" : [{ "anything-but": { "wildcard": "*/lib/*" }}]`	Sì	No
Begins with	La regione è negli Stati Uniti.	`"Region": [ {"prefix": "us-" } ]`	Sì	Sì
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" }}]}`	Sì	Sì
Empty	LastName è vuoto.	`"LastName": [""]`	Sì	Sì
Equals	Il nome è "Alice"	`"Name": [ "Alice" ]`	Sì	Sì
Equals (ignora maiuscole e minuscole)	Il nome è "Alice"	`"Name": [ { "equals-ignore-case": "alice" } ]`	Sì	Sì
Ends with	FileName termina con un'estensione.png	`"FileName": [ { "suffix": ".png" } ]`	Sì	Sì
Termina con (ignora maiuscole)	Il nome del servizio termina con le lettere «tbridge» o con qualsiasi altra variante della maiuscola, ad esempio "». TBRIDGE	`{"service" : [{ "suffix": { "equals-ignore-case": "tBridge" }}]}`	Sì	Sì

Confronto	Esempio	Sintassi delle regole	Supporto Event Bus	Supporto per tubi
Exists	ProductName esiste	`"ProductName": [ { "exists": true } ]`	Sì	Sì
Does not exist	ProductName non esiste	`"ProductName": [ { "exists": false } ]`	Sì	Sì
Not	Il tempo è qualsiasi tranne "piovoso"	`"Weather": [ { "anything-but": [ "Raining" ] } ]`	Sì	Sì
Null	UserID è nullo	`"UserID": [ null ]`	Sì	Sì
Numeric (uguale)	Il prezzo è 100	`"Price": [ { "numeric": [ "=", 100 ] } ]`	Sì	Sì
Numeric (intervallo)	Il prezzo è superiore a 10 e inferiore o uguale a 20	`"Price": [ { "numeric": [ ">", 10, "<=", 20 ] } ]`	Sì	Sì
Or	PaymentType è «Credito» o «Debito»	`"PaymentType": [ "Credit", "Debit"]`	Sì	Sì
Or (campi multipli)	La posizione è "New York" o il giorno è "lunedì".	`"$or": [ { "Location": [ "New York" ] }, { "Day": [ "Monday" ] } ]`	Sì	Sì

Confronto	Esempio	Sintassi delle regole	Supporto Event Bus	Supporto per tubi
Carattere jolly	Qualsiasi file con estensione .png, situato nella cartella "dir"	`"FileName": [ { "wildcard": "dir/*.png" } ]`	Sì	No

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 tranne che la corrispondenza 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" }}]}
}
```

È possibile utilizzare anche `equals-ignore-case` in combinazione con `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 JSON numeri. 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

Puoi utilizzare la corrispondenza degli indirizzi IP per IPv6 gli indirizzi IPv4 e. 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

Exists Matching funziona sulla presenza o l'assenza JSON di un campo nell'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 evento precedente NOT corrisponde al seguente evento perché non ha un `detail.state` campo.

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

Corrispondenza E quals-ignore-case

La quals-ignore-case corrispondenza E funziona sui valori delle stringhe 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 del programma si ARN baseranno direttamente sulla regione in cui risiedono gli autobus dell'evento. 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 criteri di corrispondenza in uno schema di eventi 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**
>
> APIsche accettano un pattern di eventi (comePutRule, `CreateArchiveUpdateArchive`, `eTestEventPattern`) genereranno un `InvalidEventPatternException` if l'utilizzo di `$or` risultati in oltre 1000 combinazioni di regole.
> Per determinare il numero di combinazioni di regole in un modello di eventi, moltiplica il numero totale di argomenti di ogni array `$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 dei modelli di eventi utilizzando la EventBridge Sandbox

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 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. Supponiamo di ACLs aver creato una regola per rilevare le modifiche in un bucket Amazon S3 e di attivare un software per modificarle nello stato desiderato. Se la regola non viene scritta con cura, la successiva modifica alla regola la ACLs riattiva, creando un ciclo infinito.

Per indicazioni su come scrivere regole e modelli di eventi precisi per ridurre al minimo tali risultati imprevisti, consulta Le migliori pratiche per le regole e Best practice.

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 AWS servizi Supported.
   - EventBridge eventi partner: seleziona tra gli eventi emessi da servizi di terze parti che supportano EventBridge, come Salesforce.
   - Inserisci il mio: inserisci il tuo evento nel 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. Eliminate la struttura degli eventi di esempio nel riquadro di JSON modifica e incollate l'evento AWS o l'evento partner al suo posto.
     6. Modifica l'evento JSON 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 EventBridge uno 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. Scegliete Genera modello di evento in JSON per generare e convalidare il modello di evento come JSON testo.
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 dell'evento.

   È anche possibile scegliere una delle seguenti opzioni:

   - Copia: copia il modello di eventi negli appunti del dispositivo.

- Prettify: semplifica la lettura del JSON testo 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. È necessario formattare lo schema come testo valido. JSON
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 modello di evento in JSON per generare e convalidare il modello di evento come JSON testo.
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.
   - Prettify: semplifica la lettura del JSON testo aggiungendo interruzioni di riga, tabulazioni e spazi.

Event pattern

Per scrivere un modello di evento personalizzato in JSON formato, effettuate le seguenti operazioni:

1. Nella sezione Metodo di creazione, per Metodo, scegliete Modello personalizzato (JSONeditor).
2. Per Schema di evento, inserisci il modello di evento personalizzato in un JSON testo formattato.
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 di evento.

   È anche possibile scegliere una delle seguenti opzioni:

   - Copia: copia il modello di eventi negli appunti del dispositivo.
   - Prettify: semplifica la lettura del JSON testo 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 che Pattern Builder venga aperto.

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 i modelli di EventBridge eventi di 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 cicli infiniti, in cui una regola viene attivata ripetutamente. Ad esempio, una regola potrebbe rilevare ACLs le modifiche in un bucket S3 e attivare il software per modificarle nello stato desiderato. Se la regola non viene scritta con cura, la successiva modifica alla regola la ACLs riattiva, creando un ciclo 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, create uno schema di eventi nella regola che corrisponda agli eventi solo se ACLs risultano in cattivo stato, 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 modelli 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 modelli degli eventi utilizzando la EventBridge console o: 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 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 ???.

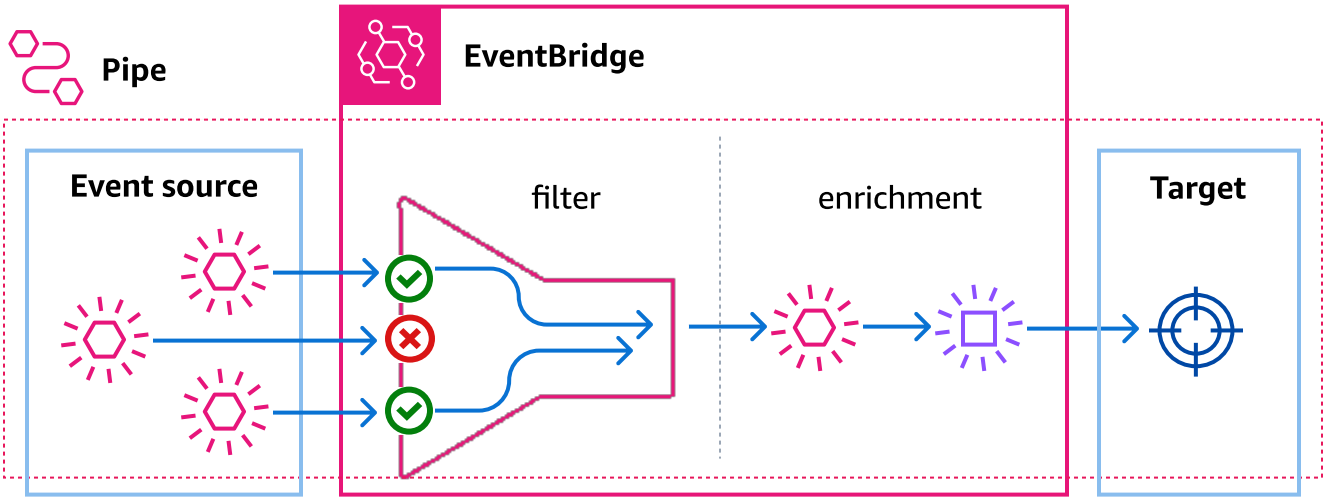
EventBridge Come funzionano i tubi

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 file API che contiene informazioni sui clienti, come gli indirizzi di spedizione.

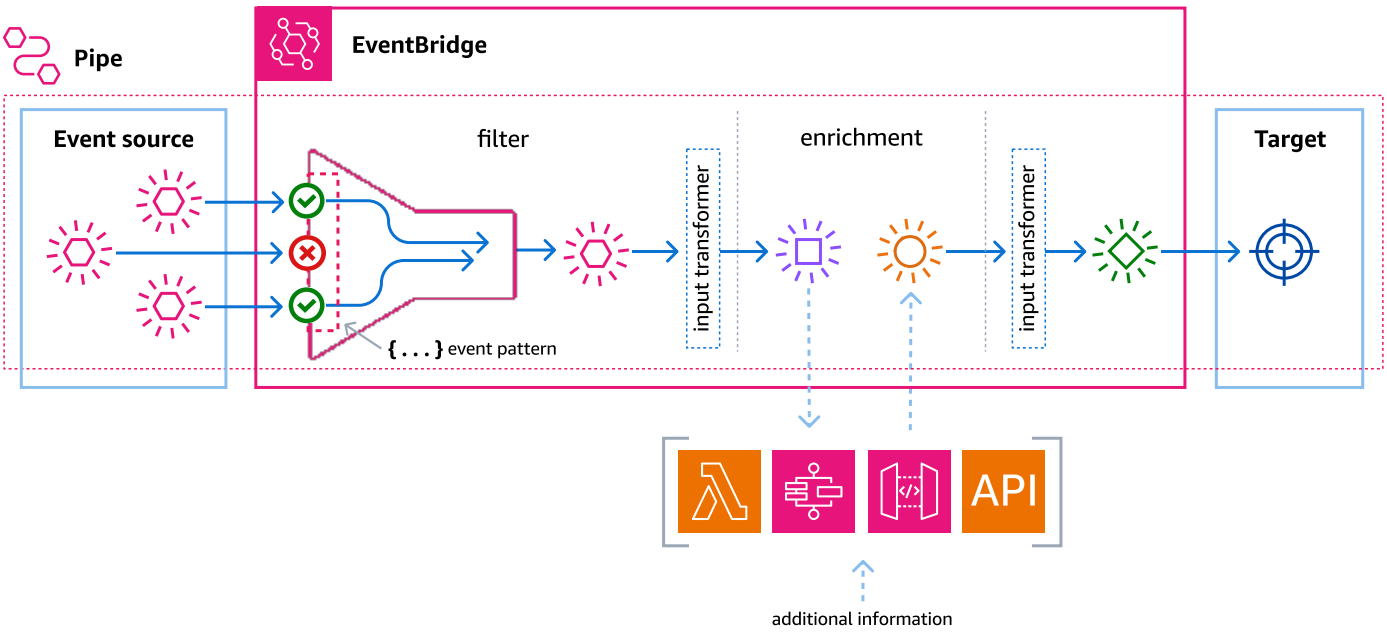
1. A questo proposito, crei una pipe con:
   - Un SQS ordine Amazon ha ricevuto una coda di messaggi come origine dell'evento.
   - Una EventBridge API destinazione come arricchimento
   - Una macchina a AWS Step Functions stati come obiettivo
2. Quindi, quando un messaggio di ricezione di un SQS ordine Amazon appare in coda, viene inviato alla tua pipe.
3. La pipe invia quindi i dati al servizio 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.

Concetti di Amazon EventBridge Pipes

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. Utilizzando l'arricchimento, puoi fare in modo che una funzione Lambda chiami `get-ticket` API il per i dettagli completi del 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 all'origine degli eventi per Amazon EventBridge Pipes

Quando configuri una pipe, puoi utilizzare un ruolo di esecuzione esistente o EventBridge crearne uno per te 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 alla fonte, usa la console EventBridge Pipes per creare un nuovo ruolo, quindi esamina le azioni elencate nella politica.

Argomenti

Autorizzazioni del ruolo di esecuzione DynamoDB

Per DynamoDB Streams 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 tuo broker di messaggi Amazon MQ.

- mq:DescribeBroker
- secretsmanager:GetSecretValue
- ec2:CreateNetworkInterface
- ec2:DeleteNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- logs:CreateLogGroup
- logs:CreateLogStream

- logs:PutLogEvents

Autorizzazioni per i ruoli di MSK esecuzione di Amazon

Per AmazonMSK, EventBridge richiede le seguenti autorizzazioni per gestire le risorse correlate al tuo MSK argomento Amazon.

> **Note**
>
> Se utilizzi l'autenticazione IAM basata sui ruoli, il tuo ruolo di esecuzione avrà bisogno delle autorizzazioni elencate 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, sono EventBridge necessarie le seguenti autorizzazioni per gestire le risorse correlate allo stream Apache Kafka autogestito.

Autorizzazioni richieste

Per creare e archiviare i log in un gruppo di log in Amazon CloudWatch Logs, la tua pipe deve avere le seguenti autorizzazioni nel suo ruolo di esecuzione:

- logs:CreateLogGroup
- logs:CreateLogStream
- logs:PutLogEvents

Autorizzazioni facoltative

La pipe potrebbe anche richiedere autorizzazioni per:

- Descrivere il segreto di Secrets Manager.
- Accedi alla tua chiave AWS Key Management Service (AWS KMS) gestita dal cliente.
- Accedi al tuo AmazonVPC.

Secrets Manager e AWS KMS autorizzazioni

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

VPCautorizzazioni

Se solo gli utenti all'interno di un cluster Apache Kafka VPC possono accedere al tuo cluster Apache Kafka autogestito, la tua pipe deve avere l'autorizzazione per accedere alle tue risorse Amazon. VPC Queste risorse includono le tue sottoretiVPC, 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 per i ruoli di SQS esecuzione di Amazon

Per AmazonSQS, EventBridge richiede le seguenti autorizzazioni per gestire le risorse correlate alla tua SQS coda Amazon.

- `sqs:ReceiveMessage`
- `sqs:DeleteMessage`
- `sqs:GetQueueAttributes`

Autorizzazioni di arricchimento e destinazione

Per effettuare API chiamate sulle risorse di tua proprietà, EventBridge Pipes necessita dell'autorizzazione appropriata. EventBridge Pipes utilizza il IAM ruolo specificato sulla pipe per le chiamate di arricchimento e di destinazione utilizzando il IAM principale`pipes.amazonaws.com`.

Creare una EventBridge pipa Amazon

EventBridge Pipes consente di creare point-to-point integrazioni tra sorgenti 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 il AWS CLI, vedete create-pipe nel AWS CLICommand Reference.

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) Per VPC, scegli quello VPC che desideri. Quindi, per le VPCsottoreti, scegli le sottoreti desiderate. Per i gruppi VPC di sicurezza, scegli i gruppi di sicurezza.
   5. Per l'autenticazione, facoltativa, attiva Usa autenticazione ed esegui le seguenti operazioni:
      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 vengono elaborati o non rimane un messaggio fallito nel batch.

> **Note**
>
> Se non si sceglie AUTOMATIC_ BISECT, è possibile restituire specifici record non riusciti e solo quelli vengono riprovati.

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, inserisci una data e un'ora utilizzando i formati YYYY /MM/DD 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 vengono elaborati o non rimane un messaggio fallito nel batch.

> **Note**
>
> Se non si sceglie AUTOMATIC_ BISECT, è possibile restituire specifici record non riusciti e solo quelli vengono riprovati.

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. Per 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 una MSK fonte Amazon utilizzando la console

1. Per Source, scegli Amazon MSK.
2. Per il MSKcluster Amazon, scegli il cluster che desideri 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) Per VPC, scegli VPC quello che desideri. Quindi, per le VPCsottoreti, scegli le sottoreti desiderate. Per i gruppi VPC di sicurezza, 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 una SQS fonte Amazon utilizzando la console

1. Per Source (Fonte), scegliere SQS.
2. Per SQScoda, scegli la coda che desideri 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 , API Amazon 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 morte (). 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 di lettere non scritte (DLQ), attiva la coda di lettere non scritte, scegli il metodo che preferisci e scegli la coda o l'argomento che desideri utilizzare. Per impostazione predefinita, le pipe appena create non utilizzano un. 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 ERROR registro. 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 quello ARN di un gruppo di log esistente.
   - Per i nuovi gruppi di log, modifica il nome del gruppo di log come desiderato.

   CloudWatch i registri 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:

- IAMrole — Poiché l'origine di una pipe non può essere modificata dopo la creazione della pipe, EventBridge verifica che il IAM ruolo fornito possa accedere all'origine.

> **Note**
>
> EventBridge non esegue la stessa convalida per gli arricchimenti o le destinazioni 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 gli arricchimenti API Gateway e di API destinazione sia 1, poiché sono supportate solo le dimensioni dei batch pari a 1.

Avvio o arresto di una pipa Amazon EventBridge

Per impostazione predefinita, una pipe è `Running` ed elabora eventi al momento della creazione.

Se crei una pipe con sorgenti AmazonSQS, Kinesis o DynamoDB, la creazione di pipe può richiedere in genere uno o due minuti.

Se crei una pipe con AmazonMSK, fonti autogestite Apache Kafka o Amazon MQ, la creazione delle pipe 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 API chiamata, imposta su. `DesiredState` `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 API chiamata, imposta il `DesiredState` parametro 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 Amazon SQS e le sorgenti di streaming, 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
- MSKStreaming Amazon
- SQSCoda Amazon
- 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

Stream Amazon DynamoDB come sorgente per Pipes EventBridge

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 per Amazon DynamoDB Streams che puoi scegliere durante la configurazione della pipe. EventBridge Pipes mantiene l'ordine dei record dal flusso di dati quando invia tali 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 riattivi un flusso DynamoDB, a tale flusso viene assegnato un nuovo Amazon Resource Name ARN () e non è più associato alla tua pipe.
>
> Se riattivi il flusso DynamoDB, dovrai creare una nuova pipe usando il nuovo stream. ARN

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 analizza i frammenti del tuo flusso DynamoDB alla ricerca di record a una frequenza base di quattro volte al secondo. Quando i record sono disponibili, EventBridge elabora l'evento e attende il risultato. Se l'elaborazione ha esito positivo, EventBridge riprende il polling finché non riceve altri record.

Per impostazione predefinita, EventBridge richiama la pipe non appena i record sono disponibili. Se il batch che EventBridge legge dalla sorgente contiene un solo record, viene elaborato un solo 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 elaborare gli eventi, EventBridge continua a leggere i record dall'origine fino alla raccolta di un batch completo, alla scadenza della finestra di batch o al raggiungimento del limite di payload di 6 MB.

È possibile anche aumentare la concorrenza elaborando più batch da ogni partizione in parallelo. EventBridge può elaborare fino a 10 batch in ogni shard contemporaneamente. Se si aumenta il numero di batch simultanei per shard, si garantisce EventBridge comunque l'elaborazione in ordine 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 che eseguono il EventBridge polling da uno shard tramite un fattore di parallelizzazione compreso tra 1 (impostazione predefinita) e 10. Ad esempio, se si imposta su `ParallelizationFactor` 2, è possibile avere un massimo di 200 esecuzioni EventBridge Pipe simultanee per elaborare 100 frammenti di dati Kinesis. Ciò permette di dimensionare verso l'alto il throughput di elaborazione quando il volume dei dati è volatile e l'`IteratorAge` è alta. 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 in streaming da una fonte, per impostazione predefinita il checkpoint si basa sul 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.

Amazon Kinesis Stream come sorgente per Pipes EventBridge

È possibile 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 quando invia i 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 un EventBridge Pipe a un consumatore con throughput condiviso (iteratore standard) o a un consumatore con throughput dedicato con fan-out avanzato.

Per gli iteratori standard, EventBridge utilizza il HTTP protocollo per eseguire il polling di ogni shard nel flusso Kinesis alla ricerca di record. 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 spinge i record a oltre /2. EventBridge HTTP 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 analizza i frammenti del tuo stream Kinesis alla ricerca di record a una frequenza base di quattro volte al secondo. Quando i record sono disponibili, EventBridge elabora l'evento e attende il risultato. Se l'elaborazione ha esito positivo, EventBridge riprende il polling finché non riceve altri record.

Per impostazione predefinita, EventBridge richiama la pipe non appena i record sono disponibili. Se il batch che EventBridge legge dalla sorgente contiene un solo record, viene elaborato un solo 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 elaborare gli eventi, EventBridge continua a leggere i record dall'origine fino alla raccolta di un batch completo, alla scadenza della finestra di batch o al raggiungimento del limite di payload di 6 MB.

È possibile anche aumentare la concorrenza elaborando più batch da ogni partizione in parallelo. EventBridge può elaborare fino a 10 batch in ogni shard contemporaneamente. Se si aumenta il numero di batch simultanei per shard, si garantisce EventBridge comunque l'elaborazione in ordine 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 che eseguono il EventBridge polling da uno shard tramite un fattore di parallelizzazione compreso tra 1 (impostazione predefinita) e 10. Ad esempio, se si imposta su `ParallelizationFactor` 2, è possibile avere un massimo di 200 esecuzioni EventBridge Pipe simultanee per elaborare 100 frammenti di dati Kinesis. Ciò permette di dimensionare verso l'alto il throughput di elaborazione quando il volume dei dati è volatile e l'`IteratorAge` è alta. 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 in streaming da una fonte, per impostazione predefinita il checkpoint si basa sul 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.

Broker di messaggi Amazon MQ come fonte in EventBridge Pipes

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 le istanze Amazon Elastic Compute Cloud (AmazonEC2) per tuo conto installando 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: non supporta l'elaborazione su più account. EventBridge 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 ActiveMQ, consulta Integrating ActiveMQ brokers with nella Amazon MQ Developer LDAP Guide.
- 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 pubblico o privato (). VPC Per quanto riguarda la modalità privataVPCs, la tua pipe deve accedere VPC a per 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 0-9-1 per consumare i messaggiAMQP. Non sono supportati altri protocolli per l'utilizzo dei messaggi. Per ulteriori informazioni sull'implementazione del protocollo 0-9-1 da parte di RabbitMQ, vedere AMQP la Guida di riferimento completa 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 consumatori viene creato con lo stesso ID della pipeUUID.

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 finché 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. Invece, il gruppo di consumatori li recupera sotto forma di byte. BLOB 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 un accesso completo con la pipe, il broker deve utilizzare un endpoint pubblico o fornire l'accesso a. VPC

Se il tuo broker Amazon MQ non è accessibile al pubblico, EventBridge deve avere accesso alle risorse Amazon Virtual Private Cloud (AmazonVPC) associate al tuo broker.

- Per accedere ai VPC tuoi broker Amazon MQ, EventBridge puoi utilizzare l'accesso a Internet in uscita per le sottoreti della tua fonte. Per le sottoreti pubbliche questo deve essere un gateway gestito. NAT Per le sottoreti private può essere un NAT gateway o il proprio. NAT Assicurati che NAT abbia un indirizzo IP pubblico e che possa connettersi a Internet.
- EventBridge Pipes supporta anche l'invio di eventi tramite AWS PrivateLink, che consente di inviare eventi da un'origine 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, consulta Creare un VPC endpoint nella Guida per l'utente. VPC AWS PrivateLink Per il nome del servizio, seleziona. `com.amazonaws.region.pipes-data`

Configura i tuoi gruppi VPC di sicurezza Amazon con le seguenti regole (come 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 in chiaro
  - 9094 per TLS
  - 9096 per SASL
  - 9098 per IAM

> **Note**
>
> La tua VPC configurazione Amazon è rilevabile tramite Amazon MQ API. Non è necessario configurarlo durante l'installazione.

Argomento Amazon Managed Streaming per Apache Kafka come fonte in Pipes EventBridge

Puoi usare 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 puoi utilizzare per creare ed eseguire applicazioni che utilizzano Apache Kafka per elaborare dati in streaming. Amazon MSK semplifica la configurazione, il ridimensionamento e la gestione dei cluster che eseguono Apache Kafka. Con AmazonMSK, puoi configurare la tua applicazione per più zone di disponibilità e per la sicurezza con AWS Identity and Access Management (IAM). Amazon MSK supporta diverse versioni open source di Kafka.

Amazon MSK come fonte funziona in modo simile all'utilizzo di Amazon Simple Queue Service (AmazonSQS) 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'MSKargomento e dell'identificatore di partizione di Amazon, insieme a un timestamp e 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`.

MSKautenticazione del cluster

EventBridge necessita dell'autorizzazione per accedere al MSK cluster Amazon, recuperare i record ed eseguire altre attività. Amazon MSK supporta diverse opzioni per controllare l'accesso dei client al MSK cluster. Per ulteriori informazioni su quale metodo di autenticazione viene utilizzato, consulta ???.

Opzioni di accesso al cluster

- Accesso non autenticato
- SASL/autenticazione SCRAM

- Autenticazione basata su ruoli IAM
- Autenticazione reciproca TLS
- Configurazione di m secret TLS
- Come EventBridge sceglie un broker bootstrap

Accesso non autenticato

Consigliamo di utilizzare solo accessi non autenticati per lo sviluppo. L'accesso non autenticato funzionerà solo se l'autenticazione IAM basata sui ruoli è disabilitata per il cluster.

SASL/autenticazione SCRAM

Amazon MSK supporta l'autenticazione Simple Authentication and Security Layer/Salted Challenge Response Authentication Mechanism (SASL/SCRAM) 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 PLAIN l'autenticazione SASL /.

Autenticazione basata su ruoli IAM

Puoi utilizzarla IAM per autenticare l'identità dei client che si connettono al MSK cluster. Se IAM l'autenticazione è attiva nel MSK cluster e non si fornisce un segreto per l'autenticazione, per impostazione predefinita utilizza EventBridge automaticamente l'autenticazione. IAM Per creare e distribuire policy basate sugli IAM utenti o sui ruoli, utilizza la console o. IAM API Per ulteriori informazioni, consulta il controllo degli IAM accessi nella Amazon Managed Streaming for Apache Kafka Developer Guide.

Per consentire la connessione EventBridge al MSK cluster, 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 azioni di Amazon MSK Kafka nella Amazon Managed Streaming for Apache Kafka Developer Guide.

Autenticazione reciproca TLS

Mutual TLS (mTLS) 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 AmazonMSK, EventBridge funge da cliente. Si configura 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 MSK cluster 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, perché tutti i broker di Amazon MSK utilizzano certificati pubblici firmati da Amazon Trust ServicesCAs, che per impostazione predefinita si EventBridge fida.

Per ulteriori informazioni su m TLS for AmazonMSK, consulta Mutual TLS Authentication nella Amazon Managed Streaming for Apache Kafka Developer Guide.

Configurazione di m secret TLS

Il AUTH segreto CLIENT CERTIFICATE _ TLS _ _ richiede un campo certificato e un campo chiave privata. Per una chiave privata crittografata, il segreto richiede una password per chiave privata. Sia il certificato che la chiave privata devono essere in PEM formato.

> **Note**
>
> EventBridge supporta gli algoritmi di crittografia a chiave privata PBES1(ma nonPBES2).

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 seguente struttura:

```
-----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-----
```

L'esempio seguente mostra il contenuto di un segreto per la mia TLS autenticazione 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 EventBridge sceglie un broker bootstrap

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 m TLS oSASL/SCRAM, sceglie EventBridge automaticamente quel metodo di autenticazione. Se non fornisci un segreto, EventBridge sceglie il metodo di autenticazione più efficace attivo nel 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:

- m TLS (segreto fornito per m) TLS
- SASL/SCRAM(segreto fornito perSASL/SCRAM)
- SASLIAM(nessun segreto fornito e IAM l'autenticazione è attiva)
- Non autenticato TLS (nessun segreto fornito e l'IAMautenticazione non è attiva)
- Testo semplice (nessun segreto fornito e sia IAM l'autenticazione che quella non autenticata non TLS sono attive)

> **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 (AmazonVPC) associate al tuo MSK cluster Amazon.

- Per accedere al VPC tuo MSK cluster Amazon, EventBridge puoi utilizzare l'accesso a Internet in uscita per le sottoreti della tua fonte. Per le sottoreti private può essere un NAT gateway o il tuo. NAT Assicurati che NAT abbia un indirizzo IP pubblico e che possa connettersi a Internet. Per le sottoreti pubbliche è necessario utilizzare VPC Endpoint (spiegato di seguito).
- EventBridge Pipes supporta anche l'invio 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. Puoi anche utilizzare VPC Endpoints per supportare la consegna dai cluster Kafka nelle sottoreti pubbliche.

  Per configurare un VPC endpoint, consulta Creare un endpoint nella Guida per l'utente. VPC AWS PrivateLink Per il nome del servizio, seleziona. `com.amazonaws.region.pipes-data`

Configura i tuoi gruppi VPC di sicurezza Amazon con le seguenti regole (come minimo):

- Regole in entrata: consenti tutto il traffico sulla porta del MSK broker Amazon 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 MSK broker Amazon per i gruppi di sicurezza specificati per la tua fonte.

  Le porte del broker includono:

  - 9092 per testo non crittografato

- 9094 per TLS
- 9096 per SASL
- 9098 per IAM

> **Note**
>
> La tua VPC configurazione Amazon è rilevabile tramite Amazon MSK API. 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. Puoi 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.

Scalabilità automatica del codice sorgente Amazon MSK

Quando crei inizialmente un MSK sorgente Amazon, 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 in Pipes EventBridge

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 (MSKAmazon) 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) o SQS 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.

VPCaccesso

Se utilizzi un ambiente Apache Kafka autogestito in cui solo gli utenti di Apache Kafka al tuo interno VPC hanno accesso ai tuoi broker Apache Kafka, devi configurare Amazon Virtual Private Cloud (Amazon) nel sorgente Apache Kafka. VPC

SASL/autenticazione SCRAM

EventBridge Pipes supporta l'autenticazione Simple Authentication and Security Layer/Salted Challenge Response Authentication Mechanism (SASL/SCRAM) con crittografia Transport Layer Security (TLS). EventBridge Pipes invia le credenziali crittografate per l'autenticazione con il cluster. Per ulteriori informazioni suSASL/SCRAMauthentication, vedere RFC 5802.

EventBridge Pipes supporta PLAIN l'autenticazioneSASL/con TLS crittografia. ConSASL/PLAINauthentication, EventBridge Pipes invia le credenziali come testo non crittografato al server.

Per SASL l'autenticazione, si memorizzano le credenziali di accesso come accesso segreto. AWS Secrets Manager

Autenticazione reciproca TLS

Mutual TLS (mTLS) 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, EventBridge Pipes funge da client. 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 Introducing mutual TLS authentication for Amazon MSK as an source.

Configurazione del segreto del certificato client

Il AUTH segreto CLIENT CERTIFICATE _ TLS _ _ richiede un campo certificato e un campo chiave privata. Per una chiave privata crittografata, il segreto richiede una password per chiave privata. Sia il certificato che la chiave privata devono essere in PEM formato.

> **Note**
>
> EventBridge Pipes supporta PBES1(ma nonPBES2) gli algoritmi di crittografia a chiave privata.

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 seguente struttura:

```
-----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-----
```

L'esempio seguente mostra il contenuto di un segreto per la mia TLS autenticazione 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 TLS crittografia con certificati firmati da una CA privata. È possibile utilizzare TLS la crittografia per l'autenticazioneVPC,SASL/SCRAMPLAIN,SASL/o m. TLS

Il segreto del certificato CA principale del server richiede un campo che contenga il certificato CA principale del broker Apache Kafka in formato. PEM Il seguente esempio illustra la struttura del segreto.

```
{
     "certificate": "-----BEGIN CERTIFICATE-----
```

```
MIID7zCCAtegAwIBAgIBADANBgkqhkiG9w0BAQsFADCBmDELMAkGA1UEBhMCVVMx
EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxJTAjBgNVBAoT
HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xOzA5BgNVBAMTMlN0YXJmaWVs
ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dG...
-----END CERTIFICATE-----"
```

Configurazione della rete

Se utilizzi un ambiente Apache Kafka autogestito che utilizza la VPC connettività privata, EventBridge devi avere accesso alle risorse Amazon Virtual Private Cloud (AmazonVPC) associate ai tuoi broker Apache Kafka.

- Per accedere al VPC tuo cluster Apache Kafka, EventBridge puoi utilizzare l'accesso a Internet in uscita per le sottoreti della tua fonte. Per le sottoreti private può essere un gateway o il tuo. NAT NAT Assicurati che NAT abbia un indirizzo IP pubblico e che possa connettersi a Internet. Per le sottoreti pubbliche è necessario utilizzare VPC Endpoint (spiegato di seguito).
- EventBridge Pipes supporta anche l'invio 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. Puoi anche utilizzare VPC Endpoints per supportare la consegna dai cluster Kafka nelle sottoreti pubbliche.

  Per configurare un VPC endpoint, consulta Creare un endpoint nella Guida per l'utente. VPC AWS PrivateLink Per il nome del servizio, seleziona. `com.amazonaws.region.pipes-data`

Configura i tuoi gruppi VPC di sicurezza Amazon con le seguenti regole (come minimo):

- Regole in entrata: consenti tutto il traffico sulla porta del broker Apache Kafka per i gruppi di sicurezza specificati per la tua fonte.
- 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 sorgente in EventBridge Pipes

Puoi usare EventBridge Pipes per ricevere record da una SQS coda Amazon. 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 (AmazonSQS). EventBridge Le pipe supportano le code standard e le code first-in, first-out (). FIFO Con AmazonSQS, puoi scaricare le attività da un componente della tua 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 , interroga fino a 10 messaggi nella coda contemporaneamente 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 esaminare i messaggi dalla coda SQS standard di Amazon finché 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 SQS coda Amazon 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 FIFO code, i record contengono attributi aggiuntivi correlati alla deduplicazione e al sequenziamento.

Quando EventBridge legge un batch, i messaggi rimangono in coda ma sono 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"
  }
]
```

FIFOcoda

```
[
  {
    "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 FIFO code, EventBridge invia i messaggi alla tua pipe nell'ordine in cui li riceve. Quando si invia un messaggio a una FIFO coda, si specifica un ID del gruppo di messaggi. Amazon SQS semplifica l'invio 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 SQS coda Amazon che funga da fonte per la tua 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 tua pipe non riesce a elaborare un messaggio più volte, Amazon SQS può inviarlo a una coda di lettere morte. Quando la tua pipe restituisce un errore, la EventBridge tiene 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 si basa sul 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 degli eventi in 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`). A `Pattern` è una rappresentazione in formato 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 maggiore chiarezza, ecco il valore del filtro `Pattern` espanso in chiaroJSON:

```
{
  "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 è in formato JSON -escape, come il SQS `body` campo Amazon, o con codifica base64, come il campo Kinesis. `data` Se i dati sono validiJSON, i modelli di input o i JSON percorsi per i parametri di destinazione possono fare riferimento direttamente al contenuto. Ad esempio, se un'origine di eventi Kinesis è validaJSON, puoi fare riferimento a una variabile utilizzando. < `$.data.someKey`>

Quando si creano modelli di eventi, è possibile filtrare in base ai campi inviati dalla fonte API 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. Questi campi sono speciali perché possono essere JSON codificati in -escape o in base64, ma quando sono validi JSON possono

essere filtrati con JSON schemi come se il corpo non fosse sfuggito. Il contenuto di questi campi può essere utilizzato senza problemi in trasformatori di input.

Filtraggio corretto dei messaggi Amazon SQS

Se un SQS messaggio Amazon non soddisfa i tuoi criteri di filtro, rimuove EventBridge automaticamente il messaggio dalla coda. Non è necessario eliminare questi messaggi manualmente in AmazonSQS.

Per AmazonSQS, il messaggio `body` può essere qualsiasi stringa. Tuttavia, questo può essere problematico se `FilterCriteria` si prevede che sia `body` in un formato validoJSON. Vale anche lo scenario inverso: se il messaggio in arrivo `body` è in un JSON formato valido, ma i criteri di filtro `body` prevedono che sia una semplice stringa, si verifica un comportamento non intenzionale.

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 `body` in arrivo 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 tuoi criteri di filtro.
Stringa normale	Valido JSON	EventBridge rilascia il messaggio.
Valido JSON	Stringa normale	EventBridge rilascia il messaggio.
Valido JSON	Nessun modello di filtro per le proprietà dei dati	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.

Formato **body** messaggio in arrivo	Formato **body** modello di filtro	Operazione risultante
Valido JSON	Valido JSON	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 sorgenti di eventi di streaming, sia il campo dati che i criteri di filtro per il campo dati devono essere in formato valido. JSON (per Kinesis, il campo dati è `data`, per Dynamo DB, il campo dati è `dynamodb`). Se uno dei due campi non è in un JSON formato 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
Valido JSON	Valido JSON	EventBridge filtri in base ai tuoi criteri di filtro.
Valido JSON	Nessun modello di filtro per le proprietà dei dati	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
Valido JSON	Non- JSON	EventBridge genera un'eccezione al momento della pipe o dell'aggiornamento. Il modello di filtro per le proprietà dei

Formato dei dati in entrata (**`data`** o **`dynamodb`**)	Formato del modello di filtro per le proprietà di dati	Operazione risultante
		dati deve essere in un JSON formato valido.
Non- JSON	Valido JSON	EventBridge cancella il record.
non- JSON	Nessun modello di filtro per le proprietà dei dati	EventBridge filtri (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'aggi ornamento della pipe. Il modello di filtro per le proprietà dei dati deve essere in un JSON formato 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 i sorgenti Apache Kafka (Amazon MSK e Apache Kafka autogestito), ci sono due campi di messaggio: e. `key` `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 con successo la funzione e riconosce i messaggi non corrispondenti quando li filtra.

I messaggi di Apache Kafka e Amazon MQ devono essere costituiti da UTF -8 stringhe codificate, 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, ad esempio UTF -16 o, o se il formato del messaggio non corrisponde al formatoASCII, elabora solo i filtri dei `FilterCriteria` metadati. 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 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 tuoi criteri di filtro.
Stringa normale	Valido JSON	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
Valido JSON	Stringa normale	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
Valido JSON	Nessun modello di filtro per le proprietà dei dati	EventBridge filtri (solo sulle altre proprietà dei metadati) in base ai criteri di filtro.
Valido JSON	Valido JSON	EventBridge filtri in base ai tuoi criteri di filtro.
Stringa codificata non 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 si filtrano gli eventi, ESM Lambda EventBridge e Pipes funzionano generalmente allo stesso modo. La differenza principale è che il `eventSourceKey` campo non è presente nei ESM payload.

Utilizzo di operatori di confronto nei filtri per tubi

Gli operatori di confronto consentono di creare modelli di eventi che corrispondono ai valori dei campi negli eventi.

Per un elenco completo degli operatori di confronto supportati per l'uso nei filtri a tubo, vedereOperatori di confronto.

Arricchimento degli eventi in Amazon EventBridge Pipes

Con la fase di arricchimento di EventBridge Pipes, puoi migliorare i dati dalla fonte prima di inviarli alla destinazione. Ad esempio, potresti ricevere eventi creati da ticket che non includono i dati completi del ticket. Utilizzando l'arricchimento, puoi fare in modo che una funzione Lambda chiami `get-ticket` API il per i dettagli completi del ticket. Le pipe possono quindi inviare tali informazioni a una destinazione.

È possibile configurare i seguenti arricchimenti quando si configura una pipe in: EventBridge

- Destinazione API
- Amazon API Gateway
- Funzione Lambda
- Macchina a stati di Step Functions

> **Note**
>
> EventBridge Pipes supporta i flussi di lavoro Express solo 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 trasmette le risposte di arricchimento direttamente al target configurato. 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 vuoi richiamare il target con un payload vuoto, restituisci un array con empty. JSON `[{}]`

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 il target.

> **Note**
>
> Per le macchine a stati Step Functions, supporta EventBridge 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. È possibile configurare i seguenti obiettivi quando si imposta una pipe in EventBridge:

- APIdestinazione
- Gateway API
- Coda di processi batch
- CloudWatch gruppo di log
- ECScompito
- Bus di eventi nello stesso account e nella stessa Regione
- Flussi di distribuzione Firehose
- Modello di valutazione di Inspector
- Flusso di Kinesis
- Funzione Lambda (SYNCo) ASYNC
- Interrogazioni sui dati del cluster Redshift API
- SageMaker Pipeline
- SNSArgomento Amazon (SNSFIFOargomenti non supportati)

- SQSCoda Amazon
- Macchina a stati di Step Functions
  - Flussi di lavoro Express (SYNCo) 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 richiamarne uno specifico. API EventBridge usa il `PipeTargetParameters`per specificare quali informazioni vengono inviate a quel destinatario. API Questi sono i seguenti:

- APIdestinazioni (I dati inviati a una API destinazione devono corrispondere alla struttura diAPI. È necessario utilizzare l'`InputTemplate`oggetto per assicurarsi che i dati siano strutturati correttamente. Se desiderate includere il payload dell'evento originale, fate riferimento a esso in `InputTemplate`.)
- APIGateway (I dati inviati a API Gateway devono corrispondere alla struttura di. API È necessario utilizzare l'`InputTemplate`oggetto per assicurarsi che i dati siano strutturati correttamente. Se desiderate includere il payload dell'evento originale, fate riferimento a esso in `InputTemplate`.)
- `PipeTargetRedshiftDataParameters`(Cluster di dati API 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 JSON percorso dinamico opzionale. È possibile utilizzare questa sintassi per specificare JSON percorsi anziché valori statici (ad esempio`$.detail.state`). L'intero valore deve essere un JSON percorso, non solo una parte di esso. Ad esempio, `RedshiftParameters.Sql` può essere `$.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 JSON percorsi dinamici dei parametri 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 una SQS fonte Amazon

Quindi, se il payload dell'evento è una JSON stringa valida, ad esempio`{"`*`someKey`*`":"`*`someValue`*`"}`, EventBridge estrae il valore dal JSON percorso e lo utilizza come parametro di destinazione. In questo esempio, EventBridge imposterebbe Kinesis `PartitionKey` su»*`someValue`*".

Autorizzazioni

Per effettuare API chiamate sulle risorse che possiedi, EventBridge Pipes necessita dell'autorizzazione appropriata. EventBridge PIpesutilizza il IAM ruolo specificato nella pipe per le chiamate di arricchimento e di destinazione utilizzando il IAM principal`pipes.amazonaws.com`.

Richiamo di destinazioni

EventBridge ha i seguenti modi per richiamare 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 una fonte non impone l'ordine, ad esempio una SQS coda Amazon standard, EventBridge può richiamare una destinazione supportata in modo 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.

AWS Batch code di lavoro, specifiche del target.

Tutti i AWS Batch `submitJob` parametri sono configurati in modo esplicito con e`BatchParameters`, come tutti i parametri Pipe, possono essere dinamici utilizzando un JSON percorso verso il payload dell'evento in entrata.

CloudWatch Registra le specifiche del target del gruppo

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 JSON percorso verso il payload dell'evento in entrata.

Informazioni specifiche sull'ECSobiettivo delle attività di Amazon

Tutti i ECS `runTask` parametri Amazon sono configurati esplicitamente tramite`EcsParameters`. Come tutti i parametri pipe, questi parametri possono essere dinamici quando si utilizza un JSON percorso verso il payload dell'evento in entrata.

Specifiche del target delle funzioni Lambda e del flusso di lavoro Step Functions

Lambda e Step Functions non dispongono di un batch. API Per elaborare batch di eventi da un'origine pipe, il batch viene convertito in un JSON array e passato come input al target Lambda o Step Functions. Per ulteriori informazioni, consulta ???.

Timestream per le specifiche del target LiveAnalytics della tabella

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 le destinazioni che lo supportano. Inoltre, il batching to richment è supportato per e. AWS Lambda 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, una pipe da uno stream Kinesis a una SQS coda Amazon può avere una dimensione batch massima configurata sull'origine di 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 sulla sorgente, gli array di JSON record vengono passati attraverso la pipe e quindi mappati al batch di un arricchimento o API di una destinazione supportati. I trasformatori di input vengono applicati separatamente su ogni singolo JSON record dell'array, non sull'array nel suo insieme. Per esempi di questi array, consulta ??? e seleziona un'origine specifica. Pipes utilizzerà il batch API per l'arricchimento o il target supportati anche se la dimensione del batch è 1. Se l'arricchimento o la destinazione non dispone di un batch API ma riceve JSON payload completi, come Lambda e Step Functions, l'intero JSON array viene inviato in un'unica richiesta. La richiesta verrà inviata come JSON array anche se la dimensione del batch è 1.

Se una pipe è configurata per il batch all'origine e la destinazione supporta il batch, è possibile restituire una serie di JSON elementi dall'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

Target	Dimensione massima batch
Flusso di Kinesis	500
Funzione Lambda	definita dal cliente
Macchina a stati di Step Functions	definita dal cliente
SNSArgomento Amazon	10
SQSCoda Amazon	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 le fonti di streaming, come Kinesis e DynamoDB, EventBridge Pipes supporta la gestione parziale degli errori in batch degli 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 AmazonSQS, questo è il`messageId`. Per Kinesis e DynamoDB, è `eventID`. EventBridge Affinché Pipes gestisca 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 — 1250

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 sorgenti rigorosamente ordinate, come Amazon SQS FIFO queues, Kinesis e DynamoDB Streams o argomenti Apache Kafka) sono ulteriormente limitate in concomitanza dalla configurazione della fonte, come il numero di gruppi di messaggi IDs per le code o il numero di shard per le code Kinesis. FIFO Poiché l'ordinamento è strettamente garantito entro questi vincoli, una pipe con un'origine ordinata non può superare tali limiti di simultaneità.

Trasformazione degli input di Amazon EventBridge Pipes

Amazon EventBridge Pipes supporta trasformatori di input opzionali per il trasferimento dei dati all'arricchimento e alla destinazione. Puoi utilizzare i trasformatori di input per rimodellare il payload di input degli JSON eventi per soddisfare le esigenze del servizio di arricchimento o di destinazione. Per Amazon API Gateway e API destinazioni, ecco come modellare l'evento di input in base al RESTful modello del vostroAPI. I trasformatori di input sono modellati come parametro `InputTemplate`. Possono essere testo libero, un JSON percorso verso il payload dell'evento o un JSON oggetto che include JSON percorsi in linea verso il payload dell'evento. 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 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 (*)

Di seguito sono riportati alcuni `InputTemplate` parametri di esempio che fanno riferimento a un payload di SQS eventi Amazon:

Stringa statica

```
InputTemplate: "Hello, sender"
```

JSONPercorso

```
InputTemplate: <$.attributes.SenderId>
```

Stringa dinamica

```
InputTemplate: "Hello, <$.attributes.SenderId>"
```

Statico JSON

```
InputTemplate: >
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3",
}
```

Dinamico JSON

```
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 ingresso in fase di esecuzione per garantire un output valido. JSON Per questo motivo, inserite tra virgolette le variabili che fanno riferimento ai parametri del JSON percorso, ma non le virgolette attorno alle variabili che si riferiscono a JSON oggetti o matrici.

Variabili riservate

I modelli di input possono utilizzare le seguenti variabili riservate:

- `<aws.pipes.pipe-arn>`— L'Amazon Resource Name (ARN) della pipe.
- `<aws.pipes.pipe-name>`: il nome della pipe.
- `<aws.pipes.source-arn>`— La ARN fonte dell'evento della pipe.
- `<aws.pipes.enrichment-arn>`— L'ARNarricchimento della pipa.
- `<aws.pipes.target-arn>`— Il bersaglio ARN della pipa.
- `<aws.pipes.event.ingestion-time>`: l'ora alla quale il trasformatore di input ha ricevuto l'evento. Si tratta di un ISO timestamp 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 dalla fonte o restituito dall'arricchimento, è. JSON Se la pipe ha un campo codificato, come il SQS `body` campo Amazon o `data` Kinesis, tali campi vengono decodificati e resi validi. JSON Poiché non è escluso, 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 EC2 evento Amazon 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"
  }
}
```

Usiamo quanto segue JSON come nostro Transformer.

```
{
  "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 essere JSON -escaped, come l'SQS`body`oggetto Amazon, o con codifica base64, come l'oggetto Kinesis. `data` Sia per il filtraggio che per la trasformazione degli input, EventBridge trasforma questi campi in campi validi in modo da poter fare riferimento direttamente ai sottovalori. JSON 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 JSON stringa valida (ad esempio`{"key": "value"}`), l'uso del trasformatore di input con dati corporei specifici della sorgente comporterà la rimozione delle virgolette all'interno del payload di origine originale. Ad esempio, `{"key": "value"}` diventerà `"{key: value}"` quando distribuito alla destinazione. Se il target richiede JSON payload validi (ad esempio, EventBridge Lambda o Step Functions), ciò causerà un errore di consegna. Per fare in modo che la destinazione riceva i dati di origine originali senza generare dati non validiJSON, avvolgi il trasformatore di input dei dati del corpo di origine. 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 validoJSON, questo campo conterrà il payload senza escape, non codificato in base64. JSON Tuttavia, se il payload non è validoJSON, EventBridge base64 codifica per i campi elencati di seguito, ad 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

Questi sono alcuni problemi comuni che si verificano quando si trasforma l'input in EventBridge pipe:

- Per le stringhe, le virgolette sono obbligatorie.
- Non è prevista alcuna convalida durante la creazione del JSON percorso per il modello.
- Se specifichi una variabile in modo che corrisponda a un JSON percorso che non esiste nell'evento, quella variabile non viene creata e non verrà visualizzata nell'output.
- JSONproprietà come `aws.pipes.event.json` possono essere utilizzate solo come valore di un JSON campo, 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 JSON percorso fa riferimento a un JSON oggetto o a un array, ma la variabile è referenziata in una stringa, EventBridge rimuove tutte le virgolette interne per garantire una stringa valida. Ad esempio, «Body is <$.body>» comporterebbe la EventBridge rimozione delle virgolette dall'oggetto.

  Pertanto, se si desidera generare un JSON oggetto basato su una singola variabile di JSON percorso, è necessario 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 validoJSON. EventBridge

Pipes non aggiunge virgolette alle variabili che rappresentano JSON oggetti o matrici. Non aggiungete virgolette per le variabili che rappresentano JSON oggetti o matrici.

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>
}
```

Risulta valido JSON con la citazione corretta:

```
{
  "pipeArn" : "arn:aws:events:us-east-2:123456789012:pipe/example",
  "pipeName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

- Per gli arricchimenti o i target Lambda o Step Functions, i batch vengono consegnati alla destinazione come JSON array, anche se la dimensione del batch è 1. Tuttavia, i trasformatori di input verranno comunque applicati ai singoli record dell'JSONarray, non all'intero array. Per ulteriori informazioni, consulta ???.

Registrazione delle prestazioni di Amazon EventBridge Pipes

EventBridge La registrazione delle pipe consente di fare in modo che EventBridge Pipes invii i record che descrivono 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 HTTP endpoint o HTTP endpoint personalizzati di proprietà di fornitori 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.
- ERROR— 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	ERROR	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	
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			

Fase	TRACE	INFO	ERROR	OFF
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			
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 sotto JSON forma di stringa. Per le richieste inviate a una API destinazione, rappresenta la HTTP richiesta 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 rispetto 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 API destinazione, rappresenta la HTTP risposta restituita da quell'endpoint.

  `awsRequest`Analogamente, 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 EventBridge tronca i campi nell'evento, il campo 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.

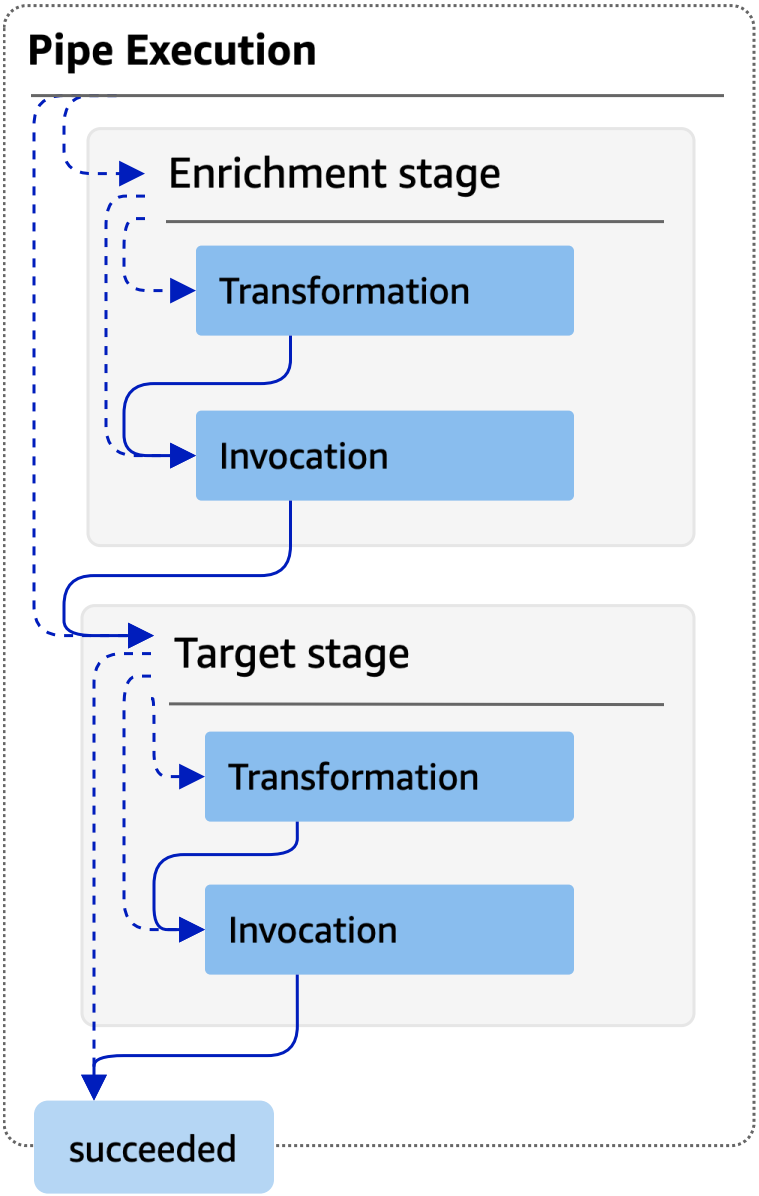
I passaggi principali di una corretta esecuzione di una pipe seguono questo flusso:

- 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.

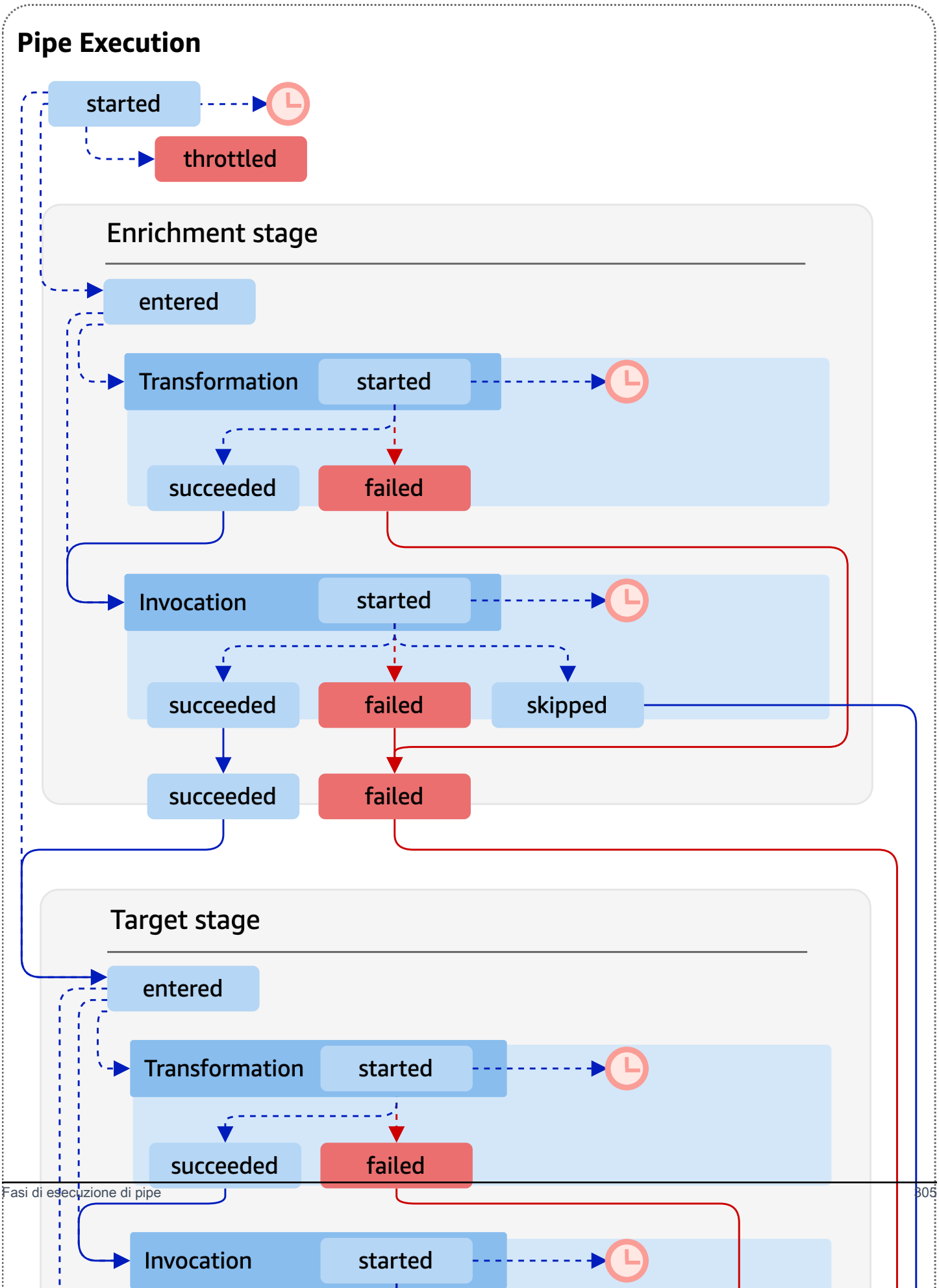
  Se non hai specificato la trasformazione o la destinazione, l'esecuzione salta la fase di destinazione.
- L'esecuzione della pipe viene completata correttamente.

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

Nota che l'invocazione della destinazione può causare un errore parziale del batch. Per ulteriori informazioni, consulta ???.

EventBridge Riferimento allo schema del registro delle pipe

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

L'Amazon Resource Name (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 hai 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 al target, in formato. JSON Per le richieste inviate a una API destinazione, rappresenta la HTTP richiesta 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 dal target, in formato. JSON Per le richieste inviate a una API destinazione, rappresenta la HTTP risposta restituita da quell'endpoint e non la risposta restituita dal servizio di API destinazione stesso.

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 HTTP di stato 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 tramite AWS CloudTrail

Puoi registrare le chiamate e l'utilizzo di Pipes CloudTrail e monitorare lo stato delle tue EventBridge pipe utilizzando le metriche. CloudWatch

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, 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

Conoscere i tipi di errori che possono verificarsi con Pipes e come EventBridge gestirli può aiutarvi a risolvere i problemi relativi alle EventBridge pipe.

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. In caso di errori 4xx quasi continui (ad esempio problemi di autorizzazione IAM o risorse mancanti), la pipe può essere disattivata automaticamente inserendo un messaggio esplicativo nel. `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 HTTP connessione durante il tentativo di richiamare il servizio clienti targer
- 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 di 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 una SQS fonte Amazon, EventBridge Pipes non conta i nuovi tentativi per errori interni rispetto al conteggio massimo di ricezione per l'SQSorigine Amazon.

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 Lambda, Step Functions, destination o Gateway del cliente richiamato in modo sincrono. API API

Per gli errori di invocazione dei clienti, Pipes esegue le seguenti operazioni: EventBridge

- 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 di quale evento si verifica per primo.
- Per le pipe con una SQS fonte Amazon, EventBridge Pipes ripete un errore del cliente fino al numero massimo di ricevute 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 le fonti di streaming, come Kinesis e DynamoDB, EventBridge Pipes supporta la gestione parziale degli errori in batch degli errori di destinazione. Per ulteriori informazioni, consulta Errori batch parziali.

Comportamento delle pipe DLQ

Una pipe eredita il comportamento di dead-letter queue (DLQ) dalla fonte:

- Se la SQS coda Amazon di origine è configurataDLQ, i messaggi vengono recapitati automaticamente lì dopo il numero di tentativi specificato.
- Per le sorgenti di flusso, come i flussi DynamoDB e Kinesis, puoi DLQ configurare un per gli eventi pipe e route. Le sorgenti di flusso DynamoDB e Kinesis supportano le code SNS Amazon e gli argomenti SQS Amazon come destinazioni. DLQ

Se specificate un `DeadLetterConfig` per una pipe con una sorgente Kinesis o DynamoDB, assicuratevi che la `MaximumRecordAgeInSeconds` proprietà sulla pipe sia inferiore a quella dell'evento source. `MaximumRecordAge` `MaximumRecordAgeInSeconds`controlla quando il pipe poller rinuncerà all'evento e lo consegnerà a DLQ e `MaximumRecordAge` controlla per quanto tempo il messaggio sarà visibile nel flusso di origine prima che venga eliminato. Pertanto, impostate un valore inferiore `MaximumRecordAgeInSeconds` a quello della fonte in `MaximumRecordAge` modo che trascorra un periodo di tempo adeguato tra il momento in cui l'evento viene inviato a e il momento in cui viene eliminato automaticamente dalla fonteDLQ, in modo da determinare il motivo per cui l'evento è stato inviato a. DLQ

Per le sorgenti Amazon MQ, DLQ possono essere configurate direttamente sul broker di messaggi.

EventBridge Pipes non supporta first-in first-out (FIFO) DLQs per le sorgenti di streaming.

EventBridge Pipes non supporta DLQ Amazon MSK stream e sorgenti di stream 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:ReceiveMessage, sqs: e sqs:) DeleteMessage GetQueueAttributes
- Errore di connessione. VPCDevi essere in grado di connetterti ai tubi. È possibile fornire l'accesso configurando un NAT gateway o un VPC endpoint a pipes-data. Per come configurare il NAT gateway o l'VPCendpoint su pipes-data, consulta la documentazione. AWS
- MSKal cluster 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 AWS KMS chiave AWS-managed, devi fornire esplicitamente l'autorizzazione di decrittografia del ruolo di esecuzione della tua pipe. A tale scopo, includi la seguente autorizzazione aggiuntiva nella politica personalizzata: CMK

```
  {
      "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.

Quindi, assicurati che le stesse autorizzazioni per KMS vengano aggiunte al tuo ruolo di esecuzione Pipe.

Questo vale per tutte le sorgenti pipe con AWS KMS CMK, tra cui:

- Amazon DynamoDB Streams
- Amazon Kinesis Data Streams
- Amazon MQ
- Amazon MSK
- Amazon SQS

Tutorial: crea una EventBridge pipe che filtra gli eventi di origine

In questo tutorial, creerai una pipe che collega una sorgente di flusso DynamoDB a una destinazione di coda AmazonSQS. 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 pipe e una coda SQS Amazon come destinazione pipe.

Per semplificare questo passaggio, puoi utilizzare il provisioning delle AWS CloudFormation risorse di origine e destinazione. Per fare ciò, creerai un CloudFormation modello che definisca 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 SQS coda Amazon, denominata`pipe-tutorial-target`, per ricevere il flusso di eventi DynamoDB dalla tua pipe.

Per creare il CloudFormation modello per il provisioning delle risorse Pipe

1. Copia il testo del JSON modello nella ??? sezione seguente.
2. Salvate il modello come JSON file (ad esempio,`~/pipe-tutorial-resources.json`).

Quindi, utilizzate il file modello appena creato per effettuare il provisioning di uno CloudFormation stack.

> **Note**
>
> Una volta creato lo CloudFormation stack, ti verranno addebitate le AWS risorse che fornisce.

Fornisci i prerequisiti del tutorial utilizzando il AWS CLI

- Esegui il CLI comando seguente, dove `--template-body` specifica la posizione del file modello:

```
aws cloudformation create-stack --stack-name pipe-tuturial-resources --template-
body file://~/pipe-tutorial-resources.json
```

Fornisci i prerequisiti del tutorial utilizzando la console CloudFormation

1. Apri la AWS CloudFormation console 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 Create 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`.
   - Per Parametri, lascia i nomi predefiniti per la tabella DynamoDB e la coda Amazon. SQS
   - Seleziona Successivo.
6. Scegli Successivo, quindi scegli Invia.

   CloudFormation crea lo stack e fornisce le risorse definite nel modello.

Per ulteriori informazioni su CloudFormation, consulta What is AWS CloudFormation? nella Guida AWS CloudFormation per l'utente.

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.

Crea la pipe usando la EventBridge 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. 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 del codice sorgente specifiche di DynamoDB.

   b. In Flusso DynamoDB, seleziona `pipe-tutorial-source`.

      Lascia Posizione di partenza impostata sul valore predefinito, `Latest`.

   c. Seleziona Successivo.
6. Specifica e testa un modello di eventi per filtrare gli eventi:

I filtri consentono 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 evento. Questo perché nell'evento il valore `eventName` è `INSERT`.

d. Seleziona 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. Specificate la vostra SQS coda Amazon come destinazione della pipe:

   a. In Dettagli, per il servizio Target, seleziona Amazon SQS queue.

   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 tubazione e visualizza la pagina dei dettagli della tubazione. 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 SQS coda di Amazon per assicurarti che la pipe abbia fornito solo gli eventi che corrispondono al nostro schema 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 Salvataggio delle 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 SQS console Amazon 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 l'evento JSON che è stato consegnato.

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.

Eliminare la pipe utilizzando la EventBridge console

1. Apri la EventBridge console Amazon 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 CloudFormation stack, per evitare che ti venga addebitato l'utilizzo continuato delle risorse fornite al suo interno.

Eliminate i prerequisiti del tutorial utilizzando il AWS CLI

- Esegui il CLI comando seguente, dove `--stack-name` specifica il nome del tuo stack:

```
aws cloudformation delete-stack --stack-name pipe-tuturial-resources
```

Eliminare i prerequisiti del tutorial utilizzando la console AWS CloudFormation

1. Apri la AWS CloudFormation console 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.

AWS CloudFormation modello per la generazione dei prerequisiti

Usa quanto JSON segue per creare un CloudFormation modello per il provisioning delle risorse di origine e di 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" }
      }
    }
  }
}
```

Generazione di 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 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 risorsa AWS: :Pipes: :Pipe per ogni pipe selezionata. Inoltre, EventBridge include le seguenti risorse nelle condizioni descritte:

- AWS: :Eventi:: ApiDestination

  Se le tue pipe includono API destinazioni, sia come arricchimenti che come obiettivi, le EventBridge include nel CloudFormation modello come AWS::Events::ApiDestination risorse.
- AWS: :Eventi:: 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: :Ruolo

Se hai EventBridge creato un nuovo ruolo di esecuzione quando hai configurato la pipe, puoi 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 il ARN ruolo.)

Considerazioni sull'utilizzo di CloudFormation modelli generati da EventBridge Pipes

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.

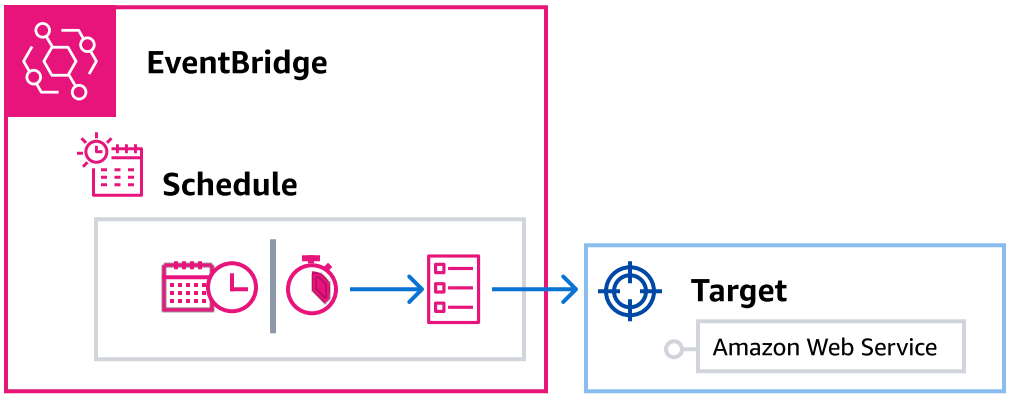
4. Scegliete CloudFormation Modello, quindi scegliete il formato in cui desiderate EventBridge generare il modello: JSONoppure YAML.

   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.

Amazon EventBridge 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. Con EventBridge Scheduler, puoi creare pianificazioni utilizzando le espressioni cron e rate per schemi ricorrenti o configurare chiamate una tantum. È possibile impostare finestre temporali flessibili per la consegna, definire limiti di nuovi tentativi e impostare il tempo massimo di conservazione per le chiamate non riuscite. API



EventBridge Scheduler è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole EventBridge pianificate, con un set più ampio di API operazioni e servizi mirati. AWS Si consiglia di utilizzare EventBridge Scheduler per richiamare gli obiettivi in base a una pianificazione.

Configurare il ruolo di esecuzione

Quando crei una nuova EventBridge pianificazione, Scheduler deve avere l'autorizzazione a richiamare l'APIoperazione di destinazione per tuo conto. Concedi queste autorizzazioni a EventBridge Scheduler utilizzando un ruolo di esecuzione. La policy di autorizzazione collegata al ruolo di esecuzione della pianificazione definisce le autorizzazioni necessarie. Queste autorizzazioni dipendono dall'obiettivo che EventBridge Scheduler API deve richiamare.

Quando si utilizza la console EventBridge Scheduler per creare una pianificazione, come nella procedura seguente, EventBridge Scheduler imposta automaticamente un ruolo di esecuzione in base all'obiettivo selezionato. Se si desidera creare una pianificazione utilizzando uno degli EventBridge Scheduler SDKs AWS CLI AWS CloudFormation, oppure è necessario disporre di un ruolo di esecuzione esistente che conceda le autorizzazioni richieste da EventBridge Scheduler 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 Scheduler. EventBridge

Creare una pianificazione.

Per creare una pianificazione utilizzando la console

1. Apri la console Amazon EventBridge Scheduler a https://console.aws.amazon.com/scheduler/casa.
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.	

Ricorrenza	Esegui questa operazione…	
	• 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 Schedule types on EventBridge Scheduler nella Amazon EventBridge Scheduler User Guide. b. Per Finestra temporale flessibile, scegli Disattivata per disattivare l'opzione o scegli una delle finestre temporali predefinite. Ad esempio,	

Ricorrenza	Esegui questa operazione…	
	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 Next (Successivo).

7. Nella pagina Seleziona destinazione, scegli l' AWS APIoperazione richiamata da Scheduler: EventBridge

   a. Per Target API, scegli Obiettivi basati su modelli.

   b. Scegli Amazon EventBridge PutEvents.

   c. In PutEvents, specifica quanto segue:

      - Per il bus degli EventBridge eventi, scegli il bus dell'evento dal menu a discesa. Ad esempio **`default`**.

        Puoi anche creare un nuovo bus di eventi nella EventBridge console selezionando Crea 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 AWS di servizio, specificate 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): per Dettagli, inserite uno schema di eventi per filtrare ulteriormente gli eventi a cui lo EventBridge Scheduler invia. EventBridge

  Per ulteriori informazioni, consulta ???.

8. Scegli Next (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 politica di ripetizione dei tentativi per la tua pianificazione, in Politica di riprova e dead-letter queue () DLQ, procedi come segue:

      - Attiva/disattiva Riprova.
      - Per Età massima dell'evento, inserisci il numero massimo di ore e minuti in cui EventBridge Scheduler deve conservare un evento non elaborato.
      - La durata massima è 24 ore.
      - Per Numero massimo di tentativi, inserisci il numero massimo di volte in cui EventBridge Scheduler riprova la pianificazione se la destinazione restituisce un errore.

        Il valore massimo è 185 tentativi.

      Con le politiche di ripetizione dei tentativi, se una pianificazione non riesce a richiamare l'obiettivo, EventBridge Scheduler esegue nuovamente la pianificazione. Se configurato, è necessario impostare il tempo di conservazione massimo e i nuovi tentativi per la pianificazione.

   c. Scegli dove EventBridge Scheduler archivia gli eventi non consegnati.

Opzione Dead-letter queue () DLQ	Esegui questa operazione…	
Non conservare	Scegliere None (Nessuno).	
Memorizza l'evento nello stesso spazio in Account AWS cui stai creando il programma	a. Scegli Seleziona una SQS coda Amazon nel mio Account AWS account. DLQ b. Scegli l'Amazon Resource Name (ARN) della SQS coda Amazon.	
Archivia l'evento in un luogo diverso Account AWS da quello in cui stai creando il programma	a. Scegli Specificare una SQS coda Amazon in altro Account AWS formato. DLQ b. Inserisci l'Amazon Resource Name (ARN) della SQS coda Amazon.	

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 una KMS chiave esistente ARN o scegli Crea un codice AWS KMS key per accedere alla AWS KMS console. Per ulteriori informazioni su come EventBridge Scheduler crittografa i dati inattivi, consulta Encryption at rest nella Amazon EventBridge Scheduler User Guide.

e. Per fare in modo che EventBridge Scheduler crei un nuovo ruolo di esecuzione per te, scegli Crea nuovo ruolo per questa pianificazione. Inserisci, quindi, un nome per Nome ruolo. Se scegli questa opzione, EventBridge Scheduler assegna al ruolo le autorizzazioni necessarie per la destinazione basata sul modello.

10. Scegli Next (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 su EventBridge Scheduler, consulta quanto segue:

- EventBridge Guida per l'utente di Scheduler
- EventBridge Riferimento allo Scheduler API
- EventBridge Prezzi Scheduler

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 o direttamente nella propria IDE utilizzando i API 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 Open 3 e Draft4. API JSONSchema

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

APIMascheratura del valore delle proprietà del registro dello schema

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, consigliamo EventBridge 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`

Individuazione di uno schema di eventi di AWS servizio in Amazon EventBridge

EventBridge include schemi per tutti i AWS servizi che generano eventi. È possibile trovare questi schemi nella EventBridge console oppure è possibile trovarli utilizzando l'azioneAPI. `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 degli schemi in Amazon EventBridge

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 utilizzandoAPI, usa `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 schema di eventi in Amazon EventBridge

È possibile creare schemi utilizzando JSON file con la APIspecifica Open Specification o JSONSchemaDraft4. È possibile creare o caricare schemi personalizzati utilizzando un modello o generando uno schema basato su un evento. EventBridge JSON Puoi anche dedurre lo schema da eventi in router di eventi. Per creare uno schema utilizzando il registro degli EventBridge schemiAPI, usa l'`CreateSchema`APIazione.

Quando scegliete tra i formati Open API 3 e JSONSchema Draft4, considerate le seguenti differenze:

- JSONSchemail formato 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`.
- Open API non supporta i collegamenti JSONSchema ipertestuali Hyper-Schema nei documenti. JSON
- Gli strumenti per Open API tendono a concentrarsi sulla fase di compilazione, mentre gli strumenti per Open JSONSchema tendono a concentrarsi sulle operazioni in fase di esecuzione, come gli strumenti client per la convalida dello schema.

Si consiglia di utilizzare il JSONSchema formato per implementare la convalida lato client in modo che gli eventi inviati siano conformi allo schema. EventBridge È possibile utilizzare JSONSchema per definire un contratto per JSON documenti validi e quindi utilizzare un validatore di JSON schemi 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.

Creazione di uno schema utilizzando un modello in Amazon EventBridge

È possibile creare uno schema da un file modello scaricato o modificando un modello direttamente nella EventBridge console.

Crea uno schema da un file modello

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 JSON modello dal seguente esempio di codice.

```
{
    "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. Per il tipo di schema, scegli Open API 3.0 o JSONSchema 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

È possibile creare uno schema direttamente nella EventBridge 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. Per il tipo di schema, scegli Open API 3.0 o JSONSchema 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 da un evento JSON in Amazon EventBridge

Se hai il nome JSON di un evento, puoi creare automaticamente uno schema per quel tipo di evento.

Per creare uno schema basato su un evento JSON

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. Per il tipo di schema, scegli Open API 3.0.

   Non è possibile utilizzarlo JSONSchema quando si crea uno schema a partire da un evento. JSON
7. Seleziona Scopri da JSON
8. Nella casella di testo sottostante JSON, incolla o trascina l'JSONorigine 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 API schema aperto 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.

Deduzione di schemi dagli eventi del bus degli eventi in EventBridge

Amazon 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 Discovered schemas nella pagina Schemas.

Se il contenuto degli eventi sul bus degli eventi cambia, EventBridge crea nuove versioni dello schema correlato. EventBridge

Considerazioni sull'avvio del rilevamento dello schema su un bus di eventi

Tenete conto delle seguenti considerazioni prima di abilitare lo schema discover su un bus di eventi:

- 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.
- EventBridge per impostazione predefinita, deduce gli schemi dagli eventi tra account diversi, ma è possibile disabilitarlo aggiornando la proprietà. `cross-account` Per ulteriori informazioni, vedere Discoverers in the Schema Registry Reference. EventBridge API

> **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 KMS key opzioni.

Per avviare o interrompere l'individuazione dello schema su un bus di eventi (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. Seleziona il bus degli eventi su cui desideri avviare o interrompere il rilevamento dello schema.
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 (AWS CLI)

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

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

Generazione di associazioni di codice per schemi di eventi in Amazon EventBridge

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. È possibile generare associazioni di codice per uno schema utilizzando la EventBridge console, lo EventBridge Schema Registry API o utilizzando un toolkit. IDE 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.

AWS integrazioni di servizi e strumenti con Amazon EventBridge

Amazon EventBridge collabora con altri AWS servizi e strumenti per elaborare eventi o richiamare una risorsa come obiettivo di una regola. Per ulteriori informazioni sulle EventBridge integrazioni con altri AWS servizi e strumenti, consulta quanto segue:

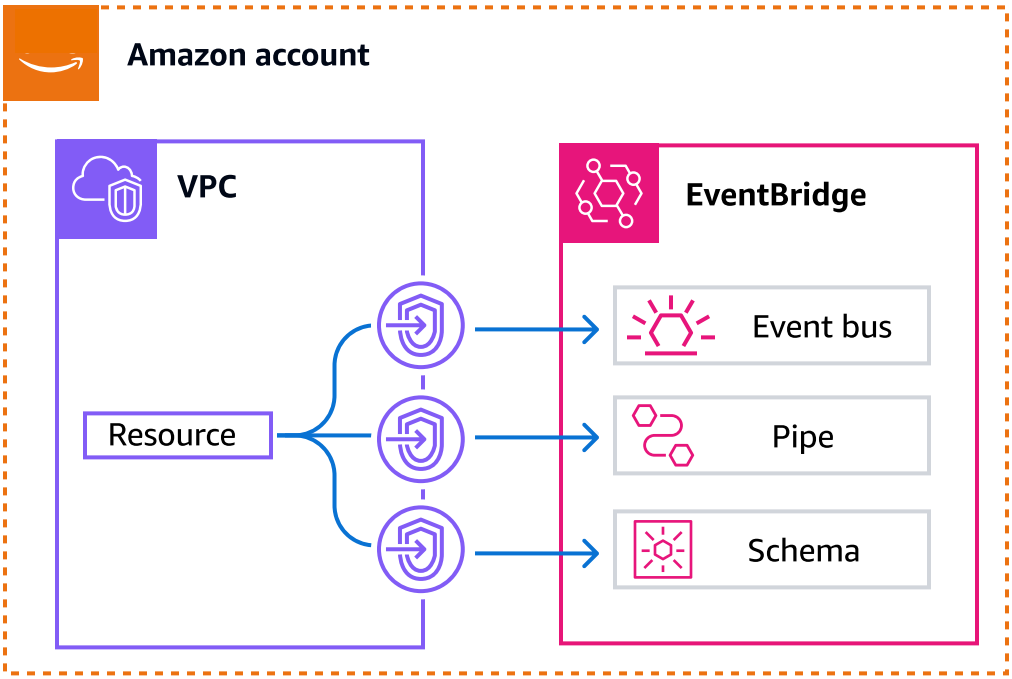
Argomenti

Utilizzo di Amazon EventBridge con VPC endpoint di interfaccia

Se utilizzi Amazon Virtual Private Cloud (AmazonVPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e EventBridge. Le tue risorse a disposizione VPC possono utilizzare questa connessione per comunicare con EventBridge.

Con aVPC, hai il controllo sulle impostazioni di rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Per connetterti VPC a EventBridge, definisci un endpoint di interfaccia VPC per. EventBridge L'endpoint fornisce una connettività affidabile e scalabile EventBridge senza richiedere un gateway Internet, un'istanza di traduzione degli indirizzi di rete (NAT) o una connessione. VPN Per ulteriori informazioni, consulta What is Amazon VPC nella Amazon VPC User Guide.

Gli VPC endpoint di interfaccia sono alimentati da AWS PrivateLink, che consente la comunicazione privata tra AWS i servizi utilizzando un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, consulta AWS PrivateLink ed VPC endpoints.



Quando utilizzi un VPC endpoint con interfaccia privata, VPC invii eventi personalizzati per EventBridge utilizzare quell'endpoint. EventBridge quindi invia tali eventi ad altri AWS servizi in base alle regole e agli obiettivi che hai configurato. Una volta inviati gli eventi a un altro servizio, puoi riceverli tramite l'endpoint pubblico o un VPC endpoint per quel servizio. Ad esempio, se crei una

regola per inviare eventi a una SQS coda Amazon, puoi configurare un VPC endpoint di interfaccia per consentire SQS ad Amazon di ricevere messaggi da quella coda VPC senza utilizzare l'endpoint pubblico.

Creazione di un endpoint per VPC EventBridge

Da utilizzare EventBridge con il tuoVPC, crea un VPC endpoint di interfaccia per EventBridge e scegli il nome di servizio appropriato EventBridge . Per ulteriori informazioni, consulta Creating an Interface Endpoint nella Amazon VPC User Guide.

- Autobus per eventi

  Nome del servizio: com.amazonaws.***region***.eventi
- Tubi

  Nome del servizio: com.amazonaws.***region***.pipe

  EventBridge Pipes supporta gli endpoint per tutte le operazioni sulle tubazioni API.

  FIPSGli endpoint Pipes supportano VPC anche gli endpoint.

  Nome del servizio: com.amazonaws.***region***.pipes-fips

  Gli endpoint Fips sono supportati nelle seguenti regioni:
  - Stati Uniti occidentali (California settentrionale)
  - US West (Oregon)
  - Stati Uniti orientali (Virginia settentrionale)
  - Stati Uniti orientali (Ohio)
  - Canada (Centrale)

  Puoi anche utilizzare un VPC endpoint per soddisfare i requisiti di rete per le sorgenti Pipes, Apache Kafka e Amazon MQ.

  Nome del servizio: com.amazonaws. ***region***.pipes-data

  Per ulteriori informazioni, fare riferimento a quanto segue:
  - Configurazione della rete Apache Kafka
  - Configurazione MSK di rete Amazon
  - Configurazione di rete Amazon MQ

> **Note**
>
> VPCgli endpoint to pipes-data non supportano VPC le policy relative alle risorse degli endpoint.
> VPCgli endpoint to pipe e pipes-fips supportano le policy relative alle risorse degli Endpoint che consentono di: VPC
>
> - Negare l'accesso a Pipe specifici. APIs
> - Limita l'accesso su alcuni APIs a Pipes specifici ARN utilizzando la chiave di condizione IAM Resource.

- Schemi

  Nome del servizio: com.amazonaws.***region***.schema

  EventBridge supporta gli endpoint per tutte le operazioni dello schema API.

Disponibilità

EventBridge attualmente supporta VPC gli endpoint 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 (Malesia)
- Asia Pacifico (Tokyo)

- Asia Pacifico (Seoul)
- 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 () UAE
- Medio Oriente (Bahrein)
- Sud America (San Paolo)
- Israele (Tel Aviv)
- AWS GovCloud (Stati Uniti occidentali)
- AWS GovCloud (Stati Uniti orientali)

EventBridge Integrazione Amazon con AWS X-Ray

È possibile utilizzare AWS X-Ray per tracciare gli eventi che EventBridge passano. 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 `PutEvents` richiesta che ha superato il contesto di traccia. X-Ray non traccia gli eventi che provengono da partner, eventi pianificati o AWS servizi 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 si dispone di un codice di tipo dead-letter queue (DLQ), l'intestazione trace viene inclusa nella `SendMessage` richiesta che invia l'evento a. DLQ Se recuperi eventi (messaggi) da DLQ by using`ReceiveMessage`, l'intestazione trace associata all'evento è inclusa nell'attributo Amazon SQS message, ma non è inclusa nel messaggio dell'evento.

Per informazioni su come un nodo EventBridge evento connette i servizi di origine e destinazione, vedere Visualizzazione della sorgente e delle destinazioni nella mappa dei servizi X-Ray nella AWS X-Ray Developer Guide.

È possibile trasmettere le seguenti informazioni sull'intestazione di traccia tramite: EventBridge

- Intestazione predefinita: SDK X-Ray compila automaticamente l'HTTPintestazione della traccia come intestazione per tutte le destinazioni di `X-Amzn-Trace-Id` HTTP invocazione. Per ulteriori informazioni sull'intestazione predefinita, consulta l'HTTPintestazione Tracing nella Developer Guide.AWS X-Ray
- **`TraceHeader`**attributo di sistema: `TraceHeader` è un PutEventsRequestEntry attributo riservato EventBridge a trasportare l'intestazione della traccia X-Ray su un bersaglio. Se lo usi

anche`PutEventsRequestEntry`, `PutEventsRequestEntry` sovrascrive l'intestazione della 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 immissione PutEvents degli eventi.

Il seguente video illustra l'uso di X-Ray EventBridge e insieme: AWS X-Ray Utilizzo per il tracciamento

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 Integrated Application Test Kit (AWS IATK) consente di implementare alcune di queste strategie per le applicazioni.

AWS IATKè una libreria software che consente di scrivere test automatici per applicazioni basate su cloud.

EventBridge integrazione con AWS 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 IATKpuò creare per te un test harness che ascolta gli eventi EventBridge di output. (Sotto il cofano, l'imbracatura è composta da una EventBridge regola che inoltra l'evento di output ad AmazonSQS.) 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 IATKoffre la possibilità di generare eventi fittizi da uno schema memorizzato nel registro degli EventBridge schemi. 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: :Eventi

  Gli esempi di modelli includono:

  - Crea una API destinazione per PagerDuty
  - Crea una API destinazione per Slack
  - Crea una connessione con ApiKey parametri di autorizzazione
  - Crea una connessione con parametri di OAuth autorizzazione

  - 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::EventSchemas
- AWS: :Tubi

  Gli esempi di modelli includono:

  - Crea una pipe con un filtro per gli eventi
- AWS: :Pianificatore

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.

Integrazioni di terze parti con Amazon EventBridge

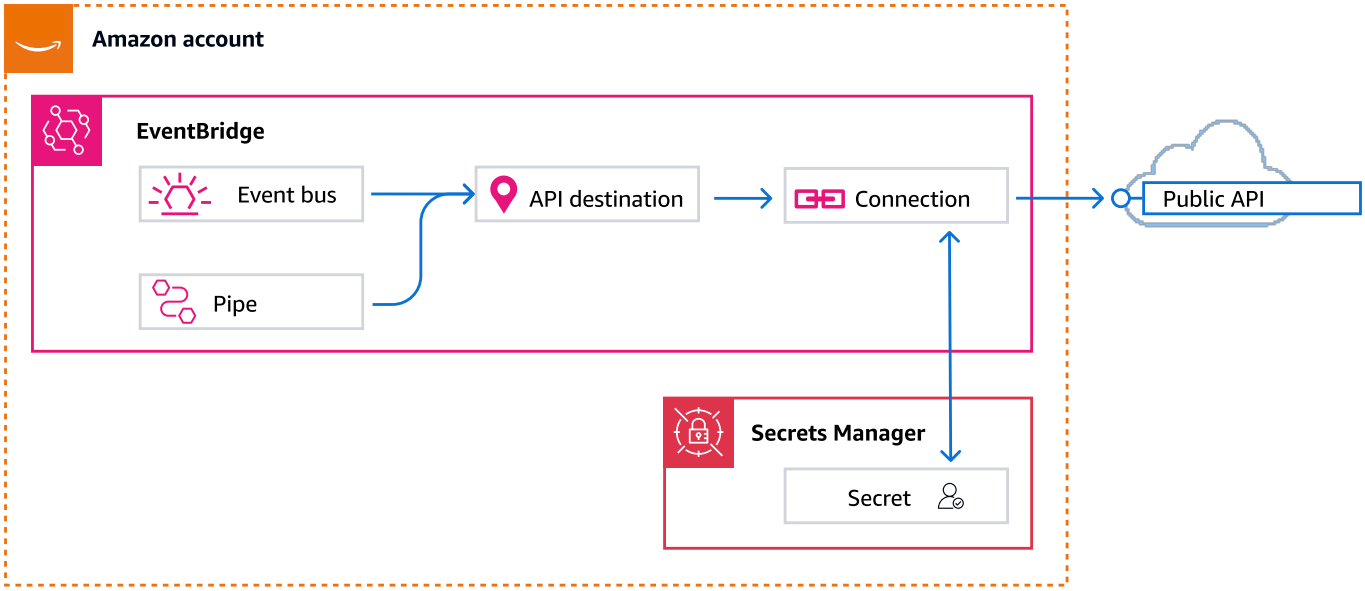
Oltre ai AWS servizi, Amazon EventBridge supporta l'integrazione con sistemi e applicazioni personalizzati e di terze parti, sia come fonti che come destinazioni di eventi.

L'integrazione di terze parti che EventBridge supporta include:

- Creazione di APIdestinazioni per inviare eventi agli HTTP endpoint pubblici da un event bus o pipe.
- Consentire al bus degli eventi di ricevere eventi da una fonte partner Software as a Service (SaaS).

APIdestinazioni come obiettivi in Amazon EventBridge

Le EventBridge APIdestinazioni Amazon sono HTTP endpoint che puoi richiamare come destinazione di una regola o pipe del bus di eventi, in modo simile a come richiami un AWS servizio o una risorsa come destinazione. Utilizzando API le destinazioni, è possibile instradare gli eventi tra AWS servizi, applicazioni SaaS (Software as a Service) integrate e applicazioni esterne AWS utilizzando API chiamate. Quando specificate una API destinazione come regola o destinazione della pipe, EventBridge richiama l'HTTPendpoint per qualsiasi evento che corrisponde al modello di evento specificato nella regola o nella pipe e quindi fornisce le informazioni sull'evento con la richiesta. Con EventBridge, è possibile utilizzare qualsiasi HTTP metodo tranne CONNECT e TRACE per la richiesta. I HTTP metodi più comuni da utilizzare sono PUT ePOST. È inoltre possibile utilizzare trasformatori di input per personalizzare l'evento in base ai parametri di uno specifico HTTP endpoint. Per ulteriori informazioni, consulta Trasformazione degli EventBridge input di Amazon.



> **Note**
>
> APIle destinazioni non supportano destinazioni private, come gli VPC endpoint di interfaccia, inclusi quelli privati HTTPS APIs nei Virtual Private Clouds (VPC) che utilizzano Network e Application Load Balancer privati e endpoint di interfaccia. VPC
> Per ulteriori informazioni, consulta ???.

> **Important**
>
> EventBridge le richieste verso un endpoint di API destinazione devono avere un timeout di esecuzione del client massimo di 5 secondi. Se l'endpoint di destinazione impiega più di 5 secondi per rispondere, scade la EventBridge richiesta. 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 mostra l'uso della API destinazione: Utilizzo delle destinazioni API

Ruolo legato ai servizi per le destinazioni API

Quando crei una connessione per una API destinazione, 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, vedere Utilizzo dei ruoli collegati ai servizi nella documentazione. IAM

Disponibilità nelle regioni

Il ruolo `AmazonEventBridgeApiDestinationsServiceRolePolicy` collegato ai servizi è supportato nelle seguenti aree: 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 nelle richieste alle destinazioni API

La sezione seguente descrive in dettaglio come EventBridge gestisce le HTTP intestazioni nelle richieste alle API destinazioni.

Intestazioni incluse nelle richieste alle destinazioni API

Oltre alle intestazioni di autorizzazione definite per la connessione utilizzata per una API destinazione, 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 alle 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 API destinazione:

- A-IM
- Accept-Charset

- Accept-Datetime
- Accept-Encoding
- Cache-Control
- Connessione
- Content-Encoding
- Content-Length
- Contenuto- MD5
- Data
- Expect
- Forwarded
- Da
- Host
- HTTP2-Impostazioni
- 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

APIcodici di errore di destinazione

Quando EventBridge tenta di inviare un evento a una API destinazione e si verifica un errore, EventBridge esegue 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 APIIe destinazioni leggono l'intestazione di HTTP risposta 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 non recapitate per elaborare eventi non consegnati in EventBridge.

Crea una API destinazione in Amazon EventBridge

La creazione di una API destinazione consente di specificare un HTTP endpoint come obiettivo di una regola.

Ogni API destinazione richiede una connessione. Una connessione specifica il tipo di autorizzazione e le credenziali da utilizzare per l'autorizzazione con l'APIendpoint di destinazione. È possibile scegliere una connessione esistente o creare una connessione contemporaneamente alla creazione della destinazione. API Per ulteriori informazioni, consulta ???

Per creare una API destinazione utilizzando la EventBridge console

1. Accedi AWS utilizzando un account con le autorizzazioni necessarie per gestire EventBridge e aprire la EventBridgeconsole.
2. Nel riquadro di navigazione a sinistra, scegli APIIle destinazioni.
3. Scorri verso il basso fino alla tabella delle APIdestinazioni, quindi scegli Crea API destinazione.
4. Nella pagina Crea API destinazione, inserisci un nome per la API destinazione. 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. Inserisci una descrizione per la API destinazione.
6. Inserire un endpoint di API destinazione per la API destinazione. L'endpoint di APIdestinazione è un endpoint di HTTP invocazione per gli eventi. Le informazioni di autorizzazione incluse nella connessione utilizzata per questa API destinazione vengono utilizzate per l'autorizzazione su questo endpoint. L'uso URL imprescindibile. HTTPS
7. Immettere il HTTPmetodo da utilizzare per connettersi all'endpoint di API destinazione.
8. (Facoltativo) Nel campo Limite di frequenza di invocazione al secondo, inserisci il numero massimo di chiamate al secondo da inviare all'endpoint di destinazione. API

   Il limite di velocità impostato può influire sulla modalità di erogazione degli eventi. EventBridge Per ulteriori informazioni, consulta Impatto della frequenza di invocazione sulla distribuzione degli eventi.
9. In Connessione, esegui una delle seguenti operazioni:
   - Scegli Usa una connessione esistente, quindi seleziona la connessione da usare per questa API destinazione.
   - Scegli Crea una nuova connessione, quindi immetti i dettagli della connessione da creare. Per ulteriori informazioni, consulta Connessioni.
10. Scegli Create (Crea) .

Creazione di regole per l'invio di eventi a una API destinazione in EventBridge

Dopo aver creato una API destinazione, è possibile selezionarla come destinazione di una regola. Per utilizzare una API destinazione come destinazione, è necessario fornire un IAM ruolo con le autorizzazioni corrette. Per ulteriori informazioni, consulta ???

La selezione di una API destinazione come destinazione fa parte della creazione della regola.

Per creare una regola che invii eventi a una API destinazione utilizzando la console

1. Segui i passaggi nella procedura ???.
2. Nel ??? passaggio, quando viene richiesto di scegliere una API destinazione come tipo di destinazione:

   a. Seleziona la EventBridge APIdestinazione.

   b. Esegui una di queste operazioni:

      - Scegli Usa una API destinazione esistente e seleziona una API destinazione esistente
      - Scegli Crea una nuova API destinazione e specifica l'impostazione necessaria per definire la nuova API destinazione.

        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.

Invio di CloudEvents eventi a API destinazioni

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 API destinazione come tipo di destinazione:
   a. Seleziona la EventBridge APIdestinazione.
   b. Esegui una di queste operazioni:
      - Scegli Usa una API destinazione esistente e seleziona una API destinazione esistente
      - Scegli Crea una nuova API destinazione e specifica l'impostazione necessaria per definire la nuova API destinazione.

        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.

APIpartner di destinazione in Amazon EventBridge

Utilizza le informazioni fornite dai seguenti AWS partner per configurare una API destinazione e una connessione per il loro servizio o applicazione.

Osservabilità nel cloud di Cisco

APIendpoint di invocazione della destinazione: URL

```
https://tenantName.observe.appdynamics.com/rest/awsevents/aws-eventbridge-integration/endpoint
```

Tipi di autorizzazione supportati:

OAuthcredenziali del client

OAuthi token vengono aggiornati quando viene restituita una risposta 401 o 407

Parametri di autorizzazione aggiuntivi necessari:

Cisco AppDynamics Client ID e Client Secret

OAuthendpoint:

```
https://tenantName.observe.appdynamics.com/auth/tenantId/default/oauth2/token
```

I seguenti parametri della coppia OAuth chiave/valore:

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 di uso API comune:

Non applicabile

Informazioni aggiuntive:

Scegliendo Cisco AppDynamics dal menu a discesa Partner destination vengono precompilate le OAuth informazioni 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

APIendpoint di invocazione della destinazione: URL

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'RESTendpoint 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

RESTProxy 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, create 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

APIendpoint di invocazione di destinazione URL

Per un elenco completo degli endpoint, vedi Reference. Coralogix API

Tipi di autorizzazione supportati

APIChiave

Parametri di autorizzazione aggiuntivi necessari

Intestazione"`x-amz-event-bridge-access-key`", il valore è la chiave Coralogix API

Documentazione di Coralogix

EventBridgeAutenticazione Amazon

APIOperazioni di uso comune

Stati Uniti: https://ingress.coralogix.us/aws/event-bridge

Singapore: ponte per https://ingress.coralogixsg.com/aws/ eventi

Irlanda: event-bridge https://ingress.coralogix.com/aws/

Stoccolma: ponte per eventi https://ingress.eu2.coralogix.com/aws/

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

APIendpoint di invocazione della destinazione URL

Per un elenco completo degli endpoint, vedi Reference. Datadog API

Tipi di autorizzazione supportati

APIChiave

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di Datadog

Autenticazione

APIOperazioni comunemente utilizzate

POST https://api.datadoghq.com/api/v1/eventi

POST https://http-intake.logs.datadoghq.com/v1/ingresso

Informazioni aggiuntive

Gli endpoint URLs variano a seconda della posizione dell'organizzazione Datadog. Per informazioni corrette URL per la tua organizzazione, consulta la documentazione.

Freshworks

APIendpoint di invocazione di destinazione URL

Per un elenco degli endpoint, vedere https://developers.freshworks.com/documentation/

Tipi di autorizzazione supportati

Basic, Key API

Parametri di autorizzazione aggiuntivi necessari

Non applicabile

Documentazione di Freshworks

Autenticazione

APIOperazioni comunemente utilizzate

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

APIendpoint di invocazione di destinazione URL

https://data.mongodb-api.com/app/*App ID*/endpoint/

Tipi di autorizzazione supportati

APIChiave

E-mail/password

JWTAutenticazione personalizzata

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di MongoDB

Dati Atlas API

Endpoints

HTTPSEndpoint personalizzati

Autenticazione

Operazioni di uso API comune

Nessuno

Informazioni aggiuntive

Nessuno

Momento

APIendpoint di invocazione della destinazione: URL

https://api.cache.*region*.prod.a.momentohq.com/cache/*cacheName*

https://api.cache.*region*.prod.a.momentohq.com/topics/*cacheName*/*topicName*

Tipi di autorizzazione supportati:

APIChiave

Parametri di autorizzazione aggiuntivi necessari:

Type	Chiave	Valore
Header	Autorizzazione	*MOMENTO_API_KEY*

Documentazione Momento:

Momento+ Amazon EventBridge

Utilizzo degli argomenti Momento API

APIriferimento per Momento Cache

APIOperazioni di uso comune:

Per le cache:`PUT`, `DELETE`

Per argomenti: `POST`

Informazioni aggiuntive:

Quando aggiorni o elimini una cache, includi i seguenti parametri della stringa di query nella configurazione dell'obiettivo della regola:

- La chiave che vuoi aggiornare nella cache di Momento
- Il Time-To-Live (TTL) per l'elemento della cache

Ad esempio, se l'evento di input includeva un `details` campo con questi valori:

```
key: $.details.key ttl_seconds: $.details.ttl_seconds
```

New Relic

APIendpoint di invocazione di destinazione URL

Per ulteriori informazioni, consulta Our EU and US region data centers.

Eventi

Stati Uniti— accounts/ https://insights-collector.newrelic.com/v1/*YOUR_NEW_RELIC_ACCOUNT_ID*/eventi

UE— https://insights-collector.eu01.nr-data.net/v1/accounts/*YOUR_NEW_RELIC_ACCOUNT_ID*/eventi

Metriche

Stati Uniti— v1 https://metric-api.newrelic.com/metric/

UE— v1 https://metric-api.eu.newrelic.com/metric/

Log

Stati Uniti— v1 https://log-api.newrelic.com/log/

UE— v1 https://log-api.eu.newrelic.com/log/

Tracce

Stati Uniti— v1 https://trace-api.newrelic.com/trace/

UE— v1 https://trace-api.eu.newrelic.com/trace/

Tipi di autorizzazione supportati

APIChiave

Documentazione di New Relic

Metrico API

Evento API

Registro API

Traccia API

APIOperazioni di uso comune

Metrico API

Evento API

Registro API

Traccia API

Informazioni aggiuntive

- Limiti metrici API
- Limiti degli eventi API
- APILimiti di registro
- APILimiti di tracciamento

Operata

APIendpoint di invocazione della destinazione: URL

- https://api.operata.io/v2/aws/events/contact-record

Tipi di autorizzazione supportati:

- Base

Parametri di autorizzazione aggiuntivi necessari:

- Nessuno

Documentazione di Operata:

- Come posso creare, visualizzare, modificare e revocare i token? API
- AWS Integrazione di Operata tramite Amazon EventBridge Scheduler Pipes

Operazioni di uso API comune:

- `POST https://api.operata.io/v2/aws/events/contact-record`

Informazioni aggiuntive:

- `username`è l'ID del gruppo Operata e la password è il tuo API token.

Salesforce

APIendpoint di invocazione di destinazione URL

- Oggetto: https:// *`myDomainName`*.my.salesforce.com/services/data/*`versionNumber`*/oggetti/*`SobjectEndpoint`*/*

Eventi della piattaforma personalizzati: https://*myDomainName*.my.salesforce.com/services/data/*versionNumber*/oggetti/*customPlatformEndpoint*/*

Per un elenco completo degli endpoint, vedi Reference Salesforce API

Tipi di autorizzazione supportati

OAuthcredenziali del client

OAUTHi token 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://*MyDomainName*-- *SandboxName*.my.salesforce.com/services /oauth2/token
- 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

RESTAPIGuida per gli sviluppatori

APIOperazioni 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 API destinazione e una regola a cui indirizzare le informazioniSalesforce, consulta???.

Slack

APIendpoint di invocazione di destinazione URL

Per un elenco di endpoint e altre risorse, consulta Using the Slack Web API

Tipi di autorizzazione supportati

OAuth2.0

OAUTHi token vengono aggiornati quando viene restituita una risposta 401 o 407.

Quando crei un'Slackapplicazione e la installi nel tuo spazio di lavoro, verrà creato un token OAuth bearer per tuo conto da utilizzare per autenticare le chiamate tramite la connessione di destinazione. API

Parametri di autorizzazione aggiuntivi necessari

Non applicabile

Documentazione di Slack

Basic app setup

Installazione con OAuth

Retrieving messages

Invio di messaggi

Sending messages using Incoming Webhooks

APIOperazioni di uso comune

https://slack.com/api/chiacchierare postMessage

Informazioni aggiuntive

Quando si configura la EventBridge regola, ci sono due configurazioni da evidenziare:

- Includi un parametro di intestazione che definisca il tipo di contenuto come "application/json;charset=utf-8".
- Utilizzate un trasformatore di input per mappare l'evento di input all'output previsto per il SlackAPI, vale a dire assicuratevi che il payload inviato a contenga coppie Slack API chiave/valore «canale» e «testo».

Shopify

APIendpoint di invocazione di destinazione URL

Per un elenco di endpoint e altre risorse e metodi, consulta Endpoints and requests

Tipi di autorizzazione supportati

OAuth, API Chiave

> **Note**
>
> OAUTHi token 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 di uso comune API

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

ShopifyAmministratore API

Splunk

APIendpoint di invocazione di destinazione URL

https://*SPLUNK_HEC_ENDPOINT*:*optional_port*/services/collector/raw

Tipi di autorizzazione supportati

Di base, chiave API

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di Splunk

Per entrambi i tipi di autorizzazione, è necessario un ID HEC token. Per ulteriori informazioni, consulta Configurare e utilizzare HTTP Event Collector nel Splunk Web.

Operazioni di uso API comune

POSThttps://*SPLUNK_HEC_ENDPOINT*:*optional_port*/servizi/collector/raw

Informazioni aggiuntive

APIChiave: quando si configura l'endpoint per EventBridge, il nome della API chiave 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

APIendpoint di invocazione di destinazione URL

HTTPGli endpoint Log e Metric Source URLs saranno diversi per ogni utente. Per ulteriori informazioni, consulta HTTPLogs and Metrics Source.

Tipi di autorizzazione supportati

Sumo Logicnon richiede l'autenticazione sulle proprie HTTP fonti perché è presente una chiave unica incorporata in. URL Per questo motivo, dovresti assicurarti di trattarlo URL come un segreto.

Quando si configura la EventBridge API destinazione, è richiesto un tipo di autorizzazione. Per soddisfare questo requisito, selezionate API Key e assegnategli un nome chiave «dummy-key» e un valore 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 gli endpoint EventBridge API Destinations e Sumo Logic HTTP Log and Metric Source.

- Per effettuare la registrazione e creare un'istanza Sumo Logic gratuita, consulta Start your free trial today.
- Per ulteriori informazioni sull'utilizzoSumo Logic, consulta HTTPLogs and Metrics Source.

Operazioni di uso comune API

POST https://endpoint4.collection.us2.sumologic.com/receiver/v1/http/*UNIQUE_ID_PER_COLLECTOR*

Informazioni aggiuntive

Nessuno

TriggerMesh

APIendpoint di invocazione di destinazione URL

Usa le informazioni nell'HTTPargomento Event Source for per formulare l'endpoint. URL Un endpoint URL include il nome della fonte 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

Fonte di eventi per HTTP

APIOperazioni di uso comune

Non applicabile

Informazioni aggiuntive

Nessuno

Zendesk

APIendpoint di invocazione di destinazione URL

https://developer.zendesk.com/rest_api/docs/supporto/ticket

Tipi di autorizzazione supportati

Di base, chiave API

Parametri di autorizzazione aggiuntivi necessari

Nessuno

Documentazione di Zendesk

Security and Authentication

APIOperazioni comunemente utilizzate

POSThttps://*your_Zendesk_subdomain*/api/v2/biglietti

Informazioni aggiuntive

APIle richieste vengono conteggiate ai EventBridge limiti di Zendesk. API 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 API chiave anziché l'autenticazione di base con le credenziali di accesso.

Connessioni per destinazioni HTTP endpoint in Amazon EventBridge

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'HTTPendpoint.

Usa connessioni con:

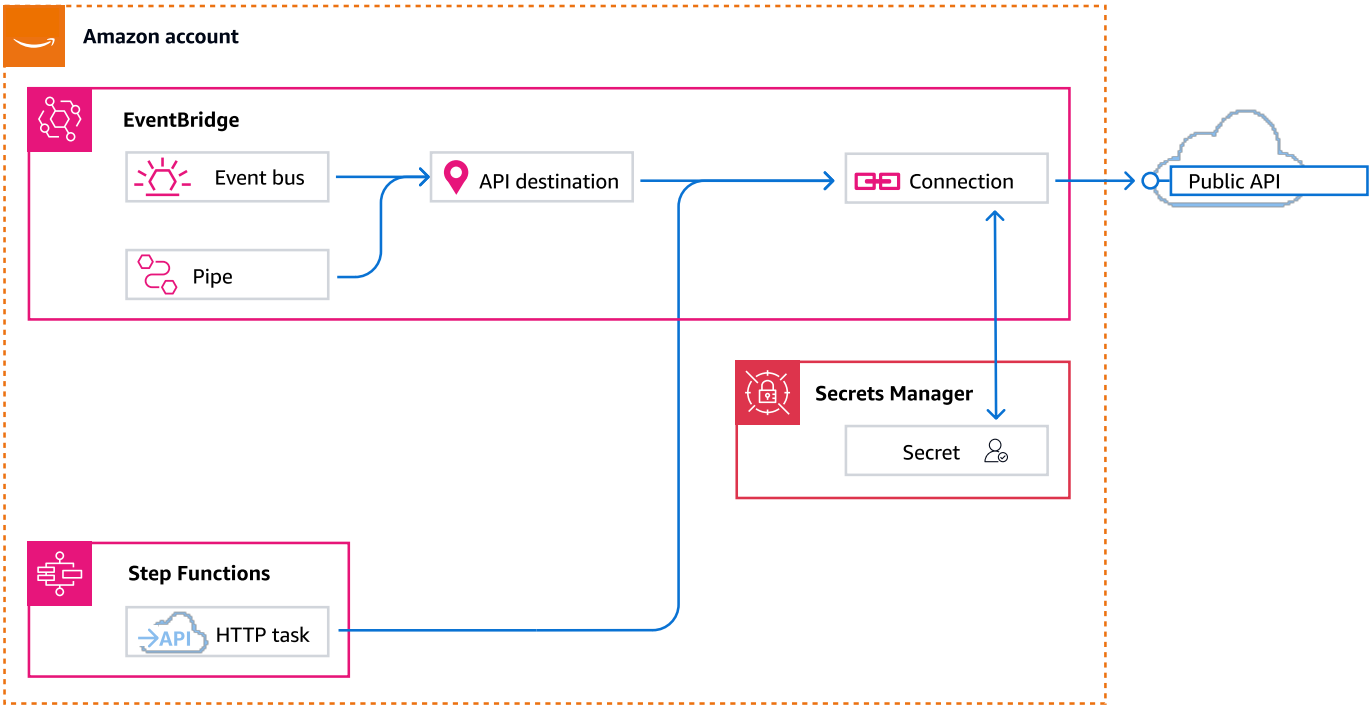
- APIdestinazioni

  Quando si crea una API destinazione, si specifica una connessione da utilizzare per tale destinazione. Puoi scegliere una connessione esistente dal tuo account o creare una connessione quando crei una API destinazione.

- HTTPattività in AWS Step Functions

  Un HTTP Task è un tipo di stato del flusso di lavoro Task che ti consente di chiamare qualsiasi terza parte pubblicaAPI, come Salesforce e Stripe, nei tuoi flussi di lavoro. L'attività utilizza una connessione per specificare il tipo di autorizzazione e le credenziali da utilizzare per l'autorizzazione della terza parte. API

  Per ulteriori informazioni, consulta Chiama terze parti APIs nei flussi di lavoro Step Functions nella Step Functions User Guide.



Metodi di autorizzazione per le connessioni

EventBridge le connessioni supportano i seguenti metodi di autorizzazione:

- Base
- APIChiave

  Per l'autorizzazione di base e tramite API chiave, EventBridge compila automaticamente le intestazioni di autorizzazione richieste.
- OAuth

  Per quanto riguarda OAuth l'autorizzazione, scambia EventBridge anche l'ID cliente e il segreto con un token di accesso e quindi lo gestisce in modo sicuro.

  OAUTHi token 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 HTTP endpoint se l'autorizzazione per l'endpoint è la stessa.

Quando si crea una connessione e si aggiungono parametri di autorizzazione, EventBridge crea un ingresso segreto. AWS Secrets Manager Il costo dell'archiviazione e dell'accesso al segreto di Secrets Manager è incluso nel costo per l'utilizzo di una API destinazione. Per ulteriori informazioni sulle migliori pratiche per l'utilizzo dei segreti con le API destinazioni, consulta AWS: :Events:: ApiDestination nella Guida per l'CloudFormation 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 HTTP endpoint in EventBridge

Per creare una connessione da utilizzare con gli HTTP endpoint 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 a sinistra, in Integrazione, scegli Connessioni.
3. Scegli Crea connessione.
4. Nella pagina Crea connessione, immetti un nome per la connessione in Nome connessione.
5. Immetti una descrizione per la connessione in Descrizione.
6. Per Tipo di autorizzazione, seleziona il tipo di autorizzazione da utilizzare per autorizzare le connessioni all'HTTPendpoint specificato per la API destinazione che utilizza questa connessione. Esegui una di queste operazioni:
   - Scegli Basic (nome utente/password), quindi inserisci il nome utente e la password da utilizzare per l'autorizzazione con l'endpoint. HTTP
   - Scegli Credenziali OAuth client, quindi inserisci l'endpoint di autorizzazione, il HTTPmetodo, l'ID client e il segreto del cliente da utilizzare per l'autorizzazione con l'endpoint.

     In Parametri OAuth Http, aggiungi eventuali parametri aggiuntivi 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 APIla chiave, quindi inserisci il nome della API chiave e il valore associato da utilizzare per l'autorizzazione della API chiave.

  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.

7. Scegli Create (Crea) .

Modifica delle connessioni tramite la EventBridge console

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 a sinistra, in Integrazione, scegli Connessioni.
3. Nella tabella Connessioni, scegli la connessione da modificare.
4. Nella pagina Dettagli di connessione, scegli Modifica.
5. Aggiorna i valori per la connessione, quindi scegli Aggiorna.

Annullare l'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 HTTP endpoint 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 a sinistra, in Integrazione, scegli Connessioni.
3. Nella tabella Connessioni, scegli la connessione.
4. Nella pagina Dettagli di connessione, scegli Rimuovi autorizzazione.
5. 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.

Eliminazione delle connessioni tramite la console EventBridge

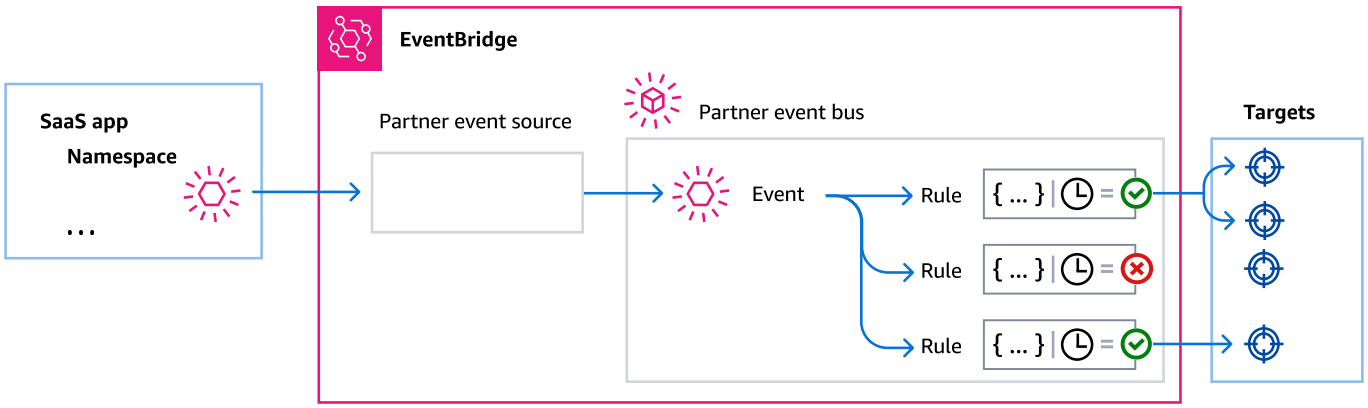
È possibile eliminare una connessione se non la si utilizza più.

Per eliminare una connessione

1. Accedi AWS utilizzando un account con le autorizzazioni necessarie per gestire EventBridge e aprire la EventBridge console.
2. Nel riquadro di navigazione a sinistra, in Integrazione, scegli Connessioni.
3. Nella tabella Connessioni, scegli la connessione.
4. Nella pagina dei dettagli della connessione, scegli Elimina.

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. Una fonte di eventi partner è una risorsa creata da un partner che puoi quindi accettare come fonte di eventi. Per accettare l'origine dell'evento partner, è necessario creare un bus di eventi personalizzato e abbinarlo all'origine dell'evento partner.



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

Argomenti

- Integrazioni di partner SaaS supportate
- Disponibilità regionale delle integrazioni dei partner SaaS
- Configurazione di Amazon EventBridge per ricevere eventi da un'integrazione SaaS
- Ricezione di eventi SaaS dalla AWS Lambda funzione URLs in Amazon EventBridge
- Ricezione di eventi da Salesforce Amazon EventBridge

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
- Partner venditore Amazon API

Disponibilità regionale delle integrazioni dei partner SaaS

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)
ap-south-1	Asia Pacifico (Mumbai)

Codice	Nome
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 () UAE
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.

   > **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.

Ricezione di eventi SaaS dalla AWS Lambda funzione URLs in Amazon EventBridge

> **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 la funzione URLs Lambda.

Il tuo bus di EventBridge eventi Amazon può utilizzare una AWS Lambda funzione URL creata da un AWS CloudFormation modello per ricevere eventi dai provider SaaS supportati. Con functionURLs, 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 della funzione Lambda URLs aumenterà i costi mensili. Per ulteriori informazioni, consulta 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.

I seguenti provider SaaS sono attualmente disponibili per l'uso con la funzione EventBridge Lambda: URLs

- GitHub
- Twilio

Argomenti

- 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

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 che utilizzano Lambda fURLs, 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 bus di eventi riceve i dati dalla funzione Lambda URL fornita a. GitHub Puoi anche creare un router di eventi selezionando Nuovo bus di eventi.
6. Nel Passaggio 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 IAM risorse.
    - Riconosco che AWS CloudFormation potrebbe creare IAM risorse 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 URL che la funzione Lambda creata 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 innanzitutto 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 che utilizzano Lambda fURLs, 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 bus di eventi riceve i dati dalla funzione Lambda URL fornita a. Twilio Puoi anche creare un router di eventi selezionando Nuovo bus di eventi.
6. Nel Passaggio 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 IAM risorse.
    - Riconosco che AWS CloudFormation potrebbe creare IAM risorse 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 la funzione LambdaURL, devi darla a Twilio in modo che i dati degli eventi possano essere inviati. Per ulteriori informazioni, consulta Configure your public URL with Twilio nella Twilio documentazione.

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 AWS CloudFormation stack non sono sincronizzati. 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 in 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 che hai creato nel passaggio 1 e scegli Avanti.
8. Scegli Next (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 in 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 che hai creato nel passaggio 1 e scegli Avanti.
8. Scegli Next (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 in AWS CloudFormation /cloudformation. https://console.aws.amazon.com

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 CloudFormation bus 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.

Servizio	Codice di errore	Descrizione
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

Usa i GitHub webhook API per verificare lo stato di consegna di ogni chiamata tramite webhook e reinviare 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 Webhook (HTTPcallback): Connection Overrides nella documentazione. Twilio

Ricezione di eventi da Salesforce Amazon EventBridge

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 API destinazioni Salesforce che utilizzano. Una volta inviato a Salesforce, l'evento può essere elaborato da flussi o trigger Apex. Per ulteriori informazioni sulla configurazione di una Salesforce API destinazione, vedere???.

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

Fase 1: Configurare 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

Fase 2: Attiva Salesforce il codice sorgente dell'evento per i partner nella EventBridge console e avvia il relay 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. Scegli Next (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. Scegli Next (Successivo).
18. (Facoltativo) Configura i filtri per i campi di dati in Amazon AppFlow.
19. Scegli Next (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 Create (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.

EventBridge Tutorial Amazon

EventBridge si integra con una serie di AWS servizi e partner SaaS. Questi tutorial sono progettati per aiutarti a familiarizzare con le basi EventBridge e con il modo in cui può far parte della tua architettura serverless.

Nozioni di base

I seguenti tutorial ti aiutano a esplorare le funzionalità e a utilizzarle. EventBridge

- Archiviazione e riproduzione di eventi
- Creazione di un'applicazione di esempio
- Dowload delle associazioni di codice
- Utilizzo del trasformatore di input

AWS tutorial

Amazon EventBridge collabora con altri AWS servizi per elaborare eventi o richiamare una AWS risorsa come obiettivo di una regola. I seguenti tutorial mostrano come integrarsi EventBridge con altri servizi. AWS

- Registrazione degli stati di un gruppo con dimensionamento automatico
- Crea una regola per le AWS API chiamate tramite CloudTrail
- Registra gli stati delle EC2 istanze Amazon
- Registra le operazioni sugli oggetti Amazon S3
- Invia eventi utilizzando schemi
- Creazione di una regola pianificata
- Invia un'e-mail quando si verificano degli eventi
- Crea una regola pianificata per le funzioni Lambda

Tutorial per provider SaaS

EventBridge può lavorare direttamente con le applicazioni e i servizi dei partner SaaS per inviare e ricevere eventi. I seguenti tutorial mostrano come integrarsi con i partner EventBridge SaaS.

- Invia eventi a Datadog

- Invia eventi a Salesforce
- Invia eventi a Zendesk

Tutorial: crea un' EventBridge applicazione Amazon di esempio

È possibile utilizzare EventBridge per indirizzare gli eventi a funzioni Lambda specifiche utilizzando le regole.

In questo tutorial, utilizzerai Node.js e il AWS CLI codice nel GitHubrepository per creare quanto segue:

- Una AWS Lambdafunzione che produce eventi per le ATM transazioni bancarie.
- Tre funzioni Lambda da utilizzare come obiettivi di una EventBridge regola.
- La regola che instrada gli eventi creati alla funzione a valle corretta in base a un modello di eventi.

Questo esempio utilizza AWS SAM modelli per definire le EventBridge regole. Per ulteriori informazioni sull'utilizzo dei AWS SAM modelli, EventBridge consulta???.

Nel repository, la atmProducersottodirectory contiene`handler.js`, che rappresenta gli eventi che producono il ATM servizio. Questo codice è un gestore Lambda scritto in Node.js e pubblica eventi EventBridge tramite l'AWS SDKutilizzo di 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 nel JavaScript modo seguente:

```
{
  // 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'ubicazioneATM, l'importo, la banca partner e il risultato della transazione.

Il `handler.js` file nella atmConsumersottodirectory 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 `console.log` istruzioni in Amazon CloudWatch Logs. Le funzioni consumer operano indipendentemente dal produttore e non conoscono l'origine degli eventi.

La logica di routing è contenuta nelle EventBridge regole distribuite dal modello dell'applicazione. AWS SAM 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 JSON oggetti 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-"
    }]
```

```
  }
}
```

Fasi:

- 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 AWS account. Crea un AWS account se non ne hai già uno.
- AWS CLI installato. Per installare AWS CLI, vedere Installazione, aggiornamento e disinstallazione della 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 AWS CLI e Git per creare le AWS risorse di cui avrai bisogno.

Per creare l'applicazione

1. Esegui l'accesso a AWS.
2. Installa Git e installalo 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`**, immettere `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`**, immettere `samconfig.toml`.

   g. In **`SAM configuration environment`**, immettere `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 atmProducerFnfunzione, quindi scegli Azioni, 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 CloudWatch i registri.

Per verificare i log

1. Apri la CloudWatch console nella stessa regione in cui hai eseguito l' AWS SAM applicazione.
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 da. 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 e seleziona il gruppo di log contenente atmConsumerCase3. 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 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 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 il/i gruppo/i di CloudWatch log di Logs

1. Apri la console CloudWatch.
2. Scegli Log, Gruppi di log.
3. Seleziona il gruppo di log.
4. Scegli Operazioni > Elimina gruppo/i di log.
5. Scegli Delete (Elimina).

Tutorial: archivia e riproduci gli eventi in Amazon EventBridge

È possibile utilizzare EventBridge per indirizzare gli eventi a AWS Lambdafunzioni specifiche utilizzando le regole.

In questo tutorial, creerai una funzione da usare come obiettivo per la EventBridge regola utilizzando la console Lambda. Quindi, creerai un archivio e una regola per archiviare gli eventi di test utilizzando la EventBridge console. Una volta che in quell'archivio sono presenti eventi, li riprodurrai.

Fasi:

- Fase 1: Creazione di una funzione Lambda
- Passaggio 2: creare l'archivio
- Passaggio 3: creare una regola
- Passaggio 4: inviare eventi di test
- Passaggio 5: riprodurre gli eventi
- Fase 6: eliminare le risorse

Fase 1: Creazione di una funzione Lambda

Innanzi tutto, crea una funzione Lambda per registrare gli eventi.

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 `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 JavaScript codice esistente con il seguente codice:

```
'use strict';

exports.handler = (event, context, callback) => {
```

```
    console.log('LogScheduledEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Seleziona Deploy (Implementa).

Passaggio 2: creare l'archivio

A questo punto, devi creare l'archivio che conterrà tutti gli eventi di test.

Per creare un archivio

1. Apri la EventBridge console Amazon 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. 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 `ARTestRule`.

   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 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. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "detail-type": [
    "customerCreated"
  ]
}
```

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 `LogScheduledEvent`.
14. Seleziona Successivo.
15. Seleziona 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. Apri la EventBridge console Amazon 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. Apri la EventBridge console Amazon 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.

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 AWS account.

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 gli EventBridge archivi

1. Apri la pagina Archivi della EventBridge console.
2. Seleziona l'archivio creato.
3. Scegli Elimina.
4. Immetti il nome dell'archivio e scegli Elimina.

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. Scegli Delete (Elimina).

Tutorial: scarica le associazioni di codice per gli schemi di eventi in EventBridge

Puoi generare associazioni di codice per schemi di eventi per velocizzare lo sviluppo di Golang, Java, Python e. TypeScript È possibile ottenere associazioni di codice per i AWS servizi esistenti, gli schemi creati dall'utente e per gli schemi generati in base agli eventi su un bus di eventi. Puoi generare associazioni di codice per uno schema mediante uno dei seguenti elementi:

- EventBridge console
- EventBridge registro dello schema API
- Hai IDE un AWS kit di strumenti

In questo tutorial si generano e si scaricano associazioni di codice da uno EventBridge schema per gli eventi di un servizio. 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. Seleziona la scheda Registro schemi di eventi AWS .
4. Trova lo schema per il AWS servizio per il quale desideri creare associazioni di codice sfogliando il registro degli 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 README file che spiega come configurare le dipendenze del pacchetto in vari framework.

Utilizzate queste associazioni di codice nel vostro codice per creare rapidamente applicazioni che utilizzano questo evento. EventBridge

Tutorial: usa i trasformatori di input per trasformare gli eventi in EventBridge

Puoi utilizzare il trasformatore di input EventBridge per personalizzare il testo di un evento prima di inviarlo alla destinazione di una regola.

A tale scopo, definite i JSON percorsi a partire dall'evento e assegnate i relativi output a diverse variabili. 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 specificate una variabile in modo che corrisponda a un JSON percorso 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 `type` variabile sul percorso $.detail-type JSON dell'evento. <type>Quindi EventBridge inserisce la variabile nel modello di input «Questo evento è stato». Il risultato è il seguente SNS messaggio Amazon.

```
"This event was of customerCreated type."
```

Fasi:

- Passaggio 1: creare un SNS argomento Amazon
- Passaggio 2: creare un SNS abbonamento Amazon
- Passaggio 3: creare una regola
- Passaggio 4: inviare eventi di test
- Passaggio 5: verificare il corretto completamento del tutorial
- Fase 6: eliminare le risorse

Passaggio 1: creare un SNS argomento Amazon

Crea un argomento da cui ricevere gli eventi EventBridge.

Per creare un argomento

1. Apri la SNS console Amazon su 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 un SNS abbonamento Amazon

Creazione di una sottoscrizione per ricevere e-mail con le informazioni trasformate.

Creazione di una sottoscrizione

1. Apri la SNS console Amazon su https://console.aws.amazon.com/sns/v3/home.
2. Nel riquadro di navigazione scegliere Subscriptions (Iscrizioni).
3. Scegli Crea sottoscrizione.
4. Per Argomento ARN, scegli l'argomento che hai creato nel passaggio 1. Per questo tutorial, scegli eventbridge-IT-test.
5. Per Protocollo, scegli E-mail.
6. Per Endpoint, immettere il proprio indirizzo e-mail.
7. Scegli Crea sottoscrizione.
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. 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 `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. Seleziona Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "detail-type": [
    "customerCreated"
  ]
}
```

10. Seleziona Successivo.
11. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
12. Per Seleziona un obiettivo, scegli l'SNSargomento dall'elenco a discesa.
13. Per Argomento, seleziona l'SNSargomento Amazon che hai creato nel 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. Seleziona Successivo.
16. Seleziona Successivo.
17. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Passaggio 4: inviare eventi di test

Ora che hai impostato l'SNSargomento e la regola, invieremo gli eventi di test per assicurarci che la regola funzioni correttamente.

Per inviare eventi di test (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. 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 AWS notifiche che corrisponde all'output previsto, hai completato con successo il tutorial.

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 AWS account.

Per eliminare l'argomento SNS

1. Apri la pagina Argomenti della SNS console.
2. Seleziona l'argomento creato.
3. Scegli Elimina.
4. Specificare **`delete me`**.
5. Scegli Elimina.

Per eliminare l'SNSabbonamento

1. Apri la pagina Abbonamenti della SNS console.
2. Seleziona la sottoscrizione creata.
3. Scegliere Delete (Elimina).
4. Scegliere Delete (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: Registra lo stato di un gruppo di Auto Scaling usando EventBridge

Puoi eseguire una AWS Lambdafunzione che registra un evento ogni volta che un gruppo di Auto Scaling avvia o termina un'istanza EC2 Amazon che indica se un evento ha avuto successo.

Per informazioni su altri scenari che utilizzano gli eventi di Amazon EC2 Auto Scaling, consulta Use EventBridge to handle Auto Scaling events nella Amazon Auto Scaling User EC2 Guide.

In questo tutorial, crei una funzione Lambda e crei una regola nella EventBridge console che richiama tale funzione quando un gruppo Amazon Auto EC2 Scaling avvia o termina un'istanza.

Fasi:

- Prerequisiti
- Fase 1: Creazione di una funzione 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

Prerequisiti

Per completare questo tutorial, avrai bisogno delle seguenti risorse:

- Un gruppo con dimensionamento automatico. Per ulteriori informazioni sulla creazione di un gruppo, consulta Creazione di un gruppo Auto Scaling utilizzando una configurazione di avvio nella Amazon Auto EC2 Scaling User Guide.

Fase 1: Creazione di 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 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. Immetti un nome per la funzione Lambda. Ad esempio, denomina la funzione `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).

Fase 2: Creazione di 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. 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 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. 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 `LogAutoScalingEvent`.
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 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 EC2 console Amazon 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 Dettagli, seleziona 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. 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. 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 CloudWatch log, significa che hai completato con successo questo tutorial. Se l'evento non è presente nei 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 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 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. Scegli Delete (Elimina).

Tutorial: crea una EventBridge regola che reagisca alle AWS API chiamate tramite CloudTrail

Puoi utilizzare EventBridge le regole di Amazon per reagire alle API chiamate 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 percorso AWS CloudTrail
- 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 percorso AWS CloudTrail

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. Per l'AWS KMS alias, digita un alias per la KMS chiave.
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 delle API chiamate.

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 interrompi un'istanza AmazonEC2.

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. Per Event source, seleziona EC2dall'elenco a discesa.
    b. Per Tipo di evento, seleziona AWS APIChiama tramite 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 tua regola interrompendo un'EC2istanza Amazon utilizzando la EC2 console Amazon. 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 EC2 console Amazon 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 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. 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'EC2istanza Amazon utilizzando EventBridge

Puoi creare una AWS Lambdafunzione che registra una modifica di stato per un'EC2istanza Amazon. 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'EC2istanza Amazon.

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. Per Event source, seleziona EC2dall'elenco a discesa.
    b. Per Tipo di evento, scegli Notifica di modifica dello stato dell'EC2istanza 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 tua regola interrompendo un'EC2istanza Amazon utilizzando la EC2 console Amazon. Attendi qualche minuto che l'istanza si fermi, quindi controlla i AWS Lambda parametri sulla CloudWatch console per verificare che la funzione funzioni.

Test della regola arrestando un'istanza

1. Apri la EC2 console Amazon 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 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 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 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. Scegli Delete (Elimina).

Tutorial: registra le operazioni a livello di oggetto di Amazon S3 utilizzando EventBridge

Puoi registrare le API operazioni a livello di oggetto sui tuoi bucket Amazon S3. Prima che Amazon EventBridge possa corrispondere a questi eventi, AWS CloudTraildevi impostare e configurare un percorso per ricevere questi eventi.

In questo tutorial, crei un CloudTrail trail, crei una AWS Lambdafunzione e quindi crei una regola nella EventBridge console che richiama quella funzione in risposta a un evento relativo ai dati S3.

Fasi:

- Passaggio 1: configura il tuo percorso 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
- Fase 6: eliminare le risorse

Passaggio 1: configura il tuo percorso AWS CloudTrail

Per registrare gli eventi relativi ai dati di un bucket S3 su AWS CloudTrail e EventBridge, devi prima creare un trail. Un trail acquisisce le API chiamate e gli eventi correlati nel tuo account e quindi invia i file di registro a un bucket S3 da te 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 all' CloudTrail 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. Per l'AWS KMS alias, digita un alias per la KMS chiave.
6. Seleziona 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. Seleziona 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 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 `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. Seleziona Deploy (Implementa).

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. 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 Simple Storage Service (S3) dall'elenco a discesa.
   b. Per Tipo di evento, seleziona APIChiamata a livello di oggetto tramite CloudTrail dall'elenco a discesa.
   c. Scegli Operazioni specifiche, quindi scegli. 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. 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 `LogS3DataEvents` che hai creato in Passaggio 1.
14. Seleziona Successivo.
15. Seleziona 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. Apri la CloudWatch console 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 CloudTrail i log nel bucket S3 che hai specificato per il percorso. Per ulteriori informazioni, consulta Ottenere e visualizzare i file di CloudTrail registro nella Guida per l'AWS CloudTrail utente.

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 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. 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: invio di eventi ad Amazon Kinesis utilizzando schemi EventBridge

Puoi inviare eventi di AWS API chiamata EventBridge a 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'EC2istanza Amazon si interrompe.

Fasi:

- Prerequisiti
- Passaggio 1: creare un flusso Amazon Kinesis
- Fase 2: Creazione di una regola
- Fase 3: Test della regola
- Passaggio 4: verificare l'invio dell'evento
- Passaggio 5: eliminare le risorse

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 tuo stream quando interrompi un'EC2istanza Amazon.

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 per i 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@ EC2InstanceStateChangeNotification 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 lo schema di eventi che hai creato, scegli Genera pattern di eventi in JSON

   EventBridge visualizza lo schema degli eventi inJSON:

```
{
  "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 tua regola, interrompi un'EC2istanza Amazon. Attendi qualche minuto che l'istanza si fermi, quindi controlla le CloudWatch metriche per verificare che la funzione funzioni.

Test della regola arrestando un'istanza

1. Apri la EC2 console Amazon 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 recuperare il record dallo stream e 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. Utilizzate un decoder Base64 per decodificare i dati in modo da poter verificare che si tratti dell'evento che è stato inviato allo stream nel modulo. 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 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 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: Creare una regola pianificata in EventBridge

Puoi eseguire EventBridge le regole in base a una pianificazione. In questo tutorial, crei uno snapshot di un volume Amazon Elastic Block Store (AmazonEBS) esistente secondo 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.

Fasi:

- 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 che viene eseguita secondo una pianificazione in Amazon EventBridge.

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. 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 Schedule (Pianifica).
7. Seleziona 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. Seleziona Successivo.
10. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
11. Per Seleziona una destinazione, scegli EBSCrea istantanea dall'elenco a discesa.
12. Per Volume ID, inserisci l'ID del EBS volume Amazon.
13. In Ruolo di esecuzione, scegli Crea un nuovo ruolo per questa risorsa specifica.
14. Seleziona Successivo.
15. Seleziona 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 EC2 console Amazon 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 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. Scegli Delete (Elimina).

Tutorial: invia un'e-mail quando si verificano eventi utilizzando Amazon EventBridge

Puoi inviare notifiche e-mail quando gli oggetti Amazon Simple Storage Service (Amazon S3) vengono creati utilizzando Amazon e Amazon EventBridge . SNS In questo tutorial, creerai un SNS argomento e un abbonamento. Quindi, creerai una regola nella EventBridge console che invia eventi a quell'argomento quando vengono `Object Created` ricevuti eventi Amazon S3.

Fasi:

- Prerequisiti
- Passaggio 1: creare un SNS argomento Amazon
- Passaggio 2: creare un SNS abbonamento Amazon
- Passaggio 3: creare una regola
- Passaggio 4: testare la regola
- Passaggio 5: eliminare le risorse

Prerequisiti

Per ricevere eventi Amazon S3 EventBridge in, devi EventBridge abilitarli nella console Amazon S3. Questo tutorial presuppone EventBridge che sia abilitato. Per ulteriori informazioni, consulta Abilitare Amazon EventBridge nella console S3.

Passaggio 1: creare un SNS argomento Amazon

Crea un argomento da cui ricevere gli eventi EventBridge.

Per creare un argomento

1. Apri la SNS console Amazon su 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 un SNS abbonamento Amazon

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 SNS console Amazon su https://console.aws.amazon.com/sns/v3/home.
2. Nel riquadro di navigazione scegliere Subscriptions (Iscrizioni).
3. Scegli Crea sottoscrizione.
4. Per Argomento ARN, scegli l'argomento che hai creato nel passaggio 1. Per questo tutorial, scegli eventbridge-test.
5. Per Protocollo, scegli E-mail.
6. Per Endpoint, immettere il proprio indirizzo e-mail.
7. Scegli Crea sottoscrizione.
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. 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 `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. Seleziona Successivo.
8. Per Event source, scegli AWS eventi o eventi per i EventBridge partner.
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. Seleziona Successivo.
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. Per Seleziona un obiettivo, scegli l'SNSargomento dall'elenco a discesa.
14. Per Argomento, seleziona l'SNSargomento Amazon che hai creato nella sezione Passaggio 1: creazione di un SNS argomento. In questo esempio, seleziona `eventbridge-test`.
15. Seleziona Successivo.
16. Seleziona Successivo.
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 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 conservarle. Eliminando AWS le risorse che non utilizzi più, eviti addebiti inutili sul tuo account. AWS

Per eliminare l'argomento SNS

1. Apri la pagina Argomenti della SNS console.
2. Seleziona l'argomento creato.
3. Scegli Elimina.
4. Specificare **`delete me`**.
5. Scegli Elimina.

Per eliminare l'SNSabbonamento

1. Apri la pagina Abbonamenti della SNS console.
2. Seleziona la sottoscrizione creata.
3. Scegliere Delete (Elimina).
4. Scegliere Delete (Elimina).

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. Scegli Delete (Elimina).

Tutorial: creare una regola EventBridge pianificata per AWS Lambda le funzioni

Puoi configurare una regola per l'esecuzione di una funzione AWS Lambda in basa a una pianificazione. Questo tutorial mostra come usare AWS Management Console o the AWS CLI per creare la regola. Se desideri utilizzare la versione AWS CLI ma non l'hai ancora installata, consulta Installazione, aggiornamento e disinstallazione della AWS CLI versione 2.

Per quanto riguarda le pianificazioni, EventBridge non fornisce una precisione di secondo livello nelle espressioni di pianificazione. La risoluzione più alta che utilizza un'espressione Cron è un minuto. A causa della natura distribuita dei servizi di destinazione EventBridge e dei servizi di destinazione, può verificarsi un ritardo di diversi secondi tra l'attivazione della regola pianificata e il momento in cui il servizio di destinazione esegue la risorsa di destinazione.

Fasi:

- Fase 1: Creazione di 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

Fase 1: Creazione di una funzione Lambda

Crea una funzione Lambda per registrare gli eventi pianificati.

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 `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.

È possibile utilizzare la console o il AWS CLI per creare la regola. Per utilizzare il AWS CLI, devi prima concedere alla regola il permesso di richiamare la tua funzione Lambda. Puoi quindi creare la regola e aggiungere la funzione Lambda come target.

Per creare una regola (console)

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 Schedule (Pianifica).
7. Seleziona 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. Seleziona 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. Seleziona Successivo.
14. Seleziona 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 EventBridge service principal (`events.amazonaws.com`) l'autorizzazione a eseguire la regola, usa il `add-permission` comando.

```
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. Apri la CloudWatch console 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 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: invia eventi a Datadog da Amazon EventBridge

È possibile utilizzare EventBridge per indirizzare gli eventi a servizi di terze parti, ad esempio. Datadog

In questo tutorial, utilizzerai la EventBridge console per creare una connessione versoDatadog, una APIdestinazione che punti e una regola verso cui indirizzare gli eventiDatadog. Datadog

Fasi:

- Prerequisiti
- Passaggio 1: creare una connessione
- Fase 2: Creare una API destinazione
- 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 Datadog.
- Una DatadogAPIchiave.
- Un EventBridge bucket Amazon Simple Storage Service (Amazon S3) abilitato per Amazon.

Passaggio 1: creare una connessione

Per inviare eventi aDatadog, devi prima stabilire una connessione a. Datadog API

Per creare la connessione

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli APIIe destinazioni.
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. Per Tipo di autorizzazione, scegli APIla chiave.
6. Per il nome della API chiave, inserisci**`DD-API-KEY`**.
7. Per Value, incolla la tua API chiave Datadog segreta.
8. Scegli Crea.

Fase 2: Creare una API destinazione

Ora che hai creato la connessione, creerai la API destinazione da utilizzare come destinazione della regola.

Per creare la API destinazione

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli APIle destinazioni.
3. Scegli Crea API destinazione.
4. Inserisci un nome e una descrizione per la API destinazione. In questo esempio, immetti **`DatadogAD`** come nome e **`Datadog API Destination`** come descrizione.
5. Per l'endpoint di API destinazione, immettere**`https://http-intake.logs.datadoghq.com/api/v2/logs`**.
6. Per HTTPmetodo, scegliete 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. 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 **`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. Seleziona Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "source": ["aws.s3"]
}
```

10. Seleziona Successivo.
11. Per i tipi di Target, scegli EventBridge APIla destinazione.
12. Per APIdestinazione, scegli Usa una API destinazione esistente, quindi scegli la `DatadogAD` destinazione creata nel 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 Successivo.
16. Seleziona Successivo.
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 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 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 API destinazione 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 la/le EventBridge API destinazione/i

1. Apri la pagina di API destinazione della EventBridge console.
2. Seleziona le API destinazioni che hai creato.
3. Scegli Elimina.
4. Inserisci il nome della API destinazione e scegli Elimina.

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. Scegli Delete (Elimina).

Tutorial: invia eventi a Salesforce da Amazon EventBridge

Puoi utilizzarlo EventBridge per indirizzare gli eventi a servizi di terze parti, ad esempio. Salesforce

In questo tutorial, utilizzerai la EventBridge console per creare una connessione versoSalesforce, una APIdestinazione che punti e una regola verso cui indirizzare gli eventiSalesforce. Salesforce

Fasi:

- Prerequisiti
- Passaggio 1: creare una connessione
- Fase 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 aSalesforce, devi prima stabilire una connessione a. Salesforce API

Per creare la connessione

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli APIIle destinazioni.
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. Per HTTPil metodo, scegli POSTdall'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 OAuthHttp Parameters, inserisci la seguente coppia chiave/valore:

Key (Chiave)	Value (Valore)
grant_type	client_credentials

11. Scegli Crea.

Fase 2: Creare una destinazione API

Ora che hai creato la connessione, creerai la API destinazione da utilizzare come destinazione della regola.

Per creare la API destinazione

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli APIle destinazioni.
3. Scegli Crea API destinazione.
4. Inserisci un nome e una descrizione per la API destinazione. In questo esempio, immetti **`SalesforceAD`** come nome e **`Salesforce API Destination`** come descrizione.

5. Per l'endpoint di API destinazione, inserisci **`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. Per il HTTPmetodo, scegli POSTdall'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 Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "source": ["aws.s3"]
}
```

10. Seleziona Successivo.
11. Per i tipi di Target, scegli EventBridge APIla destinazione.

12. Per APIdestinazione, scegli Usa una API destinazione esistente, quindi scegli la `SalesforceAD` destinazione creata nel 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 Successivo.
16. Seleziona Successivo.
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 API destinazione 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 EventBridge API destinazioni

1. Apri la pagina di API destinazione della EventBridge console.
2. Seleziona le API destinazioni che hai creato.
3. Scegli Elimina.
4. Inserisci il nome della API destinazione 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: invia eventi a Zendesk da Amazon EventBridge

Puoi utilizzarlo EventBridge per indirizzare gli eventi a servizi di terze parti come Zendesk.

In questo tutorial, utilizzerai la EventBridge console per creare una connessione versoZendesk, una APIdestinazione che punti e una regola verso cui indirizzare gli eventiZendesk. Zendesk

Fasi:

- Prerequisiti
- Passaggio 1: creare una connessione
- Fase 2: Creare una API destinazione
- 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 Zendesk.
- Un EventBridge bucket Amazon Simple Storage Service (Amazon S3) abilitato per Amazon.

Passaggio 1: creare una connessione

Per inviare eventi aZendesk, devi prima stabilire una connessione a. Zendesk API

Per creare la connessione

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli APIIe destinazioni.
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.

Fase 2: Creare una API destinazione

Ora che hai creato la connessione, creerai la API destinazione da utilizzare come destinazione della regola.

Per creare la API destinazione

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel riquadro di navigazione, scegli APIIe destinazioni.
3. Scegli Crea API destinazione.
4. Inserisci un nome e una descrizione per la API destinazione. In questo esempio, immetti **`ZendeskAD`** come nome e **`Zendesk API destination`** come descrizione.
5. Per l'endpoint di API destinazione, immettere**`https://your-subdomain.zendesk.com/api/v2/tickets.json`**, dove *`your-subdomain`* è il sottodominio associato al tuo Zendesk account.
6. Per HTTPmetodo, 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. 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 **`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. Seleziona Successivo.
8. In Event source (Origine eventi), scegli Other (Altro).
9. In Modello di eventi, immetti quanto segue:

```
{
  "source": ["aws.s3"]
}
```

10. Seleziona Successivo.
11. Per i tipi di Target, scegli EventBridge APIla destinazione.
12. Per APIdestinazione, scegli Usa una API destinazione esistente, quindi scegli la `ZendeskAD` destinazione creata nel 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 Successivo.
16. Seleziona Successivo.
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 Quando l'evento corrisponde alla regola, EventBridge chiamerà il Create Ticket. Zendesk API 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 conservarle. Eliminando AWS le risorse che non utilizzi più, eviti addebiti inutili sul tuo AWS account.

Per eliminare le EventBridge connessioni

1. Apri la pagina di API destinazione 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 la/le EventBridge API destinazione/i

1. Apri la pagina di API destinazione della EventBridge console.
2. Seleziona le API destinazioni che hai creato.
3. Scegli Elimina.
4. Inserisci il nome della API destinazione e scegli Elimina.

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. Scegli Delete (Elimina).

Utilizzo EventBridge con un AWS SDK

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione più diffusi. Ciascuno di essi SDK fornisceAPI, esempi di codice e documentazione che semplificano agli sviluppatori la creazione di applicazioni nel linguaggio preferito.

SDKdocumentazione	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 EventBridge l'utilizzo AWS SDKs.

Esempio di disponibilità

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

Esempi di codice per EventBridge l'utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzare EventBridge con un kit di sviluppo AWS software (SDK).

Le nozioni di base sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Gli scenari sono esempi di codice che mostrano come eseguire attività specifiche richiamando più funzioni all'interno di un servizio o combinandole con altre AWS servizi.

Per un elenco completo di guide per AWS SDK sviluppatori ed esempi di codice, consultaUtilizzo EventBridge con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

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 API i dettagli, vedi ListEventBusesin AWS SDK for .NET APIReference.

Java

SDKper 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.

```
/**
```

```
 * 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 API i dettagli, vedi ListEventBusesin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
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 API i dettagli, vedi il riferimento ListEventBuses AWSSDKa Kotlin API.

Esempi di codice

- Esempi di base per l'utilizzo EventBridge AWS SDKs
  - Salve EventBridge

- Impara le nozioni di base di EventBridge con un AWS SDK
- Azioni per EventBridge l'utilizzo AWS SDKs
  - Usalo DeleteRule con un AWS SDK o CLI
  - Utilizzare DescribeRule con un AWS SDK o CLI
  - Utilizzare DisableRule con un AWS SDK o CLI
  - Utilizzare EnableRule con un AWS SDK o CLI
  - Utilizzare ListRuleNamesByTarget con un AWS SDK o CLI
  - Utilizzare ListRules con un AWS SDK o CLI
  - Utilizzare ListTargetsByRule con un AWS SDK o CLI
  - Utilizzare PutEvents con un AWS SDK o CLI
  - Utilizzare PutRule con un AWS SDK o CLI
  - Utilizzare PutTargets con un AWS SDK o CLI
  - Utilizzare RemoveTargets con un AWS SDK o CLI
- Scenari di EventBridge utilizzo AWS SDKs
  - Crea e attiva una regola in Amazon EventBridge utilizzando un AWS SDK
  - Invia notifiche di eventi S3 ad Amazon EventBridge utilizzando un AWS SDK
  - Utilizzo degli eventi pianificati per richiamare una funzione Lambda

Esempi di base per l'utilizzo EventBridge AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di Amazon EventBridge con AWS SDKs.

Esempi

- Salve EventBridge
- Impara le nozioni di base di EventBridge con un AWS SDK
- Azioni per EventBridge l'utilizzo AWS SDKs
  - Usalo DeleteRule con un AWS SDK o CLI
  - Utilizzare DescribeRule con un AWS SDK o CLI
  - Utilizzare DisableRule con un AWS SDK o CLI
  - Utilizzare EnableRule con un AWS SDK o CLI

- Utilizzare ListRuleNamesByTarget con un AWS SDK o CLI
- Utilizzare ListRules con un AWS SDK o CLI
- Utilizzare ListTargetsByRule con un AWS SDK o CLI
- Utilizzare PutEvents con un AWS SDK o CLI
- Utilizzare PutRule con un AWS SDK o CLI
- Utilizzare PutTargets con un AWS SDK o CLI
- Utilizzare RemoveTargets con un AWS SDK o CLI

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 API i dettagli, vedi ListEventBusesin AWS SDK for .NET APIReference.

Java

SDKper 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.

```
/**
 * 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 API i dettagli, vedi ListEventBusesin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
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 API i dettagli, vedi il riferimento ListEventBuses AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Impara le nozioni di base di EventBridge con un AWS SDK

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 API i dettagli, consultate i seguenti argomenti in AWS SDK for .NET API Riferimento.
  - DeleteRule
  - DescribeRule
  - DisableRule

- EnableRule
- ListRuleNamesByTarget
- ListRules
- ListTargetsByRule
- PutEvents
- PutRule
- PutTargets

Java

SDKper 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.

```
/**
 * 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 API i dettagli, consulta i seguenti argomenti in AWS SDK for Java 2.x APIRiferimento.
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/*
 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 API i dettagli, consulta i seguenti argomenti in riferimento AWS SDKa Kotlin API.
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Azioni per EventBridge l'utilizzo AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole EventBridge azioni con AWS SDKs. Ogni esempio include un collegamento a GitHub, dove sono disponibili le istruzioni per la configurazione e l'esecuzione del codice.

Questi estratti chiamano EventBridge API e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. È possibile visualizzare le azioni nel contesto in. Scenari di EventBridge utilizzo AWS SDKs

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta Amazon EventBridge API Reference.

Esempi

- Usalo DeleteRule con un AWS SDK o CLI
- Utilizzare DescribeRule con un AWS SDK o CLI
- Utilizzare DisableRule con un AWS SDK o CLI
- Utilizzare EnableRule con un AWS SDK o CLI
- Utilizzare ListRuleNamesByTarget con un AWS SDK o CLI
- Utilizzare ListRules con un AWS SDK o CLI
- Utilizzare ListTargetsByRule con un AWS SDK o CLI
- Utilizzare PutEvents con un AWS SDK o CLI
- Utilizzare PutRule con un AWS SDK o CLI
- Utilizzare PutTargets con un AWS SDK o CLI
- Utilizzare RemoveTargets con un AWS SDK o CLI

Usalo **`DeleteRule`** con un AWS SDK o 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:

- Impara le nozioni di base

.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 API i dettagli, vedi DeleteRulein AWS SDK for .NET APIReference.

CLI

AWS CLI

Per eliminare una regola CloudWatch Events

Questo esempio elimina la regola denominataEC2InstanceStateChanges:

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- Per API i dettagli, vedere DeleteRulein AWS CLI Command Reference.

Java

SDKper 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 API i dettagli, vedi DeleteRulein AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteRuleByName(ruleName: String?) {
    val ruleRequest =
        DeleteRuleRequest {
            name = ruleName
        }
    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.deleteRule(ruleRequest)
        println("Successfully deleted the rule")
    }
}
```

- Per API i dettagli, vedi il riferimento DeleteRule AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`DescribeRule`** con un AWS SDK o 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:

- Impara le nozioni di base

.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 API i dettagli, vedi DescribeRulein AWS SDK for .NET APIReference.

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 API i dettagli, vedere DescribeRulein AWS CLI Command Reference.

Java

SDKper 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 API i dettagli, vedi DescribeRulein AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun 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 API i dettagli, vedi il riferimento DescribeRule AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`DisableRule`** con un AWS SDK o 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:

- Impara le nozioni di base

.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 API i dettagli, vedi DisableRulein AWS SDK for .NET APIReference.

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 API i dettagli, vedere DisableRulein AWS CLI Command Reference.

Java

SDKper 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 API i dettagli, vedi DisableRulein AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun 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 API i dettagli, vedi il riferimento DisableRule AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`EnableRule`** con un AWS SDK o 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:

- Impara le nozioni di base

.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 API i dettagli, vedi EnableRulein AWS SDK for .NET APIReference.

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 API i dettagli, vedere EnableRulein AWS CLI Command Reference.

Java

SDKper 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 API i dettagli, vedi EnableRulein AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun 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 API i dettagli, vedi il riferimento EnableRule AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare `ListRuleNamesByTarget` con un AWS SDK o 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:

- Impara le nozioni di base

.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 API i dettagli, vedi ListRuleNamesByTargetin AWS SDK for .NET APIReference.

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 API i dettagli, consulta ListRuleNamesByTarget AWS CLICommand Reference.

Java

SDKper 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 API i dettagli, vedi ListRuleNamesByTargetin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun 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 API i dettagli, vedi il riferimento ListRuleNamesByTarget AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`ListRules`** con un AWS SDK o 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:

- Impara le nozioni di base

.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 API i dettagli, vedi ListRulesin AWS SDK for .NET APIReference.

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 API i dettagli, vedere ListRulesin AWS CLI Command Reference.

Java

SDKper 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 API i dettagli, vedi ListRulesin AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun 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 API i dettagli, vedi il riferimento ListRules AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`ListTargetsByRule`** con un AWS SDK o 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:

- Impara le nozioni di base

.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 API i dettagli, vedi ListTargetsByRulein AWS SDK for .NET APIReference.

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 API i dettagli, vedere ListTargetsByRulein AWS CLI Command Reference.

Java

SDKper 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 API i dettagli, vedi ListTargetsByRulein AWS SDK for Java 2.x APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun 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 API i dettagli, vedi il riferimento ListTargetsByRule AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`PutEvents`** con un AWS SDK o 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:

- Impara le nozioni di base
- Creazione e attivazione di una regola

.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 API i dettagli, vedi PutEventsin AWS SDK for .NET APIReference.

C++

SDKper C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/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 API i dettagli, vedi PutEventsin AWS SDK for C++ APIReference.

CLI

AWS CLI

Per inviare un evento personalizzato a CloudWatch Eventi

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 API i dettagli, vedere PutEventsin AWS CLI Command Reference.

Java

SDKper 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 API i dettagli, vedi PutEventsin AWS SDK for Java 2.x APIReference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Importa i moduli SDK e client e chiama ilAPI.

```
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 API i dettagli, vedere PutEventsin AWS SDK for JavaScript APIReference.

SDKper JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create 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 API i dettagli, vedi PutEventsin AWS SDK for JavaScript APIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun 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 API i dettagli, vedi il riferimento PutEvents AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`PutRule`** con un AWS SDK o 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:

- Impara le nozioni di base
- Creazione e attivazione di una regola
- Invia notifiche di eventi a 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.

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 API i dettagli, vedi PutRulein AWS SDK for .NET APIReference.

C++

SDKper C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/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 API i dettagli, vedi PutRule AWS SDK for C++APIReference.

CLI

AWS CLI

Per creare regole relative CloudWatch agli eventi

Questo esempio crea una regola che viene attivata ogni giorno alle 9:00am ()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
 * * ? *)"
```

Questo esempio crea una regola che si attiva quando un'EC2istanza nella regione cambia stato:

```
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"
```

Questo esempio crea una regola che si attiva quando un'EC2istanza nella regione viene interrotta 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 API i dettagli, vedere PutRulein AWS CLI Command Reference.

Java

SDKper 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 API i dettagli, vedi PutRule AWS SDK for Java 2.xAPIReference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Importa i moduli SDK e client e chiama ilAPI.

```
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 API i dettagli, vedere PutRulein AWS SDK for JavaScript APIReference.

SDKper JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create 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 API i dettagli, vedi PutRule AWS SDK for JavaScriptAPIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

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 API i dettagli, vedi il riferimento PutRule AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`PutTargets`** con un AWS SDK o 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 nei seguenti esempi di codice:

- Impara le nozioni di base
- Invia notifiche di eventi a 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.

Aggiungi un SNS argomento Amazon come obiettivo 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 API i dettagli, PutTargetsconsulta AWS SDK for .NET APIReference.

C++

SDKper C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/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 API i dettagli, vedi PutTargets AWS SDK for C++APIReference.

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 API i dettagli, vedere PutTargetsin AWS CLI Command Reference.

Java

SDKper 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 SNS argomento Amazon come obiettivo 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 API i dettagli, PutTargetsconsulta AWS SDK for Java 2.x APIReference.

JavaScript

SDKper JavaScript (v3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Importa i moduli SDK e client e chiama ilAPI.

```
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 API i dettagli, vedere PutTargetsin AWS SDK for JavaScript APIReference.

SDKper JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create 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 API i dettagli, vedi PutTargets AWS SDK for JavaScriptAPIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// 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 API i dettagli, vedi il riferimento PutTargets AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzare **`RemoveTargets`** con un AWS SDK o 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 API i dettagli, vedi RemoveTargets AWS SDK for .NETAPIReference.

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 API i dettagli, consulta AWS CLI Command RemoveTargetsReference.

Java

SDKper 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 API i dettagli, vedi RemoveTargets AWS SDK for Java 2.xAPIReference.

Kotlin

SDKper Kotlin

> **Note**
>
> c'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun 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 API i dettagli, vedi il riferimento RemoveTargets AWSSDKa Kotlin API.

Per un elenco completo delle guide per AWS SDK sviluppatori e degli esempi di codice, consulta. Utilizzo EventBridge con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Scenari di EventBridge utilizzo AWS SDKs

I seguenti esempi di codice mostrano come implementare scenari comuni in EventBridge with AWS SDKs. Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno EventBridge o combinandole con altre AWS servizi. Ogni scenario include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice.

Gli scenari si basano su un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- Crea e attiva una regola in Amazon EventBridge utilizzando un AWS SDK
- Invia notifiche di eventi S3 ad Amazon EventBridge utilizzando un AWS SDK
- Utilizzo degli eventi pianificati per richiamare una funzione Lambda

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

SDKper Ruby

> **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.

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 (AmazonSNS) specificato esiste tra quelli forniti per 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 AmazonSNS.

```
# 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 poi sottoscrivi 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 inIAM.

```
# 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 inIAM.

```
# 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 (AmazonEC2) e aggiunge informazioni sull'attività correlata a un flusso di log in CloudWatch Logs.

```
# 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 API i dettagli, consulta i seguenti argomenti in AWS SDK for Ruby API Reference.
  - PutEvents
  - PutRule

Per un elenco completo di guide per AWS SDK sviluppatori ed esempi di codice, consultaUtilizzo EventBridge con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Invia notifiche di eventi S3 ad Amazon EventBridge utilizzando un AWS SDK

Il seguente esempio di codice mostra come abilitare un bucket per inviare notifiche di eventi S3 EventBridge e indirizzare le notifiche verso un SNS argomento Amazon e una coda AmazonSQS.

Java

SDKper 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.

```
   /** This method configures a bucket to send events to AWS EventBridge and
creates a rule
    * to route the S3 object created events to a topic and a queue.
    *
    * @param bucketName Name of existing bucket
```

```
    * @param topicArn ARN of existing topic to receive S3 event notifications
    * @param queueArn ARN of existing queue to receive S3 event notifications
    *
    *  An AWS CloudFormation stack sets up the bucket, queue, topic before the
method runs.
    */
   public static String setBucketNotificationToEventBridge(String bucketName,
String topicArn, String queueArn) {
       try {
           // Enable bucket to emit S3 Event notifications to EventBridge.
           s3Client.putBucketNotificationConfiguration(b -> b
                   .bucket(bucketName)
                   .notificationConfiguration(b1 -> b1
                           .eventBridgeConfiguration(
                                   SdkBuilder::build)
                   ).build()).join();

           // Create an EventBridge rule to route Object Created notifications.
           PutRuleRequest putRuleRequest = PutRuleRequest.builder()
                   .name(RULE_NAME)
                   .eventPattern("""
                           {
                             "source": ["aws.s3"],
                             "detail-type": ["Object Created"],
                             "detail": {
                               "bucket": {
                                 "name": ["%s"]
                               }
                             }
                           }
                           """.formatted(bucketName))
                   .build();

           // Add the rule to the default event bus.
           PutRuleResponse putRuleResponse =
eventBridgeClient.putRule(putRuleRequest)
                   .whenComplete((r, t) -> {
                       if (t != null) {
                           logger.error("Error creating event bus rule: " +
t.getMessage(), t);
                           throw new RuntimeException(t.getCause().getMessage(),
t);
                       }
```

```
                        logger.info("Event bus rule creation request sent
successfully. ARN is: {}", r.ruleArn());
                    }).join();

            // Add the existing SNS topic and SQS queue as targets to the rule.
            eventBridgeClient.putTargets(b -> b
                    .eventBusName("default")
                    .rule(RULE_NAME)
                    .targets(List.of (
                            Target.builder()
                                    .arn(queueArn)
                                    .id("Queue")
                                    .build(),
                            Target.builder()
                                    .arn(topicArn)
                                    .id("Topic")
                                    .build())
                            )
                    ).join();
            return putRuleResponse.ruleArn();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
```

- Per API i dettagli, consulta i seguenti argomenti in AWS SDK for Java 2.x APIRiferimento.
  - PutBucketNotificationConfiguration
  - PutRule
  - PutTargets

Per un elenco completo di guide per AWS SDK sviluppatori ed esempi di codice, consultaUtilizzo EventBridge con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

Utilizzo degli eventi pianificati per richiamare una funzione Lambda

I seguenti esempi di codice mostrano come creare una AWS Lambda funzione richiamata da un evento EventBridge pianificato di Amazon.

Java

SDKper 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, si crea una funzione Lambda utilizzando il runtime Lambda Java. API 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

SDKper 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, si crea una funzione Lambda utilizzando il runtime Lambda. JavaScript API 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

SDKper Python (Boto3)

Questo esempio mostra come registrare una AWS Lambda funzione come destinazione di un EventBridge evento Amazon pianificato. Il gestore Lambda scrive un messaggio intuitivo e i dati completi dell'evento su Amazon CloudWatch Logs per recuperarli in un secondo momento.

- Distribuzione di una funzione Lambda.
- Crea un evento EventBridge pianificato e rende la funzione Lambda la destinazione.
- Concede il permesso di EventBridge invocare la funzione Lambda.
- Stampa i dati più recenti dai CloudWatch registri per mostrare il risultato delle chiamate pianificate.
- Elimina tutte le risorse create durante la demo.

Questo esempio è visualizzato al meglio su. GitHub Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- CloudWatch Registri
- EventBridge
- Lambda

Per un elenco completo delle guide per AWS SDK gli sviluppatori e degli esempi di codice, consultaUtilizzo EventBridge con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle SDK versioni precedenti.

EventBridge Sicurezza Amazon

Amazon EventBridge usi AWS Identity and Access Management per controllare l'accesso ad altri AWS servizi e risorse. Per una panoramica di come IAM funziona, vedere Panoramica sulla gestione degli accessi nella Guida per l'IAMutente. 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 modello 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 AWS servizi utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta la sezione Privacy dei dati FAQ. Per informazioni sulla protezione dei dati in Europa, consulta il Modello di responsabilitàAWS condivisa e GDPR il post sul blog sulla AWS sicurezza.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e di configurare i singoli utenti con AWS IAM Identity Center o 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.
- UsaSSL/TLSper comunicare con AWS le risorse. Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Configurazione API e registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno AWS servizi.
- 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 FIPS 140-3 moduli crittografici convalidati per accedere AWS tramite un'interfaccia a riga di comando o unAPI, usa un endpoint. FIPS Per ulteriori informazioni sugli FIPS endpoint disponibili, vedere Federal Information Processing Standard () 140-3. FIPS

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 EventBridge o AWS servizi utilizzi in altro modo la console, API AWS CLI, o. AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Se fornisci un messaggio URL a un server esterno, ti consigliamo vivamente di non includere le informazioni sulle credenziali URL per convalidare la tua richiesta a quel server.

Crittografia dei dati in EventBridge

EventBridge fornisce sia la crittografia a riposo che la crittografia in transito per proteggere i dati:

- Crittografia a riposo

  EventBridge si integra con AWS Key Management Service (KMS) per crittografare i dati archiviati. Per impostazione predefinita, EventBridge utilizza an Chiave di proprietà di AWS per crittografare i dati degli eventi. È inoltre possibile specificare EventBridge di utilizzare invece a chiave gestita dal cliente per quanto segue.

  - Event bus: eventi personalizzati e organizzati dai 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 inattiva in Amazon EventBridge

EventBridge fornisce una crittografia trasparente lato server grazie all'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 a riposo tramite Event Bus include:

- Dati sugli eventi per eventi personalizzati e per i partner.

  Per gli event bus, i dati degli eventi includono tutti i campi contenuti nell'elemento di dettaglio dell'evento.

  EventBridge non crittografa i metadati degli eventi. Per ulteriori informazioni sui metadati degli eventi, vedere. Campi di metadati degli eventi
- Schemi 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 dei bus di eventi in EventBridge

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 dei dati in Amazon EventBridge

EventBridge utilizza an Chiave di proprietà di AWS per crittografare gli eventi AWS di servizio memorizzati sui bus degli 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 a 256 bit (AES-256) 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 IAM politiche e sovvenzioni
  - 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 la sezione 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

Autorizzazione EventBridge all'uso di un chiave gestita dal cliente

Se utilizzi una password chiave gestita dal cliente nel tuo account per proteggere il tuo bus per EventBridge eventi, 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'ID chiave fornito e KMS key ARN 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 KMS chiave 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"
        }
      }
```

Mantenimento dell'accesso alla chiave di AWS KMS crittografia in EventBridge

Per garantire che conservi EventBridge sempre l'accesso a quanto 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 un obiettivo 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.

Contesto di crittografia in Amazon EventBridge

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.

Puoi anche 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 il bus degli eventi: ARN

```
"encryptionContext": {
    "kms:EncryptionContext:aws:events:event-bus:arn": "event-bus-arn"
}
```

Crittografia degli eventi con chiavi in AWS KMS EventBridge

Puoi specificare di EventBridge utilizzare AWS KMS a per crittografare i dati (eventi personalizzati e partner) archiviati 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 KMS key opzioni.

Se specificate a chiave gestita dal cliente per un bus di eventi, avete la possibilità di specificare una coda di lettere morte (DLQ) per il bus degli eventi. EventBridge invia quindi tutti gli eventi personalizzati o dei partner che generano errori di crittografia o decrittografia. DLQ Per ulteriori informazioni, consulta DLQsper eventi crittografati.

Specificare la AWS KMS chiave utilizzata per la crittografia durante la creazione di un bus di eventi

La scelta della AWS KMS chiave utilizzata per la crittografia fa parte della creazione di un bus di eventi. L'impostazione predefinita prevede l'utilizzo del Chiave di proprietà di AWS file fornito da EventBridge.

Specificare un valore chiave gestita dal cliente per la crittografia durante la creazione di un bus di eventi (console)

- Segui queste istruzioni:

  Creazione di un router di eventi.

Per specificare un chiave gestita dal cliente per la crittografia durante la creazione di un bus di eventi (CLI)

- Durante la chiamata`create-event-bus`, utilizzate l'`kms-key-identifier`opzione per specificare il chiave gestita dal cliente modulo EventBridge da utilizzare per la crittografia sul bus degli eventi.

Facoltativamente, utilizzare `dead-letter-config` per specificare una coda di lettere morte (). DLQ

Aggiornamento della AWS KMS chiave utilizzata per la crittografia su un bus di eventi

È possibile aggiornare la AWS KMS chiave utilizzata per la crittografia inattiva su un bus di eventi esistente. Ciò include il passaggio dal valore predefinito Chiave di proprietà di AWS a a chiave gestita dal cliente, da chiave gestita dal cliente a al valore predefinito Chiave di proprietà di AWS o da uno chiave gestita dal cliente all'altro.

Per aggiornare il codice KMS key utilizzato per la crittografia su un bus di eventi (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 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 SQS coda Amazon da utilizzare come coda di lettere non scritte (DLQ) per questo bus di eventi, se disponibile.

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

Per aggiornare quello KMS key utilizzato per la crittografia su un bus di eventi (CLI)

- Durante la chiamata`update-event-bus`, utilizzate l'`kms-key-identifier`opzione per specificare il chiave gestita dal cliente modulo EventBridge da utilizzare per la crittografia sul bus degli eventi.

  Facoltativamente, utilizzare `dead-letter-config` per specificare una coda di lettere morte (). DLQ

Per aggiornare quello KMS key utilizzato per la crittografia sul bus degli eventi predefinito, utilizzando CloudFormation

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:

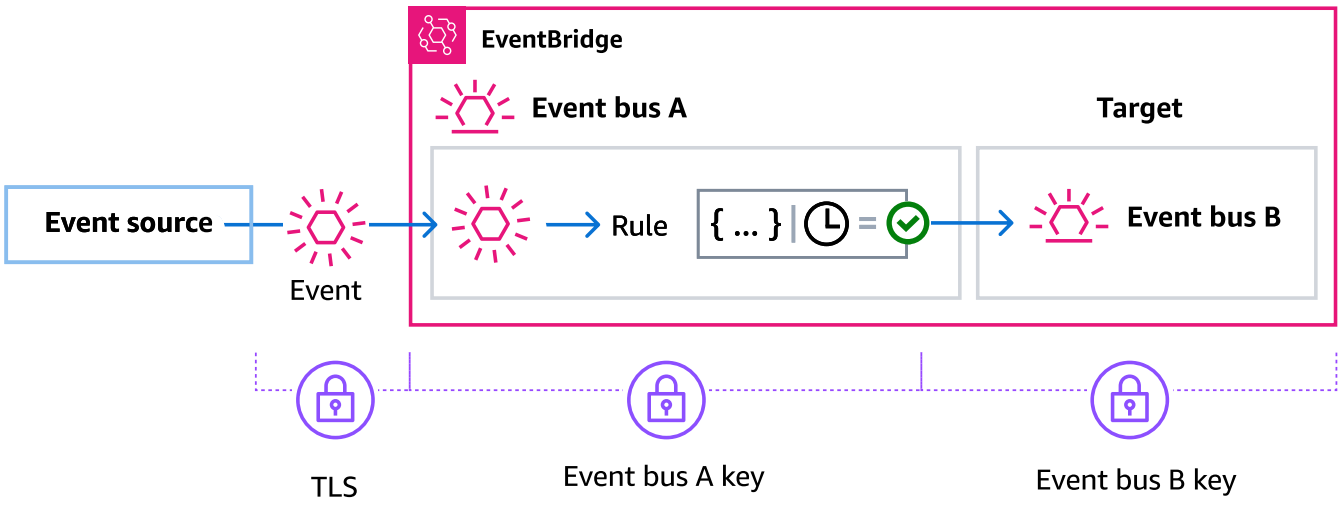
  Aggiornamento di un bus predefinito utilizzando CloudFormation.

Crittografia EventBridge attiva 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 KMS chiave di crittografia a riposo per quel bus di eventi, predefinita Chiave di proprietà di AWS o una chiave gestita dal cliente, se ne è stata specificata una. Se un evento corrisponde a una regola, EventBridge crittografa l'evento con la configurazione KMS chiave 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.

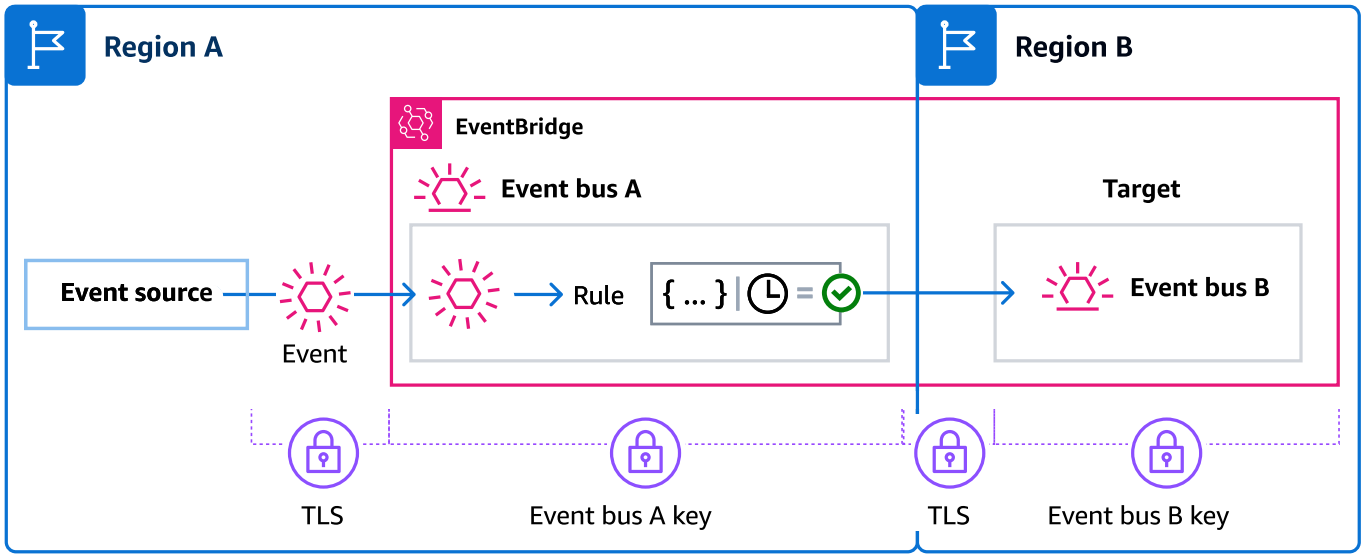
- Se la destinazione di una regola è un altro bus di eventi nella stessa AWS regione:

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:

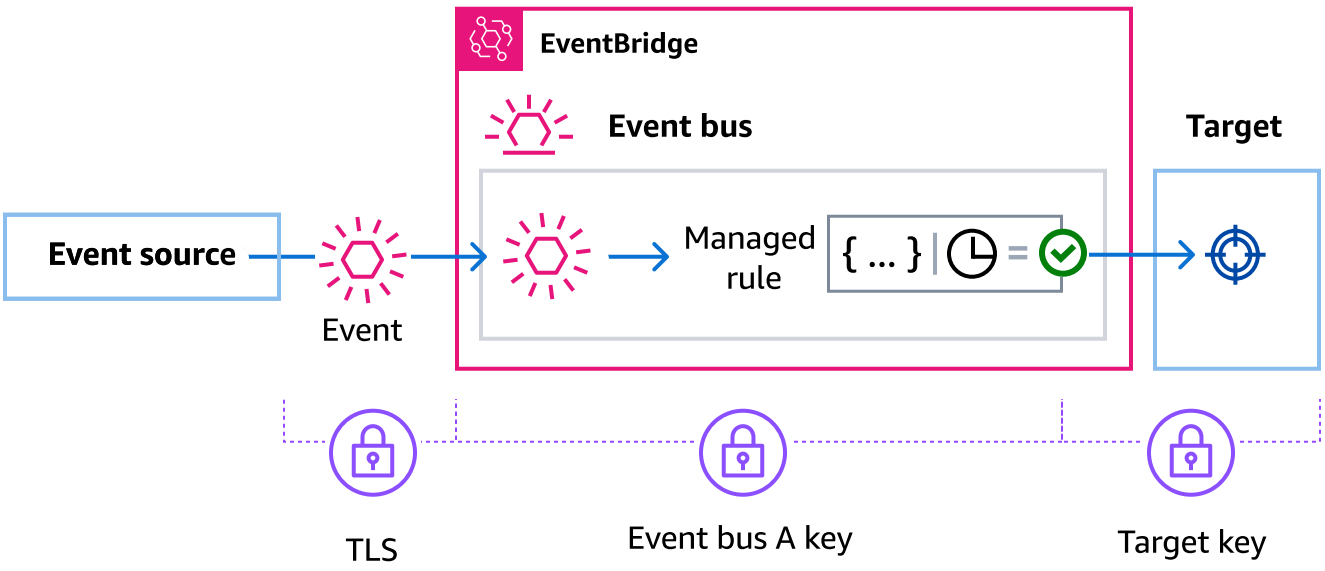
  EventBridge crittografa l'evento a riposo in base alla configurazione delle KMS chiavi sul primo bus di eventi. EventBridge utilizza TLS per inviare l'evento al secondo bus di eventi nella diversa regione, dove viene quindi crittografato in base alla configurazione della KMS chiave specificata per il bus degli eventi di destinazione.



Crittografia degli eventi per le regole gestite in EventBridge

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.

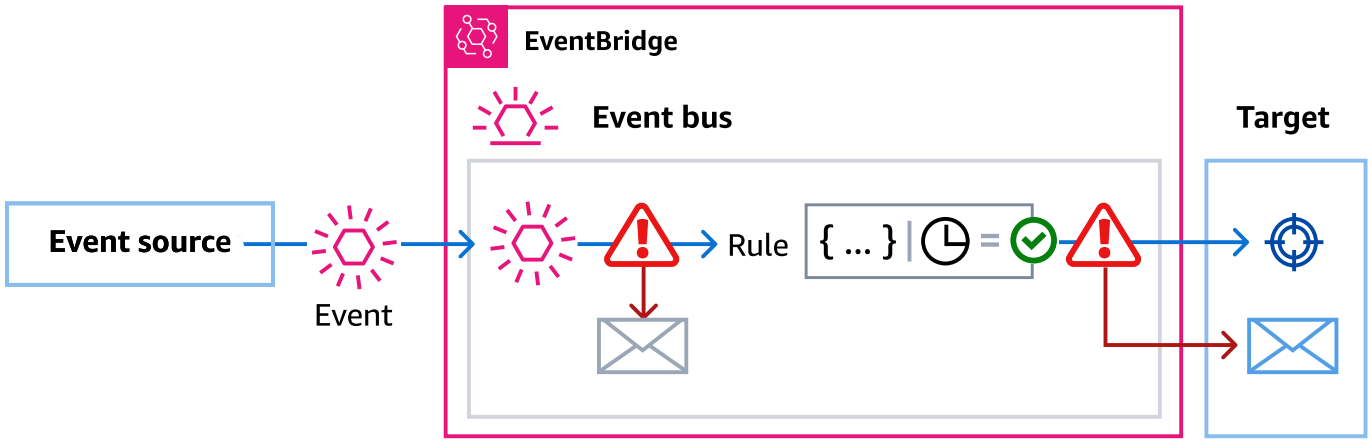


Nei casi in cui l'obiettivo della regola è un bus di eventi in un'altra regione, è necessario fornire una chiave multiregionale. Il bus degli eventi nella prima regione crittografa l'evento utilizzando chiave gestita dal cliente quanto specificato nella regola gestita. Quindi invia l'evento al bus degli 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.

Utilizzo di code di lettere morte per acquisire gli errori degli eventi crittografati in EventBridge

Se configuri chiave gestita dal cliente la crittografia su un bus di eventi, ti consigliamo di specificare una coda di lettere morte (DLQ) per quel bus di eventi. EventBridge invia eventi personalizzati e partner a questo indirizzo DLQ se rileva un errore non recuperabile 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 bus degli eventi 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 alla DLQ destinazione, se specificata.



Per ulteriori informazioni, incluse considerazioni sull'utilizzo e istruzioni sull'impostazione delle autorizzazioniDLQs, vedere. Utilizzo di code DLQ

Decrittografia degli eventi nelle code con lettere non scritte EventBridge

Una volta risolto il problema di fondo che causa un errore non recuperabile, puoi elaborare gli eventi inviati al bus o alla destinazione degli eventi. DLQs Per gli eventi crittografati, è necessario prima decrittografare l'evento per elaborarlo.

L'esempio seguente mostra come decrittografare un evento che EventBridge è stato recapitato a un bus o a una destinazione di eventi. DLQ

```
// 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);
```

Politiche basate su tag in Amazon EventBridge

In Amazon EventBridge, puoi utilizzare politiche 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:

- Etichettare le risorse in Amazon EventBridge
- Controllo dell'accesso tramite IAM tag

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. Le tue credenziali devono disporre delle autorizzazioni per accedere alle AWS risorse, ad esempio recuperare i dati degli eventi da altre risorse. AWS Le sezioni seguenti 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 AWS con uno qualsiasi dei seguenti tipi di identità:

- AWS utente root dell'account: quando ti registri AWS, fornisci un indirizzo email e una password associati al tuo account. Queste sono le tue credenziali di root e forniscono l'accesso completo a tutte le tue AWS risorse.

> **Important**
>
> Per motivi di sicurezza, ti consigliamo di utilizzare le credenziali root solo per creare un amministratore, ovvero un IAMutente con autorizzazioni complete per l'account. Potrai quindi utilizzare questo amministratore per creare altri utenti e ruoli con autorizzazioni limitate. Per ulteriori informazioni, consulta le procedure IAM consigliate e la creazione di un utente e un gruppo di amministratori nella Guida per l'IAMutente.

- IAMutente: un IAMutente è un'identità all'interno del tuo account che dispone di autorizzazioni specifiche, ad esempio l'autorizzazione a inviare i dati degli eventi a un target in EventBridge. È possibile utilizzare le credenziali di IAM accesso per accedere a AWS pagine Web sicure come il Forum di AWS discussione o il AWS Management ConsoleCentro.AWS Support

  Inoltre, puoi generare le chiavi di accesso per ogni utente. È possibile utilizzare queste chiavi quando si accede ai AWS servizi in modo programmatico per firmare crittograficamente la richiesta, tramite uno dei o utilizzando il SDKs ().AWS Command Line InterfaceAWS CLI Se non utilizzi AWS strumenti, devi firmare tu stesso la richiesta con Signature Version 4, un protocollo per l'autenticazione delle richieste in entrata. API 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.

- IAMruolo: un IAMruolo è un'altra IAM identità che puoi creare nel tuo account con autorizzazioni specifiche. È simile a un IAMutente, ma non è associato a una persona specifica. Utilizzando un IAM ruolo, è possibile ottenere chiavi di accesso temporanee per accedere a AWS servizi e risorse. IAMi ruoli con credenziali temporanee sono utili nelle seguenti situazioni:

  - Accesso utente federato: anziché creare un utente, è possibile utilizzare le identità della directory utente aziendale o di AWS Directory Service un provider di identità Web (IdP). Questi sono noti come utenti federati. AWS assegna un ruolo a un utente federato quando l'utente richiede l'accesso tramite un provider di identità. Per ulteriori informazioni sugli utenti federati, consulta Federated Users and Roles nella Guida per l'utente. IAM

  - Accesso su più account: puoi utilizzare un IAM ruolo nel tuo account per concedere a un altro account l'autorizzazione ad accedere alle risorse del tuo account. Per un esempio, vedi Tutorial: Delegare l'accesso tra AWS account utilizzando i IAM ruoli nella Guida per l'IAMutente.

  - AWS accesso al servizio: puoi utilizzare un IAM ruolo nel tuo account per concedere a un AWS servizio l'autorizzazione ad 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 AWS servizio nella Guida per l'IAMutente.

  - Applicazioni in esecuzione su Amazon EC2: per EC2 le applicazioni Amazon che richiedono l'accesso a EventBridge, puoi archiviare le chiavi di accesso nell'EC2istanza oppure utilizzare un IAM ruolo per gestire le credenziali temporanee. Per assegnare un AWS ruolo a un'EC2istanza, crei un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e fornisce credenziali temporanee alle applicazioni in esecuzione sull'EC2istanza. Per ulteriori informazioni, consulta Using Roles for Applications on Amazon EC2 nella Guida per l'IAMutente.

Controllo accessi

Per creare o accedere alle EventBridge risorse, sono necessarie credenziali e autorizzazioni valide. Ad esempio, per richiamare gli AWS Lambda obiettivi Amazon Simple Notification Service (AmazonSNS) e Amazon Simple Queue Service (AmazonSQS), devi disporre delle autorizzazioni per tali servizi.

Gestione delle autorizzazioni di accesso alle tue risorse Amazon EventBridge

È possibile gestire l'accesso a EventBridge risorse come regole o eventi utilizzando politiche basate sull'identità o sulle risorse.

EventBridge risorse

EventBridge alle risorse e alle sottorisorse sono associati Amazon Resource Names (ARNs) univoci. Si usa ARNs in EventBridge per creare modelli di eventi. Per ulteriori informazioni suARNs, consulta Amazon Resource Names (ARN) e AWS Service Namespaces nel. Riferimenti generali di Amazon Web Services

Per un elenco delle operazioni che EventBridge prevede l'utilizzo delle risorse, consulta. Riferimento alle EventBridge autorizzazioni Amazon

> **Note**
>
> La maggior parte dei servizi considera i due punti (`:`) o una barra (`/`) come lo stesso carattere inARNs. AWS Tuttavia, EventBridge utilizza una corrispondenza esatta nei modelli e nelle regole degli eventi. Assicurati di utilizzare i ARN caratteri corretti quando crei i modelli di eventi in modo che corrispondano alla ARN sintassi dell'evento a cui desideri abbinare.

La tabella seguente mostra le risorse in EventBridge.

Tipo di risorsa	ARNFormato
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	ARNFormato
Tutte le EventBridge risorse	`arn:aws:events:*`
Tutte EventBridge le risorse di proprietà dell'account specificato nella regione specificata	`arn:aws:events: region:account:*`

L'esempio seguente mostra come indicare una regola specifica (*myRule*) nella tua dichiarazione usando la suaARN.

```
"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 un'APIazione specifica non supportaARNs, usa il carattere jolly asterisco (*) nell'`Resource`elemento come segue.

```
"Resource": "*"
```

Per specificare più risorse o `PutTargets` in un'unica istruzione, separale ARNs con virgole come segue.

```
"Resource": ["arn1", "arn2"]
```

Proprietà delle risorse

Un account è proprietario delle risorse che include, indipendentemente da chi le crea. Il proprietario della risorsa è l'account dell'entità principale, l'utente root dell'account, un IAM utente o un ruolo che autentica la richiesta di creazione della risorsa. Negli esempi seguenti viene illustrato il funzionamento:

- Se utilizzi le credenziali dell'utente root del tuo account per creare una regola, quest'ultimo è il proprietario della risorsa. EventBridge
- Se crei un utente nel tuo account e concedi le autorizzazioni per creare EventBridge risorse a quell'utente, l'utente può creare EventBridge risorse. Tuttavia, il tuo account, a cui appartiene l'utente, possiede le EventBridge risorse.
- Se crei un IAM ruolo nel tuo account con le autorizzazioni per creare EventBridge risorse, chiunque possa assumere il ruolo può creare EventBridge risorse. Il tuo account, a cui appartiene il ruolo, possiede le EventBridge risorse.

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**
>
> In questa sezione viene illustrato l'utilizzo IAM nel contesto di EventBridge. Non fornisce informazioni dettagliate sul IAM servizio. Per la IAM documentazione completa, vedi Cos'èIAM? nella Guida IAM per l'utente. Per informazioni sulla sintassi e le descrizioni delle IAM politiche, vedere il riferimento alle IAM politiche nella Guida per l'IAMutente.

Le politiche associate a un'IAMidentità vengono definite politiche basate sull'identità (politiche) e le IAM politiche allegate a una risorsa sono denominate politiche basate sulle risorse. In EventBridge, è possibile utilizzare sia politiche basate sull'identità (politiche) che politiche basate sulle risorse. IAM

Argomenti

- Politiche basate sull'identità (politiche) IAM
- Politiche (politiche) basate sulle risorse IAM

Politiche basate sull'identità (politiche) IAM

È possibile allegare politiche alle identità. IAM Ad esempio, puoi eseguire le operazioni seguenti:

- Allega una politica di autorizzazioni a un utente o a un gruppo nel tuo account: per concedere a un utente l'autorizzazione a visualizzare le regole nella CloudWatch console Amazon, allega una politica di autorizzazioni a un utente o gruppo a cui appartiene l'utente.

- Allega una politica di autorizzazioni a un ruolo (concedi autorizzazioni per più account): puoi allegare una politica di autorizzazioni basata sull'identità a un ruolo per concedere autorizzazioni per più account. IAM Ad esempio, l'amministratore dell'account A può creare un ruolo per concedere autorizzazioni su più account a un altro account B o a un servizio nel modo seguente: AWS

  1. L'amministratore dell'account A crea un IAM ruolo e attribuisce una politica di autorizzazioni al ruolo che concede l'autorizzazione sulle risorse dell'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 dell'account B. In questo modo gli utenti dell'account B possono creare o accedere alle risorse nell'account A. Il responsabile della politica di fiducia può anche essere un responsabile del servizio che concede a un AWS AWS servizio l'autorizzazione necessaria per assumere il ruolo.

  Per ulteriori informazioni sull'utilizzo per IAM delegare le autorizzazioni, vedere Gestione degli accessi nella Guida per l'utente. IAM

Puoi creare IAM politiche specifiche per limitare le chiamate e le risorse a cui gli utenti del tuo account hanno accesso e quindi allegare tali politiche agli utenti. Per ulteriori informazioni su come creare IAM ruoli e per esplorare esempi di dichiarazioni IAM politiche EventBridge, consultaGestione delle autorizzazioni di accesso alle tue risorse Amazon EventBridge.

Politiche (politiche) basate sulle risorse IAM

Quando una regola viene eseguita EventBridge, vengono richiamate tutte le destinazioni associate alla regola, il che significa richiamare le AWS Lambda funzioni, pubblicare SNS sugli argomenti di Amazon o inoltrare l'evento ai flussi di Amazon Kinesis. Per effettuare API chiamate sulle risorse di tua proprietà, EventBridge è necessaria l'autorizzazione appropriata. Per le risorse LambdaSNS, Amazon e Amazon, EventBridge utilizza SQS politiche basate sulle risorse. Per gli stream Kinesis, EventBridge utilizza i ruoli. IAM

Per ulteriori informazioni su come creare IAM ruoli e per esplorare esempi di dichiarazioni politiche basate sulle risorse, consulta. EventBridge Utilizzo di politiche basate sulle risorse per Amazon EventBridge

Specificare elementi delle policy: azioni, effetti e principali

Per ogni EventBridge risorsa, EventBridge definisce un insieme di API operazioni. Per concedere le autorizzazioni per queste API operazioni, EventBridge definisce una serie di azioni che è possibile specificare in una politica. Alcune API operazioni richiedono le autorizzazioni per più di un'azione per eseguire l'APIoperazione. Per ulteriori informazioni su risorse e API operazioni, vedere EventBridge risorse eRiferimento alle EventBridge autorizzazioni Amazon.

Di seguito sono elencati gli elementi di base di una policy:

- Risorsa: utilizza un Amazon Resource Name (ARN) per identificare la risorsa a cui si applica la policy. Per ulteriori informazioni, consulta EventBridge risorse.
- 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.
- Principio: nelle politiche basate sull'identità (IAMpolicy), l'utente a cui è associata la policy 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 ulteriori informazioni sulla sintassi e sulle descrizioni delle politiche, vedere IAM IAMJSONPolicy Reference nella Guida per l'utente. IAM

Per informazioni sulle EventBridge API azioni e sulle risorse a cui si applicano, vedereRiferimento alle EventBridge autorizzazioni Amazon.

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 sulla specificazione delle condizioni in un linguaggio di policy, consulta Condition nella Guida per l'IAMutente.

Per definire le condizioni, si utilizzano chiavi di condizione. Esistono chiavi di AWS condizione e chiavi EventBridge specifiche che è possibile utilizzare a seconda delle esigenze. Per un elenco completo delle AWS chiavi, consulta Available Keys for Conditions nella Guida per l'IAMutente. Per

un elenco completo di tasti EventBridge specifici, vedereUtilizzo IAM delle condizioni di polizza in Amazon EventBridge.

Utilizzo di politiche (IAMpolitiche) basate sull'identità per Amazon EventBridge

Le politiche basate sull'identità sono politiche di autorizzazione che è possibile allegare alle identità. IAM

AWS politiche gestite per EventBridge

AWS affronta molti casi d'uso comuni fornendo IAM politiche 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, consulta le politiche AWS gestite nella Guida per l'IAMutente.

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 nel tuo account per le destinazioni. API Questa autorizzazione concede solo le autorizzazioni IAM di servizio per creare un ruolo nel tuo account specifico per API le destinazioni.
- `iam:PassRole`— EventBridge richiede questa autorizzazione per passare un ruolo di invocazione per richiamare l' EventBridge 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 Destinations. API

Di seguito viene illustrata la politicaJSON. 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 .

Di seguito viene JSON illustrata 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.

IAMruoli per l'invio di eventi

Per inoltrare gli eventi agli obiettivi, EventBridge ha bisogno di un IAM ruolo.

Per creare un IAM ruolo a cui inviare eventi EventBridge

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Per creare un IAM ruolo, segui i passaggi descritti in Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'IAMutente. 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 IAM politiche personalizzate per consentire le EventBridge autorizzazioni per azioni e risorse. È possibile allegare queste politiche personalizzate agli IAM utenti o ai gruppi che richiedono tali autorizzazioni. Per ulteriori informazioni sulle IAM politiche, vedere Panoramica delle

IAM politiche nella Guida per l'IAMutente. Per ulteriori informazioni sulla gestione e la creazione IAM di politiche personalizzate, vedere Managing IAM Policies nella Guida IAM per l'utente.

Autorizzazioni necessarie per accedere EventBridge agli obiettivi utilizzando i ruoli IAM

EventBridge gli obiettivi in genere richiedono IAM ruoli che concedono il EventBridge permesso di 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.

Le destinazioni Amazon SQSSNS, Amazon, Lambda, CloudWatch Logs e EventBridge bus non utilizzano ruoli e le autorizzazioni EventBridge devono essere concesse tramite una politica delle risorse. APIGli obiettivi Gateway possono utilizzare politiche o ruoli relativi alle risorse. IAM

Se la destinazione è una API destinazione, il ruolo specificato deve includere la seguente politica.

```
{
    "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 l'obiettivo è un'ECSattività Amazon, il ruolo specificato deve includere la seguente politica.

```
{
    "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 politica consente agli obiettivi integrati EventBridge di eseguire EC2 azioni Amazon per tuo conto. Devi utilizzare il per AWS Management Console 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 si utilizza il EventBridge API AWS SDKs, o il AWS CLI.

È possibile concedere agli utenti l'accesso a EventBridge regole specifiche impedendo loro di accedere ad altre regole. A tal fine, contrassegnate entrambi i set di regole e quindi utilizzate IAM politiche che fanno riferimento a tali tag. Per ulteriori informazioni sull'etichettatura EventBridge delle risorse, consultaEtichettare le risorse in Amazon EventBridge.

È possibile concedere una IAM politica a un utente per consentire l'accesso solo alle regole con un tag particolare. 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 delle dichiarazioni IAM politiche, vedere Controllare l'accesso utilizzando le politiche nella Guida IAM per l'utente.

EventBridge Aggiornamenti Amazon alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti alle politiche AWS gestite da EventBridge quando questo servizio ha iniziato a tracciare queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al RSS feed 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
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

Modifica	Descrizione	Data
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
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 API sulla destinazione. 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 di inclusione `iam:CreateServiceLinkedRole` e AWS Secrets Manager le autorizzazioni necessarie per l'utilizzo API delle destinazioni. 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 politiche basate sulle risorse per Amazon EventBridge

Quando viene eseguita una regola EventBridge, vengono richiamate tutte le destinazioni associate alla regola. Le regole possono richiamare AWS Lambda funzioni, pubblicare SNS argomenti su Amazon o inoltrare l'evento ai flussi Kinesis. Per effettuare API chiamate contro le risorse di tua proprietà, EventBridge sono necessarie le autorizzazioni appropriate. Per le risorse Lambda, AmazonSQS, SNS Amazon e Amazon CloudWatch Logs, EventBridge utilizza politiche basate sulle risorse. Per gli stream Kinesis, EventBridge utilizza policy basate sull'identità.

Lo usi per aggiungere autorizzazioni AWS CLI ai tuoi obiettivi. Per informazioni su come installare e configurare AWS CLI, consulta Getting Set Up with the AWS Command Line Interface nella Guida per l'AWS Command Line Interface utente.

Argomenti

- Autorizzazioni Amazon API Gateway
- CloudWatch Registra le autorizzazioni
- AWS Lambda autorizzazioni
- SNSAutorizzazioni Amazon
- SQSAutorizzazioni Amazon
- EventBridge Specifiche dei tubi

Autorizzazioni Amazon API Gateway

Per richiamare il tuo endpoint Amazon API Gateway utilizzando una EventBridge regola, aggiungi la seguente autorizzazione alla policy del tuo endpoint API Gateway.

```
{
  "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"
            ]
        }
    ]
}
```

CloudWatch Registra le autorizzazioni

Quando CloudWatch Logs è l'obiettivo di una regola, EventBridge crea flussi di log e CloudWatch Logs memorizza il testo degli eventi come voci di registro. EventBridge Per consentire la creazione del flusso di log e la registrazione degli eventi, CloudWatch Logs deve includere una politica basata sulle risorse che consenta la scrittura nei registri. EventBridge CloudWatch

Se si utilizza AWS Management Console per aggiungere i CloudWatch log come obiettivo di una regola, la policy basata sulle risorse viene creata automaticamente. Se si utilizza il AWS CLI per aggiungere la destinazione e la politica non esiste già, è necessario crearla.

L'esempio seguente consente di EventBridge 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, consultate PutResourcePolicyla guida CloudWatch Logs API Reference.

AWS Lambda autorizzazioni

Per richiamare la tua AWS Lambda funzione utilizzando una EventBridge regola, aggiungi la seguente autorizzazione alla policy della tua 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 di EventBridge richiamare le funzioni Lambda utilizzando AWS CLI

- Al prompt dei comandi, inserire 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 delle autorizzazioni che consentono di EventBridge richiamare le funzioni Lambda, consulta Using Lambda with AddPermissionScheduled Events nella Developer Guide.AWS Lambda

SNSAutorizzazioni Amazon

EventBridge Per consentire la pubblicazione su un SNS argomento Amazon, usa i `aws sns set-topic-attributes` comandi `aws sns get-topic-attributes` e.

> **Note**
>
> Non puoi utilizzare i `Condition` blocchi nelle politiche SNS tematiche di Amazon per EventBridge.

Per aggiungere autorizzazioni che consentano EventBridge di pubblicare argomenti SNS

1. Per elencare gli attributi di un SNS argomento, utilizzare 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 SNS argomento.

   ```
   {
       "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. Utilizzate un convertitore JSON da una stringa all'altra 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, consulta l'SetTopicAttributesazione in Amazon Simple Notification Service API Reference.

SQSAutorizzazioni Amazon

Per consentire a una EventBridge regola di richiamare una SQS coda Amazon, usa i comandi `aws sqs get-queue-attributes` and`aws sqs set-queue-attributes`.

Se la policy per la SQS coda è vuota, devi prima creare una policy e poi aggiungervi la dichiarazione di autorizzazione. Una nuova SQS coda ha una politica vuota.

Se la SQS coda ha già una politica, è necessario copiare la politica originale e combinarla con una nuova istruzione per aggiungervi l'istruzione di autorizzazione.

Per aggiungere autorizzazioni che consentano alle EventBridge regole di richiamare una coda SQS

1. Per SQS elencare gli attributi della coda. Al prompt dei comandi, inserire 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. Utilizzate un convertitore JSON da una stringa all'altra 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, consulta Amazon SQS Policy Examples nella Amazon Simple Queue Service Developer Guide.

EventBridge Specifiche dei tubi

EventBridge Pipes non supporta politiche basate sulle risorse e non APIs supporta condizioni politiche basate sulle risorse.

Tuttavia, se si configura l'accesso tramite pipe tramite un endpoint di interfaccia, tale VPC VPC endpoint supporta politiche di risorse che consentono di gestire l'accesso a Pipe. EventBridge APIs Per ulteriori informazioni, consulta the section called “Endpoint di interfaccia VPC”

Prevenzione interservizio confusa su più servizi in Amazon EventBridge

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel AWS, l'impersonificazione tra servizi può portare al confuso problema del vicesceriffo. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Ti consigliamo di utilizzare le chiavi di contesto aws:SourceArne aws:SourceAccountglobal condition nelle politiche delle risorse per limitare le autorizzazioni che Amazon EventBridge concede a un altro servizio alla risorsa. Utilizza aws:SourceArn se desideri consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizza aws:SourceAccount se desideri consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il modo più efficace per proteggersi dal confuso problema del vice è utilizzare la chiave di contesto ARN della condizione aws:SourceArn globale con l'intera risorsa. Se non conosci la dimensione completa ARN della risorsa o se stai specificando più risorse, usa la chiave aws:SourceArn global context condition con caratteri jolly (*) per le parti sconosciute di. ARN Ad esempio, arn:aws:*servicename*:*:*123456789012*:*.

Se il aws:SourceArn valore non contiene l'ID dell'account, ad esempio un bucket Amazon S3ARN, devi utilizzare entrambe le chiavi di contesto della condizione globale per limitare le autorizzazioni.

Bus di eventi

Per gli obiettivi delle regole del bus degli EventBridge eventi, il valore di aws:SourceArn deve essere la regola. ARN

L'esempio seguente mostra come utilizzare le chiavi di contesto aws:SourceArn e aws:SourceAccount global condition EventBridge per evitare il confuso problema del vice. Questo esempio è destinato all'uso in una politica di fiducia per i ruoli, per un ruolo utilizzato da una EventBridge regola.

```
{
  "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 Tubi

Per EventBridge i tubi, il valore di `aws:SourceArn` deve essere il tuboARN.

L'esempio seguente mostra come utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` global condition EventBridge per evitare il confuso problema del vice. Questo esempio è destinato all'uso in una politica di fiducia per i 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"
      }
    }
  }
}
```

Politiche basate sulle risorse per gli schemi Amazon EventBridge

Il registro degli EventBridge schemi supporta politiche basate sulle risorse. Una politica basata sulle risorse è una politica 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 sugli EventBridge schemi e sulle politiche basate sulle risorse, vedere quanto segue.

- Riferimento Amazon EventBridge Schemas REST API
- Politiche basate sull'identità e politiche basate sulle risorse nella guida per l'utente IAM

Supportato per APIs le politiche basate sulle risorse

È possibile utilizzare quanto segue APIs con le politiche basate sulle risorse per il registro degli EventBridge schemi.

- `DescribeRegistry`
- `UpdateRegistry`
- `DeleteRegistry`
- `ListSchemas`
- `SearchSchemas`
- `DescribeSchema`
- `CreateSchema`
- `DeleteSchema`
- `UpdateSchema`
- `ListSchemaVersions`
- `DeleteSchemaVersion`
- `DescribeCodeBinding`
- `GetCodeBindingSource`
- `PutCodeBinding`

Esempio di politica che concede tutte le azioni supportate a un account AWS

Per il registro EventBridge dello schema, è necessario allegare sempre una politica basata sulle risorse a un registro. Per concedere l'accesso a uno schema, è necessario specificare lo schema ARN e il registro ARN nella politica.

Per concedere a un utente l'accesso a tutti EventBridge gli schemi disponibiliAPIs, utilizza una politica simile alla seguente, sostituendola `"Principal"` con l'ID dell'account a cui desideri 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 politica che concede azioni di sola lettura a un account AWS

L'esempio seguente concede l'accesso a un account solo per gli schemi in sola lettura. APIs EventBridge

```
{
    "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 politiche basate sulle risorse con il registro degli EventBridge schemi per concedere l'accesso a un'organizzazione. Per ulteriori informazioni, consulta la Guida per l'utente 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 EventBridge autorizzazioni Amazon

Per specificare un'azione in una EventBridge politica, utilizzate il `events:` prefisso seguito dal nome dell'APIoperazione, come illustrato nell'esempio seguente.

```
"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 EventBridge API azioni, utilizzate il carattere * jolly come segue.

```
"Action": "events:*"
```

La tabella seguente elenca le EventBridge API operazioni e le azioni corrispondenti che è possibile specificare in una IAM politica.

EventBridge APIoperazione	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.
DescribeRule	`events:DescribeRule`	Necessario per elencare i dettagli di una regola.

EventBridge APIoperazione	Autorizzazioni richieste	Descrizione
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 EventBridge risorsa. 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.
PutTargets	`events:PutTargets`	Necessario per aggiungere target a una regola.

EventBridge APIoperazione	Autorizzazioni richieste	Descrizione
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 IAM delle condizioni di polizza in Amazon EventBridge

Per concedere le autorizzazioni, si utilizza il linguaggio delle IAM politiche in una dichiarazione politica per specificare le condizioni in cui una politica deve avere effetto. 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 si specificano più condizioni o chiavi in un'unica condizione, tutte le condizioni e le chiavi devono essere soddisfatte per EventBridge concedere l'autorizzazione. Se si specifica una singola condizione con più valori per una chiave, EventBridge concede l'autorizzazione se uno dei valori è soddisfatto.

Puoi anche utilizzare segnaposto o variabili di policy quando specifichi le condizioni. Per ulteriori informazioni, consulta Policy Variables nella Guida per l'IAMutente. Per ulteriori informazioni sulla specificazione delle condizioni in un linguaggio di IAM policy, vedere Condition nella Guida per l'IAMutente.

Per impostazione predefinita, IAM gli utenti e i ruoli non possono accedere agli eventi del tuo account. Per accedere agli eventi, un utente deve essere autorizzato all'`PutRule`APIazione. Se un IAM utente o un ruolo è autorizzato per l'`events:PutRule`azione, può creare una regola che corrisponda a determinati eventi. Tuttavia, affinché la regola sia utile, l'utente deve disporre anche delle autorizzazioni per l'`events:PutTargets`azione perché, se vuoi che la regola faccia qualcosa di più della pubblicazione di una CloudWatch metrica, devi anche aggiungere un obiettivo a una regola.

È possibile fornire una condizione nella dichiarazione politica di un IAM utente o di un ruolo che consenta all'utente o al ruolo di creare una regola che corrisponda solo a un insieme specifico di fonti 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`.

Allo stesso modo, puoi fornire una condizione nella dichiarazione politica di un IAM utente o di un ruolo che consenta all'utente o al ruolo di creare una regola che corrisponda solo a una risorsa specifica dei tuoi account. Per concedere l'accesso a una risorsa specifica, utilizza la chiave di condizione `events:TargetArn`.

L'esempio seguente è una politica che consente agli utenti di accedere a tutti gli eventi tranne EC2 gli eventi di Amazon EventBridge utilizzando un'istruzione di rifiuto sull'`PutRule`APIazione.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyPutRuleForAllEC2Events",
            "Effect": "Deny",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

EventBridge tasti di condizione

La tabella seguente mostra le chiavi di condizione e le coppie chiave/valore che è possibile utilizzare in una politica in EventBridge.

Chiave di condizione	Coppia chiave-valore	Tipi di valutazione
Leggi: SourceAccount	L'account in cui esiste la regola specificata da `aws:SourceArn` .	ID account, Null
seghe: SourceArn	La ARN regola che sta inviando l'evento.	ARN, Nullo
eventi: creatorAccount	`"events:creatorAccount":"` *`creatorAccount`* `"` In *`creatorAccount`* , utilizza l'ID dell'account che ha creato la regola. Utilizza questa condizione per autorizzare le API chiamate alle regole da un account specifico.	creatorAccount, Nullo

Chiave di condizione	Coppia chiave-valore	Tipi di valutazione
events:detail-type	`"events:detail-type":" detail-type "` Dove *detail-type* è la stringa letterale per il campo del tipo di dettaglio dell'evento, ad esempio and. `"AWS API Call via CloudTrail" "EC2 Instance State-change Notification"`	Detail-type, null
eventi: dettaglio. eventTypeCode	`"events:detail.eventTypeCode":" eventTypeCode "` In *eventTypeCode* , usa la stringa letterale per i dettagli. eventTypeCodecampo dell'evento, ad esempio`"AWS_ABUSE_DOS_REPORT"` .	eventTypeCode, Nullo
events: detail.service	`"events:detail.service":" service"` In *service*, utilizzate la stringa letterale per il campo detail.service dell'evento, ad esempio. `"ABUSE"`	service, Null

Chiave di condizione	Coppia chiave-valore	Tipi di valutazione
eventi: dettaglio. userIdentity. principalId	`"events:detail.userIdentity.principalId":" principal-id "` In *principal-id* , usa la stringa letterale per i dettagli. userIdentity. principalIdcampo dell'evento con un tipo di dettaglio `"AWS API Call via CloudTrail"` come. `"AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName."`	Principal Id, null
eventi: eventBusInvocation	`"events:eventBusInvocation":" boolean"` In *boolean*, usa true quando una regola invia un evento a una destinazione che è un bus di eventi in un altro account. Utilizza false quando viene utilizzata una `PutEvents` API chiamata.	eventBusInvocation, Nullo
eventi: ManagedBy	Utilizzato internamente dai AWS servizi. Per una regola creata da un AWS servizio per conto dell'utente, il valore è il nome 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 "` Utilizzo *source* per la stringa letterale per il campo di origine dell'evento, ad esempio `"aws.ec2" o"aws.s3"`. Per ulteriori valori possibili per *source*, vedi gli eventi di esempio in Eventi AWS relativi ai servizi in Amazon EventBridge.	Source, null
eventi: TargetArn	`"events:TargetArn" :" target-arn "` In *target-arn* , usa ARN il target per la regola, ad esempio `"arn:aws: lambda:*:*:functio n:*"` .	ArrayOfARN, Nullo

Ad esempio, dichiarazioni politiche per EventBridge, vedere Gestione delle autorizzazioni di accesso alle tue risorse Amazon EventBridge.

Argomenti

- EventBridge Specifiche dei tubi
- 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 AWS CloudTrail gli eventi per API le chiamate provenienti da un determinato PrincipalId utente
- Esempio: limitazione dell'accesso alle destinazioni

EventBridge Specifiche dei tubi

EventBridge Pipes non supporta alcuna chiave aggiuntiva IAM relativa alle condizioni della policy.

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 la chiamata proviene da una destinazione o da una richiesta tra più account. `PutEvents` API 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 falso quando la chiamata risulta da una richiesta. `PutEvents` API 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

I seguenti criteri di esempio possono essere allegati a un IAM utente. La politica A consente l'`PutRule`APIazione per tutti gli eventi, mentre la politica B lo consente `PutRule` solo se il modello di evento della regola creata corrisponde EC2 agli eventi di Amazon.

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" ]
}
```

In sostanza, l'utente con Policy B può solo creare una regola che corrisponda agli eventi provenienti da AmazonEC2; quindi, gli è consentito solo l'accesso agli eventi di AmazonEC2.

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
`{`	Sì	Sì

Modello di eventi	Consentito dalla policy A	Consentito dalla policy B
`"source": [ "aws.ec2" ] }`		
`{ "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 politica consente a un IAM utente o a un ruolo di creare una regola in cui l'origine in `EventPattern` è Amazon EC2 o AmazonECS.

```
{
    "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 evento	Consentito dalla policy
`{ "source": [ "aws.ec2" ] }`	Sì
`{ "source": [ "aws.ecs" ] }`	Sì
`{ "source": [ "aws.s3" ] }`	No
`{ "source": [ "aws.ec2", "aws.ecs" ] }`	No
`{ "detail-type": [ "AWS API Call via CloudTrail" ] }`	No

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 evento	Consentito dalla policy
`{ "source": [ "aws.ec2" ] }`	No
`{ "source": [ "aws.ecs" ] }`	No
`{`	Sì

Modello di evento	Consentito dalla policy
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	
`{ "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 politica, un IAM utente o un ruolo non può creare una regola con una `EventPattern` che non fornisce una fonte 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
`{ "detail-type": [ "EC2 Instance State-change Notification" ]`	No

Modello di eventi	Consentito dalla policy
}	

Esempio: limitazione dell'accesso **PutRule** mediante **detail.service**

Puoi limitare un IAM utente o un ruolo alla creazione di regole solo per eventi che hanno un determinato valore nel `events:details.service` campo. Il valore di `events:details.service` non è necessariamente il nome di un AWS servizio.

Questa condizione politica è utile quando si lavora con eventi relativi alla sicurezza o agli abusi. AWS Health Utilizzando questa condizione di policy, puoi limitare l'accesso a questi avvisi sensibili solo agli utenti che necessitano di visualizzarli.

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**

È possibile limitare un IAM utente o un ruolo alla creazione di regole solo per eventi che hanno un determinato valore nel `events:details.eventTypeCode` campo. Questa condizione politica è utile quando si lavora con eventi relativi alla sicurezza o all'abuso. AWS Health Utilizzando questa

condizione di policy, puoi limitare l'accesso a questi avvisi sensibili solo agli utenti che necessitano di visualizzarli.

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 AWS CloudTrail gli eventi per API le chiamate provenienti da un determinato **`PrincipalId`** utente

Tutti AWS CloudTrail gli eventi hanno il nome PrincipalId dell'utente che ha effettuato la API chiamata nel `detail.userIdentity.principalId` percorso di un evento. Utilizzando la chiave di `events:detail.userIdentity.principalId` condizione, è possibile limitare l'accesso di IAM utenti o ruoli agli CloudTrail eventi solo per coloro 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 evento	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

Esempio: limitazione dell'accesso alle destinazioni

Se un IAM utente o un ruolo dispone dell'`events:PutTargets`autorizzazione, può aggiungere qualsiasi oggetto appartenente allo stesso account alle regole a cui è autorizzato ad accedere. 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 AWS Identity and Access Management ruoli collegati ai servizi (IAM). Un ruolo collegato al servizio è un tipo unico di IAM ruolo a cui è collegato direttamente. EventBridge I ruoli collegati ai servizi sono predefiniti EventBridge e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per conto dell'utente. AWS

Un ruolo collegato al servizio semplifica la configurazione EventBridge perché non è necessario aggiungere manualmente le autorizzazioni necessarie. EventBridge definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. EventBridge Le autorizzazioni definite includono la politica di attendibilità e la politica di autorizzazione e tale politica di autorizzazione non può essere associata a nessun'altra entità. IAM

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Ciò protegge EventBridge le tue risorse perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta AWS i servizi che funzionano con IAM e cerca i servizi con Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Utilizzo dei ruoli per creare segreti per le destinazioni in API EventBridge

L'argomento seguente descrive in dettaglio l'utilizzo del ruolo AWSServiceRoleForAmazonEventBridgeApiDestinationscollegato al servizio.

Autorizzazioni relative ai ruoli collegati al servizio per EventBridge

EventBridge utilizza il ruolo collegato al servizio denominato AWSServiceRoleForAmazonEventBridgeApiDestinations— Consente l'accesso ai Secrets Manager Secrets creati da. EventBridge

Il ruolo AWSServiceRoleForAmazonEventBridgeApiDestinationscollegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `apidestinations.events.amazonaws.com`

La politica di autorizzazione dei ruoli denominata AmazonEventBridgeApiDestinationsServiceRolePolicyconsente 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 le autorizzazioni dei ruoli collegati ai servizi nella Guida per l'utente. 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 in AWS Management Console, il, o il AWS CLI AWS API, 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, vedi A new role appeared in my. 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 al AWSServiceRoleForAmazonEventBridgeApiDestinationsservizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. Tuttavia, è possibile modificare la descrizione del ruolo utilizzando. IAM Per ulteriori informazioni, consulta Modifica di un ruolo collegato al servizio nella Guida per l'IAMutente.

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 poter eliminare un ruolo collegato IAM al servizio, è necessario eliminare tutte le risorse utilizzate dal ruolo.

> **Note**
>
> Se il EventBridge servizio utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare EventBridge le risorse utilizzate dalla AWSServiceRoleForAmazonEventBridgeApiDestinations(console)

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. In Integrazioni scegli le API destinazioni, quindi scegli la scheda Connessioni.
3. Scegli la connessione, quindi scegli Elimina.

Per eliminare EventBridge le risorse utilizzate da AWSServiceRoleForAmazonEventBridgeApiDestinations(AWS CLI)

- Utilizzate il seguente comando:`delete-connection`.

Per eliminare EventBridge le risorse utilizzate da AWSServiceRoleForAmazonEventBridgeApiDestinations(API)

- Utilizzate il seguente comando:`DeleteConnection`.

Eliminazione manuale del ruolo collegato ai servizi

Usa la IAM console, il AWS CLI, o il AWS API per eliminare il ruolo AWSServiceRoleForAmazonEventBridgeApiDestinationscollegato al servizio. Per ulteriori informazioni, vedere Eliminazione di un ruolo collegato al servizio nella Guida per l'utente. IAM

Regioni supportate per i ruoli collegati ai servizi EventBridge

EventBridge supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta AWS Regioni ed endpoint.

Utilizzo dei ruoli per l'individuazione dello schema in Amazon EventBridge

L'argomento seguente descrive in dettaglio l'utilizzo del ruolo AWSServiceRoleForSchemascollegato al servizio.

Autorizzazioni relative ai ruoli collegati al servizio per EventBridge

EventBridge utilizza il ruolo collegato al servizio denominato AWSServiceRoleForSchemas— Concede le autorizzazioni alle regole gestite create dagli schemi.. Amazon EventBridge

Il ruolo AWSServiceRoleForSchemascollegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `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 le autorizzazioni dei ruoli collegati ai servizi nella Guida per l'utente. 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 nel AWS Management Console, o il AWS CLI AWS API, 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, vedi A new role appeared in my. 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 al AWSServiceRoleForSchemasservizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. Tuttavia, è possibile modificare la descrizione del ruolo utilizzando. IAM Per ulteriori informazioni, consulta Modifica di un ruolo collegato al servizio nella Guida per l'IAMutente.

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 poter eliminare un ruolo collegato IAM al servizio, è necessario eliminare tutte le risorse utilizzate dal ruolo.

> **Note**
>
> Se il EventBridge servizio utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare EventBridge le risorse utilizzate dalla 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 EventBridge le risorse utilizzate da AWSServiceRoleForSchemas(AWS CLI)

- Utilizzate il seguente comando:`delete-discoverer`.

Per eliminare EventBridge le risorse utilizzate da AWSServiceRoleForSchemas(API)

- Utilizzate il seguente comando:`DeleteDiscoverer`.

Eliminazione manuale del ruolo collegato ai servizi

Usa la IAM console, il AWS CLI, o il AWS API per eliminare il ruolo AWSServiceRoleForSchemascollegato al servizio. Per ulteriori informazioni, vedere Eliminazione di un ruolo collegato al servizio nella Guida per l'utente. IAM

Regioni supportate per i ruoli collegati ai servizi EventBridge

EventBridge supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta AWS Regioni ed endpoint.

Registrazione delle Amazon EventBridge API chiamate tramite AWS CloudTrail

Amazon EventBridge è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o un AWS servizio. CloudTrail acquisisce tutte le API chiamate EventBridge come eventi. Le chiamate acquisite includono chiamate dalla EventBridge console e chiamate in codice alle EventBridge API operazioni. 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 di IAM Identity Center.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

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 sono previsti 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 Lake Event Data Store

CloudTrail Lake ti consente di eseguire query SQL basate sui tuoi eventi. CloudTrail Lake converte gli eventi esistenti in JSON formato basato su righe in formato Apache. ORC ORCè un formato di archiviazione colonnare ottimizzato per il recupero rapido dei dati. Gli eventi vengono aggregati in archivi di dati degli eventi, che sono raccolte di eventi immutabili basate sui criteri selezionati applicando i selettori di eventi avanzati. I selettori applicati a un archivio di dati degli eventi controllano quali eventi persistono e sono disponibili per l'esecuzione della query. Per ulteriori informazioni su CloudTrail Lake, consulta Working with AWS CloudTrail Lake nella Guida per l'utente.AWS CloudTrail

CloudTrail Gli archivi e le richieste di dati sugli eventi di Lake comportano dei costi. Quando crei un datastore di eventi, scegli l'opzione di prezzo da utilizzare per tale datastore. L'opzione di prezzo determina il costo per l'importazione e l'archiviazione degli eventi, nonché il periodo di conservazione predefinito e quello massimo per il datastore di eventi. Per ulteriori informazioni sui CloudTrail prezzi, consulta Prezzi.AWS CloudTrail

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 AWS CLI console o CloudTrail API le operazioni. 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 o. AWS CLI CloudTrail APIs La CloudTrail colonna Dati APIs registrati mostra le API chiamate registrate per il tipo di risorsa. CloudTrail

Tipo di evento di dati (console)	valore resources.type	Dati registrati APIs 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	Dati registrati APIs su CloudTrail
		• UpdatePipe

Puoi configurare selettori di eventi avanzati per filtrare `resources.ARN` i campi `eventNamereadOnly`, e per registrare solo gli eventi che ritieni importanti. Per ulteriori informazioni su questi campi, vedere AdvancedFieldSelectornella Guida di AWS CloudTrail APIriferimento.

EventBridge eventi gestionali in CloudTrail

Gli eventi di gestione forniscono informazioni sulle operazioni di gestione eseguite sulle risorse di 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 la registrazione CloudTrail, vedere il Amazon EventBridge API riferimento.

EventBridge esempi di eventi

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'APIoperazione richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia stack ordinata delle API chiamate 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, vedere 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 IAM ruolo 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 AmazonSQS, Kinesis e DynamoDB, i campi and includeranno. `sourceIPAddress` `invokedBy` `pipes.amazonaws.com`

Esempio di CloudTrail registrazione per tutti gli arricchimenti, i target e le fonti AmazonSQS, 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

Revisori di terze parti SOCPCI, come FedRAMP, HIPAA valutano la sicurezza e la conformità dei AWS servizi nell'ambito di più programmi di AWS conformità.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, vedere AWS Servizi nell'ambito del programma di conformitàAWS . Per informazioni generali, vedere Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact Scaricamento dei . AWS

La vostra responsabilità di conformità durante l'utilizzo EventBridge è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide rapide su sicurezza e conformità Guide introduttive : considerazioni e passaggi sull'architettura per l'implementazione di ambienti di base incentrati sulla sicurezza e la conformità su. AWS
- Whitepaper sull'architettura per la HIPAA sicurezza e la conformità: in che modo le aziende possono utilizzare per creare applicazioni conformi. AWS HIPAA
- AWS Risorse per la conformità Risorse per AWS 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 panoramica completa dello stato di sicurezza AWS che consente di verificare la conformità agli standard e alle best practice del settore della sicurezza.

EventBridge Resilienza di Amazon

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

Sicurezza dell'infrastruttura in Amazon EventBridge

In quanto servizio gestito, Amazon EventBridge è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Si utilizzano API chiamate AWS pubblicate per accedere tramite EventBridge la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS) come (Ephemeral Diffie-Hellman) o DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale. IAM O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

È possibile richiamare queste API operazioni da qualsiasi posizione di rete e utilizzare politiche di accesso basate sulle risorse in EventBridge, che possono includere restrizioni basate sull'indirizzo IP di origine. Puoi anche utilizzare EventBridge le policy per controllare l'accesso da endpoint Amazon Virtual Private Cloud (AmazonVPC) specifici o specificiVPCs. In effetti, questo isola l'accesso alla rete a una determinata EventBridge risorsa solo da quelle specifiche all'VPCinterno della AWS rete.

Analisi della configurazione e delle vulnerabilità in Amazon EventBridge

La configurazione e i controlli IT sono una responsabilità condivisa tra voi AWS e voi, i nostri clienti. Per ulteriori informazioni, consulta il modello di responsabilità AWS condivisa.

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

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'inserimento dell'evento al completamento del primo tentativo di chiamata.	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

Il `AWS/Events` namespace include le seguenti metriche relative alle richieste. `PutEvents` API

Per le metriche che utilizzano Count come unità, Sum e 2 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

Il `AWS/Events` namespace include le seguenti metriche relative alle richieste. `PutPartnerEvents` API

> **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 gli argomenti di 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 API alle chiamate di servizio AWS globali, ma non è stata eseguita

AWS servizi globali; ad esempio Amazon Route 53 sono disponibili solo nella regione Stati Uniti orientali (Virginia settentrionale), quindi gli eventi derivanti dalle AWS API chiamate provenienti dai servizi globali sono disponibili solo in quella regione. IAM Per ulteriori informazioni, consulta Eventi AWS relativi ai servizi in Amazon EventBridge.

Il IAM ruolo associato alla mia regola viene ignorato quando viene eseguita la regola

EventBridge utilizza solo i IAM ruoli per le regole che inviano eventi ai flussi Kinesis. Per le regole che richiamano funzioni Lambda o argomenti SNS Amazon, devi fornire autorizzazioni basate sulle risorse.

Assicurati che gli AWS STS endpoint regionali siano abilitati, in modo che EventBridge possano utilizzarli quando assumeranno il ruolo che hai fornito. IAM 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 ARN i caratteri corretti durante la creazione dei pattern di eventi in modo che corrispondano alla ARN sintassi dell'evento da abbinare.

Alcuni eventi, come gli eventi di AWS API chiamata da 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. `FailedInvocations` CloudWatch Ti consigliamo di impostare un file DLQ per archiviare gli eventi che non è stato possibile consegnare correttamente a una destinazione. Per ulteriori informazioni, consulta Utilizzo di code di lettere non recapitate per elaborare eventi non consegnati in EventBridge

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 ACLs rileva le modifiche in un bucket Amazon S3 e quindi esegue il software per modificarle in un nuovo stato causa un ciclo infinito. Un modo per risolverlo consiste nel riscrivere la regola in modo ACLs che corrisponda solo a quelle in cattivo stato.

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 consegnati alla SQS coda Amazon di destinazione

Se la tua SQS coda Amazon è crittografata, devi creare una KMS chiave gestita dal cliente e includere la seguente sezione di autorizzazione nella tua KMS politica delle chiavi. Per ulteriori informazioni, consulta 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 mia regola viene eseguita, ma non vedo alcun messaggio pubblicato nel mio SNS argomento Amazon

Scenario 1

È necessaria l'autorizzazione per la pubblicazione dei messaggi nel tuo SNS argomento Amazon. 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 il comando 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 il tuo SNS argomento è crittografato, devi includere la sezione seguente nella tua politica KMS chiave.

```
{
  "Sid": "Allow EventBridge to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

Il mio SNS argomento Amazon 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 al tuo SNS argomento Amazon per tuo conto. Se elimini la regola poco dopo averla creata, EventBridge potresti non rimuovere l'autorizzazione dal tuo SNS argomento Amazon. 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 politiche basate sulle risorse per Amazon EventBridge.

Con quali chiavi di IAM condizione posso usare EventBridge?

EventBridge supporta le chiavi di condizione AWS-wide (vedi IAMe AWS STS condition context keys nella Guida per l'IAMutente), più le chiavi elencate inUtilizzo IAM delle condizioni di polizza in Amazon EventBridge.

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 Azioni, per Ogni volta che si verifica questo allarme, scegli Lo stato è ALARM.
9. Per Invia notifica a, seleziona un SNS argomento Amazon esistente o creane uno nuovo. Per creare un nuovo argomento, seleziona New list (Nuovo elenco). Digita un nome per il nuovo SNS argomento di Amazon, per esempio: myFailedRules.
10. Per Elenco e-mail, digita un elenco di indirizzi e-mail separati da virgole per ricevere una notifica quando lo stato dell'allarme cambia. ALARM
11. Scegli Crea allarme.

EventBridge Quote Amazon

Le seguenti quote si applicano alle varie aree di funzionalità di Amazon EventBridge.

Argomenti

> **Note**
>
> Per un elenco delle quote per EventBridge Scheduler, consulta Quotas for Scheduler nella Guida per l'utente di EventBridge Scheduler. EventBridge

EventBridge quote

EventBridge ha le seguenti quote.

La console Service Quotas fornisce informazioni sulle EventBridge quote. 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 API destinazioni 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
CreateEndpoint limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di richieste al secondo per. CreateEndpoint API Ulteriori richieste verranno sottopost e a limitazione (della larghezza di banda della rete).
DeleteEndpoint limite di accelerazione delle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di richieste al secondo per. DeleteEndpoint API 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. È possibile visualizzare la politica corrente e le relative dimensioni utilizzando il DescribeEventBus API.

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
PutEvents limite di accelerazione 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	Numero massimo di richieste al secondo per. PutEvents API 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		
Velocità di invocazioni per destinazione API	Ogni regione supportata: 300 al secondo	Sì	Il numero massimo di chiamate al secondo da inviare a ciascun endpoint di API destinazione per account per regione. Una volta raggiunta la quota, le future chiamate a quell'APIendpoint vengono 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 tranne. EventBridge API PutEvents Le richieste aggiuntive sono limitate

Nome	Predefinita	Adatta e	Descrizione
UpdateEndpoint limite di accelerazione nelle transazioni al secondo	Ogni regione supportata: 5 al secondo	No	Il numero massimo di richieste al secondo per. UpdateEndpoint API Ulteriori richieste verranno sottopost e a limitazione (della larghezza di banda della rete).

Inoltre, EventBridge dispone delle 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. È possibile visualizzare la politica attuale e le relative dimensioni utilizzando il. `DescribeE ventBus` API
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.

Nome	Predefinito	Descrizione
Regole contenenti caratteri jolly	Ogni Regione supportata: 30 regole per router di eventi	Numero massimo di regole, per router di eventi per account, che possono contenere filtri di 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.

PutPartnerEvents quote per regione

Se hai dei requisiti per limiti massimi più alti, contatta l'assistenza.

Regioni	Transazioni al secondo
• AWS GovCloud (Stati Uniti occidentali) • AWS GovCloud (Stati Uniti orientali) • 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)	PutPartnerEventsha un limite flessibile di 1.400 richieste di throughput al secondo e 3.600 richieste burst al secondo per impostazione predefinita in tutte le regioni.

Regioni	Transazioni al secondo
• Asia Pacific (Seul) • Asia Pacifico (Singapore) • Asia Pacifico (Sydney) • 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)	

EventBridge Schema: quote del registro

EventBridge Schema Registry ha le seguenti quote.

La console Service Quotas fornisce informazioni sulle EventBridge quote. 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
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)

EventBridge Tubi (quote)

EventBridge Pipes ha le seguenti quote. Se hai dei requisiti per limiti massimi più alti, contatta l'assistenza.

Risorsa	Regioni	Limite predefinito
Esecuzioni di pipe simultanee per account	• AWS GovCloud (Stati Uniti occidentali) • AWS GovCloud (Stati Uniti orientali) • Cina (Ningxia) • Cina (Pechino)	1000

Risorsa	Regioni	Limite predefinito
	• Asia Pacifico (Osaka-Locale) • 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 (Seoul) • Asia Pacifico (Mumbai) • Asia Pacifico (Hong Kong) • Medio Oriente (Bahrein) • Cina (Ningxia) • Cina (Pechino) • Asia Pacifico (Osaka-Locale) • Africa (Città del Capo) • Europa (Milano)	

Risorsa	Regioni	Limite predefinito
Esecuzioni di pipe simultanee per account	• Stati Uniti orientali (Virginia settentrionale) • US West (Oregon) • Europa (Irlanda)	3000
Pipe per account	Tutti	1000

Etichettare le risorse in Amazon EventBridge

Un tag è un'etichetta di attributo personalizzata che l'utente o AWS assegna 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, è possibile assegnare lo stesso tag a una EventBridge regola assegnata 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 della chiave del tag è di 128 caratteri Unicode in -8. UTF
  - Per ogni risorsa, la chiave di ciascun tag deve essere univoca.
  - I caratteri consentiti sono lettere, numeri, spazi rappresentabili in UTF -8 e i 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.
  - La lunghezza massima del valore del tag è di 256 caratteri Unicode in UTF -8.
  - I caratteri consentiti sono lettere, numeri, spazi rappresentabili in UTF -8 e i 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.

Per ulteriori informazioni sulla gestione dei tag, consulta Working with Tag Editor nella Resource Groups User Guide.

Aggiungere o rimuovere tag sui bus degli eventi

È possibile aggiungere o rimuovere tag sui bus degli eventi.

Per aggiungere o rimuovere tag su un bus di eventi (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 (AWS CLI)

- Per aggiungere tag, usa tag-resource.

  Per rimuovere i tag, usa untag-resource.

  È inoltre possibile determinare i tag su un bus di eventi utilizzando. list-tags-for-resource

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 iscriverti a un RSS feed.

Modifica	Descrizione	Data di rilascio
Aggiornamento della struttura del documento	Sono stati utilizzati i dati di visualizzazione delle pagine e l'analisi approfondita per ristrutturare le sezioni della documentazione e aumentare la visibilità di argomenti importanti. Guida alla navigazione aggiornata per ridurre la profondità complessiva. Argomenti correlati consolidati, se del caso.	4 agosto 2024
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 AWS CloudFormation modello da un bus di EventBridge eventi esistente	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

Modifica	Descrizione	Data di rilascio
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 AWS CloudFormation modello da un bus di EventBridge eventi esistente	18 novembre 2022
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

Modifica	Descrizione	Data di rilascio
È stata aggiornata la CloudWatchEventsReadOnlyAccess politica.	Aggiornata per la corrispondenza ad `AmazonEventBridgeReadOnlyAccess` . • AmazonEventBridgeReadOnlyAccess politica	1 dicembre 2022
Aggiornati i filtri dei contenuti in modelli di eventi.	Ora puoi utilizzare le opzioni di filtro `suffix`, `equals-ignore-case` e `$or` per creare modelli di eventi. • Utilizzo degli operatori di confronto nei modelli di EventBridge eventi di Amazon	14 novembre 2022
È stata aggiornata la AmazonEventBridgeFullAccess politica.	Sono state aggiunte le autorizzazioni necessarie per l'utilizzo di EventBridge Schema Registry and EventBridge Scheduler. • AmazonEventBridgeFullAccess politica	10 novembre 2022
È stata aggiornata la politica. AmazonEventBridgeReadOnlyAccess	È ora possibile visualizzare le informazioni sul registro degli EventBridge schemi e sull' EventBridge utilità di pianificazione. • AmazonEventBridgeReadOnlyAccess politica	10 novembre 2022
Aggiornati i filtri dei contenuti in modelli di eventi.	Ora puoi utilizzare le opzioni di filtro `suffix`, `equals-ignore-case` e `$or` per creare modelli di eventi. • Utilizzo degli operatori di confronto nei modelli di EventBridge eventi di Amazon	14 novembre 2022

Modifica	Descrizione	Data di rilascio
È stata aggiornata la AmazonEventBridgeFullAccess politica.	Sono state aggiunte le autorizzazioni necessarie per l'utilizzo di 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
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 regionali con endpoint globali in EventBridge • `CreateEndpoint`	7 aprile 2022

Modifica	Descrizione	Data di rilascio
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: • Creazione di un archivio per gli eventi in Amazon EventBridge. • `CreateArchive` • `StartReplay`	5 novembre 2020
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 non recapitate per elaborare eventi non consegnati in EventBridge. • `PutTargets`	12 ottobre 2020
È stato aggiunto il supporto per JSONSchema gli schemi di formato Draft4.	Amazon EventBridge ora supporta schemi in formato JSONSchema Draft 4. Ora puoi anche esportare gli schemi utilizzando. EventBridge API Per ulteriori informazioni, consulta quanto segue. • EventBridge Schemi Amazon • `Export`nello EventBridge Schema Registry API Reference.	28 settembre 2020

Modifica	Descrizione	Data di rilascio
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. • Politiche basate sulle risorse per gli schemi Amazon EventBridge • `Policy`nello Schema Registry Reference EventBridge API • RegistryPolicy Tipo di risorsa nella Guida AWS CloudFormation per l'utente	30 aprile 2020
Tag per bus di eventi	Questa versione consente di creare e gestire tag per bus di eventi. È possibile aggiungere tag durante la creazione di un bus di eventi e aggiungere o gestire tag esistenti chiamando il relativoAPI. Per ulteriori informazioni, consulta gli argomenti seguenti. • Etichettare le risorse in Amazon EventBridge • Politiche basate su tag in Amazon EventBridge • `TagResource` • `UntagResource` • `ListTagsForResource`	24 febbraio 2020
Aumento delle quote di servizio	Amazon EventBridge ha aumentato le quote per le invocazioni e per. `PutEvents` Le quote variano a seconda della Regione e possono essere aumentate se necessario.	11 febbraio 2020

Modifica	Descrizione	Data di rilascio
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: usa i trasformatori di input per trasforma re gli eventi in EventBridge Aggiunto un collegamento agli eventi di Application Auto Scaling. • Eventi di Application Auto Scaling e EventBridge • Eventi AWS relativi ai servizi in Amazon EventBridge	20 dicembre 2019
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 informazi oni, consulta gli argomenti seguenti. • Utilizzo degli eventi di Amazon Augmented AI • Eventi AWS relativi ai servizi in Amazon EventBridge	13 dicembre 2019

Modifica	Descrizione	Data di rilascio
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. • Automatizzazione di Amazon Chime con EventBridge • Eventi AWS relativi ai servizi in Amazon EventBridge	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 agli schemi API • 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
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.