aws

Guía del usuario

Amazon EventBridge

Amazon EventBridge: Guía del usuario

Table of Contents

¿Qué es Amazon EventBridge?

EventBridge es un servicio sin servidor que utiliza eventos para conectar los componentes de la aplicación entre sí, lo que facilita la creación de aplicaciones escalables basadas en eventos. La arquitectura basada en eventos es un estilo de creación de sistemas de software de acoplamiento flexible que funcionan juntos emitiendo eventos y respondiendo a ellos. La arquitectura basada en eventos puede ayudar a aumentar la agilidad y crear aplicaciones fiables y escalables.

Utilice EventBridge para enrutar eventos desde orígenes como aplicaciones propias, servicios de AWS y software de terceros a aplicaciones de consumo en toda su organización. EventBridge proporciona formas sencillas y coherentes de incorporar, filtrar, transformar y entregar eventos para que pueda crear nuevas aplicaciones rápidamente.

En el siguiente video, se ofrece una breve introducción a las características de Amazon EventBridge:

EventBridge incluye dos formas de procesar eventos: buses de eventos y canalizaciones.

- Los buses de eventos son enrutadores que reciben eventos y los envían a cero o más destinos. Los buses de eventos son ideales para enrutar eventos desde muchos orígenes a muchos destinos, con transformación opcional de eventos antes de entregarlos a un destino.

  En el siguiente video, se ofrece información general sobre los buses de eventos:

- Canalizaciones EventBridge Pipes está diseñado para integraciones punto a punto; cada canalización recibe eventos de un solo origen para su procesamiento y entrega a un único destino. Las canalizaciones también admiten transformaciones avanzadas y enriquecimiento de los eventos antes de entregarlos a un destino.

Las canalizaciones y los buses de eventos a menudo se usan juntos. Un caso de uso común es crear una canalización con un bus de eventos como destino; la canalización envía los eventos al bus de eventos, que luego los envía a varios destinos. Por ejemplo, puede crear una canalización con un flujo de DynamoDB como origen y un bus de eventos como destino. La canalización recibe los eventos del flujo de DynamoDB y los envía al bus de eventos, que, a su vez, los envía a varios destinos de acuerdo con las reglas especificadas en el bus de eventos.

EventBridge es la evolución de Amazon CloudWatch Events.

Anteriormente, EventBridge se llamaba Amazon CloudWatch Events. El bus de eventos predeterminado y las reglas que creó en CloudWatch Events también se muestran en la consola de EventBridge. EventBridge utiliza la misma API de CloudWatch Events, por lo que el código que usa la API de CloudWatch Events sigue siendo el mismo.

EventBridge se basa en las capacidades de CloudWatch Events con características como los eventos de socios, el registro de esquemas y EventBridge Pipes. Las nuevas características agregadas a EventBridge no se agregan a CloudWatch Events. Para obtener más información, consulte ???.

Todas las características a las que está acostumbrado en CloudWatch Events también están presentes en EventBridge, entre las que se incluyen:

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

Las características de EventBridge que se basan en las capacidades de los eventos y las amplían incluyen:

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

EventBridge Configuración y requisitos previos de Amazon

Para usar Amazon EventBridge, necesitas una AWS cuenta. Su cuenta le permite utilizar servicios como Amazon EC2 para generar eventos que puede ver en la EventBridge consola. También puede instalar y configurar AWS Command Line Interface (AWS CLI) para usar una interfaz de línea de comandos para ver los eventos.

Temas

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Signing in as the root user en la Guía del usuario de AWS Sign-In .
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz (consola) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .
2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Inicia sesión en la EventBridge consola de Amazon

Para iniciar sesión en la EventBridge consola de Amazon

- Inicia sesión en la EventBridge consola de Amazon AWS Management Console y ábrela en https://console.aws.amazon.com/events/.

Credenciales de la cuenta

Aunque puedes usar tus credenciales de usuario raíz para acceder EventBridge, te recomendamos que utilices una cuenta AWS Identity and Access Management (IAM) en su lugar. Si utilizas una cuenta de IAM para acceder EventBridge, debes tener los siguientes permisos.

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

Para obtener más información, consulte Autenticación.

Configure el AWS Command Line Interface

Puede utilizar el AWS CLI para realizar EventBridge operaciones.

Para obtener información sobre cómo instalar y configurar el AWS CLI, consulte Cómo configurar el AWS Command Line Interface en la Guía del AWS Command Line Interface usuario.

Puntos de conexión regionales

Debe habilitar los puntos finales regionales predeterminados para su uso EventBridge. Para obtener más información, consulte Activación y desactivación AWS STS en una AWS región en la Guía del usuario de IAM.

Cómo empezar con Amazon EventBridge

La base EventBridge es crear reglas que dirijan los eventos a un objetivo. En esta sección, va a crear una regla básica. Para ver tutoriales sobre escenarios específicos y destinos específicos, consulte Tutoriales de Amazon EventBridge.

Crear una regla en Amazon EventBridge

Para crear una regla para los eventos, debe especificar la acción que se realizará cuando EventBridge reciba un evento que coincida con el patrón de eventos de la regla. Cuando un evento coincide, EventBridge envía el evento al objetivo especificado y desencadena la acción definida en la regla.

Cuando un AWS servicio de tu AWS cuenta emite un evento, siempre va al bus de eventos predeterminado de tu cuenta. Para escribir una regla que coincida con los eventos de AWS los servicios de tu cuenta, debes asociarla al bus de eventos predeterminado.

Para crear una regla para un AWS servicio

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Elija Crear regla.
4. Escriba un nombre y una descripción para la regla.

   Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.
5. En Bus de eventos, seleccione el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione Bus de eventos predeterminado de AWS . Cuando un servicio de AWS en la cuenta emite un evento, siempre va al bus de eventos predeterminado de la cuenta.
6. En Tipo de regla, seleccione Regla con un patrón de eventos.
7. Seleccione Siguiente.
8. En Origen de evento, seleccione Servicios de AWS .
9. (Opcional) Para ver ejemplos de eventos, seleccione el tipo de evento.
10. En patrón de eventos, realice una de las siguientes acciones:

- Para usar una plantilla a fin de crear el patrón de eventos, seleccione Formulario de patrón de eventos y seleccione Origen de evento y Tipo de evento. Si eliges Todos los eventos como tipo de evento, todos los eventos emitidos por este AWS servicio coincidirán con la regla.

  Para personalizar la plantilla, seleccione Patron personalizado (editor JSON) y realice los cambios.
- Para utilizar un patrón de eventos personalizado, seleccione Patrón personalizado (editor JSON) y cree el patrón de eventos.

11. Seleccione Siguiente.
12. En Tipos de destino, seleccione Servicio de AWS .
13. En Seleccione un objetivo, elija el AWS servicio al que desee enviar la información cuando EventBridge detecte un evento que coincida con el patrón del evento.
14. Los campos mostrados varían en función del servicio que seleccione. Introduzca la información específica de este tipo de destino, según sea necesario.
15. Para muchos tipos de objetivos, EventBridge necesita permisos para enviar eventos al objetivo. En estos casos, EventBridge puede crear la función de IAM necesaria para que se ejecute la regla. Realice una de las siguientes acciones siguientes:

    - Para crear un rol de IAM automáticamente, seleccione Crear un nuevo rol para este recurso específico.
    - Para utilizar un rol de IAM que haya creado antes, seleccione Usar rol existente y seleccione el rol existente del menú desplegable.
16. (Opcional) En Configuración adicional, haga lo siguiente:

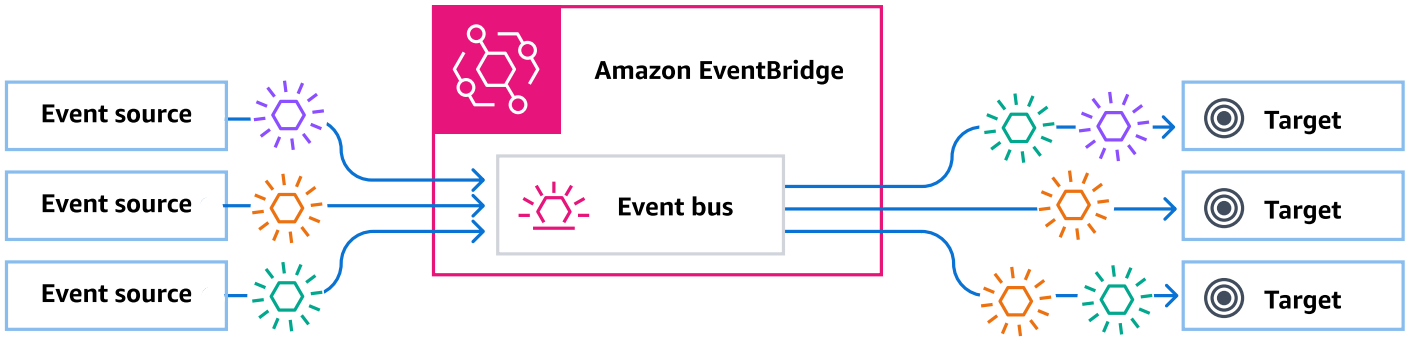
    a. En Antigüedad máxima del evento, indique un valor entre un minuto (00:01) y 24 horas (24:00).
    b. En Cantidad de reintentos, indique un número entre 0 y 185.
    c. En el caso de la cola de cartas sin salida, elija si desea utilizar una cola estándar de Amazon SQS como cola de cartas sin salida. EventBridge envía los eventos que cumplen con esta regla a la lista de espera si no se entregan correctamente al destino. Realice una de las siguientes acciones siguientes:
        - Seleccione Ninguna para no usar una cola de mensajes fallidos.

- Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para utilizarla como cola de mensajes fallidos y, a continuación, seleccione de la lista desplegable la cola que quiera usar.
- Elija Seleccione una cola de Amazon SQS en otra AWS cuenta como cola de letra muerta y, a continuación, introduzca el ARN de la cola que desee utilizar. Debe adjuntar a la cola una política basada en recursos que le conceda permiso para enviarle mensajes. EventBridge Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.

17. (Opcional) Seleccione Agregar otro destino para agregar otro destino para esta regla.
18. Seleccione Siguiente.
19. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulte EventBridge Etiquetas de Amazon.
20. Seleccione Siguiente.
21. Revise los detalles de la regla y seleccione Crear regla.

Amazon EventBridge Event Bus

Un bus de eventos es un enrutador que recibe eventos y los entrega a cero o más destinos. Los buses de eventos son ideales para direccionar eventos desde muchos orígenes a muchos destinos, con transformación opcional de eventos antes de entregarlos a un destino.



Reglas asociadas al bus de eventos evalúan los eventos a medida que llegan. Cada regla comprueba si un evento coincide con el patrón de la regla. Si el evento coincide, EventBridge envía el evento

Si asocia una regla a un bus de eventos específico, la regla solo se aplica a los eventos recibidos por ese bus de eventos.

> **Note**
>
> También puede procesar eventos mediante EventBridge Pipes. EventBridge Pipes está diseñado para point-to-point integraciones; cada canal recibe eventos de una sola fuente para su procesamiento y entrega a un único destino. Las canalizaciones también admiten transformaciones avanzadas y enriquecimiento de los eventos antes de entregarlos a un destino. Para obtener más información, consulte ???.

Temas

- Cómo funcionan los buses de eventos
- Conceptos de Amazon EventBridge Event Bus
- Crear un bus de EventBridge eventos de Amazon
- Actualización de un bus de EventBridge eventos de Amazon

- Eliminar un bus de EventBridge eventos de Amazon
- Permisos para buses de eventos de Amazon EventBridge
- Generar una plantilla de AWS CloudFormation a partir de un bus de eventos de Amazon EventBridge

Cómo funcionan los buses de eventos

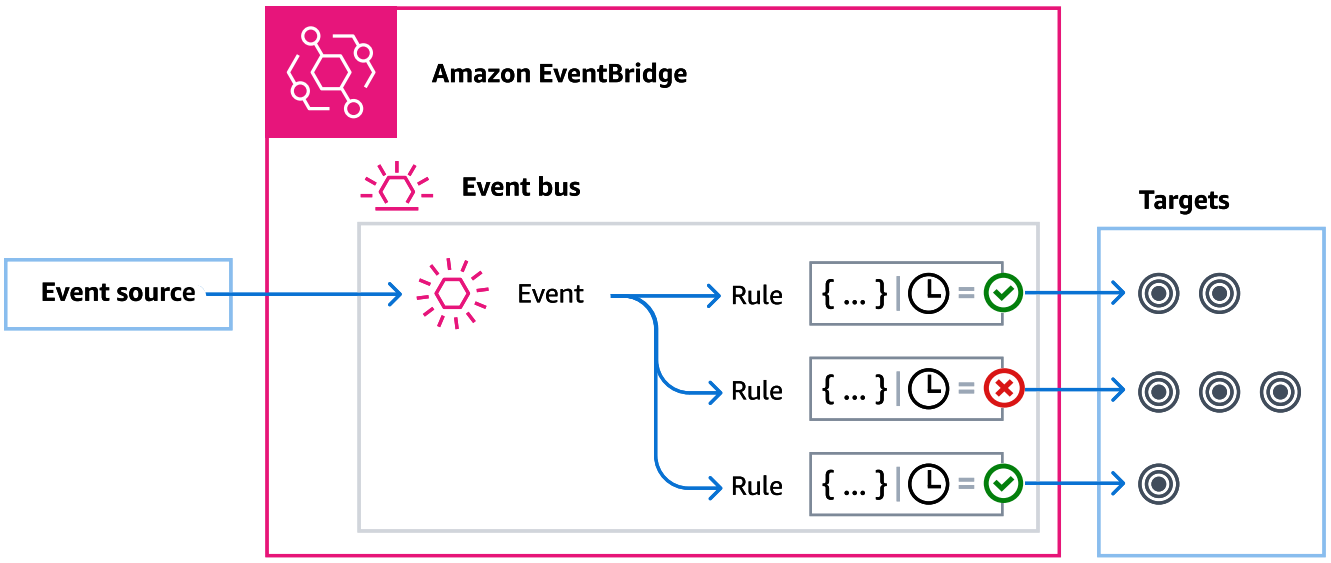
Los buses de eventos le permiten direccionar eventos desde múltiples orígenes a múltiples destinos.

En líneas generales, funciona de la siguiente manera:

1. Un origen de eventos, que puede ser un AWS servicio, una aplicación personalizada o un proveedor de SaaS, envía un evento a un bus de eventos.
2. EventBridge a continuación, evalúa el evento según cada regla definida para ese bus de eventos.

   Para cada evento que coincida con una regla, EventBridge envía el evento a los destinos especificados para esa regla. Si lo desea, como parte de la regla, también puede especificar cómo EventBridge debe transformarse el evento antes de enviarlo a los objetivos.

   Un evento puede coincidir con varias reglas y cada regla puede especificar hasta cinco destinos. (Es posible que un evento no coincida con ninguna regla, en cuyo caso no EventBridge realiza ninguna acción).



Considere un ejemplo en el que se utiliza el bus de eventos EventBridge predeterminado, que recibe automáticamente los eventos de AWS los servicios:

1. Se crea una regla en el bus de eventos predeterminado para el evento `EC2 Instance State-change Notification`:
   - Se especifica que la regla coincida con los eventos en los que una instancia de Amazon EC2 haya cambiado su `state` por `running`.

     Lo hace especificando un objeto JSON que define los atributos y valores con lo que debe coincidir un evento para activar la regla. Esto se denomina patrón de eventos.

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

   - Se especifica que el destino de la regla sea una función de Lambda determinada.
2. Cada vez que una instancia de Amazon EC2 cambia de estado, Amazon EC2 (el origen del evento) envía automáticamente ese evento al bus de eventos predeterminado.
3. EventBridge evalúa todos los eventos enviados al bus de eventos predeterminado según la regla que ha creado.

   Si el evento coincide con su regla (es decir, si el evento era una instancia de Amazon EC2 que cambiaba de estado`running`), EventBridge envía el evento al destino especificado. En este caso, es la función de Lambda.

En el siguiente vídeo se describe qué son los buses de eventos y qué hacen: Qué son los buses de eventos

En el siguiente vídeo se describen los diferentes buses de eventos y cuándo usarlos: Diferencias entre buses de eventos

Conceptos de Amazon EventBridge Event Bus

He aquí un análisis más detallado de los componentes principales de una EDA basada en buses de eventos.

Buses de eventos

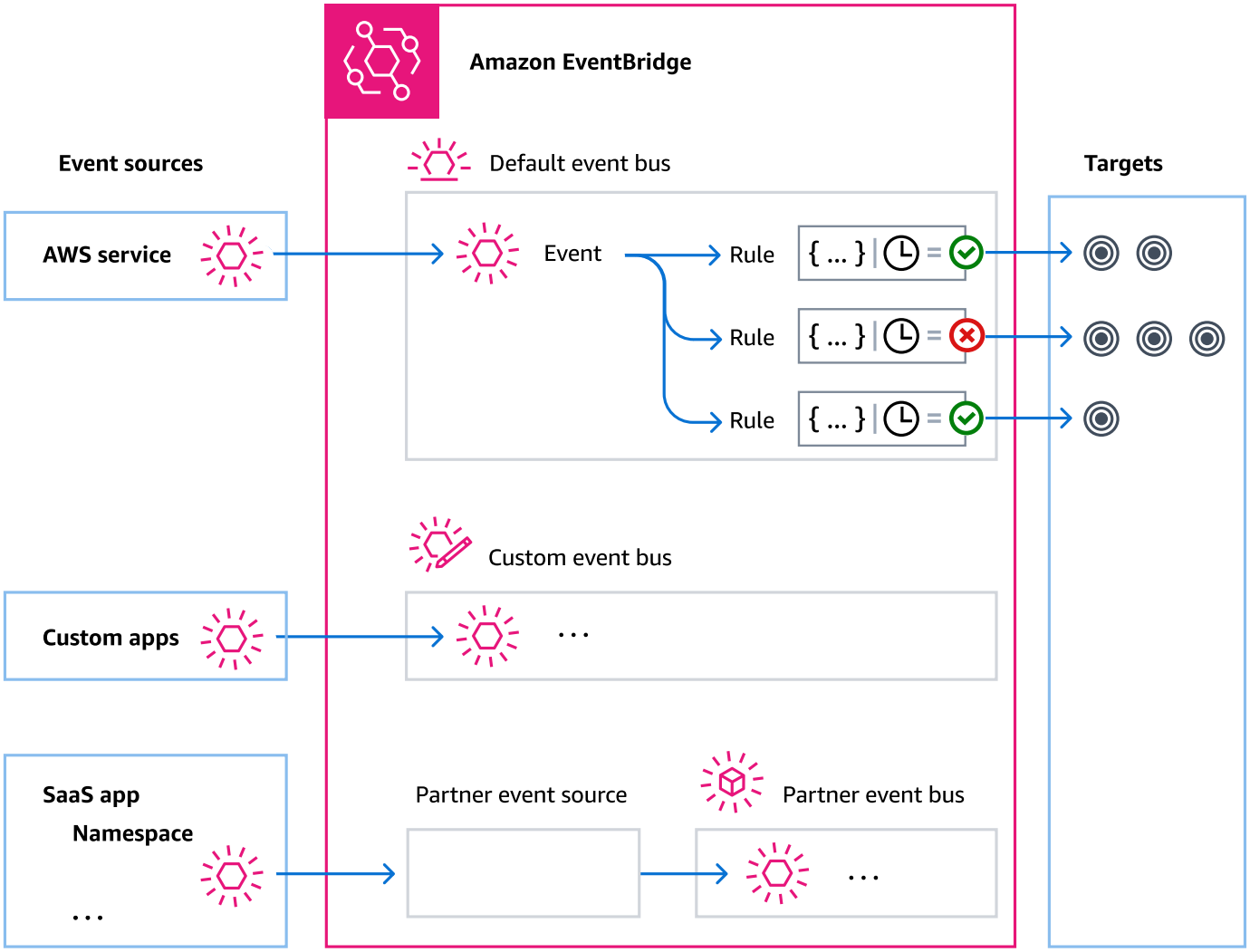
Un bus de eventos es un enrutador que recibe eventos y los entrega a cero o más destinos. Utilice un bus de eventos cuando necesite direccionar eventos desde muchos orígenes a muchos destinos, con transformación opcional de eventos antes de entregarlos a un destino.

Su cuenta incluye un bus de eventos predeterminado que recibe automáticamente los eventos de AWS los servicios. También puede:

- Crear buses de eventos adicionales, denominados buses de eventos personalizados, y especificar qué eventos reciben.
- Crear buses de eventos de socios, que reciban eventos de socios de SaaS.

Los casos de uso más comunes de buses de eventos incluyen:

- Uso de un bus de eventos como intermediario entre diferentes cargas de trabajo, servicios o sistemas.
- Uso de varios buses de eventos en sus aplicaciones para dividir el tráfico de eventos. Por ejemplo, crear un bus para procesar los eventos que contienen información de identificación personal (PII) y otro bus para los eventos que no la contienen.
- Agregación de eventos mediante el envío de eventos desde varios buses de eventos a un bus de eventos centralizado. Este bus centralizado puede estar en la misma cuenta que los otros buses, pero también puede estar en una cuenta o región diferente.



Eventos

En pocas palabras, un EventBridge evento es un objeto JSON que se envía a un bus o canal de eventos.

En el contexto de la arquitectura basada en eventos (EDA), un evento suele representar un indicador de un cambio en un recurso o entorno.

Para obtener más información, consulte ???.

Orígenes de eventos

EventBridge puede recibir eventos de fuentes de eventos, entre las que se incluyen:

- AWS servicios
- Aplicaciones personalizadas
- Software como servicio (SaaS)

Reglas

Una regla recibe eventos de entrada y los envía según corresponda a destinos para procesamiento. Puede especificar la forma en que cada regla invoca a su(s) destino(s) en función de lo siguiente:

- Un patrón de eventos, que contiene uno o más filtros para hacer coincidir los eventos. Los patrones de eventos pueden incluir filtros que coincidan con:
  - Metadatos del evento: datos sobre el evento, como el origen del evento o la cuenta o región en la que se originó el evento.
  - Datos del evento: las propiedades del evento en sí. Estas propiedades varían según el evento.
  - Contenido del evento: los valores de propiedades reales de los datos del evento.
- Una programación para invocar los destinos a intervalos regulares.

  Puede especificar una regla programada dentro EventBridge o mediante el EventBridge programador.

  > **Note**
  >
  > EventBridge ofrece Amazon EventBridge Scheduler, un programador sin servidor que le permite crear, ejecutar y gestionar tareas desde un servicio gestionado centralizado. EventBridge Scheduler es altamente personalizable y ofrece una escalabilidad mejorada en comparación con las reglas EventBridge programadas, con un conjunto más amplio de operaciones y servicios de API de destino. AWS
  > Te recomendamos que utilices EventBridge Scheduler para invocar los objetivos según un cronograma. Para obtener más información, consulte ???.

Cada regla se define para un bus de eventos específico y solo se aplica a los eventos de ese bus de eventos.

Una sola regla puede enviar un evento a un máximo de cinco destinos.

De forma predeterminada, puede configurar hasta 300 reglas por cada bus de eventos. Esta cuota se puede aumentar a miles de reglas en la consola de Service Quotas. Dado que el límite de reglas se aplica a cada bus, si necesita más reglas, puede crear buses de eventos personalizados adicionales en su cuenta.

Puede personalizar la forma en que se reciben los eventos en su cuenta creando buses de eventos con distintos permisos para distintos servicios.

Para personalizar la estructura o la fecha de un evento antes de EventBridge pasarlo a un destino, utilice el transformador de entrada para editar la información antes de que llegue al destino.

Para obtener más información, consulte ???.

Destinos

Un objetivo es un recurso o punto final al que se EventBridge envía un evento cuando el evento coincide con el patrón de eventos definido para una regla.

Un destino puede recibir varios eventos de varios buses de eventos.

Para obtener más información, consulte ???.

Características avanzadas para los buses de eventos

EventBridge incluye las siguientes funciones para ayudarle a desarrollar, gestionar y utilizar los buses de eventos.

Uso de destinos de API para habilitar las llamadas a la API de REST entre servicios

EventBridge Los destinos de las API son puntos de enlace HTTP que puede establecer como objetivo de una regla, del mismo modo que enviaría los datos de los eventos a un AWS servicio o recurso. Con los destinos de la API, puede usar llamadas a la API para dirigir eventos entre servicios de AWS , aplicaciones de SaaS integradas y sus aplicaciones fuera de AWS. Al crear un destino de la API, se especifica la conexión que se va a utilizar para él. Cada conexión incluye detalles sobre el tipo de autorización y los parámetros que se utilizarán para autorizar con el punto de conexión de destino de la API.

Archivo y reproducción de eventos para facilitar el desarrollo y la recuperación de desastres

Puede archivar o guardar los eventos y después reproducirlos más tarde desde el archivo. El archivo es útil para:

- Probar una aplicación porque tiene un almacén de eventos para usar, en lugar de tener que esperar a que se produzcan nuevos eventos.
- Hidratar un servicio nuevo cuando se pone en línea por primera vez.
- Agregar más durabilidad a sus aplicaciones basadas en eventos.

Uso del Registro de esquemas para iniciar rápidamente la creación de patrones de eventos

Al crear aplicaciones sin servidor que lo utilicen EventBridge, puede resultar útil conocer la estructura de los eventos típicos sin tener que generar el evento. La estructura de los eventos se describe en los esquemas, que están disponibles para todos los eventos generados por los AWS servicios de On. EventBridge

En el caso de los eventos que no provienen de AWS los servicios, puedes:

- Crear o subir esquemas personalizados.
- Utilice Schema Discovery para crear EventBridge automáticamente esquemas para los eventos enviados al bus de eventos.

Una vez que haya encontrado o creado un esquema para un evento, puede descargar enlaces de código para lenguajes de programación populares.

Administración de recursos y accesos con políticas

Para organizar AWS los recursos o hacer un seguimiento de los costos EventBridge, puede asignar una etiqueta o etiqueta personalizada a los AWS recursos. Al usar políticas basadas en etiquetas, puedes controlar lo que los recursos pueden y no pueden hacer dentro EventBridge de ellos.

Además de las políticas basadas en etiquetas, EventBridge admite políticas basadas en la identidad y en los recursos para controlar el acceso a ellas. EventBridge Use políticas basadas en identidad para controlar los permisos de un grupo, rol o usuario. Utilice políticas basadas en recursos para conceder permisos específicos a cada recurso, como una función de Lambda o un tema de Amazon SNS.

Crear un bus de EventBridge eventos de Amazon

Puede crear un bus de eventos personalizado para recibir eventos de las aplicaciones. Las aplicaciones también pueden enviar eventos al bus de eventos predeterminado. Al crear un bus de

eventos, puede adjuntar una política basada en recursos para conceder permisos a otras cuentas. Luego, otras cuentas pueden enviar eventos al bus de eventos de la cuenta actual.

En el siguiente vídeo se muestra cómo crear buses de eventos: Creación de un bus de eventos

Para crear un bus de eventos personalizado

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Buses de eventos.
3. Seleccione Crear bus de eventos.
4. Introduzca un nombre para el nuevo bus de eventos.
5. Elija la KMS key forma EventBridge que desee utilizar al cifrar los datos del evento almacenados en el bus del evento.

> **Note**
>
> Los buses de eventos cifrados con un clave administrada por el cliente. Para habilitar la detección de archivos o esquemas en un bus de eventos, elija usar un Clave propiedad de AWS. Para obtener más información, consulte ???.

- Elija Usar Clave propiedad de AWS EventBridge para cifrar los datos mediante un Clave propiedad de AWS.

  Clave propiedad de AWS Es una KMS key que EventBridge posee y administra para su uso en varias AWS cuentas. En general, a menos que tenga que auditar o controlar la clave de cifrado que protege sus recursos, an Clave propiedad de AWS es una buena opción.

  Esta es la opción predeterminada.
- Elija Usar clave administrada por el cliente EventBridge para cifrar los datos con la clave administrada por el cliente que especifique o cree.

  Claves administradas por el cliente están KMS keys en la AWS cuenta que usted crea, posee y administra. Tienes el control total sobre ellas KMS keys.

  a. Especifique uno existente clave administrada por el cliente o elija Crear uno nuevo KMS key.

EventBridge muestra el estado de la clave y cualquier alias clave que se haya asociado a lo especificado clave administrada por el cliente.

b. Elija la cola de Amazon SQS para utilizarla como cola de cartas muertas (DLQ) para este bus de eventos, si lo hubiera.

EventBridge envía los eventos que no se hayan cifrado correctamente al DLQ, si está configurado, para que pueda procesarlos más adelante.

6. Configure las funciones opcionales del bus de eventos:

- Especifique una política basada en recursos mediante una de las siguientes acciones:
  - Introduzca la política que incluye los permisos que se van a conceder al bus de eventos. Puede pegar una política de otro origen o introducir el formato JSON para la política. Puede usar una de las políticas de ejemplo y modificarla para su entorno.
  - Para usar una plantilla para la política, seleccione Cargar plantilla. Modifique la política según corresponda a su entorno, incluida la adición de acciones que autorice a la entidad principal de la política a utilizar.

  Para obtener más información sobre la concesión de permisos a un bus de eventos mediante políticas basadas en recursos, consulte. ???
- Habilitar un archivado (opcional)

  Puede crear un archivo de eventos para poder reproducirlos fácilmente en otro momento. Por ejemplo, puede que quiera reproducir eventos para recuperarse de errores o para validar una nueva funcionalidad de la aplicación. Para obtener más información, consulte ???.

  a. En Archivos, selecciona Activado.

  b. Especifique un nombre y una descripción para el archivo.

  > **Note**
  >
  > Los buses de eventos cifrados mediante un clave administrada por el cliente. Para habilitar la detección de archivos o esquemas en un bus de eventos, elija usar un Clave propiedad de AWS. Para obtener más información, consulte ???.

- Habilite la detección de esquemas (opcional)

Habilite la detección de esquemas para deducir EventBridge automáticamente los esquemas directamente a partir de los eventos que se ejecutan en este bus de eventos. Para obtener más información, consulte ???.

a. En Descubrimiento de esquemas, elija Activado.

> **Note**
>
> La detección de archivos y esquemas no es compatible con los buses de eventos cifrados mediante un clave administrada por el cliente. Para habilitar la detección de archivos o esquemas en un bus de eventos, elija usar un Clave propiedad de AWS. Para obtener más información, consulte ???.

- Especifique etiquetas (opcional)

  Una etiqueta es una etiqueta de atributo personalizada que se asigna a un AWS recurso. Usa etiquetas para identificar y organizar tus AWS recursos. Muchos AWS servicios admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Para obtener más información, consulte ???.

  a. En Etiquetas, elija Agregar etiqueta nueva.

  b. Especifique una clave y, si lo desea, un valor para la nueva etiqueta.

7. Seleccione Crear.

Actualización de un bus de EventBridge eventos de Amazon

Puede actualizar la configuración de los buses de eventos después de crearlos. Esto incluye el bus de eventos predeterminado, que se EventBridge crea automáticamente en tu cuenta.

Actualización del KMS key utilizado para el cifrado

> **Note**
>
> La detección de archivos y esquemas no es compatible con los buses de eventos cifrados mediante un clave administrada por el cliente. Para habilitar la detección de archivos o esquemas en un bus de eventos, elija usar un Clave propiedad de AWS. Para obtener más información, consulte ???.

Para cambiar lo que KMS key se utiliza para el cifrado en reposo en un bus de eventos mediante la EventBridge consola

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Buses de eventos.
3. Elige el bus del evento que deseas actualizar.
4. En la página de detalles del bus de eventos, selecciona la pestaña Cifrado.
5. Seleccione la KMS key forma EventBridge que desee utilizar al cifrar los datos de eventos almacenados en el bus de eventos:
   - Elija Usar Clave propiedad de AWS EventBridge para cifrar los datos mediante un. Clave propiedad de AWS

     Clave propiedad de AWS Es una KMS key que EventBridge posee y administra para su uso en varias AWS cuentas. En general, a menos que tenga que auditar o controlar la clave de cifrado que protege sus recursos, an Clave propiedad de AWS es una buena opción.

     Esta es la opción predeterminada.
   - Elija Usar clave administrada por el cliente EventBridge para cifrar los datos con la clave administrada por el cliente que especifique o cree.

     Claves administradas por el cliente están KMS keys en la AWS cuenta que usted crea, posee y administra. Tienes el control total sobre ellas KMS keys.

     a. Especifique uno existente clave administrada por el cliente o elija Crear uno nuevo KMS key.

        EventBridge muestra el estado de la clave y cualquier alias clave que se haya asociado a lo especificado clave administrada por el cliente.

     b. Elija la cola de Amazon SQS para utilizarla como cola de cartas muertas (DLQ) para este bus de eventos, si lo hubiera.

        EventBridge envía los eventos que no se hayan cifrado correctamente al DLQ, si está configurado, para que pueda procesarlos más adelante.

Actualización de los permisos en un bus de eventos

Puede conceder permisos adicionales a un bus de eventos adjuntándole una política basada en recursos. Para obtener instrucciones detalladas sobre cómo actualizar los permisos otorgados a un bus de eventos, consulte Administrar los permisos del bus de eventos.

Añadir o eliminar archivos en los buses de eventos

Un archivo le permite capturar eventos para poder reproducirlos fácilmente en otro momento. Por ejemplo, puede que quiera reproducir eventos para recuperarse de errores o para validar una nueva funcionalidad de la aplicación. Para obtener más información, consulte EventBridge archivar y reproducir.

> **Note**
>
> La detección de archivos y esquemas no es compatible con los buses de eventos cifrados mediante un clave administrada por el cliente. Para habilitar la detección de archivos o esquemas en un bus de eventos, elija usar un Clave propiedad de AWS. Para obtener más información, consulte ???.

Para añadir o eliminar un archivo de un bus de eventos mediante la EventBridge consola

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Buses de eventos.
3. Elige el bus del evento que deseas actualizar.
4. En la página de detalles del bus de eventos, selecciona la pestaña Archivos.
5. Realice una de las siguientes acciones siguientes:
   - Para añadir un archivo:
     a. Elija Crear archivo.
     b. Especifique los atributos del archivo.
     c. Elija Siguiente.
     d. Elija el patrón de eventos que desee aplicar a los eventos del archivo.
     e. Elija Crear archivo.
   - Para eliminar un archivo:

a. Para la etiqueta que quieres eliminar, selecciona Eliminar.
b. Introduce el nombre del archivo y selecciona Eliminar.

   El archivo se elimina permanentemente. No podrá deshacer esta operación.

Para crear o eliminar un archivo para un bus de eventos mediante el AWS CLI

- Para crear un archivo, utilice create-archive.

  Para eliminar permanentemente un archivo, utilice delete-archive.

Iniciar o detener la detección de esquemas en los buses de eventos

Para obtener más información sobre la detección de esquemas, consulte EventBridge esquemas.

> **Note**
>
> La detección de archivos y esquemas no es compatible con los buses de eventos cifrados mediante un clave administrada por el cliente. Para habilitar la detección de archivos o esquemas en un bus de eventos, elija usar un Clave propiedad de AWS. Para obtener más información, consulte ???.

Para iniciar o detener la detección de esquemas en un bus de eventos mediante la EventBridge consola

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Buses de eventos.
3. Elige el bus del evento que deseas actualizar.
4. Realice una de las siguientes acciones siguientes:
   - Para iniciar la detección de esquemas, elija Iniciar la detección.
   - Para detener la detección de esquemas, elija Eliminar la detección.

Para iniciar o detener la detección de esquemas en un bus de eventos mediante el AWS CLI

- Para iniciar la detección de esquemas, utilice create-discoverer.

Para detener la detección de esquemas, utilice delete-discoverer.

Añadir o eliminar etiquetas en los buses de eventos

Una etiqueta es una etiqueta de atributo personalizada que usted o AWS asigna a un AWS recurso. Usa etiquetas para identificar y organizar tus AWS recursos. Para obtener más información, consulte EventBridge etiquetas.

Para añadir o eliminar etiquetas de un bus de eventos mediante la EventBridge consola

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Buses de eventos.
3. Elige el bus del evento que deseas actualizar.
4. En la página de detalles del bus de eventos, selecciona la pestaña Etiquetas y, a continuación, selecciona Administrar etiquetas.
5. Realice una de las siguientes acciones siguientes:
   - Para añadir una etiqueta:
     a. Elija Añadir nueva etiqueta.
     b. Especifique la clave y el valor de la etiqueta
     c. Elija Actualizar.
   - Para eliminar una etiqueta:
     a. Para la etiqueta que quieres eliminar, selecciona Eliminar.
     b. Elija Actualizar.

Para añadir o eliminar etiquetas de un bus de eventos mediante el AWS CLI

- Para añadir etiquetas, utilice tag-resource.

  Para eliminar etiquetas, usa untag-resource.

Actualizar el bus de eventos predeterminado mediante AWS CloudFormation

AWS CloudFormation le permite configurar y administrar sus AWS recursos en todas las cuentas y regiones de manera centralizada y repetible al tratar la infraestructura como un código. CloudFormation lo hace permitiéndole crear plantillas que definen los recursos que desea aprovisionar y administrar.

Como EventBridge aprovisiona automáticamente el bus de eventos predeterminado en su cuenta, no puede crearlo mediante una CloudFormation plantilla, como haría normalmente con cualquier recurso que quisiera incluir en una CloudFormation pila. Para incluir el bus de eventos predeterminado en una CloudFormation pila, primero debe importarlo a una pila. Una vez que haya importado el bus de eventos predeterminado a una pila, podrá actualizar las propiedades del bus de eventos según lo desee.

Para importar un recurso existente a una CloudFormation pila nueva o existente, necesita la siguiente información:

- Un identificador único del recurso que se va a importar.

  Para los buses de eventos predeterminados, el identificador es `Name` y, a continuación, el valor del identificador es`default`.
- Una plantilla que describe con precisión las propiedades actuales del recurso existente.

  El siguiente fragmento de plantilla contiene un `AWS::Events::EventBus` recurso que describe las propiedades actuales de un bus de eventos predeterminado. En este ejemplo, el bus de eventos se ha configurado para usar un DLQ clave administrada por el cliente y un DLQ para el cifrado en reposo.

  Además, el `AWS::Events::EventBus` recurso que describe el bus de eventos predeterminado que desea importar debe incluir una `DeletionPolicy` propiedad establecida en. `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"
                }
            }
        }
    }
}
```

Para obtener más información, consulte CloudFormation Gestión de los recursos existentes en la Guía del CloudFormation usuario.

Eliminar un bus de EventBridge eventos de Amazon

Puede eliminar un bus de eventos personalizado o asociado. No puede eliminar el bus de eventos predeterminado. Al eliminar un bus de eventos, se eliminan las reglas asociadas a ese bus de eventos.

Para eliminar un bus de eventos mediante la consola EventBridge

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Buses de eventos.
3. Elige el bus de eventos que deseas eliminar.
4. Realice una de las siguientes acciones siguientes:
   - Elija Eliminar.
   - Elija el nombre del bus de eventos.

     En la página de detalles del bus de eventos, seleccione Eliminar.

Permisos para buses de eventos de Amazon EventBridge

El bus de eventos predeterminado de la cuenta de AWS solo permite eventos de una cuenta. Puede conceder permisos adicionales a un bus de eventos adjuntándole una política basada en recursos. Con una política basada en recursos, puede permitir llamadas a la API `PutEvents`, `PutRule` y `PutTargets` desde otra cuenta. También puede usar condiciones de IAM en la política para conceder permisos a una organización, aplicar etiquetas o filtrar eventos solo para los de una regla

o cuenta específica. Puede establecer una política basada en recursos para un bus de eventos al crearlo o después.

Las API de EventBridge que aceptan un parámetro `Name` de bus de eventos como `PutRule`, `PutTargets`, `DeleteRule`, `RemoveTargets`, `DisableRule` y `EnableRule` también aceptan el ARN del bus de eventos. Utilice estos parámetros para hacer referencia a los buses de eventos entre cuentas o regiones a través de las API. Por ejemplo, puede realizar llamadas `PutRule` para crear una regla en un bus de eventos de otra cuenta sin necesidad de asumir ningún rol.

Puede adjuntar las políticas de ejemplo de este tema a un rol de IAM para conceder permiso para enviar eventos a otra cuenta o región. Use los roles de IAM para establecer las políticas de control de la organización y los límites sobre quién puede enviar eventos desde su cuenta a otras. Recomendamos utilizar siempre los roles de IAM cuando el destino de una regla sea un bus de eventos. Puede adjuntar roles de IAM mediante llamadas `PutTarget`. Para obtener información sobre cómo crear una regla para enviar eventos a una cuenta o región diferente, consulte Envío y recepción de EventBridge eventos de Amazon entre AWS cuentas.

Temas

Administración de permisos para buses de eventos

Utilice el siguiente procedimiento para modificar los permisos para un bus de eventos existente. Para obtener información sobre cómo usar AWS CloudFormation para crear una política de bus de eventos, consulte AWS::Events: :EventBusPolicy.

Para administrar los permisos para un bus de eventos existente

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.

2. En el panel de navegación, seleccione Buses de eventos.
3. En Nombre, seleccione el nombre del bus de eventos para el que se van a administrar los permisos.

   Si hay una política de recursos adjunta al bus de eventos, se muestra la política.
4. Seleccione Administrar permisos y, a continuación, lleve a cabo alguna de las siguientes operaciones:
   - Introduzca la política que incluya los permisos que se van a conceder al bus de eventos. Puede pegar una política de otro origen o introducir el formato JSON para la política.
   - Para usar una plantilla para la política, seleccione Cargar plantilla. Modifique la política según corresponda a su entorno, incluida la adición de acciones que autorice al responsable de la política a utilizar.
5. Selecciones Actualizar.

En la plantilla se proporcionan ejemplos de declaraciones de política que puede personalizar para su cuenta y entorno. La plantilla no es una política válida. Puede modificar la plantilla según su caso de uso o puede copiar una de las políticas de ejemplo y personalizarla.

La plantilla carga políticas que incluyen un ejemplo de cómo conceder permisos a una cuenta para usar la acción `PutEvents`, cómo conceder permisos a una organización y cómo conceder permisos a la cuenta para gestionar las reglas de la cuenta. Puede personalizar la plantilla para la cuenta específica y, a continuación, eliminar las demás secciones de la plantilla. Más adelante en este tema se incluyen más ejemplos de políticas.

Si intenta actualizar los permisos para el bus pero la política contiene un error, un mensaje de error indica el problema específico de la política.

```
 ### 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>"
          }
        }
    }]
  }
```

Ejemplo de política: Enviar eventos al autobús predeterminado de otra cuenta

El siguiente ejemplo de política concede a la cuenta 111122223333 permiso para publicar eventos en el bus de eventos predeterminado de la cuenta 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"
        }
    ]
  }
```

Ejemplo de política: Enviar eventos a un bus personalizado de otra cuenta

El siguiente ejemplo de política concede a la cuenta 111122223333 permiso para publicar eventos en el `central-event-bus` de la cuenta 123456789012, pero solo para los eventos con un valor de origen establecido en `com.exampleCorp.webStore` y un `detail-type` establecido en `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"
        }
      }
    }
  ]
}
```

Ejemplo de política: Enviar eventos a un bus de eventos de la misma cuenta

El siguiente ejemplo de política adjunta a un bus de eventos denominado `CustomBus1` permite que el bus de eventos reciba eventos de la misma cuenta y región.

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

```
}
```

Ejemplo de política: Enviar eventos a la misma cuenta y restringir las actualizaciones

El siguiente ejemplo de política concede a la cuenta 123456789012 permiso para crear, eliminar, actualizar, deshabilitar y habilitar reglas y agregar o eliminar destinos. Limita estas reglas que coinciden con los eventos con un origen de `com.exampleCorp.webStore`, y utiliza la `"events:creatorAccount": "${aws:PrincipalAccount}"` para garantizar que solo la cuenta 123456789012 pueda modificar estas reglas y destinos una vez que se hayan creado.

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

Ejemplo de política: Enviar eventos solo desde una regla específica al bus de una región diferente

El siguiente ejemplo de política concede a la cuenta 111122223333 permiso para enviar eventos que coincidan con una regla denominada `SendToUSE1AnotherAccount` en las regiones de Medio Oriente (Baréin) y Oeste de EE. UU. (Oregón) a un bus de eventos denominado `CrossRegionBus` en la región Este de EE. UU. (Norte de Virginia) en la cuenta 123456789012. La política de ejemplo se agrega al bus de eventos denominado `CrossRegionBus` en la cuenta 123456789012. La política permite los eventos solo si coinciden con una regla especificada para el bus de eventos en la cuenta 111122223333. La declaración `Condition` restringe los eventos solo a los eventos que coinciden con las reglas con el ARN de regla especificado.

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

Ejemplo de política: Enviar eventos solo desde una región específica a una región diferente

El siguiente ejemplo de política concede a la cuenta 111122223333 permiso para enviar eventos generados en las regiones de Medio Oriente (Baréin) y Oeste de EE. UU. (Oregón) a un bus de eventos denominado `CrossRegionBus` en la región Este de EE. UU. (Norte de Virginia) en la cuenta 123456789012. La cuenta 111122223333 no tiene permiso para enviar eventos que se generen en ninguna otra región.

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

Ejemplo de política: Denegar el envío de eventos desde regiones específicas

El siguiente ejemplo de política adjunta a un bus de eventos con denominado `CrossRegionBus` de la cuenta 123456789012 permite que el bus de eventos reciba eventos de la cuenta 111122223333, pero no eventos generados en la región Oeste de EE. UU. (Oregón).

```
{
```

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

Generar una plantilla de AWS CloudFormation a partir de un bus de eventos de Amazon EventBridge

AWS CloudFormation le permite configurar y administrar sus recursos de AWS en todas las cuentas y regiones de manera centralizada y repetible al tratar la infraestructura como código. Para ello, CloudFormation le permite crear plantillas, que definen los recursos que desea aprovisionar y administrar.

EventBridge le permite generar plantillas a partir de los buses de eventos existentes en su cuenta, para ayudarle a empezar a desarrollar plantillas de CloudFormation. Además, EventBridge ofrece la opción de incluir las reglas asociadas a ese bus de eventos en su plantilla. A continuación,

puede utilizar estas plantillas como base para crear pilas de recursos bajo administración de CloudFormation.

Para obtener más información sobre CloudFormation, consulte la Guía del usuario de AWS CloudFormation.

> **Note**
>
> EventBridge no incluye reglas gestionadas en la plantilla generada.

También puede generar una plantilla a partir de una o más reglas contenidas en un bus de eventos seleccionado.

Para generar una plantilla de CloudFormation a partir de un bus de eventos

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Buses de eventos.
3. Seleccione el bus de eventos a partir del que desea generar una plantilla de CloudFormation.
4. En el menú Acciones, seleccione Plantilla de CloudFormation y, a continuación, seleccione el formato en el que quiere que EventBridge genere la plantilla: JSON o YAML.

   EventBridge muestra la plantilla, generada en el formato seleccionado. De forma predeterminada, todas las reglas asociadas al bus de eventos se incluyen en la plantilla.

   - Para generar la plantilla sin incluir reglas, desmarque Incluir reglas en este bus de eventos.
5. EventBridge le da la opción de descargar el archivo de plantilla o copiar la plantilla al portapapeles.

   - Seleccione Descargar para descargar el archivo de plantilla.
   - Para copiar la plantilla al portapapeles, seleccione Copiar.
6. Para salir de la plantilla, seleccione Cancelar.

Una vez que haya personalizado la plantilla de AWS CloudFormation según sea necesario para su caso de uso, podrá utilizarla para crear pilas en CloudFormation.

Consideraciones sobre el uso de plantillas de CloudFormation generadas a partir de Amazon EventBridge

Tenga en cuenta los siguientes factores al utilizar una plantilla de CloudFormation que haya generado a partir de un bus de eventos:

- EventBridge no incluye contraseñas en la plantilla generada.

  Puede editar la plantilla para incluir parámetros de plantilla que permitan a los usuarios especificar contraseñas u otra información confidencial al utilizar la plantilla para crear o actualizar una pila de CloudFormation.

  Además, los usuarios pueden usar Secrets Manager para crear un secreto en la región deseada y, a continuación, editar la plantilla generada para emplear parámetros dinámicos.
- Los destinos de la plantilla generada permanecen exactamente como se especificaron en el bus de eventos original. Esto puede provocar problemas entre regiones si no se edita correctamente la plantilla antes de utilizarla para crear pilas en otras regiones.

  Además, la plantilla generada no creará automáticamente los destinos descendentes.

EventBridge Eventos de Amazon

Un evento indica un cambio en un entorno, como un entorno de AWS , un servicio o aplicación de socios SaaS o una de sus propias aplicaciones o servicios. Los siguientes ejemplos corresponden a eventos:

- Amazon EC2 genera un evento cuando el estado de una instancia cambia de pendiente a ejecutándose.
- Amazon EC2 Auto Scaling genera eventos cuando lanza o finaliza instancias.
- AWS CloudTrail publica eventos cuando realiza llamadas a la API.

También puede configurar eventos programados que se generan de forma periódica.

Para obtener una lista de servicios que generan eventos, incluidos eventos de ejemplo de cada servicio, consulte Eventos de los AWS servicios y siga los enlaces de la tabla.

Los eventos se representan como objetos JSON y todos tienen una estructura similar y los mismos campos de nivel superior.

El contenido del campo de nivel superior detail será diferente en función del servicio que haya generado el evento y de cuál sea el evento. La combinación de los campos source y detail-type sirve para identificar los campos y los valores encontrados en detail. Para ver ejemplos de eventos generados por AWS los servicios, consulteEventos de los AWS servicios.

Temas

- Referencia de la estructura de un evento
- Añadir EventBridge eventos de Amazon con PutEvents
- Eventos de los AWS servicios
- Recibir eventos de un socio de SaaS de Amazon EventBridge
- Depuración de la entrega de eventos

En el siguiente vídeo se explican los aspectos básicos de los eventos: Qué es un evento

El siguiente vídeo describe las formas en que llegan los eventos EventBridge: ¿De dónde vienen los eventos

Referencia de la estructura de un evento

Los siguientes campos aparecen en todos los eventos que se envían a un bus de eventos y comprenden los metadatos del evento:

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

versión

De forma predeterminada, está definida en 0 (cero) en todos los eventos.

id

Un UUID de versión 4 generado para cada evento. Puede usar `id` para hacer un seguimiento de los eventos a medida que pasan de las reglas a los destinos.

tipo-detalle

Identifica, en combinación con el campo source, los campos y los valores que aparecen en detail.

Los eventos que son entregados por CloudTrail tienen `AWS API Call via CloudTrail` como valor`detail-type`.

source

Identifica el servicio que generó el evento. Todos los eventos que provienen de servicios de AWS empiezan por “aws”. Los eventos generados por el cliente pueden tener cualquier valor aquí,

salvo que no pueden empezar por "aws". Le recomendamos que utilice cadenas de nombres de dominio inversas que utilicen el estilo de nombres de paquetes de Java.

Para encontrar el valor correcto de un AWS servicio, consulte la tabla de claves de condición, seleccione un servicio de la lista y busque el prefijo del servicio. `source` Por ejemplo, el `source` valor de Amazon CloudFront es`aws.cloudfront`.

cuenta

El número de 12 dígitos que identifica una AWS cuenta.

hora

La marca temporal del evento, que puede especificar el servicio que origina el evento. Si el evento abarca un intervalo de tiempo, el servicio puede notificar la hora de inicio, por lo que este valor puede ser anterior al momento en que se recibe el evento.

región

Identifica la AWS región en la que se originó el evento.

resources

Esta matriz JSON contiene ARN que identifican recursos que participan en el evento. El servicio que genera el evento determina si se deben incluir estos ARN. Por ejemplo, los cambios de estado de instancia de Amazon EC2 incluyen los ARN de instancia de Amazon EC2, los eventos de escalado automático incluyen los ARN tanto para instancias como para grupos de escalado automático, pero las llamadas a la API con AWS CloudTrail no incluyen los ARN de recursos.

detail

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo. Puede ser `"{}"`.

AWS Los eventos de llamada a la API tienen objetos detallados con aproximadamente 50 campos anidados en varios niveles de profundidad.

> **Note**
>
> PutEventsacepta datos en formato JSON. Para el tipo de datos número JSON (entero), las restricciones son: un valor mínimo de -9.223.372.036.854.775.808 y un valor máximo de 9.223.372.036.854.775.807.

Example Ejemplo: Notificación de cambio de estado de una instancia de Amazon EC2

El siguiente evento en Amazon EventBridge indica que se está cancelando una instancia de Amazon EC2.

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

Información mínima necesaria para un evento personalizado válido

Al crear eventos personalizados, estos deben incluir los siguientes campos:

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

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

Añadir EventBridge eventos de Amazon con **`PutEvents`**

La `PutEvents` acción envía varios eventos EventBridge en una sola solicitud. Para obtener más información, consulte PutEventsla referencia de la EventBridge API de Amazon y la referencia de AWS CLI comandos put-events.

Cada solicitud `PutEvents` puede admitir un número limitado de entradas. Para obtener más información, consulte Cuotas de Amazon EventBridge. La operación `PutEvents` intenta procesar todas las entradas en el orden natural de la solicitud. Después de llamar`PutEvents`, EventBridge asigna a cada evento un identificador único.

Temas

- Gestión de errores con PutEvents
- Enviar eventos mediante el AWS CLI
- Calcular el tamaño de la entrada a EventBridge PutEvents un evento de Amazon

En el siguiente ejemplo, el código Java envía dos eventos idénticos 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());
    }
}
```

Tras ejecutar este código, el resultado de `PutEvents` incluye una matriz de entradas de respuesta. Cada entrada en la matriz de respuestas se corresponde con una entrada en la matriz de solicitudes siguiendo el orden de principio a fin de la solicitud y la respuesta. La matriz de respuesta `Entries` siempre incluye el mismo número de entradas que la matriz de solicitud.

Gestión de errores con **PutEvents**

De forma predeterminada, si se produce un error en una entrada individual de una solicitud, EventBridge continúa procesando el resto de las entradas de la solicitud. Una matriz de `Entries` de respuestas puede incluir entradas correctas e incorrectas. Debe detectar las entradas incorrectas e incluirlas en una llamada siguiente.

Las entradas de resultados correctas incluyen un valor `Id` y las entradas de resultados incorrectas incluyen valores `ErrorCode` y `ErrorMessage`. `ErrorCode` describe el tipo de error. `ErrorMessage` proporciona más información acerca del error. El ejemplo siguiente tiene tres entradas de resultados para una solicitud `PutEvents`. La segunda entrada no es correcta.

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

> **Note**
>
> Si publicas `PutEvents` un evento en un bus de eventos que no existe, la coincidencia de EventBridge eventos no encontrará la regla correspondiente y descartará el evento. Aunque EventBridge enviará una `200` respuesta, no rechazará la solicitud ni incluirá el evento en el `FailedEntryCount` valor de la respuesta a la solicitud.

Puede incluir las entradas incorrectas en solicitudes `PutEvents` posteriores. En primer lugar, compruebe el parámetro `FailedRecordCount` en `PutEventsResult` para confirmar si se hay entradas incorrectas. Si no es cero, puedes añadir cada `Entry` que tenga un valor `ErrorCode` que no sea nulo a una solicitud posterior. En el siguiente ejemplo se muestra un administrador de errores.

```
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);
    }
```

Enviar eventos mediante el AWS CLI

Puede utilizar el AWS CLI para enviar eventos personalizados a EventBridge fin de que se puedan procesar. En el siguiente ejemplo, se coloca un evento personalizado en 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\" }"}]'
```

También puede crear un archivo JSON que contenga eventos personalizados.

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

A continuación, para usar el AWS CLI para leer las entradas de este archivo y enviar eventos, en una línea de comandos, escriba:

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

Calcular el tamaño de la entrada a EventBridge PutEvents un evento de Amazon

Puedes enviar eventos personalizados a EventBridge mediante la `PutEvents` acción. Puede incluir en un lote varios eventos en una única solicitud para mayor eficacia. El tamaño total de entrada debe ser inferior a 256 KB. Puede calcular el tamaño de entrada antes de enviar los eventos.

> **Note**
>
> Se impone un límite de tamaño en la entrada. Incluso si la entrada es inferior al límite de tamaño, el tamaño del evento siempre EventBridge es mayor que el tamaño de la entrada debido a los caracteres y claves necesarios para la representación del evento en JSON. Para obtener más información, consulte EventBridge Eventos de Amazon.

EventBridge calcula el `PutEventsRequestEntry` tamaño de la siguiente manera:

- Si se especifica, el parámetro `Time` es de 14 bytes.
- Los parámetros `Source` y `DetailType` son el número de bytes para sus formatos cifrados con UTF-8.
- Si se especifica, el parámetro `Detail` es el número de bytes para el formato cifrado con UTF-8.
- Si se especifica, cada entrada del parámetro `Resources` es el número de bytes para sus formatos cifrados con UTF-8.

El siguiente ejemplo de código Java calcula el tamaño de un objeto `PutEventsRequestEntry` dado.

```
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**
>
> Si el tamaño de la entrada es superior a 256 KB, recomendamos subir el evento a un bucket de Amazon S3 e incluir el `Object URL` en la entrada `PutEvents`.

Eventos de los AWS servicios

Muchos AWS servicios generan eventos que EventBridge reciben. Cuando un AWS servicio de tu cuenta emite un evento, pasa al bus de eventos predeterminado de tu cuenta.

Entrega de eventos desde los AWS servicios

Cada AWS servicio que genera eventos los envía EventBridge como el mejor esfuerzo o como un intento de entrega duradero.

- La entrega con el máximo esfuerzo significa que el servicio intenta enviar todos los eventos a EventBridge, pero en algunos casos excepcionales es posible que un evento no se entregue.
- La entrega duradera significa que el servicio intentará entregar los eventos al EventBridge menos una vez con éxito.

  EventBridge aceptará todos los eventos válidos en condiciones normales. En los casos en que los eventos no se puedan entregar debido a una interrupción del EventBridge servicio, el AWS servicio volverá a intentarlo más adelante durante un máximo de 24 horas.

Una vez entregado un evento EventBridge, lo compara EventBridge con las reglas y, a continuación, sigue la política de reintentos y cualquier cola de espera especificada para los objetivos del evento.

Para obtener una lista de los AWS servicios que generan eventos, consulte. ???

Acceder a los eventos del AWS servicio mediante AWS CloudTrail

AWS CloudTrail es un servicio que registra automáticamente eventos como las llamadas a la AWS API. Puede crear EventBridge reglas que utilicen la información de CloudTrail. Para obtener más información al respecto CloudTrail, consulte ¿Qué es AWS CloudTrail? .

Todos los eventos que son entregados por CloudTrail tienen `AWS API Call via CloudTrail` como valor`detail-type`.

Para registrar eventos con un `detail-type` valor de`AWS API Call via CloudTrail`, se requiere CloudTrail un registro con el registro activado.

Si lo utiliza CloudTrail con Amazon S3, debe configurarlo CloudTrail para registrar eventos de datos. Para obtener más información, consulte Habilitar el registro de CloudTrail eventos para buckets y objetos de S3.

El propio AWS servicio o el propio servicio pueden informar EventBridge sobre algunas incidencias en los servicios. CloudTrail Por ejemplo, una llamada a la API Amazon EC2 que inicia o detiene una instancia genera EventBridge eventos y eventos intermedios. CloudTrail

CloudTrail permite que tanto las personas que llaman a la API como a los propietarios de los recursos reciban eventos en sus buckets de Amazon S3 mediante la creación de rutas y, a través de ella, entrega eventos a las personas que llaman a la API. EventBridge Los propietarios de los recursos, además de las personas que llaman a la API, pueden monitorear las llamadas a la API entre cuentas. EventBridge CloudTrailSu integración con EventBridge proporciona una forma cómoda de configurar flujos de trabajo automatizados basados en reglas en respuesta a los eventos.

No puedes usar AWS eventos de llamada a la API Put*Events que tengan un tamaño superior a 256 KB como patrones de eventos porque el tamaño máximo de cualquier solicitud de Put*Events es de 256 KB. Para obtener más información sobre las llamadas a la API que puedes usar, consulta los servicios e integraciones CloudTrail compatibles.

Recibir eventos de administración de los servicios de solo lectura AWS

Puede configurar reglas en su bus de eventos predeterminado o personalizado para recibir eventos de administración de los servicios de solo lectura mediante. AWS CloudTrail Los eventos de administración proporcionan visibilidad de las operaciones de administración que se llevan a cabo con los recursos de su AWS cuenta. Se denominan también operaciones del plano de control. Para obtener más información, consulte Registro de eventos de administración en la Guía del usuario de CloudTrail .

Para cada regla de los buses de eventos predeterminados o personalizados, puede establecer el estado de la regla para controlar los tipos de eventos que se recibirán:

- Deshabilite la regla para que los eventos EventBridge no coincidan con la regla.
- Habilite la regla para que compare los EventBridge eventos con la regla, excepto los eventos de AWS administración de solo lectura que se envíen a través de ella. CloudTrail
- Active la regla para que todos los eventos EventBridge coincidan con la regla, incluidos los eventos de administración de solo lectura que se envíen a través de ella. CloudTrail

Los autobuses de eventos asociados no reciben AWS eventos.

Algunos aspectos que se deben tener en cuenta a la hora de decidir si recibir eventos de administración de solo lectura:

- Algunos eventos de administración de solo lectura, como los eventos AWS Key Management Service `GetKeyPolicy` and `DescribeKey` o IAM `GetPolicy` y `GetRole` los eventos, se producen con un volumen mucho mayor que los eventos de cambio típicos.
- Es posible que ya esté recibiendo eventos de administración de solo lectura, si esos eventos no comienzan por, o `Describe`, `Get` o `List`. Por ejemplo, los eventos de las siguientes AWS STS API son eventos de cambio, aunque comiencen con el verbo: `Get`
  - `GetFederationToken`
  - `GetSessionToken`

  Para obtener una lista de los eventos de administración de solo lectura que no siguen la convención de `List` nomenclatura o la `Describe` convención de nomenclatura por AWS servicios, consulte. `Get` ???

Para crear una regla que reciba eventos de administración de solo lectura mediante la CLI AWS

- Utilice el comando `put-rule` para crear o actualizar la regla, utilizando parámetros para:
  - Especificar que la regla pertenece al bus de eventos predeterminado o a un bus de eventos personalizado específico.
  - Establecer el estado de la regla como `ENABLED_WITH_ALL_CLOUDTRAIL_MANAGEMENT_EVENTS`.

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

> **Note**
>
> La activación de una regla para los eventos CloudWatch de administración solo se admite mediante la AWS CLI y AWS CloudFormation las plantillas.

Example

En el siguiente ejemplo se muestra cómo determinar una coincidencia respecto de eventos específicos. La práctica recomendada consiste en definir una regla dedicada para hacer coincidir eventos específicos, para lograr una mayor claridad y facilidad de edición.

En este caso, la regla dedicada coincide con el evento de `AssumeRole` administración de AWS Security Token Service.

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

AWS servicios que generan eventos

En la siguiente tabla se muestran AWS los servicios que generan eventos. Elija el nombre del servicio para obtener más información sobre cómo EventBridge funcionan juntos ese servicio y su funcionamiento.

Cada AWS servicio que genera eventos los envía EventBridge como un intento de entrega óptimo o duradero. Para obtener más información, consulte ???.

Esta tabla incluye una representación de los AWS servicios a los que envían los eventos EventBridge, pero no incluye todos los servicios. En el caso de los servicios que no figuran en la lista y a los que se envían eventos EventBridge, se supone que la entrega se hace todo lo posible.

Servicio	Tipo de intento
Alexa for Business	Mejor forma posible
AWS Account Management	Mejor forma posible
Amazon API Gateway	Mejor forma posible
AWS AppConfig	Mejor forma posible
Amazon AppFlow	Mejor forma posible
Aplicación de escalado automático	Mejor forma posible
AWS Elaborador de perfiles de costes de aplicaciones	Mejor forma posible
AWS Application Migration Service	Mejor forma posible
Amazon Athena	Mejor forma posible
AWS Backup	Mejor forma posible
AWS Batch	Duradero
Amazon Braket	Duradero
AWS Certificate Manager	Mejor forma posible
Amazon Chime	Mejor forma posible
Amazon Cloud Directory	Mejor forma posible
AWS CloudFormation	Duradero
Amazon CloudFront	Mejor forma posible
AWS CloudHSM	Mejor forma posible
Amazon CloudSearch	Mejor forma posible
AWS CloudShell	Mejor forma posible

Servicio	Tipo de intento
Eventos de AWS CloudTrail	Mejor forma posible
Amazon CloudWatch	Duradero
Información sobre CloudWatch aplicaciones de Amazon	Mejor forma posible
Amazon CloudWatch Internet Monitor	Mejor forma posible
Amazon CloudWatch Logs	Mejor forma posible
Amazon CloudWatch Synthetics	Mejor forma posible
AWS CodeArtifact	Duradero
AWS CodeBuild	Mejor forma posible
AWS CodeCommit	Mejor forma posible
AWS CodeDeploy	Mejor forma posible
Amazon CodeGuru Profiler	Mejor forma posible
AWS CodePipeline	Mejor forma posible
AWS CodeStar	Mejor forma posible
CodeConnections	Mejor forma posible
Amazon Cognito Identity	Mejor forma posible
Grupos de usuarios de Amazon Cognito	Mejor forma posible
Amazon Cognito Sync	Mejor forma posible
AWS Config	Mejor forma posible
Amazon Connect	Mejor forma posible
Amazon Connect Voice ID	Mejor forma posible

Servicio	Tipo de intento
AWS Control Tower	Mejor forma posible
AWS Database Migration Service	Mejor forma posible
AWS Data Exchange	Mejor forma posible
Administrador de vida útil de datos de Amazon	Mejor forma posible
AWS Data Pipeline	Mejor forma posible
AWS DataSync	Mejor forma posible
AWS Device Farm	Mejor forma posible
El DevOps gurú de Amazon	Mejor forma posible
AWS Direct Connect	Mejor forma posible
AWS Directory Service	Mejor forma posible
Amazon DynamoDB	Mejor forma posible
AWS Elastic Beanstalk	Mejor forma posible
Amazon Elastic Block Store (EBS)	Mejor forma posible
Modificaciones de volumen de Amazon Elastic Block Store	Mejor forma posible
Amazon ElastiCache	Mejor forma posible
Amazon Elastic Compute Cloud (Amazon EC2)	Mejor forma posible
Amazon EC2 Auto Scaling	Mejor forma posible
Flotas de Amazon EC2	Mejor forma posible
Interrupción de instancias de spot de Amazon EC2	Mejor forma posible

Servicio	Tipo de intento
Amazon Elastic Container Registry	Mejor forma posible
Amazon Elastic Container Service	Duradero
AWS Elastic Disaster Recovery	Mejor forma posible
Amazon Elastic File System	Mejor forma posible
Amazon Elastic Kubernetes Service	Mejor forma posible
Elastic Load Balancing	Mejor forma posible
Amazon Elastic MapReduce	Mejor forma posible
Amazon Elastic Transcoder	Mejor forma posible
AWS Elemental MediaConnect	Mejor forma posible
AWS Elemental MediaConvert	Duradero
AWS Elemental MediaLive	Mejor forma posible
AWS Elemental MediaPackage	Mejor forma posible
AWS Elemental MediaStore	Duradero
Amazon EMR	Mejor forma posible
Amazon EMR en EKS	Mejor forma posible
Amazon EMR sin servidor	Mejor forma posible
Reglas EventBridge programadas de Amazon	Duradero
EventBridge Esquemas de Amazon	Mejor forma posible
AWS Fault Injection Service	Mejor forma posible
Amazon Forecast	Mejor forma posible

Servicio	Tipo de intento
Amazon GameLift	Mejor forma posible
AWS Glue	Mejor forma posible
AWS Glue DataBrew	Mejor forma posible
AWS Ground Station	Mejor forma posible
Amazon GuardDuty	Mejor forma posible
AWS Health	Mejor forma posible
AWS HealthLake	Duradero
AWS Identity and Access Management (IAM)	Mejor forma posible
IAM Access Analyzer	Mejor forma posible
Amazon Inspector Classic	Mejor forma posible
Amazon Inspector	Mejor forma posible
AWS IoT	Mejor forma posible
AWS IoT Analytics	Duradero
AWS IoT Greengrass V1	Mejor forma posible
AWS IoT Greengrass V2	Mejor forma posible
Amazon Interactive Video Service	Mejor forma posible
Amazon Kinesis	Mejor forma posible
Amazon Data Firehose	Mejor forma posible
AWS Key Management Service Eliminación de CMK	Duradero
AWS Key Management Service Rotación CMK	Mejor forma posible

Servicio	Tipo de intento
AWS Key Management Service caducidad del material clave importado	Mejor forma posible
AWS Lambda	Mejor forma posible
Amazon Location Service	Duradero
Amazon Machine Learning	Mejor forma posible
Amazon Macie	Mejor forma posible
Amazon Managed Blockchain	Mejor forma posible
AWS Managed Services	Mejor forma posible
AWS Management Console Iniciar sesión	Mejor forma posible
AWS Marketplace de medición	Mejor forma posible
AWS Migration Hub	Mejor forma posible
AWS Migration Hub Refactor Spaces	Mejor forma posible
AWS Monitorización	Mejor forma posible
AWS Network Manager	Mejor forma posible
OpenSearch Servicio Amazon	Mejor forma posible
AWS OpsWorks	Duradero
AWS OpsWorks CM	Mejor forma posible
AWS Organizations	Mejor forma posible
Amazon Polly	Mejor forma posible
AWS Private Certificate Authority	Mejor forma posible
AWS Proton	Mejor forma posible

Servicio	Tipo de intento
Amazon QLDB	Duradero
Amazon QuickSight	Mejor forma posible
Amazon RDS	Mejor forma posible
AWS Papelera de reciclaje	Mejor forma posible
Amazon Redshift	Duradero
API de datos de Amazon Redshift	Mejor forma posible
Amazon Redshift sin servidor	Mejor forma posible
AWS Resource Access Manager	Mejor forma posible
AWS Resource Groups	Mejor forma posible
AWS Resource Groups Tagging API	Mejor forma posible
Amazon Route 53	Mejor forma posible
Preparación para la recuperación de Amazon Route 53	Mejor forma posible
Amazon SageMaker	Mejor forma posible
Savings Plans	Mejor forma posible
AWS Secrets Manager	Mejor forma posible
AWS Security Hub	Duradero
AWS Security Token Service	Mejor forma posible
AWS Server Migration Service	Mejor forma posible
AWS Service Catalog	Mejor forma posible
AWS Signer	Duradero

Servicio	Tipo de intento
Amazon Simple Email Service	Mejor forma posible
Amazon Simple Storage Service (Amazon S3)	Duradero
Amazon S3 Glacier	Mejor forma posible
Amazon S3 en Outposts	Mejor forma posible
Amazon Simple Queue Service	Mejor forma posible
Amazon Simple Notification Service	Mejor forma posible
Amazon Simple Workflow Service	Mejor forma posible
AWS Step Functions	Mejor forma posible
AWS Storage Gateway	Duradero
AWS Support	Mejor forma posible
AWS Systems Manager	Mejor forma posible
Amazon Transcribe	Mejor forma posible
AWS Transfer Family	Mejor forma posible
AWS Transit Gateway	Mejor forma posible
Amazon Translate	Duradero
AWS Trusted Advisor	Mejor forma posible
AWS WAF	Mejor forma posible
AWS WAF Regional	Mejor forma posible
AWS Well-Architected Tool	Mejor forma posible
Amazon WorkDocs	Mejor forma posible

Servicio	Tipo de intento
Amazon WorkSpaces	Mejor forma posible
AWS X-Ray	Mejor forma posible

Eventos de gestión generados por AWS los servicios

En general, las API que generan eventos de administración (o de solo lectura) comienzan con los verbos`Describe`, `Get` o `List`. En la siguiente tabla se enumeran AWS los servicios y los eventos de administración que generan y que no siguen esta convención de nomenclatura. Para obtener más información sobre los eventos de administración, consulte ???.

Eventos de administración que no comienzan por **`Describe`**, **`Get`** o **`List`**

En la siguiente tabla se enumeran AWS los servicios y los eventos de administración que generan y que no siguen las convenciones de nomenclatura habituales, que consisten en empezar por `Describe`Get, o`List`.

Servicio	Nombre de evento	Tipo de evento
Alexa for Business	ResolveRoom	Llamada a la API
Alexa for Business	SearchAddressBooks	Llamada a la API
Alexa for Business	SearchContacts	Llamada a la API
Alexa for Business	SearchDevices	Llamada a la API
Alexa for Business	SearchProfiles	Llamada a la API
Alexa for Business	SearchRooms	Llamada a la API
Alexa for Business	SearchSkillGroups	Llamada a la API
Alexa for Business	SearchUsers	Llamada a la API
Analizador de acceso de IAM	ValidatePolicy	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS AdSpace Salas limpias	BatchGetSchema	Llamada a la API
AWS Amplify Creador de interfaz de usuario	ExportComponents	Llamada a la API
AWS Amplify Creador de interfaz de usuario	ExportForms	Llamada a la API
AWS Amplify Creador de interfaz de usuario	ExportThemes	Llamada a la API
OpenSearch Servicio Amazon	BatchGetCollection	Llamada a la API
Amazon API Gateway	ExportApi	Llamada a la API
AWS AppConfig	ValidateConfiguration	Llamada a la API
Amazon AppFlow	RetrieveConnectorData	Llamada a la API
Información sobre CloudWatch aplicaciones de Amazon	UpdateApplicationDashboardConfiguration	Llamada a la API
Amazon Athena	BatchGetNamedQuery	Llamada a la API
Amazon Athena	BatchGetPreparedStatement	Llamada a la API
Amazon Athena	BatchGetQueryExecution	Llamada a la API
Amazon Athena	CheckQueryCompatibility	Llamada a la API
Amazon Athena	ExportNotebook	Llamada a la API
AWS Auto Scaling	AreScalableTargetsRegistered	Llamada a la API
AWS Auto Scaling	Prueba	Llamada a la API
AWS Marketplace	SearchAgreements	Llamada a la API
AWS Backup	CreateLegalHold	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS Backup	ExportBackupPlanTemplate	Llamada a la API
AWS Backup gateway	TestHypervisorConfiguration	Llamada a la API
AWS Billing and Cost Management	AWSPaymentInstrumentGateway.Obtenga	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.DescribeMakePaymentPage	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.DescribePaymentsDashboard	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetAccountPreferences	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetAdvancePaySummary	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetAsoBulkDownload	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetBillingContactAddress	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetDocuments	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetEligiblePaymentInstruments	Acción de consola

Servicio	Nombre de evento	Tipo de evento
AWS Billing and Cost Management	AWSPaymentPortalService.GetEntitiesByIds	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetFundingDocuments	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetKybcValidationStatus	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetOneTimePasswordStatus	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentHistory	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileByArn	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileCurrencies	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfiles	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileServiceProviders	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentsDue	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetRemittanceInformation	Acción de consola

Servicio	Nombre de evento	Tipo de evento
AWS Billing and Cost Management	AWSPaymentPortalService.GetTaxInvoiceMetadata	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetTermsAndConditionsForProgramGroup	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetTransactionsHistory	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetUnappliedFunds	Acción de consola
AWS Billing and Cost Management	AWSPaymentPortalService.GetUnpaidInvoices	Acción de consola
AWS Billing and Cost Management	AWSPaymentPreferenceGateway.Obtener	Acción de consola
AWS Billing and Cost Management	CancelBulkDownload	Acción de consola
AWS Billing and Cost Management	DownloadCommercialInvoice	Acción de consola
AWS Billing and Cost Management	DownloadCsv	Acción de consola
AWS Billing and Cost Management	DownloadDoc	Acción de consola
AWS Billing and Cost Management	Descargar ECSV ForBillingPeriod	Acción de consola
AWS Billing and Cost Management	DownloadPaymentHistory	Acción de consola

Servicio	Nombre de evento	Tipo de evento
AWS Billing and Cost Management	DownloadRegistrationDocument	Acción de consola
AWS Billing and Cost Management	DownloadTaxInvoice	Acción de consola
AWS Billing and Cost Management	FindBankRedirectPaymentInstruments	Acción de consola
AWS Billing and Cost Management	Encuentra CSV ForBillingPeriod	Acción de consola
AWS Billing and Cost Management	ValidateReportDestination	Acción de consola
AWS Billing and Cost Management	VerifyChinaPaymentEligibility	Acción de consola
Amazon Braket	SearchCompilations	Llamada a la API
Amazon Braket	SearchDevices	Llamada a la API
Amazon Braket	SearchQuantumTasks	Llamada a la API
Amazon Connect Cases	BatchGetField	Llamada a la API
Amazon Connect Cases	SearchCases	Llamada a la API
Amazon Connect Cases	SearchRelatedItems	Llamada a la API
Amazon Chime	RetrieveDataExports	Llamada a la API
Amazon Chime	SearchChannels	Llamada a la API
Identidad del Amazon Chime SDK	DeleteProfile	Evento de servicio
Identidad del Amazon Chime SDK	DeleteWorkTalkAccount	Evento de servicio

Servicio	Nombre de evento	Tipo de evento
AWS Salas limpias	BatchGetSchema	Llamada a la API
Amazon Cloud Directory	BatchRead	Llamada a la API
Amazon Cloud Directory	LookupPolicy	Llamada a la API
AWS CloudFormation	DetectStackDrift	Llamada a la API
AWS CloudFormation	DetectStackResourceDrift	Llamada a la API
AWS CloudFormation	DetectStackSetDrift	Llamada a la API
AWS CloudFormation	EstimateTemplateCost	Llamada a la API
AWS CloudFormation	ValidateTemplate	Llamada a la API
AWS CloudShell	RedeemCode	Llamada a la API
AWS CloudTrail	LookupEvents	Llamada a la API
AWS CodeArtifact	ReadFromRepository	Llamada a la API
AWS CodeArtifact	SearchPackages	Llamada a la API
AWS CodeArtifact	VerifyResourcesExistForTagris	Llamada a la API
AWS CodeBuild	BatchGetBuildBatches	Llamada a la API
AWS CodeBuild	BatchGetBuilds	Llamada a la API
AWS CodeBuild	BatchGetProjects	Llamada a la API
AWS CodeBuild	BatchGetReportGroups	Llamada a la API
AWS CodeBuild	BatchGetReports	Llamada a la API
AWS CodeBuild	BatchPutCodeCoverages	Llamada a la API
AWS CodeBuild	BatchPutTestCases	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS CodeBuild	RequestBadge	Evento de servicio
AWS CodeCommit	BatchDescribeMergeConflicts	Llamada a la API
AWS CodeCommit	BatchGetCommits	Llamada a la API
AWS CodeCommit	BatchGetPullRequests	Llamada a la API
AWS CodeCommit	BatchGetRepositories	Llamada a la API
AWS CodeCommit	EvaluatePullRequestApprovalRules	Llamada a la API
AWS CodeCommit	GitPull	Llamada a la API
AWS CodeDeploy	BatchGetApplicationRevisions	Llamada a la API
AWS CodeDeploy	BatchGetApplications	Llamada a la API
AWS CodeDeploy	BatchGetDeploymentGroups	Llamada a la API
AWS CodeDeploy	BatchGetDeploymentInstances	Llamada a la API
AWS CodeDeploy	BatchGetDeployments	Llamada a la API
AWS CodeDeploy	BatchGetDeploymentTargets	Llamada a la API
AWS CodeDeploy	BatchGetOnPremisesInstances	Llamada a la API
Amazon CodeGuru Profiler	BatchGetFrameMetricData	Llamada a la API
Amazon CodeGuru Profiler	SubmitFeedback	Llamada a la API
AWS CodePipeline	PollForJobs	Llamada a la API
AWS CodePipeline	PollForThirdPartyJobs	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
CodeConnections	StartAppRegistrationHandshake	Llamada a la API
CodeConnections	StarTo AuthHandshake	Llamada a la API
CodeConnections	ValidateHostWebhook	Llamada a la API
Amazon CodeWhisperer	CreateCodeScan	Llamada a la API
Amazon CodeWhisperer	CreateProfile	Llamada a la API
Amazon CodeWhisperer	CreateUploadUrl	Llamada a la API
Amazon CodeWhisperer	GenerateRecommendations	Llamada a la API
Amazon CodeWhisperer	UpdateProfile	Llamada a la API
Amazon Cognito Identity	LookupDeveloperIdentity	Llamada a la API
Grupos de usuarios de Amazon Cognito	AdminGetDevice	Llamada a la API
Grupos de usuarios de Amazon Cognito	AdminGetUser	Llamada a la API
Grupos de usuarios de Amazon Cognito	AdminListDevices	Llamada a la API
Grupos de usuarios de Amazon Cognito	AdminListGroupsForUser	Llamada a la API
Grupos de usuarios de Amazon Cognito	AdminListUserAuthEvents	Llamada a la API
Grupos de usuarios de Amazon Cognito	Beta_Authorize_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	Confirm_GET	Evento de servicio

Servicio	Nombre de evento	Tipo de evento
Grupos de usuarios de Amazon Cognito	ConfirmForgotPassword_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	Error_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	ForgotPassword_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	IntrospectToken	Llamada a la API
Grupos de usuarios de Amazon Cognito	Login_Error_POST	Evento de servicio
Grupos de usuarios de Amazon Cognito	Login_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	Mfa_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	MfaOption_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	ResetPassword_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	Signup_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	UserInfo_GET	Evento de servicio
Grupos de usuarios de Amazon Cognito	UserInfo_POST	Evento de servicio
Amazon Cognito Sync	BulkPublish	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon Comprehend	BatchContainsPiiEntities	Llamada a la API
Amazon Comprehend	BatchDetectDominantLanguage	Llamada a la API
Amazon Comprehend	BatchDetectEntities	Llamada a la API
Amazon Comprehend	BatchDetectKeyPhrases	Llamada a la API
Amazon Comprehend	BatchDetectPiiEntities	Llamada a la API
Amazon Comprehend	BatchDetectSentiment	Llamada a la API
Amazon Comprehend	BatchDetectSyntax	Llamada a la API
Amazon Comprehend	BatchDetectTargetedSentiment	Llamada a la API
Amazon Comprehend	ClassifyDocument	Llamada a la API
Amazon Comprehend	ContainsPiiEntities	Llamada a la API
Amazon Comprehend	DetectDominantLanguage	Llamada a la API
Amazon Comprehend	DetectEntities	Llamada a la API
Amazon Comprehend	DetectKeyPhrases	Llamada a la API
Amazon Comprehend	DetectPiiEntities	Llamada a la API
Amazon Comprehend	DetectSentiment	Llamada a la API
Amazon Comprehend	DetectSyntax	Llamada a la API
Amazon Comprehend	DetectTargetedSentiment	Llamada a la API
Amazon Comprehend	DetectToxicContent	Llamada a la API
AWS Compute Optimizer	ExportAutoScalingGroupRecommendations	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS Compute Optimizer	Exportar EBS VolumeRecommendations	Llamada a la API
AWS Compute Optimizer	Exportar EC InstanceRecommendations	Llamada a la API
AWS Compute Optimizer	Exportar ECS ServiceRecommendations	Llamada a la API
AWS Compute Optimizer	ExportLambdaFunctionRecommendations	Llamada a la API
AWS Compute Optimizer	Exportar RDS InstanceRecommendations	Llamada a la API
AWS Config	BatchGetAggregateResourceConfig	Llamada a la API
AWS Config	BatchGetResourceConfig	Llamada a la API
AWS Config	SelectAggregateResourceConfig	Llamada a la API
AWS Config	SelectResourceConfig	Llamada a la API
Amazon Connect	AdminGetEmergencyAccessToken	Llamada a la API
Amazon Connect	SearchQueues	Llamada a la API
Amazon Connect	SearchRoutingProfiles	Llamada a la API
Amazon Connect	SearchSecurityProfiles	Llamada a la API
Amazon Connect	SearchUsers	Llamada a la API
AWS Glue DataBrew	SendProjectSessionAction	Llamada a la API
AWS Data Pipeline	EvaluateExpression	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS Data Pipeline	QueryObjects	Llamada a la API
AWS Data Pipeline	ValidatePipelineDefinition	Llamada a la API
AWS DataSync	VerifyResourcesExistForTagris	Llamada a la API
AWS DeepLens	BatchGetDevice	Llamada a la API
AWS DeepLens	BatchGetModel	Llamada a la API
AWS DeepLens	BatchGetProject	Llamada a la API
AWS DeepLens	CreateDeviceCertificates	Llamada a la API
AWS DeepRacer	AdminGetAccountConfig	Llamada a la API
AWS DeepRacer	AdminListAssociatedUsers	Llamada a la API
AWS DeepRacer	TestRewardFunction	Llamada a la API
AWS DeepRacer	VerifyResourcesExistForTagris	Llamada a la API
Amazon Detective	BatchGetGraphMemberDatasources	Llamada a la API
Amazon Detective	BatchGetMembershipDatasources	Llamada a la API
Amazon Detective	SearchGraph	Llamada a la API
El DevOps gurú de Amazon	SearchInsights	Llamada a la API
El DevOps gurú de Amazon	SearchOrganizationInsights	Llamada a la API
AWS Database Migration Service	BatchStartRecommendations	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS Database Migration Service	ModifyRecommendation	Llamada a la API
AWS Database Migration Service	StartRecommendations	Llamada a la API
AWS Database Migration Service	VerifyResourcesExistForTagris	Llamada a la API
AWS Directory Service	VerifyTrust	Llamada a la API
Amazon Elastic Compute Cloud	ConfirmProductInstance	Llamada a la API
Amazon Elastic Compute Cloud	ReportInstanceStatus	Llamada a la API
Amazon Elastic Container Registry	BatchCheckLayerAvailability	Llamada a la API
Amazon Elastic Container Registry	BatchGetImage	Llamada a la API
Amazon Elastic Container Registry	BatchGetImageReferrer	Llamada a la API
Amazon Elastic Container Registry	BatchGetRepository ScanningConfiguration	Llamada a la API
Amazon Elastic Container Registry	DryRunEvent	Evento de servicio
Amazon Elastic Container Registry	PolicyExecutionEvent	Evento de servicio
Amazon Elastic Container Registry Public	BatchCheckLayerAvailability	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon Elastic Container Service	DiscoverPollEndpoint	Llamada a la API
Amazon Elastic Container Service	FindSubfleetRoute	Llamada a la API
Amazon Elastic Container Service	ValidateResources	Llamada a la API
Amazon Elastic Container Service	VerifyTaskSetsExist	Llamada a la API
Amazon Elastic Kubernetes Service	AccessKubernetesApi	Llamada a la API
AWS Elastic Beanstalk	CheckDNSAvailability	Llamada a la API
AWS Elastic Beanstalk	RequestEnvironmentInfo	Llamada a la API
AWS Elastic Beanstalk	RetrieveEnvironmentInfo	Llamada a la API
AWS Elastic Beanstalk	ValidateConfigurationSettings	Llamada a la API
Amazon Elastic File System	NewClientConnection	Evento de servicio
Amazon Elastic File System	UpdateClientConnection	Evento de servicio
Amazon Elastic Transcoder	ReadJob	Llamada a la API
Amazon Elastic Transcoder	ReadPipeline	Llamada a la API
Amazon Elastic Transcoder	ReadPreset	Llamada a la API
Amazon EventBridge	TestEventPattern	Llamada a la API
Amazon EventBridge	TestScheduleExpression	Llamada a la API
Amazon FinSpace API	BatchListCatalogNodesByDataset	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon FinSpace API	BatchListNodesByDataset	Llamada a la API
Amazon FinSpace API	BatchValidateAccess	Llamada a la API
Amazon FinSpace API	CreateAuditRecordsQuery	Llamada a la API
Amazon FinSpace API	SearchDatasets	Llamada a la API
Amazon FinSpace API	SearchDatasetsV	Llamada a la API
Amazon FinSpace API	ValidateIdToken	Llamada a la API
AWS Firewall Manager	DisassociateAdminAccount	Llamada a la API
Amazon Forecast	InvokeForecastEndpoint	Llamada a la API
Amazon Forecast	QueryFeature	Llamada a la API
Amazon Forecast	QueryForecast	Llamada a la API
Amazon Forecast	QueryWhatIfForecast	Llamada a la API
Amazon Forecast	VerifyResourcesExistForTagris	Llamada a la API
Amazon Fraud Detector	BatchGetVariable	Llamada a la API
Amazon Fraud Detector	VerifyResourcesExistForTagris	Llamada a la API
FreeRTOS	VerifyEmailAddress	Llamada a la API
Amazon GameLift	RequestUploadCredentials	Llamada a la API
Amazon GameLift	ResolveAlias	Llamada a la API
Amazon GameLift	SearchGameSessions	Llamada a la API
Amazon GameLift	ValidateMatchmakingRuleSet	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon GameSparks	ExportSnapshot	Llamada a la API
Amazon Location Service	BatchGetDevicePosition	Llamada a la API
Amazon Location Service	CalculateRoute	Llamada a la API
Amazon Location Service	CalculateRouteMatrix	Llamada a la API
Amazon Location Service	SearchPlaceIndexForPosition	Llamada a la API
Amazon Location Service	SearchPlaceIndexForSuggestions	Llamada a la API
Amazon Location Service	SearchPlaceIndexForText	Llamada a la API
Amazon S3 Glacier	InitiateJob	Llamada a la API
AWS Glue	BatchGetBlueprints	Llamada a la API
AWS Glue	BatchGetColumnStatisticsForTable	Llamada a la API
AWS Glue	BatchGetCrawlers	Llamada a la API
AWS Glue	BatchGetCustomEntityTypes	Llamada a la API
AWS Glue	BatchGetDataQualityResult	Llamada a la API
AWS Glue	BatchGetDevEndpoints	Llamada a la API
AWS Glue	BatchGetJobs	Llamada a la API
AWS Glue	BatchGetML Transform	Llamada a la API
AWS Glue	BatchGetPartition	Llamada a la API
AWS Glue	BatchGetTriggers	Llamada a la API
AWS Glue	BatchGetWorkflows	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS Glue	QueryJobRuns	Llamada a la API
AWS Glue	QueryJobRunsAggregated	Llamada a la API
AWS Glue	QueryJobs	Llamada a la API
AWS Glue	QuerySchemaVersion Metadata	Llamada a la API
AWS Glue	SearchTables	Llamada a la API
AWS HealthLake	ReadResource	Llamada a la API
AWS HealthLake	SearchWithGet	Llamada a la API
AWS HealthLake	SearchWithPost	Llamada a la API
AWS Identity and Access Management	GenerateCredentialReport	Llamada a la API
AWS Identity and Access Management	GenerateOrganizationsAccess Report	Llamada a la API
AWS Identity and Access Management	GenerateServiceLas tAccessedDetails	Llamada a la API
AWS Identity and Access Management	SimulateCustomPolicy	Llamada a la API
AWS Identity and Access Management	SimulatePrincipalPolicy	Llamada a la API
AWS Tienda de identidades	IsMemberInGroups	Llamada a la API
AWS Autenticación de Identity Store	BatchGetSession	Llamada a la API
Amazon Inspector Classic	PreviewAgents	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon Inspector Classic	BatchGetAccountStatus	Llamada a la API
Amazon Inspector Classic	BatchGetFreeTrialInfo	Llamada a la API
Amazon Inspector Classic	BatchGetMember	Llamada a la API
Facturación de AWS	ValidateDocumentDeliveryS3LocationInfo	Llamada a la API
AWS IoT	SearchIndex	Llamada a la API
AWS IoT	TestAuthorization	Llamada a la API
AWS IoT	TestInvokeAuthorizer	Llamada a la API
AWS IoT	ValidateSecurityProfileBehaviors	Llamada a la API
AWS IoT Analytics	SampleChannelData	Llamada a la API
AWS IoT SiteWise	GatewaysVerifyResourcesExistForTagrisInternal	Llamada a la API
AWS IoT Things Graph	SearchEntities	Llamada a la API
AWS IoT Things Graph	SearchFlowExecutions	Llamada a la API
AWS IoT Things Graph	SearchFlowTemplates	Llamada a la API
AWS IoT Things Graph	SearchSystemInstances	Llamada a la API
AWS IoT Things Graph	SearchSystemTemplates	Llamada a la API
AWS IoT Things Graph	SearchThings	Llamada a la API
AWS IoT TwinMaker	ExecuteQuery	Llamada a la API
AWS IoT Wireless	CreateNetworkAnalyzerConfiguration	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS IoT Wireless	DeleteNetworkAnalyzerConfiguration	Llamada a la API
AWS IoT Wireless	DeregisterWirelessDevice	Llamada a la API
Amazon Interactive Video Service	BatchGetChannel	Llamada a la API
Amazon Interactive Video Service	BatchGetStreamKey	Llamada a la API
Amazon Kendra	BatchGetDocumentStatus	Llamada a la API
Amazon Kendra	Consultar	Llamada a la API
Amazon Managed Service para Apache Flink	DiscoverInputSchema	Llamada a la API
AWS Key Management Service	Decrypt	Llamada a la API
AWS Key Management Service	Encrypt	Llamada a la API
AWS Key Management Service	GenerateDataKey	Llamada a la API
AWS Key Management Service	GenerateDataKeyPair	Llamada a la API
AWS Key Management Service	GenerateDataKeyPairWithoutPlaintext	Llamada a la API
AWS Key Management Service	GenerateDataKeyWithoutPlaintext	Llamada a la API
AWS Key Management Service	GenerateMac	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS Key Management Service	GenerateRandom	Llamada a la API
AWS Key Management Service	ReEncrypt	Llamada a la API
AWS Key Management Service	Sign	Llamada a la API
AWS Key Management Service	Verificar	Llamada a la API
AWS Key Management Service	VerifyMac	Llamada a la API
AWS Lake Formation	SearchDatabasesByEtiquetas LF	Llamada a la API
AWS Lake Formation	SearchTablesByEtiquetas LF	Llamada a la API
AWS Lake Formation	StartQueryPlanning	Llamada a la API
Amazon Lex	BatchCreateCustomV ocabularyItem	Llamada a la API
Amazon Lex	BatchDeleteCustomV ocabularyItem	Llamada a la API
Amazon Lex	BatchUpdateCustomV ocabularyItem	Llamada a la API
Amazon Lex	DeleteCustomVocabulary	Llamada a la API
Amazon Lex	SearchAssociatedTranscripts	Llamada a la API
Amazon Lightsail	Crear GUI SessionAc cessDetails	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon Lightsail	DownloadDefaultKeyPair	Llamada a la API
Amazon Lightsail	IsVpcPeered	Llamada a la API
Amazon CloudWatch Logs	FilterLogEvents	Llamada a la API
Amazon Macie	BatchGetCustomDataIdentifiers	Llamada a la API
Amazon Macie	UpdateFindingsFilter	Llamada a la API
AWS Elemental MediaConnect	ManagedDescribeFlow	Llamada a la API
AWS Elemental MediaConnect	PrivateDescribeFlowMeta	Llamada a la API
AWS Application Migration Service	OperationalDescribeJobLogItems	Llamada a la API
AWS Application Migration Service	OperationalDescribeJobs	Llamada a la API
AWS Application Migration Service	OperationalDescribeReplicationConfigurationTemplates	Llamada a la API
AWS Application Migration Service	OperationalDescribeSourceServer	Llamada a la API
AWS Application Migration Service	OperationalGetLaunchConfiguration	Llamada a la API
AWS Application Migration Service	OperationalListSourceServers	Llamada a la API
AWS Application Migration Service	VerifyClientRoleForMgn	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS HealthOmics	VerifyResourceExists	Llamada a la API
AWS HealthOmics	VerifyResourcesExistForTagris	Llamada a la API
Amazon Polly	SynthesizeLongSpeech	Llamada a la API
Amazon Polly	SynthesizeSpeech	Llamada a la API
Amazon Polly	SynthesizeSpeechGet	Llamada a la API
AWS servicio que proporciona redes privadas gestionadas	Ping	Llamada a la API
AWS Proton	DeleteEnvironmentTemplateVersion	Llamada a la API
AWS Proton	DeleteServiceTemplateVersion	Llamada a la API
Amazon QLDB	ShowCatalog	Llamada a la API
Amazon QuickSight	GenerateEmbedUrlForAnonymousUser	Llamada a la API
Amazon QuickSight	GenerateEmbedUrlForRegisteredUser	Llamada a la API
Amazon QuickSight	QueryDatabase	Evento de servicio
Amazon QuickSight	SearchAnalyses	Llamada a la API
Amazon QuickSight	SearchDashboards	Llamada a la API
Amazon QuickSight	SearchDataSets	Llamada a la API
Amazon QuickSight	SearchDataSources	Llamada a la API
Amazon QuickSight	SearchFolders	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon QuickSight	SearchGroups	Llamada a la API
Amazon QuickSight	SearchUsers	Llamada a la API
Amazon Relational Database Service	DownloadCompleteDB LogFile	Llamada a la API
Amazon Relational Database Service	Descargar DB LogFilePortion	Llamada a la API
Amazon Rekognition	CompareFaces	Llamada a la API
Amazon Rekognition	DetectCustomLabels	Llamada a la API
Amazon Rekognition	DetectFaces	Llamada a la API
Amazon Rekognition	DetectLabels	Llamada a la API
Amazon Rekognition	DetectModerationLabels	Llamada a la API
Amazon Rekognition	DetectProtectiveEquipment	Llamada a la API
Amazon Rekognition	DetectText	Llamada a la API
Amazon Rekognition	RecognizeCelebrities	Llamada a la API
Amazon Rekognition	SearchFaces	Llamada a la API
Amazon Rekognition	SearchFacesByImage	Llamada a la API
Amazon Rekognition	SearchUsers	Llamada a la API
Amazon Rekognition	SearchUsersByImage	Llamada a la API
Explorador de recursos de AWS	BatchGetView	Llamada a la API
Explorador de recursos de AWS	Búsqueda	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS Resource Groups	SearchResources	Llamada a la API
AWS Resource Groups	ValidateResourceSharing	Llamada a la API
AWS RoboMaker	BatchDescribeSimulationJob	Llamada a la API
Amazon Route 53	TestDNSAnswer	Llamada a la API
Amazon Route 53 Dominios	checkAvailabilities	Llamada a la API
Amazon Route 53 Dominios	CheckDomainAvailability	Llamada a la API
Amazon Route 53 Dominios	checkDomainTransferability	Llamada a la API
Amazon Route 53 Dominios	CheckDomainTransferability	Llamada a la API
Amazon Route 53 Dominios	isEmailReachable	Llamada a la API
Amazon Route 53 Dominios	searchDomains	Llamada a la API
Amazon Route 53 Dominios	sendVerificationMessage	Llamada a la API
Amazon Route 53 Dominios	ViewBilling	Llamada a la API
Amazon Route 53 Dominios	viewBilling	Llamada a la API
Amazon CloudWatch RUM	BatchGetRumMetricDefinitions	Llamada a la API
Amazon Simple Storage Service	echo	Llamada a la API
Amazon Simple Storage Service	GenerateInventory	Evento de servicio
Amazon SageMaker	BatchDescribeModelPackage	Llamada a la API
Amazon SageMaker	DeleteModelCard	Llamada a la API
Amazon SageMaker	QueryLineage	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon SageMaker	RenderUiTemplate	Llamada a la API
Amazon SageMaker	Búsqueda	Llamada a la API
EventBridge Esquemas de Amazon	ExportSchema	Llamada a la API
EventBridge Esquemas de Amazon	SearchSchemas	Llamada a la API
Amazon SimpleDB	DomainMetadata	Llamada a la API
AWS Secrets Manager	ValidateResourcePolicy	Llamada a la API
AWS Service Catalog	ScanProvisionedProducts	Llamada a la API
AWS Service Catalog	SearchProducts	Llamada a la API
AWS Service Catalog	SearchProductsAsAdmin	Llamada a la API
AWS Service Catalog	SearchProvisionedProducts	Llamada a la API
Amazon SES	BatchGetMetricData	Llamada a la API
Amazon SES	TestRenderEmailTemplate	Llamada a la API
Amazon SES	TestRenderTemplate	Llamada a la API
Amazon Simple Notification Service	CheckIfPhoneNumber IsOptedOut	Llamada a la API
AWS SQL Workbench	BatchGetNotebookCell	Llamada a la API
AWS SQL Workbench	ExportNotebook	Llamada a la API
Amazon EC2, Systems Manager	ExecuteApi	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
AWS Systems Manager Incident Manager	DeleteContactChannel	Llamada a la API
AWS IAM Identity Center	IsMemberInGroup	Llamada a la API
AWS IAM Identity Center	SearchGroups	Llamada a la API
AWS IAM Identity Center	SearchUsers	Llamada a la API
AWS STS	AssumeRole	Llamada a la API
AWS STS	AssumeRoleWithSAML	Llamada a la API
AWS STS	AssumeRoleWithWebIdentity	Llamada a la API
AWS STS	DecodeAuthorizationMessage	Llamada a la API
AWS Configuración de impuestos	BatchGetTaxExemptions	Llamada a la API
AWS WAFV2	CheckCapacity	Llamada a la API
AWS WAFV2	GenerateMobileSdkReleaseUrl	Llamada a la API
AWS Well-Architected Tool	ExportLens	Llamada a la API
AWS Well-Architected Tool	TagResource	Llamada a la API
AWS Well-Architected Tool	UntagResource	Llamada a la API
AWS Well-Architected Tool	UpdateGlobalSettings	Llamada a la API
Amazon Connect Wisdom	QueryAssistant	Llamada a la API
Amazon Connect Wisdom	SearchContent	Llamada a la API
Amazon Connect Wisdom	SearchSessions	Llamada a la API
Amazon WorkDocs	AbortDocumentVersionUpload	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon WorkDocs	AddUsersToGroup	Llamada a la API
Amazon WorkDocs	BatchGetUsers	Llamada a la API
Amazon WorkDocs	CheckAlias	Llamada a la API
Amazon WorkDocs	CompleteDocumentVersionUpload	Llamada a la API
Amazon WorkDocs	CreateAnnotation	Llamada a la API
Amazon WorkDocs	CreateComment	Llamada a la API
Amazon WorkDocs	CreateFeedbackRequest	Llamada a la API
Amazon WorkDocs	CreateFolder	Llamada a la API
Amazon WorkDocs	CreateGroup	Llamada a la API
Amazon WorkDocs	CreateShare	Llamada a la API
Amazon WorkDocs	CreateUser	Llamada a la API
Amazon WorkDocs	DeleteAnnotation	Llamada a la API
Amazon WorkDocs	DeleteComment	Llamada a la API
Amazon WorkDocs	DeleteDocument	Llamada a la API
Amazon WorkDocs	DeleteFeedbackRequest	Llamada a la API
Amazon WorkDocs	DeleteFolder	Llamada a la API
Amazon WorkDocs	DeleteFolderContents	Llamada a la API
Amazon WorkDocs	DeleteGroup	Llamada a la API
Amazon WorkDocs	DeleteOrganizationShare	Llamada a la API
Amazon WorkDocs	DeleteUser	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon WorkDocs	DownloadDocumentVersion	Llamada a la API
Amazon WorkDocs	DownloadDocumentVersionUnderlays	Llamada a la API
Amazon WorkDocs	InitiateDocumentVersionUpload	Llamada a la API
Amazon WorkDocs	LogoutUser	Llamada a la API
Amazon WorkDocs	PaginatedOrganizationActivity	Llamada a la API
Amazon WorkDocs	PublishAnnotations	Llamada a la API
Amazon WorkDocs	PublishComments	Llamada a la API
Amazon WorkDocs	RestoreDocument	Llamada a la API
Amazon WorkDocs	RestoreFolder	Llamada a la API
Amazon WorkDocs	SearchGroups	Llamada a la API
Amazon WorkDocs	SearchOrganizationUsers	Llamada a la API
Amazon WorkDocs	TransferUserResources	Llamada a la API
Amazon WorkDocs	UpdateAnnotation	Llamada a la API
Amazon WorkDocs	UpdateComment	Llamada a la API
Amazon WorkDocs	UpdateDocument	Llamada a la API
Amazon WorkDocs	UpdateDocumentVersion	Llamada a la API
Amazon WorkDocs	UpdateFolder	Llamada a la API
Amazon WorkDocs	UpdateGroup	Llamada a la API
Amazon WorkDocs	UpdateOrganization	Llamada a la API

Servicio	Nombre de evento	Tipo de evento
Amazon WorkDocs	UpdateUser	Llamada a la API
Amazon WorkMail	AssumeImpersonationRole	Llamada a la API
Amazon WorkMail	QueryDnsRecords	Llamada a la API
Amazon WorkMail	SearchMembers	Llamada a la API
Amazon WorkMail	TestAvailabilityConfiguration	Llamada a la API
Amazon WorkMail	TestInboundMailFlowRules	Llamada a la API
Amazon WorkMail	TestOutboundMailFlowRules	Llamada a la API

EventBridge referencia detallada de eventos

EventBridge emite por sí mismo los siguientes eventos. Estos eventos se envían automáticamente al bus de eventos predeterminado, como ocurre con cualquier otro AWS servicio.

Para ver las definiciones de los campos de metadatos que se incluyen en todos los eventos, consultethe section called “Referencia de la estructura de un evento”.

Temas

- Evento programado
- Esquema creado
- Versión del esquema creada

Evento programado

A continuación se muestran los campos de detalle del `Scheduled Event` evento.

Los `detail-type` campos `source` y se incluyen porque contienen valores específicos para los EventBridge eventos. Para ver las definiciones de los demás campos de metadatos que se incluyen en todos los eventos, consultethe section called “Referencia de la estructura de un evento”.

```
{
  . . .,
```

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

`detail-type`

Identifica el tipo de evento.

Para este evento, este valor es`Scheduled Event`.

Obligatorio: sí

`source`

Identifica el servicio que generó el evento. Para EventBridge los eventos, este valor es`aws.events`.

Obligatorio: sí

`detail`

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

Obligatorio: sí

No hay campos obligatorios en este objeto para `Scheduled Event` los eventos.

Example Ejemplo de evento programado

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

Esquema creado

A continuación se muestran los campos de detalle del `Schema Created` evento.

Cuando se crea un esquema, EventBridge envía un evento `Schema Created` y un `Schema Version Created` evento.

Los `detail-type` campos `source` y se incluyen porque contienen valores específicos para los EventBridge eventos. Para ver las definiciones de los demás campos de metadatos que se incluyen en todos los eventos, consultethe section called “Referencia de la estructura de un evento”.

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

`detail-type`

Identifica el tipo de evento.

Para este evento, este valor es`Schema Created`.

Obligatorio: sí

`source`

Identifica el servicio que generó el evento. Para EventBridge los eventos, este valor es`aws.schemas`.

Obligatorio: sí

`detail`

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

Obligatorio: sí

Para este evento, estos datos incluyen:

`SchemaName`

El nombre del esquema.

Obligatorio: sí

`SchemaType`

El tipo de esquema.

Valores válidos: `OpenApi3` | `JSONSchemaDraft4`

Obligatorio: sí

`RegistryName`

Nombre del registro que contiene el esquema.

Obligatorio: sí

`CreationDate`

La fecha en que se creó el esquema.

Obligatorio: sí

`Version`

La versión del esquema.

En el `Schema Created` caso de los eventos, este valor siempre será`1`.

Obligatorio: sí

Example Ejemplo: Esquema: evento creado

```
{
  "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"
  }
}
```

Versión del esquema creada

A continuación se muestran los campos de detalle del `Schema Version Created` evento.

Cuando se crea un esquema, EventBridge envía un evento `Schema Created` y un `Schema Version Created` evento.

Los `detail-type` campos `source` y se incluyen porque contienen valores específicos para los EventBridge eventos. Para ver las definiciones de los demás campos de metadatos que se incluyen en todos los eventos, consultethe section called “Referencia de la estructura de un evento”.

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

`detail-type`

Identifica el tipo de evento.

Para este evento, este valor es`Schema Version Created`.

Obligatorio: sí

`source`

Identifica el servicio que generó el evento. Para EventBridge los eventos, este valor es`aws.schemas`.

Obligatorio: sí

`detail`

Un objeto JSON que contiene información sobre el evento. El servicio que genera el evento determina el contenido de este campo.

Obligatorio: sí

Para este evento, estos datos incluyen:

`SchemaName`

El nombre del esquema.

Obligatorio: sí

`SchemaType`

El tipo de esquema.

Valores válidos: `OpenApi3` | `JSONSchemaDraft4`

Obligatorio: sí

`RegistryName`

Nombre del registro que contiene el esquema.

Obligatorio: sí

`CreationDate`

La fecha en que se creó la versión del esquema.

Obligatorio: sí

`Version`

La versión del esquema.

Obligatorio: sí

Example Ejemplo de evento creado por la versión del esquema

```
{
  "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"
  }
}
```

Recibir eventos de un socio de SaaS de Amazon EventBridge

Para recibir eventos de aplicaciones y servicios de socios de SaaS, debe disponer de un origen de eventos de socios que le ofrezca el socio. A continuación, puede crear un bus de eventos de socios y asociarlo al origen de eventos del socio.

El siguiente vídeo describe las integraciones de SaaS con EventBridge: socios de software como servicio (SaaS)

Temas

- Integraciones de socios de SaaS compatibles
- Configuración de Amazon EventBridge para recibir eventos de una integración de SaaS
- Creación de una regla que coincida con los eventos de socios de SaaS
- Recibir eventos mediante las direcciones URL de las funciones AWS Lambda
- Recepción de eventos de Salesforce

Integraciones de socios de SaaS compatibles

EventBridge admite las siguientes integraciones de socios de SaaS:

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

- Stripe
- SugarCRM
- SugarCRM
- Symantec
- Thundra
- TriggerMesh
- Whispir
- Zendesk
- API de Amazon Seller Partner

Los orígenes de eventos de socios están disponibles en las siguientes regiones.

Code	Nombre
us-east-1	Este de EE. UU. (Norte de Virginia)
us-east-2	Este de EE. UU. (Ohio)
us-west-1	Oeste de EE. UU. (Norte de California)
us-west-2	Oeste de EE. UU. (Oregón)
ca-central-1	Canadá (centro)
eu-central-1	Europa (Fráncfort)
eu-central-2	Europa (Zúrich)
eu-west-1	Europa (Irlanda)
eu-west-2	Europa (Londres)
eu-west-3	Europa (París)
eu-north-1	Europa (Estocolmo)
eu-south-1	Europa (Milán)

Code	Nombre
eu-south-2	Europa (España)
af-south-1	África (Ciudad del Cabo)
ap-south-1	Asia-Pacífico (Bombay)
ap-south-2	Asia-Pacífico (Hyderabad)
ap-east-1	Asia-Pacífico (Hong Kong)
ap-northeast-1	Asia-Pacífico (Tokio)
ap-northeast-2	Asia-Pacífico (Seúl)
ap-northeast-3	Asia-Pacífico (Osaka)
ap-southeast-1	Asia-Pacífico (Singapur)
ap-southeast-2	Asia-Pacífico (Sídney)
ap-southeast-3	Asia-Pacífico (Yakarta)
ap-southeast-4	Asia-Pacífico (Melbourne)
cn-north-1	China (Pekín)
cn-northwest-1	China (Ningxia)
me-central-1	Medio Oriente (EAU)
me-south-1	Medio Oriente (Baréin)
sa-east-1	América del Sur (São Paulo)
il-central-1	Israel (Tel Aviv)

Configuración de Amazon EventBridge para recibir eventos de una integración de SaaS

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Orígenes de eventos de socios.
3. Busque el socio que desee y seleccione Configurar para ese socio.
4. Seleccione Copiar para copiar el ID de cuenta en el portapapeles.
5. En el panel de navegación, seleccione Orígenes de eventos de socios.
6. Vaya al sitio web del socio y siga las instrucciones para crear un origen de eventos de socio usando el ID de cuenta. El origen de eventos que cree estará disponible solo para la cuenta.
7. Vuelve a la EventBridge consola y selecciona Fuentes de eventos asociadas en el panel de navegación.
8. Seleccione el botón situado junto al origen de eventos de socios y, seguidamente, seleccione Asociar con bus de eventos.

   El estado del origen de eventos cambia de `Pending` a `Active` y el nombre del bus de eventos se actualiza para que coincida con el nombre del origen de eventos del socio. Ahora puede comenzar a crear reglas que coincidan con eventos procedentes de ese origen de eventos de socio. Para obtener más información, consulte Creación de una regla que coincida con los eventos de socios de SaaS.

   > **Note**
   >
   > Todos los eventos publicados por un socio en un origen de eventos de socios que no esté asociado a un bus de eventos se eliminarán inmediatamente. Esos eventos no se prolongarán mientras estén inactivos. EventBridge

Creación de una regla que coincida con los eventos de socios de SaaS

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Elija Crear regla.
4. Escriba un nombre y una descripción para la regla.

Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.

5. En Bus de eventos, seleccione el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione Bus de eventos predeterminado de AWS . Cuando un servicio de AWS en la cuenta emite un evento, siempre va al bus de eventos predeterminado de la cuenta.
6. En Tipo de regla, seleccione Regla con un patrón de eventos.
7. Seleccione Siguiente.
8. En Origen del evento, seleccione Otro.
9. (Opcional) Para ver ejemplos de eventos, seleccione el tipo de evento.
10. En patrón de eventos, introduzca un patrón de eventos JSON.
11. Seleccione Siguiente.
12. En Tipos de destino, seleccione Servicio de AWS .
13. En Selecciona un objetivo, elige el AWS servicio al que deseas enviar la información cuando EventBridge detecte un evento que coincida con el patrón del evento.
14. Los campos mostrados varían en función del servicio que seleccione. Introduzca la información específica de este tipo de destino, según sea necesario.
15. Para muchos tipos de objetivos, EventBridge necesita permisos para enviar eventos al objetivo. En estos casos, EventBridge puede crear la función de IAM necesaria para que se ejecute la regla. Realice una de las siguientes acciones siguientes:

    - Para crear un rol de IAM automáticamente, seleccione Crear un nuevo rol para este recurso específico.
    - Para utilizar un rol de IAM que haya creado antes, seleccione Usar rol existente y seleccione el rol existente del menú desplegable.

16. (Opcional) En Configuración adicional, haga lo siguiente:

    a. En Antigüedad máxima del evento, indique un valor entre un minuto (00:01) y 24 horas (24:00).

    b. En Cantidad de reintentos, indique un número entre 0 y 185.

    c. En el caso de la cola de cartas sin salida, elija si desea utilizar una cola estándar de Amazon SQS como cola de cartas sin salida. EventBridge envía los eventos que cumplen

con esta regla a la lista de espera si no se entregan correctamente al destino. Realice una de las siguientes acciones siguientes:

- Seleccione Ninguna para no usar una cola de mensajes fallidos.
- Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para utilizarla como cola de mensajes fallidos y, a continuación, seleccione de la lista desplegable la cola que quiera usar.
- Elija Seleccione una cola de Amazon SQS en otra AWS cuenta como cola de letra muerta y, a continuación, introduzca el ARN de la cola que desee utilizar. Debe adjuntar a la cola una política basada en recursos que le conceda permiso para enviarle mensajes. EventBridge Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.

17. (Opcional) Seleccione Agregar otro destino para agregar otro destino para esta regla.
18. Seleccione Siguiente.
19. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulte EventBridge Etiquetas de Amazon.
20. Seleccione Siguiente.
21. Revise los detalles de la regla y seleccione Crear regla.

Recibir eventos mediante las direcciones URL de las funciones AWS Lambda

> **Note**
>
> Para que nuestros socios puedan acceder al Webhook entrante, vamos a crear un Open Lambda en su AWS cuenta que está protegido a nivel de aplicación Lambda mediante la verificación de la firma de autenticación enviada por el socio externo. Revise esta configuración con el equipo de seguridad. Para obtener más información, consulte Modelo de seguridad y autenticación para URL de funciones de Lambda.

Tu bus de EventBridge eventos de Amazon puede usar una URL de AWS Lambda función creada por una AWS CloudFormation plantilla para recibir eventos de proveedores de SaaS compatibles. Con las URL de función, los datos del evento se envían a una función de Lambda. A continuación, la función convierte estos datos en un evento que puede ser ingerido EventBridge y enviado a un bus de eventos para su procesamiento. Una vez que el evento esté en un bus de eventos, puede usar reglas para filtrar los eventos, aplicar transformaciones de entrada configurada y, a continuación, dirigirlo al destino correcto.

> **Note**
>
> La creación de direcciones URL de función de Lambda aumentará sus costes mensuales. Para más información, consulte Precios de AWS Lambda.

Para configurar una conexión EventBridge, primero debe seleccionar el proveedor de SaaS con el que desea configurar una conexión. A continuación, proporciona un secreto de firma que ha creado con ese proveedor y selecciona el bus de EventBridge eventos al que desea enviar los eventos. Por último, utilizas una AWS CloudFormation plantilla y creas los recursos necesarios para completar la conexión.

Los siguientes proveedores de SaaS están disponibles actualmente para su uso con las URL de EventBridge funciones Lambda:

- GitHub
- Twilio

Temas

Configurar una conexión a GitHub:

Paso 1: Crear la pila AWS CloudFormation

Primero, usa la EventBridge consola de Amazon para crear una CloudFormation pila:

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Inicios rápidos.
3. En Webhooks entrantes con URL de Lambda, seleccione Comenzar.
4. En GitHub, seleccione Configurar.
5. En Paso 1: Seleccionar un bus de eventos, seleccione un bus de eventos de la lista desplegable. Este bus de eventos recibe datos de la URL de función de Lambda que proporciona a GitHub. También puede crear un bus de eventos seleccionando Nuevo bus de eventos.
6. En el paso 2: Configurar el uso CloudFormation, selecciona Nuevo GitHub webhook.
7. Seleccione Reconozco que el Webhook entrante que cree será de acceso público y seleccione Confirmar.
8. Escriba un nombre para la pila.
9. En los parámetros, compruebe que aparece el bus de eventos correcto y, a continuación, especifique un token seguro para el GitHubWebhookSecret. Para obtener más información sobre la creación de un token seguro, consulte Configuración del token secreto en la documentación sobre GitHub.
10. En Capacidades y transformaciones, seleccione cada una de las siguientes opciones:

    - Reconozco que eso AWS CloudFormation podría crear recursos de IAM.

- Reconozco que eso AWS CloudFormation podría crear recursos de IAM con nombres personalizados.
- Reconozco que eso AWS CloudFormation podría requerir la siguiente capacidad: **`CAPABILITY_AUTO_EXPAND`**

11. Seleccione Crear pila.

Paso 2: Crear un webhook de GitHub

A continuación, cree el webhook en GitHub. Necesitará el token seguro y la dirección URL de función de Lambda creada en el paso 2 para completar este paso. Para obtener más información, consulte Creación de webhooks en la documentación de GitHub.

Configurar una conexión a un Twilio

Paso 1: Buscar el token de autenticación de Twilio

Para configurar una conexión entre Twilio y EventBridge, primero configura la conexión Twilio con el token de autenticación, o secreto, de tu Twilio cuenta. Para obtener más información, consulte Tokens de autenticación y cómo cambiarlos en la documentación sobre Twilio.

Paso 2: Crea la pila AWS CloudFormation

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Inicios rápidos.
3. En Webhooks entrantes con URL de Lambda, seleccione Comenzar.
4. En Twilio, seleccione Configurar.
5. En Paso 1: Seleccionar un bus de eventos, seleccione un bus de eventos de la lista desplegable. Este bus de eventos recibe datos de la URL de función de Lambda que proporciona a Twilio. También puede crear un bus de eventos seleccionando Nuevo bus de eventos.
6. En el paso 2: Configurar el uso CloudFormation, selecciona Nuevo Twilio webhook.
7. Seleccione Reconozco que el Webhook entrante que cree será de acceso público y seleccione Confirmar.
8. Escriba un nombre para la pila.
9. En los parámetros, compruebe que aparece el bus de eventos correcto y, a continuación, introduzca el TwilioWebhookSecret que creó en el paso 1.
10. En Capacidades y transformaciones, seleccione cada una de las siguientes opciones:

- Reconozco que eso AWS CloudFormation podría crear recursos de IAM.
- Reconozco que eso AWS CloudFormation podría crear recursos de IAM con nombres personalizados.
- Reconozco que AWS CloudFormation podría requerir la siguiente capacidad: CAPABILITY_AUTO_EXPAND

11. Seleccione Crear pila.

Paso 3: Crear un webhook de Twilio

Después de configurar la URL de función de Lambda, debe entregársela a Twilio para enviar los datos del evento. Para obtener más información, consulte Configuración de la dirección URL pública con Twilio en la documentación sobre Twilio.

Actualizar el token de autenticación o el secreto del webhook

Actualizar el secreto de GitHub

> **Note**
>
> GitHub no admite dos secretos a la vez. Es posible que se produzca un tiempo de inactividad de los recursos mientras el GitHub secreto y el secreto de la AWS CloudFormation pila no estén sincronizados. GitHubLos mensajes que se envíen mientras los secretos no estén sincronizados fallarán debido a que las firmas son incorrectas. Espera a que los CloudFormation secretos GitHub y los secretos estén sincronizados e inténtalo de nuevo.

1. Crea un secreto de GitHub nuevo. Para obtener más información, consulte Secreto cifrados en la documentación sobre GitHub.
2. Abre la AWS CloudFormation consola en https://console.aws.amazon.com/cloudformation.
3. En el panel de navegación, seleccione Pilas.
4. Seleccione la pila para el webhook que incluye el secreto que desea actualizar.
5. Seleccione Actualizar.
6. Asegúrese de que la opción Usar plantilla actual esté seleccionada y seleccione Siguiente.
7. En GitHubWebhookSecret, desactive Usar un valor existente, introduzca el nuevo GitHub secreto que creó en el paso 1 y seleccione Siguiente.

8. Seleccione Siguiente.
9. Seleccione Actualizar pila.

Puede que el secreto tarde hasta una hora en propagarse. Para reducir este tiempo de inactividad, puede actualizar el contexto de ejecución de Lambda.

Actualizar el secreto de Twilio

> **Note**
>
> Twilio no admite dos secretos a la vez. Es posible que se produzca un tiempo de inactividad de los recursos mientras el Twilio secreto y el secreto de la AWS CloudFormation pila no estén sincronizados. Twiliolos mensajes que se envíen mientras los secretos no estén sincronizados fallarán debido a que las firmas son incorrectas. Espera a que los CloudFormation secretos Twilio y estén sincronizados e inténtalo de nuevo.

1. Crea un secreto de Twilio nuevo. Para obtener más información, consulte Tokens de autenticación y cómo cambiarlos en la documentación sobre Twilio.
2. Abre la AWS CloudFormation consola en https://console.aws.amazon.com/cloudformation.
3. En el panel de navegación, seleccione Pilas.
4. Seleccione la pila para el webhook que incluye el secreto que desea actualizar.
5. Seleccione Actualizar.
6. Asegúrese de que la opción Usar plantilla actual esté seleccionada y seleccione Siguiente.
7. En TwilioWebhookSecret, desactive Usar un valor existente, introduzca el nuevo Twilio secreto que creó en el paso 1 y seleccione Siguiente.
8. Seleccione Siguiente.
9. Seleccione Actualizar pila.

Puede que el secreto tarde hasta una hora en propagarse. Para reducir este tiempo de inactividad, puede actualizar el contexto de ejecución de Lambda.

Actualizar una función de Lambda

La función Lambda que crea la CloudFormation pila crea el webhook básico. Si desea personalizar la función Lambda para un caso de uso específico, como el registro personalizado, utilice la consola

para acceder a la función y, a continuación, utilice la CloudFormation consola Lambda para actualizar el código de la función Lambda.

Acceder a la función de Lambda

1. Abra la AWS CloudFormation consola en https://console.aws.amazon.com/cloudformation.
2. En el panel de navegación, seleccione Pilas.
3. Seleccione la pila del webhook que incluye la función de Lambda que desea actualizar.
4. Seleccione la pestaña Recursos.
5. Para abrir la función de Lambda en la consola de Lambda, en ID físico, seleccione el ID de la función de Lambda.

Ahora que ha accedido a la función de Lambda, utilice la consola de Lambda para actualizar el código de la función.

Actualizar el código de la función de Lambda

1. En Acciones, seleccione la función Exportar.
2. Seleccione Descargar paquete de implementación y guarde el archivo en su ordenador.
3. Descomprima el archivo .zip del paquete de implementación, actualice el archivo `app.py` y comprima el paquete de implementación actualizado, asegurándose de que se incluyen todos los archivos del archivo .zip original.
4. En la página de la consola de Lambda, seleccione la pestaña Código.
5. En Fuente de código, seleccione Subir desde.
6. Seleccione .zip file (Archivo .zip) y, a continuación, seleccione Subir.
    - En el selector de archivos, seleccione el archivo que actualizó, y seleccione Abrir y, luego, Guardar.
7. En Acciones, seleccione Publicar nueva versión.

Tipos de eventos disponibles

Los buses de eventos admiten actualmente los siguientes tipos de CloudFormation eventos:

- GitHub— Se admiten todos los tipos de eventos.
- Twilio: se admiten los webhooks posteriores al evento.

Cuotas, códigos de error y reintentos de entrega

Cuotas

El número de solicitudes entrantes al webhook está limitado por los AWS servicios subyacentes. En la siguiente tabla se muestran las cuotas correspondientes.

Servicio	Cuota
AWS Lambda	Predeterminada: 10 ejecuciones simultáneas Para obtener más información acerca de las cuotas, incluso cómo solicitar un aumento, consulte Cuotas de Lambda.
AWS Secrets Manager	Predeterminada: 5000 solicitudes por segundo Para obtener más información acerca de las cuotas, incluso cómo solicitar un aumento, consulte Cuotas de AWS Secrets Manager. **Note** El número de solicitudes por segundo se minimiza con el cliente de almacenamiento en cachéAWS Secrets Manager Python.
Amazon EventBridge	Tamaño máximo de entrada de 256 KB para PutEvents las acciones. EventBridge aplica cuotas tarifarias basadas en la región. Para obtener más información, consulte ???.

Códigos de error

Cada AWS servicio devuelve códigos de error específicos cuando se producen errores. En la siguiente tabla se muestran los códigos de error correspondientes.

Servicio	Código de error	Descripción
AWS Lambda	429 «TooManyRequestsException»	Se ha superado la cuota de ejecución simultánea.
AWS Secrets Manager	500 “Internal Server Error”	Se ha superado la cuota de solicitudes por segundo.
Amazon EventBridge	500 “Internal Server Error”	Se ha superado la cuota de tarifas de la región.

Reintento de entrega de eventos

Cuando se produzcan errores, puede reintentar la entrega de los eventos afectados. Cada proveedor de SaaS tiene diferentes procedimientos de reintento.

GitHub

Utilice la API de webhooks de GitHub para comprobar el estado de entrega de las llamadas de webhook y vuelva a enviar el evento, si es necesario. Para obtener más información, consulte la siguiente documentación sobre GitHub:

- Organización: reintente una entrega para un webhook de organización
- Repositorio: reintente una entrega para un webhook de repositorio
- Aplicación: reintente una entrega para un webhook de aplicación

Twilio

Los usuarios de Twilio pueden personalizar las opciones de reintento de eventos mediante anulaciones de conexión. Para obtener más información, consulte Webhooks (devoluciones de llamadas HTTP): anulaciones de conexión en la documentación sobre Twilio.

Recepción de eventos de Salesforce

Puedes usar Amazon EventBridge para recibir eventos de Salesforce las siguientes maneras:

- Al utilizar la función Salesforce's Event Bus Relay para recibir eventos directamente en un autobús de eventos EventBridge asociado.
- Configurando un flujo en Amazon AppFlow que se utiliza Salesforce como fuente de datos. AppFlow A continuación, Amazon envía Salesforce los eventos a EventBridge mediante un bus de eventos asociado.

Puede enviar la información de los eventos a Salesforce a través de los destinos de la API. Una vez enviado el evento a Salesforce, este puede ser procesado por Flujos o desencadenadores de Apex. Para obtener más información acerca de la configuración de un destino de la API de Salesforce, consulte .

Temas

- Recepción de eventos de Salesforce con Event Bus Relay
- Recibir eventos a partir del Salesforce uso de Amazon AppFlow

Recepción de eventos de Salesforce con Event Bus Relay

Paso 1: Configura Salesforce Event Bus Relay y una fuente de eventos EventBridge asociada

Al crear una configuración de retransmisión de eventos activadaSalesforce, Salesforce crea una fuente de eventos asociada EventBridge en estado pendiente.

Para configurar Salesforce Event Bus Relay

1. Configure una herramienta de API de REST
2. (Opcional) Defina un evento de plataforma
3. Cree un canal para un evento de plataforma personalizado
4. Cree un miembro del canal para asociar el evento de plataforma personalizado
5. Cree una credencial con nombre
6. Cree una configuración de retransmisión de eventos

Paso 2: Activa la fuente de eventos Salesforce asociada en la EventBridge consola e inicia la retransmisión de eventos

1. Abra la página de fuentes de eventos de los socios en la EventBridge consola.
2. Seleccione el origen de evento de socios de Salesforce que creó en el Paso 1.
3. Seleccione Asociar con bus de eventos.
4. Valide el nombre del bus de eventos de socios.
5. Seleccione Asociar.
6. Inicie la retransmisión del evento

Ahora que ha configurado e iniciado el Event Bus Relay y ha configurado la fuente de eventos asociada, puede crear una EventBridge regla que reaccione a los eventos para filtrar y enviar los datos a un destino.

Recibir eventos a partir del Salesforce uso de Amazon AppFlow

Amazon AppFlow encapsula los eventos Salesforce en un sobre de EventBridge eventos. El siguiente ejemplo muestra un Salesforce evento recibido por un bus de eventos EventBridge asociado.


```
{
    "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"
    }
}
```

Paso 1: Configurar Amazon AppFlow para que Salesforce lo utilice como fuente de eventos asociada

Para enviar eventos a EventBridge, primero debes configurar Amazon AppFlow para que Salesforce lo utilice como fuente de eventos asociada.

1. En la AppFlowconsola de Amazon, selecciona Create flow.
2. En la sección Detalles del flujo, en Nombre del flujo, indique un nombre para el flujo.
3. (Opcional) Indique una descripción para el flujo y, a continuación, seleccione Siguiente.
4. En Detalles del origen, seleccione Salesforce del menú desplegable Nombre del origen y, a continuación, seleccione Conectar para crear una conexión nueva.
5. En el cuadro de diálogo Conectar a Salesforce, seleccione Producción o Entorno aislado para el entorno de Salesforce.
6. En el campo Nombre de la conexión, introduzca un nombre exclusivo para la conexión y, a continuación, seleccione Continuar.
7. En el cuadro de diálogo de Salesforce, haga lo siguiente:
   a. Introduzca sus credenciales de inicio de sesión de Salesforce para iniciar sesión de Salesforce.
   b. Selecciona Salesforce eventos para los tipos de datos que Amazon va AppFlow a procesar.
8. En el menú desplegable Elegir Salesforce evento, selecciona el tipo de evento al que deseas realizar el envío EventBridge.
9. Para un destino, selecciona Amazon EventBridge.
10. Seleccione Crear un origen de evento de socios nuevo.
11. (Opcional) Especifique un sufijo único para el origen de evento de socios.

12. Seleccione Generar origen de evento de socios.
13. Seleccione un bucket de Amazon S3 para almacenar los archivos de carga de eventos que superen los 256 KB.
14. En la sección Desencadenador de flujo, asegúrese de que esté seleccionada la opción Ejecutar flujo en el evento. Esta configuración garantiza que el flujo se ejecute cuando se produzca un evento de Salesforce nuevo.
15. Seleccione Siguiente.
16. Para la asignación de campos, seleccione Asignar todos los campos directamente. Como alternativa, puede seleccionar los campos que sean de interés en la lista Nombre de campos de origen.

    Para obtener más información sobre la asignación de campos, consulte Asignación de campos de origen de datos.
17. Seleccione Siguiente.
18. (Opcional) Configura filtros para los campos de datos en Amazon AppFlow.
19. Seleccione Siguiente.
20. Revise la configuración y, a continuación, seleccione Crear.

Con el flujo configurado, Amazon AppFlow crea una nueva fuente de eventos de socios que, a continuación, tendrás que asociar a un bus de eventos de socios en tu cuenta.

Paso 2: EventBridge Configúralo para recibir Salesforce eventos

Asegúrese de que el AppFlow flujo de Amazon que se desencadena a partir de Salesforce eventos con EventBridge como destino esté configurado antes de seguir las instrucciones de esta sección.

Para configurarlo EventBridge para recibir Salesforce eventos

1. Abra la página de fuentes de eventos de los socios en la EventBridge consola.
2. Seleccione el origen de evento de socios de Salesforce que creó en el Paso 1.
3. Seleccione Asociar con bus de eventos.
4. Valide el nombre del bus de eventos de socios.
5. Seleccione Asociar.
6. En la AppFlow consola de Amazon, abre el flujo que has creado y selecciona Activar flujo.
7. Abre la página de reglas en la EventBridge consola.

8. Seleccione Crear regla.
9. Introduzca un nombre exclusivo para la regla.
10. Seleccione patrón de eventos en la sección Definir patrón.
11. En Patrón de coincidencia de eventos, seleccione Patrón predefinido de un servicio.
12. En la sección Proveedor de servicios, seleccione Todos los eventos.
13. En Seleccionar bus de eventos, seleccione Bus de eventos de socio o personalizado.
14. Selecciona el bus de eventos que has asociado a la fuente de eventos del AppFlow socio de Amazon.
15. En Seleccionar objetivos, elige el AWS servicio que actuará cuando se ejecute la regla. Una regla puede tener hasta cinco destinos.
16. Seleccione Crear.

El servicio de destino recibe todos los eventos de Salesforce configurados para la cuenta. Para filtrar los eventos o enviar algunos eventos a diferentes destinos, puede utilizar la opción Filtrar en función del contenido con patrones de eventos.

> **Note**
>
> En el caso de eventos de más de 256 KB, Amazon AppFlow no envía el evento completo a EventBridge. En su lugar, Amazon AppFlow coloca el evento en un bucket de S3 de tu cuenta y, a continuación, envía un evento al bucket de Amazon S3 EventBridge con un puntero. Puede usar el puntero para obtener el evento completo del bucket.

Depuración de la entrega de eventos

Los problemas relacionados con la entrega de eventos pueden ser difíciles de identificar, y EventBridge ofrece varias formas de depurar los errores en la entrega de eventos y recuperarse de ellos.

¿Cómo EventBridge vuelve a intentar entregar eventos

A veces, un evento no se entrega correctamente al destino especificado en una regla. Esto puede suceder, por ejemplo:

- Si el recurso de destino no está disponible

- Debido a las condiciones de la red

Cuando un evento no se entrega correctamente a un destino debido a errores recuperables, EventBridge vuelve a intentar enviar el evento. El tiempo durante el que se intenta y el número de reintentos se establecen en la configuración de la política de reintentos del destino. De forma predeterminada, EventBridge vuelve a intentar enviar el evento durante 24 horas y hasta 185 veces, con un retraso exponencial o un retardo aleatorio.

Si un evento no se entrega una vez agotados todos los reintentos, el evento se descarta y EventBridge no se sigue procesando.

Uso de colas con letra muerta para procesar los eventos no entregados

Para evitar perder eventos no entregados a un destino, puede configurar una cola de mensajes fallidos (DLQ) y enviarle todos los eventos fallidos para que los procese más adelante.

EventBridge Los DLQ son colas estándar de Amazon SQS EventBridge que se utilizan para almacenar eventos que no se pudieron entregar correctamente a un objetivo. Al crear una regla y añadir un destino, puede elegir si desea utilizar o no una DLQ. Al configurar una DLQ, puede retener los eventos que no se hayan entregado correctamente. A continuación, puede resolver el problema que provocó el error en la entrega del evento y procesar los eventos más tarde.

Cuando configura un DLQ para el objetivo de una regla, EventBridge envía los eventos con invocaciones fallidas a la cola de Amazon SQS seleccionada.

Los errores de eventos se gestionan de distintas formas. Algunos eventos se descartan o se envían a una DLQ sin volver a intentar la entrega. Por ejemplo, en el caso de los errores derivados de la falta de permisos para acceder a un destino o de un recurso de destino que ya no existe, todos los reintentos fallan hasta que se adopte una medida para resolver el problema subyacente. En lugar de volver a intentarlo, EventBridge envía estos eventos directamente al DLQ, si lo tiene.

Cuando se produce un error en la entrega de un evento, EventBridge publica un evento en CloudWatch las métricas de Amazon que indican que se ha producido un `invocation` error en un objetivo. Si utilizas un DLQ, se envían métricas adicionales a CloudWatch Inclusive `InvocationsSentToDLQ` y`InvocationsFailedToBeSentToDLQ`.

También puede especificar los DLQ para los buses de eventos, si los utiliza AWS KMS claves administradas por el cliente para cifrar los eventos en reposo. Para obtener más información, consulte ???.

Cada mensaje de la DLQ incluirá los siguientes atributos personalizados:

- RULE_ARN
- TARGET_ARN
- ERROR_CODE

  El siguiente es un ejemplo de los códigos de error que puede devolver una DLQ:

  - 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

Se puede devolver las siguientes condiciones:

- `MaximumRetryAttempts`
- `MaximumEventAgeInSeconds`

- `RETRY_ATTEMPTS`

En el siguiente vídeo se repasa la configuración de las DLQ: Uso de colas de mensajes fallidos (DLQ)

Temas

- Consideraciones sobre el uso de una cola de mensajes fallidos
- Concesión de permisos a la cola de mensajes fallidos
- Cómo reenviar eventos desde una cola de mensajes fallidos

Consideraciones sobre el uso de una cola de mensajes fallidos

Tenga en cuenta lo siguiente al configurar un DLQ para. EventBridge

- Solo se admiten colas estándar. No puede utilizar una cola FIFO para una entrada DLQ. EventBridge
- EventBridge incluye los metadatos del evento y los atributos del mensaje en el mensaje, incluidos: el código de error, el mensaje de error, la condición de reintento agotada, el ARN de la regla, los intentos de reintento y el ARN de destino. Puede usar estos valores para identificar un evento y la causa del error.
- Permisos para DLQ de la misma cuenta:
  - Si añade un objetivo a una regla mediante la consola y elige una cola de Amazon SQS en la misma cuenta, se adjuntará a la cola una política basada en recursos que le concede EventBridge acceso a la cola.
  - Si utiliza la `PutTargets` operación de la EventBridge API para añadir o actualizar un destino para una regla y elige una cola de Amazon SQS en la misma cuenta, debe conceder permisos manualmente a la cola seleccionada. Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.
- Permisos para usar colas de Amazon SQS desde una cuenta diferente. AWS

  - Si crea una regla desde la consola, las colas de otras cuentas no se muestran para que las seleccione. Debe proporcionar el ARN de la cola de la otra cuenta y, a continuación, adjuntar manualmente una política basada en recursos para conceder permisos a la cola. Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.
  - Si crea una regla mediante la API, debe adjuntar manualmente una política basada en recursos a las colas de SQS de otra cuenta que se utilice como cola de mensajes fallidos. Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.
- La cola de Amazon SQS que utilice debe estar en la región en la que creó la regla.

Concesión de permisos a la cola de mensajes fallidos

Para entregar correctamente los eventos a la cola, EventBridge debe tener permiso para hacerlo. Al especificar un DLQ mediante la EventBridge consola, los permisos se añaden automáticamente. Esto incluye:

- Al configurar un DLQ para el objetivo de una regla.
- Cuando configura un DLQ para un bus de eventos en el que ha especificado que EventBridge utilice un DLQ AWS KMS clave administrada por el cliente para cifrar los eventos en reposo.

  Para obtener más información, consulte ???.

Si especificas un DLQ mediante la API o utilizas una cola que se encuentra en una AWS cuenta diferente, debes crear manualmente una política basada en los recursos que conceda los permisos necesarios y, a continuación, adjuntarla a la cola.

Ejemplo de permisos de cola con letra muerta de Target

La siguiente política basada en recursos demuestra cómo conceder los permisos necesarios para enviar mensajes de eventos EventBridge a una cola de Amazon SQS. El ejemplo de política otorga al EventBridge servicio permisos para usar la `SendMessage` operación para enviar mensajes a una cola denominada «DLQ». MyEvent La cola debe estar en la región us-west-2 en la cuenta 123456789012. AWS El `Condition` estado de cuenta solo permite solicitudes que provengan de una regla denominada «MyTestRule» que se haya creado en la región us-west-2 en la cuenta 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"
    }
  }
}
```

Ejemplo de permisos de cola con letra muerta para el bus de eventos

La siguiente política basada en recursos demuestra cómo conceder los permisos necesarios al especificar un DLQ para un bus de eventos. En este caso, `aws:SourceArn` especifica el ARN del bus de eventos que envía los eventos al DLQ. También en este ejemplo, la cola debe estar en la misma región que el bus de eventos.

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

Para adjuntar la política a la cola, utilice la consola de Amazon SQS, abra la cola y, a continuación, seleccione la política de acceso y edítela. También puede utilizar la AWS CLI. Para obtener más información, consulte Permisos de Amazon SQS.

Cómo reenviar eventos desde una cola de mensajes fallidos

Los mensajes se pueden sacar de una DLQ de dos formas:

- Evitar escribir lógica de consumo de Amazon SQS: establezca la DLQ como una fuente de eventos en la función de Lambda para drenar dicha cola.
- Escriba la lógica de consumo de Amazon SQS: utilice la API AWS o el SDK de Amazon SQS AWS CLI o escriba una lógica de consumo personalizada para sondear, procesar y eliminar los mensajes del DLQ.

Patrones de EventBridge eventos de Amazon

Los patrones de eventos tienen la misma estructura que los eventos con los que coinciden. Las reglas utilizan patrones de eventos para seleccionar eventos y enviarlos a los destinos. Un patrón de eventos coincide con un evento o no lo hace.

> **Important**
>
> En EventBridge, es posible crear reglas que pueden provocar higher-than-expected cargos y estrangulamientos. Por ejemplo, puede crear inadvertidamente una regla que conduzca a un bucle infinito, en el que una regla se active de forma recursiva sin fin. Imagine que creó una regla puede detectar que las ACL han cambiado en un bucket de Amazon S3 y activar software para cambiarlas al estado deseado. Si la regla no se ha escrito minuciosamente, un nuevo cambio de las ACL vuelve a activar la regla, lo que crea un bucle infinito.
> Para obtener orientación sobre cómo escribir reglas y patrones de eventos precisos para minimizar estos resultados inesperados, consulte ??? y ???.

En el siguiente vídeo se repasan los aspectos básicos de los patrones de eventos: Cómo filtrar eventos

Temas

- Creación de patrones de eventos
- Ejemplos de eventos y patrones de eventos
- Hacer coincidir valores nulos y cadenas vacías en los patrones de EventBridge eventos de Amazon
- Matrices en los patrones de EventBridge eventos de Amazon
- Filtrado de contenido en los patrones de EventBridge eventos de Amazon
- Probar un patrón de eventos con el EventBridge Sandbox
- Mejores prácticas a la hora de definir los patrones de EventBridge eventos de Amazon

El siguiente evento muestra un AWS evento sencillo de Amazon EC2.

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

El siguiente patrón de eventos procesa todos los eventos `instance-termination` de Amazon EC2.

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

Creación de patrones de eventos

Para crear un patrón de eventos, especifique los campos con los que desea que coincida el patrón del evento. Especifique únicamente los campos que utilice para la coincidencia. El ejemplo de patrón de eventos anterior solo proporciona valores para tres campos: los campos de nivel superior `"source"` y`"detail-type"`, y el `"state"` campo dentro del campo `"detail"` objeto. EventBridge ignora todos los demás campos del evento al aplicar la regla.

Para que un patrón de eventos coincida con un evento, el evento debe contener todos los nombres de campos enumerados en el patrón. Los nombres de los campos también deben aparecer en el evento con la misma estructura de anidación.

Al escribir patrones de evento para buscar eventos, puede utilizar la API de `TestEventPattern` o el comando de la CLI `test-event-pattern` para probar si el patrón coincide con los eventos correctos. Para obtener más información, consulte TestEventPattern.

Coincidencias de valores de eventos

En un patrón de eventos, el valor que debe coincidir está en una matriz JSON, entre corchetes ("[", "]") para que pueda proporcionar varios valores. Por ejemplo, para hacer coincidir los eventos de Amazon EC2 o AWS Fargate, puede utilizar el siguiente patrón, que coincide con los eventos en los que el valor del `"source"` campo es o`"aws.ec2"`. `"aws.fargate"`

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

Consideraciones sobre la creación de patrones de eventos

A continuación, se muestran algunos aspectos que debe considerar a la hora de crear sus patrones de eventos:

- EventBridge ignora los campos del evento que no están incluidos en el patrón de eventos. El efecto es que hay un comodín `"*": "*"` para los campos que no aparecen en el patrón de eventos.
- Los valores que coinciden con patrones de eventos siguen las reglas JSON. Puede incluir cadenas entre comillas (“), números y palabras clave `true`, `false` y `null`.
- En el caso de las cadenas, EventBridge utiliza la character-by-character coincidencia exacta sin doblar mayúsculas y minúsculas ni normalizar ninguna otra cadena.
- En el caso de los números, EventBridge utiliza la representación de cadenas. Por ejemplo, 300, 300.0 y 3.0e2 no se consideran iguales.
- Si se especifican varios patrones para el mismo campo JSON, EventBridge solo usa el último.
- Tenga en cuenta que cuando EventBridge compila patrones de eventos para su uso, utiliza un punto (.) como carácter de unión.

  Esto significa que EventBridge tratará los siguientes patrones de eventos como idénticos:

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

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

Y que ambos patrones de eventos coincidirán con los dos eventos siguientes:

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

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

> **Note**
>
> Esto describe EventBridge el comportamiento actual y no se debe confiar en él para que no cambie.

- Los patrones de eventos que contienen campos duplicados no son válidos. Si un patrón contiene campos duplicados, EventBridge solo tiene en cuenta el valor final del campo.

  Por ejemplo, los siguientes patrones de eventos coincidirán con el mismo 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"] }
}
```

  Y EventBridge trata los dos eventos siguientes como idénticos:

```
## 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**
>
> Esto describe EventBridge el comportamiento actual y no se debe confiar en él para que no cambie.

Operaciones de comparación para su uso en patrones de eventos

A continuación se muestra un resumen de todos los operadores de comparación disponibles en EventBridge.

Los operadores de comparación solo funcionan en los nodos secundarios, con la excepción de `$or` y `anything-but`.

Comparación	Ejemplo	Sintaxis de reglas
Y	El valor de Location (Ubicación) es “New York” y el de Day (Día) es “Monday”	`"Location": [ "New York" ], "Day": ["Monday"]`
Cualquier cosa, pero	El estado es cualquier valor además de «inicializar».	`"state": [ { "anything-but": "initializing" } ]`
Cualquier cosa, pero (comienza con)	La región no está en los EE. UU.	`"Region": [ { "anything-but": {"prefix": "us-" } } ]`
Cualquier cosa, pero (termina con)	FileName no termina con la extensión.png.	`"FileName": [ { "anything-but": { "suffix": ".png" } } ]`
Cualquier cosa menos (ignore mayúsculas y minúsculas)	El estado es cualquier valor que no sea «inicializar» o cualquier otra variación entre mayúsculas y minúsculas, como «INICIALIZAR».	`"state": : [{ "anything-but": { "equals-ignore-case": "initializing" }}]}`
Cualquier cosa, excepto usar un comodín	FileName no es una ruta de archivo que incluya. `/lib/`	`"FilePath" : [{ "anything-but": { "wildcard": "*/lib/*" }}]`
Comienza por	La región se encuentra en EE. UU.	`"Region": [ {"prefix": "us-" } ]`
Empieza por (ignora mayúsculas y minúsculas)	El nombre del servicio comienza con las letras «eventb», independientemente de las mayúsculas y minúsculas.	`{"service" : [{ "prefix": { "equals-ignore-case": "eventb" }}]}`
Vacío	LastName está vacío.	`"LastName": [""]`
Igual a	El valor de Name (Nombre) es “Alice”	`"Name": [ "Alice" ]`

Comparación	Ejemplo	Sintaxis de reglas
Igual a (omitir mayúsculas y minúsculas)	El valor de Name (Nombre) es “Alice”	`"Name": [ { "equals-ignore-case": "alice" } ]`
Acaba con	FileName termina con una extensión .png	`"FileName": [ { "suffix": ".png" } ]`
Termina con (ignorar mayúsculas y minúsculas)	El nombre del servicio termina con las letras «tbridge» o cualquier otra variante de mayúsculas y minúsculas, como «TBRIDGE».	`{"service" : [{ "suffix": { "equals-ignore-case": "tBridge" }}]}`
Existe	ProductName existe	`"ProductName": [ { "exists": true } ]`
No existe	ProductName no existe	`"ProductName": [ { "exists": false } ]`
No	El valor de Weather (Tiempo) es cualquier valor menos “Raining” (Lluvia)	`"Weather": [ { "anything-but": [ "Raining" ] } ]`
Nulo	El valor de UserID (ID de usuario) es nulo	`"UserID": [ null ]`
Valor numérico (igual a)	El valor de Price (Precio) es 100	`"Price": [ { "numeric": [ "=", 100 ] } ]`
Valor numérico (rango)	El valor de Price (Precio) es superior a 10 e inferior o igual a 20	`"Price": [ { "numeric": [ ">", 10, "<=", 20 ] } ]`
Or (Disyunción)	PaymentType es «crédito» o «débito»	`"PaymentType": [ "Credit", "Debit"]`
O (varios campos)	El valor de Location (Ubicación) es “New York” o el de Day (Día) es “Monday”.	`"$or": [ { "Location": [ "New York" ] }, { "Day": [ "Monday" ] } ]`

Comparación	Ejemplo	Sintaxis de reglas
Comodín	Cualquier archivo con la extensión .png, ubicado dentro de la carpeta "dir"	`"FileName": [ { "wildcard": "dir/*.png" } ]`

Ejemplos de eventos y patrones de eventos

Puede usar todos los tipos de datos y valores JSON para hacer coincidir los eventos. A continuación, se muestran los eventos y los patrones de eventos que se corresponden con ellos.

Coincidencia de campo

Puede hacer coincidir el valor de un campo. Considere el siguiente evento de 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
  }
}
```

Para el evento anterior, puede utilizar el campo `"responseElements"` para que coincida.

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

Coincidencia de valor

Considere el siguiente evento de Amazon Macie, que está truncado.

```
{
  "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,
.
.
.
```

El siguiente patrón de eventos coincide con cualquier evento que tenga una puntuación de gravedad de 1 y un recuento de 2.

```
{
  "source": ["aws.macie"],
  "detail-type": ["Macie Finding"],
```

```
  "detail": {
    "severity": {
      "score": [1]
    },
    "count":[2]
  }
}
```

Hacer coincidir valores nulos y cadenas vacías en los patrones de EventBridge eventos de Amazon

> **Important**
>
> En EventBridge, es posible crear reglas que pueden provocar higher-than-expected cargos y estrangulamientos. Por ejemplo, puede crear inadvertidamente una regla que conduzca a un bucle infinito, en el que una regla se active de forma recursiva sin fin. Imagine que creó una regla puede detectar que las ACL han cambiado en un bucket de Amazon S3 y activar software para cambiarlas al estado deseado. Si la regla no se ha escrito minuciosamente, un nuevo cambio de las ACL vuelve a activar la regla, lo que crea un bucle infinito.
> Para obtener orientación sobre cómo escribir reglas y patrones de eventos precisos para minimizar estos resultados inesperados, consulte ??? y ???.

Puede crear un patrón de eventos que coincida con un campo de un evento que tenga un valor nulo o una cadena vacía. Considere el siguiente ejemplo de evento.

Consulte las mejores prácticas para evitar cargos y limitaciones superiores a lo esperado

```
{
  "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
   }
}
```

Para hacer coincidir eventos donde el valor de `eventVersion` es una cadena vacía, utilice el siguiente patrón de eventos, que coincide con el evento precedente.

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

Para hacer coincidir eventos donde el valor de `responseElements` es una cadena vacía, utilice el siguiente patrón de eventos, que coincide con el evento precedente.

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

> **Note**
>
> Los valores nulos y las cadenas vacías no son intercambiables en coincidencia de patrones. Un patrón de eventos que coincide con cadenas vacías no coincide con los valores de `null`.

Matrices en los patrones de EventBridge eventos de Amazon

El valor de cada campo de un patrón de eventos es una matriz que contiene uno o más valores. Un patrón de eventos coincide con el evento si alguno de los valores de la matriz coincide con el valor del evento. Si el valor en el evento es una matriz, entonces el patrón de eventos coincide si la intersección de la matriz de patrones de eventos y la matriz de eventos no está vacía.

> **Important**
>
> En EventBridge, es posible crear reglas que pueden provocar higher-than-expected cargos y estrangulamientos. Por ejemplo, puede crear inadvertidamente una regla que conduzca a un bucle infinito, en el que una regla se active de forma recursiva sin fin. Imagine que creó una regla puede detectar que las ACL han cambiado en un bucket de Amazon S3 y activar software para cambiarlas al estado deseado. Si la regla no se ha escrito minuciosamente, un nuevo cambio de las ACL vuelve a activar la regla, lo que crea un bucle infinito.
> Para obtener orientación sobre cómo escribir reglas y patrones de eventos precisos para minimizar estos resultados inesperados, consulte ??? y ???.

Por ejemplo, considere un patrón de eventos que incluya el siguiente campo.

```
"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",
]
```

El patrón de eventos precedente coincide con un evento que incluye el siguiente texto, ya que el primer elemento de la matriz de patrones coincide con el segundo elemento de la matriz de eventos.

```
"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"
]
```

Filtrado de contenido en los patrones de EventBridge eventos de Amazon

Amazon EventBridge admite el filtrado declarativo de contenido mediante patrones de eventos. Mediante el filtrado de contenido, puede crear patrones de eventos complejos que solo coinciden con eventos bajo condiciones muy específicas. Por ejemplo, puede crear un patrón de eventos que coincida con un evento cuando:

- Un campo del evento está dentro de un intervalo numérico específico.
- El evento proviene de una dirección IP específica.
- No existe un campo específico en el JSON del evento.

> **Important**
>
> En EventBridge, es posible crear reglas que pueden provocar higher-than-expected cargos y restricciones. Por ejemplo, puede crear inadvertidamente una regla que conduzca a un bucle infinito, en el que una regla se active de forma recursiva sin fin. Imagine que creó una regla puede detectar que las ACL han cambiado en un bucket de Amazon S3 y activar software para cambiarlas al estado deseado. Si la regla no se ha escrito minuciosamente, un nuevo cambio de las ACL vuelve a activar la regla, lo que crea un bucle infinito.
> Para obtener orientación sobre cómo escribir reglas y patrones de eventos precisos para minimizar estos resultados inesperados, consulte ??? y ???.

Tipos de filtros

- Coincidencia de prefijos
- Coincidencia de sufijos
- Coincidencia "anything-but"
- Coincidencia numérica
- Coincidencia de direcciones IP
- Coincidencia exists
- quals-ignore-caseCoincidencia E
- Coincidencia con comodines

- Ejemplo complejo con coincidencia múltiple
- Ejemplo complejo con coincidencia $or

Coincidencia de prefijos

Puede hacer coincidir un evento dependiendo del prefijo de un valor en el origen del evento. Puede utilizar la coincidencia de prefijos para los valores de cadena.

Por ejemplo, el siguiente patrón de eventos coincidiría con cualquier evento en el que el campo `"time"` comenzara con `"2017-10-02"` como `"time": "2017-10-02T18:43:48Z"`.

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

El prefijo coincide ignorando mayúsculas y minúsculas

También puede hacer coincidir el valor de un prefijo independientemente de las mayúsculas y minúsculas de los caracteres por los que comience un valor, utilizándolo `equals-ignore-case` junto con `prefix`.

Por ejemplo, el siguiente patrón de eventos coincidiría con cualquier evento en el que el `service` campo comience con la cadena de caracteres`EventB`, pero también `EVENTB` coincidiría con cualquier otro tipo de mayúscula de esos caracteres. `eventb`

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

Coincidencia de sufijos

Puede hacer coincidir un evento dependiendo del sufijo de un valor en el origen del evento. Puede utilizar la coincidencia de sufijos para los valores de cadena.

Por ejemplo, el siguiente patrón de eventos coincidiría con cualquier evento en el que el campo `"FileName"` terminara con la extensión de archivo `.png`.

```
{
```

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

El sufijo coincide sin incluir mayúsculas y minúsculas

También puede hacer coincidir el valor de un sufijo independientemente de las mayúsculas y minúsculas de los caracteres con los que termine un valor, utilizándolo `equals-ignore-case` junto con `suffix`.

Por ejemplo, el siguiente patrón de eventos coincidiría con cualquier evento en el que el `FileName` campo termine con la cadena de caracteres`.png`, pero también con cualquier `.PNG` otro tipo de mayúsculas de esos caracteres.

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

Coincidencia "anything-but"

Cualquier cosa, excepto la coincidencia, coincide con cualquier cosa excepto con lo que se especifica en la regla.

Puede utilizar la coincidencia anything-but con cadenas y valores numéricos, incluidas listas que contienen solo cadenas o solo números.

El siguiente patrón de eventos muestra la coincidencia anything-but con cadenas y números.

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

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

El siguiente patrón de eventos muestra la coincidencia anything-but con una lista de cadenas.

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

El siguiente patrón de eventos muestra la coincidencia anything-but con una lista de números.

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

Cualquier cosa menos coincidir ignorando mayúsculas y minúsculas

También se puede utilizar `equals-ignore-case` junto con`anything-but`, para hacer coincidir los valores de las cadenas independientemente de las mayúsculas y minúsculas de los caracteres.

El siguiente patrón de eventos coincide con `state` los campos que no contienen las cadenas «inicialización», «inicialización», «inicialización» o cualquier otro uso de mayúsculas para esos caracteres.

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

También puedes usarlo `equals-ignore-case` junto con `anything-but` para compararlo con una lista de valores:

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

Cualquier cosa, excepto los prefijos

Puede usarlo `prefix` junto con `anything-but` para hacer coincidir valores de cadena que no comiencen por el valor especificado. Esto incluye valores individuales o una lista de valores.

El siguiente patrón de eventos muestra cualquier evento que no tenga el prefijo `"init"` en el campo, excepto las coincidencias. `"state"`

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

El siguiente patrón de eventos muestra cualquier cosa, excepto las coincidencias, y se utiliza con una lista de valores de prefijo. Este patrón de eventos coincide con cualquier evento que no tenga el prefijo ni esté en el campo`"init"`. `"stop"` `"state"`

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

Cualquier cosa, excepto los sufijos

Puede usarlo `suffix` junto con `anything-but` para hacer coincidir valores de cadena que no terminen con el valor especificado. Esto incluye valores individuales o una lista de valores.

El siguiente patrón de eventos coincide con cualquier valor del `FileName` campo que no termine en`.txt`.

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

El siguiente patrón de eventos muestra cualquier cosa, excepto las coincidencias, y se utiliza con una lista de valores de sufijos. Este patrón de eventos coincide con todos los valores del `FileName` campo que no terminen en uno u otro. `.txt .rtf`

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

```
  }
}
```

Cualquier cosa, excepto la coincidencia mediante caracteres comodín

Puede utilizar el carácter comodín (*) dentro de los valores que especifique para cualquier cosa que no sea coincidente. Esto incluye valores individuales o una lista de valores.

El siguiente patrón de eventos coincide con cualquier valor del `FileName` campo que no contenga`/lib/`.

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

El siguiente patrón de eventos muestra cualquier cosa, excepto las coincidencias, y se utiliza con una lista de valores que incluye caracteres comodín. Este patrón de eventos coincide con cualquier valor del `FileName` campo que no contenga ninguno de los dos. `/lib/` `/bin/`

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

Para obtener más información, consulte ???.

Coincidencia numérica

La coincidencia numérica funciona con valores que son números JSON. Está limitada a valores entre -5.0e9 y +5.0e9 inclusive, con 15 dígitos de precisión o seis dígitos a la derecha del punto decimal.

A continuación, se muestra la coincidencia numérica de un patrón de eventos que solo coincide con los eventos que son verdaderos en todos los campos.

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

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

Coincidencia de direcciones IP

Puede utilizar la coincidencia de direcciones IP para direcciones IPv4 e IPv6. El siguiente patrón de eventos muestra las direcciones IP que coinciden con las direcciones IP que comienzan con 10.0.0 y terminan con un número entre 0 y 255.

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

Coincidencia exists

La coincidencia exists funciona en presencia o ausencia de un campo en la JSON del evento.

La coincidencia exists solo funciona en nodos secundarios. No funciona en nodos intermedios.

El siguiente patrón de eventos coincide con cualquier evento que tenga un campo `detail.state`.

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

El patrón de eventos precedente coincide con el siguiente evento.

```
{
  "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"
  }
}
```

El patrón de eventos anterior NO coincide con el evento siguiente porque no tiene un campo `detail.state`.

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

quals-ignore-caseCoincidencia E

La quals-ignore-case coincidencia E funciona con valores de cadena independientemente de las mayúsculas y minúsculas.

El siguiente patrón de eventos coincide con cualquier evento que tenga un campo `detail-type` que coincida con la cadena especificada, independientemente de las mayúsculas y minúsculas.

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

El patrón de eventos precedente coincide con el siguiente evento.

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

```
}
```

Coincidencia con comodines

Puede utilizar el carácter comodín (*) para hacer coincidir los valores de cadena en patrones de eventos.

> **Note**
>
> Actualmente, el carácter comodín solo se admite en las reglas del bus de eventos.

Consideraciones a la hora de utilizar caracteres comodín en los patrones de eventos:

- Puede especificar cualquier número de caracteres comodín en un valor de cadena determinado; sin embargo, no se admiten caracteres comodín consecutivos.
- EventBridge admite el uso del carácter de barra invertida (\) para especificar los caracteres literales * y\ en los filtros comodín:
  - La cadena * representa el carácter * literal
  - La cadena \\ representa el carácter \ literal

  No se admite el uso de la barra invertida para escapar de otros caracteres.

Complejidad de caracteres comodín y patrones de eventos

Existe un límite en cuanto a la complejidad de una regla que utilice caracteres comodín. Si una regla es demasiado compleja, EventBridge devuelve un `InvalidEventPatternException` al intentar crearla. Si la regla genera un error de este tipo, considere la posibilidad de utilizar las siguientes instrucciones para reducir la complejidad del patrón de eventos:

- Reduzca el número de caracteres comodín utilizados

  Utilice caracteres comodín solo cuando realmente necesite hacer coincidir varios valores posibles. Por ejemplo, considere el siguiente patrón de eventos, en el que desea hacer coincidir los buses de evento de la misma región:

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

```
}
```

En el caso anterior, muchas de las secciones de la ARN se basarán directamente en la región en la que residan los buses de evento. Por lo tanto, si utiliza la región `us-east-1`, el siguiente ejemplo podría ser un patrón menos complejo que aún coincida con los valores deseados:

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

- Reduzca las secuencias de caracteres que se repiten después de un carácter comodín

  Si la misma secuencia de caracteres aparece varias veces después de usar un comodín, aumenta la complejidad del procesamiento del patrón de eventos. Rediseñe el patrón de eventos para minimizar las secuencias repetidas. Por ejemplo, considere el siguiente ejemplo, que coincide con el nombre de archivo `doc.txt` de cualquier usuario:

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

  Si supiera que el archivo `doc.txt` solo aparecería en la ruta especificada, podría reducir la secuencia de caracteres repetidos de la siguiente manera:

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

Ejemplo complejo con coincidencia múltiple

Puede combinar varias reglas de coincidencia en un patrón de eventos más complejo. Por ejemplo, la siguiente regla combina los patrones de eventos `anything-but` y `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**
>
> Al crear patrones de eventos, si incluye una clave más de una vez, la última referencia será la que se utilice para evaluar los eventos. Por ejemplo, para el siguiente patrón:
>
> ```
> {
>   "detail": {
>     "location": [ { "prefix": "us-" } ],
>     "location": [ { "anything-but": "us-east" } ]
>   }
> }
> ```
>
> solo se tendrá en cuenta `{ "anything-but": "us-east" }` a la hora de evaluar la `location`.

Ejemplo complejo con coincidencia **`$or`**

También puede crear patrones de eventos complejos que comprueben si alguno de los valores de los campos coinciden en varios campos. Utilice `$or` para crear un patrón de eventos que coincida si alguno de los valores de varios campos coincide.

Tenga en cuenta que puede incluir otros tipos de filtros, como coincidencia numérica y matrices, en la coincidencia de patrones para los campos individuales del constructo `$or`.

El siguiente patrón de eventos coincide si se cumple alguna de las siguientes condiciones:

- El campo `c-count` es mayor que 0 o menor o igual que 5.
- El campo `d-count` tiene menos de 10.
- El campo `x-limit` es igual a 3.018e2.

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

```
      { "d-count": [ { "numeric": [ "<", 10 ] } ] },
      { "x-limit": [ { "numeric": [ "=", 3.018e2 ] } ] }
    ]
  }
}
```

> **Note**
>
> Las API que aceptan un patrón de eventos (comoPutRule, CreateArchive, UpdateArchive y TestEventPattern) arrojarán un valor InvalidEventPatternException si el uso de $or da lugar a más de 1000 combinaciones de reglas.
> Para determinar el número de combinaciones de reglas en un patrón de eventos, multiplique el número total de argumentos de cada matriz $or del patrón de eventos. Por ejemplo, el patrón anterior contiene una sola matriz $or con tres argumentos, por lo que el número total de combinaciones de reglas también es tres. Si agrega otra matriz $or con dos argumentos, el total de combinaciones de reglas sería seis.

Probar un patrón de eventos con el EventBridge Sandbox

Las reglas utilizan patrones de eventos para seleccionar eventos y enviarlos a los destinos. Los patrones de eventos tienen la misma estructura que los eventos con los que coinciden. Un patrón de eventos coincide con un evento o no lo hace.

La definición de un patrón de eventos suele formar parte del proceso más amplio de crear una regla nueva o editar una existente. Sin embargo EventBridge, al usar el Sandbox en, puede definir rápidamente un patrón de eventos y usar un evento de muestra para confirmar que el patrón coincide con los eventos deseados, sin tener que crear o editar una regla. Una vez que hayas probado tu patrón de eventos, tendrás EventBridge la opción de crear una nueva regla utilizando ese patrón de eventos directamente desde el entorno de pruebas.

Para obtener más información acerca de patrones de eventos, consulte ???.

> **Important**
>
> En EventBridge, es posible crear reglas que pueden provocar higher-than-expected cargos y estrangulamientos. Por ejemplo, puede crear inadvertidamente una regla que conduzca a un bucle infinito, en el que una regla se active de forma recursiva sin fin. Imagine que creó

una regla puede detectar que las ACL han cambiado en un bucket de Amazon S3 y activar software para cambiarlas al estado deseado. Si la regla no se ha escrito minuciosamente, un nuevo cambio de las ACL vuelve a activar la regla, lo que crea un bucle infinito.
Para obtener orientación sobre cómo escribir reglas y patrones de eventos precisos para minimizar estos resultados inesperados, consulte ??? y ???.

Para probar un patrón de eventos mediante el sandbox EventBridge

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Recursos para desarrolladores, a continuación, seleccione Entorno aislado y, en la página Entorno aislado, seleccione la pestaña Patrón de eventos.
3. En Fuente del evento, selecciona AWS eventos o eventos EventBridge asociados.
4. En la sección Eventos de muestra, seleccione un tipo de evento de muestra con el que quiera probar el patrón de eventos.

   Están disponibles los siguientes eventos de muestra:

   - AWS eventos: seleccione entre los eventos emitidos desde los compatibles Servicios de AWS.
   - EventBridge eventos de socios: selecciona entre los eventos emitidos por servicios de terceros que ofrecen soporte EventBridge, como Salesforce.
   - Escribir el mío: introduzca su propio evento en texto JSON.

     También puede utilizar un evento AWS o un evento asociado como punto de partida para crear su propio evento personalizado.

     1. Selecciona AWS eventos o eventos EventBridge asociados.
     2. Use el menú desplegable Eventos de muestra para seleccionar el evento que desea usar como punto de partida para su evento personalizado.

        EventBridge muestra el evento de muestra.
     3. Seleccione Copiar.
     4. Seleccione Escribir el mío para el tipo de evento.
     5. Elimine la estructura de eventos de ejemplo en el panel de edición de JSON y pegue el evento AWS o el asociado en su lugar.
     6. Edite la JSON del evento para crear su propio evento de muestra.

5. Seleccione un método de creación. Puede crear un patrón de eventos a partir de un EventBridge esquema o una plantilla, o puede crear un patrón de eventos personalizado.

   Existing schema

   Para usar un EventBridge esquema existente para crear el patrón de eventos, haga lo siguiente:

   1. En la sección Método de creación, en Método, seleccione Usar esquema.
   2. En la sección Patrón de eventos, para Tipo de esquema, seleccione Seleccione el esquema del registro de esquemas.
   3. Para el Registro de esquemas, seleccione el cuadro desplegable e indique el nombre de un registro de esquemas, por ejemplo, `aws.events`. También puede seleccionar una opción de la lista desplegable que aparece.
   4. En Esquema, seleccione el cuadro desplegable e indique el nombre del esquema que se va a usar. Por ejemplo, `aws.s3@ObjectDeleted`. También puede seleccionar una opción de la lista desplegable que aparece.
   5. En la sección Modelos, seleccione el botón Editar situado junto a cualquier atributo para abrir sus propiedades. Configure los campos Relación y Valor según sea necesario y, a continuación, seleccione Configurar para guardar el atributo.

      > **Note**
      >
      > Para obtener información sobre la definición de un atributo, selecciona el icono de información situado junto al nombre del atributo. Para obtener información sobre cómo configurar las propiedades de los atributos en su evento, abra la sección Nota del cuadro de diálogo de propiedades de los atributos.
      > Para eliminar las propiedades de un atributo, seleccione el botón Editar de ese atributo y, a continuación, seleccione Borrar.

   6. Seleccione Generar patrón de eventos en JSON para generar y validar el patrón de eventos como texto JSON.
   7. Para probar el evento de muestra con el patrón de prueba, seleccione Patrón de prueba.

      EventBridge muestra un cuadro de mensaje que indica si el evento de muestra coincide con el patrón de eventos.

      Puede elegir una de las siguientes opciones:

- Copiar: copia el patrón de eventos en el portapapeles de su dispositivo.
- Prettify: facilita la lectura del texto JSON al añadir saltos de línea, tabulaciones y espacios.

Custom schema

Para escribir un esquema personalizado y convertirlo en un patrón de eventos, haga lo siguiente:

1. En la sección Método de creación, en Método, seleccione Usar esquema.
2. En la sección Patrón de eventos, en Tipo de esquema, seleccione Introducir esquema.
3. Introduzca el esquema en el cuadro de texto. Debe formatear el esquema como texto JSON válido.
4. En la sección Modelos, seleccione el botón Editar situado junto a cualquier atributo para abrir sus propiedades. Configure los campos Relación y Valor según sea necesario y, a continuación, seleccione Configurar para guardar el atributo.

   > **Note**
   >
   > Para obtener información sobre la definición de un atributo, selecciona el icono de información situado junto al nombre del atributo. Para obtener información sobre cómo configurar las propiedades de los atributos en su evento, abra la sección Nota del cuadro de diálogo de propiedades de los atributos.
   > Para eliminar las propiedades de un atributo, seleccione el botón Editar de ese atributo y, a continuación, seleccione Borrar.

5. Seleccione Generar patrón de eventos en JSON para generar y validar el patrón de eventos como texto JSON.
6. Para probar el evento de muestra con el patrón de prueba, seleccione Patrón de prueba.

   EventBridge muestra un cuadro de mensaje que indica si el evento de muestra coincide con el patrón de eventos.

   Puede elegir una de las siguientes opciones:

   - Copiar: copia el patrón de eventos en el portapapeles de su dispositivo.

- Prettify: facilita la lectura del texto JSON al añadir saltos de línea, tabulaciones y espacios.

Event pattern

Para escribir un patrón de eventos personalizado en formato JSON, haga lo siguiente:

1. En la sección Método de creación, en Método, seleccione Patrón personalizado (editor JSON).
2. En Patrón de eventos, introduzca el patrón de eventos personalizado en texto con formato JSON.
3. Para probar el evento de muestra con el patrón de prueba, seleccione Patrón de prueba.

   EventBridge muestra un cuadro de mensaje que indica si el evento de muestra coincide con el patrón de eventos.

   Puede elegir una de las siguientes opciones:

   - Copiar: copia el patrón de eventos en el portapapeles de su dispositivo.
   - Prettify: facilita la lectura del texto JSON al añadir saltos de línea, tabulaciones y espacios.
   - Formulario de patrón de eventos: abre el patrón de eventos en Pattern Builder. Si el patrón no se puede renderizar en Pattern Builder tal cual, le EventBridge avisa antes de abrir Pattern Builder.

6. (Opcional) Para crear una regla con este patrón de eventos y asignarla a un bus de eventos específico, seleccione Crear regla con patrón.

   EventBridge lo lleva al paso 1 de Crear regla, que puede utilizar para crear una regla y asignarla al bus de eventos que prefiera.

   Tenga en cuenta que el Paso 2: Crear un patrón de eventos contiene la información del patrón de eventos que ya ha especificado y que puede aceptar o actualizar.

   Para obtener más información sobre cómo crear reglas, consulte ???.

Mejores prácticas a la hora de definir los patrones de EventBridge eventos de Amazon

A continuación, se muestran algunas prácticas recomendadas que se deben tener en cuenta al definir los patrones de eventos en sus reglas del bus de eventos.

Evitar escribir bucles infinitos

En EventBridge, es posible crear reglas que conduzcan a bucles infinitos, en los que una regla se activa repetidamente. Por ejemplo, una regla puede detectar que las ACL han cambiado en un bucket de S3 y activar software para cambiarlas al estado deseado. Si la regla no se ha escrito minuciosamente, un nuevo cambio de las ACL vuelve a activar la regla, lo que crea un bucle infinito.

Para evitar estos problemas, escriba los patrones de eventos de sus reglas para que sean lo más precisos posible, de modo que solo coincidan con los eventos que realmente quiere que se envíen al destino. En el ejemplo anterior, crearía un patrón de eventos que coincidiera con los eventos, de modo que las acciones desencadenadas no vuelvan a activar la misma regla. Por ejemplo, cree un patrón de eventos en la regla que coincida con los eventos solo si se descubre que las ACL están en mal estado, en lugar de hacerlo después de cualquier cambio. Para obtener más información, consulte ??? y ???.

Un bucle infinito puede generar cargos superiores a los esperados rápidamente. También puede provocar una limitación y un retraso en la entrega de los eventos. Puede controlar el límite superior de tus tasas de invocación para recibir alertas sobre picos de volumen inesperados.

Utilice la función de presupuestos para que le avise cuando los cargos superen el límite especificado. Para obtener más información, consulte Gestión de costos con presupuestos.

Hacer que los patrones de eventos sean lo más precisos posible

Cuanto más preciso sea el patrón de eventos, mayor será la probabilidad de que coincida solo con los eventos que realmente desea y evitará coincidencias inesperadas cuando se agreguen nuevos eventos a un origen de eventos o se actualicen los eventos existentes para incluir nuevas propiedades.

Los patrones de eventos pueden incluir filtros que coincidan con:

- Metadatos del evento sobre el evento, como `source`, `detail-type`, `account` `oregion`.
- Datos de eventos, es decir, los campos dentro del objeto `detail`.

- El contenido del evento o los valores reales de los campos dentro del objeto `detail`.

La mayoría de los patrones son simples, como especificar solo los filtros `source` y `detail-type`. Sin embargo, EventBridge los patrones incluyen la flexibilidad de filtrar cualquier clave o valor del evento. Además, puede aplicar filtros de contenido, como filtros `prefix` y `suffix` para mejorar la precisión de sus patrones. Para obtener más información, consulte ???.

Especifique el origen del evento y el tipo de detalle como filtros

Puede reducir la generación de bucles infinitos y la coincidencia de eventos no deseados haciendo que sus patrones de eventos sean más precisos utilizando los campos de metadatos `source` y `detail-type`.

Cuando necesite hacer coincidir valores específicos dentro de dos o más campos, utilice el operador de comparación `$or` en lugar de enumerar todos los valores posibles dentro de una sola matriz de valores.

En el caso de los eventos que se envíen directamente AWS CloudTrail, te recomendamos que utilices el `eventName` campo como filtro.

El siguiente ejemplo de patrón de eventos coincide con `CreateQueue` o `SetQueueAttributes` desde el servicio Amazon Simple Queue Service `CreateKey` o con `DisableKeyRotation` eventos del AWS Key Management Service servicio.

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

```
      "detail": {
        "eventName": [
          "CreateKey",
          "DisableKeyRotation"
        ]
      }
    }
  ]
}
```

Especificar la cuenta y la región como filtros

Inclusión de los campos `region` y `account` en el patrón de eventos le ayuda a limitar la coincidencia de eventos entre cuentas o regiones.

Especificar los filtros de contenido

El filtrado basado en el contenido puede ayudar a mejorar la precisión del patrón de eventos y, al mismo tiempo, mantener la longitud del patrón de eventos al mínimo. Por ejemplo, la coincidencia basada en un rango numérico puede resultar útil en lugar de enumerar todos los valores numéricos posibles.

Para obtener más información, consulte ???.

Limitar sus patrones de eventos para tener en cuenta las actualizaciones del origen de los eventos

Al crear patrones de eventos, debe tener en cuenta que los esquemas y dominios de los eventos pueden evolucionar y expandirse con el tiempo. Una vez más, hacer que los patrones de eventos sean lo más precisos posible le ayuda a limitar las coincidencias inesperadas si el origen del evento cambia o se expande.

Por ejemplo, supongamos que está realizando la coincidencia con eventos de un nuevo microservicio que publica eventos relacionados con los pagos. Inicialmente, el servicio usa el dominio `acme.payments` y publica un solo evento, `Payment accepted`:

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

```
    "amount": "100",
    "date": "2023-06-10",
    "currency": "USD"
    }
  }
}
```

En este punto, podría crear un patrón de eventos simple que coincida con los eventos de Pago aceptado:

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

Sin embargo, supongamos que el servicio introduce más adelante un nuevo evento para los pagos rechazados:

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

En este caso, el patrón de eventos simple que creó ahora coincidirá con ambos eventos `Payment accepted` y `Payment rejected`. EventBridge enruta ambos tipos de eventos al destino especificado para su procesamiento, lo que puede provocar errores de procesamiento y costes de procesamiento adicionales.

Para limitar el patrón de eventos a `Payment accepted` eventos únicamente, debes especificar ambos `source` y`detail-type`, como mínimo:

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

También puede especificar la cuenta y la región en el patrón de eventos para limitar aún más los eventos entre cuentas o regiones que coincidan con esta regla.

```
{
```

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

Validar patrones de eventos

Para garantizar que las reglas coincidan con los eventos deseados, le recomendamos encarecidamente que valide sus patrones de eventos. Puede validar sus patrones de eventos mediante la EventBridge consola o la API:

- En la EventBridge consola, puedes crear y probar patrones de eventos como parte de la creación de una regla o por separado mediante el Sandbox.
- Puedes probar tus patrones de eventos mediante programación mediante la acción. TestEventPattern

EventBridge Reglas de Amazon

Usted especifica EventBridge qué hacer con los eventos que se envían a cada bus de eventos. Para ello, debe crear reglas. Una regla especifica qué eventos enviar a qué destinos para su procesamiento. Una sola regla puede enviar un evento a varios destinos, que luego se ejecutan en paralelo.

Puede crear dos tipos de reglas:

- Reglas que coinciden con los datos de los eventos

  Puede crear reglas que coincidan con los eventos entrantes en función de los criterios de los datos del evento (lo que se denomina patrón de eventos). Un patrón de eventos define la estructura del evento y los campos con los que coincide una regla. Si un evento coincide con los criterios definidos en el patrón de eventos, lo EventBridge envía a los destinos que especifique.

  Para obtener más información, consulte ???.
- Reglas que se ejecutan según un cronograma

  También puede crear reglas que envíen eventos a los destinos especificados a intervalos específicos. Por ejemplo, para ejecutar una Lambda función periódicamente, puede crear una regla que se ejecute según una programación.

> **Note**
>
> EventBridge ofrece Amazon EventBridge Scheduler, un programador sin servidor que le permite crear, ejecutar y gestionar tareas desde un servicio gestionado centralizado. EventBridge Scheduler es altamente personalizable y ofrece una escalabilidad mejorada en comparación con las reglas EventBridge programadas, con un conjunto más amplio de operaciones y servicios de API de destino. AWS
> Te recomendamos que utilices EventBridge Scheduler para invocar los objetivos según un cronograma. Para obtener más información, consulte ???.

En el siguiente vídeo se repasan los aspectos básicos de las reglas: Qué son las reglas

Reglas EventBridge gestionadas por Amazon

Además de las reglas que cree, AWS los servicios pueden crear y administrar EventBridge reglas en su AWS cuenta que son necesarias para determinadas funciones de esos servicios. Se denominan reglas administradas.

Cuando un servicio crea una regla administrada, también puede crear una IAM política que conceda permiso a ese servicio para crear la regla. Las políticas de IAM creadas de esta manera se asignan de forma reducida con permisos de nivel de recursos, para permitir la creación solo de las reglas necesarias.

Puede eliminar las reglas administradas mediante la opción Forzar eliminación, pero solo debe eliminarlas si está seguro de que el otro servicio ya no necesita la regla. De lo contrario, eliminar una regla administrada hace que las características que dependen de ella dejen de funcionar.

Crear reglas de Amazon EventBridge que reaccionan a eventos

Para tomar medidas en relación con los eventos recibidos por Amazon EventBridge, puede crear reglas. Cuando un evento coincide con el patrón de eventos definido en su regla, EventBridge envía el evento al destino especificado y activa la acción definida en la regla.

En el siguiente vídeo, se explica cómo crear diferentes tipos de reglas y someterlas a prueba: Información sobre las reglas .

Utilice el siguiente procedimiento para crear una regla de Amazon EventBridge que responda a eventos.

Crear una regla que reacciona a eventos

En los pasos siguientes, se explica cómo crear una regla que EventBridge utilice para hacer coincidir los eventos a medida que se envían al bus de eventos especificado.

Pasos

- Definir la regla
- Crear el patrón de eventos
- Seleccionar los destinos
- Configure las etiquetas y revise la regla

Definir la regla

En primer lugar, escriba un nombre y la descripción de la regla para identificarla. También debe definir el bus de eventos en el que la regla busca los eventos que coincidan con un patrón de eventos.

Para definir los detalles de la regla

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Seleccione Crear regla.
4. Indique un nombre para la regla y, opcionalmente, una descripción.

Una regla no puede tener el mismo nombre que otra regla de la misma Región de AWS y del mismo bus de eventos.

5. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione Bus de eventos predeterminado de AWS. Cuando un Servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
6. En Tipo de regla, elija Regla con un patrón de evento.
7. Seleccione Siguiente.

Crear el patrón de eventos

A continuación, cree el patrón de eventos. Para ello, especifique el origen del evento, elija la base del patrón de eventos y defina los atributos y valores con los que debe coincidir. También puede generar el patrón de eventos en JSON y probarlo con un evento de muestra.

Crear el patrón de eventos

1. En Origen del evento, seleccione Eventos de AWS o eventos de socios de EventBridge.
2. (Opcional) En la sección Eventos de muestra, seleccione un tipo de evento de muestra con el que quiera probar su patrón de eventos.

   Están disponibles los siguientes eventos de muestra:

   - Eventos de AWS: seleccione uno de los eventos emitidos desde Servicios de AWS compatibles..
   - Eventos de socios de EventBridge: seleccione entre los eventos emitidos por servicios de terceros que admiten EventBridge, como Salesforce.
   - Introducir el mío: introduzca su propio evento en texto JSON.

     También puede usar un evento de AWS o un evento de socios como punto de partida para crear su propio evento personalizado.

     1. Seleccione eventos de AWS o eventos de socios de EventBridge.
     2. Use el menú desplegable Eventos de muestra para seleccionar el evento que desea usar como punto de partida para su evento personalizado.

        EventBridge muestra el evento de muestra.

3. Seleccione Copiar.
4. Seleccione Introducir el mío para Tipo de evento.
5. Elimine la estructura del evento de muestra en el panel de edición de JSON y pegue el evento de AWS o el evento de socios en su lugar.
6. Edite la JSON del evento para crear su propio evento de muestra.

3. Seleccione un método de creación. Puede crear un patrón de eventos a partir de un esquema o plantilla de EventBridge, o puede crear un patrón de eventos personalizado.

   Existing schema

   Para utilizar un esquema de EventBridge existente con el objetivo de crear el patrón de eventos, haga lo siguiente:

   1. En la sección Método de creación, en Método, seleccione Usar esquema.
   2. En la sección Patrón de eventos, para Tipo de esquema, seleccione Seleccionar esquema del registro de esquemas.
   3. Para el Registro de esquemas, seleccione el cuadro desplegable e indique el nombre de un registro de esquemas, por ejemplo, `aws.events`. También puede seleccionar una opción de la lista desplegable que aparece.
   4. En Esquema, seleccione el cuadro desplegable e indique el nombre del esquema que se va a usar. Por ejemplo, `aws.s3@ObjectDeleted`. También puede seleccionar una opción de la lista desplegable que aparece.
   5. En la sección Modelos, seleccione el botón Editar situado junto a cualquier atributo para abrir sus propiedades. Configure los campos Relación y Valor según sea necesario y, a continuación, seleccione Configurar para guardar el atributo.

      > **Note**
      >
      > Para obtener información sobre la definición de un atributo, selecciona el icono de información situado junto al nombre del atributo. Para obtener información sobre cómo configurar las propiedades de los atributos en su evento, abra la sección Nota del cuadro de diálogo de propiedades de los atributos.
      > Para eliminar las propiedades de un atributo, seleccione el botón Editar de ese atributo y, a continuación, seleccione Borrar.

6. Seleccione Generar patrón de eventos en JSON para generar y validar su patrón de eventos como texto JSON.
7. (Opcional) Para probar el evento de muestra con su patrón de prueba, seleccione Patrón de prueba.

   EventBridge muestra un cuadro de mensaje que indica si el evento de muestra coincide con el patrón de eventos.

   Puede elegir una de las siguientes opciones:

   - Copiar: copia el patrón de eventos en el portapapeles de su dispositivo.
   - Prettify: facilita la lectura del texto JSON al añadir saltos de línea, tabulaciones y espacios.

Custom schema

Para escribir un esquema personalizado y convertirlo en un patrón de eventos, haga lo siguiente:

1. En la sección Método de creación, en Método, seleccione Usar esquema.
2. En la sección Patrón de eventos, en Tipo de esquema, seleccione Introducir esquema.
3. Introduzca su esquema en el cuadro de texto. Debe formatear el esquema como texto JSON válido.
4. En la sección Modelos, seleccione el botón Editar situado junto a cualquier atributo para abrir sus propiedades. Configure los campos Relación y Valor según sea necesario y, a continuación, seleccione Configurar para guardar el atributo.

   > **Note**
   >
   > Para obtener información sobre la definición de un atributo, selecciona el icono de información situado junto al nombre del atributo. Para obtener información sobre cómo configurar las propiedades de los atributos en su evento, abra la sección Nota del cuadro de diálogo de propiedades de los atributos.
   > Para eliminar las propiedades de un atributo, seleccione el botón Editar de ese atributo y, a continuación, seleccione Borrar.

5. Seleccione Generar patrón de eventos en JSON para generar y validar su patrón de eventos como texto JSON.

6. (Opcional) Para probar el evento de muestra con su patrón de prueba, seleccione Patrón de prueba.

   EventBridge muestra un cuadro de mensaje que indica si el evento de muestra coincide con el patrón de eventos.

   Puede elegir una de las siguientes opciones:

   - Copiar: copia el patrón de eventos en el portapapeles de su dispositivo.
   - Prettify: facilita la lectura del texto JSON al añadir saltos de línea, tabulaciones y espacios.

Event pattern

Para escribir un patrón de evento personalizado en formato JSON, haga lo siguiente:

1. En la sección Método de creación, en Método, seleccione Patrón personalizado (editor JSON).
2. En Patrón de eventos, introduzca su patrón de eventos personalizado en texto con formato JSON.
3. (Opcional) Para probar el evento de muestra con su patrón de prueba, seleccione Patrón de prueba.

   EventBridge muestra un cuadro de mensaje que indica si el evento de muestra coincide con el patrón de eventos.

   Puede elegir una de las siguientes opciones:

   - Copiar: copia el patrón de eventos en el portapapeles de su dispositivo.
   - Prettify: facilita la lectura del texto JSON al añadir saltos de línea, tabulaciones y espacios.
   - Formulario de patrón de eventos: abre el patrón de eventos en Pattern Builder. Si el patrón no se puede representar en Pattern Builder tal cual, EventBridge le avisa antes de abrir Pattern Builder.

4. Seleccione Siguiente.

Seleccionar los destinos

Elija uno o más destinos para recibir los eventos que coincidan con el patrón especificado. Los destinos pueden incluir un bus de eventos de EventBridge, destinos de la API de EventBridge, incluidos socios de SaaS como Salesforce u otro Servicio de AWS.

Para seleccionar destinos

1. En Tipo de destino, seleccione uno de los siguientes tipos de destinos:

   Event bus

   Para seleccionar un bus de eventos de EventBridge, seleccione Bus de eventos de EventBridge y, a continuación, haga lo siguiente:

   - Para usar un bus de eventos en la misma Región de AWS que esta regla:
     1. Seleccione Bus de eventos en la misma cuenta y región.
     2. En Bus de eventos para destino, seleccione el cuadro desplegable e introduzca el nombre del bus de eventos. También puede seleccionar el bus de eventos en la lista desplegable.

        Para obtener más información, consulte ???.
   - Para usar un bus de eventos en una Región de AWS o una cuenta distinta:
     1. Seleccione Bus de eventos en una cuenta o región diferente.
     2. En Bus de eventos como destino, introduzca el ARN del bus de eventos que desee utilizar.

        Para obtener más información, consulte:

        - ???
        - ???

   API destination

   Para usar un destino de la API de EventBridge, seleccione Destino de la API de EventBridge y, a continuación, realice una de las siguientes acciones:

   - Para usar un destino de la API existente, selecciona Usar un destino de la API existente. A continuación, seleccione un destino de la API en la lista desplegable.

- Para crear un nuevo destino de la API, seleccione Crear un nuevo destino de la API. A continuación, proporcione los detalles del destino:
  - Nombre: introduzca un nombre para el destino.

    Los nombres deben ser únicos en su Cuenta de AWS. Los nombres pueden tener un máximo de 64 caracteres. Los caracteres válidos son A-Z, a-z, 0-9 y . _ - (guion).
  - (Opcional) Descripción: introduzca una descripción para el destino.

    Las descripciones pueden tener hasta 512 caracteres.
  - Punto de conexión de destino de la API: el punto de conexión de la URL del destino.

    La URL del punto de conexión debe empezar por **`https`**. Puede incluir el **`*`** como carácter comodín del parámetro de ruta. Puede establecer los parámetros de ruta a partir del atributo `HttpParameters` del destino.
  - Método HTTP: seleccione el método HTTP utilizado al invocar el punto de conexión.
  - (Opcional) Límite de la tasa de invocación por segundo: introduzca el número máximo de invocaciones aceptadas por segundo para este destino.

    Este valor debe ser mayor que cero. De forma predeterminada, este valor se establece en 300.
  - Conexión: elija usar una conexión nueva o existente:
    - Para usar una conexión existente, seleccione Usar una conexión existente y seleccione la conexión en la lista desplegable.
    - Para crear una conexión nueva para este destino, seleccione Crear una conexión nueva y, a continuación, defina el nombre de la conexión, el tipo de destino y el tipo de autorización. También puede agregar una descripción opcional para esta conexión.

Para obtener más información, consulte ???.

Servicio de AWS

Para usar un Servicio de AWS, seleccione y Servicio de AWS, a continuación, haga lo siguiente:

1. En Seleccionar un destino, seleccione un Servicio de AWS para usarlo como destino. Proporcione la información solicitada para el servicio que seleccione.

> **Note**
>
> Los campos que se muestran varían en función del servicio seleccionado. Para obtener más información sobre los destinos disponibles, consulte Los objetivos están disponibles en la EventBridge consola.

2. Si hay muchos tipos de destino, EventBridge necesita permisos para enviar eventos al destino. En estos casos, EventBridge puede crear el rol de IAM necesario para que se ejecute la regla.

   En Rol de ejecución, haga una de estas operaciones:

   - Para crear un nuevo rol de ejecución para esta regla:

     a. Seleccione Crear un rol nuevo para este recurso específico.

     b. Introduzca un nombre para este rol de ejecución o utilice el nombre generado por EventBridge.

   - Para usar un rol de ejecución existente para esta regla:

     a. Seleccione Usar un rol existente.

     b. Introduzca o seleccione el nombre del rol de ejecución que se va a utilizar en la lista desplegable.

3. (Opcional) En Configuración adicional, especifique cualquiera de las configuraciones opcionales disponibles para su tipo de destino:

   Event bus

   (Opcional) En Cola de mensajes fallidos, elija si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Haga una de las siguientes acciones:

   - Elija None (Ninguno) para no usar una cola de mensajes fallidos.
   - Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para utilizarla como cola de mensajes fallidos y, a continuación, seleccione de la lista desplegable la cola que quiera usar.
   - Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una

política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes.

Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.

API destination

1. (Opcional) En Configurar entrada de destino, elija cómo desea personalizar el texto enviado al destino para los eventos coincidentes. Elija una de las siguientes opciones:
   - Eventos coincidentes: EventBridge envía todo el evento de origen original al destino. Esta es la opción predeterminada.
   - Parte de los eventos coincidentes: EventBridge solo envía la parte especificada del evento de origen original al destino.

     En Especificar la parte del evento coincidente, especifique una ruta JSON que defina la parte del evento que quiere que EventBridge envíe al destino.
   - Constante (texto JSON): EventBridge envía solo el texto JSON especificado al destino. No se envía ninguna parte del evento de origen original.

     En Especificar la constante en JSON, especifique el texto JSON que quiere que EventBridge envíe al destino en lugar del evento.
   - Transformador de entrada: configure un transformador de entrada para personalizar el texto que quiere que EventBridge envíe al destino. Para obtener más información, consulte ???.

     a. Seleccione Configurar transformador de entrada.

     b. Configure el transformador de entrada siguiendo los pasos que se indican en ???.
2. (Opcional) En Política de reintentos, especifique cómo debe volver a intentar EventBridge enviar un evento a un destino después de que se produzca un error.
   - Antigüedad máxima del evento: introduzca la cantidad máxima de tiempo (en horas, minutos y segundos) para que EventBridge retenga los eventos sin procesar. El valor predeterminado es 24 horas.
   - Número de reintentos: introduzca el número máximo de veces que EventBridge debe volver a intentar enviar un evento al destino tras producirse un error. El valor predeterminado es 185 veces.

3. (Opcional) En Cola de mensajes fallidos, elija si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Haga una de las siguientes acciones:
   - Elija None (Ninguno) para no usar una cola de mensajes fallidos.
   - Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para utilizarla como cola de mensajes fallidos y, a continuación, seleccione de la lista desplegable la cola que quiera usar.
   - Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes.

     Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.

AWS service

Tenga en cuenta que es posible que EventBridge no muestre todos los campos siguientes para un servicio de AWS determinado.

1. (Opcional) En Configurar entrada de destino, elija cómo desea personalizar el texto enviado al destino para los eventos coincidentes. Elija una de las siguientes opciones:
   - Eventos coincidentes: EventBridge envía todo el evento de origen original al destino. Esta es la opción predeterminada.
   - Parte de los eventos coincidentes: EventBridge solo envía la parte especificada del evento de origen original al destino.

     En Especificar la parte del evento coincidente, especifique una ruta JSON que defina la parte del evento que quiere que EventBridge envíe al destino.
   - Constante (texto JSON): EventBridge envía solo el texto JSON especificado al destino. No se envía ninguna parte del evento de origen original.

     En Especificar la constante en JSON, especifique el texto JSON que quiere que EventBridge envíe al destino en lugar del evento.

- Transformador de entrada: configure un transformador de entrada para personalizar el texto que quiere que EventBridge envíe al destino. Para obtener más información, consulte ???.

  a. Seleccione Configurar transformador de entrada.

  b. Configure el transformador de entrada siguiendo los pasos que se indican en ???.

2. (Opcional) En Política de reintentos, especifique cómo debe volver a intentar EventBridge enviar un evento a un destino después de que se produzca un error.

   - Antigüedad máxima del evento: introduzca la cantidad máxima de tiempo (en horas, minutos y segundos) para que EventBridge retenga los eventos sin procesar. El valor predeterminado es 24 horas.
   - Número de reintentos: introduzca el número máximo de veces que EventBridge debe volver a intentar enviar un evento al destino tras producirse un error. El valor predeterminado es 185 veces.

3. (Opcional) En Cola de mensajes fallidos, elija si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Haga una de las siguientes acciones:

   - Elija None (Ninguno) para no usar una cola de mensajes fallidos.
   - Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para utilizarla como cola de mensajes fallidos y, a continuación, seleccione de la lista desplegable la cola que quiera usar.
   - Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes.

     Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.

4. (Opcional) Seleccione Agregar otro destino para agregar otro destino para esta regla.
5. Seleccione Siguiente.

Tenga en cuenta que es posible que EventBridge no muestre todos los campos siguientes para un servicio de AWS determinado.

Configure las etiquetas y revise la regla

Por último, introduzca las etiquetas que desee para la regla y, a continuación, revise y cree la regla.

Para configurar las etiquetas y revisar y crear la regla

1. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulte EventBridge Etiquetas de Amazon.
2. Seleccione Siguiente.
3. Revise los detalles de la nueva regla. Para realizar cambios en cualquier sección, pulse el botón Editar situado junto a esa sección.

   Cuando esté satisfecho con los detalles de la regla, seleccione Crear regla.

Usar el Programador de Amazon EventBridge con Amazon EventBridge

El Programador de Amazon EventBridge es un programador sin servidor que le permite crear, ejecutar y administrar tareas desde un servicio administrado y centralizado. Con el Programador de EventBridge, puede crear programadores mediante expresiones cron y de frecuencia para patrones recurrentes, o configurar invocaciones únicas. Puede configurar intervalos de tiempo flexibles para la entrega, definir límites de reintentos y establecer el tiempo máximo de retención para las invocaciones de la API.

El programador de EventBridge es altamente personalizable y ofrece una escalabilidad mejorada en comparación con las reglas programadas de EventBridge, con un conjunto más amplio de operaciones de API y servicios de AWS de destino. Se recomienda utilizar el programador de EventBridge para invocar los destinos en un programa.

Temas

Configurar el rol de ejecución

Al crear una programación nueva, el Programador de EventBridge debe tener permiso para invocar la operación de la API de destino en su nombre. Estos permisos se conceden al Programador de EventBridge mediante un rol de ejecución. La política de permisos que adjunta a la función de ejecución de su programación define los permisos necesarios. Estos permisos dependen de la API de destino que quiera que invoque el Programador de EventBridge.

Al utilizar la consola del Programador de EventBridge para crear una programación, como en el siguiente procedimiento, el Programador de EventBridge configura de forma automática un rol de ejecución en función del destino seleccionado. Si desea crear una programación con uno de los SDK del Programador de EventBridge, la AWS CLI o AWS CloudFormation, debe tener un rol de ejecución existente que conceda los permisos que el Programador de EventBridge requiere para invocar un destino. A fin de obtener más información sobre cómo configurar de forma manual un rol de ejecución para su programación, consulte Setting up the execution role en EventBridge Scheduler User Guide.

Crear una programación

Para crear una programación con la consola, realice lo siguiente:

1. Abra la consola del Programador de Amazon EventBridge enhttps://console.aws.amazon.com/scheduler/home.
2. En la página de Programaciones, elija Crear programación.
3. En la página de Especificar los detalles de la programación, en la sección de Nombre y descripción de la programación, realice lo siguiente:

   a. En Nombre de la programación, escriba un nombre para la programación. Por ejemplo, **`MyTestSchedule`**.

   b. (Opcional) En Descripción, escriba una descripción para su programación. Por ejemplo, **`My first schedule`**.

   c. En Grupo de programaciones, elija un grupo de programaciones de la lista desplegable. Si no tiene un grupo, elija predeterminado. Para crear un grupo de programaciones, elija crear mi propia programación.

   Los grupos de programaciones se utilizan para agregar etiquetas a grupos de programaciones.

4. • Elija sus opciones de programación.

Ocurrencia	Haga lo siguiente...	
Programación única Una programación única invoca solo una vez un objetivo en la fecha y hora que especifique.	En Fecha y hora, realice lo siguiente: • Ingrese una fecha válida en el formato `YYYY/MM/DD` . • Ingrese una marca de tiempo en el formato `hh:mm` de 24 horas. • En Zona horaria, elija la zona horaria.	

Ocurrencia	Haga lo siguiente...	
Programación recurrente Una programación recurrente invoca un objetivo a una velocidad que especifique mediante una expresión cron o rate.	a. En Tipo de programación, realice una de las siguientes acciones: • Para utilizar una expresión Cron para definir la programación, elija Programación basada en Cron e ingrese la expresión Cron. • Para utilizar una expresión de frecuencia para definir la programación, elija Programación basada en la frecuencia e ingrese la expresión de frecuencia. Para obtener más información sobre las expresiones cron y rate, consulte Schedule types on EventBridge Scheduler en Amazon EventBridge Scheduler User Guide. b. En Intervalo de tiempo flexible, elija Apagado para desactivar la opción o elegir uno de los periodos de	

Ocurrencia	Haga lo siguiente...	
	tiempo predefini dos. Por ejemplo, si elige 15 minutos y establece una programación recurrente para invocar su objetivo una vez cada hora, el horario se ejecuta 15 minutos después del inicio de cada hora.	

5. (Opcional) Si elige Programación recurrente en el paso anterior, en la sección de Periodo de tiempo, realice lo siguiente:

   a. En Zona horaria, elija una zona horaria.

   b. En Fecha y hora de inicio, ingrese una fecha válida en el formato `YYYY/MM/DD` y, a continuación, especifique una marca de tiempo en el formato `hh:mm` de 24 horas.

   c. En Fecha y hora de finalización, ingrese una fecha válida en el formato `YYYY/MM/DD` y, a continuación, especifique una marca de tiempo en el formato `hh:mm` de 24 horas.

6. Seleccione Siguiente.

7. En la página Seleccionar destino, elija la operación de la API de AWS que invoca el Programador de EventBridge:

   a. En API de destino, seleccione Destinos de plantillas.

   b. Seleccione Amazon EventBridge PutEvents.

   c. En PutEvents, especifique lo siguiente:

      - En Bus de eventos de EventBridge, elija el autobús en el menú desplegable. Por ejemplo, **`default`**.

        También puede crear un bus de eventos nuevo en la consola de EventBridge seleccionando Crear bus de eventos nuevo.

      - En Detail-type, introduzca el tipo de detalle de los eventos que quiere que coincidan. Por ejemplo, **`Object Created`**.

      - En Source, introduzca el nombre del servicio que es el origen de los eventos.

En Eventos de servicios de AWS, especifique el prefijo del servicio como origen. No incluya el prefijo `aws.`. Por ejemplo, para los eventos de Amazon S3, introduzca **`s3`**.

Para determinar el prefijo de un servicio, consulte Tabla de claves de condición en la Referencia de autorización del servicio. Para obtener más información sobre los valores de los eventos source y detail-tipe, consulte. ???

- (Opcional): en Detalles, introduzca un patrón de eventos para filtrar aún más los eventos que el Programador de EventBridge envía a EventBridge.

  Para obtener más información, consulte ???.

8. Seleccione Siguiente.
9. En la página Configuración, haga lo siguiente:

   a. Para activar la programación, en Estado de la programación, cambie a Habilitar programación.

   b. A fin de configurar una política de reintentos para su programación, en Política de reintento y cola de mensajes fallidos (DLQ), realice lo siguiente:

      - Cambie a Reintentar.
      - En Antigüedad máxima del evento, ingrese el máximo de horas y minutos que el Programador de EventBridge debe mantener un evento sin procesar.
      - El tiempo máximo es de 24 horas.
      - En Cantidad máxima de reintentos, ingrese el número máximo de veces que el Programador de EventBridge reintenta la programación si el objetivo devuelve un error.

        El valor máximo es 185 reintentos.

      Con las políticas de reintentos, si un programa no puede invocar su objetivo, el Programador de EventBridge vuelve a ejecutar el programa. Si se encuentra configurado, debe establecer el tiempo máximo de retención y los reintentos máximos para la programación.

   c. Elija dónde almacena los eventos no entregados el Programador de EventBridge.

Opción Cola de mensajes fallidos (DLQ)	Haga lo siguiente...
No almacenar	Seleccione Ninguno.
Guardar el evento en la misma Cuenta de AWS donde crea la programación	a. Elija Seleccionar una cola de Amazon SQS en mi Cuenta de AWS como DLQ. b. Elija el Nombre de recurso de Amazon (ARN) para la cola de Amazon SQS.
Guardar el evento en una Cuenta de AWS diferente de donde crea la programación	a. Elija Especificar una cola de Amazon SQS en otras Cuentas de AWS como DLQ. b. Ingrese el Nombre de recurso de Amazon (ARN) para la cola de Amazon SQS.

d. Para utilizar una clave administrada por el cliente a fin de cifrar la entrada de destino, en Cifrado, elija Personalizar la configuración de cifrado (avanzado).

   Si elige esta opción, ingrese un ARN de clave de KMS existente o elija Crear una AWS KMS key para navegar hasta la consola de AWS KMS. Para obtener más información sobre cómo el Programador de EventBridge cifra los datos en reposo, consulte Encryption at rest en Amazon EventBridge Scheduler User Guide.

e. Para que el Programador de EventBridge cree un rol de ejecución nuevo en su nombre, elija Crear un nuevo rol para esta programación. A continuación, ingrese un nombre para el Nombre de rol. Si elige esta opción, el Programador de EventBridge adjunta al rol los permisos necesarios para el objetivo creado con la plantilla.

10. Seleccione Siguiente.

11. En la página de Revisar y crear una programación, revise los detalles de su programación. En cada sección, elija Editar para volver a ese paso y editar sus detalles.
12. Elija Crear programación.

    Puede ver una lista de sus programaciones nuevas y existentes en la página de Programaciones. En la columna Estado, verifique que su programación nueva se encuentre Habilitada.

Recursos relacionados

Para obtener más información sobre el Programador de EventBridge, consulte lo siguiente:

- EventBridge Scheduler User Guide
- Referencia de la API del Programador de EventBridge
- Precios del Programador de EventBridge

Crear una regla de Amazon EventBridge que se ejecuta según una programación

Una regla se puede ejecutar en respuesta a un evento o en determinados intervalos de tiempo. Por ejemplo, para ejecutar una función AWS Lambda periódicamente, puede crear una regla que se ejecute según una programación.

> **Note**
>
> El Programador de Amazon EventBridge es un programador sin servidor que le permite crear, ejecutar y administrar tareas desde un servicio administrado y centralizado. El programador de EventBridge es altamente personalizable y ofrece una escalabilidad mejorada en comparación con las reglas programadas de EventBridge, con un conjunto más amplio de operaciones de API y servicios de AWS de destino.
> Se recomienda utilizar el Programador de EventBridge para invocar los destinos en una programación. Para obtener más información, consulte ???.

En EventBridge, puede crear dos tipos de reglas programadas:

- Reglas que se ejecutan con una frecuencia regular

  EventBridge ejecuta estas reglas a intervalos regulares; por ejemplo, cada 20 minutos.

  Para especificar la frecuencia de una regla programada, debe definir una expresión de frecuencia.
- Reglas que se ejecutan en momentos concretos

  EventBridge ejecuta estas reglas a horas y fechas específicas; por ejemplo, a las 8:00 a.m. PST el primer lunes de cada mes.

  Para especificar la hora y las fechas en que se ejecuta una regla programada, debe definir una expresión cron.

Las expresiones de frecuencia son más sencillas de definir, mientras que las expresiones cron ofrecen un control detallado de la programación. Por ejemplo, con una expresión cron, puede definir una regla que se ejecute a una hora especificada de un determinado día de cada semana o mes. Por el contrario, las expresiones de frecuencia ejecutan una regla con una frecuencia regular, como una vez cada hora o una vez cada día.

Todos los eventos programados utilizan la zona horaria UTC+0 y la precisión mínima para una programación es de 1 minuto.

> **Note**
>
> EventBridge no proporciona precisión de segundo nivel en expresiones de programación. La mejor resolución al utilizar una expresión cron es 1 minuto. Debido a la naturaleza distribuida de EventBridge y los servicios de destino, puede producirse un retraso de varios segundos entre el momento en que la regla programada se activa y el momento en que el servicio de destino ejecuta el recurso de destino.

En el siguiente vídeo se ofrece una visión general de la programación de tareas: Crear tareas programadas con EventBridge

Temas

- Crear una regla que se ejecuta según una programación
- Referencia de expresiones cron

- Referencia de expresiones de frecuencia

Crear una regla que se ejecuta según una programación

En los pasos siguientes, se explica cómo crear una regla de EventBridge que se ejecuta según una programación regular.

> **Note**
>
> Puede crear reglas programadas utilizando solo el bus de eventos predeterminado.

Pasos

- Definir la regla
- Definir la programación
- Seleccionar los destinos
- Configure las etiquetas y revise la regla

Definir la regla

En primer lugar, escriba un nombre y la descripción de la regla para identificarla.

Para definir los detalles de la regla

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Seleccione Crear regla.
4. Indique un nombre para la regla y, opcionalmente, una descripción.

   Una regla no puede tener el mismo nombre que otra regla de la misma Región de AWS y del mismo bus de eventos.
5. En Bus de eventos, seleccione el bus de eventos predeterminado. Puede crear reglas programadas utilizando solo el bus de eventos predeterminado.
6. Para que la regla surta efecto en cuanto la cree, asegúrese de que la opción Activar la regla en el bus de eventos seleccionado esté habilitada.

7. En Tipo de regla, seleccione Programación.

   En este punto, puede elegir continuar con la creación de una regla que se ejecute según una programación o usar el Programador de Amazon EventBridge.

8. Elija cómo quiere continuar:

   - Usar el Programador de Amazon EventBridge para crear su programación

     > **Note**
     >
     > El Programador de Amazon EventBridge es un programador sin servidor que le permite crear, ejecutar y administrar tareas desde un servicio administrado y centralizado. Proporciona una funcionalidad de programación única y recurrente, independientemente de las reglas y los buses de eventos. El programador de EventBridge es altamente personalizable y ofrece una escalabilidad mejorada en comparación con las reglas programadas de EventBridge, con un conjunto más amplio de operaciones de API y servicios de AWS de destino.
     > Se recomienda utilizar el Programador de EventBridge para invocar los destinos en una programación. Para obtener más información, consulte ¿Qué es el Programador de Amazon EventBridge? en la Guía del usuario del Programador de Amazon EventBridge.

     1. Seleccione Continuar en el Programador de EventBridge

        EventBridge abre la consola del Programador de EventBridge en la página Crear programación.

     2. Cree la programación en la consola del Programador de EventBridge.

   - Siga utilizando EventBridge para crear una regla programada para el bus de eventos predeterminado

     1. Seleccione Continuar para crear la regla.

Definir la programación

A continuación, defina el patrón de programación.

Para definir el patrón de programación

1. En Patrón de programación, elija si desea que la programación se ejecute a una hora concreta o con una frecuencia regular:

   Specific time

   1. Seleccione Una programación detallada que se ejecute a una hora concreta, como las 8:00 a. m. PST el primer lunes de cada mes.
   2. En Expresión Cron, especifique los campos para definir la expresión cron que EventBridge debe utilizar para determinar cuándo ejecutar esta regla programada.

      Una vez que haya especificado todos los campos, EventBridge mostrará las diez fechas siguientes en las que EventBridge ejecutará esta regla programada. Puede elegir si desea mostrar esas fechas en UTC o en la zona horaria local.

      Para obtener más información sobre la creación de una expresión cron, consulte ???.

   Regular rate

   1. Seleccione Una programación que se ejecute con una frecuencia regular, por ejemplo, cada 10 minutos.
   2. En Expresión de frecuencia, especifique los campos Valor y Unidad para definir la frecuencia con la que EventBridge debe ejecutar esta regla programada.

      Para obtener más información sobre la creación de una expresión de frecuencia, consulte ???.
2. Seleccione Siguiente.

Seleccionar los destinos

Elija uno o más destinos para recibir los eventos que coincidan con el patrón especificado. Los destino pueden incluir un bus de eventos de EventBridge, destinos de la API de EventBridge, incluidos socios de SaaS como Salesforce u otro Servicio de AWS.

Para seleccionar destinos

1. En Tipo de destino, seleccione uno de los siguientes tipos de destinos:

Event bus

Para seleccionar un bus de eventos de EventBridge, seleccione Bus de eventos de EventBridge y, a continuación, haga lo siguiente:

- Para usar un bus de eventos en la misma Región de AWS que esta regla:
  1. Seleccione Bus de eventos en la misma cuenta y región.
  2. En Bus de eventos para destino, seleccione el cuadro desplegable e introduzca el nombre del bus de eventos. También puede seleccionar el bus de eventos en la lista desplegable.

     Para obtener más información, consulte ???.
- Para usar un bus de eventos en una Región de AWS o una cuenta distinta:
  1. Seleccione Bus de eventos en una cuenta o región diferente.
  2. En Bus de eventos como destino, introduzca el ARN del bus de eventos que desee utilizar.

     Para obtener más información, consulte:
     - ???
     - ???

API destination

Para usar un destino de la API de EventBridge, seleccione el destino de la API de EventBridge y, a continuación, realice una de las siguientes acciones:

- Para usar un destino de la API existente, selecciona Usar un destino de la API existente. A continuación, seleccione un destino de la API en la lista desplegable.
- Para crear un nuevo destino de la API, seleccione Crear un nuevo destino de la API. A continuación, proporcione los detalles del destino:
  - Nombre: introduzca un nombre para el destino.

    Los nombres deben ser únicos en su Cuenta de AWS. Los nombres pueden tener un máximo de 64 caracteres. Los caracteres válidos son A-Z, a-z, 0-9 y . _ - (guion).
  - (Opcional) Descripción: introduzca una descripción para el destino.

Las descripciones pueden tener hasta 512 caracteres.

- Punto de conexión de destino de la API: el punto de conexión de la URL del destino.

  La URL del punto de conexión debe empezar por **`https`**. Puede incluir el **`*`** como carácter comodín del parámetro de ruta. Puede establecer los parámetros de ruta a partir del atributo `HttpParameters` del destino.
- Método HTTP: seleccione el método HTTP utilizado al invocar el punto de conexión.
- (Opcional) Límite de la tasa de invocación por segundo: introduzca el número máximo de invocaciones aceptadas por segundo para este destino.

  Este valor debe ser mayor que cero. De forma predeterminada, este valor se establece en 300.
- Conexión: elija usar una conexión nueva o existente:
  - Para usar una conexión existente, seleccione Usar una conexión existente y seleccione la conexión en la lista desplegable.
  - Para crear una conexión nueva para este destino, seleccione Crear una conexión nueva y, a continuación, defina el nombre de la conexión, el tipo de destino y el tipo de autorización. También puede agregar una descripción opcional para esta conexión.

Para obtener más información, consulte ???.

Servicio de AWS

Para usar un Servicio de AWS, seleccione y Servicio de AWS, a continuación, haga lo siguiente:

1. En Seleccionar un destino, seleccione un Servicio de AWS para usarlo como destino. Proporcione la información solicitada para el servicio que seleccione.

   > **Note**
   >
   > Los campos que se muestran varían en función del servicio seleccionado. Para obtener más información sobre los destinos disponibles, consulte Los objetivos están disponibles en la EventBridge consola.

2. Si hay muchos tipos de destino, EventBridge necesita permisos para enviar eventos al destino. En estos casos, EventBridge puede crear el rol de IAM necesario para que se ejecute la regla.

En Rol de ejecución, haga una de estas operaciones:

- Para crear un nuevo rol de ejecución para esta regla:
  a. Seleccione Crear un rol nuevo para este recurso específico.
  b. Introduzca un nombre para este rol de ejecución o utilice el nombre generado por EventBridge.
- Para usar un rol de ejecución existente para esta regla:
  a. Seleccione Usar un rol existente.
  b. Introduzca o seleccione el nombre del rol de ejecución que se va a utilizar en la lista desplegable.

3. (Opcional) En Configuración adicional, especifique cualquiera de las configuraciones opcionales disponibles para su tipo de destino:

   Event bus

   (Opcional) En Cola de mensajes fallidos, elija si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Haga una de las siguientes acciones:

   - Seleccione Ninguna para no usar una cola de mensajes fallidos.
   - Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para utilizarla como cola de mensajes fallidos y, a continuación, seleccione de la lista desplegable la cola que quiera usar.
   - Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes.

     Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.

API destination

1. (Opcional) En Configurar entrada de destino, elija cómo desea personalizar el texto enviado al destino para los eventos coincidentes. Elija una de las siguientes opciones:
   - Eventos coincidentes: EventBridge envía todo el evento de origen original al destino. Esta es la opción predeterminada.
   - Parte de los eventos coincidentes: EventBridge solo envía la parte especificada del evento de origen original al destino.

     En Especificar la parte del evento coincidente, especifique una ruta JSON que defina la parte del evento que quiere que EventBridge envíe al destino.
   - Constante (texto JSON): EventBridge envía solo el texto JSON especificado al destino. No se envía ninguna parte del evento de origen original.

     En Especificar la constante en JSON, especifique el texto JSON que quiere que EventBridge envíe al destino en lugar del evento.
   - Transformador de entrada: configure un transformador de entrada para personalizar el texto que quiere que EventBridge envíe al destino. Para obtener más información, consulte ???.

     a. Seleccione Configurar transformador de entrada.

     b. Configure el transformador de entrada siguiendo los pasos que se indican en ???.
2. (Opcional) En Política de reintentos, especifique cómo debe volver a intentar EventBridge enviar un evento a un destino después de que se produzca un error.
   - Antigüedad máxima del evento: introduzca la cantidad máxima de tiempo (en horas, minutos y segundos) para que EventBridge retenga los eventos sin procesar. El valor predeterminado es 24 horas.
   - Número de reintentos: introduzca el número máximo de veces que EventBridge debe volver a intentar enviar un evento al destino tras producirse un error. El valor predeterminado es 185 veces.
3. (Opcional) En Cola de mensajes fallidos, elija si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Haga una de las siguientes acciones:
   - Elija None (Ninguno) para no usar una cola de mensajes fallidos.

- Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para utilizarla como cola de mensajes fallidos y, a continuación, seleccione de la lista desplegable la cola que quiera usar.
- Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes.

  Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.

AWS service

Tenga en cuenta que es posible que EventBridge no muestre todos los campos siguientes para un servicio de AWS determinado.

1. (Opcional) En Configurar entrada de destino, elija cómo desea personalizar el texto enviado al destino para los eventos coincidentes. Elija una de las siguientes opciones:
   - Eventos coincidentes: EventBridge envía todo el evento de origen original al destino. Esta es la opción predeterminada.
   - Parte de los eventos coincidentes: EventBridge solo envía la parte especificada del evento de origen original al destino.

     En Especificar la parte del evento coincidente, especifique una ruta JSON que defina la parte del evento que quiere que EventBridge envíe al destino.
   - Constante (texto JSON): EventBridge envía solo el texto JSON especificado al destino. No se envía ninguna parte del evento de origen original.

     En Especificar la constante en JSON, especifique el texto JSON que quiere que EventBridge envíe al destino en lugar del evento.
   - Transformador de entrada: configure un transformador de entrada para personalizar el texto que quiere que EventBridge envíe al destino. Para obtener más información, consulte ???.

     a. Seleccione Configurar transformador de entrada.

     b. Configure el transformador de entrada siguiendo los pasos que se indican en ???.

2. (Opcional) En Política de reintentos, especifique cómo debe volver a intentar EventBridge enviar un evento a un destino después de que se produzca un error.
   - Antigüedad máxima del evento: introduzca la cantidad máxima de tiempo (en horas, minutos y segundos) para que EventBridge retenga los eventos sin procesar. El valor predeterminado es 24 horas.
   - Número de reintentos: introduzca el número máximo de veces que EventBridge debe volver a intentar enviar un evento al destino tras producirse un error. El valor predeterminado es 185 veces.
3. (Opcional) En Cola de mensajes fallidos, elija si desea utilizar una cola de Amazon SQS estándar como cola de mensajes fallidos. EventBridge envía eventos que coincidan con esta regla a la cola de mensajes fallidos si no se entregan correctamente al destino. Haga una de las siguientes acciones:
   - Elija None (Ninguno) para no usar una cola de mensajes fallidos.
   - Elija Seleccionar una cola de Amazon SQS en la cuenta de AWS actual para utilizarla como cola de mensajes fallidos y, a continuación, seleccione de la lista desplegable la cola que quiera usar.
   - Elija Seleccionar una cola de Amazon SQS en otra cuenta de AWS como cola de mensajes fallidos y luego ingrese el ARN de la cola que quiera usar. Debe asociar una política basada en recursos a la cola que conceda permiso a EventBridge para enviarle mensajes.

     Para obtener más información, consulte Concesión de permisos a la cola de mensajes fallidos.
4. (Opcional) Seleccione Agregar otro destino para agregar otro destino para esta regla.
5. Seleccione Siguiente.

Configure las etiquetas y revise la regla

Por último, introduzca las etiquetas que desee para la regla y, a continuación, revise y cree la regla.

Para configurar las etiquetas y revisar y crear la regla

1. (Opcional) Introduzca una o varias etiquetas para la regla. Para obtener más información, consulte EventBridge Etiquetas de Amazon.
2. Seleccione Siguiente.

3. Revise los detalles de la nueva regla. Para realizar cambios en cualquier sección, pulse el botón Editar situado junto a esa sección.

   Cuando esté satisfecho con los detalles de la regla, seleccione Crear regla.

Referencia de expresiones cron

Las expresiones Cron tienen seis campos obligatorios, que están separados por un espacio en blanco.

Sintaxis

```
cron(fields)
```

Campo	Valores	Caracteres comodín
Minutos	0-59	, - * /
Horas	0-23	, - * /
Día del mes	1-31	, - * ? / L W
Mes	1-12 o JAN-DEC	, - * /
Día de la semana	1-7 o SUN-SAT	, - * ? L #
Año	1970-2199	, - * /

Caracteres comodín

- El carácter comodín , (coma) incluye valores adicionales. En el campo Mes, JAN, FEB, MAR incluiría enero, febrero y marzo.
- El carácter comodín - (guion) especifica los intervalos. En el campo Día, 1-15 incluiría los días del 1 al 15 del mes especificado.
- El * (asterisco) incluye todos los valores del campo. En el campo Horas, * incluye cada hora. No puede utilizar * en los campos Día del mes y Día de la semana. Si lo utiliza en uno, debe utilizar ? en el otro.

- El comodín / (barra inclinada) especifica incrementos. En el campo Minutos, puede escribir 1/10 para especificar cada décimo minuto, empezando desde el primer minuto de la hora (por ejemplo, los minutos 11, 21 y 31, etc.).
- El comodín ? (signo de interrogación) especifica uno u otro. En el campo Día del mes puede escribir 7 y si cualquier día de la semana fuera aceptable, podría escribir ? en el campo Día de la semana.
- El comodín L en los campos Día del mes o Día de la semana especifica el último día del mes o de la semana.
- El comodín **`W`** en el campo Día del mes especifica un día de la semana. En el campo Día del mes, **`3W`** especifica el día de la semana más cercano al tercer día del mes.
- El comodín # en el campo Día de la semana especifica una instancia concreta del día de la semana de un mes. Por ejemplo, 3#2 sería el segundo martes del mes: el número 3 hace referencia al martes, ya que es el tercer día de la semana en el calendario anglosajón, mientras que 2 hace referencia al segundo día de ese tipo dentro de un mes.

> **Note**
>
> Si utiliza un carácter '#', solo puede definir una expresión en el campo Día de la semana. Por ejemplo, `"3#1,6#3"` no es válido porque se interpreta como dos expresiones.

Limitaciones

- No se pueden especificar los campos Día del mes y Día de la semana en la misma expresión Cron. Si especifica un valor o un * (asterisco) en uno de los campos, debe utilizar un ? (signo de interrogación) en el otro.
- No se admiten las expresiones Cron que conducen a frecuencias superiores a 1 minuto.

Ejemplos

Puede utilizar las siguientes cadenas cron de ejemplo al crear una regla con programa.

Minutos	Horas	Día del mes	Mes	Día de la semana	Año	Significado
0	10	*	*	?	*	Ejecutar a las 10:00 h (UTC+0) todos los días
15	12	*	*	?	*	Ejecutar a las 12:15 h (UTC+0) todos los días
0	18	?	*	MON-FRI	*	Ejecutar a las 18:00 h (UTC+0) de lunes a viernes
0	8	1	*	?	*	Ejecutar a las 8:00 h (UTC +0) cada primer día del mes
0/15	*	*	*	?	*	Ejecutar cada 15 minutos
0/10	*	?	*	MON-FRI	*	Ejecutar cada 10 minutos de lunes a viernes

Minutos	Horas	Día del mes	Mes	Día de la semana	Año	Significado
0/5	8-17	?	*	MON-FRI	*	Ejecutar cada 5 minutos de lunes a viernes entre las 8:00 y las 17:55 h (UTC+0)
0/30	20-2	?	*	MON-FRI	*	Ejecutar cada 30 minutos de lunes a viernes, entre las 22:00 h del día de inicio y las 2:00 h del día siguiente (UTC) Ejecutar de 12:00 a 2:00 h el lunes por la mañana (UTC).

En el siguiente ejemplo se crea una regla que se ejecuta cada día a las 12:00 h (UTC+0).

```
aws events put-rule --schedule-expression "cron(0 12 * * ? *)" --name MyRule1
```

En el siguiente ejemplo se crea una regla que se ejecuta cada día a las 14:05 y a las 14:35 h (UTC +0).

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

El siguiente ejemplo crea una regla que se activa a las 10:15 UTC+0 el último viernes de cada mes, entre los años 2019 y 2022.

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

Referencia de expresiones de frecuencia

Una expresión de frecuencia comienza cuando se crea una regla de evento programado y, a continuación, se ejecuta en una programación definida.

Las expresiones de frecuencia tienen dos campos obligatorios, que están separados por un espacio en blanco.

Sintaxis

```
rate(value unit)
```

valor

Un número positivo.

unidad

La unidad de tiempo. Se requieren diferentes unidades para valores de 1, como `minute`, y valores superiores a 1, como `minutes`.

Valores válidos: minuto | minutos | hora | horas | día | días

Limitaciones

Si el valor es igual a 1, entonces la unidad debe ser singular. Si el valor es mayor que 1, la unidad debe ser plural. Por ejemplo, la frecuencia(1 horas) y la frecuencia(5 hora) no son válidas, pero la frecuencia(1 hora) y la frecuencia(5 horas) son válidas.

Ejemplos

Los siguientes ejemplos muestran cómo utilizar expresiones de frecuencia con el comando `put-rule` de la AWS CLI. El primer ejemplo activa la regla cada minuto, el siguiente ejemplo la activa cada cinco minutos, el tercer ejemplo la activa una vez cada hora y el último ejemplo la activa una vez al día.

```
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
```

Deshabilitar o eliminar una regla de Amazon EventBridge

Para impedir que una regla procese eventos o se ejecute según una programación, puede eliminarla o deshabilitarla. En los pasos siguientes, se explica cómo eliminar o deshabilitar una regla de EventBridge.

Para eliminar o deshabilitar una regla

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.

   En Bus de eventos, seleccione el bus de eventos asociado a la regla.
3. Haga una de las siguientes acciones:

   a. Para eliminar una regla, seleccione el botón que aparece junto a la regla y seleccione Acciones, Eliminar, Eliminar.

      Si la regla es una regla administrada, introduzca el nombre de la regla para confirmar que se trata de una regla administrada y que la eliminación puede detener la funcionalidad en el servicio que ha creado la regla. Para continuar, introduzca el nombre de la regla y elija Forzar eliminación.

   b. Para deshabilitar temporalmente una regla, seleccione el botón que aparece junto a la regla y elija Deshabilitar, Deshabilitar.

      No puede deshabilitar una regla administrada.

Prácticas recomendadas a la hora de definir reglas de Amazon EventBridge

A continuación, se muestran algunas prácticas recomendadas que se deben tener en cuenta al crear reglas para los buses de eventos.

Establecer un destino único para cada regla

Si bien puede especificar hasta cinco destinos para una regla determinada, le resultará más fácil administrar las reglas si especifica un único destino para cada regla. Si más de un destino necesita recibir el mismo conjunto de eventos, se recomienda duplicar la regla para enviar los mismos

eventos a destinos diferentes. Esta encapsulación simplifica el mantenimiento de las reglas: si las necesidades de los destinos de eventos varían con el tiempo, puede actualizar cada regla y su patrón de eventos de forma independiente de las demás.

Establecer permisos de reglas

Puede habilitar los componentes o servicios de la aplicación que consumen muchos eventos para que controlen la administración de sus propias reglas. Un enfoque arquitectónico común que adoptan los clientes consiste en aislar estos componentes o servicios de la aplicación mediante cuentas de AWS independientes. Para habilitar el flujo de eventos de una cuenta a otra, debe crear una regla en un bus de eventos que dirija los eventos a un bus de eventos de otra cuenta. Puede habilitar los equipos o servicios que consumen muchos eventos para que controlen la administración de sus propias reglas. Para ello, especifique los permisos adecuados para sus cuentas mediante políticas de recursos. Esto funciona en todas las cuentas y regiones.

Para obtener más información, consulte ???.

Para ver un ejemplo de políticas de recursos, consulte Patrones de diseño de varias cuentas con Amazon EventBridge en GitHub.

Supervisar el rendimiento de las reglas

Supervise sus reglas para asegurarse de que funcionan según lo esperado:

- Supervisar la métrica `TriggeredRules` para detectar puntos de datos faltantes o anomalías puede ayudarte a detectar discrepancias si un editor ha realizado un cambio radical. Para obtener más información, consulte ???.
- La alarma en caso de anomalías o el recuento máximo esperado también pueden ayudar a detectar si una regla coincide con nuevos eventos. Esto puede suceder cuando los publicadores, incluidos los servicios de AWS y los socios de SaaS, introducen nuevos eventos al habilitar nuevos casos de uso y características. Cuando estos nuevos eventos son inesperados y generan un volumen superior a la velocidad de procesamiento del destino final, pueden provocar una acumulación de eventos pendientes.

  Este procesamiento de eventos inesperados también puede generar cargos de facturación no deseados.

  También puede provocar una limitación de las reglas cuando la cuenta supera su cuota de servicio de invocaciones de destino totales por segundo. EventBridge seguirá intentando entregar eventos

que coincidan con reglas limitadas y lo volverá a intentar en un plazo máximo de 24 horas o según se describe en la política de reintentos personalizada del destino. Puede detectar y activar las reglas limitadas mediante la métrica `ThrottledRules`

- En los casos de uso de baja latencia, también puede supervisar la latencia mediante `IngestionToInvocationStartLatency`, que proporciona una indicación del estado del bus de eventos. Cualquier período prolongado de alta latencia de más de 30 segundos puede indicar una interrupción del servicio o una limitación de las reglas.

Usar Amazon EventBridge y plantillas de AWS Serverless Application Model

Puede crear y probar reglas manualmente en la consola de EventBridge, lo que puede ayudarte en el proceso de desarrollo a medida que refina los patrones de eventos. Sin embargo, una vez que esté preparado para implementar su aplicación, será más fácil utilizar un marco como AWS SAM para lanzar todos los recursos sin servidor de forma coherente.

Usaremos esta aplicación de ejemplo para analizar las formas en que puede usar las plantillas de AWS SAM para crear recursos de EventBridge. El archivo template.yaml de este ejemplo es una plantilla de AWS SAM que define cuatro funciones de AWS Lambda y muestra dos formas diferentes de integrar las funciones de Lambda con EventBridge.

Para ver esta aplicación de ejemplo, consulte ???.

Existen dos métodos para usar EventBridge y plantillas de AWS SAM. Para integraciones sencillas en las que una sola regla invoca una función de Lambda, se recomienda el método de Plantilla combinada. Si tiene una lógica de enrutamiento compleja o se conecta a recursos fuera de su plantilla de AWS SAM, la mejor opción es el método de Plantillas separadas.

Métodos:

- Plantilla combinada
- Plantilla separada

Plantilla combinada

El primer método utiliza la propiedad `Events` para configurar la regla de EventBridge. El siguiente código de ejemplo define un evento que invoca la función de Lambda.

> **Note**
>
> En este ejemplo, se crea automáticamente la regla en el bus de eventos predeterminado, que existe en todas las cuentas de AWS. Para asociar la regla a un bus de eventos personalizado, puede el `EventBusName` a la plantilla.

```
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"
```

Este código YAML equivale a un patrón de eventos de la consola de EventBridge. En YAML, solo necesita definir el patrón de eventos y AWS SAM crea automáticamente un rol de IAM con los permisos necesarios.

Plantilla separada

En el segundo método para definir una configuración de EventBridge en AWS SAM, los recursos se separan más claramente en la plantilla.

1. En primer lugar, defina la función de Lambda:

   ```
   atmConsumerCase1Fn:
     Type: AWS::Serverless::Function
     Properties:
       CodeUri: atmConsumer/
       Handler: handler.case1Handler
       Runtime: nodejs12.x
   ```

2. A continuación, defina la regla mediante un recurso de `AWS::Events::Rule`. Las propiedades definen el patrón de eventos y también pueden especificar destinos. Puede definir varios destino de forma explícita.

   ```
   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. Por último, defina un recurso de `AWS::Lambda::Permission` que conceda permiso a EventBridge para invocar el destino.

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

Generar una plantilla de AWS CloudFormation a partir de reglas de Amazon EventBridge

AWS CloudFormation le permite configurar y administrar sus recursos de AWS en todas las cuentas y regiones de manera centralizada y repetible al tratar la infraestructura como código. Para ello, CloudFormation le permite crear plantillas, que definen los recursos que desea aprovisionar y administrar.

EventBridge le permite generar plantillas a partir de las reglas existentes en su cuenta, para ayudarte a empezar a desarrollar plantillas de CloudFormation. Puede seleccionar una sola regla o varias reglas para incluirlas en la plantilla. A continuación, puede utilizar estas plantillas como base para crear pilas de recursos bajo administración de CloudFormation.

Para obtener más información sobre CloudFormation, consulte la Guía del usuario de AWS CloudFormation.

> **Note**
>
> EventBridge no incluye reglas gestionadas en la plantilla generada.

También puede generar una plantilla a partir de un bus de eventos existente, incluidas las reglas que contiene el bus de eventos.

Para generar una plantilla de AWS CloudFormation a partir de una o varias reglas

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. En Seleccionar bus de eventos, elija el bus de eventos que contenga las reglas que desee incluir en la plantilla.
4. En Reglas, elija las reglas que desee incluir en la plantilla de AWS CloudFormation generada.

   Para una sola regla, también puede elegir el nombre de la regla para mostrar la página de detalles de la regla.
5. Seleccione Plantilla de CloudFormation y, a continuación, elija el formato en el que desea que EventBridge genere la plantilla: JSON o YAML.

   EventBridge muestra la plantilla, generada en el formato seleccionado.
6. EventBridge le da la opción de descargar el archivo de plantilla o copiar la plantilla al portapapeles.

   - Seleccione Descargar para descargar el archivo de plantilla.
   - Para copiar la plantilla al portapapeles, seleccione Copiar.
7. Para salir de la plantilla, seleccione Cancelar.

Una vez que haya personalizado la plantilla de AWS CloudFormation según sea necesario para su caso de uso, podrá utilizarla para crear pilas en AWS CloudFormation.

Consideraciones sobre el uso de plantillas de CloudFormation generadas a partir de Amazon EventBridge

Tenga en cuenta los siguientes factores al utilizar una plantilla de CloudFormation que haya generado a partir de EventBridge:

- EventBridge no incluye contraseñas en la plantilla generada.

  Puede editar la plantilla para incluir parámetros de plantilla que permitan a los usuarios especificar contraseñas u otra información confidencial al utilizar la plantilla para crear o actualizar una pila de CloudFormation.

  Además, los usuarios pueden usar Secrets Manager para crear un secreto en la región deseada y, a continuación, editar la plantilla generada para emplear parámetros dinámicos.
- Los destinos de la plantilla generada permanecen exactamente como se especificaron en el bus de eventos original. Esto puede provocar problemas entre regiones si no se edita correctamente la plantilla antes de utilizarla para crear pilas en otras regiones.

  Además, la plantilla generada no crea automáticamente los destinos descendentes.

EventBridge Objetivos de Amazon

Un objetivo es un recurso o punto final que EventBridge envía un evento cuando el evento coincide con el patrón de eventos definido para una regla. La regla procesa los datos del evento y envía la información pertinente al destino. Para entregar datos de eventos a un destino, EventBridge necesita permiso para acceder al recurso de destino. Puede definir hasta cinco destinos para cada regla.

Cuando se agregan destinos a una regla y esa regla se ejecuta poco después, es posible que los destinos actualizados no se invoquen inmediatamente. Espere un breve periodo para que los cambios surtan efecto.

En el siguiente vídeo se describen los aspectos básicos de los destinos: Qué es un destino

Los objetivos están disponibles en la EventBridge consola

Puede configurar los siguientes objetivos para los eventos de la EventBridge consola:

- Destino de la API
- API Gateway
- AWS AppSync;
- Cola de trabajos por lotes
- CloudWatch grupo de registros
- CodeBuild proyecto
- CodePipeline
- Llamada a la API `CreateSnapshot` de Amazon EBS
- Generador de Imágenes de EC2
- Llamada a la API `RebootInstances` de EC2
- Llamada a la API `StopInstances` de EC2
- Llamada a la API `TerminateInstances` de EC2
- Tarea de ECS
- Bus de eventos en una cuenta o región diferente

- Bus de eventos en la misma cuenta y región
- Flujo de entrega de Firehose
- Flujo de trabajo de Glue
- Plan de respuesta del Administrador de incidentes
- Plantilla de evaluación del inspector
- Flujo de Kinesis
- Función de Lambda (ASYNC)
- Consultas de API de datos de clústeres de Amazon Redshift
- Consultas de API de datos de grupos de trabajo de Amazon Redshift sin servidor
- SageMaker Tubería
- Tema de Amazon SNS

  EventBridge no admite los temas de FIFO (primero en entrar, primero en salir) de Amazon SNS.
- Cola de Amazon SQS
- Máquina de estado de Step Functions (ASYNC)
- Automatización de Systems Manager
- Systems Manager OpsItem
- Systems Manager Run Command

Parámetros de destino

Algunos destinos no envían la información de la carga útil del evento al objetivo, sino que tratan el evento como un desencadenante para invocar una API específica. EventBridge usa los parámetros de Target para determinar qué ocurre con ese objetivo. Estos incluyen los siguientes:

- Destinos de la API (Los datos enviados a un destino de la API deben coincidir con la estructura de la API. Debe usar el objeto `InputTransformer` para asegurarse de que los datos estén estructurados correctamente. Si desea incluir la carga del evento original, haga referencia a ella en el objeto `InputTransformer`.)
- API Gateway (Los datos enviados a API Gateway deben coincidir con la estructura de la API. Debe usar el objeto `InputTransformer` para asegurarse de que los datos estén estructurados correctamente. Si desea incluir la carga del evento original, haga referencia a ella en el objeto `InputTransformer`.)

- Amazon EC2 Image Builder
- `RedshiftDataParameters` (Clústeres de API de datos de Amazon Redshift)
- `SageMakerPipelineParameters`(Amazon SageMaker Runtime Model Building Pipelines)

> **Note**
>
> EventBridge no admite toda la sintaxis de rutas JSON y la evalúa en tiempo de ejecución. La sintaxis admitida incluye:
>
> - notación de puntos (por ejemplo, `$.detail`)
> - guiones
> - guiones bajos
> - Caracteres alfanuméricos
> - índices de matrices
> - caracteres comodín (*)

Parámetros de ruta dinámicos

Algunos parámetros de destino admiten sintaxis de ruta JSON dinámica opcional. Esta sintaxis le permite especificar rutas JSON en lugar de valores estáticos (por ejemplo`$.detail.state`). El valor completo debe ser una ruta JSON, no solo una parte de ella. Por ejemplo, `RedshiftParameters.Sql` puede ser `$.detail.state`, pero no puede ser `"SELECT * FROM $.detail.state"`. Estas rutas se sustituyen dinámicamente en tiempo de ejecución por datos de la propia carga del evento en la ruta especificada. Los parámetros de ruta dinámicos no pueden hacer referencia a valores nuevos o transformados que resulten de la transformación de entrada. La sintaxis admitida para las rutas JSON con parámetros dinámicos es la misma que cuando se transforma la entrada. Para obtener más información, consulte ???

La sintaxis dinámica se puede utilizar en todos los campos non-enum de cadena de estos parámetros:

- `EcsParameters`
- `HttpParameters` (excepto claves `HeaderParameters`)
- `RedshiftDataParameters`

- `SageMakerPipelineParameters`

Permisos

Para realizar llamadas a la API en los recursos de su propiedad, EventBridge necesita el permiso adecuado. Para AWS Lambda los recursos de Amazon SNS, EventBridge utiliza políticas basadas en recursos. En el caso de las instancias EC2, las transmisiones de datos de Kinesis y las máquinas de estado Step Functions EventBridge , utilizan las funciones de IAM que se especifican en el parámetro en`RoleARN`. `PutTargets` Puede invocar un punto de conexión de API Gateway con la autorización de la IAM configurada, pero la función es opcional si no ha configurado la autorización. Para obtener más información, consulte Amazon EventBridge y AWS Identity and Access Management.

Si otra cuenta se encuentra en la misma región y le dio permiso, puede enviar eventos a esa cuenta. Para obtener más información, consulte Envío y recepción de EventBridge eventos de Amazon entre AWS cuentas.

Si el destino está cifrado, debe incluir la siguiente sección en la política de claves de KMS.

```
{
    "Sid": "Allow EventBridge to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": "events.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*"
}
```

EventBridge características específicas del objetivo

AWS Batch colas de trabajos

Algunos parámetros se AWS Batch `submitJob` pueden configurar mediante BatchParameters.

Otros se pueden especificar la carga del evento. Si la carga útil del evento (pasada o vía InputTransformers) contiene las siguientes claves, se asignan a los parámetros de la `submitJob` solicitud:

- `ContainerOverrides: containerOverrides`

  > **Note**
  >
  > Esto incluye solo el comando, el entorno, la memoria y las vCPU

- `DependsOn: dependsOn`

  > **Note**
  >
  > Esto incluye solo jobId

- `Parameters: parameters`

CloudWatch Grupo de registros

Si no utilizas uno InputTransformercon un objetivo de CloudWatch Logs, la carga útil del evento se utiliza como mensaje de registro y el origen del evento como marca de tiempo. Si utilizas una InputTransformer, la plantilla debe ser:

```
{"timestamp":<timestamp>,"message":<message>}
```

EventBridge agrupa las entradas enviadas a un flujo de registro; por lo tanto, EventBridge puede entregar uno o varios eventos a un flujo de registro, según el tráfico.

CodeBuild proyecto

Si se utiliza InputTransformerspara dar forma al evento de entrada a un objetivo para que coincida con la CodeBuild StartBuildRequestestructura, los parámetros se asignarán uno a uno y se transferirán a. `codeBuild.StartBuild`

Tarea de Amazon ECS

Si suele dar forma InputTransformersal evento de entrada a un Target para que coincida con la RunTask TaskOverrideestructura de Amazon ECS, los parámetros se asignarán 1 a 1 y se transferirán a. `ecs.RunTask`

Plan de respuesta del Administrador de Incidentes

Si el evento coincidente proviene de CloudWatch Alarms, los detalles del cambio de estado de la alarma se incluyen en los detalles del activador de la StartIncidentRequest llamada a Incident Manager.

Configurar destinos

Aprenda a configurar los ajustes de los EventBridge objetivos.

Destinos:

- Destinos de la API
- Amazon EventBridge apunta a Amazon API Gateway
- AWS AppSync objetivos para Amazon EventBridge
- Conexiones para destinos de punto final HTTP
- Envío y recepción de EventBridge eventos de Amazon entre AWS cuentas
- Envío y recepción de EventBridge eventos de Amazon entre AWS regiones
- Envío y recepción de EventBridge eventos de Amazon entre autobuses de eventos de la misma cuenta y región

Destinos de la API

Los destinos de las EventBridge API de Amazon son puntos de enlace HTTP que se pueden invocar como destino de una regla, de forma similar a como se invoca un AWS servicio o recurso como destino. Con los destinos de API, puede enrutar eventos entre AWS servicios, aplicaciones de software como servicio (SaaS) integradas y sus aplicaciones fuera de ellas AWS mediante llamadas a la API. Cuando especificas un destino de API como destino de una regla, EventBridge invoca el punto final HTTP para cualquier evento que coincida con el patrón de eventos especificado en la regla y, a continuación, entrega la información del evento junto con la solicitud. Con EventBridge, puedes usar cualquier método HTTP excepto CONNECT y TRACE para la solicitud. Los métodos HTTP más habituales son PUT y POST. También puede usar transformadores de entrada para personalizar el evento según los parámetros de un punto de conexión HTTP específico. Para obtener más información, consulte Transformación EventBridge de entradas de Amazon.

> **Note**
>
> Los destinos de API no admiten destinos privados, como los puntos de enlace de VPC de interfaz, incluidas las API HTTPS privadas en nubes privadas virtuales (VPC) que utilizan Application Load Balancer privados y puntos de enlace de VPC de interfaz.
> Para obtener más información, consulte ???.

> **Important**
>
> EventBridge las solicitudes a un punto final de destino de la API deben tener un tiempo de espera máximo de ejecución por parte del cliente de 5 segundos. Si el punto final de destino tarda más de 5 segundos en responder, se agota el EventBridge tiempo de espera de la solicitud. EventBridge Reintenta las solicitudes agotadas hasta los máximos configurados en tu política de reintentos. De forma predeterminada, los máximos son 24 horas y 185 veces. Una vez transcurrido el número máximo de reintentos, los eventos se envían a cola de mensajes fallidos, si dispone de ella. De lo contrario, el evento se descarta.

En el siguiente vídeo se muestra el uso del destino de la API: Uso de destinos de la API

En este tema:

- Crear un destino de la API
- Creación de reglas que envíen eventos a un destino de la API
- Rol vinculado a un servicio para los destinos de la API
- Encabezados en solicitudes a destinos de la API
- Códigos de error de destinos de la API
- Cómo afecta la tasa de invocación a la entrega del evento
- Envío de CloudEvents eventos a destinos de API
- Socios de destinos de la API

Crear un destino de la API

Cada destino de la API requiere una conexión. Una conexión especifica el tipo de autorización y las credenciales que se utilizarán para autorizar con el punto de conexión de destino de la API. Puede elegir una conexión existente o crear una conexión a la vez que crea el destino de la API. Para obtener más información, consulte ???.

Para crear un destino de API mediante la consola EventBridge

1. Inicie sesión AWS con una cuenta que tenga permisos para administrar EventBridge y abrir la EventBridgeconsola.
2. En el panel de navegación izquierdo, seleccione Destinos de la API.
3. Desplácese hacia abajo hasta la tabla Destinos de la API y, a continuación, seleccione Crear destino de la API.
4. En la página Crear destino de la API, indique un nombre para el destino de la API. Puede utilizar hasta 64 caracteres en mayúscula o minúscula, números, punto (.), guion (-) o guion bajo (_).

   El nombre debe ser exclusivo de la cuenta en la región actual.
5. Escriba una descripción para el destino de la API que se va a crear.
6. Introduzca un punto de conexión de destino de la API para el destino de la API. El punto de conexión de destino de la API es un destino de punto de conexión de invocación HTTP para eventos. La información de autorización que incluye en la conexión utilizada para este destino de la API se utiliza para autorizar en este punto de conexión. La URL debe utilizar HTTPS.
7. Introduzca el método HTTP que se utilizará para conectarse al punto de conexión de destino de la API.

8. (Opcional) En el campo Límite de tasa de invocación por segundo, indique el número máximo de invocaciones por segundo que se enviarán al punto de conexión de destino de la API.

   El límite de velocidad que establezcas puede afectar a la forma en que se EventBridge distribuyen los eventos. Para obtener más información, consulte Cómo afecta la tasa de invocación a la entrega del evento.
9. En Conexión, lleve a cabo alguna de las siguientes operaciones:
   - Seleccione Usar una conexión existente y, a continuación, seleccione la conexión que desea usar para este destino de la API.
   - Seleccione Crear una conexión nueva y, a continuación, introduzca los detalles de la conexión que desea crear. Para obtener más información, consulte Conexiones.
10. Seleccione Crear.

Creación de reglas que envíen eventos a un destino de la API

Tras crear un destino de la API, puede seleccionarlo como destino de una regla. Para utilizar un destino de la API como destino, debe proporcionar un rol de IAM con los permisos correspondientes. Para obtener más información, consulte ???.

La selección de un destino de la API como destino forma parte de la creación de la regla.

Para crear una regla que envíe eventos a un destino de la API mediante la consola

1. Siga los pasos que se indican en el procedimiento ???.
2. En el ??? paso, cuando se te pida que elijas un destino de API como tipo de destino:
   a. Selecciona el destino EventBridge de la API.
   b. Realice una de las siguientes acciones siguientes:
      - Elige Usar un destino de API existente y selecciona un destino de API existente
      - Elige Crear un nuevo destino de API y especifica la configuración necesaria para definir tu nuevo destino de API.

        Para obtener más información sobre cómo especificar los ajustes necesarios, consulte???.
   c. (Opcional): para especificar los parámetros de cabecera del evento, en Parámetros de cabecera, seleccione Añadir parámetro de cabecera.

A continuación, especifique la clave y el valor del parámetro de encabezado.

d. (Opcional): para especificar los parámetros de cadena de consulta para el evento, en Parámetros de cadena de consulta, elija Añadir parámetro de cadena de consulta.

A continuación, especifique la clave y el valor del parámetro de cadena de consulta.

3. Complete la creación de la regla siguiendo los pasos del procedimiento.

Rol vinculado a un servicio para los destinos de la API

Al crear una conexión para un destino de API, se agrega a su cuenta un rol vinculado al servicio denominado AWS ServiceRoleForAmazonEventBridgeApiDestinations. EventBridge usa el rol vinculado al servicio para crear y almacenar un secreto en Secrets Manager.
Para conceder los permisos necesarios al rol vinculado al servicio, EventBridge adjunta la AmazonEventBridgeApiDestinationsServiceRolePolicypolítica al rol. La política limita los permisos concedidos únicamente a los necesarios para que el rol interactúe con el secreto de la conexión. No se incluye ningún otro permiso y el rol solo puede interactuar con las conexiones de la cuenta para administrar el secreto.

La siguiente política es la `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/*"
        }
    ]
}
```

Para obtener más información acerca del uso de los roles vinculados a servicios, consulte Uso de roles vinculados a servicios en la documentación sobre IAM.

El rol `AmazonEventBridgeApiDestinationsServiceRolePolicy` vinculado al servicio se admite en las siguientes regiones: AWS

- Este de EE. UU. (Norte de Virginia)
- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Norte de California)
- Oeste de EE. UU. (Oregón)
- África (Ciudad del Cabo)
- Asia-Pacífico (Hong Kong)
- Asia-Pacífico (Bombay)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Tokio)
- Canadá (centro)
- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milán)
- Europa (París)
- Europa (Estocolmo)
- América del Sur (São Paulo)
- China (Ningxia)
- China (Pekín)

Encabezados en solicitudes a destinos de la API

En la siguiente sección, se detalla cómo se EventBridge gestionan los encabezados HTTP en las solicitudes a los destinos de la API.

Encabezados se incluyen en solicitudes a destinos de la API

Además de los encabezados de autorización definidos para la conexión utilizada para un destino de API, EventBridge incluye los siguientes encabezados en cada solicitud.

Clave de encabezado	Valor del encabezado
User-Agent	Amazonía//EventBridgeApiDestinations
Contenido-Tipo	Si no se especifica ningún valor de tipo de contenido personalizado, EventBridge incluye el siguiente valor predeterminado como tipo de contenido: aplicación/json; charset=utf-8
Range	bytes=0-1048575
Accept-Encoding	gzip, deflate
Connection	close
Content-Length	Un encabezado de entidad que indica el tamaño del cuerpo de la entidad, en bytes, enviado al destinatario.
Host	Un encabezado de solicitud que especifica el número de host y puerto del servidor al que se envía la solicitud.

Encabezados que no se pueden anular en las solicitudes a los destinos de la API

EventBridge no permite anular los siguientes encabezados:

- User-Agent
- Range

Se EventBridge eliminan los encabezados de las solicitudes a los destinos de la API

EventBridge elimina los siguientes encabezados de todas las solicitudes de destino de la API:

- A-IM
- Accept-Charset
- Accept-Datetime
- Accept-Encoding
- Cache-Control
- Connection
- Content-Encoding
- Content-Length
- Content-MD5
- Date
- Expect
- Forwarded
- De
- Host
- HTTP2-Settings
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Origen
- Pragma
- Proxy-Authorization
- Range
- Referer
- TE
- Trailer

- Transfer-Encoding
- User-Agent
- Upgrade
- Via
- Advertencia

Códigos de error de destinos de la API

Cuando EventBridge intenta enviar un evento a un destino de API y se produce un error, EventBridge hace lo siguiente:

- Se vuelven a intentar enviar los eventos asociados a los códigos de error 409, 429 y 5xx.
- No se vuelva a intentar enviar los eventos asociados a los códigos de error 1xx, 2xx, 3xx y 4xx (excepto 429).

EventBridge Los destinos de la API leen el encabezado de respuesta HTTP estándar `Retry-After` para saber cuánto tiempo deben esperar antes de realizar una solicitud de seguimiento. EventBridge elige el valor más conservador entre la política de reintentos definida y el `Retry-After` encabezado. Si `Retry-After` el valor es negativo, EventBridge deja de reintentar la entrega para ese evento.

Cómo afecta la tasa de invocación a la entrega del evento

Si establece la tasa de invocación por segundo en un valor muy inferior al número de invocaciones generadas, es posible que los eventos no se entreguen dentro del tiempo de reintento de 24 horas especificado para los eventos. Por ejemplo, si establece la tasa de invocación en 10 invocaciones por segundo, pero se generan miles de eventos por segundo, rápidamente tendrá una acumulación de eventos pendientes de entrega que supera las 24 horas. Para que no se pierdan eventos, configure una cola de mensajes fallidos a la que enviar los eventos con invocaciones fallidas para poder procesarlos más adelante. Para obtener más información, consulte Uso de colas con letra muerta para procesar los eventos no entregados.

Envío de CloudEvents eventos a destinos de API

CloudEvents es una especificación independiente del proveedor para el formato de eventos, con el objetivo de proporcionar interoperabilidad entre servicios, plataformas y sistemas. Se puede utilizar

EventBridge para transformar los eventos AWS de servicio CloudEvents antes de que se envíen a un destino, como un destino de API.

> **Note**
>
> El siguiente procedimiento explica cómo transformar los eventos de origen en modo estructurado CloudEvents. En la CloudEvents especificación, un mensaje en modo estructurado es aquel en el que todo el evento (atributos y datos) está codificado en la carga útil del evento.

Para obtener más información sobre la CloudEvents especificación, consulta cloudevents.io.

Para transformar los AWS eventos al formato mediante la consola CloudEvents

Para transformar los eventos al CloudEvents formato anterior a su entrega a un destino, comience por crear una regla de bus de eventos. Como parte de la definición de la regla, se utiliza un transformador de entrada para EventBridge transformar los eventos antes de enviarlos al destino que especifique.

1. Siga los pasos que se indican en el procedimiento ???.
2. En el ??? paso, cuando se te pida que elijas un destino de API como tipo de destino:
   a. Selecciona el destino EventBridge de la API.
   b. Realice una de las siguientes acciones siguientes:
      - Elige Usar un destino de API existente y selecciona un destino de API existente
      - Elige Crear un nuevo destino de API y especifica la configuración necesaria para definir tu nuevo destino de API.

        Para obtener más información sobre cómo especificar los ajustes necesarios, consulte???.
   c. Especifique los parámetros de encabezado Content-Type necesarios para los CloudEvents eventos:
      - En Parámetros de encabezado, elija Agregar parámetro de encabezado.
      - En clave, especifique`Content-Type`.

        Para el valor, especifique`application/cloudevents+json; charset=UTF-8`.

3. Especifique un rol de ejecución para su objetivo.
4. Defina un transformador de entrada para transformar los datos del evento de origen en el siguiente CloudEvents formato:

   a. En Ajustes adicionales, en Configurar la entrada de destino, elija Transformador de entrada.

      A continuación, selecciona Configurar transformador de entrada.

   b. En Transformador de entrada objetivo, especifique la ruta de entrada.

      En la siguiente ruta de entrada, el atributo de región es un atributo de extensión personalizado del CloudEvents formato. Como tal, no es obligatorio para cumplir con la CloudEvents especificación.

      CloudEvents permite usar y crear atributos de extensión no definidos en la especificación principal. Para obtener más información, incluida una lista de los atributos de extensión conocidos, consulte los atributos de CloudEvents extensión en la documentación de la CloudEvents especificación correspondiente GitHub.

      ```
      {
        "detail": "$.detail",
        "detail-type": "$.detail-type",
        "id": "$.id",
        "region": "$.region",
        "source": "$.source",
        "time": "$.time"
      }
      ```

   c. En Plantilla, introduzca la plantilla para transformar los datos del evento de origen en el CloudEvents formato.

      En la siguiente plantilla, no `region` es estrictamente obligatorio, ya que el `region` atributo de la ruta de entrada es un atributo de extensión de la CloudEvents especificación.

      ```
      {
        "specversion":"1.0",
        "id":<id>,
        "source":<source>,
        "type":<detail-type>,
        "time":<time>,
        "region":<region>,
        "data":<detail>
      ```

```
}
```

5. Complete la creación de la regla siguiendo los pasos del procedimiento.

Socios de destinos de la API

Utilice la información proporcionada por los siguientes AWS socios para configurar un destino y una conexión de API para su servicio o aplicación.

Observabilidad en la nube de Cisco

URL del punto de conexión de invocación de destino de la API:

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

Tipos de autorización compatibles:

Credenciales del cliente OAuth

Los tokens OAuth se actualizan cuando se devuelve una respuesta 401 o 407

Se requieren parámetros de autorización adicionales:

ID de cliente y secreto de AppDynamics cliente de Cisco

Punto final de OAuth:

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

Los siguientes parámetros del par clave/valor de OAuth:

Tipo	Clave	Valor
Campo corporal	grant_type	client_credentials
Encabezado	Contenido-Tipo	aplicación/x-www-form-urlencoded; charset=utf-8

Documentación AppDynamics de Cisco:

AWS ingestión de eventos

Operaciones de la API de uso frecuente;

No aplicable

Información adicional:

Al seleccionar Cisco en el menú desplegable AppDynamics de destinos para socios, se rellena previamente la información de OAuth necesaria, incluidos los pares de encabezado y cuerpo clave/valor necesarios para las llamadas a la API.

Para obtener información adicional, consulte la recopilación de AWS eventos en la documentación de Cisco. AppDynamics

Confluente

URL del punto de conexión de invocación de destino de la API:

Por lo general, el siguiente formato:

`https://random-id.region.aws.confluent.cloud:443/kafka/v3/clusters/cluster-id/topics/topic-name/records`

Para obtener más información, consulte Buscar la dirección del punto final de REST y el ID del clúster en la documentación de Confluent.

Tipos de autorización compatibles:

Basic

Se requieren parámetros de autorización adicionales:

No aplicable

Documentación de Confluent:

Registros de producción

Proxy REST confluente para Apache Kafka

Operaciones de la API de uso frecuente;

`POST`

Información adicional:

Para transformar los datos del evento en un mensaje que el punto final pueda procesar, cree un transformador de entrada de destino.

- Para generar un registro sin especificar una clave de partición de Kafka, utilice la siguiente plantilla para su transformador de entrada. No se requiere ninguna ruta de entrada.

```
{
  "value":{
      "type":"JSON",
      "data":aws.events.event.json
  },
}
```

- Para generar un registro utilizando un campo de datos de eventos como clave de partición de Kafka, siga el ejemplo de ruta de entrada y plantilla que se muestra a continuación. En este ejemplo se define la ruta de entrada del `orderId` campo y, a continuación, se especifica ese campo como clave de partición.

  En primer lugar, defina la ruta de entrada para el campo de datos del evento:

```
{
  "orderId":"$.detail.orderId"
}
```

  A continuación, utilice la plantilla del transformador de entrada para especificar el campo de datos como clave de partición:

```
{
  "value":{
    "type":"JSON",
    "data":aws.events.event.json
  },
  "key":{
    "data":"<orderId>",
    "type":"STRING"
  }
}
```

Coralogix

URL del punto de conexión de invocación de destino de la API

Para obtener una lista completa de puntos de conexión, consulte Referencia de la API de Coralogix.

Tipos de autorización compatibles

Clave de API

Se requieren parámetros de autorización adicionales

Encabezado `"x-amz-event-bridge-access-key"`, el valor es la clave de API Coralogix

Documentación de Coralogix

EventBridgeAutenticación de Amazon

Operaciones de la API de uso frecuente

EE. UU.: https://ingress.coralogix.us/aws/event-bridge

Singapur: https://ingress.coralogixsg.com/aws/event-bridge

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

Estocolmo: https://ingress.eu2.coralogix.com/aws/event-bridge

India: https://ingress.coralogix.in/aws/event-bridge

Información adicional

Los eventos se almacenan como entradas de registro con `applicationName=[AWS Account]` y `subsystemName=[event.source]`.

Datadog

URL del punto de conexión de invocación de destino de la API

Para obtener una lista completa de puntos de conexión, consulte Referencia de la API de Datadog.

Tipos de autorización compatibles

Clave de API

Se requieren parámetros de autorización adicionales

Ninguna

Documentación de Datadog

Autenticación

Operaciones de la API de uso frecuente

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

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

Información adicional

Las URL de los puntos de conexión varían en función de la ubicación de la organización de Datadog. Para ver la URL correcta de la organización, consulte la documentación.

Freshworks

URL del punto de conexión de invocación de destino de la API

Para obtener una lista de puntos de conexión, consulte https://developers.freshworks.com/documentation/

Tipos de autorización compatibles

Basic, clave de API

Se requieren parámetros de autorización adicionales

No aplicable

Documentación de Freshworks

Autenticación

Operaciones de la API de uso frecuente

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

Información adicional

Ninguna

MongoDB

URL del punto de conexión de invocación de destino de la API

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

Tipos de autorización compatibles

Clave de API

Correo electrónico/Contraseña

Autenticación JWT personalizada

Se requieren parámetros de autorización adicionales

Ninguna

Documentación de MongoDB

API de datos de Atlas

Puntos de conexión

Puntos de conexión HTTPS personalizados

Autenticación

Operaciones de la API de uso frecuente

Ninguna

Información adicional

Ninguna

New Relic

URL del punto de conexión de invocación de destino de la API

Para obtener más información, consulte nuestros centros de datos regionales de la UE y EE. UU..

Eventos

EE. UU: https://insights-collector.newrelic.com/v1/accounts/*YOUR_NEW_RELIC_ACCOUNT_ID*/events

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

Métricas

EE. UU: https://metric-api.newrelic.com/metric/v1

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

Registros

EE. UU.: https://log-api.newrelic.com/log/v1

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

Rastros

EE. UU.: https://trace-api.newrelic.com/trace/v1

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

Tipos de autorización compatibles

Clave de API

Documentación de New Relic

API de métricas

API de eventos

API de registros

API de rastros

Operaciones de la API de uso frecuente

API de métricas

API de eventos

API de registros

API de rastros

Información adicional

Límites de la API de métricas

Límites de la API de eventos

Límites de la API de registros

Límites de la API de rastros

Operata

URL del punto de conexión de invocación de destino de la API:

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

Tipos de autorización compatibles:

Basic

Se requieren parámetros de autorización adicionales:

Ninguna

Documentación de Operata:

¿Cómo puedo crear, ver, cambiar y revocar los tokens de la API?

AWS Integración de Operata mediante Amazon EventBridge Scheduler Pipes

Operaciones de la API de uso frecuente;

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

Información adicional:

El `username` es el ID del grupo Operata y la contraseña es el token de la API.

Salesforce

URL del punto de conexión de invocación de destino de la API

Objeto: myDomainName https://.my.salesforce.com/services/data/ VersionNumber /sobjects*//* *SobjectEndpoint*

Eventos de plataforma personalizados: myDomainName https://.my.salesforce.com/services/data/ VersionNumber */sobjects/ /* customPlatformEndpoint*

Para obtener una lista completa de puntos de conexión, consulte Referencia de la API de Salesforce.

Tipos de autorización compatibles

Credenciales del cliente OAuth

Los tokens OAUTH se actualizan cuando se devuelve una respuesta 401 o 407.

Se requieren parámetros de autorización adicionales

ID de cliente de Salesforce Connected App y secreto de cliente.

Uno de los siguientes puntos de conexión de autorización:

- Producción ***MyDomainName***: https://.my.salesforce.com. /services/oauth2/token
- Sandbox sin dominios mejorados: https://-- .my. salesforce.com/services /oauth2/token *MyDomainName SandboxName*
- Sandbox con dominios mejorados: https://-- .sandbox.my.salesforce.com/services/oauth2/token ***MyDomainName SandboxName***

El siguiente par clave/valor:

Clave	Valor
grant_type	client_credentials

Documentación de Salesforce

Guía para desarrolladores de la API de REST

Operaciones de la API de uso frecuente

Trabajo con metadatos de objeto

Trabajo con registros

Información adicional

Para ver un tutorial en el que se explica cómo usar la EventBridge consola para crear una conexiónSalesforce, un destino de API y una regla a la que dirigir la informaciónSalesforce, consulte. ???

Slack

URL del punto de conexión de invocación de destino de la API

Para ver una lista de puntos de conexión y otros recursos, consulte Uso de la API Slack Web

Tipos de autorización compatibles

OAuth 2.0

Los tokens OAUTH se actualizan cuando se devuelve una respuesta 401 o 407.

Cuando crea una aplicación de Slack y la instala en el espacio de trabajo, se creará en su nombre un token de portador de OAuth que se utilizará para autenticar las llamadas desde la conexión de destino de la API.

Se requieren parámetros de autorización adicionales

No aplicable

Documentación de Slack

Configuración básica de la aplicación

Instalación con OAuth

Recuperación de mensajes

Envío de mensajes

Envío de mensajes con Webhooks entrantes

Operaciones de la API de uso frecuente

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

Información adicional

Al configurar la EventBridge regla, hay dos configuraciones que hay que destacar:

- Incluya un parámetro de encabezado que defina el tipo de contenido como “application/json;charset=utf-8”.
- Use un transformador de entrada para asignar el evento de entrada al resultado esperado para la API de Slack, es decir, asegúrese de que la carga enviada a la API de Slack tenga los pares clave/valor “canal” y “texto”.

Shopify

URL del punto de conexión de invocación de destino de la API

Para obtener una lista de puntos de conexión y otros recursos y métodos, consulte Endpoints and requests

Tipos de autorización compatibles

OAuth, clave de API

> **Note**
>
> Los tokens OAUTH se actualizan cuando se devuelve una respuesta 401 o 407.

Se requieren parámetros de autorización adicionales

No aplicable

Documentación de Shopify

Información general sobre autenticación y autorización

Operaciones de la API de uso frecuente

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

Información adicional

Crear una aplicación

Entrega de EventBridge webhook en Amazon

Access tokens for custom apps in the Shopify admin

Producto

API de administrador de Shopify

Splunk

URL del punto de conexión de invocación de destino de la API

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

Tipos de autorización compatibles

Basic, clave de API

Se requieren parámetros de autorización adicionales

Ninguna

Documentación de Splunk

Para ambos tipos de autorización, necesita un identificador de token HEC. Para obtener más información, consulte Configurar y usar el recolector de eventos HTTP en la web de Splunk.

Operaciones de la API de uso frecuente

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

Información adicional

Clave de API: al configurar el punto final EventBridge, el nombre de la clave de API es «Autorización» y el valor es el ID del token HEC de Splunk.

Básico (nombre de usuario/contraseña): al configurar el punto final para EventBridge, el nombre de usuario es «Splunk» y la contraseña es el ID del token HEC de Splunk.

Sumo Logic

URL del punto de conexión de invocación de destino de la API

Las URL de punto de conexión del origen HTTP de registros y métricas serán diferentes para cada usuario. Para obtener más información, consulte Origen HTTP de registros y métricas.

Tipos de autorización compatibles

Sumo Logic no requiere autenticación en sus orígenes HTTP porque hay una clave única integrada en la URL. Por este motivo, debe tratar a la URL como un secreto.

Al configurar el destino de la EventBridge API, se requiere un tipo de autorización. Para cumplir este requisito, seleccione la clave de API y asígnele el nombre de clave “dummy-key” y el valor de clave “dummy-value”.

Se requieren parámetros de autorización adicionales

No aplicable

Documentación de Sumo Logic

Sumo Logicya ha creado fuentes alojadas para recopilar registros y métricas de muchos AWS servicios, y puedes usar la información de su sitio web para trabajar con esas fuentes. Para obtener más información, consulte Amazon Web Services.

Si está generando eventos personalizados desde una aplicación y desea enviarlos Sumo Logic como registros o métricas, utilice los destinos de EventBridge API y los puntos de enlace Sumo Logic HTTP de origen de registros y métricas.

- Para registrarse y crear una instancia de Sumo Logic gratuita, consulte Comenzar la prueba gratuita hoy.
- Para obtener más información sobre el uso de Sumo Logic, Origen HTTP de registros y métricas.

Operaciones de la API de uso frecuente

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

Información adicional

Ninguna

TriggerMesh

URL del punto de conexión de invocación de destino de la API

Utilice la información del tema Origen de eventos para HTTP para formular la URL del punto de conexión. La URL de un punto de conexión incluye el nombre del origen del evento y el espacio de nombres de usuario en el siguiente formato:

https://*source-name*.*user-namespace*.cloud.triggermesh.io

Incluya los parámetros de autorización Basic en la solicitud al punto de conexión.

Tipos de autorización compatibles

Basic

Se requieren parámetros de autorización adicionales

Ninguna

Documentación de TriggerMesh

Origen de eventos para HTTP

Operaciones de la API de uso frecuente

No aplicable

Información adicional

Ninguna

Zendesk

URL del punto de conexión de invocación de destino de la API

https://developer.zendesk.com/rest_api/docs/support/tickets

Tipos de autorización compatibles

Basic, clave de API

Se requieren parámetros de autorización adicionales

Ninguna

Documentación de Zendesk

Seguridad y autenticación

Operaciones de la API de uso frecuente

POST https://*your_Zendesk_subdomain*/api/v2/tickets

Información adicional

Las solicitudes de API se EventBridge tienen en cuenta para descontar los límites de la API de Zendesk. Para obtener información sobre los límites de Zendesk para su plan, consulte Límites de uso.

Para proteger mejor su cuenta y sus datos, le recomendamos usar una clave de API en lugar de la autenticación básica con credenciales de inicio de sesión.

Amazon EventBridge apunta a Amazon API Gateway

Puede usar Amazon API Gateway para crear, publicar, mantener y supervisar API. Amazon EventBridge admite el envío de eventos a un punto final de API Gateway. Al especificar un punto de conexión de API Gateway como destino, cada evento enviado al destino se asigna a una solicitud enviada al punto de conexión.

> **Important**
>
> EventBridge admite el uso de puntos finales regionales y optimizados para API Gateway Edge como objetivos. Actualmente, no se admiten puntos de conexión privados. Para obtener más información sobre puntos de conexión, consulte https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-api-endpoint-types.html.

Puede utilizar un destino de API Gateway para los siguientes casos de uso:

- Para invocar una API especificada por el cliente alojada en API Gateway en función de eventos AWS o de terceros.
- Para invocar un punto de conexión de forma periódica y de forma programada.

La información del evento EventBridge JSON se envía como el cuerpo de la solicitud HTTP a tu punto final. Puede especificar los demás atributos de la solicitud en el campo `HttpParameters` del destino de la siguiente manera:

- `PathParameterValues` muestra los valores que corresponden secuencialmente a cualquier variable de ruta del ARN del punto de conexión, por ejemplo `"arn:aws:execute-api:us-east-1:112233445566:myapi/dev/POST/pets/*/*"`.

- `QueryStringParameters`representa los parámetros de la cadena de consulta que se EventBridge anexan al punto final invocado.
- `HeaderParameters` define los encabezados HTTP que se van a agregar a la solicitud.

> **Note**
>
> Por motivos de seguridad, actualmente no se permiten las siguientes claves de encabezado HTTP:
>
> - Cualquiera que tenga el prefijo `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`

Parámetros dinámicos

Al invocar un destino de API Gateway, puede añadir datos de forma dinámica a los eventos que se envían al destino. Para obtener más información, consulte the section called “Parámetros de destino”.

Reintentos de invocación

Como ocurre con todos los objetivos, EventBridge vuelve a intentar algunas invocaciones fallidas. En el caso de API Gateway, EventBridge vuelve a intentar las respuestas enviadas con un código de

estado HTTP 5xx o 429 durante un máximo de 24 horas, con retardo y fluctuación exponenciales. Después de eso, EventBridge publica una `FailedInvocations` métrica en Amazon CloudWatch. EventBridge no vuelve a intentar otros errores HTTP de 4xx.

Tiempo de espera

EventBridge regla Las solicitudes de API Gateway deben tener un tiempo de espera máximo de ejecución del cliente de 5 segundos. Si API Gateway tarda más de 5 segundos en responder, agota el EventBridge tiempo de espera de la solicitud y, a continuación, vuelve a intentarlo.

EventBridge Las solicitudes de API Gateway de Pipes tienen un tiempo de espera máximo de 29 segundos, el máximo de API Gateway.

AWS AppSync objetivos para Amazon EventBridge

AWS AppSync permite a los desarrolladores conectar sus aplicaciones y servicios a datos y eventos con API GraphQL y Pub/Sub seguras, sin servidor y de alto rendimiento. Con AWS AppSync, puede publicar actualizaciones de datos en tiempo real en sus aplicaciones con mutaciones de GraphQL. EventBridge admite la llamada a una operación de mutación de GraphQL válida para eventos coincidentes. Cuando especificas una mutación de AWS AppSync API como objetivo, AWS AppSync procesa el evento mediante una operación de mutación, que luego puede activar suscripciones vinculadas a la mutación.

> **Note**
>
> EventBridge admite las API AWS AppSync públicas de GraphQL. EventBridge actualmente no es compatible con las API AWS AppSync privadas.

Puedes usar un objetivo de la API de AWS AppSync GraphQL para los siguientes casos de uso:

- Para insertar, transformar y almacenar datos de eventos en los orígenes de datos configurados.
- Para enviar notificaciones en tiempo real a los clientes de aplicaciones conectados.

> **Note**
>
> AWS AppSync los objetivos solo admiten la llamada a las API de AWS AppSync GraphQL mediante el tipo de `AWS_IAM` autorización.

Para obtener más información sobre las API de AWS AppSync GraphQL, consulte GraphQL y su AWS AppSync arquitectura en la Guía para desarrolladores.AWS AppSync

Para especificar un AWS AppSync objetivo para una EventBridge regla mediante la consola

1. Cree o edite la regla.
2. En Destino, especifique el objetivo eligiendo servicio de AWS y, a continuación, AWS AppSync.
3. Especifique la operación de mutación que se analizará y ejecutará, junto con el conjunto de selección.

   - Elija la AWS AppSync API y, a continuación, la mutación de la API de GraphQL que desee invocar.
   - En Configurar parámetros y conjunto de selecciones, elija crear un conjunto de selección mediante una asignación de clave-valor o un transformador de entrada.

     Key-value mapping

     Para usar la asignación de clave-valor para crear el conjunto de selección:

     - Especifique las variables para los parámetros de la API. Cada variable puede ser un valor estático o una expresión de ruta JSON dinámica para la carga útil del evento.
     - En Conjunto de selección, elige las variables que desea incluir en la respuesta.

     Input transformer

     Si desea usar un transformador de entrada para crear el conjunto de selección:

     - Especifique una ruta de entrada que defina las variables que se usarán.
     - Especifique una plantilla de entrada para definir y dar formato a la información que desea que se pase al destino.

     Para obtener más información, consulte ???.
4. Para Rol de ejecución, elija si desea crear un nuevo rol o use un rol existente.
5. Complete la creación o edición de la regla.

Ejemplo: AWS AppSync objetivos para Amazon EventBridge

En el siguiente ejemplo, veremos cómo especificar un AWS AppSync objetivo para una EventBridge regla, incluida la definición de una transformación de entrada para formatear los eventos para su entrega.

Supongamos que tiene una API de AWS AppSync GraphQL`Ec2EventAPI`, definida por el siguiente esquema:

```
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"])
}
```

Los clientes de aplicaciones que utilizan esta API pueden suscribirse a la suscripción de `subscribeToEvent`, que se desencadena por la mutación de `pushEvent`.

Puedes crear una EventBridge regla con un objetivo que envíe eventos a la AppSync API a través de la `pushEvent` mutación. Cuando se invoque la mutación, cualquier cliente que esté suscrito recibirá el evento.

Para especificar esta API como el destino de una EventBridge regla, debes hacer lo siguiente:

1. Establezca el nombre de recurso de Amazon (ARN) del destino de la regla en el ARN de punto de conexión de GraphQL de la API de `Ec2EventAPI`.
2. Especifique la operación de GraphQL de mutación como parámetro de destino:

```
mutation CreatePushEvent($id: ID!, $statusCode: String, $instanceId: String) {
  pushEvent(id: $input, statusCode: $statusCode, instanceId: $instanceId) {
    id
    statusCode
    instanceId
  }
```

```
}
```

Su conjunto de selección de mutaciones debe incluir todos los campos a los que desee suscribirse en su suscripción a GraphQL.

3. Configure un transformador de entrada para especificar cómo se utilizan los datos de los eventos coincidentes en su operación.

   Supongamos que ha seleccionado el evento de ejemplo "`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."
  }
}
```

Puede definir las siguientes variables para usarlas en su plantilla, usando la ruta de entrada del transformador de entrada de destino:

```
{
  "id": "$.id",
  "statusCode": "$.detail.StatusCode",
  "EC2InstanceId": "$.detail.EC2InstanceId"
}
```

Componga la plantilla del transformador de entrada para definir las variables que se EventBridge transferirán a la operación de AWS AppSync mutación. La plantilla debe evaluarse en JSON. Dada nuestra ruta de entrada, puede crear la siguiente plantilla:

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

Conexiones para destinos de punto final HTTP

Una conexión define el método de autorización y las credenciales EventBridge que se utilizarán para conectarse a un punto final HTTP determinado. Al configurar los ajustes de autorización y crear una conexión, se crea una entrada secreta AWS Secrets Manager para almacenar de forma segura la información de autorización. También puede agregar parámetros adicionales para incluirlos en la conexión según corresponda para su destino de punto final HTTP.

Utilice las conexiones con:

- Destinos de la API

  Al crear un destino de la API, se especifica la conexión que se va a utilizar para él. Puedes elegir una conexión existente desde tu cuenta o crear una conexión al crear un destino de API.

Métodos de autorización para las conexiones

EventBridge las conexiones admiten los siguientes métodos de autorización:

- Basic
- Clave de API

  Para la autorización de claves básicas y de API, EventBridge rellena automáticamente los encabezados de autorización necesarios.
- OAuth

  Para la autorización de OAuth, EventBridge también intercambia tu ID de cliente y tu secreto por un token de acceso y, a continuación, los administra de forma segura.

  Los tokens OAUTH se actualizan cuando se devuelve una respuesta 401 o 407.

Al crear una conexión, también puede incluir el encabezado, el cuerpo y los parámetros de consulta necesarios para la autorización con un punto de conexión. Puedes usar la misma conexión para más de un punto final HTTP si la autorización del punto final es la misma.

Al crear una conexión y añadir parámetros de autorización, EventBridge crea una entrada secreta AWS Secrets Manager. El coste tanto de almacenar como de acceder al secreto de Secrets Manager está incluido con el cargo por utilizar un destino de la API. Para obtener más información sobre las prácticas recomendadas para el uso de secretos con los destinos de las API, consulta AWS::Events::ApiDestinationla Guía del CloudFormation usuario.

> **Note**
>
> Para crear o actualizar correctamente una conexión, debe usar una cuenta que tenga permiso para usar Secrets Manager. El permiso necesario está incluido en la AmazonEventBridgeFullAccess política. Se concede el mismo permiso al rol vinculado al servicio que se creó en la cuenta para la conexión.

Crear conexiones para destinos de punto final HTTP

Para crear una conexión para utilizarla con puntos de conexión HTTP mediante la consola EventBridge

1. Inicie sesión AWS con una cuenta que tenga permisos para administrar EventBridge y abrir la EventBridge consola.
2. En el panel de navegación izquierdo, seleccione Destinos de la API.

3. Desplácese hacia abajo hasta la tabla de destinos de la API y, a continuación, seleccione la pestaña Conexiones.
4. Seleccione Crear conexión.
5. En la página Crear conexión, indique un nombre de conexión para la conexión.
6. Escriba una descripción para la conexión.
7. En Tipo de autorización, seleccione el tipo de autorización que se utilizará para autorizar las conexiones con el punto de conexión HTTP especificado para el destino de la API que utiliza esta conexión. Realice una de las siguientes acciones siguientes:
   - Seleccione Basic (Nombre de usuario/Contraseña) y, a continuación, introduzca el nombre de usuario y la contraseña que se utilizarán para autorizar con el punto de conexión HTTP.
   - Seleccione Credenciales de Cliente de OAuth y, a continuación, indique el Punto de conexión de autorización, el Método HTTP, el ID de cliente y el Secreto de cliente que se van a utilizar para autorizar con el punto de conexión.

     En Parámetros Http de OAuth, añada los parámetros adicionales que desee incluir en la autorización con el punto de conexión de autorización. Seleccione un parámetro de la lista desplegable y, a continuación, introduzca una clave y un valor. Para incluir un parámetro adicional, seleccione Añadir parámetro.

     En Parámetros Http de invocación, añada parámetros adicionales para incluirlos en la solicitud de autorización. Para añadir un parámetro, seleccione un parámetro de la lista desplegable y, a continuación, introduzca una clave y un valor. Para incluir un parámetro adicional, seleccione Añadir parámetro.
   - Seleccione Clave de API y, a continuación, introduzca el nombre de la clave de API y el valor asociado que se utilizará para la autorización de la clave de API.

     En Parámetros Http de invocación, añada parámetros adicionales para incluirlos en la solicitud de autorización. Para añadir un parámetro, seleccione un parámetro de la lista desplegable y, a continuación, introduzca una clave y un valor. Para incluir un parámetro adicional, seleccione Añadir parámetro.
8. Seleccione Crear.

Edición de conexiones mediante la EventBridge consola

Puede editar las conexiones existentes.

Para editar una conexión mediante la EventBridge consola

1. Inicie sesión AWS con una cuenta que tenga permisos para administrar EventBridge y abrir la EventBridge consola.
2. En el panel de navegación izquierdo, seleccione Destinos de la API.
3. Desplácese hacia abajo hasta la tabla de destinos de la API y, a continuación, seleccione la pestaña Conexiones.
4. En la tabla Conexiones, seleccione la conexión que desee editar.
5. En la página Detalles de la conexión, seleccione Editar.
6. Actualice los valores de la conexión y, a continuación, seleccione Actualizar.

Desautorizar las conexiones mediante la consola EventBridge

Al desautorizar una conexión, se eliminan todos los parámetros de autorización. Al eliminar los parámetros de autorización, se elimina el secreto de la conexión, por lo que puede reutilizarla sin tener que crear otra nueva.

> **Note**
>
> Debe actualizar cualquier punto de enlace HTTP que utilice la conexión desautorizada para que utilice una conexión diferente para enviar correctamente las solicitudes al punto de enlace HTTP.

Para desautorizar una conexión

1. Inicie sesión AWS con una cuenta que tenga permisos para administrar EventBridge y abrir la EventBridge consola.
2. En el panel de navegación izquierdo, seleccione Destinos de la API.
3. Desplácese hacia abajo hasta la tabla de destinos de la API y, a continuación, seleccione la pestaña Conexiones.
4. En la tabla Conexiones, seleccione la conexión.
5. En la página Detalles de la conexión, seleccione Desautorizar.
6. En el cuadro de diálogo ¿Desautorizar la conexión?, indique el nombre de la conexión y, a continuación, seleccione Desautorizar.

El estado de la conexión cambia a Desautorizando hasta que se completa el proceso. A continuación, el estado cambia a Desautorizada. Ahora puede editar la conexión para añadir nuevos parámetros de autorización.

Envío y recepción de EventBridge eventos de Amazon entre AWS cuentas

Puede configurarlos EventBridge para enviar y recibir eventos entre los buses de eventos de las AWS cuentas. Al EventBridge configurar el envío o la recepción de eventos entre cuentas, puede especificar qué AWS cuentas pueden enviar o recibir eventos desde el bus de eventos de su cuenta. También puede permitir o denegar eventos de reglas específicas asociadas al bus de eventos o eventos de orígenes específicos. Para obtener más información, consulta Cómo simplificar el acceso entre cuentas con las políticas de recursos de Amazon EventBridge

> **Note**
>
> Si las utilizas AWS Organizations, puedes especificar una organización y conceder acceso a todas las cuentas de esa organización. Además, el bus de eventos de envío debe tener roles de IAM asociados al enviar eventos a otra cuenta. Para obtener más información, consulte ¿Qué es AWS Organizations? en la Guía del usuario de AWS Organizations .

> **Note**
>
> Si utiliza un plan de respuesta del Administrador de incidentes como destino, todos los planes de respuesta que se comparten con su cuenta estarán disponibles de forma predeterminada.

Puede enviar y recibir eventos entre grupos de eventos de AWS cuentas de la misma región en todas las regiones y entre cuentas de diferentes regiones, siempre que la región de destino sea una región de destino multiregional compatible.

Los pasos EventBridge para configurar el envío o la recepción de eventos desde un bus de eventos de una cuenta diferente son los siguientes:

- En la cuenta receptora, edite los permisos de un bus de eventos para permitir que determinadas AWS cuentas, una organización o todas AWS las cuentas envíen eventos a la cuenta receptora.

- En la cuenta remitente, configure una o varias reglas que tengan como destino el bus de eventos de la cuenta receptora.

  Si la cuenta del remitente hereda los permisos para enviar eventos de una AWS organización, la cuenta del remitente también debe tener una función de IAM con políticas que le permitan enviar eventos a la cuenta receptora. Si la usa AWS Management Console para crear la regla dirigida al bus de eventos de la cuenta receptora, la función se crea automáticamente. Si usa el AWS CLI, debe crear el rol manualmente.
- En la cuenta receptora, configure una o varias reglas que coincidan con eventos procedentes de la cuenta remitente.

Los eventos enviados de una cuenta a otra se cargan a la cuenta remitente como eventos personalizados. No se cobra nada a la cuenta receptora. Para obtener más información, consulta los EventBridge precios de Amazon.

Si una cuenta receptora configura una regla que envíe los eventos recibidos de una cuenta remitente a una tercera cuenta, estos eventos no se envían a la tercera cuenta.

Si tienes tres buses de eventos en la misma cuenta y configuras una regla en el primer bus de eventos para reenviar los eventos del segundo bus de eventos a un tercer bus de eventos, esos eventos no se envían al tercer bus de eventos.

El siguiente vídeo muestra el enrutamiento de eventos entre cuentas: Enrutamiento de eventos a buses de otras AWS cuentas

Otorga permisos para permitir eventos de otras AWS cuentas

Para recibir eventos desde otras organizaciones o cuentas, primero debe editar los permisos en el bus de eventos en el que desea recibir los eventos. El bus de eventos predeterminado acepta eventos de AWS servicios, otras AWS cuentas autorizadas y `PutEvents` llamadas. Los permisos para un bus de eventos se conceden o deniegan mediante una política basada en recursos adjunta al bus de eventos. En la política, puedes conceder permisos a otras AWS cuentas con el ID de la cuenta o a una AWS organización con el ID de la organización. Para obtener más información sobre los permisos del bus de eventos, incluidos ejemplos de políticas, consulte Permisos para buses de eventos de Amazon EventBridge.

> **Note**
>
> EventBridge ahora requiere que todos los nuevos destinos de bus de eventos entre cuentas agreguen funciones de IAM. Esto solo se aplica a los destinos de bus de eventos creados después del 2 de marzo de 2023. Las aplicaciones creadas sin un rol de IAM antes de esa fecha no se ven afectadas. Sin embargo, recomendamos añadir roles de IAM para permitir que los usuarios accedan a los recursos de otra cuenta, ya que así se garantizan los límites organizativos mediante políticas de control de servicios (SCP) para determinar quién puede enviar y recibir eventos de las cuentas de su organización.

> **Important**
>
> Si eliges recibir eventos de todas las AWS cuentas, asegúrate de crear reglas que coincidan únicamente con los eventos que quieres recibir de otras cuentas. Para crear reglas más seguras, asegúrese de que el patrón de eventos de cada regla contiene un campo `Account` con el ID de una o varias cuentas desde las que desea recibir eventos. Las reglas que tienen un patrón de eventos que contiene un campo Account (Cuenta) no coinciden con los eventos enviados desde cuentas que no aparecen en el campo `Account`. Para obtener más información, consulte EventBridge Eventos de Amazon.

Reglas para los eventos entre AWS cuentas

Si tu cuenta está configurada para recibir eventos de los buses de eventos de otras AWS cuentas, puedes escribir reglas que coincidan con esos eventos. Establezca el patrón de eventos de la regla para que coincida con los eventos que recibe de la otra cuenta.

A menos que especifique `account` en el patrón de eventos de una regla, cualquiera de las reglas de la cuenta, ya sean nuevas o existentes, que coincidan con los eventos que recibe de los buses de eventos de otras cuentas se activa en función de dichos eventos. Si recibe eventos de los buses de eventos de otra cuenta y desea que solamente se active una regla en ese patrón de eventos cuando se genere desde su propia cuenta, debe agregar `account` y especificar su propio ID de cuenta en el patrón de eventos de la regla.

Si configuraste tu AWS cuenta para aceptar eventos de los autobuses de eventos en todas las AWS cuentas, te recomendamos encarecidamente que las añadas `account` a todas las EventBridge reglas de tu cuenta. Esto evita que las reglas de tu cuenta se activen en eventos de AWS cuentas

desconocidas. Cuando especifique el campo `account` de la regla, puede especificar los ID de varias cuentas de AWS en dicho campo.

Para que una regla se active en un evento coincidente desde cualquier bus de eventos de la AWS cuenta a la que hayas concedido permisos, no especifiques un asterisco (*) en el `account` campo de la regla. Si lo hace, no se encontrarán coincidencias de ningún evento, porque * no aparece nunca en el campo `account` de un evento. En lugar de ello, omita el campo `account` de la regla.

Crear reglas que envíen eventos entre AWS cuentas

Especificar un bus de eventos en otra cuenta como destino forma parte de la creación de la regla.

Para crear una regla que envíe eventos a una AWS cuenta diferente mediante la consola

1. Siga los pasos que se indican en el procedimiento ???.
2. En el paso ???, cuando se le pida que seleccione un tipo de destino:
   a. Seleccione el bus de EventBridge eventos.
   b. Seleccione Bus de eventos en una cuenta o región diferente.
   c. Para Bus de eventos como destino, introduzca el ARN del bus de eventos que desee utilizar.
3. Siga los pasos del procedimiento para crear la regla.

Envío y recepción de EventBridge eventos de Amazon entre AWS regiones

Puede configurarlos EventBridge para enviar y recibir eventos entre AWS regiones. También puede permitir o denegar eventos de regiones específicas, reglas específicas asociadas al bus de eventos o eventos de orígenes específicos. Para obtener más información, consulte Introducción al enrutamiento de eventos entre regiones con Amazon EventBridge

Las siguientes regiones son regiones de destino compatibles:

- Este de EE. UU. (Norte de Virginia)
- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Norte de California)
- Oeste de EE. UU. (Oregón)
- África (Ciudad del Cabo)

- Asia Pacific (Hong Kong)
- Asia-Pacífico (Tokio)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Bombay)
- Asia-Pacífico (Hyderabad)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Yakarta)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Melbourne)
- Canadá (centro)
- Oeste de Canadá (Calgary
- Europa (Fráncfort)
- Europa (España)
- Europa (Zúrich)
- Europa (Estocolmo)
- Europa (Milán)
- Europa (Irlanda)
- Europa (Londres)
- Europa (París)
- Israel (Tel Aviv)
- Medio Oriente (EAU)
- Medio Oriente (Baréin)
- América del Sur (São Paulo)

El siguiente vídeo describe el enrutamiento de eventos entre regiones mediante https://console.aws.amazon.com/events/, AWS CloudFormation, y AWS Serverless Application Model: el enrutamiento de eventos entre regiones

Crear reglas que envíen eventos a una región diferente AWS

Especificar un bus de eventos en otra AWS región como destino forma parte de la creación de la regla.

Para crear una regla que envíe eventos a otra AWS cuenta mediante la consola

1. Siga los pasos que se indican en el procedimiento ???.
2. En el paso ???, cuando se le pida que seleccione un tipo de destino:
    a. Seleccione el bus de EventBridge eventos.
    b. Seleccione Bus de eventos en una cuenta o región diferente.
    c. Para Bus de eventos como destino, introduzca el ARN del bus de eventos que desee utilizar.
3. Siga los pasos del procedimiento para crear la regla.

Envío y recepción de EventBridge eventos de Amazon entre autobuses de eventos de la misma cuenta y región

Puede configurarlo EventBridge para enviar y recibir eventos entre buses de eventos de la misma AWS cuenta y región.

Cuando se configura EventBridge para enviar o recibir eventos entre buses de eventos, se utilizan las funciones de IAM en el bus de eventos del remitente para dar permiso al bus de eventos del remitente para enviar los eventos al bus de eventos del receptor. Se utilizan políticas basadas en recursos en el bus de eventos receptor para dar permiso al bus de eventos receptor para recibir eventos del bus de eventos remitente. También puede permitir o denegar eventos de determinados buses de eventos, reglas específicas asociadas al bus de eventos o eventos de orígenes específicos. Para obtener más información sobre los permisos del bus de eventos, incluidos ejemplos de políticas, consulte Permisos para buses de eventos de Amazon EventBridge.

Los pasos EventBridge para configurar el envío o la recepción de eventos entre los buses de eventos de su cuenta incluyen los siguientes:

- Para usar un rol de IAM existente, debe conceder bien los permisos del bus de eventos remitente al bus de eventos receptor o bien los permisos del bus de eventos receptor al bus de eventos remitente.

- En la bus de eventos remitente, configure una o varias reglas que tengan como destino el bus de eventos receptor y cree un rol de IAM. Para ver un ejemplo de la política que se debe adjuntar al rol, consulte ???.
- En el bus de eventos receptor, edite los permisos para permitir que los eventos se transfieran desde el otro bus de eventos.
- En el evento receptor, configure una o varias reglas que coincidan con eventos procedentes del bus de eventos remitente.

> **Note**
>
> EventBridge no puede enrutar los eventos recibidos de un bus de eventos remitente a un tercer bus de eventos.

Los eventos enviados de un bus de eventos a otro se cobran como eventos personalizados. Para obtener más información, consulte Precios de Amazon EventBridge.

Crear reglas que envíen eventos a un bus de eventos diferente de la misma AWS cuenta y región

Para enviar eventos a otro bus de eventos, debe crear una regla con un bus de eventos como destino. La especificación de un bus de eventos en la misma AWS cuenta y región que el objetivo forma parte de la creación de la regla.

Para crear una regla que envíe eventos a un bus de eventos diferente en la misma AWS cuenta y región mediante la consola

1. Siga los pasos que se indican en el procedimiento ???.
2. En el paso ???, cuando se le pida que seleccione un tipo de destino:
   a. Seleccione el bus de EventBridge eventos.
   b. Selecciona Event Bus en la misma AWS cuenta y región.
   c. En Bus de eventos como destino, seleccione un bus de eventos en la lista desplegable.
3. Siga los pasos del procedimiento para crear la regla.

Transformación EventBridge de entradas de Amazon

Puedes personalizar el texto de un evento antes de EventBridge pasar la información al objetivo de una regla. Con el transformador de entrada de la consola o la API, se definen variables que utilizan la ruta JSON para hacer referencia a los valores del origen del evento original. El evento transformado se envía a un destino en lugar del evento original. Sin embargo, los parámetros de ruta dinámicos deben hacer referencia al evento original, no al evento transformado. Puede definir hasta 100 variables asignando a cada una un valor desde la entrada. A continuación, puede utilizar esas variables en la Plantilla de entrada como **<*`nombre-variable`*>**.

Para ver un tutorial sobre el uso del transformador de entrada, consulte ???.

> **Note**
>
> EventBridge no admite toda la sintaxis de JSON Path y la evalúa en tiempo de ejecución. La sintaxis admitida incluye:
>
> - notación de puntos (por ejemplo, `$.detail`)
> - guiones
> - guiones bajos
> - Caracteres alfanuméricos
> - índices de matrices
> - caracteres comodín (*)

En este tema:

Variables predefinidas

Hay variables predefinidas que puede usar sin definir una ruta JSON. Estas variables están reservadas y no se pueden crear variables con estos nombres:

- `aws.events.rule-arn`— El nombre del recurso de Amazon (ARN) de la EventBridge regla.
- `aws.events.rule-name`— El nombre de la EventBridge regla.
- `aws.events.event.ingestion-time`— La hora a la que se recibió el evento EventBridge. Se trata de una marca de tiempo ISO 8601. Esta variable la genera EventBridge y no se puede sobrescribir.
- `aws.events.event` — La carga del evento original en formato JSON (sin el campo `detail`). Solo se puede usar como valor para un campo JSON, ya que su contenido no está oculto.
- `aws.events.event.json` — La carga del evento original en formato JSON (con el campo `detail`). Solo se puede usar como valor para un campo JSON, ya que su contenido no está oculto.

Ejemplos de transformación de entradas

A continuación se muestra un evento de Amazon EC2 de ejemplo.

```
{
  "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"
  }
}
```

Cuando defina una regla en la consola, seleccione la opción Transformador de entrada en Configurar entrada. Esta opción muestra dos cuadros de texto: Ruta de entrada y Plantilla de entrada.

El cuadro de texto Ruta de entrada se utiliza para definir variables. Utilice la ruta JSON para hacer referencia a los elementos en su evento y almacenar esos valores en variables. Por ejemplo, puede crear una ruta de entrada para hacer referencia a valores en el evento de ejemplo escribiendo lo siguiente en el primer cuadro de texto. También puede usar paréntesis e índices para obtener elementos de las matrices.

> **Note**
>
> EventBridge reemplaza los transformadores de entrada en tiempo de ejecución para garantizar una salida JSON válida. Por este motivo, debe colocar comillas alrededor de las variables que hagan referencia a los parámetros de ruta JSON, pero no alrededor de las variables que hagan referencia a objetos o matrices JSON.

```
{
  "timestamp" : "$.time",
  "instance" : "$.detail.instance-id",
  "state" : "$.detail.state",
  "resource" : "$.resources[0]"
}
```

Esto define cuatro variables: `<timestamp>`, `<instance>`, `<state>` y `<resource>`. Puede hacer referencia a estas variables al crear su plantilla de entrada.

La plantilla de entrada es una plantilla para la información que desea pasar a su destino. Puede crear una plantilla que pase una cadena o JSON al destino. Con el evento anterior y la ruta de entrada, los siguientes ejemplos de plantilla de entrada transformarán el evento en la salida de ejemplo antes de enrutarlo a un destino.

Descripción	Plantilla	Salida
Cadena simple	`"instance <instance> is in <state>"`	`"instance i-0123456789 is in RUNNING"`

Descripción	Plantilla	Salida
Cadena con comillas ocultas	`"instance \"<instance> \" is in <state>"`	`"instance \"i-01234 56789\" is in RUNNING"` Tenga en cuenta que este es el comportamiento de la EventBridge consola. La AWS CLI aplica barras diagonale s como carácter de escape y el resultado es `"instance "i-0123456789" is in RUNNING"`.
JSON simple	`{ "instance" : <instance>, "state": <state> }`	`{ "instance" : "i-0123456789", "state": "RUNNING" }`
JSON con cadenas y variables	`{ "instance" : <instance >, "state": "<state>", "instanceStatus": "instance \"<instance> \" is in <state>" }`	`{ "instance" : "i-012345 6789", "state": "RUNNING", "instanceStatus": "instance \"i-01234 56789\" is in RUNNING" }`

Descripción	Plantilla	Salida
JSON con una mezcla de variables e información estática	`{ "instance" : <instance>, "state": [ 9, <state>, true ], "Transformed" : "Yes" }`	`{ "instance" : "i-0123456789", "state": [ 9, "RUNNING", true ], "Transformed" : "Yes" }`
Inclusión de variables reservadas en 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 } }`
Inclusión de variables reservadas en una cadena	`"<aws.events.rule-name> triggered"`	`"example triggered"`
Grupo de CloudWatch registros de Amazon	`{ "timestamp" : <timestamp>, "message": "instance \"<instance>\" is in <state>" }`	`{ "timestamp" : 2015-11-11T21:29:54Z, "message": "instance "i-0123456789" is in RUNNING }`

Transformar la entrada mediante la EventBridge API

Para obtener información sobre el uso de la EventBridge API para transformar la entrada, consulte Usar un transformador de entrada para extraer datos de un evento e ingresarlos en el destino.

Transformar la entrada mediante AWS CloudFormation

Para obtener información sobre el uso AWS CloudFormation para transformar la entrada, consulte AWS::Events::Rule InputTransformer.

Problemas comunes con la transformación de entradas

Estos son algunos de los problemas más comunes al transformar la entrada en EventBridge:

- Para cadenas, se requieren comillas.
- No hay validación al crear la ruta JSON para la plantilla.
- Si especifica una variable que coincida con una ruta JSON que no existe en el evento, dicha variable no se crea ni aparece en la salida.
- Las propiedades JSON como `aws.events.event.json` solo se pueden usar como el valor de un campo JSON, no en línea en otras cadenas.
- EventBridge no escapa a los valores extraídos por la ruta de entrada al rellenar la plantilla de entrada de un objetivo.
- Si una ruta JSON hace referencia a un objeto o matriz JSON, pero se hace referencia a la variable en una cadena, EventBridge elimina las comillas internas para garantizar que la cadena sea válida. Por ejemplo, en el caso de una variable `<detail>` apuntada a «El detalle es<detail>»`$.detail`, se EventBridge eliminarían las comillas del objeto.

  Por lo tanto, si quiere generar un objeto JSON basado en una única variable de ruta JSON, debe colocarlo como clave. En este ejemplo, `{"detail": <detail>}`.
- No se requieren comillas para las variables que representan cadenas. Están permitidas, pero añaden comillas EventBridge automáticamente a los valores de las variables de cadena durante la transformación, para garantizar que el resultado de la transformación sea un JSON válido. EventBridge no añade comillas a las variables que representan objetos o matrices de JSON. No añada comillas a las variables que representan objetos o matrices JSON.

  Por ejemplo, la siguiente plantilla de entrada incluye variables que representan tanto cadenas como objetos JSON:

```
{
  "ruleArn" : <aws.events.rule-arn>,
  "ruleName" : <aws.events.rule-name>,
  "originalEvent" : <aws.events.event.json>
}
```

El resultado es un formato JSON válido con las comillas adecuadas:

```
{
  "ruleArn" : "arn:aws:events:us-east-2:123456789012:rule/example",
  "ruleName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

- Para la salida de texto (que no sea JSON) como cadenas de varias líneas, escribe cada línea independiente de la plantilla de entrada entre comillas dobles.

  Por ejemplo, si comparas los eventos de Amazon Inspector Finding con el siguiente patrón de eventos:

  ```
  {
    "detail": {
      "severity": ["HIGH"],
      "status": ["ACTIVE"]
    },
    "detail-type": ["Inspector2 Finding"],
    "source": ["inspector2"]
  }
  ```

  Y usando la siguiente ruta de entrada:

  ```
  {
    "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"
}
```

Puede usar la siguiente plantilla de entrada para generar una salida de cadena multilínea:

```
"<severity> severity finding <title>"
"Description: <description>"
"ARN: \"<arn>\""
"Type: <type>"
"AWS Account: <account>"
"Region: <region>"
"EC2 Instance: <instance>"
"Platform: <platform>"
"AMI: <ami>"
```

Configuración de un transformador de entrada como parte de la creación de una regla

Como parte de la creación de una regla, puede especificar un transformador de entrada que se utilizará EventBridge para procesar los eventos coincidentes antes de enviarlos al destino especificado. Puede configurar transformadores de entrada para destinos que sean AWS servicios o destinos de API.

Para crear un transformador de entrada de destino como parte de una regla

1. Siga los pasos para crear una regla que se detallan en ???.
2. En el paso 3: Seleccionar destino(s), expanda Configuración adicional.
3. En Configurar entrada de destino, seleccione Transformador de entrada en el menú desplegable.

   Haga clic en Configurar transformador de entrada.

   EventBridge muestra el cuadro de diálogo Configurar el transformador de entrada.

4. En la sección Evento de muestra, seleccione un tipo de evento de muestra con el que desee probar su patrón de eventos. Puede elegir un AWS evento, un evento asociado o introducir su propio evento personalizado.

   AWS events

   Seleccione uno de los eventos emitidos desde Servicios de AWS compatibles.

   1. Seleccione Eventos de AWS .
   2. En Ejemplos de eventos, selecciona el AWS evento que desees. Los eventos se organizan por AWS servicio.

      Al seleccionar un evento, EventBridge rellena el evento de muestra.

      Por ejemplo, si elige S3 Object Created, EventBridge muestra un ejemplo de evento S3 Object Created.
   3. (Opcional) También puede seleccionar Copiar para copiar el evento de muestra en el portapapeles de tu dispositivo.

   Partner events

   Seleccione entre los eventos emitidos por servicios de terceros compatibles EventBridge, como Salesforce.

   1. Seleccione los eventos de los EventBridge socios.
   2. En Eventos de ejemplo, seleccione el evento de socio deseado. Los eventos se organizan por socio.

      Al seleccionar un evento, EventBridge rellena el evento de muestra.
   3. (Opcional) También puede seleccionar Copiar para copiar el evento de muestra en el portapapeles de tu dispositivo.

   Enter your own

   Introduzca su propio evento en texto JSON.

   1. Seleccione Introducir el suyo.

2. EventBridge rellena el evento de muestra con una plantilla de los atributos de evento necesarios.
3. Edite el evento de muestra y añádalo según desee. El evento de muestra debe tener un formato JSON válido.
4. (Opcional) También puede elegir una de las siguientes opciones:
   - Copiar: copia el evento de muestra en el portapapeles de su dispositivo.
   - Prettify: facilita la lectura del texto JSON al añadir saltos de línea, tabulaciones y espacios.

5. (Opcional) Amplíe la sección Ejemplos de rutas de entrada, plantillas y salidas para ver ejemplos de:

   - Cómo se utilizan las rutas JSON para definir las variables que representan datos de eventos
   - Cómo se pueden usar esas variables en una plantilla de transformador de entrada
   - La salida resultante que se EventBridge envía al objetivo

   Para ver ejemplos más detallados de transformaciones de entradas, consulte ???.

6. En la sección Transformador de entrada de destino, defina las variables que desee utilizar en la plantilla de entrada.

   Variables que utilizan la ruta JSON para hacer referencia a valores en el origen del evento original. A continuación, puede hacer referencia a esas variables en la plantilla de entrada para incluir los datos del evento de origen original en el evento transformado que EventBridge pasa al destino. Puede definir hasta 100 variables. El transformador de entrada debe tener un formato JSON válido.

   Por ejemplo, supongamos que ha elegido el AWS evento S3 Object Created como su evento de muestra para este transformador de entrada. A continuación, podría definir las siguientes variables para usarlas en su plantilla:

```
{
  "requester": "$.detail.requester",
  "key": "$.detail.object.key",
  "bucket": "$.detail.bucket.name"
}
```

(Opcional) También puede elegir Copiar para copiar el transformador de entrada al portapapeles de su dispositivo.

7. En la sección Plantilla, redacte la plantilla que desee usar para determinar qué EventBridge pasará al objetivo.

   Puede usar formato JSON, cadenas, información estática, variables que haya definido y variables reservadas. Para ver ejemplos más detallados de transformaciones de entradas, consulte ???.

   Suponga, por ejemplo, que ha definido las variables en el ejemplo anterior. A continuación, podría crear la siguiente plantilla, que haga referencia a esas variables, así como a las variables reservadas y a la información estática.

   ```
   {
       "message": "<requester> has created the object \"<key>\" in the bucket
    \"<bucket>\"",
       "RuleName": <aws.events.rule-name>,
       "ruleArn" : <aws.events.rule-arn>,
       "Transformed": "Yes"
   }
   ```

   (Opcional) También puede elegir Copiar para copiar la plantilla al portapapeles de su dispositivo.

8. Para probar su plantilla, seleccione Generar salida.

   EventBridge procesa el evento de muestra en función de la plantilla de entrada y muestra la salida transformada generada en Salida. Esta es la información que EventBridge se transferirá al destino en lugar del evento de origen original.

   El resultado generado para la plantilla de entrada de ejemplo descrita anteriormente sería el siguiente:

   ```
   {
       "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"
   }
   ```

(Opcional) También puede elegir Copiar para copiar la salida generada al portapapeles de su dispositivo.

9. Seleccione Confirmar.
10. Siga el resto de los pasos para crear una regla, tal como se detalla en ???.

Probar un transformador de entrada objetivo con el EventBridge Sandbox

Puedes usar transformadores de entrada para personalizar el texto de un evento antes de EventBridge pasar la información al objetivo de una regla.

La configuración de un transformador de entrada suele formar parte de un proceso más amplio que consiste en especificar un destino mientras se crea una regla nueva o se edita una existente. Sin embargo EventBridge, al usar Sandbox in, puede configurar rápidamente un transformador de entrada y usar un evento de muestra para confirmar que está obteniendo el resultado deseado, sin tener que crear o editar una regla.

Para obtener más información sobre transformaciones de entradas, consulte ???.

Para probar un transformador de entrada de destino

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En Recursos para desarrolladores, seleccione Entorno aislado y, en la página Entorno aislado, seleccione la pestaña Transformador de entrada de destino.
3. En la sección Evento de muestra, seleccione un tipo de evento de muestra con el que desee probar su patrón de eventos. Puedes elegir un AWS evento, un evento asociado o introducir tu propio evento personalizado.

   AWS events

   Seleccione uno de los eventos emitidos desde Servicios de AWS compatibles.

   1. Seleccione Eventos de AWS .
   2. En Ejemplos de eventos, selecciona el AWS evento que desees. Los eventos se organizan por AWS servicio.

      Al seleccionar un evento, EventBridge rellena el evento de muestra.

Por ejemplo, si elige S3 Object Created, EventBridge muestra un ejemplo de evento S3 Object Created.

3. (Opcional) También puede seleccionar Copiar para copiar el evento de muestra en el portapapeles de tu dispositivo.

Partner events

Seleccione entre los eventos emitidos por servicios de terceros compatibles EventBridge, como Salesforce.

1. Seleccione los eventos de los EventBridge socios.
2. En Eventos de ejemplo, seleccione el evento de socio deseado. Los eventos se organizan por socio.

   Al seleccionar un evento, EventBridge rellena el evento de muestra.
3. (Opcional) También puede seleccionar Copiar para copiar el evento de muestra en el portapapeles de tu dispositivo.

Enter your own

Introduzca su propio evento en texto JSON.

1. Seleccione Introducir el suyo.
2. EventBridge rellena el evento de muestra con una plantilla de los atributos de evento necesarios.
3. Edite el evento de muestra y añádalo según desee. El evento de muestra debe tener un formato JSON válido.
4. (Opcional) También puede elegir una de las siguientes opciones:
   - Copiar: copia el evento de muestra en el portapapeles de su dispositivo.
   - Prettify: facilita la lectura del texto JSON al añadir saltos de línea, tabulaciones y espacios.

4. (Opcional) Amplíe la sección Ejemplos de rutas de entrada, plantillas y salidas para ver ejemplos de:

- Cómo se utilizan las rutas JSON para definir las variables que representan datos de eventos

- Cómo se pueden usar esas variables en una plantilla de transformador de entrada
- La salida resultante que se EventBridge envía al objetivo

Para ver ejemplos más detallados de transformaciones de entradas, consulte ???.

5. En la sección Transformador de entrada de destino, defina las variables que desee utilizar en la plantilla de entrada.

   Variables que utilizan la ruta JSON para hacer referencia a valores en el origen del evento original. A continuación, puede hacer referencia a esas variables en la plantilla de entrada para incluir los datos del evento de origen original en el evento transformado que EventBridge pasa al destino. Puede definir hasta 100 variables. El transformador de entrada debe tener un formato JSON válido.

   Por ejemplo, supongamos que ha elegido el AWS evento S3 Object Created como su evento de muestra para este transformador de entrada. A continuación, podría definir las siguientes variables para usarlas en su plantilla:

   ```
   {
     "requester": "$.detail.requester",
     "key": "$.detail.object.key",
     "bucket": "$.detail.bucket.name"
   }
   ```

   (Opcional) También puede elegir Copiar para copiar el transformador de entrada al portapapeles de su dispositivo.

6. En la sección Plantilla, redacte la plantilla que desee usar para determinar qué EventBridge pasará al objetivo.

   Puede usar formato JSON, cadenas, información estática, variables que haya definido y variables reservadas. Para ver ejemplos más detallados de transformaciones de entradas, consulte ???.

   Suponga, por ejemplo, que ha definido las variables en el ejemplo anterior. A continuación, podría crear la siguiente plantilla, que haga referencia a esas variables, así como a las variables reservadas y a la información estática.

   ```
   {
   ```

```
    "message": "<requester> has created the object \"<key>\" in the bucket
 \"<bucket>\"",
    "RuleName": <aws.events.rule-name>,
    "ruleArn" : <aws.events.rule-arn>,
    "Transformed": "Yes"
}
```

(Opcional) También puede elegir Copiar para copiar la plantilla al portapapeles de su dispositivo.

7. Para probar su plantilla, seleccione Generar salida.

   EventBridge procesa el evento de muestra en función de la plantilla de entrada y muestra la salida transformada generada en Salida. Esta es la información que EventBridge se transferirá al destino en lugar del evento de origen original.

   El resultado generado para la plantilla de entrada de ejemplo descrita anteriormente sería el siguiente:

```
{
    "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"
}
```

   (Opcional) También puede elegir Copiar para copiar la salida generada al portapapeles de su dispositivo.

Archivo y reproducción de Amazon EventBridge

En EventBridge, puede crear un archivo de eventos para reproducirlos fácilmente en otro momento. Por ejemplo, puede que quiera reproducir eventos para recuperarse de errores o para validar una nueva funcionalidad de la aplicación.

> **Note**
>
> Es posible que se produzca un retraso entre la publicación de un evento en un bus de eventos y su llegada al archivo. Le recomendamos que retrase la reproducción de los eventos archivados durante 10 minutos para asegurarse de que se reproduzcan todos los eventos.

En el siguiente vídeo, se muestra el uso de la función de archivo y reproducción: Creación de archivos y reproducciones

Temas

- Archivar eventos de Amazon EventBridge
- Reproducción de eventos de Amazon EventBridge archivados

Archivar eventos de Amazon EventBridge

Al crear un archivo en EventBridge, puede determinar qué eventos se envían al archivo especificando un patrón de eventos. EventBridge envía los eventos que coinciden con el patrón de eventos al archivo. También se establece el período de retención para almacenar los eventos en el archivo antes de descartarlos.

De forma predeterminada, EventBridge cifra los datos de los eventos de un archivo mediante el estándar de cifrado avanzado (AES-256) de 256 bits con una AWS CMK propia, lo que ayuda a proteger los datos contra el acceso no autorizado.

> **Note**
>
> Los `SizeBytes` valores `EventCount` y de la `DescribeArchive`operación tienen un período de conciliación de 24 horas. Por lo tanto, es posible que cualquier evento que haya caducado recientemente o que se haya archivado recientemente no se refleje inmediatamente en estos valores.

Para crear un archivo de todos los eventos

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Archivos.
3. Elija Crear archivo.
4. En Detalles del archivo, introduzca un nombre para el archivo. El nombre debe ser único para la cuenta de la región seleccionada.

   No se puede cambiar el nombre después de crear el archivo.
5. (Opcional) Escriba una descripción del archivo.
6. En Origen, seleccione el bus de eventos que emite los eventos para enviarlos al archivo.
7. En Período de retención, realice una de las acciones siguientes:
   - Elija Indefinido para retener los eventos en el archivo y no eliminarlos nunca.
   - Indique el número de días durante el que se han de retener los eventos. Tras el número de días especificado, EventBridge borra los eventos del archivo.
8. Elija Siguiente.
9. En Patrón de evento, seleccione Sin filtrado de eventos.

10. Elija Crear archivo.

Para crear un archivo con un patrón de evento

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Archivos.
3. Elija Crear archivo.
4. En Detalles del archivo, introduzca un nombre para el archivo. El nombre debe ser único para la cuenta de la región seleccionada.

   No se puede cambiar el nombre después de crear el archivo.
5. (Opcional) Escriba una descripción del archivo.
6. En Origen, seleccione el bus de eventos que emite los eventos para enviarlos al archivo.
7. En Período de retención, realice una de las acciones siguientes:
   - Elija Indefinido* para retener los eventos en el archivo y no eliminarlos nunca.
   - Indique el número de días durante el que se han de retener los eventos. Tras el número de días especificado, EventBridge borra los eventos del archivo.
8. Elija Siguiente.
9. En Patrón de eventos, seleccione Filtrar eventos por coincidencia de patrones de eventos.
10. Realice una de las siguientes acciones siguientes:
    - Seleccione Creador de patrones y, a continuación, elija el proveedor de servicios. Si elige AWS, seleccione también el nombre del servicio de AWS y el tipo de evento que desee utilizar en el patrón.
    - Seleccione Editor JSON para crear un patrón manualmente. También puede copiar el patrón de una regla y, a continuación, pegarlo en el editor JSON.
11. Elija Crear archivo.

Para confirmar que los eventos se han enviado correctamente al archivo, puedes utilizar el `DescribeArchive`funcionamiento de la EventBridge API para comprobar si `EventCount` refleja el número de eventos del archivo. Si es 0, no hay eventos en el archivo.

Reproducción de eventos de Amazon EventBridge archivados

Después de crear un archivo, puede reproducir los eventos desde el archivo. Por ejemplo, si actualiza una aplicación con una funcionalidad adicional, puede reproducir los eventos históricos para que estos se vuelvan a procesar y así mantener la coherencia de la aplicación. También puede utilizar un archivo para reproducir eventos con una nueva funcionalidad. Al reproducir los eventos, puede especificar el archivo desde el que se van a reproducir, la hora de inicio y finalización de la reproducción, el bus de eventos o una o más reglas para reproducir los eventos.

Los eventos no se reproducen necesariamente en el mismo orden en que se agregaron al archivo. Una reproducción procesa los eventos para que reproduzcan en función de la hora del evento y los reproduce en intervalos de un minuto. Si especifica una hora de inicio y una hora de finalización del evento que abarquen un intervalo de 20 minutos, los eventos se reproducirán desde el primer minuto de ese intervalo de 20 minutos. A continuación, se reproducen los eventos del segundo minuto. Puede usar la operación `DescribeReplay` de la API de EventBridge para determinar el progreso de una reproducción. `EventLastReplayedTime` devuelve la marca de tiempo del último evento reproducido.

Los eventos se reproducen en función del límite de transacciones `PutEvents` por segundo de la cuenta AWS, pero de forma independiente. Puede solicitar un aumento del límite de PutEvents. Para obtener más información, consulte Amazon EventBridge Quotas.

> **Note**
>
> Puede tener un máximo de 10 reproducciones simultáneas activas por cuenta por región de AWS.

Para iniciar la reproducción de un evento

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Reproducciones.
3. Elija Iniciar reproducción nueva.
4. Indique un Nombre para la reproducción y, si lo desea, introduzca una Descripción.
5. En Origen, seleccione el archivo desde el que desea reproducir los eventos.
6. Para destino, solo puede reproducir los eventos en el mismo bus de eventos que los emitió.
7. En Especificar reglas, realice una de las siguientes acciones:

- Seleccione Todas las reglas para reproducir eventos para todas las reglas.
- Elija Especificar reglas y, a continuación, seleccione la regla o reglas para reproducir los eventos.

8. En Periodo de reproducción, especifique la fecha, la hora y la zona horaria para la hora de inicio y la hora de finalización. Solo se reproducen los eventos que se produjeron entre la hora de inicio y la hora de finalización.
9. Elija Iniciar la reproducción.

Cuando se reproducen los eventos del archivo, el estado de la reproducción es Completado.

Si inicia una reproducción y desea interrumpirla, puede cancelarla siempre que el estado sea Iniciando o Ejecutando.

Para cancelar una reproducción

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Reproducciones.
3. Seleccione la reproducción que desea cancelar.
4. Elija Cancelar.

Amazon EventBridge Pipes

Amazon EventBridge Pipes conecta las fuentes con los objetivos. Los tubos están diseñados para la point-to-point integración entre las fuentes y los objetivos compatibles, y permiten realizar transformaciones y enriquecimientos avanzados. Reduce la necesidad de conocimientos especializados y código de integración al desarrollar arquitecturas basadas en eventos, lo que fomenta la coherencia de las aplicaciones de la empresa. Para configurar una canalización, elija el origen, agregue filtros opcionales, defina el enriquecimiento opcional y elija el destino de los datos del evento.

> **Note**
>
> También puede direccionar los eventos mediante buses de eventos. Los buses de eventos son ideales para many-to-many enrutar eventos entre servicios basados en eventos. Para obtener más información, consulte ???.

Cómo funcionan las tuberías EventBridge

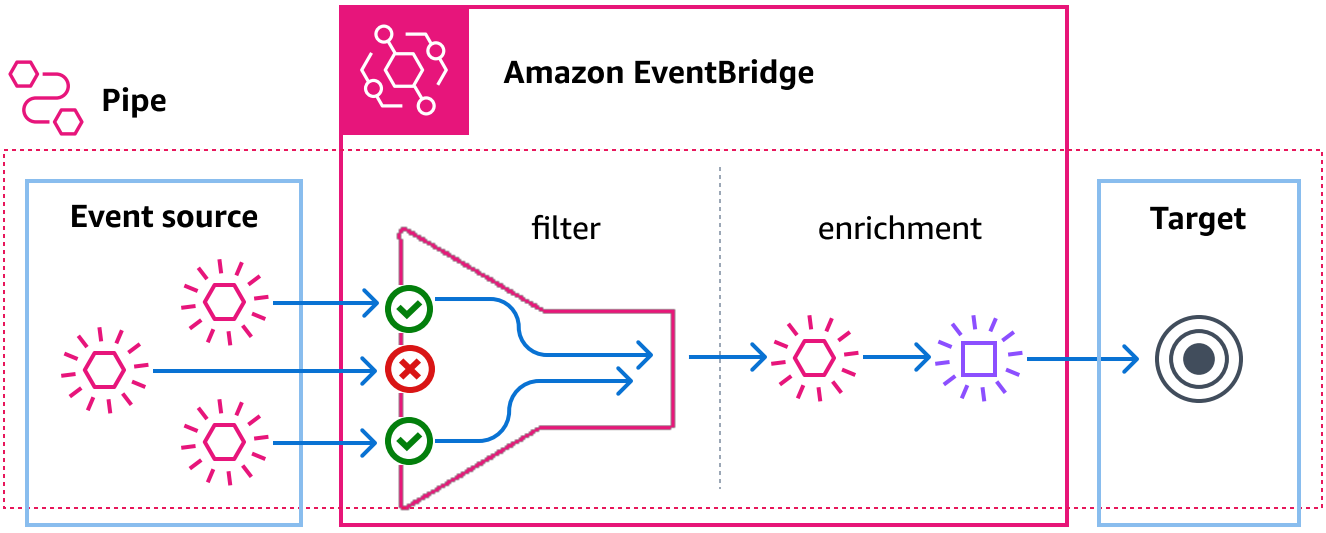
A un nivel alto, así es como funciona EventBridge Pipes:

1. Usted crea una canalización en la cuenta. Esto incluye:
   - Especificar uno de los orígenes de eventos compatibles en el que quiere que su canal reciba los eventos.
   - Si lo desea, puede configurar un filtro para que la canalización solo procese un subconjunto de los eventos que recibe del origen.
   - Si lo desea, puede configurar un paso de enriquecimiento que mejora los datos del evento antes de enviarlos al destino.
   - Especificar uno de los destinos compatibles al que quiere que su canalización envíe los eventos.
2. El origen de eventos comienza a enviar eventos a la canalización y la canalización procesa el evento antes de enviarlo al destino.
   - Si ha configurado un filtro, la canalización evalúa el evento y solo lo envía al destino si coincide con ese filtro.

     Solo se le cobrará por los eventos que coincidan con el filtro.

- Si ha configurado un enriquecimiento, la canalización lo enriquece en el evento antes de enviarlo al destino.

  Si los eventos se agrupan en lotes, el enriquecimiento mantiene el orden de los eventos en el lote.



Por ejemplo, una canalización podría usarse para crear un sistema de comercio electrónico. Supongamos que tiene una API que contiene información del cliente, como las direcciones de envío.

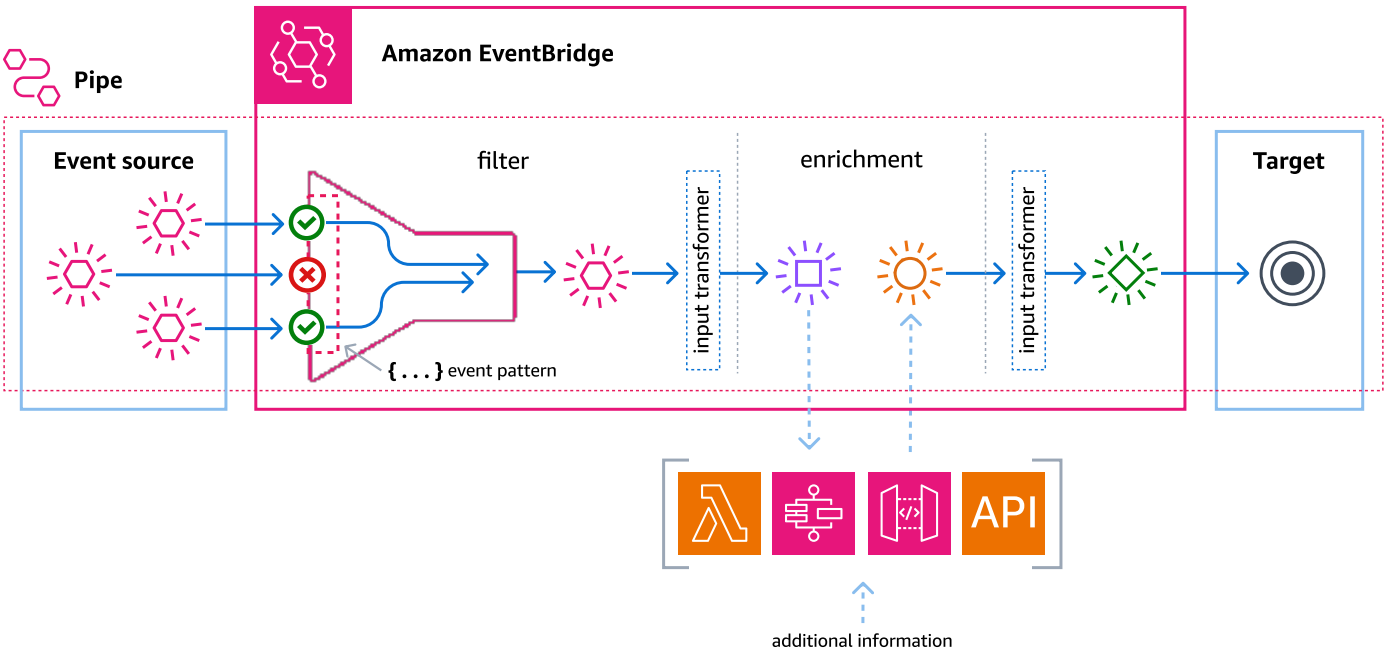
1. Entonces, puede crear una canalización con lo siguiente:
   - Una cola de mensajes recibidos por un pedido de Amazon SQS como origen del evento.
   - Un destino EventBridge de API como enriquecimiento
   - Una máquina de AWS Step Functions estados como objetivo
2. A continuación, cuando un mensaje recibido por un pedido de Amazon SQS aparezca en la cola, se enviará a su canal.
3. A continuación, la tubería envía esos datos a la EventBridge API Destination Enrichment, que devuelve la información del cliente correspondiente a ese pedido.
4. Por último, la tubería envía los datos enriquecidos a la máquina de AWS Step Functions estados, que procesa el pedido.

EventBridge Conceptos de tuberías

He aquí un análisis más detallado de los componentes básicos de EventBridge Pipes.

Canalización

Una canalización dirige los eventos desde un origen único a un destino único. La canalización también incluye la posibilidad de filtrar eventos específicos y de realizar enriquecimientos de los datos del evento antes de enviarlos al destino.



Origen

EventBridge Pipes recibe datos de eventos de diversas fuentes, aplica filtros y enriquecimientos opcionales a esos datos y los envía a un destino. Si un origen impone un orden a los eventos enviados a las canalizaciones, ese orden se mantiene durante todo el proceso hasta el destino.

Para obtener más información sobre orígenes, consulte ???.

Filtros

Una canalización puede filtrar los eventos de un origen determinado y procesar solo un subconjunto de ellos. Para configurar el filtrado en una canalización, se define un patrón de eventos que la canalización utilizará para determinar qué eventos enviar al destino.

Solo se le cobrará por los eventos que coincidan con el filtro.

Para obtener más información, consulte ???.

Enriquecimiento

Con el paso de enriquecimiento de EventBridge Pipes, puede mejorar los datos de la fuente antes de enviarlos al destino. Por ejemplo, es posible que reciba eventos de creación de tíckets que no incluyan todos los datos de la entrada. Con el enriquecimiento, puede hacer que una función de Lambda llame a la API `get-ticket` para obtener todos los detalles del tícket. Luego, la canalización puede enviar esa información a un destino.

Para obtener más información sobre el enriquecimiento de los datos de eventos, consulte ???.

Destino

Una vez filtrados y enriquecidos los datos del evento, puede especificar la canalización para enviarlos a un destino específico, como una transmisión de Amazon Kinesis o un grupo de CloudWatch registros de Amazon. Para obtener una lista de los destinos disponibles, consulte ???.

Puede transformar los datos una vez mejorados y antes de que la canalización los envíe al destino. Para obtener más información, consulte ???.

Varias canalizaciones, cada una con un origen diferente, pueden enviar eventos al mismo destino.

También puede usar canalizaciones y buses de eventos juntos para enviar eventos a varios destinos. Un caso de uso común es crear una canalización con un bus de eventos como destino; la canalización envía los eventos al bus de eventos, que luego los envía a varios destinos. Por ejemplo, puede crear una canalización con un flujo de DynamoDB como origen y un bus de eventos como destino. La canalización recibe los eventos del flujo de DynamoDB y los envía al bus de eventos, que, a su vez, los envía a varios destinos de acuerdo con las reglas especificadas en el bus de eventos.

Permisos para Amazon EventBridge Pipes

Al configurar una canalización, puede utilizar una función de ejecución existente o hacer que EventBridge cree una para usted con los permisos necesarios. Los permisos que requiere EventBridge Pipes varían en función del tipo de origen y se indican a continuación. Si está configurando su propio rol de ejecución, debe añadir estos permisos usted mismo.

> **Note**
>
> Si no está seguro de cuáles son los permisos detallados exactos necesarios para acceder a la fuente, utilice la consola de EventBridge Pipes para crear un nuevo rol y, a continuación, inspeccione las acciones que se indican en la política.

Temas

Permisos de rol de ejecución de DynamoDB

Para DynamoDB Streams, EventBridge Pipes necesita los siguientes permisos para administrar los recursos relacionados con el flujo de datos de DynamoDB.

- `dynamodb:DescribeStream`
- `dynamodb:GetRecords`
- `dynamodb:GetShardIterator`
- `dynamodb:ListStreams`

Para enviar los registros de los lotes con errores a la cola de mensajes fallidos de la canalización, su rol de ejecución de canalización necesita el siguiente permiso:

- `sqs:SendMessage`

Permisos de rol de ejecución de Kinesis

Para Kinesis, EventBridge Pipes necesita los siguientes permisos para administrar los recursos relacionados con el flujo de datos de Kinesis.

- kinesis:DescribeStream
- kinesis:DescribeStreamSummary
- kinesis:GetRecords
- kinesis:GetShardIterator
- kinesis:ListShards
- kinesis:ListStreams
- kinesis:SubscribeToShard

Para enviar los registros de los lotes con errores a la cola de mensajes fallidos de la canalización, su rol de ejecución de canalización necesita el siguiente permiso:

- sqs:SendMessage

Permisos de rol de ejecución de Amazon MQ

Para Amazon MQ, EventBridge Pipes necesita los siguientes permisos para administrar los recursos relacionados con el agente de mensajes de Amazon MQ.

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

- logs:PutLogEvents

Permisos de rol de ejecución de Amazon MSK

Para Amazon MSK, EventBridge Pipes necesita los siguientes permisos para administrar los recursos relacionados con el tema de Amazon MSK.

> **Note**
>
> Si utiliza la autenticación basada en roles de IAM, el rol de ejecución necesitará los permisos que se indican en ???, además de los que se indican a continuación.

- kafka:DescribeClusterV2
- kafka:GetBootstrapBrokers
- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeVpcs
- ec2:DeleteNetworkInterface
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- logs:CreateLogGroup
- logs:CreateLogStream
- logs:PutLogEvents

Permisos del rol de ejecución de Apache Kafka autoadministrado

Para Apache Kafka autoadministrado, EventBridge requiere los siguientes permisos para gestionar los recursos relacionados con el flujo de Apache Kafka autoadministrado.

Permisos necesarios

Para crear y almacenar registros en un grupo de registros en Registros de Amazon CloudWatch, su canalización debe tener los siguientes permisos en el rol de ejecución:

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

Permisos opcionales

Es posible que la canalización también necesite permisos para:

- Describir el secreto de Secrets Manager.
- Acceder a su clave administrada por el cliente de AWS Key Management Service (AWS KMS).
- Acceder a su Amazon VPC.

Secrets Manager y permisos de AWS KMS

En función del tipo de control de acceso que configure para los agentes de Apache Kafka, es posible que su canalización necesite permiso para acceder a su secreto de Secrets Manager o para descifrar su clave administrada por el cliente de AWS KMS. Para acceder a estos recursos, el rol de ejecución de la función debe tener los siguientes permisos:

- secretsmanager:GetSecretValue
- kms:Decrypt

Permisos de VPC

Si solo los usuarios de una VPC pueden acceder al clúster de Apache Kafka autoadministrado, su canalización debe tener permiso para acceder a sus recursos de Amazon VPC. Estos recursos incluyen su VPC, subredes, grupos de seguridad e interfaces de red. Para acceder a estos recursos, el rol de ejecución de su canalización debe tener los siguientes permisos:

- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeVpcs
- ec2:DeleteNetworkInterface
- ec2:DescribeSubnets

- `ec2:DescribeSecurityGroups`

Permisos de rol de ejecución de Amazon SQS

Para Amazon SQS, EventBridge necesita los siguientes permisos para administrar los recursos relacionados con la cola de Amazon SQS.

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

Permisos de enriquecimiento y destino

Para realizar llamadas a la API en los recursos que posee, EventBridge Pipes necesita los permisos adecuados. EventBridge Pipes utiliza el rol de IAM que usted especifique en la canalización para llamadas de enriquecimiento y destino utilizando la entidad principal de IAM `pipes.amazonaws.com`.

Creando una EventBridge pipa de Amazon

EventBridge Pipes le permite crear point-to-point integraciones entre las fuentes y los destinos, incluidas las transformaciones y el enriquecimiento avanzados de los eventos. Para crear una EventBridge tubería, lleve a cabo los siguientes pasos:

1. ???
2. ???
3. ???
4. ???
5. ???

Para obtener información sobre cómo crear una tubería mediante la AWS CLI, consulte create-pipe en la Referencia de comandos de la AWS CLI.

Especificación de un origen

Para empezar, especifique el origen desde el que desea que la canalización reciba los eventos.

Para especificar un origen de la canalización mediante la consola

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Canalizaciones.
3. Seleccione Crear canalización.
4. Escriba un nombre para la canalización.
5. (Opcional) Agregue una descripción de la canalización.
6. En la pestaña Crear canalización, en Origen, seleccione el tipo de origen que desee especificar para esta canalización y configure el origen.

   Las propiedades de configuración difieren en función del tipo de origen que seleccione:

   Confluent

   Para configurar una transmisión de Confluent Cloud como fuente, mediante la consola

   1. En Source, elija Confluent Cloud.
   2. En Servidores de arranque, introduzca el par de direcciones `host:port` de sus agentes.
   3. En Nombre de tema, introduzca el nombre del tema desde donde leerá la canalización.
   4. (Opcional) En VPC, elija el VPC que desee. A continuación, en Subredes de VPC, seleccione las subredes deseadas. En Grupos de seguridad de VPC, seleccione los grupos de seguridad.
   5. Para la autenticación (opcional), active Usar autenticación y haga lo siguiente:
      a. En Método de autenticación, seleccione el tipo de autenticación.
      b. En Clave secreta, seleccione la clave secreta.

      Para obtener más información, consulte Autenticarse en los recursos de Confluent Cloud en la documentación de Confluent.
   6. (Opcional) En Configuración adicional: opcional, haga lo siguiente:
      a. En Posición inicial, seleccione una de las siguientes opciones:
         - Última: comience a leer el flujo con el registro más reciente de la partición.
         - Recortar horizonte: comience a leer el flujo con el último registro sin recortar de la partición. Este es el registro más antiguo de la partición.
      b. En Tamaño de lote: opcional, introduzca un número máximo de registros para cada lote. El valor predeterminado es 100.

c. En Ventana de lote: opcional, introduzca un número máximo de segundos para recopilar los registros antes de continuar.

DynamoDB

1. En Origen, seleccione DynamoDB.
2. En flujo de DynamoDB, seleccione el flujo que desee usar como origen.
3. En Posición inicial, seleccione una de las siguientes opciones:
   - Última: comience a leer el flujo con el registro más reciente de la partición.
   - Recortar horizonte: comience a leer el flujo con el último registro sin recortar de la partición. Este es el registro más antiguo de la partición.
4. (Opcional) En Configuración adicional: opcional, haga lo siguiente:
   a. En Tamaño de lote: opcional, introduzca un número máximo de registros para cada lote. El valor predeterminado es 100.
   b. En Ventana de lote: opcional, introduzca un número máximo de segundos para recopilar los registros antes de continuar.
   c. En Lotes simultáneos por partición: opcional, introduzca el número de lotes de la misma partición que se pueden leer al mismo tiempo.
   d. En Fallo parcial de elementos del lote, seleccione lo siguiente:
      - AUTOMATIC_BISECT: Divide en dos cada lote e intenta volver a enviar cada mitad hasta que se procesan todos los registros o aparece un mensaje de error en el lote.

> **Note**
>
> Si no selecciona AUTOMATIC_BISECT, puede devolver determinados registros fallidos y solo se intentará volver a enviar esos registros.

Kinesis

Para configurar un origen de Kinesis con la consola

1. En Origen, seleccione Kinesis.
2. En Flujo de Kinesis, seleccione el flujo que desee usar como origen.
3. En Posición inicial, seleccione una de las siguientes opciones:

- Última: comience a leer el flujo con el registro más reciente de la partición.
- Recortar horizonte: comience a leer el flujo con el último registro sin recortar de la partición. Este es el registro más antiguo de la partición.
- En marca de tiempo: comience a leer el flujo a partir de una hora específica. En Marca de tiempo, introduzca una fecha y una hora con los formatos AAAA/MM/DD y hh:mm:ss.

4. (Opcional) En Configuración adicional: opcional, haga lo siguiente:
   a. En Tamaño de lote: opcional, introduzca un número máximo de registros para cada lote. El valor predeterminado es 100.
   b. (Opcional) En Ventana de lote: opcional, introduzca un número máximo de segundos para recopilar los registros antes de continuar.
   c. En Lotes simultáneos por partición: opcional, introduzca el número de lotes de la misma partición que se pueden leer al mismo tiempo.
   d. En Fallo parcial de elementos del lote, seleccione lo siguiente:
      - AUTOMATIC_BISECT: Divide en dos cada lote e intenta volver a enviar cada mitad hasta que se procesan todos los registros o aparece un mensaje de error en el lote.

> **Note**
>
> Si no selecciona AUTOMATIC_BISECT, puede devolver determinados registros fallidos y solo se intentará volver a enviar esos registros.

Amazon MQ

Para configurar un origen de Amazon MQ con la consola

1. En Tipo de origen, seleccione Amazon MQ.
2. En agente de Amazon MQ, seleccione el flujo que desee usar como origen.
3. En Nombre de cola, introduzca el nombre de la cola desde donde leerá la canalización.
4. En Método de autenticación, seleccione BASIC_AUTH.
5. En Clave secreta, seleccione la clave secreta.
6. (Opcional) En Configuración adicional: opcional, haga lo siguiente:
   a. En Tamaño de lote: opcional, introduzca un número máximo de mensajes para cada lote. El valor predeterminado es 100.

b. En Ventana de lote: opcional, introduzca un número máximo de segundos para recopilar los registros antes de continuar.

Amazon MSK

Para configurar un origen de Amazon MSK con la consola

1. En Origen, seleccione Amazon MSK.
2. En Clúster de Amazon MSK, seleccione el clúster que desee usar.
3. En Nombre de tema, introduzca el nombre del tema desde donde leerá la canalización.
4. (Opcional) En ID del grupo de consumidores: opcional, introduzca el ID de un grupo de consumidores al que desea que se una la canalización.
5. (Opcional) En Autenticación: opcional, active Usar autenticación y haga lo siguiente:
   a. En Método de autenticación, seleccione el tipo que desees.
   b. En Clave secreta, seleccione la clave secreta.
6. (Opcional) En Configuración adicional: opcional, haga lo siguiente:
   a. En Tamaño de lote: opcional, introduzca un número máximo de registros para cada lote. El valor predeterminado es 100.
   b. En Ventana de lote: opcional, introduzca un número máximo de segundos para recopilar los registros antes de continuar.
   c. En Posición inicial, seleccione una de las siguientes opciones:
      - Última: comience a leer el tema con el registro más reciente de la partición.
      - Recortar horizonte: comience a leer el tema con el último registro sin recortar de la partición. Este es el registro más antiguo de la partición.

> **Note**
>
> Recortar horizonte es lo mismo que Primero para Apache Kafka.

Self managed Apache Kafka

Para configurar una origen de Apache Kafka autogestionado mediante la consola

1. En Origen, seleccione Apache Kafka autogestionado.

2. En Servidores de arranque, introduzca el par de direcciones `host:port` de sus agentes.
3. En Nombre de tema, introduzca el nombre del tema desde donde leerá la canalización.
4. (Opcional) En VPC, elija el VPC que desee. A continuación, en Subredes de VPC, seleccione las subredes deseadas. En Grupos de seguridad de VPC, seleccione los grupos de seguridad.
5. (Opcional) En Autenticación: opcional, active Usar autenticación y haga lo siguiente:
   a. En Método de autenticación, seleccione el tipo de autenticación.
   b. En Clave secreta, seleccione la clave secreta.
6. (Opcional) En Configuración adicional: opcional, haga lo siguiente:
   a. En Posición inicial, seleccione una de las siguientes opciones:
      - Última: comience a leer el flujo con el registro más reciente de la partición.
      - Recortar horizonte: comience a leer el flujo con el último registro sin recortar de la partición. Este es el registro más antiguo de la partición.
   b. En Tamaño de lote: opcional, introduzca un número máximo de registros para cada lote. El valor predeterminado es 100.
   c. En Ventana de lote: opcional, introduzca un número máximo de segundos para recopilar los registros antes de continuar.

Amazon SQS

Para configurar un origen de Amazon SQS con la consola

1. En Origen, seleccione SQS.
2. En Cola de SQS, seleccione la cola que desee utilizar.
3. (Opcional) En Configuración adicional: opcional, haga lo siguiente:
   a. En Tamaño de lote: opcional, introduzca un número máximo de registros para cada lote. El valor predeterminado es 100.
   b. En Ventana de lote: opcional, introduzca un número máximo de segundos para recopilar los registros antes de continuar.

Configuración del filtrado de eventos (opcional)

Puede añadir filtros a su canalización de forma que solo envíe un subconjunto de eventos del origen al destino.

Para configurar el filtrado mediante la consola

1. Seleccione Filtrado.
2. En Evento de muestra: opcional, verá un evento de muestra que puede usar para crear su patrón de eventos, o puede introducir su propio evento seleccionando Introduce el suyo.
3. En Patrón de eventos, introduce el patrón de eventos que quiere usar para filtrar los eventos. Para obtener más información sobre la construcción de filtros, consulte. ???

   A continuación se muestra un ejemplo de patrón de eventos que solo envía eventos con el valor Seattle en el campo Ciudad.

   ```
   {
     "data": {
       "City": ["Seattle"]
     }
   }
   ```

Ahora que se están filtrando los eventos, puede añadir un enriquecimiento opcional y un destino para la canalización.

Definición del enriquecimiento de eventos (opcional)

Puede enviar los datos del evento para su enriquecimiento a una función Lambda, a una máquina de estado AWS Step Functions , a Amazon API Gateway o a un destino de API.

Para seleccionar el enriquecimiento

1. Seleccione Enriquecimiento.
2. En Detalles, en Servicio, seleccione el servicio y la configuración relacionada que desee usar para el enriquecimiento.

También puede transformar los datos antes de enviarlos para su enriquecimiento.

(Opcional) Para definir el transformador de entrada

1. Seleccione Transformador de entrada de enriquecimiento: opcional.
2. En Eventos de muestra/Carga de evento, seleccione el tipo de evento de muestra.
3. En Transformador, introduzca la sintaxis del transformador, por ejemplo `"Event happened at <$.detail.field>."`, donde `<$.detail.field>` es una referencia a un campo del evento de muestra. También puede hacer doble clic en un campo del evento de muestra para añadirlo al transformador.
4. En Salida, compruebe que la salida tenga el aspecto que desea.

Ahora que los datos se han filtrado y enriquecido, debe definir un destino al que enviar los datos del evento.

Configuración de un destino

Para configurar un destino

1. Seleccione Destino.
2. En Detalles, en Servicio de destino, seleccione el destino. Los campos mostrados varían en función del destino que seleccione. Introduzca la información específica de este tipo de destino, según sea necesario.

También puede transformar los datos antes de enviarlos al destino.

(Opcional) Para definir el transformador de entrada

1. Seleccione Transformador de entrada de destino: opcional.
2. En Eventos de muestra/Carga de evento, seleccione el tipo de evento de muestra.
3. En Transformador, introduzca la sintaxis del transformador, por ejemplo `"Event happened at <$.detail.field>."`, donde `<$.detail.field>` es una referencia a un campo del evento de muestra. También puede hacer doble clic en un campo del evento de muestra para añadirlo al transformador.
4. En Salida, compruebe que la salida tenga el aspecto que desea.

Ahora que la canalización está configurada, asegúrese de que sus ajustes estén configurados correctamente.

Configuración de los ajustes de la canalización

Una canalización está activa por defecto, pero puede desactivarla. También puede especificar los permisos de la canalización, configurar el registro de la canalización y añadir etiquetas.

Para configurar los ajustes de la canalización

1. Seleccione la pestaña Ajustes de la canalización.
2. Por defecto, las canalizaciones recién creadas están activas desde su creación. Si quiere crear una canalización inactiva, en Activación, en Activar canalización, desactive la opción Activa.
3. En Permisos, en Rol de ejecución, realice una de las siguientes acciones:

   a. Para EventBridge crear una nueva función de ejecución para esta canalización, elija Crear una nueva función para este recurso específico. En Nombre del rol, si lo desea, puede editar el nombre del rol.

   b. Para usar un rol de ejecución existente, seleccione Usar un rol existente. En Nombre del rol, seleccione el rol.
4. (Opcional) Si ha especificado un DynamoDB flujo Kinesis o flujo como fuente de canalización, puede configurar una política de reintentos y una cola de mensajes sin salida (DLQ).

   En Política de reintentos y cola de mensajes fallidos: opcional, haga lo siguiente:

   En Política de reintentos, haga lo siguiente:

   a. Si quieres activar las políticas de reintentos, active Reintento. De forma predeterminada, las canalizaciones recién creadas no tienen activada la política de reintentos.

   b. En Antigüedad máxima del evento, introduzca un valor entre un minuto (00:01) y 24 horas (24:00).

   c. En Cantidad de reintentos, introduzca un número entre 0 y 185.

   d. Si desea usar una cola de mensajes fallidos (DLQ), active Cola de mensajes fallidos, seleccione el método que prefiera y elija la cola o el tema que quiera usar. De forma predeterminada, las canalizaciones recién creadas no utilizan una DLQ.
5. (Opcional) En Registros: opcional, puede configurar la forma en que las canalizaciones de EventBridge envían la información de registro a los servicios compatibles, incluida la forma de configurar esos registros.

   Para obtener más información sobre los registros de canalizaciones, consulte ???.

CloudWatch Los registros se seleccionan como destino de registro de forma predeterminada, al igual que el nivel de registro. `ERROR` Por lo tanto, de forma predeterminada, EventBridge Pipes crea un nuevo grupo de CloudWatch registros al que envía los registros que contienen el `ERROR` nivel de detalle.

Para que EventBridge Pipes envíe los registros de registro a cualquiera de los destinos de registro compatibles, haga lo siguiente:

a. En Registros: opcional, seleccione los destinos a los que quiere que se envíen los registros.

b. En Nivel de registro, elija el nivel de información EventBridge que desee incluir en los registros de registro. El nivel de registro de `ERROR` está seleccionado de forma predeterminada.

   Para obtener más información, consulte ???.

c. Seleccione Incluir datos de ejecución si EventBridge desea incluir la información de carga útil del evento y la información de solicitud y respuesta de servicio en los registros.

   Para obtener más información, consulte ???.

d. Configure cada destino de registro que haya seleccionado:

   En el CloudWatch Logs caso de los registros, en CloudWatch los registros, haga lo siguiente:

   - Para el grupo de CloudWatch registros, elija si desea EventBridge crear un nuevo grupo de registros o puede seleccionar un grupo de registros existente o especificar el ARN de un grupo de registros existente.
   - Para los grupos de registro nuevos, edite el nombre del grupo de registro según desee.

   CloudWatch los registros están seleccionados de forma predeterminada.

   Para los registros de Firehose transmisión, en registro de Firehose transmisión, seleccione la Firehose transmisión.

   Para Amazon S3 los registros, en S3 logs, haga lo siguiente:

   - Introduzca el nombre del bucket que utilizará como destino del registro.
   - Introduzca el ID de AWS cuenta del propietario del bucket.

- Introduzca el texto de prefijo que desee utilizar cuando EventBridge crea objetos S3.

  Para obtener más información, consulte Organizar objetos mediante prefijos en la Guía del usuario de Amazon Simple Storage Service .
- Elige cómo quieres EventBridge formatear los registros de S3:
  - `json`: JSON
  - `plain`: Texto sin formato
  - `w3c`: Formato de archivo de registro extendido de W3C

6. (Opcional) En Etiquetas; opcional, seleccione Añadir etiqueta nueva e introduzca una o varias etiquetas para la regla. Para obtener más información, consulte ???.
7. Seleccione Crear canalización.

Validación de los parámetros de configuración

Tras crear una tubería, EventBridge valida los siguientes parámetros de configuración:

- Función de IAM: dado que la fuente de una tubería no se puede cambiar una vez creada la tubería, EventBridge verifica que la función de IAM proporcionada pueda acceder a la fuente.

  > **Note**
  >
  > EventBridge no realiza la misma validación para los enriquecimientos o los objetivos, ya que se pueden actualizar una vez creada la canalización.

- Procesamiento por lotes: EventBridge valida que el tamaño del lote de la fuente no supere el tamaño máximo del lote del objetivo. Si lo hace, EventBridge requiere un tamaño de lote inferior. Además, si un destino no admite el procesamiento por lotes, no puede configurar el procesamiento por lotes EventBridge para el origen.
- Enriquecimientos: EventBridge valida que el tamaño del lote para los enriquecimientos de API Gateway y API de destino sea 1, ya que solo se admiten tamaños de lote de 1.

Inicio o detención de una canalización

Por defecto, una canalización está `Running` y procesa los eventos cuando se crea.

Si crea una canalización con orígenes de Amazon SQS, Kinesis o DynamoDB, la creación de la canalización normalmente puede tardar uno o dos minutos.

Si crea una canalización con orígenes de Amazon MSK, Apache Kafka autoadministrado o Amazon MQ, la creación de canalizaciones puede tardar hasta diez minutos.

Para crear una canalización sin procesar eventos, utilice la consola

- Desactive la configuración Activar canalización.

Para crear una canalización sin procesar eventos mediante programación

- En la llamada a la API, establezca el `DesiredState` en `Stopped`.

Para iniciar o detener una canalización existente mediante la consola

- En la pestaña Configuración de canalización, en Activación, para Activar canalización, active o desactive la opción Activa.

Para iniciar o detener una canalización existente mediante programación

- En la llamada a la API, establezca el parámetro `DesiredState` en `RUNNING` o `STOPPED`.

Puede haber un retraso entre el momento en que una canalización esté `STOPPED` y el momento en que deja de procesar eventos:

- En el caso de Amazon SQS y de los orígenes de flujo, este retraso suele ser inferior a dos minutos.
- En el caso de los orígenes de Amazon MQ y Apache Kafka, este retraso puede ser de hasta quince minutos.

Fuentes de Amazon EventBridge Pipes

EventBridge Pipes recibe datos de eventos de diversas fuentes, aplica filtros y enriquecimientos opcionales a esos datos y los envía a un destino.

Si una fuente impone el orden de los eventos enviados a EventBridge Pipes, ese orden se mantiene durante todo el proceso hasta el destino.

Los siguientes AWS servicios se pueden especificar como fuentes para EventBridge Pipes:

- Flujo de Amazon DynamoDB
- Flujo de Amazon Kinesis
- Agente de Amazon MQ
- Flujo de Amazon MSK
- Cola de Amazon SQS
- Transmisión de Apache Kafka

  Al especificar una transmisión de Apache Kafka como fuente de canalización, puede especificar una transmisión de Apache Kafka que administre usted mismo o una que administre un proveedor externo, como:

  - Confluent Cloud
  - CloudKarafka
  - Redpanda

Flujo de Amazon DynamoDB como origen

Puede usar EventBridge Pipes para recibir registros en un flujo de DynamoDB. A continuación, si lo desea, puede filtrar o enriquecer estos registros antes de enviarlos a un destino para su procesamiento. Al configurar la canalización, puede elegir ajustes específicos para Amazon DynamoDB Streams. EventBridge Pipes mantiene el orden de los registros del flujo de datos al enviar esos datos al destino.

> **Important**
>
> Al deshabilitar un flujo de DynamoDB que es el origen de una canalización, esa canalización queda inutilizada, aunque se vuelva a habilitar el flujo. Esto puede suceder por los motivos siguientes:
>
> - No puede detener, iniciar ni actualizar una canalización cuyo origen esté deshabilitado.

- No puede actualizar una canalización con un origen nuevo después de crearla. Al volver a habilitar el flujo de DynamoDB, a ese flujo se le asigna un nuevo nombre de recurso de Amazon (ARN) y deja de estar asociado a la canalización.

Si vuelve a habilitar el flujo de DynamoDB, tendrá que crear una nueva canalización con el nuevo ARN del flujo.

Evento de ejemplo

En el siguiente evento de ejemplo se muestra la información que recibe la canalización. Puede usar este evento para crear y filtrar sus patrones de eventos o para definir la transformación de entrada. No todos los campos se pueden filtrar. Para obtener más información sobre cómo filtrar campos, consulte ???.


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

Flujos de sondeo y procesamiento por lotes

EventBridge sondea las particiones de su flujo de DynamoDB y busca registros cuatro veces por segundo. Cuando hay registros disponibles, EventBridge procesa el evento y espera el resultado. Si el procesamiento se realiza correctamente, EventBridge reanuda el sondeo hasta que recibe más registros.

De forma predeterminada, EventBridge invoca su canalización tan pronto como los registros estén disponibles. Si el lote que EventBridge lee desde el origen solo tiene un registro, solo se procesa un evento. Para evitar procesar un número de registros pequeño, puede indicar a la canalización que almacene en búfer registros hasta cinco minutos configurando un plazo de procesamiento por lotes. Antes de procesar los eventos, EventBridge continúa leyendo los registros del origen hasta que haya recopilado un lote completo, venza el plazo de procesamiento por lotes o el lote alcance el límite de carga de 6 MB.

También puede aumentar la simultaneidad procesando varios lotes de cada partición en paralelo. EventBridge puede procesar hasta 10 lotes en cada partición simultáneamente. Si aumenta el número de lotes simultáneos por partición, EventBridge sigue garantizando el procesamiento en orden a nivel de clave de partición.

Configure los ajustes de `ParallelizationFactor` para procesar una partición de un flujo de datos de Kinesis o DynamoDB con más de una ejecución de canalización simultáneamente. Puede especificar el número de lotes simultáneos que EventBridge sondea desde una partición a través de un factor de paralelización de 1 (predeterminado) a 10. Por ejemplo, al establecer `ParallelizationFactor` como 2, puede tener 200 ejecuciones de EventBridge Pipe simultáneas como máximo para procesar 100 particiones de datos de Kinesis. Esto ayuda a escalar el rendimiento de procesamiento cuando el volumen de datos es volátil y el `IteratorAge` es alto. Tenga en cuenta que el factor de paralelización no funcionará si está utilizando la agregación de Kinesis.

Posición inicial de flujos y sondeo

Tenga en cuenta que el sondeo de flujos durante la creación de canalizaciones y las actualizaciones es, en última instancia, coherente.

- Durante la creación de canalizaciones, es posible que se demore varios minutos en iniciar el sondeo de los eventos del flujo.
- Durante las actualizaciones de las canalizaciones, es posible que se demore varios minutos en detener y reiniciar el sondeo de los eventos del flujo.

Esto significa que, si especifica `LATEST` como posición inicial del flujo, la canalización podría omitir eventos durante la creación de canalizaciones o las actualizaciones. Para garantizar que no se pierda ningún evento, especifique la posición inicial del flujo como `TRIM_HORIZON`.

Informes de fallos de artículos de lote

Cuando EventBridge consume y procesa datos de flujos de un origen, de forma predeterminada asigna puntos de control hasta el número de secuencia más alto de un lote solo cuando el lote ese procesa correctamente por completo. Para evitar el reprocesamiento de los mensajes procesados correctamente en un lote con errores, puede configurar el enriquecimiento o el destino para que devuelva un objeto que indique qué mensajes se han procesado correctamente y cuáles no. Esto se denomina respuesta parcial por lotes.

Para obtener más información, consulte ???.

Condiciones de éxito y fracaso

EventBridge trata un lote como un éxito completo si devuelve cualquiera de los siguientes elementos:

- Una lista `batchItemFailure` vacía
- Una lista `batchItemFailure` nula
- Un `EventResponse` vacío
- Un `EventResponse` nulo

EventBridge trata un lote como un error completo si devuelve cualquiera de los siguientes elementos:

- Una cadena `itemIdentifier` vacía
- Un `itemIdentifier` nulo
- Un `itemIdentifier` con un mal nombre de clave

Eventbridge intenta volver a procesar los mensajes fallidos conforme a su estrategia de reintentos.

Flujo de Amazon Kinesis como origen

Puede usar EventBridge Pipes para recibir registros en un flujo de datos de Kinesis. A continuación, si lo desea, puede filtrar o enriquecer estos registros antes de enviarlos a uno de los destinos disponibles para su procesamiento. Al configurar la canalización, puede elegir ajustes específicos de Kinesis. EventBridge Pipes mantiene el orden de los registros del flujo de datos al enviar esos datos al destino.

Un flujo de datos de Kinesis es un conjunto de particiones. Cada partición contiene una secuencia de registros de datos. Un consumidor es una aplicación que procesa los datos procedentes de

un flujo de datos de Kinesis. Puede asignar una canalización de EventBridge a un consumidor de rendimiento compartido (iterador estándar) o a un consumidor de rendimiento dedicado con distribución ramificada mejorada.

Para iteradores estándar, EventBridge usa el protocolo HTTP para sondear cada partición en su flujo de Kinesis para registros. La canalización comparte el rendimiento de lectura con otros consumidores de la partición.

Para minimizar la latencia y maximizar el rendimiento de lectura, puede crear un consumidor de flujo de datos con distribución ramificada mejorada. Los consumidores de flujos obtienen una conexión dedicada a cada partición que no afecta a las demás aplicaciones que leen el flujo. El rendimiento dedicado puede ser útil si hay muchas aplicaciones que leen los mismos datos, o si se está reprocesando un flujo con registros de gran tamaño. Kinesis publica los registros en EventBridge sobre HTTP/2. Para obtener información detallada sobre los flujos de datos de Kinesis, consulte Lectura de datos de Amazon Kinesis Data Streams.

Evento de ejemplo

En el siguiente evento de ejemplo se muestra la información que recibe la canalización. Puede usar este evento para crear y filtrar sus patrones de eventos o para definir la transformación de entrada. No todos los campos se pueden filtrar. Para obtener más información sobre los campos que puede filtrar, consulte ???.


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

Flujos de sondeo y procesamiento por lotes

EventBridge sondea las particiones de su flujo de Kinesis y busca registros cuatro veces por segundo. Cuando hay registros disponibles, EventBridge procesa el evento y espera el resultado. Si el procesamiento se realiza correctamente, EventBridge reanuda el sondeo hasta que recibe más registros.

De forma predeterminada, EventBridge invoca su canalización tan pronto como los registros estén disponibles. Si el lote que EventBridge lee desde el origen solo tiene un registro, solo se procesa un evento. Para evitar procesar un número de registros pequeño, puede indicar a la canalización que almacene en búfer registros hasta cinco minutos configurando un plazo de procesamiento por lotes. Antes de procesar los eventos, EventBridge continúa leyendo los registros del origen hasta que haya recopilado un lote completo, venza el plazo de procesamiento por lotes o el lote alcance el límite de carga de 6 MB.

También puede aumentar la concurrencia procesando varios lotes de cada partición en paralelo. EventBridge puede procesar hasta 10 lotes en cada partición simultáneamente. Si aumenta el número de lotes simultáneos por partición, EventBridge sigue garantizando el procesamiento en orden a nivel de clave de partición.

Configure los ajustes de `ParallelizationFactor` para procesar una partición de un flujo de datos de Kinesis o DynamoDB con más de una ejecución de canalización simultáneamente. Puede especificar el número de lotes simultáneos que EventBridge sondea desde una partición a través de un factor de paralelización de 1 (predeterminado) a 10. Por ejemplo, al establecer `ParallelizationFactor` como 2, puede tener 200 ejecuciones de EventBridge Pipe simultáneas

como máximo para procesar 100 particiones de datos de Kinesis. Esto ayuda a escalar el rendimiento de procesamiento cuando el volumen de datos es volátil y el `IteratorAge` es alto. Tenga en cuenta que el factor de paralelización no funcionará si está utilizando la agregación de Kinesis.

Posición inicial de flujos y sondeo

Tenga en cuenta que el sondeo de flujos durante la creación de canalizaciones y las actualizaciones es, en última instancia, coherente.

- Durante la creación de canalizaciones, es posible que se demore varios minutos en iniciar el sondeo de los eventos del flujo.
- Durante las actualizaciones de las canalizaciones, es posible que se demore varios minutos en detener y reiniciar el sondeo de los eventos del flujo.

Esto significa que, si especifica `LATEST` como posición inicial del flujo, la canalización podría omitir eventos durante la creación de canalizaciones o las actualizaciones. Para garantizar que no se pierda ningún evento, especifique la posición inicial del flujo como `TRIM_HORIZON` o `AT_TIMESTAMP`.

Informes de fallos de artículos de lote

Cuando EventBridge consume y procesa datos de flujos de un origen, de forma predeterminada asigna puntos de control hasta el número de secuencia más alto de un lote solo cuando el lote ese procesa correctamente por completo. Para evitar el reprocesamiento de los mensajes procesados correctamente en un lote con errores, puede configurar el enriquecimiento o el destino para que devuelva un objeto que indique qué mensajes se han procesado correctamente y cuáles no. Esto se denomina respuesta parcial por lotes.

Para obtener más información, consulte ???.

Condiciones de éxito y fracaso

EventBridge trata un lote como un éxito completo si devuelve cualquiera de los siguientes elementos:

- Una lista `batchItemFailure` vacía
- Una lista `batchItemFailure` nula
- Un `EventResponse` vacío
- Un `EventResponse` nulo

EventBridge trata un lote como un error completo si devuelve cualquiera de los siguientes elementos:

- Una `itemIdentifier` cadena vacía
- Un `itemIdentifier` nulo
- Un `itemIdentifier` con un mal nombre de clave

Eventbridge intenta volver a procesar los mensajes fallidos conforme a su estrategia de reintentos.

Agente de mensajes Amazon MQ como origen

Puedes usar EventBridge Pipes para recibir registros de un agente de mensajes de Amazon MQ. A continuación, si lo desea, puede filtrar o enriquecer estos registros antes de enviarlos a uno de los destinos disponibles para su procesamiento. Existen ajustes específicos de Amazon MQ que puedes elegir al configurar una tubería. EventBridge Pipes mantiene el orden de los registros del agente de mensajes al enviar esos datos al destino.

Amazon MQ es un servicio de agente de mensajes administrado para Apache ActiveMQ y RabbitMQ. Un agente de mensajes permite que las aplicaciones de software y los componentes se comuniquen mediante diferentes lenguajes de programación, sistemas operativos y protocolos de mensajería formales con temas o colas como destinos de eventos.

Amazon MQ también puede administrar instancias de Amazon Elastic Compute Cloud (Amazon EC2) en su nombre instalando agentes de ActiveMQ o RabbitMQ. Una vez instalado el agente, este proporciona diferentes topologías de red y cubre otras necesidades de infraestructura para sus instancias.

El origen de Amazon MQ tiene las siguientes restricciones de configuración:

- Cuenta cruzada: EventBridge no admite el procesamiento entre cuentas. No puedes usar EventBridge para procesar registros de un agente de mensajes de Amazon MQ que esté en una cuenta diferente AWS .
- Autenticación: para ActiveMQ, solo se admite SimpleAuthenticationPluginActiveMQ. Para RabbitMQ, solo se admite el mecanismo de autenticación PLAIN. Para administrar las credenciales, utilice AWS Secrets Manager. Para obtener más información acerca de la autenticación de ActiveMQ, consulte Integración de agentes de ActiveMQ con LDAP en la Guía para desarrolladores de Amazon MQ.
- Cuota de conexión: los agentes tienen un número máximo de conexiones permitidas por protocolo de nivel de cable. Esta cuota se basa en el tipo de instancia del broker. Para obtener

más información, consulte la sección Agentes de *Cuotas en Amazon MQ* en la Guía para desarrolladores de Amazon MQ.

- Conectividad: puede crear agentes en una nube privada virtual (VPC) pública o privada. En el caso de las VPC privadas, su canalización necesita acceso a la VPC para recibir mensajes.
- Destinos de eventos: solo se admiten los destinos de cola. Sin embargo, puede utilizar un tema virtual, que se comporta como un tema internamente y como una cola externamente cuando interactúa con sus canalizaciones. Para obtener más información, consulte Destinos virtuales en el sitio web de Apache ActiveMQ y Virtual Hosts en el sitio web RabbitMQ.
- Topología de red: para ActiveMQ, solo se admite una instancia única o agente en espera por canalización. Para RabbitMQ, solo se admite una implementación de agente o clúster de una instancia única por cada canalización. Los agentes de instancia única requieren un punto de conexión de conmutación por error. Para obtener más información acerca de estos modos de implementación de agente, consulte Arquitectura de agente Active MQ y Arquitectura de agente de Rabbit MQ en la Guía para desarrolladores de Amazon MQ.
- Protocolos: los protocolos compatibles dependen de la integración de Amazon MQ que utilice.
  - Para las integraciones de ActiveMQ EventBridge , utiliza OpenWire el protocolo /Java Message Service (JMS) para consumir los mensajes. El consumo de mensajes no es compatible con ningún otro protocolo. EventBridge solo admite las BytesMessageoperaciones TextMessagey dentro del protocolo JMS. Para obtener más información sobre el OpenWire protocolo, consulte el OpenWiresitio web de Apache ActiveMQ.
  - Para las integraciones de RabbitMQ, EventBridge utiliza el protocolo AMQP 0-9-1 para consumir los mensajes. No se admiten otros protocolos para consumir mensajes. Para obtener más información acerca de la implementación de RabbitMQ del protocolo AMQP 0-9-1, consulte la Guía de referencia completa de AMQP 0-9-1 en el sitio web de RabbitMQ.

EventBridge admite automáticamente las versiones más recientes de ActiveMQ y RabbitMQ compatibles con Amazon MQ. Para obtener las últimas versiones compatibles, consulte Notas de la versión de Amazon MQ en la Guía para desarrolladores de Amazon MQ.

> **Note**
>
> De forma predeterminada, Amazon MQ tiene una ventana de mantenimiento semanal para los agentes. Durante esa ventana de tiempo, los corredores no están disponibles. En el caso de los corredores que no estén en modo de espera, EventBridge no procesarán los mensajes hasta que finalice el período.

Eventos de ejemplo

En el siguiente evento de ejemplo se muestra la información que recibe la canalización. Puede usar este evento para crear y filtrar sus patrones de eventos o para definir la transformación de entrada. No todos los campos se pueden filtrar. Para obtener más información sobre los campos que puede filtrar, consulte ???.

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=="
  }
]
```

Grupo de consumidores

Para interactuar con Amazon MQ, EventBridge crea un grupo de consumidores que pueda leer información de tus agentes de Amazon MQ. El grupo de consumidores se crea con el mismo ID que el UUID de la canalización.

En el caso de las fuentes de Amazon MQ EventBridge , agrupa los registros y los envía a su función en una sola carga útil. Para controlar el comportamiento, puede configurar el plazo de procesamiento por lotes y el tamaño del lote. EventBridge extrae los mensajes hasta que ocurre una de las siguientes situaciones:

- Los registros procesados alcanzan un tamaño máximo de carga de 6 MB.
- El plazo de procesamiento por lotes caduca.
- El número de registros alcanza el tamaño total del lote.

EventBridge convierte el lote en una sola carga útil y, a continuación, invoca la función. Los mensajes no se mantienen ni se deserializan. En su lugar, el grupo de consumidores los recupera como un BLOB de bytes. A continuación, los codifica con base64 en una carga JSON. Si la canalización devuelve un error en alguno de los mensajes de un lote, EventBridge vuelve a intentarlo con todo el lote de mensajes hasta que el procesamiento se realice correctamente o los mensajes caduquen.

Configuración de red

De forma predeterminada, los agentes de Amazon MQ se crean con el indicador `PubliclyAccessible` establecido en falso. Solo cuando `PubliclyAccessible` se establece en verdadero que el agente recibe una dirección IP pública. Para tener acceso completo con su canalización, el agente debe utilizar un punto de conexión público o proporcionar acceso a la VPC.

Si su agente de Amazon MQ no es de acceso público, EventBridge debe tener acceso a los recursos de Amazon Virtual Private Cloud (Amazon VPC) asociados a su agente.

- Para acceder a la VPC de sus agentes de Amazon MQ EventBridge , puede utilizar el acceso saliente a Internet para las subredes de su fuente. Para las subredes públicas, debe ser una puerta de enlace NAT administrada. Para las subredes privadas, puede ser una puerta de enlace NAT o su propia NAT. Asegúrese de que la NAT tiene una dirección IP pública y puede conectarse a Internet.
- EventBridge Pipes también admite la entrega directa de eventos AWS PrivateLink, lo que le permite enviar eventos desde una fuente de eventos ubicada en un Amazon Virtual Private Cloud (Amazon VPC) a un destino de Pipes sin tener que atravesar la red pública de Internet. Puedes usar Pipes para sondear desde Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogestionado y Amazon MQ fuentes que residen en una subred privada sin necesidad de implementar una puerta de enlace a Internet, configurar reglas de firewall o configurar servidores proxy.

  Para configurar un punto de enlace de VPC, consulte Crear un punto de enlace de VPC en la Guía del usuario.AWS PrivateLink Para el nombre del servicio, seleccione. `com.amazonaws.region.pipes-data`

Configure sus grupos de seguridad de Amazon VPC con las siguientes reglas (como mínimo):

- Reglas de entrada: permita todo el tráfico en el puerto de agente de Amazon MQ para los grupos de seguridad especificados para su fuente.
- Reglas de salida: permiten todo el tráfico en el puerto 443 para todos los destinos. Permita todo el tráfico en el puerto de agente de Amazon MQ para los grupos de seguridad especificados para su fuente.

  Los puertos de bróker incluyen:

  - 9092 para texto sin formato
  - 9094 para TLS
  - 9096 para SASL
  - 9098 para IAM

> **Note**
>
> La configuración de Amazon VPC se puede detectar a través de la API de Amazon MQ. No tiene que configurarla durante la configuración.

Tema de Amazon Managed Streaming para Apache Kafka como origen

Puedes usar EventBridge Pipes para recibir registros de un tema de Amazon Managed Streaming for Apache Kafka (Amazon MSK). Si lo desea, también puede filtrar o enriquecer estos registros antes de enviarlos a uno de los destinos disponibles para su procesamiento. Existen ajustes específicos de Amazon MSK que puede elegir al configurar una tubería. EventBridge Pipes mantiene el orden de los registros del agente de mensajes al enviar esos datos al destino.

Amazon MSK es un servicio completamente administrado que puede usar para crear y ejecutar aplicaciones que utilizan Apache Kafka para procesar datos de streaming. Amazon MSK simplifica la configuración, el escalado y la administración de clústeres que ejecutan Apache Kafka. Con Amazon MSK, puede configurar su aplicación para varias zonas de disponibilidad y para garantizar la seguridad con AWS Identity and Access Management (IAM). Amazon MSK es compatible con múltiples versiones de código abierto de Kafka.

Amazon MSK como fuente funciona de forma similar al uso de Amazon Simple Queue Service (Amazon SQS) o Amazon Kinesis. EventBridgesondea internamente los mensajes nuevos de la fuente y, a continuación, invoca al destino de forma sincrónica. EventBridge lee los mensajes por lotes y los proporciona a su función como carga útil de eventos. El tamaño máximo del lote es configurable. (El valor predeterminado es 100 mensajes).

En el caso de las fuentes basadas en Apache Kafka, EventBridge admite parámetros de control del procesamiento, como las ventanas de procesamiento por lotes y el tamaño del lote.

EventBridge lee los mensajes de forma secuencial para cada partición. Después de EventBridge procesar cada lote, confirma las compensaciones de los mensajes de ese lote. Si el objetivo de la canalización devuelve un error para alguno de los mensajes de un lote, EventBridge vuelve a intentarlo con todo el lote de mensajes hasta que el procesamiento se realice correctamente o los mensajes caduquen.

EventBridge envía el lote de mensajes en caso de que invoque al destino. La carga de eventos contiene una matriz de mensajes. Cada elemento de matriz contiene detalles del tema y el

identificador de partición de Amazon MSK, junto con una marca de hora y un mensaje codificado en base64.

Eventos de ejemplo

En el siguiente evento de ejemplo se muestra la información que recibe la canalización. Puede usar este evento para crear y filtrar sus patrones de eventos o para definir la transformación de entrada. No todos los campos se pueden filtrar. Para obtener más información sobre los campos que puede filtrar, consulte ???.

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

Sondeo y posición inicial de flujos

Tenga en cuenta que el sondeo de flujos durante la creación de canalizaciones y las actualizaciones es, en última instancia, coherente.

- Durante la creación de canalizaciones, es posible que se demore varios minutos en iniciar el sondeo de los eventos del flujo.
- Durante las actualizaciones de las canalizaciones, es posible que se demore varios minutos en detener y reiniciar el sondeo de los eventos del flujo.

Esto significa que, si especifica `LATEST` como posición inicial del flujo, la canalización podría omitir eventos durante la creación de canalizaciones o las actualizaciones. Para garantizar que no se pierda ningún evento, especifique la posición inicial del flujo como `TRIM_HORIZON`.

Autenticación de clústeres de MSK

EventBridge necesita permiso para acceder al clúster de Amazon MSK, recuperar registros y realizar otras tareas. Amazon MSK admite varias opciones para controlar el acceso de los clientes al clúster de MSK. Para obtener más información acerca de este método de autenticación que se utiliza, consulte ???.

Opciones de acceso al clúster

- Acceso sin autenticar
- Autenticación SASL/SCRAM
- Autenticación basada en roles de IAM
- Autenticación TLS mutua
- Configuración del secreto de mTLS
- ¿Cómo elegir EventBridge un bróker bootstrap

Acceso sin autenticar

Recomendamos utilizar únicamente el acceso no autenticado para el desarrollo. El acceso no autenticado solo funcionará si la autenticación basada en roles de IAM está deshabilitada para el clúster.

Autenticación SASL/SCRAM

Amazon MSK admite autenticación simple y autenticación de capa de seguridad/mecanismo de autenticación de respuesta por desafío saltado (SASL/SCRAM) con cifrado de seguridad de la capa de transporte (TLS). EventBridge Para conectarse al clúster, debe almacenar las credenciales de autenticación (credenciales de inicio de sesión) en secreto. AWS Secrets Manager

Para obtener más información sobre Secrets Manager, consulte Autenticación de usuario y contraseña con AWS Secrets Manager (en la Guía para desarrolladores de Amazon Managed Streaming for Apache Kafka.

Amazon MSK no admite la autenticación SASL/PLAIN.

Autenticación basada en roles de IAM

Puede utilizar IAM para autenticar la identidad de los clientes que se conectan al clúster de MSK. Si la autenticación de IAM está activa en tu clúster de MSK y no proporcionas un secreto para la autenticación, EventBridge se utilizará automáticamente la autenticación de IAM de forma predeterminada. Para crear e implementar políticas basadas en roles o usuarios de IAM, utilice la consola o la API de IAM. Para obtener más información, consulte IAM access control (Control de acceso de IAM) en la Guía para desarrolladores de Amazon Managed Streaming for Apache Kafka.

Para poder conectarte EventBridge al clúster de MSK, leer registros y realizar otras acciones necesarias, añade los siguientes permisos a la función de ejecución de tus tuberías.

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

Puede asignar estos permisos a un clúster, un tema y un grupo específicos. Para obtener más información, consulte Acciones de Amazon MSK para Kafka en la Guía para desarrolladores de Amazon Managed Streaming for Apache Kafka.

Autenticación TLS mutua

TLS mutua (mTLS) proporciona autenticación bidireccional entre el cliente y el servidor. El cliente envía un certificado al servidor para que el servidor verifique el cliente, mientras que el servidor envía un certificado al cliente para que el cliente verifique el servidor.

En el caso de Amazon MSK, EventBridge actúa como cliente. Configura un certificado de cliente (como secreto en Secrets Manager) para autenticarse EventBridge con los agentes de su clúster de MSK. El certificado de servidor debe estar firmado por una entidad de certificación que esté en el almacén de confianza del servidor. El clúster de MSK envía un certificado de servidor para autenticar EventBridge a los corredores. EventBridge El certificado del servidor debe estar firmado por una entidad emisora de certificados que se encuentre en el almacén de AWS confianza.

Amazon MSK no admite certificados de servidor autofirmados, ya que todos los agentes de Amazon MSK utilizan certificados públicos firmados por las CA de Amazon Trust Services, que son de confianza de forma predeterminada EventBridge .

Para obtener más información sobre mTLS para Amazon MSK, consulte Mutual TLS Authentication (Autenticación con TLS mutua) en la Guía para desarrolladores de Amazon Managed Streaming for Apache Kafka.

Configuración del secreto de mTLS

El secreto CLIENT_CERTIFICATE_TLS_AUTH requiere un campo de certificado y un campo de clave privada. Para una clave privada cifrada, el secreto requiere una contraseña de clave privada. El certificado y la clave privada deben estar en formato PEM.

> **Note**
>
> EventBridge admite los algoritmos de cifrado de clave privada PBES1 (pero no PBES2).

El campo de certificado debe contener una lista de certificados y debe comenzar por el certificado de cliente, seguido de cualquier certificado intermedio, y finalizar con el certificado raíz. Cada certificado debe comenzar en una nueva línea con la siguiente estructura:

```
-----BEGIN CERTIFICATE-----
        <certificate contents>
-----END CERTIFICATE-----
```

Secrets Manager admite secretos de hasta 65 536 bytes, que supone suficiente espacio para cadenas de certificados largas.

El formato de la clave privada debe ser PKCS #8, con la siguiente estructura:

```
-----BEGIN PRIVATE KEY-----
         <private key contents>
-----END PRIVATE KEY-----
```

Para una clave privada cifrada, utilice la siguiente estructura:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
          <private key contents>
-----END ENCRYPTED PRIVATE KEY-----
```

El siguiente ejemplo muestra el contenido de un secreto para la autenticación de mTLS mediante una clave privada cifrada. Para una clave privada cifrada, incluya la contraseña de la clave privada en el secreto.

```
{
 "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-----"
}
```

¿Cómo elegir EventBridge un bróker bootstrap

EventBridge elige un agente de arranque en función de los métodos de autenticación disponibles en su clúster y de si proporciona un secreto para la autenticación. Si proporciona un secreto para mTLS o SASL/SCRAM, elige EventBridge automáticamente ese método de autenticación. Si no proporciona un secreto, EventBridge elija el método de autenticación más seguro que esté activo en su clúster. El siguiente es el orden de prioridad en el que se EventBridge selecciona un intermediario, desde la autenticación más fuerte hasta la más débil:

- mTLS (secreto proporcionado para mTLS)
- SASL/SCRAM (secreto proporcionado para SASL/SCRAM)
- SASL IAM (no se proporciona secreto y la autenticación de IAM está activa)
- TLS no autenticada (no se proporciona secreto y la autenticación de IAM no está activa)
- Texto sin formato (no se proporciona secreto y tanto la autenticación de IAM como la TLS no autenticada no están activas)

> **Note**
>
> Si no EventBridge puede conectarse al tipo de corredor más seguro, no intentará conectarse a un tipo de corredor diferente (más débil). Si EventBridge quiere elegir un tipo de intermediario más débil, desactive todos los métodos de autenticación más seguros de su clúster.

Configuración de red

EventBridge debe tener acceso a los recursos de Amazon Virtual Private Cloud (Amazon VPC) asociados a su clúster de Amazon MSK.

- Para acceder a la VPC de su clúster de Amazon MSK, EventBridge puede utilizar el acceso saliente a Internet para las subredes de su fuente. Para las subredes públicas, debe ser una puerta de enlace NAT administrada. Para las subredes privadas, puede ser una puerta de enlace NAT o su propia NAT. Asegúrese de que la NAT tiene una dirección IP pública y puede conectarse a Internet.
- EventBridge Pipes también admite la entrega directa de eventos AWS PrivateLink, lo que le permite enviar eventos desde una fuente de eventos ubicada en un Amazon Virtual Private Cloud (Amazon VPC) a un destino de Pipes sin tener que atravesar la red pública de Internet. Puedes usar Pipes para sondear desde Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogestionado y Amazon MQ fuentes que residen en una subred privada sin necesidad de implementar una puerta de enlace a Internet, configurar reglas de firewall o configurar servidores proxy.

  Para configurar un punto de enlace de VPC, consulte Crear un punto de enlace de VPC en la Guía del usuario.AWS PrivateLink Para el nombre del servicio, seleccione. `com.amazonaws.region.pipes-data`

Configure sus grupos de seguridad de Amazon VPC con las siguientes reglas (como mínimo):

- Reglas de entrada: permita todo el tráfico en el puerto del agente de Amazon MSK para los grupos de seguridad especificados para su fuente.
- Reglas de salida: permiten todo el tráfico en el puerto 443 para todos los destinos. Permita todo el tráfico en el puerto del broker de Amazon MSK para los grupos de seguridad especificados para su fuente.

  Los puertos de bróker incluyen:

  - 9092 para texto sin formato
  - 9094 para TLS
  - 9096 para SASL
  - 9098 para IAM

> **Note**
>
> La configuración de Amazon VPC se puede detectar a través de la API de Amazon MSK. No tiene que configurarla durante la configuración.

ID del grupo de consumidores personalizable

Al configurar Apache Kafka como origen, puede especificar un ID de grupo de consumidores. Este ID de grupo de consumidores es un identificador existente para el grupo de consumidores de Apache Kafka al que desea que se una su canalización. Puede utilizar esta función para migrar cualquier configuración de procesamiento de registros de Apache Kafka en curso de otros consumidores a otra. EventBridge

Si especifica un ID de grupo de consumidores y hay otros sondeadores activos dentro de ese grupo de consumidores, Apache Kafka distribuirá los mensajes entre todos los consumidores. En otras palabras, EventBridge no recibe todos los mensajes relacionados con el tema de Apache Kafka. Si EventBridge quiere gestionar todos los mensajes del tema, desactive cualquier otro sondeo de ese grupo de consumidores.

Además, si especificas un ID de grupo de consumidores y Apache Kafka encuentra un grupo de consumidores válido existente con el mismo ID, EventBridge ignora el `StartingPosition` parámetro de tu canal. En su lugar, EventBridge comienza a procesar los registros de acuerdo con la compensación comprometida del grupo de consumidores. Si especificas un ID de grupo de consumidores y Apache Kafka no encuentra un grupo de consumidores existente, EventBridge configura tu fuente con el especificado. `StartingPosition`

El ID del grupo de consumidores que especifique debe ser único entre todos los orígenes de eventos de Apache Kafka. Tras crear una canalización con el ID de grupo de consumidores especificado, no puede actualizar este valor.

Escalado automático del origen de Amazon MSK

Al crear inicialmente una fuente de Amazon MSK, EventBridge asigna un consumidor para procesar todas las particiones del tema de Apache Kafka. Cada consumidor tiene varios procesadores que se ejecutan en paralelo para gestionar el aumento de las cargas de trabajo. Además, aumenta o reduce EventBridge automáticamente el número de consumidores en función de la carga de trabajo. Para conservar el orden de mensajes en cada partición, el número máximo de consumidores es un consumidor por partición en el tema.

En intervalos de un minuto, EventBridge evalúa el desfase de compensación por consumo de todas las secciones del tema. Si el retraso es demasiado alto, la partición recibe los mensajes más rápido de lo que EventBridge puede procesarlos. Si es necesario, EventBridge añade o elimina consumidores del tema. El proceso de escalado para agregar o eliminar consumidores se produce dentro de los tres minutos posteriores a la evaluación.

Si tu público objetivo está sobrecargado, EventBridge reduce el número de consumidores. Esta acción reduce la carga de trabajo de la canalización al reducir el número de mensajes que los consumidores pueden recuperar y enviar a la canalización.

Apache Kafka transmite como fuente

Apache Kafka es una plataforma de secuencia de eventos de código abierto que admite cargas de trabajo como canalizaciones de datos y análisis de streaming. Puede utilizar Amazon Managed Streaming for Apache Kafka (Amazon MSK) o un clúster de Apache Kafka autogestionado. En AWS terminología, un clúster autogestionado se refiere a cualquier clúster de Apache Kafka que no esté hospedado por. AWS Esto incluye tanto los clústeres que gestione usted mismo como los alojados por un proveedor externo, como Confluent CloudCloudKarafka, o Redpanda.

Para obtener más información sobre otras opciones de AWS alojamiento para su clúster, consulte Prácticas recomendadas para ejecutar Apache Kafka en el blog AWS sobre AWS macrodatos.

Apache Kafka como fuente funciona de forma similar a cuando se utiliza Amazon Simple Queue Service (Amazon SQS) o Amazon Kinesis. EventBridgesondea internamente los mensajes nuevos de la fuente y, a continuación, invoca al destino de forma sincrónica. EventBridge lee los mensajes por lotes y los proporciona a su función como carga útil de eventos. El tamaño máximo del lote es configurable. (El valor predeterminado es 100 mensajes).

En el caso de las fuentes basadas en Apache Kafka, EventBridge admite parámetros de control del procesamiento, como las ventanas de procesamiento por lotes y el tamaño del lote.

EventBridge envía el lote de mensajes del parámetro de evento cuando invoca la tubería. La carga de eventos contiene una matriz de mensajes. Cada elemento de la matriz contiene detalles del tema Apache Kafka y el identificador de partición de Apache Kafka, junto con una marca de tiempo y un mensaje codificado en base64.

Eventos de ejemplo

En el siguiente evento de ejemplo se muestra la información que recibe la canalización. Puede usar este evento para crear y filtrar sus patrones de eventos o para definir la transformación de entrada.

No todos los campos se pueden filtrar. Para obtener más información sobre los campos que puede filtrar, consulte ???.

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

Autenticación de clústeres de Apache Kafka

EventBridge Pipes admite varios métodos para autenticarse con su clúster Apache Kafka autogestionado. Asegúrese de configurar el clúster de Apache Kafka para que utilice uno de estos

métodos de autenticación admitidos. Para obtener más información acerca de la seguridad de Apache Kafka, consulte la sección Seguridad de la documentación de Apache Kafka.

Acceso mediante VPC

Si utiliza un entorno Apache Kafka autogestionado en el que solo los usuarios de Apache Kafka de su VPC tienen acceso a sus agentes de Apache Kafka, debe configurar Amazon Virtual Private Cloud (Amazon VPC) en el código fuente de Apache Kafka.

Autenticación SASL/SCRAM

EventBridge Pipes admite la autenticación simple y la autenticación por capa de seguridad/mecanismo de autenticación por respuesta a desafíos salados (SASL/SCRAM) con cifrado de seguridad de capa de transporte (TLS). EventBridge Pipes envía las credenciales cifradas para autenticarse en el clúster. Para obtener más información acerca de la autenticación SASL/SCRAM, consulte RFC 5802.

EventBridge Pipes admite la autenticación SASL/PLAIN con cifrado TLS. Con la autenticación SASL/PLAIN, EventBridge Pipes envía las credenciales como texto sin cifrar (sin cifrar) al servidor.

Para la autenticación SASL, almacene las credenciales de inicio de sesión como secreto en AWS Secrets Manager.

Autenticación TLS mutua

TLS mutua (mTLS) proporciona autenticación bidireccional entre el cliente y el servidor. El cliente envía un certificado al servidor para que el servidor verifique el cliente, mientras que el servidor envía un certificado al cliente para que el cliente verifique el servidor.

En Apache Kafka autogestionado, EventBridge Pipes actúa como cliente. Configura un certificado de cliente (como secreto en Secrets Manager) para autenticar a EventBridge Pipes con sus agentes de Apache Kafka. El certificado de servidor debe estar firmado por una entidad de certificación que esté en el almacén de confianza del servidor.

El clúster de Apache Kafka envía un certificado de servidor a EventBridge Pipes para autenticar a los agentes de Apache Kafka con Pipes. EventBridge El certificado de servidor puede ser un certificado de entidad de certificación pública o un certificado autofirmado o de entidad de certificación privada. El certificado de CA público debe estar firmado por una CA que se encuentre en el almacén de confianza de EventBridge Pipes. Para un certificado de CA privado autofirmado, se configura el

certificado de CA raíz del servidor (como secreto en Secrets Manager). EventBridge Pipes usa el certificado raíz para verificar los agentes de Apache Kafka.

Para obtener más información acerca de mTLS, consulte Presentación de la autenticación de TLS mutua para Amazon MSK como origen.

Configuración del secreto de certificado de cliente

El secreto CLIENT_CERTIFICATE_TLS_AUTH requiere un campo de certificado y un campo de clave privada. Para una clave privada cifrada, el secreto requiere una contraseña de clave privada. El certificado y la clave privada deben estar en formato PEM.

> **Note**
>
> EventBridge Pipes admite los algoritmos de cifrado de clave privada PBES1 (pero no PBES2).

El campo de certificado debe contener una lista de certificados y debe comenzar por el certificado de cliente, seguido de cualquier certificado intermedio, y finalizar con el certificado raíz. Cada certificado debe comenzar en una nueva línea con la siguiente estructura:

```
-----BEGIN CERTIFICATE-----
        <certificate contents>
-----END CERTIFICATE-----
```

Secrets Manager admite secretos de hasta 65 536 bytes, que supone suficiente espacio para cadenas de certificados largas.

El formato de la clave privada debe ser PKCS #8, con la siguiente estructura:

```
-----BEGIN PRIVATE KEY-----
         <private key contents>
-----END PRIVATE KEY-----
```

Para una clave privada cifrada, utilice la siguiente estructura:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
          <private key contents>
-----END ENCRYPTED PRIVATE KEY-----
```

El siguiente ejemplo muestra el contenido de un secreto para la autenticación de mTLS mediante una clave privada cifrada. Para una clave privada cifrada, incluya la contraseña de la clave privada en el secreto.

```
{
 "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-----"
}
```

Configuración del secreto de certificado de entidad de certificación raíz del servidor

Cree este secreto si sus agentes de Apache Kafka utilizan cifrado TLS con certificados firmados por una entidad de certificación privada. Puede utilizar el cifrado TLS para autenticación VPC, SASL/SCRAM, SASL/PLAIN o mTLS.

El secreto de certificado de entidad de certificación raíz del servidor requiere un campo que contenga el certificado de entidad de certificación raíz del agente de Apache Kafka en formato PEM. La estructura del secreto se muestra en el ejemplo siguiente.

```
{
     "certificate": "-----BEGIN CERTIFICATE-----
  MIID7zCCAtegAwIBAgIBADANBgkqhkiG9w0BAQsFADCBmDELMAkGA1UEBhMCVVMx
  EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxJTAjBgNVBAoT
  HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xOzA5BgNVBAMTMlN0YXJmaWVs
  ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dG...
```

```
-----END CERTIFICATE-----"
```

Configuración de red

Si utiliza un entorno Apache Kafka autogestionado que utiliza conectividad de VPC privada EventBridge , debe tener acceso a los recursos de Amazon Virtual Private Cloud (Amazon VPC) asociados a sus agentes de Apache Kafka.

- Para acceder a la VPC de su clúster de Apache Kafka, EventBridge puede utilizar el acceso saliente a Internet para las subredes de su fuente. Para las subredes públicas, debe ser una puerta de enlace NAT administrada. Para las subredes privadas, puede ser una puerta de enlace NAT o su propia NAT. Asegúrese de que la NAT tiene una dirección IP pública y puede conectarse a Internet.
- EventBridge Pipes también admite la entrega directa de eventos AWS PrivateLink, lo que te permite enviar eventos desde una fuente de eventos ubicada en un Amazon Virtual Private Cloud (Amazon VPC) a un destino de Pipes sin tener que atravesar la red pública de Internet. Puedes usar Pipes para sondear desde Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogestionado y Amazon MQ fuentes que residen en una subred privada sin necesidad de implementar una puerta de enlace a Internet, configurar reglas de firewall o configurar servidores proxy.

  Para configurar un punto de enlace de VPC, consulte Crear un punto de enlace de VPC en la Guía del usuario.AWS PrivateLink Para el nombre del servicio, seleccione. `com.amazonaws.region.pipes-data`

Configure sus grupos de seguridad de Amazon VPC con las siguientes reglas (como mínimo):

- Reglas de entrada: permita que todo el tráfico en el puerto del broker Apache Kafka esté destinado a los grupos de seguridad especificados para su origen.
- Reglas de salida: permiten todo el tráfico en el puerto 443 para todos los destinos. Permita todo el tráfico en el puerto del broker de Apache Kafka para los grupos de seguridad especificados para su origen.

  Los puertos de broker incluyen:

  - 9092 para texto sin formato
  - 9094 para TLS
  - 9096 para SASL

- 9098 para IAM

Escalado automático para consumidores con fuentes de Apache Kafka

Al crear inicialmente una fuente de Apache Kafka, EventBridge asigna un consumidor para procesar todas las particiones del tema de Kafka. Cada consumidor tiene varios procesadores que se ejecutan en paralelo para gestionar el aumento de las cargas de trabajo. Además, aumenta o reduce EventBridge automáticamente el número de consumidores en función de la carga de trabajo. Para conservar el orden de mensajes en cada partición, el número máximo de consumidores es un consumidor por partición en el tema.

En intervalos de un minuto, EventBridge evalúa el desfase de compensación por consumo de todas las secciones del tema. Si el retraso es demasiado alto, la partición recibe los mensajes más rápido de lo que EventBridge puede procesarlos. Si es necesario, EventBridge añade o elimina consumidores del tema. El proceso de escalado para agregar o eliminar consumidores se produce dentro de los tres minutos posteriores a la evaluación.

Si tu público objetivo está sobrecargado, EventBridge reduce el número de consumidores. Esta acción reduce la carga de trabajo de la función al reducir el número de mensajes que los consumidores pueden recuperar y enviar a la función.

Amazon Simple Queue Service como origen

Puede usar EventBridge Pipes para recibir registros de una cola de Amazon SQS. A continuación, si lo desea, puede filtrar o enriquecer estos registros antes de enviarlos a un destino disponible para su procesamiento.

Puede utilizar una tubería para procesar los mensajes de una cola de Amazon Simple Queue Service (Amazon SQS). EventBridge Las tuberías admiten colas estándar y colas FIFO (primero en entrar y primero en salir). Con Amazon SQS, puede descargar tareas de un componente de su aplicación enviándolas a una cola para, a continuación, procesarlas de forma asíncrona.

EventBridge sondea la cola e invoca la canalización de forma sincrónica con un evento que contiene los mensajes de la cola. EventBridge lee los mensajes por lotes e invoca tu canal una vez por cada lote. Cuando su canal procese correctamente un lote, EventBridge elimina sus mensajes de la cola.

De forma predeterminada, EventBridge sondea simultáneamente hasta 10 mensajes de la cola y envía ese lote a la canalización. Para evitar invocar a la canalización con un número de registros pequeño, puede indicar al origen del evento que almacene en búfer registros hasta 5 minutos

configurando una ventana de lote. Antes de invocar la tubería, EventBridge continúa sondeando los mensajes de la cola estándar de Amazon SQS hasta que ocurra una de estas cosas:

- La ventana de lote caduca.
- Se alcanza la cuota de tamaño de carga de invocación.
- Se alcanza el tamaño máximo del lote configurado.

> **Note**
>
> Si utilizas una ventana por lotes y tu cola de Amazon SQS tiene poco tráfico, es EventBridge posible que esperes hasta 20 segundos antes de invocar tu canalización. Esto sucederá incluso si establece una ventana de lote inferior a 20 segundos. Para las colas FIFO, los registros contienen atributos adicionales relacionados con la deduplicación y la secuenciación.

Cuando EventBridge lee un lote, los mensajes permanecen en la cola, pero permanecen ocultos durante el tiempo de espera de visibilidad de la cola. Si tu canal procesa el lote correctamente, EventBridge elimina los mensajes de la cola. De forma predeterminada, si la función detecta un error al procesar un lote, todos los mensajes de ese lote se vuelven a ver en la cola. Por este motivo, el código de su canalización debe ser capaz de procesar el mismo mensaje varias veces sin efectos secundarios no deseados. Puede modificar este comportamiento de reprocesamiento si incluye errores de elementos por lotes en la respuesta de la canalización. En el ejemplo siguiente se muestra un evento para un lote de dos mensajes.

Eventos de ejemplo

En el siguiente evento de ejemplo se muestra la información que recibe la canalización. Puede usar este evento para crear y filtrar sus patrones de eventos o para definir la transformación de entrada. No todos los campos se pueden filtrar. Para obtener más información sobre los campos que puede filtrar, consulte ???.

Cola estándar

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

Cola FIFO

```
[
  {
    "messageId": "11d6ee51-4cc7-4302-9e22-7cd8afdaadf5",
    "receiptHandle": "AQEBBX8nesZEXmkhsmZeyIE8iQAMig7qw...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1573251510774",
      "SequenceNumber": "18849496460467696128",
      "MessageGroupId": "1",
```

```
      "SenderId": "AIDAIO23YVJENQZJOL4VO",
      "MessageDeduplicationId": "1",
      "ApproximateFirstReceiveTimestamp": "1573251510774"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:fifo.fifo",
    "awsRegion": "us-east-2"
  }
]
```

Escalado y procesamiento

En el caso de las colas estándar, EventBridge utiliza un sondeo largo para sondear una cola hasta que se active. Cuando hay mensajes disponibles, EventBridge lee hasta cinco lotes y los envía a la tubería. Si los mensajes siguen disponibles, EventBridge aumenta el número de procesos que leen los lotes en hasta 300 instancias más por minuto. El número máximo de lotes que una canalización puede procesar simultáneamente es 1 000.

En el caso de las colas FIFO, EventBridge envía los mensajes a la canalización en el orden en que los recibe. Cuando envíe un mensaje a una cola FIFO, especifique un ID de grupo de mensajes. Amazon SQS facilita la entrega de mensajes del mismo grupo a EventBridge, en orden. EventBridge clasifica los mensajes recibidos en grupos y envía solo un lote a la vez para cada grupo. Si la canalización devuelve un error, intentará realizar todos los reintentos con los mensajes afectados antes de EventBridge recibir más mensajes del mismo grupo.

Configurar una cola para usarla con Pipes EventBridge

Cree una cola de SQS que sirva como origen para la canalización. A continuación, configura la cola para que tu canal tenga tiempo de procesar cada lote de eventos y, a medida que vaya aumentando la escala, pueda volver EventBridge a intentarlo en respuesta a errores de regulación.

Para permitir que su canalización tenga tiempo para procesar cada lote de registros, establezca el tiempo de espera de visibilidad de la cola de origen hasta al menos seis veces el tiempo de ejecución combinado de los componentes de destino y enriquecimiento de la canalización. El tiempo adicional permite volver EventBridge a intentarlo si la tubería se ralentiza mientras se procesa un lote anterior.

Si su canalización no puede procesar un mensaje en repetidas ocasiones, Amazon SQS podrá enviar dicho mensaje a una cola de mensajes fallidos. Cuando la pipa devuelve un error, lo EventBridge mantiene en espera. Después de que se supera el tiempo de espera de visibilidad,

EventBridge recibe el mensaje de nuevo. Para enviar mensajes a una segunda cola después de recibir varias veces, configure una cola de mensajes fallidos en la cola de origen.

> **Note**
>
> Asegúrese de configurar la cola de mensajes fallidos en la cola de origen y no en la canalización. La cola de mensajes fallidos que configure en una canalización se utiliza para la cola de invocación asíncrona de la canalización, no para las colas de origen de eventos.

Si la canalización devuelve un error o no puede invocarse porque está en la máxima simultaneidad, es posible que el procesado tenga éxito tras algunos intentos adicionales. Para dar a los mensajes más oportunidades de ser procesados antes de enviarlos a la cola de mensajes fallidos, establezca el `maxReceiveCount` de la política de redirección de la cola de origen a como mínimo 5.

Informes de fallos de elementos de lote

Cuando EventBridge consume y procesa datos en streaming desde una fuente, de forma predeterminada, selecciona el número de secuencia más alto de un lote, pero solo cuando el lote es un éxito total. Para evitar el reprocesamiento de los mensajes procesados correctamente en un lote con errores, puede configurar el enriquecimiento o el destino para que devuelva un objeto que indique qué mensajes se han procesado correctamente y cuáles no. Esto se denomina respuesta parcial por lotes.

Para obtener más información, consulte ???.

Condiciones de éxito y fracaso

Si devuelve alguno de los siguientes datos, considerará EventBridge que el lote se ha realizado correctamente:

- Una lista `batchItemFailure` vacía
- Una lista `batchItemFailure` nula
- Una `EventResponse` vacía
- Una `EventResponse` nula

Si devuelve alguna de las siguientes opciones, EventBridge considerará el lote como un error total:

- Una cadena `itemIdentifier` vacía

- Una `itemIdentifier` nula
- Un `itemIdentifier` con un mal nombre de clave

EventBridge Los reintentos fallan en función de su estrategia de reintentos.

Filtrado EventBridge de Amazon Pipes

Con EventBridge Pipes, puede filtrar los eventos de una fuente determinada y procesar solo un subconjunto de ellos. Este filtrado funciona de la misma manera que el filtrado en un bus de EventBridge eventos o en un mapeo de origen de eventos de Lambda, mediante patrones de eventos. Para obtener más información acerca de patrones de eventos, consulte ???.

Un objeto `FilterCriteria` de criterios de filtro es una estructura que consta de una lista de filtros (`Filters`). Cada filtro es una estructura que define un patrón de filtrado (`Pattern`). Un `Pattern` es una representación de cadenas de una regla de filtro de JSON. Un objeto `FilterCriteria` es similar al siguiente ejemplo:

```
{
  "Filters": [
    {"Pattern": "{ \"Metadata1\": [ rule1 ], \"data\": { \"Data1\": [ rule2 ] }}"
    }
  ]
}
```

Para mayor claridad, este es el valor del `Pattern` del filtro ampliado en JSON no cifrado:

```
{
  "Metadata1": [ pattern1 ],
  "data": {"Data1": [ pattern2 ]}
}
```

Las partes principales de un objeto `FilterCriteria` son las propiedades de metadatos y las propiedades de datos.

- Las propiedades de metadatos son los campos del objeto de eventos. En el ejemplo, `FilterCriteria.Metadata1` hace referencia a una propiedad de metadatos.
- Las propiedades de datos son los campos del cuerpo del evento. En el ejemplo, `FilterCriteria.Data1` hace referencia a una propiedad de datos.

Por ejemplo, supongamos que su flujo de Kinesis contiene un evento como este:

```
{
  "kinesisSchemaVersion": "1.0",
  "partitionKey": "1",
  "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
  "data": {"City": "Seattle",
    "State": "WA",
    "Temperature": "46",
    "Month": "December"
  },
  "approximateArrivalTimestamp": 1545084650.987
}
```

Cuando el evento fluya por la canalización, tendrá el siguiente aspecto con el campo `data` codificado en 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"
}
```

Las propiedades de metadatos del evento de Kinesis son cualquier campo fuera del objeto `data`, como `partitionKey` o `sequenceNumber`.

Las propiedades de datos del evento de Kinesis son cualquier campo dentro del objeto `data`, como `City` o `Temperature`.

Al filtrar para obtener coincidencias con este evento, puede usar filtros en los campos decodificados. Por ejemplo, para filtrar en `partitionKey` y `City`, usaría el siguiente filtro:

```
{
  "partitionKey": [
    "1"
  ],
  "data": {
    "City": [
      "Seattle"
    ]
  }
}
```

Al crear filtros de eventos, EventBridge Pipes puede acceder al contenido del evento. Este contenido tiene un objeto JSON oculto, como el campo `body` de Amazon SQS, o codificado en base64, como el campo `data` de Kinesis. Si los datos son un objeto JSON válido, las plantillas de entrada o las rutas JSON de los parámetros de destino pueden hacer referencia al contenido directamente. Por ejemplo, si un origen de eventos de Kinesis es un objeto JSON válido, puede hacer referencia a una variable mediante `<$.data.someKey>`.

Al crear patrones de eventos, puede filtrar en función de los campos enviados por la API de origen y no de los campos agregados por la operación de sondeo. Los siguientes campos no se pueden usar en los patrones de eventos:

- `awsRegion`
- `eventSource`
- `eventSourceARN`
- `eventVersion`
- `eventID`
- `eventName`
- `invokeIdentityArn`
- `eventSourceKey`

Campos de mensaje y datos

Cada fuente EventBridge de Pipe contiene un campo que contiene el mensaje o los datos principales. Los denominamos campos mensaje o campos datos. Estos campos son especiales porque su contenido puede tener un objeto JSON oculto o estar codificado en base64, pero cuando son objetos JSON válidos, se pueden filtrar con patrones JSON como si el cuerpo no estuviera

oculto. El contenido de estos campos también se puede utilizar sin problemas en los transformadores de entrada.

Filtrado correcto de mensajes de Amazon SQS

Si un mensaje de Amazon SQS no cumple los criterios de filtrado, lo elimina EventBridge automáticamente de la cola. No tiene que eliminar manualmente estos mensajes en Amazon SQS.

Para Amazon SQS, el `body` del mensaje puede ser cualquier cadena. Sin embargo, esto puede ser problemático si los `FilterCriteria` esperan que el formato JSON del `body` sea válido. El escenario inverso también es problemático: si el `body` del mensaje entrante está en un formato JSON válido pero los criterios de filtro esperan que el `body` sea una cadena sin formato, puede producirse un comportamiento no deseado.

Para evitar este problema, asegúrese de que el formato del `body` de los `FilterCriteria` coincida con el formato esperado del `body` de los mensajes que recibe de la cola. Antes de filtrar los mensajes, evalúa EventBridge automáticamente el formato del mensaje entrante `body` y el patrón de filtrado. `body` Si hay una discrepancia, borra el EventBridge mensaje. En la siguiente tabla se resume esta evaluación:

Formato del **body** del mensaje entrante	Formato del **body** del patrón de filtro	Acción resultante
Cadena sin formato	Cadena sin formato	EventBridge filtra en función de sus criterios de filtrado.
Cadena sin formato	Sin patrón de filtro para las propiedades de datos	EventBridge filtra (solo en las demás propiedades de los metadatos) en función de sus criterios de filtrado.
Cadena sin formato	JSON válido	EventBridge elimina el mensaje.
JSON válido	Cadena sin formato	EventBridge deja caer el mensaje.
JSON válido	Sin patrón de filtro para las propiedades de datos	EventBridge filtra (solo en las demás propiedades de los

Formato del **body** del mensaje entrante	Formato del **body** del patrón de filtro	Acción resultante
		metadatos) en función de sus criterios de filtrado.
JSON válido	JSON válido	EventBridge filtra en función de sus criterios de filtrado.

Si no los incluyes `body` como parte de tu lista`FilterCriteria`, EventBridge omite esta verificación.

Filtrado correcto de mensajes de Kinesis y DynamoDB

Una vez que los criterios de filtro procesan un registro de Kinesis o DynamoDB, el iterador de flujos supera este registro. Si el registro no cumple los criterios de filtro, no tiene que eliminarlo manualmente del origen de eventos. Tras el periodo de retención, Kinesis y DynamoDB eliminan automáticamente estos registros antiguos. Si quiere que los registros se eliminen antes, consulte Changing the Data Retention Period (Cambiar el periodo de retención de datos).

Para filtrar correctamente los eventos de los orígenes de eventos de flujos, el formato JSON del campo de datos y de los criterios de filtro del campo de datos debe ser válido. (Para Kinesis, el campo de datos es `data`. Para DynamoDB, el campo de datos es `dynamodb`). Si alguno de los campos no tiene un formato JSON válido, borra EventBridge el mensaje o genera una excepción. En la siguiente tabla se resume el comportamiento específico:

Formato de los datos entrantes (**`data`** o **`dynamodb`**)	Formato del patrón de filtro para las propiedades de datos	Acción resultante
JSON válido	JSON válido	EventBridge filtra en función de tus criterios de filtrado.
JSON válido	Sin patrón de filtro para las propiedades de datos	EventBridge filtra (solo en las demás propiedades de los metadatos) en función de sus criterios de filtrado.

Formato de los datos entrantes (**`data`** o **`dynamodb`**)	Formato del patrón de filtro para las propiedades de datos	Acción resultante
JSON válido	No JSON	EventBridge genera una excepción en el momento de la canalización o actualización. El formato JSON del patrón de filtro de las propiedades de datos debe ser válido.
No JSON	JSON válido	EventBridge elimina el registro.
No JSON	Sin patrón de filtro para las propiedades de datos	EventBridge filtra (solo en las demás propiedades de los metadatos) en función de sus criterios de filtrado.
No JSON	No JSON	EventBridge genera una excepción en el momento de la creación o actualización de la tubería. El formato JSON del patrón de filtro de las propiedades de datos debe ser válido.

Filtrado correcto de mensajes de Amazon Managed Streaming para Apache Kafka, Apache Kafka autoadministrado y Amazon MQ

En el caso de los orígenes de Amazon MQ, el campo de mensaje es `data`. En el caso de los orígenes de Apache Kafka (Amazon MSK y Apache Kafka autoadministrado), hay dos campos de mensaje: `key` y `value`.

EventBridge elimina los mensajes que no coinciden con todos los campos incluidos en el filtro. En el caso de Apache Kafka, EventBridge compila las compensaciones de los mensajes coincidentes y no coincidentes después de invocar correctamente la función. En Amazon MQ, EventBridge reconoce

los mensajes coincidentes después de invocar correctamente la función y reconoce los mensajes no coincidentes al filtrarlos.

Los mensajes de Apache Kafka y Amazon MQ deben ser cadenas codificadas en UTF-8, cadenas simples o en formato JSON. Esto se debe a que EventBridge decodifica las matrices de bytes de Apache Kafka y Amazon MQ en UTF-8 antes de aplicar los criterios de filtrado. Si sus mensajes utilizan otra codificación, como UTF-16 o ASCII, o si el formato del mensaje no coincide con el formato, procesa únicamente los filtros de metadatos. `FilterCriteria` EventBridge En la siguiente tabla se resume el comportamiento específico:

Formato del mensaje entrante (**`data`** o **`key`** y **`value`**)	Formato del patrón de filtro para las propiedades del mensaje	Acción resultante
Cadena sin formato	Cadena sin formato	EventBridge filtra en función de sus criterios de filtrado.
Cadena sin formato	Sin patrón de filtro para las propiedades de datos	EventBridge filtra (solo en las demás propiedades de los metadatos) en función de sus criterios de filtrado.
Cadena sin formato	JSON válido	EventBridge filtra (solo en las demás propiedades de los metadatos) en función de sus criterios de filtrado.
JSON válido	Cadena sin formato	EventBridge filtra (solo en las demás propiedades de los metadatos) en función de sus criterios de filtrado.
JSON válido	Sin patrón de filtro para las propiedades de datos	EventBridge filtra (solo en las demás propiedades de los metadatos) en función de sus criterios de filtrado.

Formato del mensaje entrante (**`data`** o **`key`** y **`value`**)	Formato del patrón de filtro para las propiedades del mensaje	Acción resultante
JSON válido	JSON válido	EventBridge filtra en función de sus criterios de filtrado.
Cadena no codificada con UTF-8	JSON, cadena sin formato o sin patrón	EventBridge filtra (solo en las demás propiedades de los metadatos) en función de sus criterios de filtrado.

Diferencias entre Lambda ESM y Pipes EventBridge

Al filtrar eventos, Lambda ESM y EventBridge Pipes funcionan generalmente de la misma manera. La principal diferencia es que el campo `eventSourceKey` no está presente en las cargas de ESM.

Enriquecimiento de eventos de Amazon EventBridge Pipes

Con el paso de enriquecimiento de EventBridge Pipes, puede mejorar los datos del origen antes de enviarlos al destino. Por ejemplo, es posible que reciba eventos de creación de tíckets que no incluyan todos los datos de la entrada. Con el enriquecimiento, puede hacer que una función de Lambda llame a la API `get-ticket` para obtener todos los detalles del tícket. Luego, EventBridge Pipes puede enviar esa información a un destino.

Puede configurar los siguientes enriquecimientos al configurar una canalización en EventBridge:

- Destino de la API
- Amazon API Gateway
- Función de Lambda
- Máquina de estado de Step Functions

> **Note**
>
> EventBridge Pipes solo admite flujos de trabajo rápidos como enriquecimientos.

EventBridge invoca los enriquecimientos de forma sincrónica porque debe esperar una respuesta del enriquecimiento antes de invocar el destino.

Las respuestas de enriquecimiento están limitadas a un tamaño máximo de 6 MB.

También puede transformar los datos que recibe del origen antes de enviarlos para su enriquecimiento. Para obtener más información, consulte ???.

Filtrar eventos mediante el enriquecimiento

EventBridge Pipes transfiere las respuestas de enriquecimiento directamente al destino configurado. Incluye las respuestas de matriz para los destinos que admiten lotes. Para obtener más información sobre este comportamiento por lotes, consulte ???. También puede usar el enriquecimiento como filtro y transferir menos eventos de los que recibió del origen. Si no quiere invocar el destino, devuelva una respuesta vacía, como `""`, `{}` o `[]`.

> **Note**
>
> Si quiere invocar el destino con una carga vacía, devuelva una matriz con un JSON `[{}]` vacío.

Invocación de enriquecimientos

EventBridge invoca los enriquecimientos de forma sincrónica (tipo de invocación establecida en `REQUEST_RESPONSE`) porque debe esperar una respuesta del enriquecimiento antes de invocar el destino.

> **Note**
>
> En el caso de las máquinas de estados Step Functions, EventBridge solo admite los flujos de trabajo rápidos como enriquecimientos, ya que se pueden invocar de forma sincrónica.

Objetivos de Amazon EventBridge Pipes

Puede enviar los datos de la canalización a un destino específico. Puede configurar los siguientes objetivos al configurar una tubería en EventBridge:

- Destino de la API

- API Gateway
- Cola de trabajos por lotes
- CloudWatch grupo de registros
- Tarea de ECS
- Bus de eventos en la misma cuenta y región
- Flujo de entrega de Firehose
- Plantilla de evaluación del inspector
- Flujo de Kinesis
- Función de Lambda (SYNC o ASYNC)
- Consultas de API de datos de clústeres de Redshift
- SageMaker Tubería
- Tema de Amazon SNS (no se admiten temas FIFO de Amazon SNS)
- Cola de Amazon SQS
- Máquina de estado de Step Functions
  - Flujos de trabajo rápido (SYNC o ASYNC)
  - Flujos de trabajo estándar (ASYNC)
- Timestream para LiveAnalytics mesa

Parámetros de destino

Algunos servicios de destino no envían la carga útil del evento al destino, sino que tratan el evento como un desencadenante para invocar una API específica. EventBridge usa el `PipeTargetParameters`para especificar qué información se envía a esa API. Estos incluyen los siguientes:

- Destinos de la API (Los datos enviados a un destino de la API deben coincidir con la estructura de la API. Debe usar el objeto `InputTemplate` para asegurarse de que los datos estén estructurados correctamente. Si desea incluir la carga del evento original, haga referencia a ella en el objeto `InputTemplate`.)
- API Gateway (Los datos enviados a API Gateway deben coincidir con la estructura de la API. Debe usar el objeto `InputTemplate` para asegurarse de que los datos estén estructurados correctamente. Si desea incluir la carga del evento original, haga referencia a ella en el objeto `InputTemplate`.)

- `PipeTargetRedshiftDataParameters` (Clústeres de API de datos de Amazon Redshift)
- `PipeTargetSageMakerPipelineParameters`(Amazon SageMaker Runtime Model Building Pipelines)
- `PipeTargetBatchJobParameters` (AWS Batch)

> **Note**
>
> EventBridge no admite toda la sintaxis de rutas JSON y la evalúa en tiempo de ejecución. La sintaxis admitida incluye:
>
> - notación de puntos (por ejemplo, `$.detail`)
> - guiones
> - guiones bajos
> - Caracteres alfanuméricos
> - índices de matrices
> - caracteres comodín (*)

Parámetros de ruta dinámicos

EventBridge Los parámetros de destino de Pipes admiten la sintaxis de ruta JSON dinámica opcional. Puede usar esta sintaxis para especificar rutas JSON en lugar de valores estáticos (por ejemplo, `$.detail.state`). El valor completo debe ser una ruta JSON, no solo una parte de ella. Por ejemplo, `RedshiftParameters.Sql` puede ser `$.detail.state`, pero no puede ser `"SELECT * FROM $.detail.state"`. Estas rutas se sustituyen dinámicamente en tiempo de ejecución por datos de la propia carga del evento en la ruta especificada. Los parámetros de ruta dinámicos no pueden hacer referencia a valores nuevos o transformados que resulten de la transformación de entrada. La sintaxis admitida para las rutas JSON con parámetros dinámicos es la misma que cuando se transforma la entrada. Para obtener más información, consulte ???.

La sintaxis dinámica se puede utilizar en todos los campos de cadena y sin enumeración de todos los parámetros de enriquecimiento y destino de EventBridge Pipes, excepto:

- `PipeTargetCloudWatchLogsParameters.LogStreamName`
- `PipeTargetEventBridgeEventBusParameters.EndpointId`
- `PipeEnrichmentHttpParameters.HeaderParameters`

- `PipeTargetHttpParameters.HeaderParameters`

Por ejemplo, para establecer el objetivo `PartitionKey` de un canal de Kinesis en una clave personalizada del evento de origen, defina el. KinesisTargetParameter PartitionKeypara:

- `"$.data.someKey"` para un origen de Kinesis
- `"$.body.someKey"` para un origen de Amazon SQS

A continuación, si la carga útil del evento es una cadena JSON válida, por ejemplo`{"someKey":"someValue"}`, EventBridge extrae el valor de la ruta JSON y lo utiliza como parámetro de destino. En este ejemplo, EventBridge establecería Kinesis en "`PartitionKeySomeValue`».

Permisos

Para realizar llamadas a la API en los recursos de su propiedad, EventBridge Pipes necesita el permiso adecuado. EventBridge PiPEs usa la función de IAM que especifiques en el proceso de enriquecimiento y dirige las llamadas mediante el principio de IAM. `pipes.amazonaws.com`

Invocación de destinos

EventBridge tiene las siguientes formas de invocar un objetivo:

- Sincrónicamente (tipo de invocación establecido en`REQUEST_RESPONSE`): EventBridge espera una respuesta del objetivo antes de continuar.
- De forma asíncrona (tipo de invocación establecido en`FIRE_AND_FORGET`): EventBridge no espera una respuesta para continuar.

De forma predeterminada, en el caso de las canalizaciones con fuentes ordenadas, EventBridge invoca los destinos de forma sincrónica, ya que se necesita una respuesta del destino antes de pasar al siguiente evento.

Si una fuente no hace cumplir el orden, como una cola estándar de Amazon SQS, EventBridge puede invocar un destino compatible de forma sincrónica o asíncrona.

Con las funciones de Lambda y las máquinas de estado Step Functions, puede configurar el tipo de invocación.

> **Note**
>
> Para las máquinas de estado Step Functions, los flujos de trabajo estándar se deben invocar de forma asíncrona.

EventBridge Canaliza las especificaciones del objetivo

AWS Batch colas de trabajos

Todos los AWS Batch `submitJob` parámetros se configuran de forma explícita con la carga útil del evento entrante y`BatchParameters`, como ocurre con todos los parámetros de Pipe, pueden ser dinámicos mediante una ruta JSON a la carga útil del evento entrante.

CloudWatch Grupo de registros

Tanto si utiliza un transformador de entrada como si no, la carga del evento se utiliza como mensaje de registro. Puede configurar el `Timestamp` (o el `LogStreamName` explícito del destino) mediante `CloudWatchLogsParameters` en `PipeTarget`. Como todos los parámetros de canalización, estos parámetros pueden ser dinámicos cuando se usa una ruta JSON con la carga del evento entrante.

Tarea de Amazon ECS

Todos los parámetros `runTask` e Amazon ECS se configuran de forma explícita mediante `EcsParameters`. Como todos los parámetros de canalización, estos parámetros pueden ser dinámicos cuando se usa una ruta JSON con la carga del evento entrante.

Funciones de Lambda y flujos de trabajo de Step Functions

Lambda y Step Functions no tienen una API de lotes. Para procesar lotes de eventos de un origen de canalización, el lote se convierte en una matriz JSON y se transfiere como entrada al destino de Lambda o Step Functions. Para obtener más información, consulte ???.

Timestream para LiveAnalytics tabla

A la hora de especificar una LiveAnalytics tabla Timestream de formulario como objetivo de tubería, se incluyen las siguientes:

- Actualmente, las transmisiones de Apache Kafka (incluidas las de proveedores externos Amazon MSK o de terceros) no se admiten como fuente canalizada.
- Si ha especificado una DynamoDB transmisión Kinesis o como fuente de canalización, debe especificar el número de reintentos.

  Para obtener más información, consulte ???.

Procesamiento por lotes y simultaneidad de Amazon EventBridge Pipes

Comportamiento de procesamiento por lotes

EventBridge Pipes admite el procesamiento por lotes desde el origen hasta los destinos que lo admiten. Además, se admite el procesamiento por lotes y el enriquecimiento para AWS Lambda y AWS Step Functions. Como los diferentes servicios admiten diferentes niveles de procesamiento por lotes, no se puede configurar una canalización con un tamaño de lote mayor que el que admite el destino. Por ejemplo, las fuentes de streaming de Amazon Kinesis admiten un tamaño de lote máximo de 10 000 registros, pero Amazon Simple Queue Service admite un máximo de 10 mensajes por lote como destino. Por lo tanto, una canalización desde un flujo de Kinesis a una cola de Amazon SQS puede tener un tamaño máximo de lote configurado en el origen de 10.

Si configura una canalización con un enriquecimiento o un destino que no admite el procesamiento por lotes, no podrá activar el procesamiento por lotes en la origen.

Cuando se activa el procesamiento por lotes en el origen, las matrices de registros JSON pasan por la canalización y, a continuación, se asignan a la API del lote de un enriquecimiento o destino compatible. Los transformadores de entrada se aplican por separado a cada registro JSON individual de la matriz, no a la matriz en su conjunto. Para ver ejemplos de estas matrices, consulte ??? y seleccione un origen específico. Las canalizaciones utilizarán la API de lote para el enriquecimiento o el destino admitidos, incluso si el tamaño del lote es 1. Si el enriquecimiento o el destino no tiene una API de lote, pero recibe cargas completas de JSON, como Lambda y Step Functions, toda la matriz JSON se envía en una sola solicitud. La solicitud se enviará como una matriz JSON incluso si el tamaño del lote es 1.

Si una canalización está configurada para el procesamiento por lotes en el origen y el destino admite el procesamiento por lotes, puedes devolver una matriz de elementos JSON a partir de su enriquecimiento. Esta matriz puede incluir una matriz más corta o más larga que el origen original.

Sin embargo, si la matriz es mayor que el tamaño de lote admitido por el destino, la canalización no invocará el destino.

Destinos procesables por lotes admitidos

Destino	Tamaño máximo de lote
CloudWatch Registros	10 000
EventBridge autobús de eventos	10
Firehose Stream	500
Flujo de Kinesis	500
Función de Lambda	definido por el cliente
Máquina de estado de Step Functions	definido por el cliente
Tema de Amazon SNS	10
Cola de Amazon SQS	10

Los siguientes enriquecimientos y destinos reciben la carga completa del evento por lotes para su procesamiento y están limitados por el tamaño total de la carga del evento, más que por el tamaño del lote:

- Máquina de estado de Step Functions (262 144 caracteres)
- Función de Lambda (6 MB)

Fallo de lote parcial

Para Amazon SQS y fuentes de streaming, como Kinesis y DynamoDB, Pipes admite la gestión parcial de errores por lotes de los EventBridge errores de destino. Si el destino admite el procesamiento por lotes y solo una parte del lote tiene éxito, vuelve a intentar agrupar EventBridge automáticamente el resto de la carga útil. En el caso del contenido más up-to-date enriquecido, este reintento se realiza a lo largo de todo el proceso, incluida la reinvocación de cualquier enriquecimiento configurado.

No se admite la gestión de fallos parciales de lotes.

Para los destinos Lambda y Step Functions, también puede especificar un error parcial devolviendo una carga con una estructura definida desde el destino. Esto indica los eventos que se deben volver a intentar.

Ejemplo de estructura de carga con fallo parcial

```
{
  "batchItemFailures": [
    {
      "itemIdentifier": "id2"
    },
    {
      "itemIdentifier": "id4"
    }
]
```

En el ejemplo, el `itemIdentifier` coincide con el ID de los eventos gestionados por su destino desde su origen original. Para Amazon SQS, este es el `messageId`. Para Kinesis y DynamoDB, este es el `eventID`. Para que EventBridge Pipes pueda gestionar adecuadamente los fallos parciales de los lotes producidos por los objetivos, estos campos deben incluirse en cualquier carga útil de matriz que devuelva el enriquecimiento.

Comportamiento de rendimiento y simultaneidad

Cada evento o lote de eventos recibido por una canalización que se dirige a un enriquecimiento o un destino se considera una ejecución de canalización. Una canalización en estado `STARTED` sondea continuamente los eventos desde el origen y se amplía o reduce en función de la acumulación de trabajo disponible y de los ajustes de procesamiento por lotes configurados.

Para conocer las cuotas de ejecuciones simultáneas de canalizaciones y el número de canalizaciones por cuenta y región, consulte ???.

De forma predeterminada, una sola canalización se escalará hasta el siguiente número máximo de ejecuciones simultáneas, en función del origen:

- DynamoDB: las ejecuciones simultáneas pueden ascender tanto como el `ParallelizationFactor` configurado en la canalización multiplicado por el número de particiones del flujo.

- Apache Kafka: las ejecuciones simultáneas pueden ascender tanto como el número de particiones sobre el tema, hasta 1 000.
- Kinesis: las ejecuciones simultáneas pueden ascender tanto como el `ParallelizationFactor` configurado en la canalización multiplicado por el número de particiones del flujo.
- Amazon MQ: 5
- Amazon SQS: 1250

Si necesita un rendimiento máximo de sondeo o límites de simultaneidad más altos, [póngase en contacto con el servicio de asistencia](#).

> **Note**
>
> Los límites de ejecución se consideran limitaciones de seguridad máximas. Si bien el sondeo no se limita por debajo de estos valores, es posible que una canalización o una cuenta superen estos valores recomendados.

Las ejecuciones de canalizaciones están limitadas a un máximo de 5 minutos, incluidos el enriquecimiento y el procesamiento objetivo. Este límite no se puede aumentar.

Las canalizaciones con orígenes estrictamente ordenados (como las colas FIFO de Amazon SQS, los flujos de Kinesis y DynamoDB, o los temas de Apache Kafka) están limitadas aún más en cuanto a la simultaneidad por la configuración del origen, por ejemplo, el número de ID de grupos de mensajes para las colas FIFO o el número de particiones para las colas de Kinesis. Como el orden está estrictamente garantizado dentro de estas restricciones, una canalización con un origen ordenado no puede superar esos límites de simultaneidad.

Transformación de entradas de Amazon EventBridge Pipes

Amazon EventBridge Pipes admite transformadores de entrada opcionales al transferir datos al enriquecimiento y al destino. Puede utilizar los transformadores de entrada para modificar la forma de la carga de entrada del evento JSON a fin de satisfacer las necesidades del servicio de enriquecimiento o de destino. En el caso de Amazon API Gateway y los destinos de API, así es como se configura el evento de entrada según el modelo RESTful de la API. Los transformadores de entrada se modelan como un parámetro `InputTemplate`. Pueden ser texto libre, una ruta JSON a la carga del evento o un objeto JSON que incluye rutas JSON en línea a la carga del evento.

Para el enriquecimiento, la carga del evento proviene del origen. En el caso de los destinos, la carga del evento es la que se devuelve desde el enriquecimiento, si hay alguna configurada en la canalización. Además de los datos específicos del servicio en la carga del evento, puede usar variables reservadas en su `InputTemplate` para hacer referencia a datos para la canalización.

Para acceder a los elementos de una matriz, use la notación entre corchetes.

> **Note**
>
> EventBridge no admite toda la sintaxis de ruta JSON y la evalúa en tiempo de ejecución. La sintaxis admitida incluye:
>
> - notación de puntos (por ejemplo, `$.detail`)
> - guiones
> - guiones bajos
> - Caracteres alfanuméricos
> - índices de matrices
> - caracteres comodín (*)

Los siguientes son ejemplos de parámetros `InputTemplate` que hacen referencia a una carga de eventos de Amazon SQS:

Cadena estática

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

Ruta JSON

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

Cadena dinámica

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

JSON estático

```
InputTemplate: >
```

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

JSON dinámico

```
InputTemplate: >
{
  "key1": "value1"
  "key2": <$.body.key>,
  "d": <aws.pipes.event.ingestion-time>
}
```

Uso de la notación entre corchetes para acceder a los elementos de una matriz:

```
InputTemplate: >
{
  "key1": "value1"
  "key2": <$.body.Records[3]>,
  "d": <aws.pipes.event.ingestion-time>
}
```

> **Note**
>
> EventBridge reemplaza a los transformadores de entrada en tiempo de ejecución para garantizar una salida JSON válida. Por este motivo, debe colocar comillas alrededor de las variables que hagan referencia a los parámetros de ruta JSON, pero no alrededor de las variables que hagan referencia a objetos o matrices JSON.

Variables reservadas

Las plantillas de entrada pueden utilizar las siguientes variables reservadas:

- `<aws.pipes.pipe-arn>`: el Nombre de recurso de Amazon (ARN) para la canalización.
- `<aws.pipes.pipe-name>`: el nombre de la canalización.
- `<aws.pipes.source-arn>`: el ARN del origen del evento de la canalización.

- `<aws.pipes.enrichment-arn>`: el ARN del enriquecimiento de la canalización.
- `<aws.pipes.target-arn>`: el ARN del destino de la canalización.
- `<aws.pipes.event.ingestion-time>`: la hora a la que el transformador de entrada recibió el evento. Se trata de una marca de tiempo ISO 8601. Esta hora es diferente para el transformador de entrada de enriquecimiento y el transformador de entrada de destino, según el momento en que el enriquecimiento haya completado el procesamiento del evento.
- `<aws.pipes.event>`: el evento recibido pro el transformador de entrada.

  En el caso de un transformador de entrada de enriquecimiento, este es el evento del origen. Contiene la carga original del origen, además de metadatos específicos de los servicios adicionales. Consulte los temas en ??? para obtener ejemplos específicos de este servicio.

  En el caso de un transformador de entrada de destino, este es el evento devuelto por el enriquecimiento, si hay alguno configurado, sin metadatos adicionales. Por lo tanto, una carga devuelta por el enriquecimiento puede no ser un valor JSON. Si no se configura ningún enriquecimiento en la canalización, se trata del evento del origen con los metadatos.
- `<aws.pipes.event.json>`: igual que `aws.pipes.event`, pero la variable solo tiene un valor si la carga original, ya sea del origen o devuelta por el enriquecimiento, es un valor JSON. Si la canalización tiene un campo codificado, como el campo `body` de Amazon SQS o el campo `data` de Kinesis, esos campos se decodifican y se convierten en un valor JSON válido. Como no está oculta, la variable solo se puede usar como valor para un campo JSON. Para obtener más información, consulte ???.

Ejemplos de transformación de entrada

El siguiente es un ejemplo de un evento de Amazon EC2 que podemos usar como evento de muestra.

```
{
  "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"
  }
}
```

Usemos el siguiente valor JSON como transformador.

```
{
  "instance" : <$.detail.instance-id>,
  "state": <$.detail.state>,
  "pipeArn" : <aws.pipes.pipe-arn>,
  "pipeName" : <aws.pipes.pipe-name>,
  "originalEvent" : <aws.pipes.event.json>
}
```

Se generará la siguiente salida:

```
{
  "instance" : "i-0123456789",
  "state": "RUNNING",
  "pipeArn" : "arn:aws:pipe:us-east-1:123456789012:pipe/example",
  "pipeName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

Análisis implícito de datos del cuerpo

Los siguientes campos de la carga entrante pueden ser valores JSON ocultos, como el objeto `body` de Amazon SQS, o valores codificados en base64, como el objeto `data` de Kinesis. Tanto para el filtrado como para la transformación de entradas, EventBridge transforma estos campos en un valor JSON válido para que se pueda hacer referencia a los subvalores directamente. Por ejemplo, <$.data.someKey> para Kinesis.

Para que el destino reciba la carga original sin ningún metadato adicional, utilice un transformador de entrada con estos datos del cuerpo, específicos del origen. Por ejemplo, <$.body> para Amazon

SQS o <$.data> para Kinesis. Si la carga original es una cadena JSON válida (por ejemplo, `{"key": "value"}`), el uso del transformador de entrada con datos del cuerpo específicos del origen hará que se eliminen las comillas de la carga de origen original. Por ejemplo, `{"key": "value"}` se convertirá en `"{key: value}"` cuando se entregue al destino. Si el destino requiere cargas JSON válidas (por ejemplo, EventBridge Lambda o Step Functions), se producirá un error en la entrega. Para que el destino reciba los datos de origen originales sin generar un valor JSON no válido, empaquete el transformador de entrada de los datos del cuerpo del origen en formato JSON. Por ejemplo, `{"data": <$.data>}`.

El análisis del cuerpo implícito también se puede utilizar para rellenar dinámicamente los valores de la mayoría de los parámetros de enriquecimiento o de destino de la canalización. Para obtener más información, consulte ???

> **Note**
>
> Si la carga original es un valor JSON válido, este campo contendrá el valor JSON sin ocultar ni codificar en base64. Sin embargo, si la carga no es un valor JSON válido, EventBridge codifica en base64 los campos que se indican a continuación, con la excepción de Amazon SQS.

- Active MQ — `data`
- Kinesis – `data`
- Amazon MSK, y `key` `value`
- Rabbit MQ — `data`
- Apache Kafka autoadministrado; – `key` y `value`
- Amazon SQS – `body`

Problemas comunes con la transformación de entradas

Estos son algunos problemas comunes cuando se transforma la entrada en EventBridge:

- Para cadenas, se requieren comillas.
- No hay validación al crear la ruta JSON para la plantilla.
- Si especifica una variable que coincida con una ruta JSON que no existe en el evento, dicha variable no se crea ni aparece en la salida.

- Las propiedades JSON como `aws.pipes.event.json` solo se pueden usar como el valor de un campo JSON, no en línea en otras cadenas.
- EventBridge no oculta valores extraídos por la ruta de entrada al rellenar la plantilla de entrada para un destino.
- Si una ruta JSON hace referencia a un objeto o matriz JSON, pero se hace referencia a la variable en una cadena, EventBridge elimina las comillas internas para garantizar la validez de la cadena. Por ejemplo, en "Cuerpo es <$.body>", EventBridge eliminaría las comillas del objeto.

  Por lo tanto, si quiere generar un objeto JSON basado en una única variable de ruta JSON, debe colocarlo como clave. En este ejemplo, `{"body": <$.body>}`.
- No se requieren comillas para las variables que representan cadenas. Están permitidas, pero EventBridge Pipes añade automáticamente comillas a los valores de las variables de cadena durante la transformación, para garantizar que el resultado de la transformación tenga un formato JSON válido. EventBridge Pipes no añade comillas a las variables que representan objetos o matrices JSON. No añada comillas a las variables que representan objetos o matrices JSON.

  Por ejemplo, la siguiente plantilla de entrada incluye variables que representan tanto cadenas como objetos JSON:

  ```
  {
    "pipeArn" : <aws.pipes.pipe-arn>,
    "pipeName" : <aws.pipes.pipe-name>,
    "originalEvent" : <aws.pipes.event.json>
  }
  ```

  El resultado es un formato JSON válido con las comillas adecuadas:

  ```
  {
    "pipeArn" : "arn:aws:events:us-east-2:123456789012:pipe/example",
    "pipeName" : "example",
    "originalEvent" : {
      ... // commented for brevity
    }
  }
  ```
- Para los enriquecimientos o destinos de Lambda o Step Functions, los lotes se entregan al destino como matrices JSON, incluso si el tamaño del lote es 1. Sin embargo, los transformadores de entrada se seguirán aplicando a los registros individuales de la matriz JSON, no a la matriz en su conjunto. Para obtener más información, consulte ???.

Registrar Amazon EventBridge Pipes

EventBridge El registro de tuberías le permite hacer que EventBridge Pipes envíe registros que detallen el rendimiento de las tuberías a AWS los servicios compatibles. Utilice los registros para obtener información sobre el rendimiento de ejecución de su canalización y para ayudar a solucionar problemas y depurar.

Puede seleccionar los siguientes AWS servicios como destinos de registro a los que EventBridge Pipes entrega los registros:

- CloudWatch Registros

  EventBridge entrega los registros al grupo de CloudWatch registros especificado.

  Utilice CloudWatch los registros para centralizar los registros de todos los sistemas, aplicaciones y AWS servicios que utilice en un único servicio altamente escalable. Para obtener más información, consulte Trabajar con grupos de registros y transmisiones de CloudWatch registros en la Guía del usuario de Amazon Logs.
- Registros de transmisión Firehose

  EventBridge entrega los registros a un flujo de entrega de Firehose.

  Amazon Data Firehose es un servicio totalmente gestionado para entregar datos de streaming en tiempo real a destinos como determinados AWS servicios, así como a cualquier punto de enlace HTTP personalizado o punto de enlace HTTP propiedad de proveedores de servicios externos compatibles. Para obtener más información, consulte Creación de un flujo de entrega de Amazon Data Firehose en la Guía del usuario de Amazon Data Firehose.
- Registros de Amazon S3

  EventBridge entrega los registros como objetos de Amazon S3 al bucket especificado.

  Amazon S3 es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes del sector. Para obtener más información, consulte Cargar, descargar y trabajar con objetos en Amazon S3 en la Guía del usuario de Amazon Simple Storage Service.

Cómo funciona el registro EventBridge de Amazon Pipes

Cada evento o lote de eventos recibido por una canalización que se dirige a un enriquecimiento o un destino se considera una ejecución de canalización. Si está activado, EventBridge genera un registro de registro para cada paso de ejecución que realiza a medida que se procesa el lote de eventos. La información contenida en el registro se aplica al lote de eventos, ya sea un evento único o hasta 10.000 eventos.

Puede configurar el tamaño del lote de eventos en la canalización de origen y destino. Para obtener más información, consulte ???.

Los datos de registro enviados a cada destino de registro son los mismos.

Si se configura un destino de Amazon CloudWatch Logs, los registros de registro entregados a todos los destinos tienen un límite de 256 KB. Los campos se truncarán según sea necesario.

Puede personalizar los registros que se EventBridge envían a los destinos de registro seleccionados de la siguiente manera:

- Puede especificar el nivel de registro, que determina los pasos de ejecución para los que se EventBridge envían los registros a los destinos de registro seleccionados. Para obtener más información, consulte ???.
- Puede especificar si EventBridge Pipes incluye los datos de ejecución en los registros de los pasos de ejecución cuando sea relevante. Estos datos incluyen:
  - La carga del lote de eventos
  - La solicitud enviada al servicio de AWS enriquecimiento o de destino
  - La respuesta devuelta por el servicio de AWS enriquecimiento o de destino

  Para obtener más información, consulte ???.

Especificar el nivel EventBridge de registro de Pipes

Puede especificar los tipos de pasos de ejecución para los que se EventBridge envían los registros a los destinos de registro seleccionados.

Elija entre los siguientes niveles de detalles para incluirlos en los entradas de registros. El nivel de registro se aplica a todos los destinos de registro especificados para la canalización. Cada nivel de registro incluye los pasos de ejecución de los niveles de registro anteriores.

- DESACTIVADO: EventBridge no envía ningún registro a ningún destino de registro especificado. Este es el valor predeterminado.
- ERROR: EventBridge envía todos los registros relacionados con los errores generados durante la ejecución de la canalización a los destinos de registro especificados.
- INFORMACIÓN: EventBridge envía todos los registros relacionados con los errores y selecciona otros pasos realizados durante la ejecución de la canalización a los destinos de registro especificados.
- TRACE: EventBridge envía todos los registros generados durante cualquier paso de la ejecución de la canalización a los destinos de registro especificados.

En la EventBridge consola, CloudWatch los registros se seleccionan como destino de registro de forma predeterminada, al igual que el nivel de `ERROR` registro. Por lo tanto, de forma predeterminada, EventBridge Pipes crea un nuevo grupo de CloudWatch registros al que envía los registros que contienen el `ERROR` nivel de detalle. Al configurar los registros mediante programación, no se selecciona ningún valor predeterminado.

En la siguiente tabla se enumeran los pasos de ejecución incluidos en cada nivel de registro.

Paso	TRACE	INFO	ERROR	OFF
Ejecución errónea	x	x	x	
Ejecución errónea parcialmente	x	x	x	
Ejecución iniciada	x	x		
Ejecución correcta	x	x		
Ejecución limitada	x	x	x	
Tiempo de espera de la ejecución	x	x	x	
Invocación de enriquecimiento errónea	x	x	x	

Paso	TRACE	INFO	ERROR	OFF
Invocación de enriquecimiento omitida	x	x		
Invocación de enriquecimiento iniciada	x			
Invocación de enriquecimiento correcta	x			
Fase de enriquecimiento iniciada	x	x		
Fase de enriquecimiento errónea	x	x	x	
Fase de enriquecimiento correcta	x	x		
Transformación de enriqueci miento errónea	x	x	x	
Transformación del enriqueci miento iniciada	x			
Transformación del enriqueci miento correcta	x			
Invocación de destino errónea	x	x	x	
Invocación de destino parcialmente errónea	x	x	x	
Invocación de destino omitida	x			
Invocación de destino iniciada	x			
Invocación de destino correcta	x			

Paso	TRACE	INFO	ERROR	OFF
Fase de destino iniciada	x	x		
Fase de destino errónea	x	x	x	
Fase de destino parcialmente errónea	x	x	x	
Fase de destino omitida	x			
Fase de destino correcta	x	x		
Transformación de destino errónea	x	x	x	
Transformación de destino iniciada	x			
Transformación de destino correcta	x			

Incluir datos de ejecución en los registros EventBridge de Pipes

Puede especificar que se incluyan datos de ejecución en los registros que genera. EventBridge Los datos de ejecución incluyen campos que representan la carga del lote de eventos, así como la solicitud enviada y la respuesta del enriquecimiento y el destino.

Los datos de ejecución son útiles para solucionar problemas y depurar errores. En el campo `payload` figura el contenido real de cada evento incluido en el lote, lo que permite correlacionar eventos individuales con una ejecución de canalización específica.

Si opta por incluir los datos de ejecución, estos se incluyen para todos los destinos de registro especificados para la canalización.

> **Important**
>
> Estos campos pueden contener información confidencial. EventBridge no intenta censurar el contenido de estos campos durante el registro.

Al incluir los datos de ejecución, EventBridge agrega los siguientes campos a los registros pertinentes:

- **`payload`**

  Representa el contenido del lote de eventos que la canalización está procesando.

  EventBridge incluye el `payload` campo en los registros generados en los pasos en los que es posible que se haya actualizado el contenido del lote de eventos. Esto incluye los siguientes pasos:

  - `EXECUTION_STARTED`
  - `ENRICHMENT_TRANSFORMATION_SUCCEEDED`
  - `ENRICHMENT_STAGE_SUCCEEDED`
  - `TARGET_TRANSFORMATION_SUCCEEDED`
  - `TARGET_STAGE_SUCCEEDED`

- **`awsRequest`**

  Representa la solicitud enviada al enriquecimiento o al destino como una cadena JSON. En el caso de las solicitudes enviadas a un destino de API, esto representa la solicitud HTTP enviada a ese punto de conexión.

  EventBridge incluye el `awsRequest` campo en los registros generados en las etapas finales del enriquecimiento y la segmentación, es decir, después de haber EventBridge ejecutado o intentado ejecutar la solicitud en relación con el servicio de enriquecimiento o destino especificado. Esto incluye los siguientes pasos:

  - `ENRICHMENT_INVOCATION_FAILED`
  - `ENRICHMENT_INVOCATION_SUCCEEDED`
  - `TARGET_INVOCATION_FAILED`
  - `TARGET_INVOCATION_PARTIALLY_FAILED`
  - `TARGET_INVOCATION_SUCCEEDED`

- **`awsResponse`**

  Representa la respuesta devuelta por el enriquecimiento o el destino, en formato JSON. En el caso de las solicitudes enviadas a un destino de API, esto representa la respuesta HTTP devuelta por ese punto de conexión.

Del mismo modo`awsRequest`, EventBridge incluye el `awsResponse` campo en los registros generados en las etapas finales del enriquecimiento y la segmentación, es decir, después de haber EventBridge ejecutado o intentado ejecutar una solicitud contra el servicio de enriquecimiento o objetivo especificado y haber recibido una respuesta. Esto incluye los siguientes pasos:

- `ENRICHMENT_INVOCATION_FAILED`
- `ENRICHMENT_INVOCATION_SUCCEEDED`
- `TARGET_INVOCATION_FAILED`
- `TARGET_INVOCATION_PARTIALLY_FAILED`
- `TARGET_INVOCATION_SUCCEEDED`

Para obtener más información sobre los pasos de ejecución de canalizaciones, consulte. ???

Truncar los datos de ejecución en los registros de Pipes EventBridge

Si opta por EventBridge incluir los datos de ejecución en los registros de una tubería, existe la posibilidad de que un registro supere el límite de tamaño de 256 KB. Para evitarlo, trunca EventBridge automáticamente los campos de datos de ejecución, en el siguiente orden. EventBridge trunca cada campo por completo antes de avanzar para truncar el siguiente campo. EventBridge trunca los datos del campo simplemente eliminando los caracteres del final de la cadena de datos; no se intenta truncar en función de la importancia de los datos, y el truncamiento invalidará el formato JSON.

- `payload`
- `awsRequest`
- `awsResponse`

Si trunca los campos en ese caso, el `truncatedFields` campo EventBridge incluye una lista de los campos de datos truncados.

Informes de errores en los registros de Pipes EventBridge

EventBridge también incluye datos de error, cuando están disponibles, en los pasos de ejecución de Pipe que representan los estados de fallo. Estos pasos incluyen:

- `ExecutionThrottled`

- `ExecutionTimeout`
- `ExecutionFailed`
- `ExecutionPartiallyFailed`
- `EnrichmentTransformationFailed`
- `EnrichmentInvocationFailed`
- `EnrichmentStageFailed`
- `TargetTransformationFailed`
- `TargetInvocationFailed`
- `TargetInvocationPartiallyFailed`
- `TargetStageFailed`
- `TargetStagePartiallyFailed`

EventBridge Canaliza los pasos de ejecución

Comprender el flujo de los pasos de ejecución de canalizaciones puede ayudarle a solucionar problemas o depurar el rendimiento de la canalización mediante registros.

Cada evento o lote de eventos recibido por una canalización que se dirige a un enriquecimiento o un destino se considera una ejecución de canalización. Si está activado, EventBridge genera un registro para cada paso de ejecución que realiza a medida que se procesa el lote de eventos.

A grandes rasgos, la ejecución consta de dos fases, o conjuntos de pasos: enriquecimiento y destino. Cada una de estas fases consta de los pasos de transformación e invocación.

Los pasos principales de una ejecución de canalización correcta siguen este flujo:

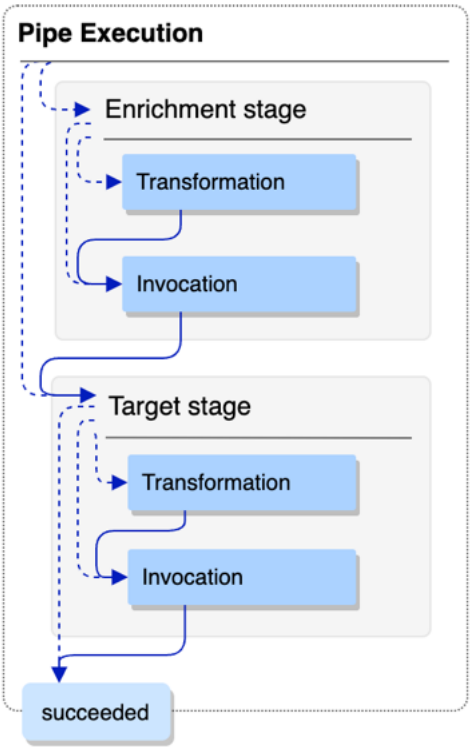
- Se inicia la ejecución de la canalización.
- La ejecución entra en la fase de enriquecimiento si se ha especificado un enriquecimiento para los eventos. Si no ha especificado un enriquecimiento, la ejecución pasa a la fase de destino.

  En la fase de enriquecimiento, la canalización realiza cualquier transformación que haya especificado y, a continuación, invoca el enriquecimiento.
- En la fase de enriquecimiento, la canalización realiza cualquier transformación que haya especificado y, a continuación, invoca el enriquecimiento.

  Si no ha especificado ninguna transformación ni destino, la ejecución omite la fase de destino.

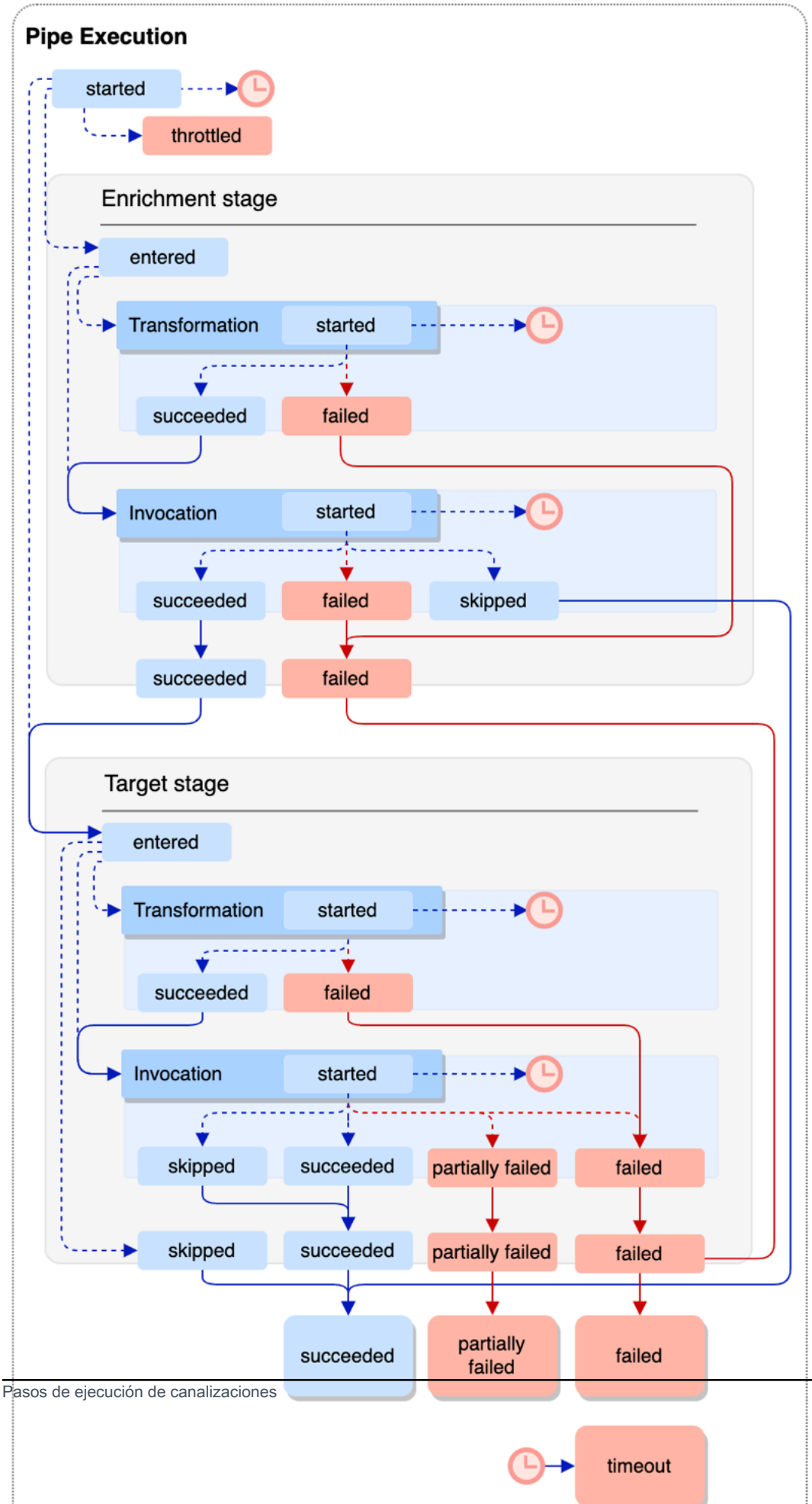
- La ejecución de la canalización se completa correctamente.

En el siguiente diagrama se muestra este flujo. Las rutas divergentes adoptan la forma de líneas punteadas.



El siguiente diagrama presenta una vista detallada del flujo de ejecución de la canalización, con todos los pasos de ejecución posibles representados. De nuevo, las rutas divergentes adoptan la forma de líneas punteadas

Para obtener una lista completa de pasos de ejecución de la canalización, consulte ???.


Pipe Execution
started
throttled
Enrichment stage
entered
Transformation
started
succeeded
failed
Invocation
started
succeeded
failed
skipped
succeeded
failed
Target stage
entered
Transformation
started
succeeded
failed
Invocation
started
skipped
succeeded
partially failed
failed
skipped
succeeded
partially failed
failed
succeeded
partially failed
failed
timeout

Tenga en cuenta que la invocación de destino puede provocar un fallo parcial del lote. Para obtener más información, consulte ???.

EventBridge Referencia del esquema de registro de tuberías

La siguiente referencia detalla el esquema de los registros de EventBridge Pipes.

Cada entrada de registros representa un paso de ejecución de la canalización y puede contener hasta 10.000 eventos si el origen y el destino de la canalización se han configurado para el procesamiento por lotes.

Para obtener más información, consulte ???.

```
{
    "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

El ID de la ejecución de la canalización.

Cada evento o lote de eventos recibido por una canalización que se dirige a un enriquecimiento o un destino se considera una ejecución de canalización. Para obtener más información, consulte ???.

Marca de tiempo

La fecha y la hora en que se emitió el evento de registro.

Unidad: milisegundos

messageType

El paso de ejecución de la canalización para el que se generó el registro.

Para obtener más información acerca de los pasos de ejecución, consulte ???.

resourceArn

El Nombre de recurso de Amazon (ARN) para la canalización.

logLevel

El nivel de detalle especificado para el registro de la canalización.

Valores válidos: `ERROR` | `INFO` | `TRACE`

Para obtener más información, consulte ???.

payload

El contenido del lote de eventos que la canalización está procesando.

EventBridge incluye este campo solo si ha especificado incluir datos de ejecución en los registros de esta canalización. Para obtener más información, consulte ???.

> **Important**
>
> Estos campos pueden contener información confidencial. EventBridge no intenta censurar el contenido de estos campos durante el registro.

Para obtener más información, consulte ???.

awsRequest

La solicitud enviada al enriquecimiento o destino, en formato JSON. En el caso de las solicitudes enviadas a un destino de API, esto representa la solicitud HTTP enviada a ese punto de conexión.

EventBridge incluye este campo solo si ha especificado incluir datos de ejecución en los registros de esta canalización. Para obtener más información, consulte ???.

> **Important**
>
> Estos campos pueden contener información confidencial. EventBridge no intenta censurar el contenido de estos campos durante el registro.

Para obtener más información, consulte ???.

awsResponse

La respuesta devuelta por el enriquecimiento o el destino, en formato JSON. En el caso de las solicitudes enviadas a un destino de API, esto representa la respuesta HTTP devuelta por ese punto de conexión, y no la respuesta devuelta por el servicio de destino de API propiamente dicho.

EventBridge incluye este campo solo si ha especificado incluir datos de ejecución en los registros de esta canalización. Para obtener más información, consulte ???.

> **Important**
>
> Estos campos pueden contener información confidencial. EventBridge no intenta censurar el contenido de estos campos durante el registro.

Para obtener más información, consulte ???.

truncatedFields

Se EventBridge ha truncado una lista de todos los campos de datos de ejecución para mantener el registro por debajo del límite de tamaño de 256 KB.

Si EventBridge no ha tenido que truncar ninguno de los campos de datos de ejecución, este campo está presente pero. `null`

Para obtener más información, consulte ???.

error

Contiene información sobre errores generados durante este paso de ejecución de la canalización.

Si no se generó ningún error durante este paso de ejecución de la canalización, este campo está presente pero es `null`.

httpStatusCode

El código de estado HTTP devuelto por el servicio al que se llama.

message

El mensaje de error que devuelve el servicio al que se llama.

details

Cualquier información de error detallada devuelta por el servicio al que se llama.

awsService

El nombre del servicio al que se llama.

requestId

El ID de solicitud para esta solicitud del servicio al que se llama.

Registro y supervisión de Amazon EventBridge Pipes mediante AWS CloudTrail Amazon CloudWatch Logs

Puede registrar EventBridge las invocaciones de Pipes y usar CloudTrail y monitorear el estado de sus tuberías mediante CloudWatch métricas.

CloudWatch métricas

EventBridge Pipes envía métricas a Amazon CloudWatch cada minuto para todo, desde la ejecución de una tubería que se limita hasta la invocación exitosa de un objetivo.

Métrica	Descripción	Dimensiones	Unidades
`Concurrency`	El número de ejecuciones simultáneas de una canalización.	AwsAccountId	Ninguna
`Duration`	Duración de la ejecución de la canalización.	PipeName	Milisegundos
`EventCount`	El número de eventos que ha procesado una canalización.	PipeName	Ninguna
`EventSize`	El tamaño de la carga del evento que invocó la canalización.	PipeName	Bytes

Métrica	Descripción	Dimensiones	Unidades
`ExecutionThrottled`	Número de ejecuciones de una tubería que se limitaron. **Note** Este valor será `0` si no se limitó ninguna ejecución.	AwsAccountId, PipeName	Ninguna
`ExecutionTimeout`	Cuántas veces se agotó el tiempo de espera de una ejecución de canalización antes de completar la ejecución. **Note** Este valor será `0` si no se agotó el tiempo de espera de ninguna ejecución .	PipeName	Ninguna
`ExecutionFailed`	Cuántas ejecuciones de una canalización fueron erróneas. **Note** Este valor será `0` si no hubo ninguna ejecución errónea.	PipeName	Ninguna
`ExecutionPartiallyFailed`	Cuántas ejecuciones de una canalización fueron parcialmente erróneas. **Note** Este valor será `0` si no hubo ninguna ejecución parcialmente errónea.	PipeName	Ninguna

Métrica	Descripción	Dimensiones	Unidades
`EnrichmentStageDuration`	Cuánto tiempo tardó en completarse la fase de enriquecimiento.	PipeName	Milisegundos
`EnrichmentStageFailed`	Cuántas ejecuciones de la fase de enriquecimiento de una canalización fueron erróneas. **Note** Este valor será `0` si no hubo ninguna ejecución errónea.	PipeName	Ninguna
`Invocations`	Número total de invocaciones.	AwsAccountId, PipeName	Ninguna
`TargetStageDuration`	Cuánto tiempo tardó en completarse la fase de destino.	PipeName	Milisegundos
`TargetStageFailed`	Cuántas ejecuciones de la fase de destino de una canalización fueron erróneas. **Note** Este valor será `0` si no hubo ninguna ejecución errónea.	PipeName	Ninguna

Métrica	Descripción	Dimensiones	Unidades
`TargetStagePartiallyFailed`	Cuántas ejecuciones de la fase de destino de una canalización fueron parcialmente erróneas. **Note** Este valor será `0` si no hubo ninguna ejecución de la fase de destino parcialmente errónea.	PipeName	Ninguna
`TargetStageSkipped`	Cuántas ejecuciones de la fase de destino de una canalización se omitieron (por ejemplo, debido a que el enriquecimiento devolvió una carga vacía).	PipeName	Recuento

Dimensiones de las métricas CloudWatch

CloudWatch Las métricas tienen dimensiones o atributos que se pueden ordenar, que se muestran a continuación.

Dimensión	Descripción
`AwsAccountId`	Filtra las métricas disponibles por ID de cuenta.
`PipeName`	Filtra las métricas disponibles por nombre de canalización.

Solución de problemas y gestión de errores de Amazon EventBridge Pipes

Comportamiento de los reintentos y gestión de errores

EventBridge Pipes vuelve a intentar automáticamente el enriquecimiento y la invocación de destino en caso de que se produzca un error reintentable AWS en el servicio de origen, el enriquecimiento o los servicios de destino, o. EventBridge Sin embargo, si las implementaciones de enriquecimiento

o destino de los clientes devuelven errores, el rendimiento de sondeo de las canalizaciones irá disminuyendo gradualmente. En caso de que se produzcan errores 4xx casi continuos de hasta cuatro veces (como problemas de autorización con IAM o ausencia de recursos), la canalización se puede desactivar automáticamente con un mensaje explicativo en el valor `StateReason`.

Errores de invocación de canalizaciones y comportamiento de los reintentos

Al invocar una canalización, se pueden producir dos tipos principales de errores: errores de canalización internos y errores de invocación del cliente.

Errores de canalización internos

Los errores internos de Pipe son errores que se producen por aspectos de la invocación gestionados por el servicio Pipes. EventBridge

Estos tipos de errores pueden incluir problemas como:

- Un error de conexión HTTP al intentar invocar el servicio de destino del cliente
- Una caída transitoria de la disponibilidad en el propio servicio de canalización.

En general, EventBridge Pipes reintenta los errores internos un número indefinido de veces y solo se detiene cuando el registro caduca en la fuente.

En el caso de las tuberías con una fuente de flujo, EventBridge Pipes no cuenta los reintentos por errores internos con respecto al número máximo de reintentos especificado en la política de reintentos de la fuente de transmisión. En el caso de las canalizaciones con una fuente de Amazon SQS, EventBridge Pipes no cuenta los reintentos de errores internos en relación con el recuento máximo de recepción de la fuente de Amazon SQS.

Errores de invocación del cliente

Los errores de invocación del cliente son errores que se derivan de la configuración o del código administrado por el usuario.

Estos tipos de errores pueden incluir problemas como:

- Los permisos necesarios para invocar el destino son insuficientes.
- Un error lógico en un punto de conexión de Lambda, Step Functions, un destino de API o API Gateway de un cliente invocado de forma sincrónica.

En el caso de los errores de invocación por parte de los clientes, EventBridge Pipes hace lo siguiente:

- En el caso de las canalizaciones con una fuente de flujo, EventBridge Pipes vuelve a intentarlo hasta los tiempos máximos de reintentos configurados en la política de reintentos de canalización o hasta que venza la antigüedad máxima del registro, lo que ocurra primero.
- En el caso de las canalizaciones con una fuente de Amazon SQS, EventBridge Pipes vuelve a intentar un error del cliente hasta el recuento máximo de recepciones de la cola de fuentes.
- En el caso de las canalizaciones con una fuente de Apache Kafka o Amazon MQ EventBridge , vuelve a intentar los errores del cliente de la misma manera que reintenta los errores internos.

En el caso de las tuberías con objetivos de cálculo, debe invocar la canalización de forma sincrónica para que EventBridge Pipes detecte cualquier error de tiempo de ejecución derivado de la lógica de cálculo del cliente y pueda volver a intentarlo con esos errores. Las canalizaciones no pueden volver a intentar subsanar errores que aparecen en la lógica de un flujo de trabajo estándar de Step Functions, ya que este destino debe invocarse de forma asíncrona.

Para Amazon SQS y fuentes de streaming, como Kinesis y DynamoDB, Pipes admite la gestión parcial de errores por lotes de los EventBridge errores de destino. Para obtener más información, consulte Fallo parcial de lotes.

Comportamiento de la DLQ de una canalización

Una canalización hereda el comportamiento de la cola de mensajes fallidos (DLQ) del origen:

- Si la cola de Amazon SQS de origen tiene una DLQ configurada, los mensajes se entregan automáticamente allí tras el número de intentos especificado.
- Para los orígenes de flujo, como los flujos de DynamoDB y Kinesis, puede configurar una DLQ para los eventos de canalización y ruta. Los orígenes de flujo de DynamoDB y Kinesis admiten colas de Amazon SQS y temas de Amazon SNS como destinos de DLQ.

Si especifica una `DeadLetterConfig` para una canalización con un origen de Kinesis o DynamoDB, asegúrese de que la propiedad `MaximumRecordAgeInSeconds` de la canalización sea inferior a la propiedad `MaximumRecordAge` del evento de origen. `MaximumRecordAgeInSeconds`controla el momento en que el sondeador de la canalización cancelará el evento y lo enviará a la DLQ y `MaximumRecordAge` controla cuánto tiempo estará visible el mensaje en el flujo del origen antes de que se elimine. Por lo tanto, establezca la propiedad

`MaximumRecordAgeInSeconds` en un valor inferior al de la propiedad `MaximumRecordAge` del origen para que haya suficiente tiempo entre el momento en que el evento se envía a la DLQ y el momento en que el origen lo elimina automáticamente para determinar el motivo por el que el evento pasó a la DLQ.

Para orígenes de Amazon MQ, la DLQ se puede configurar directamente en el agente de mensajes.

EventBridge Pipes no admite los DLQ de tipo «primero en entrar, primero en salir» (FIFO) para las fuentes de transmisión.

EventBridge Pipes no admite DLQ para las fuentes de transmisión de Amazon MSK y Apache Kafka autogestionadas.

Estados de fallo de canalizaciones

La creación, la eliminación y la actualización de canalizaciones son operaciones asíncronas que pueden provocar un estado de fallo. Del mismo modo, una canalización puede desactivarse automáticamente debido a fallos. En todos los casos, el valor `StateReason` de la canalización proporciona información que ayuda a solucionar el fallo.

A continuación se muestra una lista de valores `StateReason` posibles:

- Flujo no encontrado. Para reanudar el procesamiento, elimine la canalización y cree una nueva.
- Pipes no tiene los permisos necesarios para realizar operaciones de cola (sqs:ReceiveMessage, sqs: y sqs:) DeleteMessage GetQueueAttributes
- Error de conexión. Su VPC debe poder conectarse a canalizaciones. Puede proporcionar acceso configurando una puerta de enlace NAT o un punto final de VPC a los datos de canalización. Para saber cómo configurar la puerta de enlace NAT o el punto final de VPC para canalizar datos, consulte la documentación. AWS
- El clúster MSK no tiene grupos de seguridad asociados

Una canalización se puede detener automáticamente con un valor `StateReason` actualizado. Algunas de las causas posibles son:

- Un flujo de trabajo estándar de Step Functions configurado como un enriquecimiento.
- Un flujo de trabajo estándar de Step Functions configurado como un destino que debe invocarse de forma sincrónica.

Fallos de cifrado personalizado

Si configuras una fuente para que utilice una clave de cifrado AWS KMS personalizada (CMK), en lugar de una AWS KMS clave AWS gestionada, debes conceder de forma explícita el permiso de descifrado de la función de ejecución de tu canal. Para ello, incluya el siguiente permiso adicional en la política de CMK personalizada:

```
  {
      "Sid": "Allow Pipes access",
      "Effect": "Allow",
      "Principal": {
          "AWS": "arn:aws:iam::01234567890:role/service-role/
Amazon_EventBridge_Pipe_DDBStreamSourcePipe_12345678"
      },
      "Action": "kms:Decrypt",
      "Resource": "*"
  }
```

Sustituya el rol anterior por el rol de ejecución de su canalización.

Esto es válido para todas las fuentes canalizadas con AWS KMS CMK, incluidas las siguientes:

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

Tutorial: Crear una canalización de EventBridge que filtre eventos de origen

En este tutorial, creará una canalización que conecte un origen de flujo de DynamoDB a un destino de cola de Amazon SQS. Tendrá que especificar un patrón de eventos para la canalización que se utilizará al filtrar los eventos para entregarlos a la cola. A continuación, probará la canalización para asegurarse de que solo se entreguen los eventos deseados.

Requisitos previos: crear el origen y el destino

Antes de crear la canalización, tendrá que crear el origen y el destino a los que se va a conectar. En este caso, un flujo de datos de Amazon DynamoDB como origen de la canalización y una cola de Amazon SQS como destino de la canalización.

Para simplificar este paso, puede utilizar AWS CloudFormation para aprovisionar los recursos de origen y destino. Para ello, creará una plantilla de CloudFormation que definirá los siguientes recursos:

- El origen de la canalización

  Una tabla de Amazon DynamoDB, denominada `pipe-tutorial-source`, con un flujo habilitado para proporcionar un flujo ordenado de información sobre los cambios que se realizan en los elementos de la tabla de DynamoDB.

- El tipo de destino

  Una cola de Amazon SQS, denominada `pipe-tutorial-target`, para recibir el flujo de eventos de DynamoDB desde su canalización.

Para crear la plantilla de CloudFormation para el aprovisionamiento de recursos de canalización

1. Copie el texto de la plantilla JSON de la sección ???, a continuación.
2. Guarde la plantilla como un archivo JSON (por ejemplo, `~/pipe-tutorial-resources.json`).

A continuación, utilice el archivo de plantilla que acaba de crear para aprovisionar una pila de CloudFormation.

> **Note**
>
> Una vez que cree su pila de CloudFormation, se le cobrarán los recursos de AWS que esta aprovisione.

Aprovisionar los requisitos previos del tutorial mediante la CLI de AWS

- Ejecute el siguiente comando de la CLI, donde `--template-body` especifica la ubicación de su archivo de plantilla:

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

Aprovisionar los requisitos previos del tutorial mediante la consola de CloudFormation

1. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
2. Seleccione Pilas y, seguidamente, seleccione Crear pila, y elija la opción con recursos nuevos (estándar).

   CloudFormation muestra el asistente Crear pila.
3. En Requisito previo: preparar la plantilla, deje seleccionada la opción predeterminada La plantilla está lista.
4. En Especificar plantilla, seleccione Cargar un archivo de plantilla y, a continuación, seleccione el archivo y elija Siguiente.
5. Configure la pila y los recursos que aprovisionará:
   - En Nombre de pila, escriba `pipe-tuturial-resources`.
   - En Parámetros, deje los nombres predeterminados para la tabla de DynamoDB y la cola de Amazon SQS.
   - Seleccione Siguiente.
6. Seleccione Siguiente y, a continuación, seleccione Enviar.

   CloudFormation crea la pila y aprovisiona los recursos que se definen en la plantilla.

Para obtener más información acerca de CloudFormation, consulte ¿Qué es AWS CloudFormation? en la Guía del usuario de AWS CloudFormation.

Paso 1: Crear la canalización

Con el origen y el destino de la canalización aprovisionados, ahora puede crear la canalización para conectar los dos servicios.

Crear la canalización mediante la consola de EventBridge

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Canalizaciones.
3. Seleccione Crear canalización.
4. En Nombre, asigne un nombre a su canalización `pipe-tutorial`.
5. Especifique el origen de flujo de datos de DynamoDB:

   a. En Detalles, en Origen, seleccione Flujo de datos de DynamoDB.

      EventBridge muestra los ajustes de configuración de origen específicos de DynamoDB.

   b. En Flujo de DynamoDB, seleccione. `pipe-tutorial-source`

      Deje Posición inicial establecida en el valor predeterminado, `Latest`.

   c. Seleccione Siguiente.
6. Especifique y pruebe un patrón de eventos para filtrar eventos:

   El filtrado le permite controlar qué eventos envían las canalizaciones al enriquecimiento y al destino. La canalización solo envía al enriquecimiento o al destino los eventos que coinciden con el patrón de eventos.

   Para obtener más información, consulte ???.

   > **Note**
   >
   > Solo se le facturarán los eventos que se envíen al enriquecimiento o al destino.

   a. En Evento de muestra (opcional), deje Eventos de AWS seleccionados y asegúrese de que el Evento 1 de muestra del flujo de DynamoDB esté seleccionado.

      Este es el evento de muestra que utilizará para probar nuestro patrón de eventos.

   b. En Patrón de eventos, introduzca el siguiente patrón de eventos:

```
{
  "eventName": ["INSERT", "MODIFY"]
}
```

c. Seleccione Patrón de prueba.

   EventBridge muestra un mensaje que indica que el evento de muestra coincide con el patrón de eventos. Se debe a que el evento de muestra tiene un valor `eventName` de `INSERT`.

d. Seleccione Siguiente.

7. Seleccione Siguiente para omitir la especificación de un enriquecimiento.

   En este ejemplo, no seleccionará un enriquecimiento. Los enriquecimientos le permiten seleccionar un servicio para mejorar los datos del origen antes de enviarlos al destino. Para obtener más información, consulte ???.

8. Especifique su cola de Amazon SQS como destino de la canalización:

   a. En Detalles, en Servicio de destino, seleccione Cola de Amazon SQS.

   b. En Cola, seleccione `pipe-tutorial-target`.

   c. Deje vacía la sección Transformador de entrada de destino.

      Para obtener más información, consulte ???.

9. Seleccione Crear canalización

   EventBridge crea la canalización y muestra la página de detalles de la canalización. La canalización estará lista cuando su estado se actualice a`Running`.

Paso 2: Confirmar los eventos de los filtros de la canalización

La canalización está configurada, pero aún no ha recibido los eventos de la tabla.

Para probar la canalización, debe actualizar las entradas de la tabla de DynamoDB. Cada actualización generará eventos que el flujo de DynamoDB envía a nuestra canalización. Algunos coincidirán con el patrón de eventos que especificó, otros no. A continuación, puede examinar la cola de Amazon SQS para asegurarse de que la canalización solo entregó los eventos que coincidían con nuestro patrón de eventos.

Actualizar los elementos de la tabla para generar eventos

1. Abra la consola de DynamoDB en https://console.aws.amazon.com/dynamodb/.

2. En la barra de navegación izquierda, seleccione Tablas. Seleccione la tabla de `pipe-tutorial-source`.

   DynamoDB muestra la página de detalles de la tabla para `pipe-tutorial-source`.

3. Seleccione Explorar los elementos de la tabla y, a continuación, selecciones Crear elemento.

   DynamoDB muestra la página Crear elemento.

4. En Atributos, cree un elemento de tabla nuevo:

   a. En Álbum, escriba `Album A`.

   b. En Artista, escriba `Artist A`.

   c. Seleccione Crear elemento.

5. Actualice el elemento de la tabla:

   a. En Elementos devueltos, seleccione Álbum A.

   b. Seleccione Añadir atributo nuevo y, a continuación, seleccione Cadena.

   c. Introduzca un valor de `Song` nuevo, con un valor de `Song A`.

   d. Seleccione Guardar cambios.

6. Elimine el elemento de la tabla:

   a. En Elementos devueltos, marque Álbum A.

   b. En el menú Acciones, seleccione Detectar elementos.

Ha realizado tres actualizaciones del elemento de la tabla; esto genera tres eventos para el flujo de datos de DynamoDB:

- Un evento `INSERT` al crear el elemento.
- Un evento `MODIFY` al agregar un atributo al elemento.
- Un evento `REMOVE` al eliminar el elemento.

Sin embargo, el patrón de eventos que especificó para la canalización debe filtrar los eventos que no lo sean eventos `INSERT` o `MODIFY`. A continuación, confirme que la canalización envió los eventos esperados a la cola.

Confirmar que los eventos esperados se enviaron a la cola

1. Abra la consola de Amazon SQS en https://console.aws.amazon.com/sqs/.
2. Seleccione la cola de `pipe-tutorial-target`.

   Amazon SQS muestra la página de detalles de la cola.
3. Seleccione Enviar y recibir mensajes y, a continuación, en Recibir mensajes, seleccione Sondear mensajes.

   La cola sondea la canalización y, a continuación, muestra los eventos que recibe.
4. Elija el nombre del evento para ver el JSON del evento que se entregó.

Debe haber dos eventos en la cola: uno con un `eventName` de `INSERT` y otro con un `eventName` de `MODIFY`. Sin embargo, la canalización no proporcionó el evento para eliminar el elemento de la tabla, ya que ese evento tenía un `eventName` de `REMOVE`, que no coincidía con el patrón de eventos que especificó en la canalización.

Paso 3: Eliminar los recursos

En primer lugar, elimine la propia canalización.

Eliminar la canalización mediante la consola de EventBridge

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Canalizaciones.
3. Seleccione la canalización de `pipe-tutorial` y seleccione Eliminar.

A continuación, elimine la pila de CloudFormation para evitar que se le facture por el uso continuo de los recursos aprovisionados en ella.

Eliminar los requisitos previos del tutorial mediante la CLI de AWS

- Ejecute el siguiente comando de la CLI, donde `--stack-name` especifica el nombre de su pila:

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

Eliminar los requisitos previos del tutorial mediante la consola de AWS CloudFormation

1. Abra la consola de AWS CloudFormation en https://console.aws.amazon.com/cloudformation.
2. En la página Pilas, selecciona la pila y, a continuación, seleccione Eliminar.
3. Seleccione Eliminar para confirmar su acción.

Plantilla de AWS CloudFormation para generar requisitos previos

Utilice el siguiente JSON para crear una plantilla de CloudFormation para aprovisionar los recursos de origen y destino necesarios para este 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" }
      }
    }
  }
}
```

Generar una AWS CloudFormation plantilla a partir de EventBridge tuberías

AWS CloudFormation le permite configurar y administrar sus AWS recursos en todas las cuentas y regiones de forma centralizada y repetible al tratar la infraestructura como un código. CloudFormation lo hace permitiéndole crear plantillas que definen los recursos que desea aprovisionar y administrar.

EventBridge le permite generar plantillas a partir de las canalizaciones existentes en su cuenta, como una ayuda que le ayudará a empezar a desarrollar CloudFormation plantillas con rapidez. Puede seleccionar una sola canalización o varias para incluirlas en la plantilla. A continuación, puede utilizar estas plantillas como base para crear pilas de recursos gestionados CloudFormation .

Para obtener más información CloudFormation, consulte la Guía del AWS CloudFormation usuario.

Para los autobuses de eventos, puede generar CloudFormation plantillas a partir de los autobuses de eventos y las reglas de los autobuses de eventos.

Recursos incluidos en las plantillas EventBridge de Pipe

Cuando se EventBridge genera la CloudFormation plantilla, se crea un AWS::Pipes::Piperecurso para cada tubería seleccionada. Además, EventBridge incluye los siguientes recursos en las condiciones descritas:

- AWS::Events::ApiDestination

  Si sus canalizaciones incluyen destinos de API, ya sea como enriquecimientos o como objetivos, EventBridge inclúyalos en la CloudFormation plantilla como AWS::Events::ApiDestination recursos.
- AWS::Events::EventBus

  Si sus canalizaciones incluyen un bus de eventos como destino, EventBridge inclúyalo en la CloudFormation plantilla como AWS::Events::EventBus recurso.
- AWS::IAM::Role

  Si ha EventBridge creado un nuevo rol de ejecución al configurar la canalización, puede optar por EventBridge incluir ese rol en la plantilla como un AWS::IAM::Role recurso. EventBridge no incluye los roles que cree. (En cualquier caso, la `RoleArn` propiedad del AWS::Pipes::Pipe recurso contiene el ARN del rol).

Consideraciones a la hora de utilizar CloudFormation plantillas generadas a partir de Pipes EventBridge

Tenga en cuenta los siguientes factores al utilizar una CloudFormation plantilla a partir de la cual ha generado EventBridge:

- EventBridge no incluye ninguna contraseña en la plantilla de generación.

Puede editar la plantilla para incluir parámetros de plantilla que permitan a los usuarios especificar contraseñas u otra información confidencial al utilizar la plantilla para crear o actualizar una CloudFormation pila.

Además, los usuarios pueden usar Secrets Manager para crear un secreto en la región deseada y, a continuación, editar la plantilla generada para emplear parámetros dinámicos.

- Los destinos de la plantilla generada permanecen exactamente como se especificaron en la canalización original. Esto puede provocar problemas entre regiones si no se edita correctamente la plantilla antes de utilizarla para crear pilas en otras regiones.

  Además, la plantilla generada no crea automáticamente los destinos descendentes.

Generar una CloudFormation plantilla a partir de EventBridge Pipes

Para generar una CloudFormation plantilla a partir de una o más tuberías mediante la EventBridge consola, haga lo siguiente:

Para generar una CloudFormation plantilla a partir de una o más tuberías

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, elija Canalizaciones.
3. En Tubos, elija uno o más tubos que desee incluir en la CloudFormation plantilla generada.

   Para una canalización única, también puede elegir el nombre de la canalización para mostrar la página de detalles de la canalización.
4. Elija CloudFormation Plantilla y, a continuación, elija el formato en el que EventBridge desea generar la plantilla: JSON o YAML.

   EventBridge muestra la plantilla, generada en el formato seleccionado.
5. Si ha EventBridge creado una nueva función de ejecución para alguna de las tuberías seleccionadas y EventBridge desea incluir esas funciones en la plantilla, seleccione Incluir las IAM funciones creadas por la consola en su nombre.
6. EventBridge le da la opción de descargar el archivo de plantilla o copiarla en el portapapeles.

   - Seleccione Descargar para descargar el archivo de plantilla.
   - Para copiar la plantilla al portapapeles, seleccione Copiar.

7. Para salir de la plantilla, seleccione Cancelar.

Tolerancia de las aplicaciones a los errores regionales con puntos de conexión globales y replicación de eventos

Puede mejorar la disponibilidad de su aplicación con los puntos de conexión EventBridge globales de Amazon. Los puntos de conexión globales ayudan a hacer que la aplicación sea tolerante a los errores regionales sin coste adicional. Para empezar, debe asignar una comprobación de estado de Amazon Route 53 al punto de conexión. Cuando se inicia la conmutación por error, la comprobación de estado indica “mal estado”. A los pocos minutos del inicio de la conmutación por error, todos los eventos personalizados se direccionan a un bus de eventos en la región secundaria, donde son procesados. Una vez que la comprobación de estado indica “buen estado”, el bus de eventos de la región principal procesa los eventos.

Al utilizar puntos de conexión globales, puede habilitar la replicación de eventos. La replicación de eventos envía todos los eventos personalizados a los buses de eventos de las regiones principal y secundaria utilizando reglas administradas.

> **Note**
>
> Si utiliza buses personalizados, necesitará un bus personalizado en cada región con el mismo nombre y en la misma cuenta para que la conmutación por error funcione correctamente.

Temas

- Objetivos de tiempo de recuperación y punto de recuperación
- Replicación de eventos
- Crear un punto de conexión global
- Trabajar con puntos finales globales mediante un SDK AWS
- Regiones disponibles
- Prácticas recomendadas para trabajar con puntos de conexión globales de Amazon EventBridge
- Plantilla de AWS CloudFormation para configurar la comprobación de estado de Route 53

Objetivos de tiempo de recuperación y punto de recuperación

El objetivo de tiempo de recuperación (RTO) es el tiempo que tarda la región secundaria en empezar a recibir eventos tras un error. En el caso de RTO, el tiempo incluye el período de tiempo para activar CloudWatch las alarmas y actualizar los estados de las comprobaciones de estado de Route 53. El objetivo de punto de recuperación (RPO) es la medida de los datos que quedarán sin procesar durante una fallo. Para el RPO, el tiempo incluye los eventos que no se replican en la región secundaria y que permanecen atrapados en la región principal hasta que el servicio o la región se recuperen. Con puntos de conexión globales, si sigue nuestras directrices prescriptivas para la configuración de alarmas, puede esperar que el RTO y el RPO duren 360 segundos, con un máximo de 420 segundos.

Replicación de eventos

Los eventos se procesan en la región secundaria de forma asíncrona. Esto significa que no se garantiza que los eventos se procesen al mismo tiempo en ambas regiones. Cuando se activa la conmutación por error, los eventos son procesados por la región secundaria y serán procesados por la región principal cuando esté disponible. La habilitación de la replicación de eventos aumentará sus costes mensuales. Para obtener más información, consulta los EventBridgeprecios de Amazon

Recomendamos habilitar la replicación de eventos al configurar los puntos de conexión globales por los siguientes motivos:

- La replicación de eventos le ayuda a comprobar que los puntos de conexión globales están configurados correctamente. Esto ayuda a garantizar que estará protegido en caso de una conmutación por error.
- La replicación de eventos es necesaria para recuperarse automáticamente de una conmutación por error. Si no tiene habilitada la replicación de eventos, tendrá que restablecer manualmente la comprobación de estado de Route 53 a “buen estado” para que los eventos vuelvan a la región principal.

Carga de eventos replicados

A continuación, se muestra un ejemplo de carga de evento replicado:

> **Note**
>
> Para `region`, la región desde la que se replicó el evento aparece en la lista.

```
{
    "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"
    }
}
```

Crear un punto de conexión global

Complete los siguientes pasos para configurar un punto de conexión global:

1. Asegúrese de que los buses d e eventos y las reglas coincidan tanto en la región principal como en la secundaria.
2. Cree una comprobación de estado de Route 53 para supervisar los buses de eventos. Si necesita ayuda para crear la comprobación de estado, elija Comprobación de estado nueva al crear el punto de conexión global.
3. Crear el punto de conexión global.

Una vez que haya configurado la comprobación de estado de Route 53, puede crear un punto de conexión global.

Para crear un punto de conexión global mediante la consola

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.

2. En el panel de navegación, seleccione Puntos de conexión de entrada.
3. Seleccione Crear punto de conexión.
4. Escriba un nombre y la descripción del punto de conexión.
5. Para Bus de eventos de la región principal, seleccione el bus de eventos al que desee asociar el punto de conexión.
6. En Región secundaria, seleccione la región a la que quiere dirigir los eventos en caso de que se produzca una conmutación por error.

> **Note**
>
> El bus de eventos de la región secundaria se rellena automáticamente y no se puede editar.

7. En Comprobación de estado de Route 53 para desencadenar conmutación por error y recuperación, seleccione la comprobación de estado que supervisará el punto de conexión. Si aún no tienes un chequeo de estado, selecciona Nuevo chequeo de estado para abrir la AWS CloudFormation consola y crear un chequeo de estado usando una CloudFormation plantilla.

> **Note**
>
> Si faltan datos, se producirá un error en la comprobación de estado. Si solo necesitas enviar los eventos de forma intermitente, considera la posibilidad de utilizar una más larga MinimumEvaluationPeriodo tratar los datos faltantes como «ausentes» en lugar de como «incorrectos».

8. (Opcional) Para Replicación de eventos, haga lo siguiente:
   a. Seleccione Replicación de eventos habilitada.
   b. Para Rol de ejecución, seleccione si crear un nuevo rol de AWS Identity and Access Management o usar un rol existente. Haga lo siguiente:
      - Seleccione Crear un rol nuevo para este recurso específico. Si lo desea, puede actualizar el nombre del rol para crear un rol nuevo.
      - Seleccione Usar rol existente. A continuación, en Rol de ejecución, seleccione el rol que desee utilizar.
9. Seleccione Crear.

Para crear un punto de conexión global con la API

Para crear un punto final global mediante la EventBridge API, consulta la referencia CreateEndpointde la EventBridge API de Amazon.

Para crear un punto de conexión global usando AWS CloudFormation

Para crear un punto final global mediante la AWS CloudFormation API, consulte AWS::Events::Endpointsla Guía del AWS CloudFormation usuario.

Trabajar con puntos finales globales mediante un SDK AWS

> **Note**
>
> La compatibilidad para C++ estará disponible próximamente.

Cuando utilices un AWS SDK para trabajar con puntos finales globales, ten en cuenta lo siguiente:

- Necesitarás tener instalada la biblioteca AWS Common Runtime (CRT) para tu SDK específico. Si no tiene la biblioteca CRT instalada, recibirá un mensaje de excepción en el que se indica lo que se debe instalar. Para más información, consulte los siguientes temas:
  - Bibliotecas Common Runtime (CRT) de AWS
  - awslabs/ aws-crt-java
  - aserras/ aws-crt-nodejs
  - aserras/ aws-crt-python
- Una vez que haya creado un punto de conexión global, tendrá que añadir el `endpointId` y `EventBusName` a todas las llamadas `PutEvents` que utilice.
- Los puntos de conexión globales admiten la versión 4A de Signature. Esta versión de SigV4 permite que las solicitudes se firmen para múltiples Regiones de AWS. Esto es útil en operaciones de API que podrían dar como resultado el acceso a datos desde una de multirregiones. Al usar el AWS SDK, usted proporciona sus credenciales y las solicitudes a los puntos finales globales utilizarán la versión 4A de Signature sin necesidad de configuración adicional. Para obtener más información acerca de SigV4A, consulte Firma de solicitudes de API de AWS en la AWS Referencia general.

Si solicitas credenciales temporales desde el AWS STS punto de enlace global (sts.amazonaws.com), vende credenciales que, de forma predeterminada, no AWS STS son compatibles con SigV4a. Consulte Gestión AWS STS en una AWS región en la Guía del usuario para obtener más información.AWS Identity and Access Management

Regiones disponibles

Las siguientes regiones admiten puntos de conexión globales:

- Este de EE. UU. (Norte de Virginia)
- Este de EE. UU. (Ohio)
- Oeste de EE. UU. (Norte de California)
- Oeste de EE. UU. (Oregón)
- Canadá (centro)
- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milán)
- Europa (París)
- Europa (Estocolmo)
- Asia-Pacífico (Bombay)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Tokio)
- América del Sur (São Paulo)

Prácticas recomendadas para trabajar con puntos de conexión globales de Amazon EventBridge

Se recomiendan las siguientes prácticas para configurar puntos de conexión globales.

Temas

Habilitar la replicación de eventos

Le recomendamos encarecidamente que active la replicación y procese sus eventos en la región secundaria que asigne a su punto de conexión global. Esto garantiza que su aplicación de la región secundaria esté configurada correctamente. También debe activar la replicación para garantizar la recuperación automática en la región principal una vez que se haya mitigado un problema.

Los ID de los eventos pueden cambiar entre llamadas a la API, por lo que para correlacionar los eventos entre regiones es necesario disponer de un identificador único e inmutable. Los consumidores también deben diseñarse teniendo en cuenta la idempotencia. De esta forma, si replica eventos o los reproduce desde archivos, no habrá efectos secundarios por el procesamiento de los eventos en ambas regiones.

Cómo evitar la limitación de eventos

Para evitar que los eventos se limiten, le recomendamos que actualice sus límites de `PutEvents` y destinos para que sean uniformes en todas las regiones.

Usar métricas de suscriptor en las comprobaciones de estado de Amazon Route 53

Evite incluir métricas de suscriptor en sus comprobaciones de estado de Amazon Route 53. La inclusión de estas métricas puede provocar que el publicador se conmute por error a las regiones secundarias si un suscriptor encuentra un problema a pesar de que todos los demás suscriptores permanecen en buen estado en la región principal. Si uno de sus suscriptores no procesa los eventos en la región principal, debe activar la replicación para asegurarse de que el suscriptor de la región secundaria pueda procesar los eventos correctamente.

Plantilla de AWS CloudFormation para configurar la comprobación de estado de Route 53

Cuando utilice puntos de conexión globales, debe someterse a una comprobación de estado de Route 53 para supervisar el estado de sus regiones. La siguiente plantilla define una alarma de Amazon CloudWatch y la utiliza para definir una comprobación de estado de Route 53.

Temas

Plantilla de AWS CloudFormation para definir una comprobación de estado de Route 53

Use la siguiente plantilla para definir su comprobación de estado de 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
```

Los ID de los eventos pueden cambiar entre llamadas a la API, por lo que para correlacionar los eventos entre regiones es necesario disponer de un identificador único e inmutable. Los consumidores también deben diseñarse teniendo en cuenta la idempotencia. De esta forma, si replica eventos o los reproduce desde archivos, no habrá efectos secundarios por el procesamiento de los eventos en ambas regiones.

Propiedades de la plantilla de alarmas de CloudWatch

> **Note**
>
> Para todos los campos **`editable`**, tenga en cuenta el rendimiento por segundo. Si solo envía los eventos de forma intermitente, considere la posibilidad de cambiar la comprobación de estado para utilizar un período de evaluación más largo o, en su lugar, tratar los datos faltantes como `missing`, en lugar de `breaching`.

Las siguientes propiedades se utilizan en la sección de alarmas de CloudWatch de la plantilla:

Métrica	Descripción
`AlarmDescription`	La descripción de la alarma. Valor predeterminado: **`High Latency in Amazon EventBridge`**
`MetricName`	El nombre de la métrica asociada a la alarma. Esto es necesario para alarmas basadas en métricas. Para alarmas basadas en expresiones matemáticas, se utiliza en cambio `Metrics` y no se puede especificar `MetricName` . Valor predeterminado: ingestionToInvocationStartLatency
`Namespace`	El espacio de nombres de la métrica asociada a la alarma. Esto es necesario para alarmas basadas en métricas. Para alarmas basadas en expresiones matemáticas, no puede especificar `Namespace` . En cambio, debe utilizar `Metrics`. Valor predeterminado: `AWS/Events`
`Statistic`	La estadística para la métrica asociada a la alarma, aparte de percentil. Valor predeterminado: Promedio
`Period`	El periodo, en segundos, durante el cual se aplica la estadística. Esto es necesario para alarmas basadas en métricas. Los valores válidos son 10, 30, 60 y cualquier múltiplo de 60.

Métrica	Descripción
	Valor predeterminado: **`60`**
`EvaluationPeriods`	El número de periodos en los que los datos se comparan con el umbral especificado. Si configura una alarma que requiere que se infrinjan varios puntos de datos consecutivos para desencadenar la alarma, este valor especifica ese número. Si configura una alarma “M de N”, este valor es la N y `DatapointsToAlarm` es la M. Valor predeterminado: **`5`**
`Threshold`	El valor para comparar con la estadística especificada. Valor predeterminado: **`30,000`**
`ComparisonOperator`	La operación aritmética que debe usar al comparar el umbral y la estadística especificados. El valor de estadística especificado se utiliza como el primer operando. Valor predeterminado: `GreaterThanThreshold`
`TreatMissingData`	Establece cómo administra esta alarma los puntos de datos que faltan. Los valores aceptados son: `breaching` , `notBreaching` , `ignore` y `missing` Valor predeterminado: `breaching`

Propiedades de la plantilla de comprobación de estado de Route 53

> **Note**
>
> Para todos los campos **`editable`**, tenga en cuenta el rendimiento por segundo. Si solo envía los eventos de forma intermitente, considere la posibilidad de cambiar la comprobación de estado para utilizar un período de evaluación más largo o, en su lugar, tratar los datos faltantes como `missing`, en lugar de `breaching`.

Las siguientes propiedades se utilizan en la sección de comprobación de estado de Route 53 de la plantilla:

Métrica	Descripción
`HealthCheckName`	El nombre de la comprobación de estado. Valor predeterminado: **`LatencyFailuresHealthCheck`**
`InsufficientDataHealthStatus`	Cuando CloudWatch no tiene suficientes datos sobre la métrica para determinar el estado de alarma, el estado que desea que Amazon Route 53 asigne a la comprobación de estado Valores válidos: • `Healthy`: Route 53 considera que la comprobación de estado es correcta. • `Unhealthy` : Route 53 considera que la comprobación de estado no es correcta. • `LastKnownStatus` : Route 53 utiliza el estado de la comprobación de estado de la última vez que CloudWatch tenía datos suficientes para determinar el estado de alarma. En el caso de las nuevas comprobaciones de estado que no tienen un último estado conocido, el estado predeterminado para la comprobación de estado es correcto. Valor predeterminado: Mal estado **Note** Este campo se actualiza en función de la entrada en el campo `TreatMissingData` . Si `TreatingMissingData` se establece en`Missing`, se actualizará a `LastKnownStatus` . Si `TreatingMissingData` se establece en `Breaching` , se actualizará a `Unhealthy` .

EventBridge Esquemas de Amazon

Un esquema define la estructura de los eventos a EventBridge los que se envían. EventBridge proporciona esquemas para todos los eventos generados por los AWS servicios. También puede crear o subir esquemas personalizados o inferir esquemas de forma directa de eventos en un bus de eventos. Una vez que tenga un esquema para un evento, puede descargar enlaces de código para lenguajes de programación populares y agilizar el desarrollo. Puede trabajar con enlaces de código para los esquemas y administrarlos desde la EventBridge consola, mediante la API, o directamente en su IDE mediante los kits de herramientas. AWS Para crear aplicaciones sin servidor que usen eventos, utilice AWS Serverless Application Model.

> **Note**
>
> Cuando se utiliza la característica transformador de entrada, el evento original se infiere mediante la detección del esquema, no el evento transformado que se envía al destino.

EventBridge admite los formatos OpenAPI 3 y JSONSchema Draft4.

En el caso de AWSAWS Toolkit for JetBrains y Toolkit for VS Code, puede explorar o buscar esquemas y descargar los enlaces de código de los esquemas directamente en su IDE.

En el siguiente vídeo se ofrece una descripción general de los esquemas y registros de esquemas: Uso del registro de esquemas

Temas

- Enmascaramiento de valores de propiedades de la API de registro de esquemas
- Búsqueda de un EventBridge esquema de Amazon
- Registros EventBridge de esquemas de Amazon
- Creación de un EventBridge esquema de Amazon
- EventBridge Encuadernaciones de código de Amazon

Enmascaramiento de valores de propiedades de la API de registro de esquemas

Algunos valores de propiedades de los eventos que se utilizan para crear un registro de esquemas pueden contener información confidencia l del cliente. Para proteger la información del cliente, los valores se enmascararán con asteriscos (*). Como estamos ocultando estos valores, se EventBridge recomienda no crear aplicaciones que dependan explícitamente de las siguientes propiedades o sus valores:

- CreateSchema— La `Content` propiedad del cuerpo `requestParameters`
- GetDiscoveredSchema— La `Events` propiedad del `requestParameters` cuerpo y la `Content` propiedad del `responseElements` cuerpo
- SearchSchemas— La `keywords` propiedad del `requestParameters`
- UpdateSchema— La `Content` propiedad del `requestParameters`

Búsqueda de un EventBridge esquema de Amazon

EventBridge incluye esquemas para todos los AWS servicios que generan eventos. Puede encontrar estos esquemas en la EventBridge consola o puede encontrarlos mediante la acción de la API. `SearchSchemas`

Para buscar esquemas de AWS servicios en la consola EventBridge

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Esquemas.
3. En la página Esquemas, seleccione Registro de esquemas de eventos de AWS .
   <result>

   Se muestra la primera página de esquemas disponible.
   </result>
4. Para buscar un esquema, en Buscar esquemas de AWS eventos, ingresa un término de búsqueda.

   Una búsqueda devuelve coincidencias para el nombre y el contenido de los esquemas disponibles y, seguidamente, muestra qué versiones del esquema contiene coincidencias.
5. Abra un esquema de eventos seleccionando el nombre del esquema.

Registros EventBridge de esquemas de Amazon

Los registros de esquemas son contenedores para esquemas. Los registros de esquemas recopilan y organizan esquemas en grupos lógicos. Los registros de esquemas predeterminados son:

- Todos los esquemas: todos los esquemas de los registros de AWS eventos, detectados y personalizados.
- AWS registro de esquemas de eventos: los esquemas integrados.
- Registro de esquemas detectados: los esquemas detectados por Schema Discovery.

También puede crear registros personalizados para organizar los esquemas que cree o suba.

Para crear un registro personalizado

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Esquemas y, a continuación, Crear registro.
3. En la página Detalles del registro, escriba un nombre en Nombre.
4. (Opcional) Escriba una descripción para el registro nuevo.
5. Seleccione Crear.

Para crear un esquema personalizado en el registro nuevo, seleccione Crear esquema personalizado. Para agregar un esquema a su registro, selecciónelo cuando cree un esquema nuevo.

Para crear un registro mediante la API, utilice `CreateRegistry`. Para obtener más información, consulte la referencia de la API de Amazon EventBridge Schema Registry.

Para obtener más información sobre cómo utilizar el registro de EventBridge esquemas AWS CloudFormation, consulte la referencia de tipos de EventSchemas recursos en AWS CloudFormation.

Creación de un EventBridge esquema de Amazon

Los esquemas se crean mediante archivos JSON con la especificación OpenAPI o la especificación JSONSchema Draft4. Puede crear o cargar sus propios esquemas EventBridge utilizando una plantilla o generando un esquema basado en el JSON de un evento. También puede inferir el esquema a partir de los eventos de un bus de eventos. Para crear un esquema mediante la API de registro de EventBridge esquemas, usa la acción de la `CreateSchema`API.

Cuando seleccione entre los formatos OpenAPI 3 y JSONSchema Draft4, tenga en cuenta las siguientes diferencias:

- El formato JSONSchema admite palabras clave adicionales que no son compatibles con OpenAPI, como `$schema, additionalItems`.
- Existen pequeñas diferencias en la forma en que se gestionan las palabras clave, como `type` y `format`.
- OpenAPI no admite hipervínculos de hiperesquema JSONSchema en documentos JSON.
- Las herramientas para OpenAPI tienden a centrarse en el tiempo de compilación, mientras que las herramientas para JSONSchema tienden a centrarse en las operaciones de tiempo de ejecución, como las herramientas de cliente para la validación de esquemas.

Recomendamos usar el formato JSONSchema para implementar la validación del lado del cliente, de modo que los eventos enviados EventBridge se ajusten al esquema. Puede usar JSONSchema para definir un contrato para documentos JSON válidos y, a continuación, usar un validador de esquemas JSON antes de enviar los eventos asociados.

Cuando tenga un esquema nuevo, podrá descargar los enlaces de código que te ayudarán a crear aplicaciones para eventos con ese esquema.

Temas

- Crear un esquema mediante una plantilla
- Editar una plantilla de esquema directamente en la consola
- Cree un esquema a partir del formato JSON de un evento
- Cree un esquema a partir de los eventos de un bus de eventos

Crear un esquema mediante una plantilla

Puede crear un esquema a partir de una plantilla o editarla directamente en la consola. EventBridge Para obtener la plantilla, debe descargarla de la consola. Puede editar la plantilla para que el esquema coincida con sus eventos. A continuación, suba la nueva plantilla a través de la consola.

Para descargar la plantilla de esquema

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Registro de esquema.
3. En la sección Introducción en Plantilla de esquema, seleccione Descargar.

Si lo prefiere, puede copiar la plantilla JSON desde el siguiente ejemplo de código.

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

Para subir una plantilla de esquema

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Esquemas y, a continuación, Crear esquema.
3. (Opcional) Seleccione o cree un registro de esquema.
4. En Detalles del esquema, escriba un nombre para su esquema.
5. (Opcional) Escriba una descripción para su esquema.
6. En Tipo de esquema, seleccione OpenAPI 3.0 o JSON Schema Draft 4.
7. En la pestaña Crear, en el cuadro de texto, arrastre su archivo de esquema al cuadro de texto o pegue el origen del esquema.
8. Seleccione Crear.

Editar una plantilla de esquema directamente en la consola

Para editar un esquema en la consola

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Esquemas y, a continuación, Crear esquema.
3. (Opcional) Seleccione o cree un registro de esquema.
4. En Detalles del esquema, escriba un nombre para su esquema.
5. En Tipo de esquema, seleccione OpenAPI 3.0 o JSON Schema Draft 4.

6. (Opcional) Escriba una descripción para el esquema que desea crear.
7. En la pestaña Crear, seleccione Cargar plantilla.
8. En el cuadro de texto, edite la plantilla para que el esquema coincida con sus eventos.
9. Seleccione Crear.

Cree un esquema a partir del formato JSON de un evento

Si tiene el formato JSON de un evento, puede crear automáticamente un esquema para ese tipo de evento.

Para crear un esquema basado en el formato JSON de un evento

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Esquemas y, a continuación, Crear esquema.
3. (Opcional) Seleccione o cree un registro de esquema.
4. En Detalles del esquema, escriba un nombre para su esquema.
5. (Opcional) Escriba una descripción para el esquema creado.
6. En Tipo de esquema, seleccione OpenAPI 3.0.

   No puede usar JSONSchema al crear un esquema a partir del formato JSON de un evento.
7. Seleccione Descubrir de JSON
8. En el cuadro de texto en JSON, pegue o arrastre el origen JSON de un evento.

   Por ejemplo, puedes pegar el código fuente de este AWS Step Functions evento si se produce un error en la ejecución.

```
{
    "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. Seleccione Detectar esquema.
10. EventBridge genera un esquema OpenAPI para el evento. Por ejemplo, se genera el siguiente esquema para el evento Step Functions anterior.

```
{
  "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 vez generado el esquema, seleccione Crear.

Cree un esquema a partir de los eventos de un bus de eventos

EventBridge puede deducir esquemas mediante el descubrimiento de eventos. Para inferir los esquemas, se activa la detección de eventos en un bus de eventos y cada esquema único se agrega al registro de esquemas, incluidos los de los eventos entre cuentas. Los esquemas descubiertos EventBridge aparecen en el registro de esquemas descubiertos de la página Esquemas.

Si el contenido de los eventos del bus de eventos cambia, EventBridge crea nuevas versiones del esquema relacionado. EventBridge

> **Note**
>
> La habilitación de la detección de eventos en un bus de eventos puede generar costes. Los primeros cinco millones de eventos procesados cada mes son gratuitos.

> **Note**
>
> EventBridge deduce esquemas a partir de eventos entre cuentas de forma predeterminada, pero puede deshabilitarlo actualizando la propiedad. `cross-account` Para obtener más información, consulte Discoverers in the EventBridge Schema Registry API Reference.

Para habilitar la detección de esquemas en un bus de eventos

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Buses de eventos.
3. Realice una de las siguientes acciones siguientes:
   - Para habilitar la detección en el Bus de eventos predeterminado, seleccione Comenzar detección.
   - Para habilitar la detección en un Bus de eventos personalizado, seleccione el botón de opción del bus de eventos personalizado y seleccione Comenzar detección.

EventBridge Encuadernaciones de código de Amazon

Puede generar enlaces de código para esquemas de eventos para acelerar el desarrollo en Golang, Java, Python y. TypeScript Los enlaces de código están disponibles para los eventos de servicios de AWS , los esquemas que crea y los esquemas que genera en función de eventos en un bus de eventos. Puede generar enlaces de código para un esquema mediante la EventBridge consola, la API de EventBridge Schema Registry o en su IDE con un kit de herramientas. AWS

Para generar enlaces de código a partir de un esquema EventBridge

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Esquemas.
3. Encuentre un esquema para el que desee enlaces de código, ya sea buscando en los registros de esquemas o buscando un esquema.
4. Seleccione el nombre del esquema.
5. En la página Detalles del esquema, en la sección Versión, seleccione Descargar enlaces de código.
6. En la página Descargar enlaces de código, seleccione el lenguaje de los enlaces de código que desea descargar.
7. Seleccione Descargar.

   La descarga puede tardar unos segundos en comenzar. El archivo descargado es un archivo zip de enlaces de código para el lenguaje seleccionado.

Herramientas y servicios relacionados con Amazon EventBridge

Amazon EventBridge trabaja con otras herramientas y servicios de AWS para procesar eventos o invocar un recurso como destino de una regla. Para obtener más información acerca de las integraciones de EventBridge con otros servicios de AWS, consulte lo siguiente.

Temas

- Usar Amazon EventBridge con puntos de conexión de VPC de tipo interfaz
- Integración de Amazon EventBridge con AWS X-Ray
- Uso EventBridge con el kit de prueba de aplicaciones AWS integrado
- Incluir EventBridge recursos de Amazon en AWS CloudFormation pilas

Usar Amazon EventBridge con puntos de conexión de VPC de tipo interfaz

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus recursos de AWS, puede establecer una conexión privada entre su VPC y EventBridge. Los recursos de su VPC pueden utilizar esta conexión para comunicarse con EventBridge.

Con una VPC, puede controlar la configuración de la red, como el rango de direcciones IP, las subredes, las tablas de enrutamiento y las puertas de enlace de red. Para conectar su VPC a EventBridge, debe definir un punto de conexión de VPC de tipo interfaz para EventBridge. El punto de conexión ofrece conectividad fiable y escalable con EventBridge sin necesidad de utilizar una puerta de enlace de Internet, una instancia de traducción de direcciones de red (NAT) o una conexión de VPN. Para obtener más información, consulte Qué es Amazon VPC en la Guía del usuario de Amazon VPC.

Los puntos de conexión de VPC de tipo interfaz utilizan la tecnología AWS PrivateLink, que permite la comunicación privada entre los servicios de AWS mediante una interfaz de red elástica con direcciones IP privadas. Para obtener más información, consulte Puntos de conexión de VPC y AWS PrivateLink.

Cuando utiliza un punto de conexión de VPC de tipo interfaz privada, los eventos personalizados que la VPC envía a EventBridge utilizan ese punto de conexión. A continuación, EventBridge envía esos eventos a otros servicios de AWS en función de las reglas y los destinos que haya configurado. Una vez que los eventos se envían a otro servicio, puede recibirlos a través del punto de conexión público o de un punto de conexión de VPC para ese servicio. Por ejemplo, si crea una regla para enviar eventos a una cola de Amazon SQS, puede configurar un punto de conexión de VPC de tipo interfaz para que Amazon SQS reciba mensajes de esa cola en su VPC sin utilizar el punto de conexión público.

Disponibilidad

Actualmente, EventBridge admite puntos de conexión de VPC en las regiones siguientes:

- Este de EE. UU. (Ohio)
- Este de EE. UU. (Norte de Virginia)
- Oeste de EE. UU. (Norte de California)
- Oeste de EE. UU. (Oregón)

- África (Ciudad del Cabo)
- Asia-Pacífico (Bombay)
- Asia-Pacífico (Hyderabad)
- Asia-Pacífico (Hong Kong)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Yakarta)
- Asia-Pacífico (Melbourne)
- Asia-Pacífico (Tokio)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Osaka)
- Canadá (Centro)
- Oeste de Canadá (Calgary)
- China (Pekín)
- China (Ningxia)
- Europa (Fráncfort)
- Europa (Zúrich)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milán)
- Europa (España)
- Europa (París)
- Europa (Estocolmo)
- Oriente Medio (EAU)
- Medio Oriente (Baréin)
- América del Sur (São Paulo)
- Israel (Tel Aviv)
- AWS GovCloud (EE. UU. Oeste)
- AWS GovCloud (Este de EE. UU.)

Crear un punto de conexión de VPC para EventBridge

Para utilizar EventBridge con su VPC, cree un punto de conexión de VPC de tipo interfaz para EventBridge y seleccione com.amazonaws.***Region***.events como nombre del servicio. Para obtener más información, consulte Crear un punto de conexión de tipo interfaz en la Guía del usuario de Amazon VPC.

Aspectos específicos de EventBridge Pipes

El soporte completo de EventBridge Pipes para los puntos de conexión de VPC de tipo interfaz no está disponible. Para utilizar los siguientes orígenes en una VPC con EventBridge Pipes, consulte lo siguiente:

- Configuración de la red de Amazon MSK
- Configuración de la red de Apache Kafka autoadministrado
- Configuración de la red de Amazon MQ

Integración de Amazon EventBridge con AWS X-Ray

Se puede utilizar AWS X-Ray para rastrear los eventos que se transfieren a través de EventBridge. EventBridge transfiere el encabezado de seguimiento original al destino para que los servicios de destino puedan realizar un seguimiento, analizar y depurar.

EventBridge puede transferir un encabezado de seguimiento para un evento solo si el evento proviene de una solicitud `PutEvents` que pasó el contexto de seguimiento. X-Ray no rastrea los eventos que se originan en socios externos, eventos programados o servicios de AWS, y estos orígenes de eventos no aparecen en el mapa de servicios de X-Ray.

X-Ray valida los encabezados de seguimiento y los encabezados de seguimiento que no son válidos se descartan. Sin embargo, el evento sigue procesado.

> **Important**
>
> El encabezado de seguimiento no está disponible en el evento que se envía al destino de invocación.
>
> - Si tiene un archivo de eventos, el encabezado de seguimiento no está disponible en los eventos archivados. Si reproduce los eventos archivados, el encabezado de seguimiento no está incluido.
> - Si tiene una cola de mensajes fallidos (DLQ), el encabezado de seguimiento se incluye en la solicitud `SendMessage` que envía el evento a la DLQ. Si recupera eventos (mensajes) de la DLQ mediante el uso de `ReceiveMessage`, el encabezado de seguimiento asociado al evento se incluye en el atributo de mensaje de Amazon SQS, pero no se incluye en el mensaje del evento.

Para obtener información sobre cómo un nodo de eventos de EventBridge conecta los servicios de origen y destino, consulte Visualizar orígenes y destinos en el mapa del servicios de X-Ray en la Guía para desarrolladores de AWS X-Ray.

Puede transferir la siguiente información del encabezado de seguimiento a través de EventBridge:

- Encabezado HTTP predeterminado: el SDK de X-Ray rellena automáticamente el encabezado de seguimiento como encabezado HTTP `X-Amzn-Trace-Id` para todos los destinos de invocación. Para obtener más información sobre el encabezado HTTP predeterminado, consulte Encabezado de seguimiento en la Guía para desarrolladores de AWS X-Ray.

- Atributo del sistema **`TraceHeader`**: `TraceHeader` es un atributo PutEventsRequestEntry reservado por EventBridge para transportar el encabezado de seguimiento de X-Ray a un destino. Si también usa `PutEventsRequestEntry`, `PutEventsRequestEntry` anula el encabezado de seguimiento HTTP.

> **Note**
>
> El encabezado de seguimiento no cuenta para el tamaño del evento `PutEventsRequestEntry`. Para obtener más información, consulte Calcular el tamaño de la entrada a EventBridge PutEvents un evento de Amazon.

En el siguiente vídeo se muestra el uso conjunto de X-Ray y EventBridge: Uso de AWS X-Ray para el seguimiento

Uso EventBridge con el kit de prueba de aplicaciones AWS integrado

Al crear aplicaciones compuestas por servicios sin servidor, como Lambda o Step Functions EventBridge, muchos de los componentes de la arquitectura no se pueden implementar en el escritorio, sino que solo existen en AWS la nube. A diferencia de trabajar con aplicaciones implementadas localmente, estos tipos de aplicaciones se benefician de las estrategias basadas en la nube para realizar pruebas automatizadas. AWS El kit integrado de pruebas de aplicaciones (AWS IATK) le ayuda a implementar algunas de estas estrategias para sus aplicaciones.

AWS El IATK es una biblioteca de software que le ayuda a escribir pruebas automatizadas para aplicaciones basadas en la nube.

EventBridge integración con IATK AWS

Puede usar EventBridge eventos y buses de eventos con AWS IATK para implementar sus pruebas automatizadas, que incluyen:

Implementación de herramientas de ejecución de pruebas

Para escribir pruebas de integración para arquitecturas basadas en eventos, establezca límites lógicos dividiendo su aplicación en subsistemas. Una técnica útil para probar subsistemas consiste en crear herramientas de ejecución de pruebas, es decir, recursos que crea específicamente para probar subsistemas.

Por ejemplo, una prueba de integración puede iniciar un proceso de subsistema al transferirle un evento de prueba de entrada. AWS IATK puede crearle un arnés de pruebas que escuche los eventos de salida. EventBridge (Bajo el capó, el arnés está compuesto por una EventBridge regla que reenvía el evento de salida a Amazon SQS). A continuación, la prueba de integración consulta a continuación la herramienta para examinar la salida y determinar si la prueba se supera o no.

Generación de eventos simulados

AWS IATK le permite generar eventos simulados a partir de un esquema almacenado en el registro de esquemas. EventBridge Esto le permite generar un evento simulado e invocar a cualquier consumidor (como una función de Lambda o una máquina de estados Step Functions) con el evento generado.

Para obtener más información, consulte la descripción general del kit de prueba de aplicaciones AWS integradas en. GitHub

Incluir EventBridge recursos de Amazon en AWS CloudFormation pilas

AWS CloudFormation le permite configurar y administrar sus AWS recursos en todas las cuentas y regiones de forma centralizada y repetible al tratar la infraestructura como un código. CloudFormation lo hace permitiéndole crear plantillas que definen los recursos que desea aprovisionar y administrar. Estos recursos pueden incluir EventBridge elementos como los autobuses y las reglas de los eventos, las canalizaciones, los esquemas y los horarios, entre otros. Utilice estos recursos para incluir EventBridge funcionalidades en los paquetes de tecnología que aprovisiona y administra. CloudFormation

EventBridge Recursos de Amazon disponibles en AWS CloudFormation

EventBridge proporciona recursos para su uso en CloudFormation plantillas en los siguientes espacios de nombres de recursos:

- AWS::Events

  Entre los ejemplos de plantillas se incluyen:

  - Cree un destino de API para PagerDuty
  - Creación de un destino de API para Slack
  - Cree una conexión con parámetros ApiKey de autorización
  - Creación de una conexión con parámetros de autorización OAuth
  - Creación de un punto de conexión global con replicación de eventos
  - Denegación de directiva mediante múltiples entidades y acciones
  - Concesión de permisos a una organización mediante un bus de eventos personalizado
  - Creación de una regla interregional
  - Creación de una regla que incluya una cola de mensajes fallidos para un destino
  - Invocación normal de una función de Lambda
  - Invocación a la función de Lambda en respuesta a un evento
  - Notificación de un tema en respuesta a una entrada de registro
- AWS::EventEsquemas
- AWS::Pipes

  Entre los ejemplos de plantillas se incluyen:

  - Creación de una canalización con un filtro de eventos
- AWS::Scheduler

Generación de definiciones EventBridge de recursos de Amazon para AWS CloudFormation plantillas

Como ayuda para ayudarte a empezar a desarrollar CloudFormation plantillas, la EventBridge consola te permite crear CloudFormation plantillas a partir de los procesos, reglas y canales de eventos existentes en tu cuenta.

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

Administrar el bus de eventos predeterminado AWS CloudFormation

Como EventBridge aprovisiona automáticamente el bus de eventos predeterminado en su cuenta, no puede crearlo mediante una CloudFormation plantilla, como haría normalmente con cualquier recurso que quisiera incluir en una CloudFormation pila. Para incluir el bus de eventos predeterminado en una CloudFormation pila, primero debe importarlo a una pila. Una vez que haya importado el bus de eventos predeterminado a una pila, podrá actualizar las propiedades del bus de eventos según lo desee.

Para obtener más información, consulte ???.

Gestione los eventos de la AWS CloudFormation pila mediante EventBridge

Además de incluir EventBridge recursos en tus CloudFormation pilas, puedes utilizarlos EventBridge para gestionar los eventos generados por las propias CloudFormation pilas. CloudFormation envía eventos EventBridge cada vez que se realiza una operación de creación, actualización, eliminación o detección de desviaciones en una pila. CloudFormation también envía eventos a EventBridge para cambiar el estado de los conjuntos de pilas y de las instancias de conjuntos de pilas. Puede usar EventBridge reglas para dirigir los eventos a sus objetivos definidos.

Para obtener más información, consulte Administrar CloudFormation eventos mediante EventBridge la Guía del AWS CloudFormation usuario.

Tutoriales de Amazon EventBridge

EventBridge se integra con una serie de servicios de AWS y socios de SaaS. Estos tutoriales están diseñados para ayudarle a familiarizarse con los conceptos básicos de EventBridge y enseñarle cómo puede formar parte de su arquitectura sin servidor.

Tutoriales:

- Tutoriales de introducción a Amazon EventBridge
- Tutoriales de Amazon EventBridge para la integración con otros servicios de AWS
- Tutoriales de Amazon EventBridge para la integración con proveedores de SaaS

Tutoriales de introducción a Amazon EventBridge

Los siguientes tutoriales le ayudan a explorar las características de EventBridge y le enseñan a utilizarlas.

Tutoriales:

- Archivo y reproducción de eventos de Amazon EventBridge
- Creación de una aplicación de muestra de Amazon EventBridge
- Tutorial: Descarga de enlaces de código para eventos mediante el registro de esquemas de EventBridge
- Tutorial: Uso del transformador de entrada para personalizar lo que EventBridge transfiere al destino de eventos

Archivo y reproducción de eventos de Amazon EventBridge

Puede usar EventBridge para enrutar eventos a funciones de AWS Lambda específicas mediante reglas.

En este tutorial, creará una función para usarla como destino de la regla de EventBridge mediante la consola de Lambda. A continuación, creará un archivo y una regla que archivarán los eventos de prueba mediante la consola de EventBridge. Cuando haya eventos en ese archivo, los reproducirá.

Pasos:

- Paso 1: Crear una función de Lambda
- Paso 2: Crear un archivo
- Paso 3: Crear regla
- Paso 4: Enviar eventos de prueba
- Paso 5: Reproducir eventos
- Paso 6: Eliminar los recursos

Paso 1: Crear una función de Lambda

En primer lugar, cree una función de Lambda para registrar los eventos.

Para crear una función de Lambda:

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. Elija Create function (Crear función).
3. Elija Author from scratch (Crear desde cero).
4. Introduzca un nombre y la descripción de la función Lambda. Por ejemplo, asigne un nombre a la función `LogScheduledEvent`.
5. Deje el resto de las opciones como predeterminadas y elija Crear función.
6. En la pestaña Código de la página de funciones, haga doble clic en index.js.
7. Sustituya el código JavaScript existente por el siguiente código:

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogScheduledEvent');
```

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

8. Elija Deploy (Implementar).

Paso 2: Crear un archivo

A continuación, cree el archivo que contendrá todos los eventos de prueba.

Para crear un archivo

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, elija Archivos.
3. Elija Crear archivo.
4. Escriba un nombre y una descripción para el archivo. Por ejemplo, llame al archivo `ArchiveTest`.
5. Deje el resto de las opciones como predeterminadas y elija Siguiente.
6. Elija Crear archivo.

Paso 3: Crear regla

Cree una regla para archivar los eventos que se envían al bus de eventos.

Para crear una regla

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Elija Crear regla.
4. Escriba un nombre y una descripción para la regla. Por ejemplo, llame a la regla `ARTestRule`.

   Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.
5. En Event bus (Bus de eventos), elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.

6. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).
7. Elija Next (Siguiente).
8. En Event source (Origen del evento), elija Other (Otro).
9. En Patrón de evento, introduzca lo siguiente:

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

10. Elija Next (Siguiente).
11. En Target types (Tipos de destino), elija AWS service.
12. En Seleccionar un destino, elija Función de Lambda en la lista desplegable.
13. En Función, seleccione la función de Lambda que creó en la sección Paso 1: Crear una función de Lambda. En este ejemplo, seleccione `LogScheduledEvent`.
14. Elija Next (Siguiente).
15. Elija Next (Siguiente).
16. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 4: Enviar eventos de prueba

Ahora que ha configurado el archivo y la regla, enviaremos los eventos de prueba para asegurarnos de que el archivo funciona correctamente.

> **Note**
>
> Los eventos pueden tardar algún tiempo en llegar al archivo.

Para enviar eventos de prueba (consola)

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, elija Event Buses (Buses de eventos).
3. En el icono del bus de eventos predeterminado, elija Acciones, Enviar eventos.
4. Introduzca un origen de eventos. Por ejemplo, `TestEvent`.

5. En Detail type (Tipo de detalle), introduzca `customerCreated`.
6. En Detalle del evento, introduzca `{}`.
7. Seleccione Send (enviar).

Paso 5: Reproducir eventos

Una vez que los eventos de prueba estén en el archivo, podrá reproducirlos.

Para reproducir eventos archivados (consola)

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, elija Reproducciones.
3. Elija Iniciar reproducción nueva.
4. Escriba un nombre y una descripción para la reproducción. Por ejemplo, llame a la reproducción `ReplayTest`.
5. En Origen, seleccione el archivo que creó en la sección Paso 2: Crear archivo.
6. En Período de reproducción, haga lo siguiente.
    a. En Hora de inicio, seleccione la fecha en la que envió los eventos de prueba y una hora antes de enviarlos. Por ejemplo, `2021/08/11` y `08:00:00`.
    b. En Hora de finalización, seleccione la fecha y la hora actuales. Por ejemplo, `2021/08/11` y `09:15:00`.
7. Elija Iniciar la reproducción.

Paso 6: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar las funciones de Lambda

1. Abra la página de Functions (Funciones) en la consola de Lambda.
2. Seleccione las funciones que ha creado.
3. Elija Actions (Acciones), Delete (Eliminar).
4. Elija Eliminar.

Para eliminar los archivos de EventBridge

1. Abra la página Archivos en la consola de EventBridge.
2. Seleccione los archivos que creó.
3. Elija Eliminar.
4. Introduzca el nombre del archivo y seleccione Eliminar.

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Creación de una aplicación de muestra de Amazon EventBridge

Puede usar EventBridge para enrutar eventos a funciones de Lambda específicas mediante reglas.

En este tutorial, usará la AWS CLI, Node.js, y el código del repositorio de GitHub para crear lo siguiente:

- Una función de AWS Lambda que produce eventos para las transacciones bancarias en cajeros automáticos.
- Tres funciones de Lambda para usar como destinos de una regla de EventBridge.
- y la regla que enruta los eventos creados a la función descendente correcta en función de un patrón de eventos.

En este ejemplo, se utilizan plantillas de AWS SAM para definir las reglas de EventBridge. Para obtener más información sobre el uso de plantillas de AWS SAM con EventBridge, consulte. ???

En el repositorio, el subdirectorio atmProducer contiene `handler.js`, que representa el servicio de cajero automático que produce eventos. Este código es un controlador de Lambda escrito en Node.js y publica eventos en EventBridge a través del SDK de AWS mediante esta línea de código JavaScript.

```
const result = await eventbridge.putEvents(params).promise()
```

Este directorio también contiene `events.js` una lista de varias transacciones de prueba en una matriz de entradas. Un único evento se define en JavaScript de la siguiente manera:

```
{
  // 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 sección Detalle del evento especifica los atributos de la transacción. Incluyen la ubicación del cajero automático, el importe, el banco asociado y el resultado de la transacción.

El archivo `handler.js` del subdirectorio atmConsumer contiene tres funciones:

```
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))
}
```

Cada función recibe eventos de transacciones, que se registran a través de las instrucciones de `console.log` en Registros de Amazon CloudWatch. Las funciones de consumo funcionan de forma independiente del productor y desconocen el origen de los eventos.

La lógica de enrutamiento está incluida en las reglas de EventBridge que implementa la plantilla de AWS SAM de la aplicación. Las reglas evalúan el flujo entrante de eventos y enrutan los eventos coincidentes a las funciones de Lambda de destino.

Las reglas utilizan patrones de eventos que son objetos JSON con la misma estructura que los eventos con los que coinciden. Este es el patrón de eventos de una de las reglas.

```
{
  "detail-type": ["transaction"],
  "source": ["custom.myATMapp"],
```

```
  "detail": {
    "location": [{
      "prefix": "NY-"
    }]
  }
}
```

Pasos:

- Requisitos previos
- Paso 1: Crear una aplicación
- Paso 2: Ejecutar la aplicación
- Paso 3: Verificar los registros y el funcionamiento de la aplicación
- Paso 4: Eliminar los recursos

Requisitos previos

Para completar este tutorial necesitará los siguientes recursos:

- Una cuenta de AWS. Cree una cuenta de AWS si no dispone todavía de una.
- AWS CLI instalada. Para instalar la AWS CLI más reciente, consulte Instalación, actualización y desinstalación de la versión 2 de la AWS CLI.
- Se instaló la versión 12.x de Node.js. Para instalar Node.js, consulte Descargas.

Paso 1: Crear una aplicación

Para configurar la aplicación de ejemplo, usará la AWS CLI y Git para crear los recursos de AWS que necesitará.

Para crear la aplicación

1. Inicie sesión en AWS.
2. Instale Git e instale la CLI de AWS Serverless Application Model en su máquina local.
3. Cree un directorio nuevo y, a continuación, vaya a ese directorio en un terminal.
4. En la línea de comandos, introduzca `git clone https://github.com/aws-samples/amazon-eventbridge-producer-consumer-example`.
5. En la línea de comandos, ejecute el comando siguiente:

```
cd ./amazon-eventbridge-producer-consumer-example
sam deploy --guided
```

6. Haga lo siguiente en el terminal:

   a. En **`Stack Name`**, escriba un nombre para la pila. Por ejemplo, llame a la pila `Test`.
   b. En **`AWS Region`**, introduzca la región. Por ejemplo, `us-west-2`.
   c. En **`Confirm changes before deploy`**, introduzca Y.
   d. En **`Allow SAM CLI IAM role creation`**, introduzca Y
   e. En **`Save arguments to configuration file`**, introduzca Y
   f. En **`SAM configuration file`**, introduzca `samconfig.toml`.
   g. En **`SAM configuration environment`**, introduzca `default`.

Paso 2: Ejecutar la aplicación

Ahora que ha configurado los recursos, utilizará la consola para probar las funciones.

Para ejecutar la aplicación

1. Abra la Consola de Lambda en la misma región en la que implementó la aplicación de AWS SAM.
2. Hay cuatro funciones de Lambda con el prefijo atm-demo. Seleccione la función atmProducerFN y, a continuación, elija Acciones, Prueba.
3. Escriba `Test` en Nombre.
4. Seleccione Test (Probar).

Paso 3: Verificar los registros y el funcionamiento de la aplicación

Ahora que ha ejecutado la aplicación, utilizará la consola para comprobar los Registros de CloudWatch.

Para consultar los registros

1. Abra la Consola de CloudWatch en la misma región en la que implementó la aplicación de AWS SAM.
2. Elija Logs (Registros) y, a continuación, elija Log groups (Grupo de registros).

3. Seleccione el grupo de registro que contiene atmConsumerCase1. Verá dos flujos que representan las dos transacciones aprobadas por el cajero automático. Elija el flujo de registro para ver el resultado.
4. Vuelva a la lista de grupos de registro y, a continuación, seleccione el grupo de registro que contiene atmConsumerCase2. Verá dos flujos que representan las dos transacciones que coinciden con el filtro de ubicación Nueva York.
5. Vuelva a la lista de grupos de registro y, a continuación, seleccione el grupo de registro que contiene atmConsumerCase3. Abra el flujo para ver las transacciones denegadas.

Paso 4: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Para eliminar las funciones de Lambda

1. Abra la página de Functions (Funciones) en la consola de Lambda.
2. Seleccione las funciones que creó.
3. Elija Actions (Acciones), Delete (Eliminar).
4. Elija Eliminar.

Para eliminar los grupos de registro de Registros de CloudWatch

1. Abra la consola de CloudWatch.
2. Elija Registros, Grupos de registro.
3. Seleccione los grupos de registro que se crearon en este tutorial.
4. Elija Acciones, Eliminar grupo(s) de registro(s).
5. Elija Eliminar.

Tutorial: Descarga de enlaces de código para eventos mediante el registro de esquemas de EventBridge

Puede generar enlaces de código para esquemas de eventos para acelerar el desarrollo de Golang, Java, Python y TypeScript. Puede obtener enlaces de código para los servicios de AWSexistentes, los esquemas que cree y los esquemas que genere en función de eventos en un bus de eventos. Puede generar enlaces de código para un esquema, usando uno de los siguientes:

- La consola de EventBridge
- La API de registro de esquemas de EventBridge
- Su IDE con un kit de herramientas de AWS

En este tutorial, generará y descargará enlaces de código desde un esquema de EventBridge para los eventos de un servicio de AWS.

Para generar enlaces de código a partir de un esquema de EventBridge

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, elija Schemas (Esquemas).
3. Seleccione la pestaña Registro de esquemas de eventos de AWS.
4. Encuentre el esquema para un servicio de AWS para el que desee enlaces de código, ya sea explorando el registro de esquemas o buscando un esquema.
5. Seleccione el nombre del esquema.
6. En la página Detalles del esquema, en la sección Versión, seleccione Descargar enlaces de código.
7. En la página Download code bindings (Descargar enlaces de código), seleccione el lenguaje de los enlaces de código que desea descargar.
8. Seleccione Download (Descargar).

   La descarga puede tardar unos segundos en comenzar. El archivo de descarga será un archivo .zip de enlaces de código para el lenguaje seleccionado.
9. Descomprima el archivo descargado y agréguelo al proyecto.

   El paquete descargado contiene un archivo README que explica cómo configurar las dependencias del paquete en varios marcos.

Utilice estos enlaces de código en su propio código para ayudar a crear aplicaciones rápidamente utilizando este evento de EventBridge.

Tutorial: Uso del transformador de entrada para personalizar lo que EventBridge transfiere al destino de eventos

Puede usar el transformador de entrada de EventBridge para personalizar el texto de un evento antes de enviarlo al destino de una regla.

Para ello, debe definir varias rutas de JSON a partir del evento y asignar sus salidas a distintas variables. A continuación, puede utilizar esas variables en la plantilla de entrada. Los caracteres < y > no pueden utilizar un carácter de escape. Para obtener más información, consulte Transformación EventBridge de entradas de Amazon

> **Note**
>
> Si especifica una variable que coincida con una ruta JSON que no existe en el evento, dicha variable no se crea ni aparece en la salida.

En este tutorial, creará una regla que haga coincidir un evento con `detail-type: "customerCreated"`. El transformador de entrada asigna la variable `type` a la ruta JSON $.detail-type desde el evento. A continuación, EventBridge coloca la variable en la plantilla de entrada «Este evento era <tipo>». El resultado es el siguiente mensaje de Amazon SNS.

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

Pasos:

- Paso 1: Crear un tema de Amazon SNS
- Paso 2: Crear una suscripción de Amazon SNS
- Paso 3: Crear una regla
- Paso 4: Enviar eventos de prueba
- Paso 5: Confirmar el éxito
- Paso 6: Eliminar los recursos

Paso 1: Crear un tema de Amazon SNS

Cree un tema para recibir los eventos de EventBridge.

Para crear un tema

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. En el panel de navegación, elija Temas.
3. Elija Create new topic (Crear nuevo tema).
4. En Type (Tipo), seleccione Standard (Estándar).
5. Escriba **`eventbridge-IT-test`** como nombre del tema.
6. Elija Create new topic (Crear nuevo tema).

Paso 2: Crear una suscripción de Amazon SNS

Cree una suscripción para recibir correos electrónicos con la información transformada.

Para crear una suscripción

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. En el panel de navegación, seleccione Subscriptions.
3. Seleccione Create subscription.
4. En la lista ARN de tema, seleccione el tema que creó en el paso 1. Para este tutorial, elija eventbridge-IT-test.
5. En Protocol (Protocolo), elija Email (Correo electrónico).
6. En Punto de enlace, introduzca su dirección de correo electrónico.
7. Seleccione Create subscription.
8. Confirme la suscripción seleccionando Confirmar suscripción en el correo electrónico que reciba de las notificaciones de AWS.

Paso 3: Crear una regla

Cree una regla para usar el transformador de entrada para personalizar la información del estado de la instancia que se envía a un destino.

Para crear una regla

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.

3. Elija Create rule.
4. Escriba un nombre y una descripción de la regla. Por ejemplo, llame a la regla `ARTestRule`
5. En Event bus (Bus de eventos), elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
6. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).
7. Elija Next (Siguiente).
8. En Event source (Origen del evento), elija Other (Otro).
9. En Patrón de evento, introduzca lo siguiente:

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

10. Elija Next (Siguiente).
11. En Target types (Tipos de destino), elija AWS service.
12. En Seleccionar un destino, elija Tema de SNS en la lista desplegable.
13. En Tema, seleccione el tema de Amazon SNS que creó en el paso 1. Para este tutorial, elija eventbridge-IT-test.
14. En Configuración adicional, haga lo siguiente:

    a. En Configurar entrada de destino, seleccione Transformador de entrada en la lista desplegable.

    b. Elija Configurar transformador de entrada)

    c. En Eventos de muestra, introduzca lo siguiente:

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

    d. En Transformador de entrada de destino, haga lo siguiente:

        i. En Ruta de entrada, introduzca lo siguiente:

```
{"detail-type":"$.detail-type"}
```

ii. En Plantilla de entrada, introduzca lo siguiente:

```
"This event was of <detail-type> type."
```

e. Elija Confirmar.

15. Elija Next (Siguiente).
16. Elija Next (Siguiente).
17. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 4: Enviar eventos de prueba

Ahora que ha configurado el tema de SNS y la regla, enviaremos los eventos de prueba para asegurarnos de que la regla funciona correctamente.

Para enviar eventos de prueba (consola)

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, elija Event Buses (Buses de eventos).
3. En el icono del bus de eventos predeterminado, elija Acciones, Enviar eventos.
4. Introduzca un origen de eventos. Por ejemplo, `TestEvent`.
5. En Detail type (Tipo de detalle), introduzca `customerCreated`.
6. En Detalle del evento, introduzca `{}`.
7. Seleccione Send (enviar).

Paso 5: Confirmar el éxito

Si recibe un correo electrónico de notificaciones de AWS que coincide con el resultado esperado, ha completado correctamente el tutorial.

Paso 6: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar el tema de SNS

1. Abra la página Temas de la consola de SNS.
2. Seleccione el tema que creó.
3. Elija Eliminar (Delete).
4. Escriba **`delete me`**.
5. Elija Eliminar (Delete).

Para eliminar la suscripción a SNS

1. Abra la página Suscripciones en la consola de SNS.
2. Seleccione la suscripción que creó.
3. Elija Eliminar (Delete).
4. Elija Eliminar (Delete).

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.
3. Elija Eliminar (Delete).
4. Elija Eliminar (Delete).

Tutoriales de Amazon EventBridge para la integración con otros servicios de AWS

Amazon EventBridge trabaja con otros servicios de AWS para procesar eventos o invocar un recurso de AWS como destino de una regla. En los siguientes tutoriales, se explica cómo integrar EventBridge con otros servicios de AWS.

Tutoriales:

- Tutorial: Registro del estado de un grupo de Auto Scaling con EventBridge
- Tutorial: Registra las llamadas a la AWS API mediante EventBridge
- Tutorial: Registre el estado de una instancia de Amazon EC2 mediante EventBridge
- Tutorial: Registro de operaciones en el nivel de objetos de Amazon S3 con EventBridge
- Tutorial: envíe eventos a una transmisión de Amazon Kinesis utilizando EventBridge y el esquema aws.events
- Tutorial: Programación de instantáneas de Amazon EBS automatizadas utilizando EventBridge
- Tutorial: Envío de una notificación cuando se crea un objeto de Amazon S3
- Tutorial: Programación de funciones de AWS Lambda con EventBridge

Tutorial: Registro del estado de un grupo de Auto Scaling con EventBridge

Puede ejecutar una función de AWS Lambda que registre un evento siempre que un grupo de Auto Scaling lance o termine una instancia de Amazon EC2 que indique si el evento se realizó correctamente.

Para obtener información sobre escenarios adicionales en los que se utilizan eventos de Amazon EC2 Auto Scaling, consulte Uso de EventBridge para gestionar eventos de Auto Scaling en la Guía del usuario de Amazon EC2 Auto Scaling.

En este tutorial, se crea una función de Lambda y una regla en la consola de EventBridge que llama a esa función cuando un grupo de Amazon EC2 Auto Scaling lanza o termina una instancia.

Pasos:

- Requisitos previos
- Paso 1: Crear una función Lambda
- Paso 2: Crear una regla
- Paso 3: Probar la regla
- Paso 4: Confirmar el éxito
- Paso 5: Eliminar los recursos

Requisitos previos

Para completar este tutorial necesitará los siguientes recursos:

- Un grupo de Auto Scaling. Para obtener más información, consulte Creación de un grupo de Auto Scaling mediante una configuración de lanzamiento en la Guía del usuario de Amazon EC2 Auto Scaling.

Paso 1: Crear una función Lambda

Cree una función Lambda para registrar los eventos de escalado ascendente y descendente para su grupo de Auto Scaling.

Para crear una función Lambda

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.

2. Elija Create function (Crear función).
3. Elija Author from scratch (Crear desde cero).
4. Escriba el nombre de la función de Lambda.. Por ejemplo, asigne un nombre a la función `LogAutoScalingEvent`.
5. Deje el resto de las opciones como predeterminadas y elija Crear función.
6. En la pestaña Código de la página de funciones, haga doble clic en index.js.
7. Sustituya el código existente por el código siguiente.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogAutoScalingEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Elija Deploy (Implementar).

Paso 2: Crear una regla

Cree una regla para ejecutar la función de Lambda que creó en el paso 1. La regla se ejecuta cuando el grupo de Auto Scaling inicia o detiene una instancia.

Para crear una regla

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.
3. Elija Create rule.
4. Escriba un nombre y una descripción de la regla. Por ejemplo, llame a la regla `TestRule`
5. En Event bus (Bus de eventos), elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
6. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).
7. Elija Next (Siguiente).
8. En Event source (Origen del evento), elija AWS services (Servicios de).

9. En Event pattern (Patrón de evento), realice una de las siguientes acciones:
    a. En Origen del evento, seleccione Auto Scaling en la lista desplegable.
    b. En Tipo de evento, seleccione Lanzamiento y finalización de la instancia en la lista desplegable.
    c. Elija Cualquier evento de instancia y Cualquier nombre de grupo.
10. Elija Next (Siguiente).
11. En Target types (Tipos de destino), elija AWS service.
12. En Seleccionar un destino, elija Función de Lambda en la lista desplegable.
13. En Función, seleccione la función de Lambda que creó en la sección Paso 1: Crear una función de Lambda. En este ejemplo, seleccione `LogAutoScalingEvent`.
14. Elija Next (Siguiente).
15. Elija Next (Siguiente).
16. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 3: Probar la regla

Puede probar la regla manualmente escalando un grupo de Auto Scaling para que lance una instancia. Espere unos minutos a que se produzca el evento de escalado ascendente y, a continuación, verifique que la función de Lambda se ha invocado.

Para probar la regla con un grupo de Auto Scaling

1. Para aumentar el tamaño de su grupo de Auto Scaling, haga lo siguiente:
    a. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
    b. En el panel de navegación, seleccione Auto Scaling, Auto Scaling Groups (Grupos de Auto Scaling).
    c. Seleccione la casilla del grupo de Auto Scaling correspondiente.
    d. En la pestaña Details, seleccione Edit. En Desired, aumente la capacidad deseada en 1. Por ejemplo, si el valor actual es 2, introduzca 3. La capacidad deseada debe ser menor o igual que el tamaño máximo del grupo. Si el nuevo valor de Desired es mayor que Max, debe actualizar Max. Cuando haya terminado, elija Save.
2. Para ver la salida de la función de Lambda, haga lo siguiente:

a. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
b. En el panel de navegación, elija Logs.
c. Seleccione el nombre del grupo de registros para la función Lambda (`/aws/lambda/function-name`).
d. Seleccione el nombre del flujo de registro para ver los datos proporcionados por la función para la instancia que ha lanzado.

3. (Opcional) Cuando haya terminado, puede reducir la capacidad deseada en uno para que el grupo de Auto Scaling vuelva a su tamaño anterior.

Paso 4: Confirmar el éxito

Si ve el evento de Lambda en los registros de CloudWatch, significa que ha completado correctamente este tutorial. Si el evento no está en sus registros de CloudWatch, comience a solucionar problemas verificando que la regla se haya creado correctamente y, si la regla parece correcta, compruebe que el código de la función de Lambda sea correcto.

Paso 5: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Para eliminar las funciones de Lambda

1. Abra la página de Functions (Funciones) en la consola de Lambda.
2. Seleccione las funciones que creó.
3. Elija Actions (Acciones), Delete (Eliminar).
4. Elija Eliminar.

Tutorial: Registra las llamadas a la AWS API mediante EventBridge

Puedes usar EventBridge las reglas de Amazon para reaccionar a las llamadas a la API realizadas por un AWS servicio registrado por AWS CloudTrail.

En este tutorial, creará una AWS CloudTrailruta, una función Lambda y una regla en la EventBridge consola. La regla invoca la función de Lambda cuando se detiene una instancia de Amazon EC2.

Pasos:

- Paso 1: Crear una ruta AWS CloudTrail
- Paso 2: Crear una función de AWS Lambda
- Paso 3: Crear una regla
- Paso 4: Probar la regla
- Paso 5: Confirmar el éxito
- Paso 6: Eliminar los recursos

Paso 1: Crear una ruta AWS CloudTrail

Si ya tiene configurado un registro de seguimiento, vaya al paso 2.

Para crear un registro de seguimiento

1. Abre la CloudTrail consola en https://console.aws.amazon.com/cloudtrail/.
2. Elija Trails (Registros de seguimiento), Create trail (Crear un registro de seguimiento).
3. En Trail name, escriba un nombre para el registro de seguimiento.
4. En Ubicación de almacenamiento, en Crear un bucket de S3 nuevo.
5. En alias de AWS KMS , escriba el alias para la clave KMS.
6. Elija Siguiente.
7. Elija Siguiente.
8. Elija Create Trail (Crear registro de seguimiento).

Paso 2: Crear una función de AWS Lambda

Cree una función de Lambda para registrar los eventos de llamada a la API.

Cómo crear una función de Lambda

1. Abra la AWS Lambda consola en https://console.aws.amazon.com/lambda/.
2. Elija Crear función.
3. Elija Crear desde cero.
4. Introduzca un nombre y la descripción de la función de Lambda. Por ejemplo, asigne un nombre a la función `LogEC2StopInstance`.
5. Deje el resto de las opciones como predeterminadas y elija Crear función.
6. En la pestaña Código de la página de funciones, haga doble clic en index.js.
7. Sustituya el código existente por el código siguiente.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2StopInstance');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Elija Implementar.

Paso 3: Crear una regla

Cree una regla para ejecutar su función de Lambda que creó en el paso 2 siempre que detenga una instancia de Amazon EC2.

Para crear una regla

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Elija Crear regla.
4. Escriba un nombre y una descripción para la regla. Por ejemplo, llame a la regla `TestRule`
5. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un servicio de AWS en la cuenta emite un evento, siempre va al bus de eventos predeterminado de la cuenta.
6. En Tipo de regla, seleccione Regla con un patrón de eventos.

7. Seleccione Siguiente.
8. En Origen de evento, seleccione Servicios de AWS .
9. En Event pattern (Patrón de evento), realice una de las siguientes acciones:
    a. En Origen del evento, seleccione EC2 en la lista desplegable.
    b. Para el tipo de evento, selecciona AWS API Call via CloudTrail en la lista desplegable.
    c. Elija Operaciones específicas y escriba `StopInstances`.
10. Seleccione Siguiente.
11. En Tipos de destino, seleccione Servicio de AWS .
12. En Seleccionar un destino, elija Función de Lambda en la lista desplegable.
13. En Función, seleccione la función de Lambda que creó en la sección Paso 1: Crear una función de Lambda. En este ejemplo, seleccione `LogEC2StopInstance`.
14. Elija Siguiente.
15. Seleccione Siguiente.
16. Revise los detalles de la regla y seleccione Crear regla.

Paso 4: Probar la regla

Puede probar la regla parando una instancia de Amazon EC2 mediante la consola de Amazon EC2. Espera unos minutos a que se detenga la instancia y, a continuación, comprueba tus AWS Lambda métricas en la CloudWatch consola para comprobar que la función se ha ejecutado.

Para probar la regla parando una instancia

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Lance una instancia. Para obtener más información, consulte Lance Your Instance en la Guía del usuario de Amazon EC2.
3. Detenga la instancia. Para obtener más información, consulte Stop and Start Your Instance en la Guía del usuario de Amazon EC2.
4. Para ver la salida de la función de Lambda, haga lo siguiente:
    a. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
    b. En el panel de navegación, elija Logs (Registros).
    c. Seleccione el nombre del grupo de registros para la función de Lambda (`/aws/lambda/`*`function-name`*).

d. Seleccione el nombre del flujo de registro para ver los datos proporcionados por la función para la instancia que ha detenido.

5. (Opcional) Cuando haya finalizado, termine la instancia detenida. Para obtener más información, consulte Finalizar su instancia en la Guía del usuario de Amazon EC2.

Paso 5: Confirmar el éxito

Si ve el evento Lambda en los CloudWatch registros, significa que ha completado correctamente este tutorial. Si el evento no está en sus CloudWatch registros, comience a solucionar problemas verificando que la regla se haya creado correctamente y, si la regla parece correcta, compruebe que el código de la función Lambda sea correcto.

Paso 6: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Al eliminar AWS los recursos que ya no utilizas, evitas que se hagan cargos innecesarios a tu AWS cuenta.

Para eliminar la (s) EventBridge regla (s)

1. Abre la página de reglas de la EventBridge consola.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Para eliminar las funciones de Lambda

1. Abra la página de Funciones en la consola de Lambda.
2. Seleccione las funciones que creó.
3. Elija Actions (Acciones), Delete (Eliminar).
4. Elija Eliminar.

Para eliminar la (s) CloudTrail ruta (s)

1. Abre la página de senderos de la CloudTrail consola.
2. Seleccione los registros de seguimiento que creó.

3. Elija Eliminar.
4. Elija Eliminar.

Tutorial: Registre el estado de una instancia de Amazon EC2 mediante EventBridge

Puede crear una función de AWS Lambda que registre un cambio de estado de una instancia de Amazon EC2. Tiene la opción de crear una regla que ejecute la función de Lambda cuando haya una transición de estado o una transición a uno o varios estados de interés. En este tutorial, puede registrar el lanzamiento de una nueva instancia.

Pasos:

- Paso 1: Crear una función de AWS Lambda
- Paso 2: Crear una regla
- Paso 3: Probar la regla
- Paso 4: Confirmar el éxito
- Paso 5: Eliminar los recursos

Paso 1: Crear una función de AWS Lambda

Cree una función de Lambda para registrar los eventos de cambio de estado. Cuando cree su reglan en el Paso 2, especifique esta función.

Cómo crear una función de Lambda

1. Abra la AWS Lambda consola en https://console.aws.amazon.com/lambda/.
2. Elija Crear función.
3. Elija Crear desde cero.
4. Introduzca un nombre y la descripción de la función de Lambda. Por ejemplo, asigne un nombre a la función `LogEC2InstanceStateChange`.
5. Deje el resto de las opciones como predeterminadas y elija Crear función.
6. En la pestaña Código de la página de funciones, haga doble clic en index.js.
7. Sustituya el código existente por el código siguiente.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2InstanceStateChange');
    console.log('Received event:', JSON.stringify(event, null, 2));
```

```
    callback(null, 'Finished');
};
```

8. Elija Deploy (Implementar).

Paso 2: Crear una regla

Cree una regla para ejecutar la función de Lambda que creó en el paso 1. La regla se ejecuta al lanzar una instancia de Amazon EC2.

Para crear la EventBridge regla

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Elija Crear regla.
4. Escriba un nombre y una descripción para la regla. Por ejemplo, llame a la regla `TestRule`
5. En Bus de eventos, elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un servicio de AWS en la cuenta emite un evento, siempre va al bus de eventos predeterminado de la cuenta.
6. En Tipo de regla, seleccione Regla con un patrón de eventos.
7. Seleccione Siguiente.
8. En Origen de evento, seleccione Servicios de AWS .
9. En Event pattern (Patrón de evento), realice una de las siguientes acciones:
    a. En Origen del evento, seleccione EC2 en la lista desplegable.
    b. En Tipo de evento, elija Notificación de cambio de estado de instancia de EC2 en la lista desplegable.
    c. Elija Estados específicos y elija Ejecutar de la lista desplegable.
    d. Elija Cualquier instancia
10. Seleccione Siguiente.
11. En Tipos de destino, seleccione Servicio de AWS .
12. En Seleccionar un destino, elija Función de Lambda en la lista desplegable.
13. En Función, seleccione la función de Lambda que creó en la sección Paso 1: Crear una función de Lambda. En este ejemplo, seleccione `LogEC2InstanceStateChange`.

14. Elija Siguiente.
15. Seleccione Siguiente.
16. Revise los detalles de la regla y seleccione Crear regla.

Paso 3: Probar la regla

Puede probar la regla parando una instancia de Amazon EC2 mediante la consola de Amazon EC2. Espera unos minutos a que se detenga la instancia y, a continuación, comprueba tus AWS Lambda métricas en la CloudWatch consola para comprobar que la función se ha ejecutado.

Para probar la regla parando una instancia

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Lance una instancia. Para obtener más información, consulte Lance Your Instance en la Guía del usuario de Amazon EC2.
3. Detenga la instancia. Para obtener más información, consulte Stop and Start Your Instance en la Guía del usuario de Amazon EC2.
4. Para ver la salida de la función de Lambda, haga lo siguiente:
   a. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
   b. En el panel de navegación, elija Logs (Registros).
   c. Seleccione el nombre del grupo de registros para la función de Lambda (`/aws/lambda/function-name`).
   d. Seleccione el nombre del flujo de registro para ver los datos proporcionados por la función para la instancia que ha detenido.
5. (Opcional) Cuando haya finalizado, termine la instancia detenida. Para obtener más información, consulte Finalizar su instancia en la Guía del usuario de Amazon EC2.

Paso 4: Confirmar el éxito

Si ve el evento Lambda en los CloudWatch registros, significa que ha completado correctamente este tutorial. Si el evento no está en sus CloudWatch registros, comience a solucionar problemas verificando que la regla se haya creado correctamente y, si la regla parece correcta, compruebe que el código de la función Lambda sea correcto.

Paso 5: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Al eliminar AWS los recursos que ya no utilizas, evitas que se hagan cargos innecesarios a tu AWS cuenta.

Para eliminar la (s) EventBridge regla (s)

1. Abre la página de reglas de la EventBridge consola.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Para eliminar las funciones de Lambda

1. Abra la página de Funciones en la consola de Lambda.
2. Seleccione las funciones que creó.
3. Elija Actions (Acciones), Delete (Eliminar).
4. Elija Eliminar.

Tutorial: Registro de operaciones en el nivel de objetos de Amazon S3 con EventBridge

Puede registrar las operaciones de API en el nivel de objetos que tienen lugar en los buckets de Amazon S3. Antes de que Amazon EventBridge pueda asignar estos eventos, debe utilizar AWS CloudTrail para configurar un registro de seguimiento para recibir estos eventos.

En este tutorial, creará un registro de seguimiento de CloudTrail, creará una función de AWS Lambda y, a continuación, creará una regla en la consola de EventBridge que invoque esa función en respuesta a un evento de datos de S3.

Pasos:

- Paso 1: configuración del registro de seguimiento de AWS CloudTrail
- Paso 2: Crear una función de AWS Lambda
- Paso 3: Crear una regla
- Paso 4: Probar la regla de
- Paso 5: Confirmar el éxito
- Paso 6: Eliminar los recursos

Paso 1: configuración del registro de seguimiento de AWS CloudTrail

Para registrar eventos de datos para un bucket de S3 en AWS CloudTrail y EventBridge, primero cree un registro de seguimiento. Un registro de seguimiento captura las llamadas a la API y los eventos relacionados de la cuenta y entrega los archivos de registro en un bucket de S3 especificado. Puede actualizar un registro de seguimiento existente o crear uno.

Para obtener más información, consulte Eventos de datos en la Guía del usuario de AWS CloudTrail.

Para crear un registro de seguimiento

1. Abra la consola de CloudTrail en https://console.aws.amazon.com/cloudtrail/.
2. Elija Trails (Registros de seguimiento), Create trail (Crear un registro de seguimiento).
3. En Trail name, escriba un nombre para el registro de seguimiento.
4. En Ubicación de almacenamiento, en Crear un bucket de S3 nuevo.
5. En alias de AWS KMS, escriba el alias para la clave KMS.
6. Elija Next (Siguiente).

7. En Tipo de evento, elija Eventos de datos
8. En Eventos de datos, realice una de las siguientes operaciones:
   - Para registrar eventos de datos de todos los objetos de Amazon S3 en un bucket, especifique un bucket de S3 y un prefijo vacío. Cuando un evento se produce en un objeto de dicho bucket de , el registro de seguimiento procesa y registra el evento.
   - Para registrar los eventos de datos para objetos de Amazon S3 concretos, especifique un bucket de S3 y el prefijo del objeto. Cuando un evento se produce en un objeto en dicho bucket de y el objeto comienza por el prefijo indicado, el registro de seguimiento procesa y registra el evento.
9. En cada recurso, elija si desea registrar los eventos de tipo Lectura, Escritura o ambos.
10. Elija Next (Siguiente).
11. Elija Create Trail (Crear registro de seguimiento).

Paso 2: Crear una función de AWS Lambda

Cree una función Lambda para registrar eventos de datos para sus buckets de S3.

Para crear una función Lambda

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. Elija Create function (Crear función).
3. Elija Author from scratch (Crear desde cero).
4. Introduzca un nombre y la descripción de la función Lambda. Por ejemplo, asigne un nombre a la función `LogS3DataEvents`.
5. Deje el resto de las opciones como predeterminadas y elija Crear función.
6. En la pestaña Código de la página de funciones, haga doble clic en index.js.
7. Sustituya el código existente por el código siguiente.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogS3DataEvents');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Elija Deploy (Implementar).

Paso 3: Crear una regla

Cree una regla para ejecutar la función de Lambda que creó en el paso 2. Esta regla se ejecuta en respuesta a un evento de datos de Amazon S3.

Para crear una regla

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.
3. Elija Create rule.
4. Escriba un nombre y una descripción de la regla. Por ejemplo, llame a la regla `TestRule`
5. En Event bus (Bus de eventos), elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione predeterminado. Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
6. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).
7. Elija Next (Siguiente).
8. En Event source (Origen del evento), elija AWS services (Servicios de).
9. En Event pattern (Patrón de evento), realice una de las siguientes acciones:
    a. En Origen del evento, seleccione Simple Storage Service (S3) en la lista desplegable.
    b. En Tipo de evento, seleccione Llamada a la API en el nivel de objetos de mediante CloudTrail en la lista desplegable.
    c. Elija Specific operation(s) (Operaciones específicas) y, a continuación, elija PutObject.
    d. De forma predeterminada, la regla coincide con los eventos de datos de todos los buckets de la región. Para asignar eventos de datos a buckets específicos, elija Specify bucket(s) by name y especifique uno o varios buckets.
10. Elija Next (Siguiente).
11. En Target types (Tipos de destino), elija AWS service.
12. En Seleccionar un destino, elija Función de Lambda en la lista desplegable.
13. En Función, seleccione la función de Lambda de `LogS3DataEvents` que creó en el paso 1.
14. Elija Next (Siguiente).

15. Elija Next (Siguiente).
16. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 4: Probar la regla de

Para probar la regla, coloque un objeto en su bucket de S3. Puede verificar que se invocó su función de Lambda.

Para ver los registros de su función de Lambda

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs.
3. Seleccione el nombre del grupo de registros para la función Lambda (`/aws/lambda/`*`function-name`*).
4. Seleccione el nombre del flujo de registro para ver los datos proporcionados por la función para la instancia que ha lanzado.

También puede comprobar sus registros de CloudTrail en el bucket de S3 especificado para su registro de seguimiento. Para obtener más información, consulte Obtención y visualización de los archivos de registro de CloudTrail en la Guía del usuario de AWS CloudTrail.

Paso 5: Confirmar el éxito

Si ve el evento de Lambda en los registros de CloudWatch, significa que ha completado correctamente este tutorial. Si el evento no está en sus registros de CloudWatch, comience a solucionar problemas verificando que la regla se haya creado correctamente y, si la regla parece correcta, compruebe que el código de la función de Lambda sea correcto.

Paso 6: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.

3. Elija Eliminar.
4. Elija Eliminar.

Para eliminar las funciones de Lambda

1. Abra la página de Functions (Funciones) en la consola de Lambda.
2. Seleccione las funciones que creó.
3. Elija Actions (Acciones), Delete (Eliminar).
4. Elija Eliminar.

Para eliminar los registros de seguimiento de CloudTrail

1. Abra la página Trails (Registros de seguimiento) de la consola de CloudTrail.
2. Seleccione los registros de seguimiento que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Tutorial: envíe eventos a una transmisión de Amazon Kinesis utilizando EventBridge y el esquema **`aws.events`**

Puede enviar eventos de llamadas a la AWS API EventBridge a una transmisión de Amazon Kinesis, crear aplicaciones de Kinesis Data Streams y procesar grandes cantidades de datos. En este tutorial, creará una transmisión de Kinesis y, a continuación, creará una regla en la EventBridge consola que envíe eventos a esa transmisión cuando se detenga una instancia de Amazon EC2.

Pasos:

- Requisitos previos
- Paso 1: Crear un fljujo de Amazon Kinesis
- Paso 2: Crear una regla
- Paso 3: Probar la regla
- Paso 4: Verificar el envío del evento
- Paso 5: Eliminar los recursos

Requisitos previos

En este tutorial, utilizará lo siguiente:

- Úselo AWS CLI para trabajar con las transmisiones de Kinesis.

  Para instalar la AWS CLI, consulte Instalación, actualización y desinstalación de la AWS CLI versión 2.

> **Note**
>
> En este tutorial se utilizan AWS los eventos y el registro de `aws.events` esquemas integrado. También puede crear una EventBridge regla basada en el esquema de sus eventos personalizados agregándolos manualmente a un registro de esquemas personalizados o mediante la detección de esquemas.
> Para obtener más información sobre los esquemas, consulte ???. Para obtener más información sobre cómo crear una regla con otras opciones de patrones de eventos, consulte ???.

Paso 1: Crear un fljujo de Amazon Kinesis

Para crear una transmisión, utilice el comando en una línea de `create-stream` AWS CLI comandos.

```
aws kinesis create-stream --stream-name test --shard-count 1
```

Cuando el estado del flujo sea `ACTIVE`, el flujo está listo. Para comprobar el estado del flujo, utilice el comando `describe-stream`.

```
aws kinesis describe-stream --stream-name test
```

Paso 2: Crear una regla

Cree una regla para enviar eventos al flujo cuando se pare una instancia de Amazon EC2.

Para crear una regla

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Elija Crear regla.
4. Escriba un nombre y una descripción para la regla. Por ejemplo, llame a la regla `TestRule`
5. En Bus de eventos, seleccione el valor predeterminado.
6. En Tipo de regla, elija Regla con un patrón de evento.
7. Elija Siguiente.
8. En Fuente del evento, selecciona AWS eventos o eventos EventBridge asociados.
9. En Método de creación, elija Usar esquema.
10. En Event pattern (Patrón de evento), realice una de las siguientes acciones:

    a. En Tipo de esquema, elija Seleccionar esquema del registro de esquemas.

    b. En Registro de esquemas, elija aws.events en la lista desplegable.

    c. En Schema, elija aws.ec2 @EC2 en la InstanceStateChangeNotification lista desplegable.

    EventBridge muestra el esquema de eventos en Modelos.

    EventBridge muestra un asterisco rojo junto a las propiedades necesarias para el evento, no para el patrón del evento.

d. En Modelos, defina las siguientes propiedades del filtro de eventos:

   i. Seleccione + Editar junto a la propiedad state.

      Deje el campo Relación vacío. En Valor, introduzca `running`. Elija Establecer.

   ii. Seleccione + Editar junto a la propiedad source.

      Deje el campo Relación vacío. En Valor, introduzca `aws.ec2`. Elija Establecer.

   iii. Seleccione + Editar junto a la propiedad detail-type.

      Deje el campo Relación vacío. En Valor, introduzca `EC2 Instance State-change Notification`. Elija Establecer.

e. Para ver el patrón de eventos que ha creado, elija Generar patrón de eventos en JSON

   EventBridge muestra el patrón de eventos en JSON:

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

11. Seleccione Siguiente.
12. En Tipos de destino, seleccione Servicio de AWS .
13. En Seleccionar un destino, elija Flujo de Kinesis en la lista desplegable.
14. En Flujo, seleccione el flujo de Kinesis que creó en la sección Paso 1: Crear un flujo de Amazon Kinesis. En este ejemplo, seleccione `test`.
15. En Rol de ejecución, elija Crear un rol nuevo para este recurso específico.
16. Elija Siguiente.
17. Seleccione Siguiente.
18. Revise los detalles de la regla y seleccione Crear regla.

Paso 3: Probar la regla

Para probar la regla, pare una instancia de Amazon EC2. Espera unos minutos a que la instancia se detenga y, a continuación, comprueba tus CloudWatch métricas para comprobar que la función se ha ejecutado.

Para probar la regla parando una instancia

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Lance una instancia. Para obtener más información, consulte Lance Your Instance en la Guía del usuario de Amazon EC2.
3. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
4. En el panel de navegación, seleccione Reglas.

   Elija el nombre de la regla que ha creado y elija Metrics for the rule (Métricas para la regla).
5. (Opcional) Cuando haya finalizado, termine la instancia. Para obtener más información, consulte Finalizar su instancia en la Guía del usuario de Amazon EC2.

Paso 4: Verificar el envío del evento

Puede utilizar el AWS CLI para obtener el registro de la transmisión y comprobar que el evento se ha enviado.

Para obtener el registro

1. Para empezar a leer el flujo de Kinesis, en el símbolo del sistema, utilice el comando `get-shard-iterator`.

   ```
   aws kinesis get-shard-iterator --shard-id shardId-000000000000 --shard-iterator-type TRIM_HORIZON --stream-name test
   ```

   A continuación, se muestra un ejemplo del resultado.

   ```
   {
       "ShardIterator": "AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp
   +KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd
   +efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg="
   }
   ```

2. Para obtener el registro, utilice el siguiente comando `get-records`. Use el iterador de partición del resultado del paso anterior.

```
aws kinesis get-records --shard-
iterator AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp
+KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd
+efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg=
```

   Si el comando se realiza correctamente, solicita registros del flujo para el fragmento especificado. Puede recibir cero o más registros. Los registros devueltos podrían no representar todos los registros del flujo. Si no recibe los datos que espera, siga llamando a `get-records`.

3. Los registros de Kinesis están codificados en Base64. Use un decodificador Base64 para decodificar los datos y comprobar que se trata del evento que se envió al flujo en formato JSON.

Paso 5: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Al eliminar AWS los recursos que ya no utilizas, evitas cargos innecesarios en tu AWS cuenta.

Para eliminar la (s) EventBridge regla (s)

1. Abre la página de reglas de la EventBridge consola.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Para eliminar los flujos de Kinesis

1. Abra la página Flujos de datos de la consola de Kinesis.
2. Seleccione los flujos que creó.
3. Elija Acciones, Eliminar.
4. Escriba eliminar en el campo y seleccione Eliminar.

Tutorial: Programación de instantáneas de Amazon EBS automatizadas utilizando EventBridge

Puedes ejecutar las reglas de EventBridge conforme a una programación. En este tutorial, creará una instantánea automatizada de un volumen de Amazon Elastic Block Store (Amazon EBS) existente en una programación. Puede seleccionar una frecuencia fija para que se cree una instantánea cada pocos minutos o utilizar una expresión cron para crear la instantánea a una hora concreta del día.

> **Important**
>
> Para crear reglas con destinos integrados, debe usar la AWS Management Console.

Pasos:

- Paso 1: crear la tabla
- Paso 2: Probar la regla
- Paso 3: Confirmar el éxito
- Paso 4: Eliminar los recursos

Paso 1: crear la tabla

Cree una regla que realice instantáneas de manera programada. Puede utilizar una expresión de frecuencia o una expresión Cron para especificar la programación. Para obtener más información, consulte Crear una regla de Amazon EventBridge que se ejecuta según una programación.

Para crear una regla de

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.
3. Elija Create rule.
4. Escriba un nombre y una descripción de la regla.

   Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.

5. En Event bus (Bus de eventos), elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de la cuenta, seleccione Bus de eventos predeterminado de AWS. Cuando un servicio de AWS en la cuenta emite un evento, siempre va al bus de eventos predeterminado de la cuenta.
6. En Rule type (Tipo de regla), elija Schedule (Programación).
7. Elija Next (Siguiente).
8. En Patrón de programación, elija Una programación que se ejecute con una frecuencia regular, por ejemplo, cada 10 minutos. e introducir **5** y elegir Minutos en la lista desplegable.
9. Elija Next (Siguiente).
10. En Target types (Tipos de destino), elija AWS service.
11. En Seleccionar un destino, elija EBS Create Snapshot en la lista desplegable.
12. En ID de volumen, introduzca el ID de volumen del volumen de Amazon EBS.
13. En Rol de ejecución, elija Crear un rol nuevo para este recurso específico.
14. Elija Next (Siguiente).
15. Elija Next (Siguiente).
16. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 2: Probar la regla

Puede verificar que la regla funciona consultando la primera instantánea después de tomarla.

Para probar la regla

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En el panel de navegación, elija Elastic Block Store, Snapshots (Instantáneas).
3. Compruebe que la primera instantánea aparezca en la lista.

Paso 3: Confirmar el éxito

Si ve una instantánea en la lista, significa que ha completado correctamente este tutorial. Si la instantánea no está en la lista, comience a solucionar el problema comprobando que la regla se creó correctamente.

Paso 4: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en la cuenta de AWS.

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Tutorial: Envío de una notificación cuando se crea un objeto de Amazon S3

Puede enviar notificaciones por correo electrónico cuando se creen objetos de Amazon Simple Storage Service (Amazon S3) mediante Amazon EventBridge y Amazon SNS. En este tutorial, creará un tema y una suscripción de SNS. A continuación, creará una regla en la consola de EventBridge que envíe los eventos a ese tema cuando se reciban eventos `Object Created` de Amazon S3.

Pasos:

- Requisitos previos
- Paso 1: Crear un tema de Amazon SNS
- Paso 2: Crear una suscripción de Amazon SNS
- Paso 3: Crear una regla
- Paso 4: Probar la regla
- Paso 5: Eliminar los recursos

Requisitos previos

Para recibir eventos de Amazon S3 en EventBridge, debe habilitar EventBridge en la consola de Amazon S3. En este tutorial se supone que EventBridge está activado. Para obtener más información, consulte Habilitación de Amazon EventBridge en la consola de S3.

Paso 1: Crear un tema de Amazon SNS

Cree un tema para recibir los eventos de EventBridge.

Para crear un tema

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. En el panel de navegación, elija Temas.
3. Elija Create new topic (Crear nuevo tema).
4. En Type (Tipo), seleccione Standard (Estándar).
5. Escriba **`eventbridge-test`** como nombre del tema.
6. Elija Create new topic (Crear nuevo tema).

Paso 2: Crear una suscripción de Amazon SNS

Cree una suscripción para recibir notificaciones por correo electrónico de Amazon S3 cuando el tema reciba eventos.

Para crear una suscripción

1. Abra la consola de Amazon SNS en https://console.aws.amazon.com/sns/v3/home.
2. En el panel de navegación, seleccione Subscriptions.
3. Seleccione Create subscription.
4. En la lista ARN de tema, seleccione el tema que creó en el paso 1. Para este tutorial, elija eventbridge-test.
5. En Protocol (Protocolo), elija Email (Correo electrónico).
6. En Punto de enlace, introduzca su dirección de correo electrónico.
7. Seleccione Create subscription.
8. Confirme la suscripción seleccionando Confirmar suscripción en el correo electrónico que reciba de las notificaciones de AWS.

Paso 3: Crear una regla

Cree una regla para enviar eventos a su tema cuando se cree un objeto de Amazon S3.

Para crear una regla de

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.
3. Elija Create rule.
4. Escriba un nombre y una descripción de la regla. Por ejemplo, llame a la regla `s3-test`
5. En Bus de eventos, seleccione el valor predeterminado.
6. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).
7. Elija Next (Siguiente).
8. En Event source (Origen del evento), elija AWS events or EventBridge partner events (Eventos o eventos de socios de EventBridge).
9. En Método de creación, seleccione Formulario de patrón de eventos.

10. En Event pattern (Patrón de evento), realice una de las siguientes acciones:
    a. En Origen del evento, seleccione Servicios de AWS en la lista desplegable.
    b. En Servicio de AWS, seleccione Simple Storage Service (S3) en la lista desplegable.
    c. En Tipo de evento, elija Notificación de evento de Amazon S3 en la lista desplegable.
    d. Seleccione Eventos específicos y elija Objeto creado en la lista desplegable.
    e. Elija Agregar bucket
11. Elija Next (Siguiente).
12. En Target types (Tipos de destino), elija AWS service.
13. En Seleccionar un destino, elija Tema de SNS en la lista desplegable.
14. En Tema, seleccione el tema de Amazon SNS que creó en la sección Paso 1: Crear un tema de SNS. En este ejemplo, seleccione `eventbridge-test`.
15. Elija Next (Siguiente).
16. Elija Next (Siguiente).
17. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 4: Probar la regla

Para probar su regla, cree un objeto de Amazon S3 cargando un archivo en un bucket habilitado para Eventbridge. A continuación, espere unos minutos y compruebe si ha recibido un correo de las notificaciones de AWS.

Paso 5: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar el tema de SNS

1. Abra la página Temas de la consola de SNS.
2. Seleccione el tema que creó.
3. Elija Eliminar (Delete).
4. Escriba **`delete me`**.
5. Elija Eliminar (Delete).

Para eliminar la suscripción a SNS

1. Abra la página Suscripciones en la consola de SNS.
2. Seleccione la suscripción que creó.
3. Elija Eliminar (Delete).
4. Elija Eliminar (Delete).

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.
3. Elija Eliminar (Delete).
4. Elija Eliminar (Delete).

Tutorial: Programación de funciones de AWS Lambda con EventBridge

Puede configurar una regla para ejecutar una función de AWS Lambda conforme a una programación. Este tutorial muestra cómo utilizar la AWS Management Console o la AWS CLI para crear la regla. Si desea utilizar la AWS CLI pero no la ha instalado, consulte Instalación, actualización y desinstalación de la versión 2 de la AWS CLI.

Para programaciones, EventBridge no proporciona precisión de segundo nivel en expresiones de programación. La mejor resolución al utilizar una expresión cron es 1 minuto. Debido a la naturaleza distribuida de EventBridge y los servicios de destino, puede producirse un retraso de varios segundos entre el momento en que la regla programada se activa y el momento en que el servicio de destino ejecuta el recurso de destino.

Pasos:

- Paso 1: Crear una función Lambda
- Paso 2: Crear una regla
- Paso 3: Comprobar la regla
- Paso 4: Confirmar el éxito
- Paso 5: Eliminar los recursos

Paso 1: Crear una función Lambda

Cree una función Lambda para registrar los eventos programados.

Para crear una función Lambda

1. Abra la consola de AWS Lambda en https://console.aws.amazon.com/lambda/.
2. Elija Create function (Crear función).
3. Elija Author from scratch (Crear desde cero).
4. Introduzca un nombre y la descripción de la función Lambda. Por ejemplo, asigne un nombre a la función `LogScheduledEvent`.
5. Deje el resto de las opciones como predeterminadas y elija Crear función.
6. En la pestaña Código de la página de funciones, haga doble clic en index.js.
7. Sustituya el código existente por el código siguiente.

```
'use strict';
```

```
exports.handler = (event, context, callback) => {
    console.log('LogScheduledEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Elija Deploy (Implementar).

Paso 2: Crear una regla

Cree una regla para ejecutar la función de Lambda que creó en el paso 1 conforme a una programación.

Puede utilizar la consola o la AWS CLI para crear la regla. Para usar la AWS CLI, primero debe conceder permiso a la regla para invocar su función de Lambda. A continuación, puede crear la regla y agregar la función de Lambda como destino.

Para crear una regla (consola)

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.
3. Elija Create rule.
4. Escriba un nombre y una descripción de la regla.

   Una regla no puede tener el mismo nombre que otra regla de la misma región y del mismo bus de eventos.
5. En Event bus (Bus de eventos), elija el bus de eventos que desea asociar a esta regla. Si desea que esta regla coincida con eventos procedentes de su cuenta, seleccione Bus de eventos predeterminado de AWS. Cuando un servicio de AWS en su cuenta emite un evento, siempre va al bus de eventos predeterminado de su cuenta.
6. En Rule type (Tipo de regla), elija Schedule (Programación).
7. Elija Next (Siguiente).
8. En Patrón de programación, elija Una programación que se ejecute con una frecuencia regular, por ejemplo, cada 10 minutos. e introducir **5** y elegir Minutos en la lista desplegable.
9. Elija Next (Siguiente).
10. En Target types (Tipos de destino), elija AWS service.

11. En Seleccionar un destino, elija Función de Lambda en la lista desplegable.
12. En Función, seleccione la función de Lambda que creó en la sección Paso 1: Crear una función de Lambda. En este ejemplo, seleccione `LogScheduledEvent`.
13. Elija Next (Siguiente).
14. Elija Next (Siguiente).
15. Revise los detalles de la regla y elija Create rule (Crear regla).

Para crear una regla (AWS CLI)

1. Para crear una regla que se ejecute según una programación, utilice el comando `put-rule`.

```
aws events put-rule \
--name my-scheduled-rule \
--schedule-expression 'rate(5 minutes)'
```

Cuando se ejecuta esta regla, crea un evento y, a continuación, lo envía a los destinos. El siguiente es un evento de ejemplo.

```
{
    "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. Para conceder a la entidad principal del servicio (`events.amazonaws.com`) de EventBridge permiso para ejecutar la regla, utilice el comando `add-permission`.

```
aws lambda add-permission \
--function-name LogScheduledEvent \
--statement-id my-scheduled-event \
--action 'lambda:InvokeFunction' \
```

```
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule
```

3. Cree el archivo `targets.json` con el siguiente contenido.

```
[
  {
    "Id": "1",
    "Arn": "arn:aws:lambda:us-east-1:123456789012:function:LogScheduledEvent"
  }
]
```

4. Para agregar a la regla la función de Lambda que creó en el paso 1, utilice el comando `put-targets`.

```
aws events put-targets --rule my-scheduled-rule --targets file://targets.json
```

Paso 3: Comprobar la regla

Al menos cinco minutos después de completar el paso 2, puede comprobar que se invocó la función de Lambda.

Para ver la salida de la función de Lambda

1. Abra la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs.
3. Seleccione el nombre del grupo de registros para la función Lambda (`/aws/lambda/function-name`).
4. Seleccione el nombre del flujo de registro para ver los datos proporcionados por la función para la instancia que ha lanzado.

Paso 4: Confirmar el éxito

Si ve el evento de Lambda en los registros de CloudWatch, significa que ha completado correctamente este tutorial. Si el evento no está en sus registros de CloudWatch, comience a solucionar problemas verificando que la regla se haya creado correctamente y, si la regla parece correcta, compruebe que el código de la función de Lambda sea correcto.

Paso 5: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Para eliminar las funciones de Lambda

1. Abra la página de Functions (Funciones) en la consola de Lambda.
2. Seleccione las funciones que creó.
3. Elija Actions (Acciones), Delete (Eliminar).
4. Elija Eliminar.

Tutoriales de Amazon EventBridge para la integración con proveedores de SaaS

EventBridge puede trabajar directamente con las aplicaciones y los servicios de los socios de SaaS para enviar y recibir eventos. En los siguientes tutoriales, se explica cómo integrar EventBridge con socios de SaaS.

Tutoriales:

- Tutorial: Creación de una conexión a Datadog como destino de API
- Tutorial: Creación de una conexión a Salesforce como destino de API
- Tutorial: Creación de una conexión a Zendesk como destino de API

Tutorial: Creación de una conexión a Datadog como destino de API

Puede usar EventBridge para enrutar eventos a servicios de terceros, como Datadog.

En este tutorial, utilizará la consola de EventBridge para crear una conexión a Datadog, un destino de API que apunte a Datadog, y una regla para dirigir eventos a Datadog.

Pasos:

- Requisitos previos
- Paso 1: Crear una conexión
- Paso 2: Crear un destino de la API
- Paso 3: Crear regla
- Paso 4: Probar la regla
- Paso 5: Eliminar los recursos

Requisitos previos

Para completar este tutorial necesitará los siguientes recursos:

- Una cuenta de Datadog.
- Una clave de API de Datadog.
- Un bucket de Amazon Simple Storage Service (Amazon S3) habilitado para EventBridge.

Paso 1: Crear una conexión

Para enviar eventos a Datadog, primero tendrá que establecer una conexión con la API de Datadog.

Para crear la conexión

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Destinos de API.
3. Elija la pestaña Conexiones y, a continuación, elija Crear conexión.
4. Escriba un nombre y una descripción para la conexión. Por ejemplo, introduzca **`Datadog`** como nombre y **`Datadog API Connection`** como descripción.
5. En Tipo de autorización, elija Clave de API.

6. En Nombre de clave de API, escriba **`DD-API-KEY`**.
7. En Valor, pegue su clave de API secreta de Datadog.
8. Seleccione Create (Crear).

Paso 2: Crear un destino de la API

Ahora que ha creado la conexión, debe crear el destino de la API para usarlo como destino de la regla.

Para crear el destino de la API

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Destinos de API.
3. Elija Crear destino de API.
4. Escriba un nombre y una descripción para el destino de la API. Por ejemplo, introduzca **`DatadogAD`** para el nombre y **`Datadog API Destination`** para la descripción.
5. En Punto de conexión de destino de la API, introduzca **`https://http-intake.logs.datadoghq.com/api/v2/logs`**.
6. En HTTP method (Método HTTP), elija POST.
7. En Límite de frecuencia de invocación, introduzca **`300`**.
8. En Conexión, elija Usar una conexión existente y elija la conexión de `Datadog` que creó en el paso 1.
9. Seleccione Create (Crear).

Paso 3: Crear regla

A continuación, debe crear una regla para enviar eventos a Datadog cuando se cree un objeto de Amazon S3.

Para crear una regla

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.
3. Elija Create rule.

4. Escriba un nombre y una descripción de la regla. Por ejemplo, introduzca **`DatadogRule`** para el nombre y **`Rule to send events to Datadog for S3 object creation`** para la descripción.
5. En Event bus (Bus de eventos), elija Default (Predeterminado).
6. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).
7. Elija Next (Siguiente).
8. En Event source (Origen del evento), elija Other (Otro).
9. En Patrón de evento, introduzca lo siguiente:

```
{
  "source": ["aws.s3"]
}
```

10. Elija Next (Siguiente).
11. En Tipos de destino, elija Destino de la API de EventBridge.
12. En Destino de la API, seleccione Usar un destino de API existente y, a continuación, elija el destino de `DatadogAD` que creó en el paso 2.
13. En Rol de ejecución, elija Crear un rol nuevo para este recurso específico.
14. En Configuración adicional, haga lo siguiente:
    a. En Configurar entrada de destino, seleccione Transformador de entrada en la lista desplegable.
    b. Elija Configurar transformador de entrada
    c. En Eventos de muestra, introduzca lo siguiente:

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

    d. En Transformador de entrada de destino, haga lo siguiente:
        i. En Ruta de entrada, introduzca lo siguiente:

```
{"detail":"$.detail"}
```

        ii. En Plantilla de entrada, introduzca lo siguiente:

```
{"message": <detail>}
```

e. Elija Confirmar.

15. Elija Next (Siguiente).
16. Elija Next (Siguiente).
17. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 4: Probar la regla

Para probar su regla, cree un objeto de Amazon S3 cargando un archivo en un bucket habilitado para Eventbridge. El objeto creado se registrará en la consola de Registros de Datadog.

Paso 5: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar las conexiones de EventBridge

1. Abra la página Destinos de la API en la consola de EventBridge.
2. Elija la pestaña Connections (Conexiones).
3. Seleccione las conexiones que creó.
4. Elija Eliminar (Delete).
5. Introduzca el nombre de la conexión y elija Eliminar.

Para eliminar los destinos de la API de EventBridge

1. Abra la página Destinos de la API en la consola de EventBridge.
2. Seleccione los destinos de la API que creó.
3. Elija Eliminar (Delete).
4. Introduzca el nombre del destino de la API y elija Eliminar.

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.

2. Seleccione las reglas que creó.
3. Elija Eliminar (Delete).
4. Elija Eliminar (Delete).

Tutorial: Creación de una conexión a Salesforce como destino de API

Se puede utilizar EventBridge para enrutar eventos a servicios de terceros, como Salesforce.

En este tutorial, utilizarás la EventBridge consola para crear una conexiónSalesforce, un destino de API al que apunte y una regla a la que dirigir los eventosSalesforce. Salesforce

Pasos:

- Requisitos previos
- Paso 1: Crear una conexión
- Paso 2: Crear un destino de la API
- Paso 3: Crear regla
- Paso 4: Probar la regla
- Paso 5: Eliminar los recursos

Requisitos previos

Para completar este tutorial necesitará los siguientes recursos:

- Una cuenta de Salesforce.
- Una aplicación de Salesforce conectada.
- Un token de seguridad de Salesforce.
- Un evento de plataforma personalizado de Salesforce.
- Un bucket EventBridge de Amazon Simple Storage Service (Amazon S3) habilitado.

Paso 1: Crear una conexión

Para enviar eventos a Salesforce, primero tendrá que establecer una conexión con la API de Salesforce.

Para crear la conexión

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Destinos de API.
3. Elija la pestaña Conexiones y, a continuación, elija Crear conexión.

4. Escriba un nombre y una descripción para la conexión. Por ejemplo, introduzca **`Salesforce`** como nombre y **`Salesforce API Connection`** como descripción.
5. En Tipo de destino, elija Socios y, en Destinos de socios, seleccione Salesforce de la lista desplegable.
6. En Punto de conexión de autorización, introduzca uno de los siguientes:

   - Si utiliza una organización de producción, introduzca **`https://`*`MyDomainName`*`.my.salesforce.com./services/oauth2/token`**
   - Si utiliza un entorno aislado sin dominios mejorados, introduzca **`https://`*`MyDomainName`*`--`*`SandboxName`*`.my. salesforce.com/services /oauth2/token`**
   - Si utiliza un entorno aislado con dominios mejorados, introduzca **`https://`*`MyDomainName`*`--`*`SandboxName`*`.sandbox.my.salesforce.com/services/oauth2/token`**
7. En Método HTTP, elija POST en la lista desplegable.
8. En ID de cliente, introduzca el ID de cliente de la aplicación de Salesforce conectada.
9. En Secreto de cliente, introduzca el secreto de cliente de la aplicación de Salesforce conectada.
10. Para los parámetros HTTP de OAuth, introduce el siguiente par clave/valor:

Clave	Valor
grant_type	client_credentials

11. Seleccione Crear.

Paso 2: Crear un destino de la API

Ahora que ha creado la conexión, debe crear el destino de la API para usarlo como destino de la regla.

Para crear el destino de la API

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Destinos de API.
3. Elija Crear destino de API.
4. Escriba un nombre y una descripción para el destino de la API. Por ejemplo, introduzca **`SalesforceAD`** para el nombre y **`Salesforce API Destination`** para la descripción.

5. En Punto de conexión de destino de la API, introduzca **`https://MyDomainName.my.salesforce.com/services/data/v54.0/sobjects/MyEvent__e`**, donde MyEvent__E es el evento de la plataforma al que quiere enviar la información.
6. En Método HTTP, elija POST en la lista desplegable.
7. En Límite de frecuencia de invocación, introduzca **`300`**.
8. En Conexión, elija Usar una conexión existente y elija la conexión de `Salesforce` que creó en el paso 1.
9. Seleccione Crear.

Paso 3: Crear regla

A continuación, debe crear una regla para enviar eventos a Salesforce cuando se cree un objeto de Amazon S3.

Para crear una regla

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Reglas.
3. Seleccione Crear regla.
4. Escriba un nombre y una descripción para la regla. Por ejemplo, introduzca **`SalesforceRule`** para el nombre y **`Rule to send events to Salesforce for S3 object creation`** para la descripción.
5. En Event bus (Bus de eventos), elija Default (Predeterminado).
6. En Tipo de regla, elija Regla con un patrón de evento.
7. Seleccione Siguiente.
8. En Origen del evento, seleccione Otro.
9. En Patrón de evento, introduzca lo siguiente:

```
{
  "source": ["aws.s3"]
}
```

10. Elija Siguiente.
11. Para los tipos de destino, elige el destino EventBridge de la API.

12. En Destino de la API, seleccione Usar un destino de API existente y, a continuación, elija el destino de `SalesforceAD` que creó en el paso 2.
13. En Rol de ejecución, elija Crear un rol nuevo para este recurso específico.
14. En Configuración adicional, haga lo siguiente:

    a. En Configurar entrada de destino, seleccione Transformador de entrada en la lista desplegable.

    b. Elija Configure input transformer (Configurar transformador de entrada).

    c. En Eventos de muestra, introduzca lo siguiente:

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

    d. En Transformador de entrada de destino, haga lo siguiente:

        i. En Ruta de entrada, introduzca lo siguiente:

        ```
        {"detail":"$.detail"}
        ```

        ii. En Plantilla de entrada, introduzca lo siguiente:

        ```
        {"message": <detail>}
        ```

    e. Elija Confirmar.
15. Elija Siguiente.
16. Seleccione Siguiente.
17. Revise los detalles de la regla y seleccione Crear regla.

Paso 4: Probar la regla

Para probar la regla, cree un objeto de Amazon S3 cargando un archivo en un bucket EventBridge habilitado. La información sobre el objeto creado se enviará al evento de la plataforma de Salesforce.

Paso 5: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Al eliminar AWS los recursos que ya no utiliza, evita cargos innecesarios en su AWS cuenta.

Para eliminar la (s) EventBridge conexión (es)

1. Abra la página de destino de la API de la EventBridge consola.
2. Elija la pestaña Connections (Conexiones).
3. Seleccione las conexiones que creó.
4. Elija Eliminar.
5. Introduzca el nombre de la conexión y elija Eliminar.

Para eliminar los destinos de la EventBridge API

1. Abre la página de destino de la API de la EventBridge consola.
2. Seleccione los destinos de la API que creó.
3. Elija Eliminar.
4. Introduzca el nombre del destino de la API y elija Eliminar.

Para eliminar la (s) EventBridge regla (s)

1. Abre la página de reglas de la EventBridge consola.
2. Seleccione las reglas que creó.
3. Elija Eliminar.
4. Elija Eliminar.

Tutorial: Creación de una conexión a Zendesk como destino de API

Puede usar EventBridge para dirigir eventos a servicios de terceros, como Zendesk.

En este tutorial, utilizará la consola de EventBridge para crear una conexión a Zendesk, un destino de API que apunte a Zendesk, y una regla para dirigir eventos a Zendesk.

Pasos:

- Requisitos previos
- Paso 1: Crear una conexión
- Paso 2: Crear un destino de la API
- Paso 3: Crear regla
- Paso 4: Probar la regla
- Paso 5: Eliminar los recursos

Requisitos previos

Para completar este tutorial necesitará los siguientes recursos:

- Una cuenta de Zendesk.
- Un bucket de Amazon Simple Storage Service (Amazon S3) habilitado para EventBridge.

Paso 1: Crear una conexión

Para enviar eventos a Zendesk, primero tendrá que establecer una conexión con la API de Zendesk.

Para crear la conexión

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Destinos de API.
3. Elija la pestaña Conexiones y, a continuación, elija Crear conexión.
4. Escriba un nombre y una descripción para la conexión. Por ejemplo, introduzca **`Zendesk`** para el nombre y **`Connection to Zendesk API`** para la descripción.
5. En Tipo de autorización, elija Basic (Nombre de usuario y contraseña).
6. En Nombre de usuario, introduzca su nombre de usuario de Zendesk.

7. En Contraseña, introduzca su contraseña de Zendesk.
8. Seleccione Create (Crear).

Paso 2: Crear un destino de la API

Ahora que ha creado la conexión, debe crear el destino de la API para usarlo como destino de la regla.

Para crear el destino de la API

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Destinos de API.
3. Elija Crear destino de API.
4. Escriba un nombre y una descripción para el destino de la API. Por ejemplo, introduzca **`ZendeskAD`** para el nombre y **`Zendesk API destination`** para la descripción.
5. En Punto de conexión de destino de la API, introduzca **`https://`*`your-subdomain`*`.zendesk.com/api/v2/tickets.json`**, donde *`your-subdomain`* es el subdominio asociado a su cuenta de Zendesk.
6. En HTTP method (Método HTTP), elija POST.
7. En Límite de frecuencia de invocación, introduzca **`10`**.
8. En Conexión, elija Usar una conexión existente y elija la conexión de `Zendesk` que creó en el paso 1.
9. Seleccione Create (Crear).

Paso 3: Crear regla

A continuación, cree una regla para enviar eventos a Zendesk cuando se cree un objeto de Amazon S3.

Para crear una regla de

1. Abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación, seleccione Rules.
3. Elija Create rule.

4. Escriba un nombre y una descripción de la regla. Por ejemplo, introduzca **`ZendeskRule`** para el nombre y **`Rule to send events to Zendesk when S3 objects are created`** para la descripción.
5. En Event bus (Bus de eventos), elija Default (Predeterminado).
6. En Rule type (Tipo de regla), elija Rule with an event pattern (Regla con un patrón de evento).
7. Elija Next (Siguiente).
8. En Event source (Origen del evento), elija Other (Otro).
9. En Patrón de evento, introduzca lo siguiente:

```
{
  "source": ["aws.s3"]
}
```

10. Elija Next (Siguiente).
11. En Tipos de destino, elija Destino de la API de EventBridge.
12. En Destino de la API, seleccione Usar un destino de API existente y, a continuación, elija el destino de `ZendeskAD` que creó en el paso 2.
13. En Rol de ejecución, elija Crear un rol nuevo para este recurso específico.
14. En Configuración adicional, haga lo siguiente:
    a. En Configurar entrada de destino, seleccione Transformador de entrada en la lista desplegable.
    b. Elija Configurar transformador de entrada
    c. En Eventos de muestra, introduzca lo siguiente:

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

    d. En Transformador de entrada de destino, haga lo siguiente:
        i. En Ruta de entrada, introduzca lo siguiente:

```
{"detail":"$.detail"}
```

        ii. En Plantilla de entrada, introduzca lo siguiente:

```
{"message": <detail>}
```

e. Elija Confirmar.

15. Elija Next (Siguiente).
16. Elija Next (Siguiente).
17. Revise los detalles de la regla y elija Create rule (Crear regla).

Paso 4: Probar la regla

Para probar su regla, cree un objeto de Amazon S3 cargando un archivo en un bucket habilitado para Eventbridge. Cuando el evento cumpla con la regla, EventBridge llamará a la API crear ticket de Zendesk. El nuevo ticket aparecerá en el panel de control de Zendesk.

Paso 5: Eliminar los recursos

A menos que desee conservar los recursos que creó para este tutorial, puede eliminarlos ahora. Si elimina los recursos de AWS que ya no utiliza, evitará gastos innecesarios en su cuenta de AWS.

Para eliminar las conexiones de EventBridge

1. Abra la página Destinos de la API en la consola de EventBridge.
2. Elija la pestaña Connections (Conexiones).
3. Seleccione las conexiones que creó.
4. Elija Eliminar (Delete).
5. Introduzca el nombre de la conexión y elija Eliminar.

Para eliminar los destinos de la API de EventBridge

1. Abra la página Destinos de la API en la consola de EventBridge.
2. Seleccione los destinos de la API que creó.
3. Elija Eliminar (Delete).
4. Introduzca el nombre del destino de la API y elija Eliminar.

Para eliminar las reglas de EventBridge

1. Abra la página Reglas en la consola de EventBridge.
2. Seleccione las reglas que creó.
3. Elija Eliminar (Delete).
4. Elija Eliminar (Delete).

EventBridge Utilizándolo con un AWS SDK

AWS Los kits de desarrollo de software (SDK) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK for C++	AWS SDK for C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK for Go	AWS SDK for Go ejemplos de código
AWS SDK for Java	AWS SDK for Java ejemplos de código
AWS SDK for JavaScript	AWS SDK for JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK for .NET	AWS SDK for .NET ejemplos de código
AWS SDK for PHP	AWS SDK for PHP ejemplos de código
AWS Tools for PowerShell	Herramientas para ejemplos PowerShell de código
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) ejemplos de código
AWS SDK for Ruby	AWS SDK for Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Para ver ejemplos específicos de EventBridge, consulte[Ejemplos de código para EventBridge usar los AWS SDK](#).

Ejemplo de disponibilidad

¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Ejemplos de código para EventBridge usar los AWS SDK

Los siguientes ejemplos de código muestran cómo usarlo EventBridge con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Los ejemplos con varios servicios son aplicaciones de muestra que funcionan con varios Servicios de AWS.

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Introducción

Hola EventBridge

Los siguientes ejemplos de código muestran cómo empezar a usarlo EventBridge.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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();
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListEventBusesla Referencia AWS SDK for .NET de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListEventBuseslа Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}")
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListEventBuseslа referencia sobre el AWS SDK para la API de Kotlin.

Ejemplos de código

Acciones para EventBridge usar los SDK AWS

Los siguientes ejemplos de código muestran cómo realizar EventBridge acciones individuales con los AWS SDK. Estos extractos se denominan EventBridge API y son fragmentos de código de programas más grandes que deben ejecutarse en su contexto. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para obtener una lista completa, consulta la referencia de las EventBridge API de Amazon.

Ejemplos

- Úselo EnableRule con un AWS SDK o CLI
- Úselo ListRuleNamesByTarget con un AWS SDK o CLI
- Úselo ListRules con un AWS SDK o CLI
- Úselo ListTargetsByRule con un AWS SDK o CLI
- Úselo PutEvents con un AWS SDK o CLI
- Úselo PutRule con un AWS SDK o CLI
- Úselo PutTargets con un AWS SDK o CLI
- Úselo RemoveTargets con un AWS SDK o CLI

Úselo **`DeleteRule`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DeleteRule`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Eliminar una regla por su nombre.

```
/// <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;
    }
```

- Para obtener más información sobre la API, consulta DeleteRulela Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Para eliminar una regla de CloudWatch eventos

En este ejemplo se elimina la regla denominada InstanceStateChanges EC2:

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- Para obtener más información sobre la API, consulte la Referencia DeleteRulede AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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");
    }
```

- Para obtener más información sobre la API, consulta DeleteRulela Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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")
    }
}
```

- Para obtener más información sobre la API, consulta DeleteRulela referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DescribeRule`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DescribeRule`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Obtener el estado de una regla mediante la descripción de la regla.

```
   /// <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;
   }
```

- Para obtener más información sobre la API, consulta DescribeRulela Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Para mostrar información sobre una regla de CloudWatch eventos

En este ejemplo se muestra información sobre la regla denominada DailyLambdaFunction:

```
aws events describe-rule --name "DailyLambdaFunction"
```

- Para obtener más información sobre la API, consulte DescribeRulela Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
        }
    }
```

- Para obtener más información sobre la API, consulta DescribeRulela Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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")
    }
}
```

- Para obtener más información sobre la API, consulta DescribeRulela referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DisableRule`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DisableRule`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Deshabilitar una regla por su nombre de regla.

```
/// <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;
}
```

- Para obtener más información sobre la API, consulta DisableRulela Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Para deshabilitar una regla de CloudWatch eventos

En este ejemplo, se deshabilita la regla denominada DailyLambdaFunction. La regla no se elimina:

```
aws events disable-rule --name "DailyLambdaFunction"
```

- Para obtener más información sobre la API, consulte DisableRulela Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Deshabilitar una regla por su nombre de regla.

```
   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);
```

```
        }
    }
```

- Para obtener más información sobre la API, consulta DisableRulela Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
        }
    }
}
```

- Para obtener más información sobre la API, consulta DisableRulela referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`EnableRule`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `EnableRule`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Habilitar una regla por su nombre de regla.

```
/// <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;
}
```

- Para obtener más información sobre la API, consulta EnableRulela Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Para habilitar una regla de CloudWatch eventos

Este ejemplo habilita la regla denominada DailyLambdaFunction, que estaba deshabilitada anteriormente:

```
aws events enable-rule --name "DailyLambdaFunction"
```

- Para obtener más información sobre la API, consulte EnableRulela Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Habilitar una regla por su nombre de regla.

```
   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);
        }
    }
```

- Para obtener más información sobre la API, consulta EnableRulela Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
        }
    }
}
```

- Para obtener más información sobre la API, consulta EnableRulela referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListRuleNamesByTarget`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ListRuleNamesByTarget`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enumerar todos los nombres de regla por el destino.

```
    /// <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;
    }
```

- Para obtener más información sobre la API, consulta ListRuleNamesByTargetla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Visualización de una lista de todas las reglas que tienen un destino especificado

En este ejemplo se muestran todas las reglas que tienen como destino la función Lambda denominada MyFunctionName "»:

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-
east-1:123456789012:function:MyFunctionName"
```

- Para obtener más información sobre la API, consulte ListRuleNamesByTargetla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enumerar todos los nombres de regla por el destino.

```
   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);
       }
   }
```

- Para obtener más información sobre la API, consulta ListRuleNamesByTargetla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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")
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListRuleNamesByTargetla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListRules`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ListRules`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enumerar todas las reglas de un bus de eventos.

```
   /// <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;
   }
```

- Para obtener más información sobre la API, consulta ListRulesla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Para mostrar una lista de todas las reglas de CloudWatch eventos

En este ejemplo se muestran todas las reglas de CloudWatch eventos de la región:

```
aws events list-rules
```

Para mostrar una lista de reglas de CloudWatch eventos que comiencen por una cadena determinada.

En este ejemplo, se muestran todas las reglas de CloudWatch eventos de la región cuyo nombre comience por «Daily»:

```
aws events list-rules --name-prefix "Daily"
```

- Para obtener más información sobre la API, consulte ListRulesla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Habilitar una regla por su nombre de regla.

```
    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);
        }
    }
```

- Para obtener más información sobre la API, consulta ListRulesla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}")
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListRulesla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`ListTargetsByRule`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `ListTargetsByRule`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enumerar todos los destinos para una regla por el nombre de regla.

```
/// <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;
    }
```

- Para obtener más información sobre la API, consulta ListTargetsByRulela Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Para mostrar todos los objetivos de una regla de CloudWatch eventos

En este ejemplo se muestran todos los objetivos de la regla denominada DailyLambdaFunction:

```
aws events list-targets-by-rule --rule  "DailyLambdaFunction"
```

- Para obtener más información sobre la API, consulte ListTargetsByRulela Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enumerar todos los destinos para una regla por el nombre de regla.

```
    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());
        }
    }
```

- Para obtener más información sobre la API, consulta ListTargetsByRulela Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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}")
        }
    }
}
```

- Para obtener más información sobre la API, consulta ListTargetsByRulela referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`PutEvents`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `PutEvents`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en los siguientes ejemplos de código:

- Crear y activar una regla
- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Enviar un evento que coincida con el patrón personalizado de una regla.

```
    /// <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;
    }
```

- Para obtener más información sobre la API, consulta PutEventsla Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#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>
```

Enviar el 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;
        }
```

- Para obtener más información sobre la API, consulta PutEventsla Referencia AWS SDK for C++ de la API.

CLI

AWS CLI

Para enviar un evento personalizado a CloudWatch Events

En este ejemplo, se envía un evento personalizado a CloudWatch Events. El evento está incluido en el archivo putevents.json:

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

Este es el contenido del archivo 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"
   }
]
```

- Para obtener más información sobre la API, consulte PutEventsla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
    }
```

- Para obtener más información sobre la API, consulta PutEventsla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import {
  EventBridgeClient,
  PutEventsCommand,
} from "@aws-sdk/client-eventbridge";

export const putEvents = async (
  source = "eventbridge.integration.test",
  detailType = "greeting",
  resources = [],
) => {
  const client = new EventBridgeClient({});
```

```
  const response = await client.send(
    new PutEventsCommand({
      Entries: [
        {
          Detail: JSON.stringify({ greeting: "Hello there." }),
          DetailType: detailType,
          Resources: resources,
          Source: source,
        },
      ],
    }),
  );

  console.log("PutEvents response:");
  console.log(response);
  // PutEvents response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: '3d0df73d-dcea-4a23-ae0d-f5556a3ac109',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   Entries: [ { EventId: '51620841-5af4-6402-d9bc-b77734991eb5' } ],
  //   FailedEntryCount: 0
  // }

  return response;
};
```

- Para obtener más información sobre la API, consulta PutEventsla Referencia AWS SDK for JavaScript de la API.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
  }
});
```

- Para obtener más información sobre la API, consulta PutEventsla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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)
    }
}
```

- Para obtener más información sobre la API, consulta PutEventsla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`PutRule`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `PutRule`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en los siguientes ejemplos de código:

- Crear y activar una regla
- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear una regla que se active cuando se agregue un objeto a un bucket de 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;
    }
```

Crear una regla que use un patrón personalizado.

```
    /// <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;
    }
```

- Para obtener más información sobre la API, consulta PutRulela Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#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>
```

Crear la regla.

```
        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;
        }
```

- Para obtener más información sobre la API, consulta PutRulela Referencia AWS SDK for C ++ de la API.

CLI

AWS CLI

Para crear reglas de CloudWatch eventos

Este ejemplo crea una regla que se activa cada día a las 9:00 (UTC). Si utiliza put-targets para añadir una función de Lambda como destino de esta regla, puede ejecutar la función de Lambda todos los días a la hora especificada:

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9
 * * ? *)"
```

En este ejemplo se crea una regla que se activa cuando una instancia EC2 de la región cambia de estado:

```
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"
```

En este ejemplo, se crea una regla que se activa cuando se detiene o se termina una instancia de EC2 de la región:

```
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"
```

- Para obtener más información sobre la API, consulte PutRulela Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear una regla programada.

```
   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);
       }
   }
```

Crear una regla que se active cuando se agregue un objeto a un bucket de 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);
       }
   }
```

- Para obtener más información sobre la API, consulta PutRulela Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import { EventBridgeClient, PutRuleCommand } from "@aws-sdk/client-eventbridge";

export const putRule = async (
  ruleName = "some-rule",
  source = "some-source",
) => {
  const client = new EventBridgeClient({});

  const response = await client.send(
    new PutRuleCommand({
      Name: ruleName,
      EventPattern: JSON.stringify({ source: [source] }),
      State: "ENABLED",
      EventBusName: "default",
    }),
  );

  console.log("PutRule response:");
  console.log(response);
  // PutRule response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: 'd7292ced-1544-421b-842f-596326bc7072',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   RuleArn: 'arn:aws:events:us-east-1:xxxxxxxxxxxx:rule/
EventBridgeTestRule-1696280037720'
```

```
  // }
  return response;
};
```

- Para obtener más información sobre la API, consulta PutRulela Referencia AWS SDK for JavaScript de la API.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
  }
});
```

- Para obtener más información sobre la API, consulta PutRulela Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Crear una regla programada.

```
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}")
    }
}
```

Crear una regla que se active cuando se agregue un objeto a un bucket de 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}")
    }
}
```

- Para obtener más información sobre la API, consulta PutRulela referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`PutTargets`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `PutTargets`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Introducción a las reglas y los destinos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Agregue un tema de Amazon SNS como destino de una regla.

```
    /// <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;
    }
```

Agregue un transformador de entrada al destino de una regla.

```
    /// <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;
    }
```

- Para obtener más información sobre la API, consulta PutTargetsla Referencia AWS SDK for .NET de la API.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#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>
```

Añada el destino.

```
        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;
        }
```

- Para obtener más información sobre la API, consulta PutTargetsla Referencia AWS SDK for C++ de la API.

CLI

AWS CLI

Para añadir objetivos a las reglas de CloudWatch Events

En este ejemplo, se añade una función de Lambda como destino de una regla:

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="1","Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

En este ejemplo, se establece una transmisión de Amazon Kinesis como destino, de modo que los eventos detectados por esta regla se retransmitan a la transmisión:

```
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"
```

En este ejemplo, se establecen dos transmisiones de Amazon Kinesis como destinos para una regla:

```
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"
```

- Para obtener más información sobre la API, consulte PutTargetsla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Agregue un tema de Amazon SNS como destino de una regla.

```
    // 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 + ".");
    }
```

Agregue un transformador de entrada al destino de una regla.

```
    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);
        }
    }
```

- Para obtener más información sobre la API, consulta PutTargetsla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Importar el SDK y los módulos de cliente, y llamar a la API.

```
import {
  EventBridgeClient,
  PutTargetsCommand,
} from "@aws-sdk/client-eventbridge";

export const putTarget = async (
  existingRuleName = "some-rule",
  targetArn = "arn:aws:lambda:us-east-1:000000000000:function:test-func",
  uniqueId = Date.now().toString(),
) => {
  const client = new EventBridgeClient({});
  const response = await client.send(
    new PutTargetsCommand({
      Rule: existingRuleName,
      Targets: [
        {
          Arn: targetArn,
```

```
          Id: uniqueId,
        },
      ],
    }),
  );

  console.log("PutTargets response:");
  console.log(response);
  // PutTargets response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: 'f5b23b9a-2c17-45c1-ad5c-f926c3692e3d',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   FailedEntries: [],
  //   FailedEntryCount: 0
  // }

  return response;
};
```

- Para obtener más información sobre la API, consulta PutTargetsla Referencia AWS SDK for JavaScript de la API.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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);
  }
});
```

- Para obtener más información sobre la API, consulta PutTargetsla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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.")
    }
}
```

Agregue un transformador de entrada al destino de una regla.

```
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)
    }
}
```

- Para obtener más información sobre la API, consulta PutTargetsla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`RemoveTargets`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `RemoveTargets`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Eliminar todos los destinos de una regla por el nombre de regla.

```
    /// <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;
    }
```

- Para obtener más información sobre la API, consulta RemoveTargetsla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

Eliminación de un destino de un evento

En este ejemplo, la transmisión de Amazon Kinesis denominada MyStream 1 deja de ser el objetivo de la regla. DailyLambdaFunction Cuando DailyLambdaFunction se creó, esta transmisión se estableció como un objetivo con un ID de Target1:

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- Para obtener más información sobre la API, consulte RemoveTargetsla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Eliminar todos los destinos de una regla por el nombre de regla.

```
   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");
       }
   }
```

- Para obtener más información sobre la API, consulta RemoveTargetsla Referencia AWS SDK for Java 2.x de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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")
            }
        }
    }
}
```

- Para obtener más información sobre la API, consulta RemoveTargetsla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulta EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Escenarios de EventBridge uso de AWS los SDK

Los siguientes ejemplos de código muestran cómo implementar escenarios comunes EventBridge con los AWS SDK. Estos escenarios muestran cómo realizar tareas específicas mediante la invocación de varias funciones internas EventBridge. Cada escenario incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código.

Ejemplos

- Crea y activa una regla en Amazon EventBridge mediante un AWS SDK
- Comience con EventBridge las reglas y los objetivos mediante un AWS SDK

Crea y activa una regla en Amazon EventBridge mediante un AWS SDK

El siguiente ejemplo de código muestra cómo crear y activar una regla en Amazon EventBridge.

Ruby

SDK para Ruby

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Llamar a las funciones en el orden correcto.

```
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"
```

Comprobar si el tema de Amazon Simple Notification Service (Amazon SNS) especificado existe entre los que se proporcionan para esta función.

```
# 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
```

Comprobar si el tema especificado existe entre los disponibles para el intermediario en Amazon SNS.

```
# Checks whether the specified topic exists among those available to the
# caller in Amazon SNS.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   exit 1 unless topic_exists?(
#     Aws::SNS::Client.new(region: 'us-east-1'),
```

```
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def topic_exists?(sns_client, topic_arn)
  puts "Searching for topic with ARN '#{topic_arn}'..."
  response = sns_client.list_topics
  if response.topics.count.positive?
    if topic_found?(response.topics, topic_arn)
      puts "Topic found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.topics.count.positive?
        if topic_found?(response.topics, topic_arn)
          puts "Topic found."
          return true
        end
      end
    end
  end
  puts "Topic not found."
  return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end
```

Crear un tema en Amazon SNS y después suscribe una dirección de correo electrónico para recibir notificaciones sobre dicho tema.

```
# 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
```

Compruebe si el rol especificado AWS Identity and Access Management (IAM) existe entre los que se proporcionan a esta función.

```
# 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
```

Comprobar si el rol especificado existe entre los disponibles para el intermediario en IAM.

```
# Checks whether the specified role exists among those available to the
# caller in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   exit 1 unless role_exists?(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
def role_exists?(iam_client, role_arn)
  puts "Searching for role with ARN '#{role_arn}'..."
  response = iam_client.list_roles
  if response.roles.count.positive?
    if role_found?(response.roles, role_arn)
      puts "Role found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.roles.count.positive?
        if role_found?(response.roles, role_arn)
          puts "Role found."
          return true
        end
      end
    end
  end
  puts "Role not found."
  return false
rescue StandardError => e
  puts "Role not found: #{e.message}"
  return false
end
```

Crear un rol en IAM.

```
# Creates a role in AWS Identity and Access Management (IAM).
# This role is used by a rule in Amazon EventBridge to allow
# that rule to operate within the caller's account.
# This role is designed to be used specifically by this code example.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to create.
# @return [String] The ARN of the role that was created.
# @example
#   puts create_role(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def create_role(iam_client, role_name)
  puts "Creating the role named '#{role_name}'..."
  response = iam_client.create_role(
    assume_role_policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "",
          'Effect': "Allow",
          'Principal': {
            'Service': "events.amazonaws.com"
          },
          'Action': "sts:AssumeRole"
        }
      ]
    }.to_json,
    path: "/",
    role_name: role_name
  )
  puts "Role created with ARN '#{response.role.arn}'."
  puts "Adding access policy to role..."
  iam_client.put_role_policy(
    policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
```

```
          'Sid': "CloudWatchEventsFullAccess",
          'Effect': "Allow",
          'Resource': "*",
          'Action': "events:*"
        },
        {
          'Sid': "IAMPassRoleForCloudWatchEvents",
          'Effect': "Allow",
          'Resource': "arn:aws:iam::*:role/AWS_Events_Invoke_Targets",
          'Action': "iam:PassRole"
        }
      ]
    }.to_json,
    policy_name: "CloudWatchEventsPolicy",
    role_name: role_name
  )
  puts "Access policy added to role."
  return response.role.arn
rescue StandardError => e
  puts "Error creating role or adding policy to it: #{e.message}"
  puts "If the role was created, you must add the access policy " \
    "to the role yourself, or delete the role yourself and try again."
  return "Error"
end
```

Comprueba si la EventBridge regla especificada existe entre las que se proporcionan a esta función.

```
# 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
```

Comprueba si la regla especificada existe entre las disponibles para la persona que llama. 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 regla en 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
```

Comprueba si el grupo de registros especificado existe entre los disponibles para la persona que llama en 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
```

Cree un grupo de CloudWatch registros en 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
```

Escribe un evento en una secuencia de CloudWatch registros en 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.",
#     '495426724868310740095796045676565788214806863282469607 3EX'
#   )
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
```

Reinicie una instancia de Amazon Elastic Compute Cloud (Amazon EC2) y añada información sobre la actividad relacionada a una secuencia de registros en Logs. CloudWatch

```
# Restarts an Amazon EC2 instance
# and adds information about the related activity to a log stream
# in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - The Amazon EC2 instance to restart.
# - The log group in Amazon CloudWatch Logs to add related activity
#   information to.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client]
#   An initialized Amazon CloudWatch Logs client.
# @param instance_id [String] The ID of the instance.
# @param log_group_name [String] The name of the log group.
# @return [Boolean] true if the instance was restarted and the information
#   was written to the log stream; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX',
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  log_stream_name = "#{Time.now.year}/#{Time.now.month}/#{Time.now.day}/" \
    "#{SecureRandom.uuid}"
  cloudwatchlogs_client.create_log_stream(
    log_group_name: log_group_name,
    log_stream_name: log_stream_name
  )
  sequence_token = ""

  puts "Attempting to stop the instance with the ID '#{instance_id}'. " \
```

```
    "This might take a few minutes..."
  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts "Instance stopped."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' stopped.",
    sequence_token
  )

  puts "Attempting to restart the instance. This might take a few minutes..."
  ec2_client.start_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
  puts "Instance restarted."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' restarted.",
    sequence_token
  )

  return true
rescue StandardError => e
  puts "Error creating log stream or stopping or restarting the instance: " \
    "#{e.message}"
  log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Error stopping or starting instance '#{instance_id}': #{e.message}",
    sequence_token
  )
  return false
end
```

Muestra información sobre la actividad de una regla en 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
```

Muestra la información de registro de todos los flujos de registros de un grupo de CloudWatch registros.

```
# 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
```

Muestra un recordatorio a la persona que llama para que limpie manualmente AWS los recursos asociados que ya no necesite.

```
# 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__
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK for Ruby .
  - PutEvents
  - PutRule

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Comience con EventBridge las reglas y los objetivos mediante un AWS SDK

En el siguiente ejemplo de código, se muestra cómo:

- Crear una regla y agregarle un destino.
- Habilitar y deshabilitar reglas.
- Enumerar y actualizar reglas y destinos.
- Enviar eventos y, después, limpiar los recursos

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Ejecutar un escenario interactivo en un símbolo del sistema.

```
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 clase que abarque EventBridge las operaciones.

```
/// <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;
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK for .NET .
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Java

SDK para Java 2.x

> **Note**
>
> Hay más en marcha. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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 "";
    }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la referencia de la API de AWS SDK for Java 2.x .
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de 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
    }
}
```

- Para obtener información acerca de la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Kotlin.
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de servicios cruzados para EventBridge usar los SDK AWS

Las siguientes aplicaciones de ejemplo utilizan AWS los SDK para combinarlos EventBridge con otros. Servicios de AWS Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar la aplicación.

Ejemplos

- Usar eventos programados para invocar una función de Lambda

Usar eventos programados para invocar una función de Lambda

Los siguientes ejemplos de código muestran cómo crear una AWS Lambda función invocada por un evento EventBridge programado de Amazon.

Java

SDK para Java 2.x

Muestra cómo crear un evento EventBridge programado de Amazon que invoque una AWS Lambda función. Configure EventBridge para usar una expresión cron para programar cuándo se invoca la función Lambda. En este ejemplo, creará una función de Lambda utilizando la API de tiempo de ejecución de Lambda Java. En este ejemplo, se invocan diferentes AWS servicios para realizar un caso de uso específico. Este ejemplo indica cómo crear una aplicación que envíe un mensaje de texto a sus empleados para felicitarles por su primer aniversario.

Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

JavaScript

SDK para JavaScript (v3)

Muestra cómo crear un evento EventBridge programado de Amazon que invoque una AWS Lambda función. Configure EventBridge para usar una expresión cron para programar cuándo se invoca la función Lambda. En este ejemplo, se crea una función de Lambda mediante la API de tiempo de ejecución de JavaScript Lambda. En este ejemplo, se invocan diferentes AWS servicios para realizar un caso de uso específico. Este ejemplo indica cómo crear una aplicación que envíe un mensaje de texto a sus empleados para felicitarles por su primer aniversario.

Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en GitHub.

Este ejemplo también está disponible en la guía para desarrolladores de AWS SDK for JavaScript v3.

Servicios utilizados en este ejemplo

- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

Python

SDK para Python (Boto3)

En este ejemplo se muestra cómo registrar una AWS Lambda función como destino de un EventBridge evento programado de Amazon. El controlador Lambda escribe un mensaje descriptivo y los datos completos del evento en Amazon CloudWatch Logs para su posterior recuperación.

- Implementa una función de Lambda.
- Crea un evento EventBridge programado y convierte la función Lambda en el objetivo.
- Otorga permiso para EventBridge invocar la función Lambda.
- Imprime los datos más recientes de CloudWatch los registros para mostrar el resultado de las invocaciones programadas.
- Limpia todos los recursos creados durante la demostración.

Es mejor ver este ejemplo en GitHub. Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- CloudWatch Registros
- EventBridge
- Lambda

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulte EventBridge Utilizándolo con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Amazon EventBridge seguridad

Amazon EventBridge AWS Identity and Access Management se utiliza para controlar el acceso a otros AWS servicios y recursos. Para obtener información general sobre cómo funciona IAM, consulte Información general sobre la administración del acceso en la Guía del usuario de IAM. Para obtener información general de credenciales de seguridad, consulte Credenciales de seguridad de AWS en la Referencia general de Amazon Web Services.

Temas

Protección de datos en Amazon EventBridge

El modelo de responsabilidad AWS compartida modelo se aplica a la protección de datos en Amazon EventBridge. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja EventBridge o Servicios de AWS utiliza la consola, la API o los SDK. AWS CLI AWS Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o

diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Cifrado de datos para buses de EventBridge eventos

EventBridge proporciona cifrado en reposo y cifrado en tránsito para proteger los datos del evento:

- Cifrado en reposo

  EventBridge se integra con AWS Key Management Service (KMS) para cifrar los datos de eventos almacenados en los buses de eventos. De forma predeterminada, EventBridge utiliza an Clave propiedad de AWS para cifrar los datos de los eventos. También puede especificar que se utilice en su lugar clave administrada por el cliente para eventos personalizados y de socios. EventBridge
- Cifrado en tránsito

  EventBridge cifra los datos que se transfieren entre EventBridge y otros servicios mediante la seguridad de la capa de transporte (TLS). En el caso de los buses de eventos, esto incluye tanto durante el envío de un evento como cuando se EventBridge envía un evento a un objetivo de regla. EventBridge

Cifrado en reposo para los buses de eventos

EventBridge proporciona un cifrado transparente del lado del servidor mediante la integración con AWS Key Management Service (KMS). El cifrado de los datos en reposo de forma predeterminada ayuda a reducir la sobrecarga operativa y la complejidad que implica la protección de los datos confidenciales. Al mismo tiempo, le permite crear aplicaciones seguras que cumplen con los estrictos requisitos normativos y de conformidad con el cifrado.

Los datos del bus de eventos EventBridge cifrados en reposo incluyen:

- Datos de eventos para AWSeventos personalizados y de socios.

  En el caso de los buses de eventos, los datos del evento incluyen todos los campos contenidos en el ??? elemento del evento.

  EventBridge no cifra los metadatos del evento. Para obtener más información sobre los metadatos de los eventos, consulte???.
- Patrones de eventos
- Transformadores de entrada

De forma predeterminada, EventBridge utiliza an Clave propiedad de AWS para cifrar los datos de los eventos. También puede especificar que se utilice en su lugar clave administrada por el cliente para eventos personalizados y de socios. EventBridge

Consideraciones de seguridad para el cifrado del bus de eventos

Le recomendamos encarecidamente que nunca coloque información confidencial o sensible en los siguientes campos, ya que no están cifrados en reposo:

- Nombres de los buses de eventos
- Nombres de reglas
- Recursos compartidos, como etiquetas

KMS key opciones para el cifrado del bus de eventos

EventBridge utiliza y Clave propiedad de AWS para cifrar los eventos del AWS servicio almacenados en los buses de eventos.

Para cada bus de eventos, puede elegir el tipo de KMS key EventBridge usos para cifrar los eventos personalizados y asociados almacenados en ese bus:

- Clave propiedad de AWS

  De forma predeterminada, EventBridge cifra los datos con el estándar de cifrado avanzado (AES-256) de 256 bits bajo un Clave propiedad de AWS, lo que ayuda a proteger sus datos contra el acceso no autorizado.

  No puede ver, administrar, usar ni auditar su uso Claves propiedad de AWS. Sin embargo, no tiene que realizar ninguna acción ni cambiar ningún programa para proteger las claves que cifran sus datos.

  En general, a menos que tengas que auditar o controlar la clave de cifrado que protege tus recursos, an Clave propiedad de AWS es una buena opción. Claves propiedad de AWS son completamente gratuitos (sin cuotas mensuales ni de uso) y no se descontarán de las AWS KMS cuotas de su cuenta. No es necesario crear ni mantener la clave ni su política de claves.

  Para obtener más información, consulte las claves propiedad de AWS en la Guía para desarrolladores de AWS Key Management Service .
- Clave administrada por el cliente

EventBridge admite el uso de un sistema simétrico clave administrada por el cliente que usted crea, posee y administra. Como tiene el control total de este tipo de sistemas KMS key, puede realizar tareas como las siguientes:

- Establecer y mantener políticas de claves
- Establecer y mantener concesiones y políticas de IAM
- Habilitar y deshabilitar políticas de claves
- Rotar el material criptográfico
- Agregar etiquetas.
- Crear alias de clave
- Programar la eliminación de claves

Para obtener más información, consulte Claves administradas por el cliente en la Guía para desarrolladores de AWS Key Management Service .

EventBridge admite claves multirregionales y el acceso a las claves entre cuentas.

Claves administradas por el cliente incurren en una cuota mensual. Para obtener más información, consulta AWS Key Management Service los precios y las cuotas en la Guía para AWS Key Management Service desarrolladores.

> **Note**
>
> EventBridge no admite las siguientes funciones en los buses de eventos cifrados mediante claves administradas por el cliente:
>
> - Archivos
> - Descubrimiento de esquemas

Para obtener más información, consulte ???.

Cifrar eventos con claves administradas por el cliente

Puede especificar que EventBridge utilice AWS KMS clave administrada por el cliente a para cifrar los datos (eventos personalizados y asociados) almacenados en un bus de eventos, en lugar de utilizar un Clave propiedad de AWS como es el predeterminado. Puede especificar un clave

administrada por el cliente al crear o actualizar un bus de eventos. También puede actualizar el bus de eventos predeterminado para usarlo también clave administrada por el cliente para eventos personalizados y asociados. Para obtener más información, consulte ???.

Si especifica un clave administrada por el cliente para un bus de eventos, tiene la opción de especificar una cola de espera (DLQ) para el bus de eventos. EventBridge a continuación, envía a ese DLQ cualquier evento personalizado o asociado que genere errores de cifrado o descifrado. Para obtener más información, consulte ???.

Especificar una clave administrada por el cliente para el cifrado al crear un bus de eventos (mediante la consola)

- Siga estas instrucciones:

  ???.

Especificar un clave administrada por el cliente para el cifrado al crear un bus de eventos (mediante la CLI)

- Al llamar`create-event-bus`, utilice la `kms-key-identifier` opción para especificar el clave administrada por el cliente for que se va EventBridge a utilizar para el cifrado en el bus de eventos.

  Si lo desea, puede utilizar `dead-letter-config` esta opción para especificar una cola de cartas muertas (DLQ).

Actualización de un bus de eventos para usarlo como cifrado (mediante clave administrada por el cliente la consola)

- Siga estas instrucciones:

  ???.

Actualización de un bus de eventos para usar un clave administrada por el cliente para el cifrado (mediante la CLI)

- Al llamar`update-event-bus`, utilice la `kms-key-identifier` opción para especificar el foro que se utilizará clave administrada por el cliente EventBridge para el cifrado en el bus de eventos.

Si lo desea, puede utilizar `dead-letter-config` esta opción para especificar una cola de cartas muertas (DLQ).

Actualizar el bus de eventos predeterminado para usar un para el cifrado mediante clave administrada por el clienteCloudFormation

Como EventBridge aprovisiona automáticamente el bus de eventos predeterminado en tu cuenta, no puedes crearlo mediante una CloudFormation plantilla, como harías normalmente con cualquier recurso que quisieras incluir en una CloudFormation pila. Para incluir el bus de eventos predeterminado en una CloudFormation pila, primero debe importarlo a una pila. Una vez que haya importado el bus de eventos predeterminado a una pila, podrá actualizar las propiedades del bus de eventos según lo desee.

- Siga estas instrucciones:

  ???.

Autorizar el uso EventBridge de un clave administrada por el cliente

Si utilizas una clave administrada por el cliente en tu cuenta para proteger el autobús de tu EventBridge evento, las políticas al respecto KMS key deben dar EventBridge permiso para usarlo en tu nombre. Estos permisos se proporcionan en una política clave.

EventBridge no necesita autorización adicional para usar la opción predeterminada Clave propiedad de AWS para proteger los EventBridge recursos de su AWS cuenta.

EventBridge requiere los siguientes permisos en un claves administradas por el cliente:

- `kms:DescribeKey`

  EventBridge requiere este permiso para recuperar el KMS key ARN del identificador de clave proporcionado y para comprobar que la clave es simétrica.

- `kms:GenerateDataKey`

  EventBridge requiere este permiso para generar una clave de datos como clave de cifrado para los datos del evento.

- `kms:Decrypt`

EventBridge requiere este permiso para descifrar la clave de datos cifrada y almacenada con los datos cifrados del evento.

EventBridge lo usa para hacer coincidir las reglas; los usuarios nunca tienen acceso a los datos.

El siguiente ejemplo de política clave proporciona los permisos necesarios:

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

Seguridad cuando se utiliza claves administradas por el cliente para el cifrado EventBridge del bus de eventos

Como práctica recomendada de seguridad, añada una `aws:SourceArn` clave o `kms:EncryptionContext:aws:events:event-bus:arn` condicionada a la política de AWS KMS claves. `aws:sourceAccount` La clave de condición IAM global ayuda a garantizar que se EventBridge utilice la clave KMS solo para el bus o la cuenta especificados.

En el siguiente ejemplo, se muestra cómo seguir esta práctica recomendada en su IAM política:

```
{
      "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"
        }
      }
```

Gestión del claves administradas por el cliente cifrado EventBridge del bus de eventos

Para garantizar que EventBridge siempre se mantenga el acceso a lo necesario clave administrada por el cliente:

- No elimine un clave administrada por el cliente hasta que esté seguro de que se han procesado todos los eventos cifrados con él.

  Cuando realice alguna de las siguientes operaciones, conserve el material clave anterior para asegurarse de que EventBridge puede seguir utilizándolo para los eventos cifrados anteriormente:

  - Rotación automática de claves
  - Rotación manual de claves
  - Actualización de un alias clave

  En general, si está pensando en eliminar una AWS KMS clave, desactívela primero y configure una CloudWatch alarma o un mecanismo similar para asegurarse de que nunca necesitará usar la clave para descifrar datos cifrados.
- No elimine la política de claves que proporciona EventBridge los permisos para usar la clave.

Otras consideraciones incluyen:

- Especifique claves administradas por el cliente los objetivos de las reglas, según corresponda.

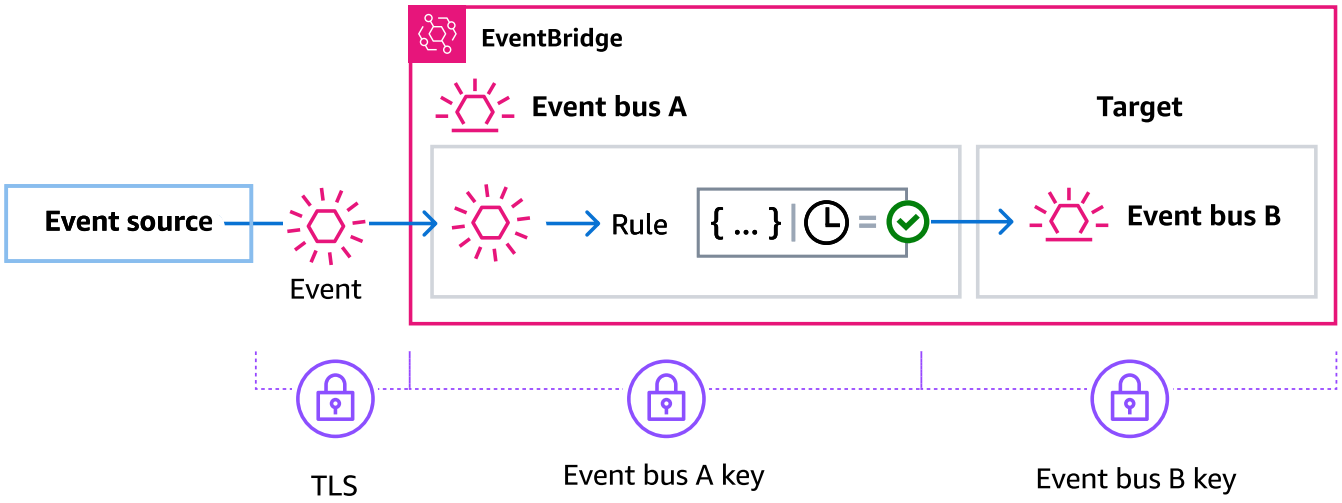
Cuando se EventBridge envía un evento a un destino de regla, el evento se envía mediante Transport Layer Security (TLS). Sin embargo, el cifrado que se aplique al evento cuando se almacene en el destino depende del cifrado que haya configurado en el propio destino.

Cifrado de eventos cuando un bus de eventos es el objetivo de la regla

Cuando se envía un evento personalizado o asociado a un bus de eventos, EventBridge cifra ese evento de acuerdo con la configuración de clave KMS de cifrado en reposo para ese bus de eventos, ya sea la predeterminada Clave propiedad de AWS o una clave administrada por el cliente, si se ha especificado alguna. Si un evento coincide con una regla, EventBridge cifra el evento con la configuración de claves KMS para ese bus de eventos hasta que el evento se envíe al destino de la regla, a menos que el destino de la regla sea otro bus de eventos.

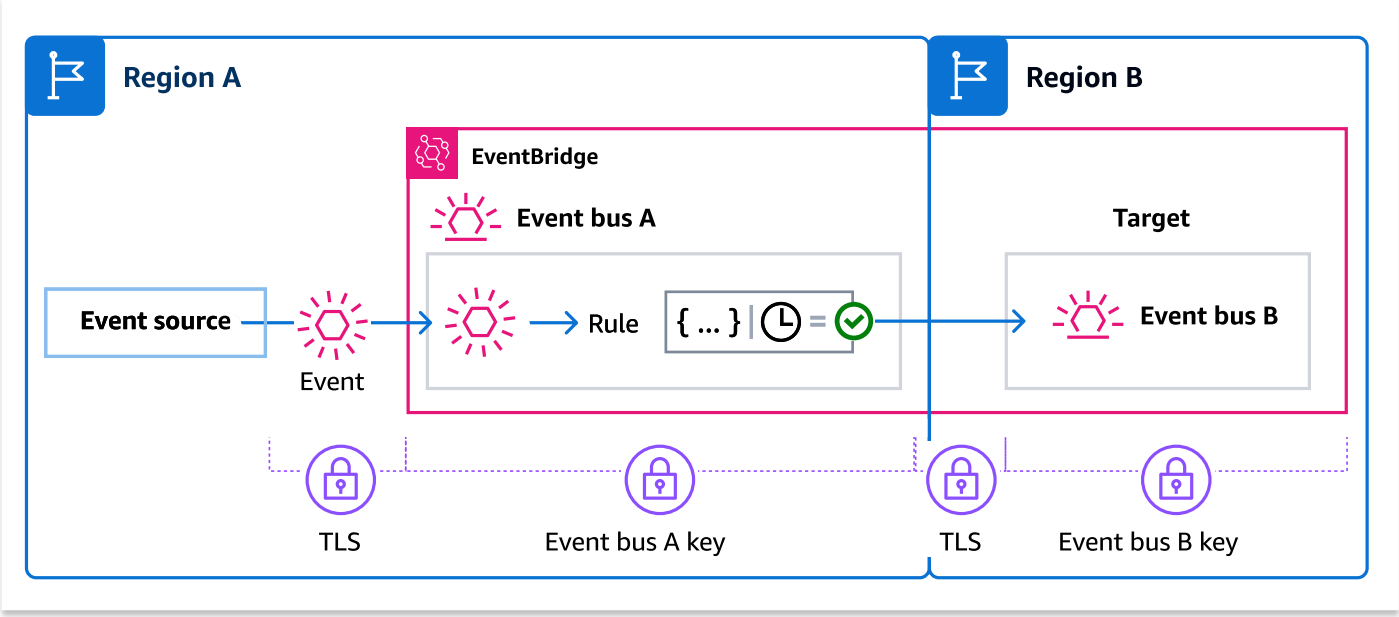
- Si el objetivo de una regla es otro bus de eventos en la misma AWS región:

  Si el bus de eventos de destino tiene una especificada clave administrada por el cliente, EventBridge cifra el evento con el bus clave administrada por el cliente de eventos de destino para su entrega.



- Si el objetivo de una regla es otro bus de eventos en una AWS región diferente:

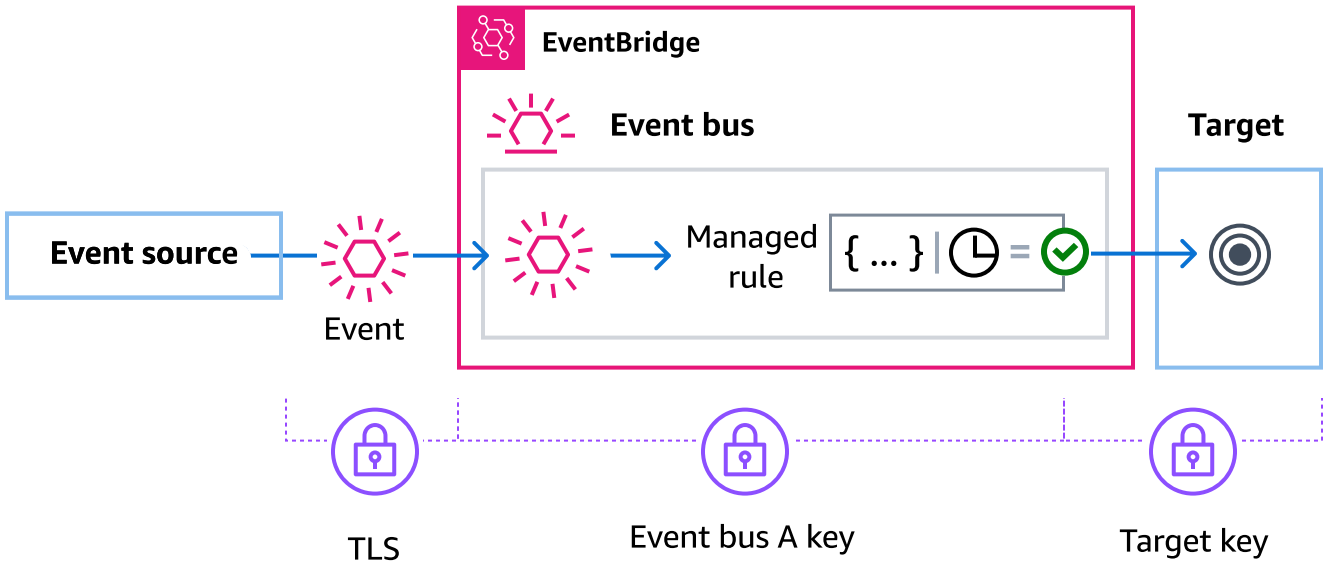
  EventBridge cifra el evento en reposo según la configuración de la clave KMS del primer bus de eventos. EventBridge usa TLS para enviar el evento al segundo bus de eventos de la región diferente, donde, a continuación, se cifra de acuerdo con la configuración de claves KMS especificada para el bus de eventos de destino.



Cifrado de eventos para reglas gestionadas

AWS los servicios pueden crear y administrar las reglas de bus de eventos en su AWS cuenta que son necesarias para determinadas funciones de esos servicios. Como parte de una regla gestionada, el AWS servicio puede especificar que se EventBridge utilice lo clave administrada por el cliente especificado para el objetivo de la regla. Esto le da la flexibilidad de especificar cuál clave administrada por el cliente usar en función del objetivo de la regla.

En estos casos, una vez que un evento personalizado o asociado coincide con la regla administrada, EventBridge utiliza el destino clave administrada por el cliente especificado por la regla administrada para cifrar el evento hasta que se envíe al destino de la regla. Esto es independiente de si el bus de eventos se ha configurado para usar el suyo propio clave administrada por el cliente para el cifrado. Este es el caso incluso si el destino de la regla gestionada es otro bus de eventos y ese bus de eventos tiene el suyo propio clave administrada por el cliente especificado para el cifrado. EventBridge sigue utilizando el destino clave administrada por el cliente especificado en la regla gestionada hasta que el evento se envíe a un destino que no sea un bus de eventos.



En los casos en los que el objetivo de la regla sea un bus de eventos de otra región, debe proporcionar una clave multirregional. El bus de eventos de la primera región cifra el evento mediante la clave administrada por el cliente especificada en la regla gestionada. A continuación, envía el evento al bus de eventos de destino de la segunda región. Ese bus de eventos debe poder seguir utilizándolo clave administrada por el cliente hasta que envíe el evento a su destino.

EventBridge contexto de cifrado del bus de eventos

Un contexto de cifrado es un conjunto de pares de clave-valor que contienen datos no secretos arbitrarios. Cuando se incluye un contexto de cifrado en una solicitud para cifrar datos, AWS KMS vincula criptográficamente el contexto de cifrado a los datos cifrados. Para descifrar los datos, es necesario pasar el mismo contexto de cifrado.

También puede utilizar el contexto de cifrado como condición para la autorización en políticas y concesiones.

Para los buses de eventos, EventBridge utiliza el mismo contexto de cifrado en todas las operaciones AWS KMS criptográficas. Si utiliza una clave gestionada por el cliente para proteger sus EventBridge recursos, puede utilizar el contexto de cifrado para identificar su uso KMS key en los registros y registros de auditoría. También aparece en texto sin formato en registros, como AWS CloudTrail y Amazon CloudWatch Logs.

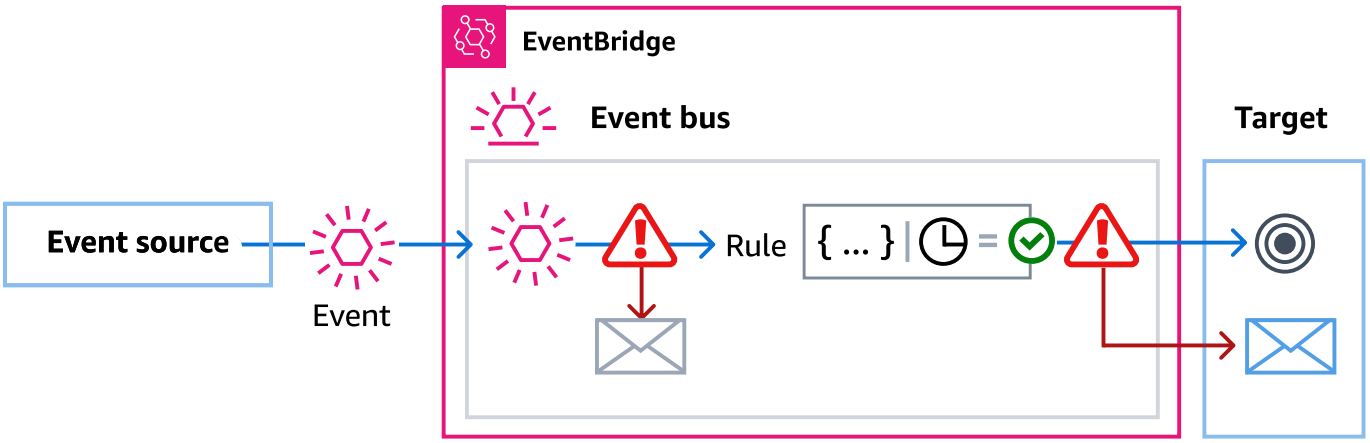
En sus solicitudes AWS KMS, EventBridge utiliza un contexto de cifrado con un único par clave-valor, que contiene el ARN del bus de eventos:

```
"encryptionContext": {
    "kms:EncryptionContext:aws:events:event-bus:arn": "event-bus-arn"
}
```

Utiliza colas de texto sin efecto para capturar los errores de eventos cifrados

Si configura el clave administrada por el cliente cifrado en un bus de eventos, le recomendamos que especifique una cola de mensajes sin salida (DLQ) para ese bus de eventos. EventBridge envía eventos personalizados y asociados a este DLQ si encuentra un error irrecuperable al procesar el evento en el bus de eventos. Un error irrecuperable es aquel en el que es necesaria una acción por parte del usuario para resolver el problema subyacente, como por ejemplo la inhabilitación o la ausencia del objeto especificado clave administrada por el cliente .

- Si se produce un error de cifrado o descifrado no recuperable mientras EventBridge se procesa el evento en el bus de eventos, el evento se envía al DLQ para el bus de eventos, si se especifica alguno.
- Si se produce un error de cifrado o descifrado no recuperable al intentar enviar el evento a un destino, el evento se envía al DLQ del destino, si se especifica alguno. EventBridge



Para obtener más información, incluidas las consideraciones a la hora de utilizar los DLQ e instrucciones sobre cómo configurar los permisos, consulte. ???

Descifrar eventos en colas de letra muerta EventBridge

Una vez que haya resuelto el problema subyacente que está causando un error irrecuperable, puede procesar los eventos enviados al bus de eventos o al DLQ de destino. En el caso de los eventos cifrados, primero debe descifrar el evento para poder procesarlo.

El siguiente ejemplo muestra cómo descifrar un evento que se EventBridge ha enviado a un bus de eventos o a un DLQ de destino.

```
// 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);
```

Políticas basadas en etiquetas

En Amazon EventBridge puede utilizar políticas basadas en etiquetas para controlar el acceso a los recursos.

Por ejemplo, puede restringir el acceso a los recursos que incluyen una etiqueta con la clave `environment` y el valor `production`. Las siguientes políticas de ejemplo deniegan a los recursos con esta etiqueta la capacidad de crear, eliminar o modificar etiquetas, reglas o buses de eventos para los recursos que se han etiquetado como `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"}
            }
        }
    ]
}
```

Para obtener más información sobre el etiquetado, consulte lo siguiente.

- EventBridge Etiquetas de Amazon
- Control del acceso mediante etiquetas de IAM

Amazon EventBridge y AWS Identity and Access Management

Para acceder a Amazon EventBridge, necesitas credenciales que AWS puedas usar para autenticar tus solicitudes. Sus credenciales deben tener permisos para acceder a los recursos de AWS como, por ejemplo, para recuperar datos de evento desde otros recursos de AWS. En las siguientes secciones, se proporcionan detalles sobre cómo puedes usar AWS Identity and Access Management(IAM) y cómo ayudarte EventBridge a proteger tus recursos controlando quién puede acceder a ellos.

Temas

Autenticación

Puede obtener acceso a AWS con los siguientes tipos de identidades:

- Usuario raíz de la cuenta de AWS: cuando se registra en AWS, proporciona una dirección de email y una contraseña asociada a su cuenta. Estas son las credenciales raíz y proporcionan acceso completo a todos los recursos de AWS.

> **Important**
>
> Por motivos de seguridad, le recomendamos que solamente utilice las credenciales raíz para crear un administrador, que es un usuario de IAM con todos los permisos necesarios para administrar la cuenta. A continuación, puede utilizar este administrador para crear otros usuarios y roles con permisos limitados. Para obtener más información, consulte

Prácticas recomendadas de IAM y Creación de un grupo y usuario administrador en la Guía del usuario de IAM.

- Usuario de IAM: un usuario de IAM es una identidad de tu cuenta que tiene permisos específicos, por ejemplo, el permiso para enviar datos de eventos a un objetivo. EventBridge Puede utilizar estas credenciales para iniciar sesión en IAM para proteger páginas web de AWS, como la AWS Management Console, los foros de debate de AWS o el centro de AWS Support.

  Además de las credenciales de inicio de sesión, puede generar claves de acceso para cada usuario. Puede utilizar estas claves al acceder a los servicios de AWS de manera programática para firmar criptográficamente su solicitud, ya sea a través de uno de los SDK o mediante la AWS Command Line Interface (AWS CLI). Si no utiliza las herramientas de AWS, debe firmar la solicitud usted mismo con Signature Version 4, un protocolo para autenticar solicitudes entrantes de la API. Para obtener más información acerca de la autenticación de solicitudes, consulte Proceso de firma Signature Version 4 en la Referencia general de Amazon Web Services.
- Rol de IAM: un rol de IAM es otra identidad de IAM que puede crear en la cuenta y que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una persona determinada. Un rol de IAM le permite obtener claves de acceso temporal para acceder a los recursos y servicios de AWS. Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:
  - Acceso de usuarios federados: en lugar de crear un usuario, puede usar identidades de AWS Directory Service, el directorio de usuarios de la compañía o un proveedor de identidades (IdP) web. A estas identidades se les llama usuarios federados. AWS asigna un rol a un usuario federado cuando el usuario solicita acceso a través de un proveedor de identidad. Para obtener más información acerca de los usuarios federados, consulte Usuarios federados y roles en la Guía del usuario de IAM.
  - Acceso entre cuentas: puede utilizar un rol de IAM en su cuenta para conceder permiso a otra cuenta para que acceda a los recursos de su cuenta. Para ver un ejemplo, consulte Tutorial: Delegación del acceso entre cuentas de AWS mediante roles de IAM en la Guía del usuario de IAM.
  - Acceso a servicios de AWS: puede utilizar un rol de IAM de su cuenta para conceder permiso a un servicio de AWS a fin de acceder a los recursos de su cuenta. Por ejemplo, puede crear un rol que permita a Amazon Redshift cargar datos almacenados en un bucket de Amazon S3 en un clúster de Amazon Redshift. Para obtener más información, consulte Creación de un rol para delegarle permisos a un servicio de AWS en la Guía del usuario de IAM.

- Aplicaciones que se ejecutan en Amazon EC2: en el caso de las aplicaciones de Amazon EC2 EventBridge a las que es necesario acceder, puede almacenar las claves de acceso en la instancia EC2 o utilizar una función de IAM para gestionar las credenciales temporales. Para asignar un rol de AWS a una instancia de EC2, cree un perfil de instancia asociado a la instancia. Un perfil de instancia contiene el rol y proporciona credenciales temporales a aplicaciones que se ejecutan en la instancia de EC2. Para obtener más información, consulte Uso de roles para aplicaciones en Amazon EC2 en la Guía del usuario de IAM.

Control de acceso

Para crear EventBridge recursos o acceder a ellos, necesita credenciales y permisos válidos. Por ejemplo, para invocar destinos de AWS Lambda, Amazon Simple Notification Service (Amazon SNS) y Amazon Simple Queue Service (Amazon SQS), debe tener permisos para estos servicios.

Administrar permisos de acceso para los recursos de Amazon EventBridge

Usted administra el acceso a los recursos de EventBridge como reglas o eventos utilizando políticas basadas en identidades o basadas en recursos.

Recursos de EventBridge

Los recursos y los subrecursos de EventBridge tienen nombres de recurso de Amazon (ARN) únicos asociados a ellos. Los ARN se utilizan en EventBridge para crear patrones de eventos. Para obtener más información sobre los ARN, consulte Nombres de recursos de Amazon (ARN) y espacios de nombres de servicios de AWS en la Referencia general de Amazon Web Services.

Para obtener una lista de las operaciones que EventBridge proporciona para trabajar con recursos, consulte Referencia de permisos de Amazon EventBridge.

> **Note**
>
> La mayoría de los servicios de AWS tratan el carácter de dos puntos (`:`) o la barra diagonal (`/`) como el mismo carácter en los ARN. Sin embargo, EventBridge utiliza una coincidencia exacta en las reglas y los patrones de eventos. Asegúrese de usar los caracteres de ARN correctos al crear patrones de eventos para que coincidan con la sintaxis de ARN en el evento que desee asignar.

En la tabla siguiente, se muestran los recursos de EventBridge.

Tipo de recurso	Formato de ARN
Archivado	`arn:aws:events:` *`region`*`:`*`account`*`:archive/` *`archive-name`*
Reproducción	`arn:aws:events:` *`region`*`:`*`account`*`:replay/`*`replay-name`*
Regla	`arn:aws:events:` *`region`*`:`*`account`*`:rule/[event-bus-name]/`*`rule-name`*
Bus de eventos	`arn:aws:events:` *`region`*`:`*`account`*`:event-bus/` *`event-bus-name`*

Tipo de recurso	Formato de ARN
Todos los recursos de EventBridge	`arn:aws:events:*`
Todos los recursos de EventBridge que pertenecen a la cuenta especificada en la región indicada	`arn:aws:events: region:account:*`

Los siguientes ejemplos muestran cómo indicar una regla específica (*myRule*) en su instrucción usando su ARN.

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/myRule"
```

Para especificar todas las reglas que pertenezcan a una cuenta específica mediante el comodín asterisco (*) del modo siguiente.

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/*"
```

Para especificar todos los recursos, o si una acción de API específica no admite ARN, utilice el comodín asterisco (*) en el elemento `Resource` del siguiente modo.

```
"Resource": "*"
```

Para especificar varios recursos o `PutTargets` en una única instrucción, separe sus ARN con comas, tal y como se indica a continuación.

```
"Resource": ["arn1", "arn2"]
```

Propiedad del recurso

Una cuenta es la propietaria de los recursos de la cuenta, independientemente de quién los haya creado. El propietario de los recursos es la cuenta de la entidad principal, el usuario raíz de la cuenta, un usuario o un rol de IAM que autentica la solicitud que crea el recurso. Los siguientes ejemplos ilustran cómo funciona:

- Si utiliza las credenciales de usuario raíz de su cuenta para crear una regla, la cuenta será la propietaria del recurso de EventBridge.
- Si crea un usuario en su cuenta y le concede permisos para crear recursos de EventBridge, el usuario podrá crear recursos de EventBridge. Sin embargo, su cuenta, a la que pertenece el usuario, será la propietaria de los recursos de EventBridge.
- Si crea un rol de IAM en su cuenta con permisos para crear recursos de EventBridge, cualquier persona que pueda asumir el rol podrá crear recursos de EventBridge. Su cuenta, a la que pertenece el rol, será la propietaria de los recursos de EventBridge.

Administrar el acceso a los recursos

Una política de permisos describe quién tiene acceso a qué. En la siguiente sección se explican las opciones disponibles para crear políticas de permisos.

> **Note**
>
> En esta sección se explica el uso de IAM en el contexto de EventBridge. No se proporciona información detallada sobre el servicio de IAM. Para ver la documentación completa de IAM, consulte What is IAM? (¿Qué es IAM?) en la Guía del usuario de IAM. Para obtener más información sobre la sintaxis y descripciones de las políticas de IAM, consulte Referencia de políticas de IAM en la Guía del usuario de IAM.

Las políticas asociadas a una identidad de IAM se denominan políticas basadas en identidad (políticas de IAM) y las políticas asociadas a un recurso se denominan políticas basadas en recursos. En EventBridge, puede utilizar políticas basadas en identidades (políticas de IAM) y políticas basadas en recursos.

Temas

- Políticas basadas en identidades (políticas de IAM)
- Políticas basadas en recursos (políticas de IAM)

Políticas basadas en identidades (políticas de IAM)

Puede asociar políticas a identidades de IAM. Por ejemplo, puede hacer lo siguiente:

- Asociar una política de permisos a un usuario o un grupo en su cuenta: para conceder permiso a un usuario a fin de ver las reglas en la consola de Amazon CloudWatch, puede asociar una política de permisos a un usuario o a un grupo al que pertenezca el usuario.
- Adjuntar una política de permisos a un rol (conceder permisos para cuentas cruzadas): puede adjuntar una política de permisos basada en identidades a un rol de IAM para conceder permisos para cuentas cruzadas. Por ejemplo, el administrador de la cuenta A puede crear un rol para concederles permisos a las cuentas cruzadas a otra cuenta B o a un servicio de AWS, tal y como se indica a continuación:

  1. El administrador de la Cuenta A crea un rol de IAM y adjunta una política de permisos al rol que concede permiso sobre los recursos de la Cuenta A.
  2. El administrador de la cuenta A asocia una política de confianza al rol que identifica la cuenta B como la entidad principal que puede asumir el rol.
  3. El administrador de la Cuenta B puede delegar permisos para asumir el rol de cualquier usuario de la Cuenta B. De este modo, los usuarios de la Cuenta B podrán crear recursos en la Cuenta A u obtener acceso a ellos. La entidad principal de la política de confianza también puede ser la entidad principal de un servicio de AWS si desea conceder a un servicio de AWS el permiso necesario para asumir el rol.

  Para obtener más información sobre el uso de IAM para delegar permisos, consulte Administración de accesos en la Guía del usuario de IAM.

Puede crear políticas de IAM específicas para restringir las llamadas y los recursos a los que los usuarios de su cuenta tienen acceso y, a continuación, asociar esas políticas a usuarios de . Para obtener más información sobre cómo crear roles de IAM y para ver instrucciones de políticas de IAM de ejemplo para EventBridge, consulte Administrar permisos de acceso para los recursos de Amazon EventBridge.

Políticas basadas en recursos (políticas de IAM)

Cuando se ejecuta una regla en EventBridge, se invocan todos los destinos asociados a la regla, es decir, se invocan las funciones AWS Lambda, que publican en los temas de Amazon SNS o que transfieren el evento a los flujos de Amazon Kinesis. Para realizar llamadas a la API en los recursos que posee, EventBridge necesita los permisos adecuados. Para los recursos de Lambda, Amazon SNS y Amazon SQS, EventBridge utiliza políticas basadas en recursos. Para los flujos de Kinesis, EventBridge utiliza roles de IAM.

Para obtener más información sobre cómo crear roles de IAM; y para ver instrucciones de políticas basadas en recursos de ejemplo para EventBridge, consulte Uso de políticas basadas en recursos para Amazon EventBridge.

Especificar elementos de las políticas: acciones, efectos y entidades principales

Para cada recurso de EventBridge, EventBridge define un conjunto de operaciones de API. Para conceder permisos para estas operaciones de API, EventBridge define un conjunto de acciones que usted puede especificar en una política. Algunas operaciones de API requieren permisos para más de una acción para poder realizar la operación de API. Para obtener más información sobre los recursos y las operaciones de API, consulte Recursos de EventBridge y Referencia de permisos de Amazon EventBridge.

A continuación, se indican los elementos básicos de la política:

- Recurso: utilice un nombre de recurso de Amazon (ARN) para identificar el recurso al que se aplica la política. Para obtener más información, consulte Recursos de EventBridge.
- Acción: utilice palabras clave para identificar las operaciones del recurso que desea permitir o denegar. Por ejemplo, cuando se concede el permiso `events:Describe`, el usuario puede realizar la operación `Describe`.
- Efecto: especifique permitir o denegar. Si no concede acceso de forma explícita (permitir) a un recurso, el acceso se deniega. También puede denegar explícitamente el acceso a un recurso para asegurarse de que un usuario no pueda obtener acceso a él, aunque otra política le conceda acceso.
- Entidad principal: en las políticas basadas en identidades (políticas de IAM), el usuario al que se asocia esta política es la entidad principal implícita. Para las políticas basadas en recursos, debe especificar el usuario, la cuenta, el servicio u otra entidad que desee que reciba permisos (se aplica solo a las políticas basadas en recursos).

Para obtener más información acerca de la sintaxis y las descripciones de las políticas de IAM, consulte Referencia de políticas JSON de IAM en la Guía del usuario de IAM.

Para obtener información sobre las acciones de la API de EventBridge y los recursos a los que se aplican, consulte Referencia de permisos de Amazon EventBridge.

Especificar las condiciones de una política

Al conceder permisos, puede utilizar el lenguaje de la política de acceso para especificar las condiciones en las que se debe aplicar una política. Por ejemplo, es posible que desee que solo se aplique una política después de una fecha específica. Para obtener más información sobre cómo especificar condiciones en un lenguaje de política, consulte Condition en la Guía del usuario de IAM.

Para definir condiciones, se usan claves de condición. Hay claves de condición de AWS y claves específicas de EventBridge que pueden utilizar cuando corresponda. Para ver una lista completa de claves de AWS, consulte Claves disponibles para las condiciones en la Guía del usuario de IAM. Para obtener una lista completa de las claves específicas de EventBrige, consulte Usar condiciones de las políticas de IAM para control de acceso preciso.

Uso de políticas basadas en la identidad (políticas de IAM) para Amazon EventBridge

Las políticas basadas en identidades son políticas de permisos que se adjuntan a identidades de IAM.

Temas

AWS políticas gestionadas para EventBridge

AWS aborda muchos casos de uso comunes al proporcionar políticas de IAM independientes que son creadas y administradas por. AWS Las políticas administradas o predefinidas otorgan los permisos necesarios para casos de uso comunes, por lo que no es necesario investigar qué permisos se necesitan. Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Las siguientes políticas AWS gestionadas que puede adjuntar a los usuarios de su cuenta son específicas de: EventBridge

- AmazonEventBridgeFullAccess— Otorga acceso completo a EventBridge, incluidos EventBridge Pipes, EventBridge Schemas y EventBridge Scheduler.
- AmazonEventBridgeReadOnlyAccess— Otorga acceso de solo lectura a EventBridge, incluidos EventBridge Pipes, Schemas y Scheduler EventBridge . EventBridge

AmazonEventBridgeFullAccess política

La AmazonEventBridgeFullAccess política concede permisos para usar todas EventBridge las acciones, así como los siguientes permisos:

- `iam:CreateServiceLinkedRole`— EventBridge requiere este permiso para crear el rol de servicio en tu cuenta para los destinos de la API. Este permiso solo otorga al servicio de IAM permisos para crear un rol en la cuenta específico para los destinos de la API.

- `iam:PassRole`— EventBridge requiere este permiso para pasar un rol de invocación EventBridge al objetivo de una regla.
- Permisos de Secrets Manager: EventBridge requiere estos permisos para administrar los secretos de tu cuenta cuando proporcionas credenciales a través del recurso de conexión para autorizar los destinos de la API.

En el siguiente JSON se muestra la AmazonEventBridgeFullAccess política.

```
{
    "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**
>
> La información de esta sección también se aplica a la política `CloudWatchEventsFullAccess`. Sin embargo, se recomienda encarecidamente que utilice Amazon EventBridge en lugar de Amazon CloudWatch Events.

AmazonEventBridgeReadOnlyAccess política

La AmazonEventBridgeReadOnlyAccess política otorga permisos para usar todas las EventBridge acciones de lectura.

En el siguiente JSON se muestra la AmazonEventBridgeReadOnlyAccess política.

```
{
    "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**
>
> La información de esta sección también se aplica a la política `CloudWatchEventsReadOnlyAccess`. Sin embargo, se recomienda encarecidamente que utilice Amazon EventBridge en lugar de Amazon CloudWatch Events.

EventBridge Políticas gestionadas específicas del esquema

Un esquema define la estructura de los eventos a los que se envían. EventBridge EventBridge proporciona esquemas para todos los eventos generados por los AWS servicios. Están disponibles las siguientes políticas AWS gestionadas específicas de EventBridge los esquemas:

- AmazonEventBridgeSchemasServiceRolePolicy
- AmazonEventBridgeSchemasFullAccess

- AmazonEventBridgeSchemasReadOnlyAccess

EventBridge Políticas gestionadas específicas del programador

Amazon EventBridge Scheduler es un programador sin servidor que le permite crear, ejecutar y gestionar tareas desde un servicio gestionado centralizado. Para ver las políticas AWS administradas específicas de EventBridge Scheduler, consulte las políticas AWS administradas de Scheduler en la Guía del usuario de EventBridge Scheduler. EventBridge

EventBridge Políticas gestionadas específicas de Pipes

Amazon EventBridge Pipes conecta las fuentes de eventos con los objetivos. Reduce la necesidad de conocimientos especializados y códigos de integración a la hora de desarrollar arquitecturas basadas en eventos. Esto ayuda a garantizar la coherencia en todas las aplicaciones de su empresa. Están disponibles las siguientes políticas AWS gestionadas específicas para EventBridge Pipes:

- AmazonEventBridgePipesFullAccess

  Proporciona acceso completo a Amazon EventBridge Pipes.

  > **Note**
  >
  > Esta política establece lo `iam:PassRole` siguiente: EventBridge Pipes necesita este permiso para transferir un rol de invocación EventBridge a fin de crear e iniciar canalizaciones.

- AmazonEventBridgePipesReadOnlyAccess

  Proporciona acceso de solo lectura a Amazon EventBridge Pipes.

- AmazonEventBridgePipesOperatorAccess

  Proporciona acceso de solo lectura y mediante operador (es decir, la capacidad de detener y empezar a ejecutar Pipes) a Amazon EventBridge Pipes.

Roles de IAM; para enviar eventos

Para retransmitir eventos a los objetivos, EventBridge necesita una función de IAM.

Para crear un rol de IAM para enviar eventos a EventBridge

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Para crear un rol de IAM, siga los pasos de la Guía del usuario de IAM sobre cómo crear un rol para delegar permisos a un AWS servicio. Cuando siga los pasos, haga lo siguiente:
   - En Nombre del rol, utilice un nombre que sea exclusivo dentro de su cuenta.
   - En Select Role Type, elija AWS Service Roles y, a continuación, Amazon EventBridge. Esto otorga EventBridge permisos para asumir el rol.
   - En Attach Policy, elija AmazonEventBridgeFullAccess.

También puedes crear tus propias políticas de IAM personalizadas para permitir permisos para EventBridge acciones y recursos. Puede asociar estas políticas personalizadas a los grupos o usuarios de IAM que requieran esos permisos. Para obtener más información acerca de las políticas de IAM, consulte Descripción general de las políticas de IAM en la Guía del usuario de IAM. Para obtener más información sobre cómo administrar y crear políticas de IAM personalizadas, consulte Administrar políticas de IAM en la Guía del usuario de IAM.

Permisos necesarios para acceder EventBridge a los objetivos mediante funciones de IAM

EventBridge Los objetivos suelen requerir funciones de IAM que concedan permiso para EventBridge invocar el objetivo. A continuación se muestran algunos ejemplos de varios AWS servicios y objetivos. Para otros, utilice la EventBridge consola para crear una regla y crear un nuevo rol que se creará con una política con permisos bien definidos y preconfigurados.

Amazon SQS, Amazon SNS, CloudWatch Lambda EventBridge , Logs y los destinos de bus no utilizan funciones y los EventBridge permisos deben concederse mediante una política de recursos. Los objetivos de API Gateway pueden usar políticas de recursos o roles de IAM.

Si el destino es un destino de API, el rol que especifique debe incluir la siguiente política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "events:InvokeApiDestination" ],
            "Resource": [ "arn:aws:events:::api-destination/*" ]
```

```
        }
     ]
}
```

Si el destino es un flujo de Kinesis, el rol utilizado para enviar datos de eventos a dicho destino debe incluir la siguiente política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

Si el destino es el comando ejecutar de Systems Manager y especifica uno o varios valores de `InstanceIds` para el comando, el rol que especifique debe incluir la siguiente política.

```
{
     "Version": "2012-10-17",
     "Statement": [
          {
               "Action": "ssm:SendCommand",
               "Effect": "Allow",
               "Resource": [
                    "arn:aws:ec2:region:accountId:instance/instanceIds",
                    "arn:aws:ssm:region:*:document/documentName"
               ]
          }
     ]
}
```

Si el destino es el comando ejecutar de Systems Manager y especifica una o varias etiquetas para el comando, el rol que especifique debe incluir la siguiente política.

```
{
```

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

Si el destino es una máquina de AWS Step Functions estados, la función que especifique debe incluir la siguiente política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "states:StartExecution" ],
            "Resource": [ "arn:aws:states:*:*:stateMachine:*" ]
        }
     ]
}
```

Si el destino es una tarea de Amazon ECS, el rol que especifique debe incluir la siguiente política.

```
{
```

```
    "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 siguiente política permite que los objetivos integrados EventBridge realicen acciones de Amazon EC2 en su nombre. Debe utilizarla para AWS Management Console crear reglas con objetivos integrados.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TargetInvocationAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:RebootInstances",
                "ec2:StopInstances",
```

```
                "ec2:TerminateInstances",
                "ec2:CreateSnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

La siguiente política permite EventBridge retransmitir eventos a las transmisiones de Kinesis de su cuenta.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KinesisAccess",
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

Ejemplo de política administrada por el cliente: uso del etiquetado para controlar el acceso a las reglas

El siguiente ejemplo muestra una política de usuario que concede permisos para EventBridge realizar acciones. Esta política funciona cuando usas la EventBridge API, AWS los SDK o el AWS CLI.

Puedes conceder a los usuarios el acceso a EventBridge reglas específicas y, al mismo tiempo, impedir que accedan a otras reglas. Para ello, etiquete ambos conjuntos de reglas y, a continuación, utilice políticas de IAM para hacer referencia a esas etiquetas. Para obtener más información sobre el etiquetado de EventBridge recursos, consulte EventBridge Etiquetas de Amazon.

Puede conceder una política de IAM a un usuario para permitir el acceso únicamente a las reglas con una determinada etiqueta. Usted elige las reglas a las que desea conceder el acceso etiquetándolas con esa etiqueta concreta. Por ejemplo, la siguiente política concede a un usuario acceso a reglas con el valor de `Prod` para la clave de etiqueta `Stack`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "events:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Stack": "Prod"
                }
            }
        }
    ]
}
```

Para obtener más información acerca del uso de instrucciones de política de IAM, consulte Control del acceso mediante las políticas en la Guía del usuario de IAM.

Amazon EventBridge actualiza las políticas AWS gestionadas

Consulta los detalles sobre las actualizaciones de las políticas AWS gestionadas EventBridge desde que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbase a la fuente RSS de la página del historial del EventBridge documento.

Cambio	Descripción	Fecha
AmazonEventBridgeFullAccess: política actualizada	AWS GovCloud (US) Regions únicamente El siguiente permiso no está incluido, ya que no se utiliza: • `iam:CreateServiceLinkedRole` permiso para EventBridge Schema Registry	9 de mayo de 2024

Cambio	Descripción	Fecha
AmazonEventBridgeSchemasFullAccess: política actualizada	AWS GovCloud (US) Regions únicamente El siguiente permiso no está incluido, ya que no se utiliza: • `iam:CreateServiceLinkedRole` permiso para EventBridge Schema Registry	9 de mayo de 2024
AmazonEventBridgePipesFullAccess— Se agregó una nueva política	EventBridge se agregó una política administrada para obtener todos los permisos de uso de EventBridge Pipes.	1 de diciembre de 2022
AmazonEventBridgePipesReadOnlyAccess— Se ha añadido una nueva política	EventBridge se agregó una política administrada de permisos para ver los recursos de información de EventBridge Pipes.	1 de diciembre de 2022
AmazonEventBridgePipesOperatorAccess— Se agregó una nueva política	EventBridge se agregó una política administrada de permisos para ver la información de EventBridge las tuberías, así como para iniciar y detener las tuberías en funcionamiento.	1 de diciembre de 2022
AmazonEventBridgeFullAccess: actualización de una política actual	EventBridge actualizó la política para incluir los permisos necesarios para usar EventBridge las funciones de Pipes.	1 de diciembre de 2022

Cambio	Descripción	Fecha
AmazonEventBridgeReadOnlyAccess: actualización de una política actual	EventBridge se agregaron los permisos necesarios para ver los recursos de información de EventBridge Pipes. Se agregaron las siguientes acciones: • `pipes:DescribePipe` • `pipes:ListPipes` • `pipes:ListTagsForResource`	1 de diciembre de 2022
CloudWatchEventsReadOnlyAccess: actualización de una política actual	Actualizado para que coincida AmazonEventBridgeReadOnlyAccess.	1 de diciembre de 2022
CloudWatchEventsFullAccess: actualización de una política actual	Actualizado para que coincida AmazonEventBridgeFullAccess.	1 de diciembre de 2022

Cambio	Descripción	Fecha
AmazonEventBridgeFullAccess: actualización de una política actual	EventBridge actualizó la política para incluir los permisos necesarios para usar los esquemas y las funciones del programador. Se agregaron los siguientes permisos: • EventBridge Acciones del registro de esquemas • EventBridge Acciones del programador • `iam:CreateServiceLinkedRole` permiso para EventBridge Schema Registry • `iam:PassRole` permiso para EventBridge Scheduler	10 de noviembre de 2022

Cambio	Descripción	Fecha
AmazonEventBridgeReadOnlyAccess: actualización de una política actual	EventBridge se agregaron los permisos necesarios para ver los recursos de información del esquema y del programador. Se agregaron las siguientes acciones: • `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 de noviembre de 2022

Cambio	Descripción	Fecha
	• `schemas:ListTagsForResource` • `schemas:SearchSchemas` • `scheduler:GetSchedule` • `scheduler:GetScheduleGroup` • `scheduler:ListSchedules` • `scheduler:ListScheduleGroups` • `scheduler:ListTagsForResource`	
AmazonEventBridgeReadOnlyAccess: actualización de una política actual	EventBridge se agregaron los permisos necesarios para ver la información del punto final. Se agregaron las siguientes acciones: • `events:ListEndpoints` • `events:DescribeEndpoint`	7 de abril de 2022

Cambio	Descripción	Fecha
AmazonEventBridgeReadOnlyAccess: actualización de una política actual	EventBridge se agregaron los permisos necesarios para ver la información de conexión y destino de la API. Se agregaron las siguientes acciones: • `events:DescribeConnection` • `events:ListConnections` • `events:DescribeApiDestination` • `events:ListApiDestinations`	4 de marzo de 2021

Cambio	Descripción	Fecha
AmazonEventBridgeFullAccess: actualización de una política actual	EventBridge actualizó la política para incluir `iam:CreateServiceLinkedRole` los AWS Secrets Manager permisos necesarios para usar los destinos de la API. Se agregaron las siguientes acciones: • `secretsmanager:CreateSecret` • `secretsmanager:UpdateSecret` • `secretsmanager:DeleteSecret` • `secretsmanager:GetSecretValue` • `secretsmanager:PutSecretValue`	4 de marzo de 2021
EventBridge comenzó a rastrear los cambios	EventBridge comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	4 de marzo de 2021

Uso de políticas basadas en recursos para Amazon EventBridge

Cuando se ejecuta una regla en EventBridge, se invocan todos los destinos asociados a la regla. Las reglas pueden invocar funciones de AWS Lambda, publicar en temas de Amazon SNS o retransmitir el evento en flujos de Kinesis. Para realizar llamadas a la API contra los recursos que usted posee, EventBridge necesita los permisos adecuados. Para los recursos de Lambda, Amazon SNS, Amazon SQS y Registros de Amazon CloudWatch, EventBridge utiliza políticas basadas en recursos. Para los flujos de Kinesis, EventBridge utiliza políticas basadas en identidades.

Puede usar AWS CLI para agregar permisos a sus destinos. Para obtener más información sobre cómo instalar y configurar la AWS CLI, consulte Configuración inicial de la AWS Command Line Interface en la Guía del usuario de AWS Command Line Interface.

Temas

Permisos de Amazon API Gateway

Para invocar su punto de conexión de Amazon API Gateway mediante una regla de EventBridge, agregue el siguiente permiso a la política de su punto de conexión de 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"
            ]
        }
    ]
}
```

Permisos de Registros de CloudWatch

Cuando Registros de CloudWatch es el destino de una regla, EventBridge crea flujos de registro y Registros de CloudWatch almacena el texto de los eventos como entradas de registro. Para permitir que EventBridge cree el flujo de registro y registre los eventos, Registros de CloudWatch debe incluir una política basada en recursos que permita a EventBridge escribir en Registros de CloudWatch.

Si utiliza la AWS Management Console para agregar Registros de CloudWatch como el destino de una regla, la política basada en recursos se crea automáticamente. Si utiliza la AWS CLI para agregar el destino y la política aún no existe, debe crearla.

Este ejemplo permite a EventBridge escribir en todos los grupos de registro que tienen nombres que empiezan por `/aws/events/`. Si utiliza una política de nomenclatura diferente para estos tipos de registros, ajuste la política en consecuencia.

```
{
    "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"
}
```

Para obtener más información, consulte PutResourcePolicy en la Guía de referencia de la API de Registros de CloudWatch.

Permisos de AWS Lambda

Para invocar la función de AWS Lambda utilizando una regla de EventBridge, agregue el siguiente permiso a la política de su función de 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"
}
```

Para agregar los permisos anteriores que permiten a EventBridge invocar funciones de Lambda mediante la AWS CLI

- En el símbolo del sistema, escriba el siguiente comando.

```
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"
```

Para obtener más información sobre la configuración de permisos que permiten a EventBridge invocar funciones de Lambda, consulte AddPermission y Uso de Lambda con eventos programados en la Guía para desarrolladores de AWS Lambda.

Permisos de Amazon SNS

Para permitir a EventBridge publicar en un tema de Amazon SNS, utilice los comandos `aws sns get-topic-attributes` y `aws sns set-topic-attributes`.

> **Note**
>
> No puede usar bloques `Condition` en las políticas de temas de Amazon SNS para EventBridge.

Para agregar permisos que permitan a EventBridge publicar temas de SNS

1. Utilice el siguiente comando para ver una lista de los atributos de un tema de SNS.

   ```
   aws sns get-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name"
   ```

   El siguiente ejemplo muestra el resultado de un tema de SNS nuevo.

   ```
   {
       "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. Utilice un conversor de JSON a cadena para convertir la siguiente instrucción en una cadena.

```
{
  "Sid": "PublishEventsToMyTopic",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "arn:aws:sns:region:account-id:topic-name"
}
```

Después de convertir la instrucción en una cadena, debe parecerse al siguiente ejemplo.

```
{\"Sid\":\"PublishEventsToMyTopic\",\"Effect\":\"Allow\",\"Principal\":
{\"Service\":\"events.amazonaws.com\"},\"Action\":\"sns:Publish\",\"Resource\":
\"arn:aws:sns:region:account-id:topic-name\"}
```

3. Agregue la cadena que creó en el paso anterior a la colección `"Statement"` incluida en el atributo `"Policy"`.
4. Para especificar la política nueva, use el 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\"}]}"
```

Para obtener más información, consulte la acción SetTopicAttributes en la Referencia de la API de Amazon Simple Notification Service.

Permisos de Amazon SQS

Para permitir que una regla de EventBridge invoque una cola de Amazon SQS, utilice los comandos `aws sqs get-queue-attributes` y `aws sqs set-queue-attributes`.

Si la política de la cola de SQS está vacía, primero debe crear una política y, a continuación, agregarle la instrucción de permisos. Una nueva cola de SQS tiene una política vacía.

Si la cola de SQS ya tiene una política, debe copiar la política original y combinarla con una nueva instrucción para agregarle la instrucción de permisos.

Para agregar permisos que permitan a las reglas de EventBridge invocar una cola de SQS

1. Para enumerar los atributos de la cola de SQS. En el símbolo del sistema, escriba el siguiente comando.

```
aws sqs get-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attribute-names Policy
```

2. Agregue la siguiente instrucción.

```
{
      "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. Utilice un conversor de JSON a cadena para convertir la instrucción anterior en una cadena. Después de convertir la política en una cadena, debe parecerse al siguiente ejemplo.

```
{\"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. Cree un archivo denominado `set-queue-attributes.json` con el siguiente contenido.

```
{
    "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. Establezca el atributo de política mediante el archivo `set-queue-attributes.json` que acaba de crear como entrada, tal y como se muestra en el siguiente comando.

```
aws sqs set-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attributes file://set-queue-attributes.json
```

Para obtener más información, consulte Ejemplos de políticas de Amazon SQS en la Guía para desarrolladores de Amazon Simple Queue Service.

Aspectos específicos de EventBridge Pipes

EventBridge Pipes no admite políticas basadas en recursos y no tiene API que admitan políticas basadas en recursos.

Prevención del suplente confuso entre servicios

El problema del suplente confuso es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación entre servicios puede dar lugar al problema del suplente confuso. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos

de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición global `aws:SourceArn` o `aws:SourceAccount` en las políticas de recursos para limitar los permisos que Amazon EventBridge concede a otro servicio para el recurso. Utilice `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utilice `aws:SourceAccount` si quiere permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios.

La forma más eficaz de protegerse contra el problema del suplente confuso es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si está especificando varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con caracteres comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un ARN de bucket de Amazon S3, debe utilizar ambas claves de contexto de condición global para limitar los permisos.

Buses de eventos

Para los destinos de las reglas del bus de eventos de EventBridge, el valor de `aws:SourceArn` debe ser el ARN de la regla.

El siguiente ejemplo muestra cómo se pueden utilizar las claves contextuales de condición global `aws:SourceArn` y `aws:SourceAccount` en EventBridge para evitar el problema del adjunto confundido. Este ejemplo se utiliza en una política de confianza de roles, para un rol utilizado por una regla de EventBridge.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
    ],
    "Condition": {
      "ArnLike": {
```

```
        "aws:SourceArn": "arn:aws:events:*:123456789012:rule/myRule"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

EventBridge Pipes

En el caso de EventBridge Pipes, el valor de `aws:SourceArn` debe ser el ARN de la canalización.

El siguiente ejemplo muestra cómo se pueden utilizar las claves contextuales de condición global `aws:SourceArn` y `aws:SourceAccount` en EventBridge para evitar el problema del adjunto confundido. Este ejemplo se utiliza en una política de confianza de roles, para un rol utilizado por 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"
      }
    }
  }
}
```

Políticas basadas en recursos para los esquemas de Amazon EventBridge

El registro de esquemas de EventBridge admite políticas basadas en recursos. Una política de recurso es una política asociada a un recurso en lugar de a una identidad de IAM. Por ejemplo, en Amazon Simple Storage Service (Amazon S3), una política de recursos se asocia a un bucket de Amazon S3.

Para obtener más información sobre los esquemas de EventBridge y las políticas basadas en recursos, consulte los siguientes temas.

- Referencia de tipo de recurso de esquemas de Amazon EventBridge
- Políticas basadas en identidad y políticas basadas en recursos en la Guía del usuario de IAM

API compatibles con las políticas basadas en recursos

Puedes usar las siguientes API con políticas basadas en recursos para el registro de esquemas de EventBridge.

- `DescribeRegistry`
- `UpdateRegistry`
- `DeleteRegistry`
- `ListSchemas`
- `SearchSchemas`
- `DescribeSchema`
- `CreateSchema`
- `DeleteSchema`
- `UpdateSchema`
- `ListSchemaVersions`
- `DeleteSchemaVersion`
- `DescribeCodeBinding`
- `GetCodeBindingSource`
- `PutCodeBinding`

Ejemplo de política que concede todas las acciones compatibles a una cuenta AWS

Para el registro de esquemas de EventBridge, siempre debe asociar una política basada en recursos a un registro. Para conceder acceso a un esquema, debe especificar el ARN del esquema y el ARN del registro en la política.

Para conceder a un usuario acceso a todas las API disponibles para los esquemas de EventBridge, utiliza una política similar a la siguiente, sustituyéndola por el `"Principal"` ID de cuenta de la cuenta a la que quieres conceder acceso.

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

Ejemplo de política que concede acciones de solo lectura a una cuenta AWS

El siguiente ejemplo concede acceso a una cuenta solo para las API de solo lectura de los esquemas de 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*"
            ]
        }
    ]
}
```

Ejemplo de política que concede todas las acciones a una organización

Puedes usar políticas basadas en recursos con el registro de esquemas de EventBridge para conceder acceso a una organización. Para obtener más información, consulte la Guía del usuario de AWS Organizations. El siguiente ejemplo otorga a la organización un ID de `o-a1b2c3d4e5` acceso al registro de esquemas.

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

Referencia de permisos de Amazon EventBridge

Para especificar una acción en una política de EventBridge, use el prefijo `events:` seguido del nombre de operación de API, como se muestra en el ejemplo siguiente.

```
"Action": "events:PutRule"
```

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo.

```
"Action": ["events:action1", "events:action2"]
```

Para especificar varias acciones, también puede utilizar caracteres comodín. Por ejemplo, puede especificar todas las acciones que comiencen por la palabra `"Put"` del siguiente modo.

```
"Action": "events:Put*"
```

Para especificar todas las acciones de la API de EventBridge, use el carácter comodín `*` del siguiente modo.

```
"Action": "events:*"
```

En la siguiente tabla se enumeran las operaciones de la API de EventBridge y las acciones correspondientes que puede especificar en una política de IAM.

Operación de la API de EventBridge	Permisos necesarios	Descripción
DeleteRule	`events:DeleteRule`	Necesario para eliminar una regla.
DescribeEventBus	`events:DescribeEventBus`	Necesario para enumerar las cuentas que pueden escribir eventos en el bus de eventos actual de la cuenta.
DescribeRule	`events:DescribeRule`	Necesario para mostrar detalles acerca de una regla.

Operación de la API de EventBridge	Permisos necesarios	Descripción
DisableRule	`events:DisableRule`	Necesario para deshabilitar una regla.
EnableRule	`events:EnableRule`	Necesario para habilitar una regla.
ListRuleNamesByTarget	`events:ListRuleNamesByTarget`	Necesario para enumerar reglas asociadas a un destino.
ListRules	`events:ListRules`	Necesario para enumerar todas las reglas de su cuenta.
ListTagsForResource	`events:ListTagsForResource`	Necesario para generar una lista de todas las etiquetas asociadas a un recurso de EventBridge. Actualmente, solo se pueden etiquetar las reglas.
ListTargetsByRule	`events:ListTargetsByRule`	Necesario para enumerar todos los destinos asociados a una regla.
PutEvents	`events:PutEvents`	Necesario para agregar eventos personalizados que se pueden asignar a reglas.
PutPermission	`events:PutPermission`	Necesario para dar permiso a otra cuenta para escribir eventos en el bus de eventos predeterminado de esta cuenta.
PutRule	`events:PutRule`	Necesario para crear o actualizar una regla.

Operación de la API de EventBridge	Permisos necesarios	Descripción
PutTargets	`events:PutTargets`	Necesario para añadir destinos a una regla.
RemovePermission	`events:RemovePermission`	Necesario para revocar el permiso de otra cuenta para escribir eventos en el bus de eventos predeterminado de esta cuenta.
RemoveTargets	`events:RemoveTargets`	Necesario para eliminar un destino de una regla.
TestEventPattern	`events:TestEventPattern`	Necesario para probar un patrón de evento con respecto a un evento dado.

Usar condiciones de las políticas de IAM para control de acceso preciso

A la hora de conceder permisos, se utiliza el lenguaje de las políticas de IAM para especificar las condiciones en la que se debe aplicar una política. Por ejemplo, puede tener una política que solo se aplique después de una fecha específica.

Una condición de una política consta de pares clave-valor. Las claves de condición no distinguen entre mayúsculas y minúsculas.

Si especifica varias condiciones o claves en una única condición, para que EventBridge conceda el permiso deben cumplirse todas las condiciones y claves. Si especifica una única condición con varios valores para una clave, EventBridge concede el permiso si se cumple uno de los valores.

También puede utilizar marcadores de posición o variables de políticas al especificar condiciones. Para obtener más información, consulte Variables de políticas en la Guía del usuario de IAM. Para obtener más información sobre cómo especificar condiciones en un lenguaje de políticas de IAM, consulte Condición en la Guía del usuario de IAM.

De forma predeterminada, los usuarios y los roles de IAM no pueden acceder a los eventos en su cuenta. Para acceder a los eventos, un usuario debe estar autorizado para la acción del API `PutRule`. Si un usuario o un rol de IAM recibe autorización para la acción `events:PutRule`, puede crear una regla que coincida con determinados eventos. Sin embargo, para que la regla sea útil, el usuario también debe tener permisos para la acción `events:PutTargets` porque, si quiere que la regla haga algo más que publicar una métrica de CloudWatch, también debe añadir un destino a la regla.

Puede proporcionar una condición en la instrucción de política de un usuario o rol de IAM que permita a este crear una regla que solo coincida con un conjunto específico de orígenes y tipos de detalles. Para conceder acceso a orígenes y tipos de eventos específicos, utilice las claves de condición `events:source` y `events:detail-type`.

Del mismo modo, puede proporcionar una condición en la instrucción de política de un usuario o rol de IAM que permita a este crear una regla que solo coincida con un recurso específico de sus cuentas. Para conceder acceso a un recurso específico, utilice la clave de condición `events:TargetArn`.

El siguiente ejemplo es una política que permite a los usuarios acceder a todos los eventos de EventBridge, excepto a los de Amazon EC2, mediante una instrucción de denegación en la acción de la API `PutRule`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyPutRuleForAllEC2Events",
            "Effect": "Deny",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

Claves de condición de EventBridge

En la siguiente tabla se muestran las claves de condición y los pares de claves y valores que puede usar en una política de EventBridge.

Clave de condición	Par clave-valor	Tipos de evaluación
aws:SourceAccount	La cuenta en la que reside la regla especificada por `aws:SourceArn` .	ID de cuenta, Null
aws:SourceArn	El ARN de la regla que envía el evento.	ARN, Null
events:creatorAccount	`"events:creatorAccount":"` *`creatorAccount`* `"` En *`CreatorAccount`* , utilice el ID de la cuenta que creó la regla. Use esta condición para autorizar las llamadas a la API en reglas de una cuenta específica.	creatorAccount, Null

Clave de condición	Par clave-valor	Tipos de evaluación
events:detail-type	`"events:detail-type":"` *`detail-type`* `"` Donde *`detail-type`* es la cadena literal del campo detail-type del evento, como `"AWS API Call via CloudTrail"` y `"EC2 Instance State-change Notification"` .	Tipo de detalle, Null
events: detail.eventTypeCode	`"events:detail.eventTypeCode":"` *`eventTypeCode`* `"` En *`eventTypeCode`* , use la cadena literal para el campo detail.eventTypeCode del evento, como `"AWS_ABUSE_DOS_REPORT"` .	eventTypeCode, Null
events: detail.service	`"events:detail.service":"` *`service`*`"` En *`service`*, use la cadena literal para el campo detail.service del evento, como `"ABUSE"`.	servicio, Null

Clave de condición	Par clave-valor	Tipos de evaluación
events: detail.userIdentity.principalId	`"events:detail.userIdentity.principalId":" principal-id "` En *principal-id* , use la cadena literal para el campo detail.userIdentity.principalId del evento con detail-type `"AWS API Call via CloudTrail"` , como `"AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName."` .	ID de entidad principal, Null
events:eventBusInvocation	`"events:eventBusInvocation":" boolean"` En *booleano*, use true cuando una regla envíe un evento a un destino que sea un bus de eventos de otra cuenta. Usa false cuando se utilice una llamada a la API `PutEvents` .	eventBusInvocation, Null
events:ManagedBy	Utilizado internamente por los servicios de AWS. Si un servicio de AWS crea una regla en su nombre, el valor será el nombre de la entidad principal del servicio que la creó.	No está diseñado para su uso en las políticas de clientes.

Clave de condición	Par clave-valor	Tipos de evaluación
events:source	`"events:source":" source "` Use *source* para la cadena literal para el campo de origen del evento, como `"aws.ec2"` o `"aws.s3"`. Para ver más valores posibles de *source*, consulte los eventos de ejemplo en Eventos de los AWS servicios.	Origen, Null
events:TargetArn	`"events:TargetArn" :" target-arn "` Para *target-arn* , use el ARN del destino para la regla, por ejemplo `"arn:aws:lambda:*:*:function:*"` .	ArrayOfARN, Null

Para obtener ejemplos de instrucciones de política, consulte Administrar permisos de acceso para los recursos de Amazon EventBridge.

Temas

- Aspectos específicos de EventBridge Pipes
- Ejemplo: uso de la condición creatorAccount
- Ejemplo: Uso de la condición eventBusInvocation
- Ejemplo: Limitar el acceso a un origen específico
- Ejemplo: Definir varios orígenes que puedan utilizarse en un patrón de eventos individualmente
- Ejemplo: Definir un origen y un DetailType que puedan utilizarse en un patrón de eventos
- Ejemplo: Comprobar que el origen se ha definido en el patrón de eventos
- Ejemplo: Definir una lista de orígenes permitidos en un patrón de eventos con varios orígenes
- Ejemplo: Limitar el acceso a PutRule mediante detail.service

- Ejemplo: Limitar el acceso a PutRule mediante detail.eventTypeCode
- Ejemplo: Garantizar que solo se permitan eventos de AWS CloudTrail para llamadas a la API de un PrincipalId determinado
- Ejemplo: Limitar el acceso a los destinos

Aspectos específicos de EventBridge Pipes

EventBridge Pipes no admite ninguna clave de condición adicional de las políticas de IAM.

Ejemplo: uso de la condición **`creatorAccount`**

En el siguiente ejemplo de instrucción de política se muestra cómo utilizar la condición `creatorAccount` en una política para permitir la creación de reglas únicamente si la cuenta especificada como `creatorAccount` es la cuenta que creó la regla.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForOwnedRules",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "events:creatorAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
}
```

Ejemplo: Uso de la condición **`eventBusInvocation`**

La `eventBusInvocation` indica si la invocación se origina en un destino entre cuentas o en una solicitud de API `PutEvents`. El valor es true cuando la invocación es el resultado de una regla que incluye un destino entre cuentas, como cuando el deestino es un bus de eventos de otra cuenta. El valor es false cuando la invocación es el resultado de una solicitud de API `PutEvents`. El siguiente ejemplo indica una invocación desde un destino entre cuentas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountInvocationEventsOnly",
      "Effect": "Allow",
      "Action": "events:PutEvents",
      "Resource": "*",
      "Condition": {
        "BoolIfExists": {
          "events:eventBusInvocation": "true"
        }
      }
    }
  ]
}
```

Ejemplo: Limitar el acceso a un origen específico

Las siguientes políticas de ejemplo se pueden adjuntar a un usuario de IAM;. La Política A permite la acción de la API `PutRule` para todos los eventos, mientras que la Política B permite `PutRule` únicamente si el patrón de eventos de la regla que se crea coincide con eventos de Amazon EC2.

Política A: permitir todos los eventos

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEvents",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*"
        }
    ]
    }
```

Política B: permitir eventos solo desde Amazon EC2

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

```
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEC2Events",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

`EventPattern` es un argumento obligatorio para `PutRule`. Por lo tanto, si el usuario con la Política B llama a `PutRule` con un patrón de eventos como el siguiente:

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

La regla se crearía porque la política permite este origen específico, es decir, `"aws.ec2"`. Sin embargo, si el usuario con la Política B llama a `PutRule` con un patrón de eventos como el siguiente, la creación de la regla se denegaría porque la política no permite este origen específico: es decir, `"aws.s3"`.

```
{
    "source": [ "aws.s3" ]
}
```

Básicamente, al usuario con la Política B solo se le permite crear una regla que coincida con los eventos procedentes de Amazon EC2, por lo tanto, solo se le permite el acceso a los eventos desde Amazon EC2.

Consulte la tabla siguiente para una comparación de Política A y Política B.

Patrón de eventos	Permitidos por la Política A	Permitidos por la Política B
`{`	Sí	Sí

Patrón de eventos	Permitidos por la Política A	Permitidos por la Política B
`"source": [ "aws.ec2" ] }`		
`{ "source": [ "aws.ec2", "aws.s3" ] }`	Sí	No (el origen aws.s3 no está permitido)
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sí	Sí
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sí	No (el origen debe especificarse)

Ejemplo: Definir varios orígenes que puedan utilizarse en un patrón de eventos individualmente

El siguiente ejemplo permite a un usuario o rol de IAM crear una regla cuyo origen en el `EventPattern` sea Amazon EC2 o Amazon ECS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsEC2OrECS",
```

```
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": [ "aws.ec2", "aws.ecs" ]
                }
            }
        }
    ]
}
```

En la tabla siguiente se ofrecen ejemplos de patrones de eventos que esta política permite o deniega.

Patrón de eventos	Permitidos por la política
`{ "source": [ "aws.ec2" ] }`	Sí
`{ "source": [ "aws.ecs" ] }`	Sí
`{ "source": [ "aws.s3" ] }`	No
`{ "source": [ "aws.ec2", "aws.ecs" ] }`	No
`{ "detail-type": [ "AWS API Call via CloudTrail" ] }`	No

Ejemplo: Definir un origen y un **`DetailType`** que puedan utilizarse en un patrón de eventos

La siguiente política permite eventos solo desde el origen `aws.ec2` con `DetailType` igual 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"
                }
            }
        }
    ]
}
```

En la tabla siguiente se ofrecen ejemplos de patrones de eventos que esta política permite o deniega.

Patrón de eventos	Permitidos por la política
`{ "source": [ "aws.ec2" ] }`	No
`{ "source": [ "aws.ecs" ] }`	No
`{`	Sí

Patrón de eventos	Permitidos por la política

<pre>"source": ["aws.ec2"],
"detail-type": ["EC2
Instance State-change Notificat
ion"]
}</pre>	

<pre>{
"source": ["aws.ec2"],
"detail-type": ["EC2
Instance Health Failed"]
}</pre>
	No

<pre>{
"detail-type": ["EC2
Instance State-change Notificat
ion"]
}</pre>
	No

Ejemplo: Comprobar que el origen se ha definido en el patrón de eventos

La siguiente política permite a los usuarios crear reglas con `EventPatterns` que tienen el campo de origen. Con esta política, un usuario o un rol de IAM no puede crear una regla con un `EventPattern` que no proporcione un origen específico.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsSpecified",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "Null": {
                    "events:source": "false"
                }
            }
        }
```

```
    ]
}
```

En la tabla siguiente se ofrecen ejemplos de patrones de eventos que esta política permite o deniega.

Patrón de eventos	Permitidos por la Política
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sí
`{ "source": [ "aws.ecs", "aws.ec2" ] }`	Sí
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	No

Ejemplo: Definir una lista de orígenes permitidos en un patrón de eventos con varios orígenes

La siguiente política permite a los usuarios crear reglas con `EventPatterns` que puede tener varios orígenes en ellas. Cada origen del patrón de eventos debe ser miembro de la lista proporcionada en la condición. Cuando utilice la condición `ForAllValues`, asegúrese de que al menos uno de los elementos de la lista de condiciones esté definido.

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

En la tabla siguiente se ofrecen ejemplos de patrones de eventos que esta política permite o deniega.

Patrón de eventos	Permitidos por la Política
`{ "source": [ "aws.ec2" ] }`	Sí
`{ "source": [ "aws.ec2", "aws.s3" ] }`	Sí
`{ "source": [ "aws.ec2", "aws.autoscaling" ] }`	No
`{ "detail-type": [ "EC2 Instance State-change Notification" ]`	No

Patrón de eventos	Permitidos por la Política
}	

Ejemplo: Limitar el acceso a **`PutRule`** mediante **`detail.service`**

Puede restringir un usuario o un rol de IAM de forma que solo pueda crear reglas para eventos que tengan un determinado valor en el campo `events:details.service`. El valor de `events:details.service` no es necesariamente el nombre de un servicio de AWS.

Esta condición de política resulta útil al trabajar con eventos de AWS Health relacionados con la seguridad o el abuso. Al utilizar esta condición de política, puede limitar el acceso a estas alertas sensibles únicamente a aquellos usuarios que necesiten verlas.

Por ejemplo, la siguiente política permite la creación de reglas solo para los eventos cuyo donde el valor de `events:details.service` sea `ABUSE`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleEventsWithDetailServiceEC2",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail.service": "ABUSE"
                }
            }
        }
    ]
}
```

Ejemplo: Limitar el acceso a **`PutRule`** mediante **`detail.eventTypeCode`**

Puede restringir un usuario o un rol de IAM de forma que solo pueda crear reglas para eventos que tengan un determinado valor en el campo `events:details.eventTypeCode`. Esta condición de política resulta útil al trabajar con eventos de AWS Health relacionados con la seguridad o el abuso.

Al utilizar esta condición de política, puede limitar el acceso a estas alertas sensibles únicamente a aquellos usuarios que necesiten verlas.

Por ejemplo, la siguiente política permite la creación de reglas solo para los eventos cuyo donde el valor de `events:details.eventTypeCode` sea `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"
                }
            }
        }
    ]
}
```

Ejemplo: Garantizar que solo se permitan eventos de AWS CloudTrail para llamadas a la API de un **`PrincipalId`** determinado

Todos los eventos de AWS CloudTrail tienen el PrincipalId del usuario que hizo la llamada al API en la ruta `detail.userIdentity.principalId` de un evento. Usando la clave de condición `events:detail.userIdentity.principalId`, puede limitar el acceso de los usuarios o roles de IAM a los eventos de CloudTrail únicamente para los procedentes de una cuenta específica.

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

En la tabla siguiente se ofrecen ejemplos de patrones de eventos que esta política permite o deniega.

Patrón de eventos	Permitidos por la política
`{ "detail-type": [ "AWS API Call via CloudTrail" ] }`	No
`{ "detail-type": [ "AWS API Call via CloudTrail" ], "detail.userIdentity.principalId": [ "AIDAJ45Q7YFFAREXAMPLE" ] }`	Sí
`{ "detail-type": [ "AWS API Call via CloudTrail" ], "detail.userIdentity.principalId": [ "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName" ] }`	No

Ejemplo: Limitar el acceso a los destinos

Si un usuario o un rol de IAM; tiene permiso `events:PutTargets`, puede añadir cualquier destino en la misma cuenta a las reglas a las que se les permita el acceso. La siguiente política limita la adición de destinos a solo una regla específica: `MyRule` en la cuenta `123456789012`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRule",
            "Effect": "Allow",
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule"
        }
    ]
}
```

Para limitar los destinos que se pueden añadir a la regla, utilice la clave de condición `events:TargetArn`. Por ejemplo, puede limitar destinos solo a funciones de Lambda, como en el siguiente ejemplo.

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

Uso de roles vinculados a servicios de EventBridge

Amazon EventBridge utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que está vinculado directamente a EventBridge. Los roles vinculados a servicios están predefinidos por EventBridge e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Temas

Uso de roles para crear secretos para destinos de API

Amazon EventBridge utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que está vinculado directamente a EventBridge. Los roles vinculados a servicios están predefinidos por EventBridge e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Con un rol vinculado a servicios, resulta más sencillo configurar EventBridge, porque no es preciso agregar los permisos necesarios manualmente. EventBridge define los permisos de los roles vinculados con su propio servicio y, a menos que esté definido de otra manera, solo EventBridge puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo puede eliminar un rol vinculado a servicios después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de EventBridge, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Sí en la columna Roles vinculados a servicios. Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios de EventBridge

EventBridgeutiliza el rol vinculado al servicio denominado AWSServiceRoleForAmazonEventBridgeApiDestinations— Permite el acceso a los Secrets Manager Secrets creados por. EventBridge

El rol vinculado al servicio AWSServiceRoleForAmazonEventBridgeApiDestinations depende de los siguientes servicios para asumir el rol:

- `apidestinations.events.amazonaws.com`

La política de permisos de roles denominada AmazonEventBridgeApiDestinationsServiceRolePolítica permite EventBridge realizar las siguientes acciones en los recursos especificados:

- Acción: `create, describe, update and delete secrets; get and put secret values` en `secrets created for all connections by EventBridge`

Debe configurar los permisos para permitir a sus usuarios, grupos o funciones, crear, editar o eliminar la descripción de un rol vinculado al servicio. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado a un servicio de EventBridge

No necesita crear manualmente un rol vinculado a servicios. Al crear una conexión en laAWS Management Console, la AWS CLI o la AWS API, EventBridge crea automáticamente el rol vinculado al servicio.

> **Important**
>
> Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol. Si utilizabas el EventBridge servicio antes del 11 de febrero de 2021, cuando comenzó a admitir roles vinculados al servicio, entonces EventBridge creaste el AWSServiceRoleForAmazonEventBridgeApiDestinationsrol en tu cuenta. Para obtener más información, consulte Un nuevo rol ha aparecido en mi Cuenta de AWS.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al crear una conexión, vuelve a EventBridge crear el rol vinculado al servicio para ti.

Editar un rol vinculado a un servicio para EventBridge

EventBridge no le permite editar el rol vinculado a servicios AWSServiceRoleForAmazonEventBridgeApiDestinations. Después de crear un rol vinculado al

servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminación de un rol vinculado a un servicio de EventBridge

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a servicios, recomendamos que elimine dicho rol. De esta forma, no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Limpiar un rol vinculado a servicios

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero debe eliminar los recursos que utiliza el rol.

> **Note**
>
> Si el servicio EventBridge está utilizando el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de EventBridge utilizados por AWSServiceRoleForAmazonEventBridgeApiDestinations (consola)

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En Integraciones, elige los destinos de las API y, a continuación, selecciona la pestaña Conexiones.
3. Elige la conexión y, a continuación, selecciona Eliminar.

Para eliminar los recursos de EventBridge utilizados por AWSServiceRoleForAmazonEventBridgeApiDestinations (CLI de AWS)

- Usa el siguiente comando:`delete-connection`.

Para eliminar los recursos de EventBridge utilizados por AWSServiceRoleForAmazonEventBridgeApiDestinations (API)

- Usa el siguiente comando:`DeleteConnection`.

Eliminación manual de un rol vinculado a servicios

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a servicios de AWSServiceRoleForAmazonEventBridgeApiDestinations. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a un servicio de EventBridge

EventBridge admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio esté disponible. Para obtener más información, consulte Puntos de conexión y Regiones de AWS.

Uso de roles para el descubrimiento de esquemas

Amazon EventBridge utiliza roles vinculados a servicios de AWS Identity and Access Management (IAM). Un rol vinculado a servicios es un tipo único de rol de IAM que está vinculado directamente a EventBridge. Los roles vinculados a servicios están predefinidos por EventBridge e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Con un rol vinculado a servicios, resulta más sencillo configurar EventBridge, porque no es preciso agregar los permisos necesarios manualmente. EventBridge define los permisos de los roles vinculados con su propio servicio y, a menos que esté definido de otra manera, solo EventBridge puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo puede eliminar un rol vinculado a servicios después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de EventBridge, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Sí en la columna Roles vinculados a servicios. Seleccione una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios de EventBridge

EventBridgeutiliza el rol vinculado al servicio denominado AWSServiceRoleForSchemas: Otorga permisos a las reglas administradas creadas por Amazon EventBridge los esquemas.

El rol vinculado al servicio AWSServiceRoleForSchemas depende de los siguientes servicios para asumir el rol:

- `schemas.amazonaws.com`

La política de permisos de roles denominada AmazonEventBridgeSchemasServiceRolePolicypermite EventBridge realizar las siguientes acciones en los recursos especificados:

- Acción: `put, enable, disable, and delete rules; put and remove targets; list targets per rule` en `all managed rules created by EventBridge`

Debe configurar los permisos para permitir a sus usuarios, grupos o funciones, crear, editar o eliminar la descripción de un rol vinculado al servicio. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Creación de un rol vinculado a un servicio de EventBridge

No necesita crear manualmente un rol vinculado a servicios. Al realizar una detección de esquemas en la AWS Management ConsoleAWS CLI, la o la AWS API, se EventBridge crea automáticamente el rol vinculado al servicio.

> **Important**
>
> Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol. Si utilizaba el EventBridge servicio antes del 27 de noviembre de 2019, cuando comenzó a admitir funciones vinculadas al servicio, EventBridge creó la AWSServiceRoleForSchemasfunción en su cuenta. Para obtener más información, consulte Un nuevo rol ha aparecido en mi Cuenta de AWS.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al realizar una detección de esquemas, vuelve a EventBridge crear el rol vinculado al servicio para usted.

Editar un rol vinculado a un servicio para EventBridge

EventBridge no le permite editar el rol vinculado a servicios AWSServiceRoleForSchemas. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminación de un rol vinculado a un servicio de EventBridge

Si ya no necesita utilizar una característica o servicio que requiere un rol vinculado a servicios, recomendamos que elimine dicho rol. De esta forma, no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado a servicios antes de eliminarlo manualmente.

Limpiar un rol vinculado a servicios

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero debe eliminar los recursos que utiliza el rol.

> **Note**
>
> Si el servicio EventBridge está utilizando el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de EventBridge utilizados por AWSServiceRoleForSchemas (consola)

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En Autobuses, selecciona Autobuses de eventos y, a continuación, elige un autobús de eventos.
3. Selecciona Stop Discovery.

Para eliminar los recursos de EventBridge utilizados por AWSServiceRoleForSchemas (CLI de AWS)

- Usa el siguiente comando:`delete-discoverer`.

Para eliminar los recursos de EventBridge utilizados por AWSServiceRoleForSchemas (API)

- Usa el siguiente comando:`DeleteDiscoverer`.

Eliminación manual de un rol vinculado a servicios

Utilice la consola de IAM, la AWS CLI o la API de AWS para eliminar el rol vinculado a servicios de AWSServiceRoleForSchemas. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a un servicio de EventBridge

EventBridge admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio esté disponible. Para obtener más información, consulte Puntos de conexión y Regiones de AWS.

Registro de llamadas a la Amazon EventBridge API mediante AWS CloudTrail

Amazon EventBridge está integrado con AWS CloudTrailun servicio que proporciona un registro de las acciones realizadas por un usuario, rol o un Servicio de AWS. CloudTrail captura todas las llamadas a la API EventBridge como eventos. Las llamadas capturadas incluyen llamadas desde la EventBridge consola y llamadas en código a las operaciones de la EventBridge API. Con la información recopilada por CloudTrail, puede determinar a qué solicitud se realizó EventBridge, la dirección IP desde la que se realizó la solicitud, cuándo se realizó y detalles adicionales.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales del usuario raíz o del usuario.
- Si la solicitud se realizó en nombre de un usuario de IAM Identity Center.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro Servicio de AWS.

CloudTrail está activa en tu cuenta Cuenta de AWS al crear la cuenta y automáticamente tienes acceso al historial de CloudTrail eventos. El historial de CloudTrail eventos proporciona un registro visible, consultable, descargable e inmutable de los últimos 90 días de eventos de gestión registrados en un. Región de AWSPara obtener más información, consulte Uso del historial de CloudTrail eventos en la Guía del usuario.AWS CloudTrail La visualización del historial de eventos no conlleva ningún CloudTrail cargo.

Para tener un registro continuo de los eventos de Cuenta de AWS los últimos 90 días, crea un almacén de datos de eventos de senderos o CloudTraillagos.

CloudTrail senderos

Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. Todos los senderos creados con él AWS Management Console son multirregionales. Puede crear un registro de seguimiento de una sola región o de varias regiones mediante la AWS CLI. Se recomienda crear un sendero multirregional, ya que puedes capturar toda la actividad de tu Regiones de AWS cuenta. Si crea un registro de seguimiento de una sola región, solo podrá ver los eventos registrados en la Región de AWS del registro de seguimiento. Para obtener

más información acerca de los registros de seguimiento, consulte Creación de un registro de seguimiento para su Cuenta de AWS y Creación de un registro de seguimiento para una organización en la Guía del usuario de AWS CloudTrail .

Puede enviar una copia de sus eventos de administración en curso a su bucket de Amazon S3 sin coste alguno CloudTrail mediante la creación de una ruta; sin embargo, hay cargos por almacenamiento en Amazon S3. Para obtener más información sobre CloudTrail los precios, consulte AWS CloudTrail Precios. Para obtener información acerca de los precios de Amazon S3, consulte Precios de Amazon S3.

CloudTrail Almacenes de datos de eventos en Lake

CloudTrail Lake le permite ejecutar consultas basadas en SQL en sus eventos. CloudTrail Lake convierte los eventos existentes en formato JSON basado en filas al formato Apache ORC. ORC es un formato de almacenamiento en columnas optimizado para una recuperación rápida de datos. Los eventos se agregan en almacenes de datos de eventos, que son recopilaciones inmutables de eventos en función de criterios que se seleccionan aplicando selectores de eventos avanzados. Los selectores que se aplican a un almacén de datos de eventos controlan los eventos que perduran y están disponibles para la consulta. Para obtener más información sobre CloudTrail Lake, consulte Cómo trabajar con AWS CloudTrail Lake en la Guía del AWS CloudTrail usuario.

CloudTrail Los almacenes de datos y las consultas sobre eventos de Lake conllevan costes. Cuando crea un almacén de datos de eventos, elige la opción de precios que desea utilizar para él. La opción de precios determina el costo de la incorporación y el almacenamiento de los eventos, así como el periodo de retención predeterminado y máximo del almacén de datos de eventos. Para obtener más información sobre CloudTrail los precios, consulte AWS CloudTrail Precios.

EventBridge eventos de datos en CloudTrail

Los eventos de datos proporcionan información sobre las operaciones de recursos realizadas en o dentro de un recurso (por ejemplo, leer o escribir en un objeto de Amazon S3). Se denominan también operaciones del plano de datos. Los eventos de datos suelen ser actividades de gran volumen. De forma predeterminada, CloudTrail no registra los eventos de datos. El historial de CloudTrail eventos no registra los eventos de datos.

Se aplican cargos adicionales a los eventos de datos. Para obtener más información sobre CloudTrail los precios, consulta AWS CloudTrail Precios.

Puede registrar eventos de datos para los tipos de EventBridge recursos mediante la CloudTrail consola o las operaciones de la CloudTrail API. AWS CLI Para obtener más información sobre cómo registrar los eventos de datos, consulte Registro de eventos de datos con la AWS Management Console y Registro de eventos de datos con la AWS Command Line Interface en la Guía del usuario de AWS CloudTrail .

En la siguiente tabla se enumeran los tipos de EventBridge recursos para los que puede registrar eventos de datos. La columna Tipo de evento de datos (consola) muestra el valor que se puede elegir en la lista de tipos de eventos de datos de la CloudTrail consola. La columna de valores resources.type muestra el `resources.type` valor que se debe especificar al configurar los selectores de eventos avanzados mediante las API o. AWS CLI CloudTrail La CloudTrail columna API de datos en la que se ha registrado muestra las llamadas a la API registradas CloudTrail para el tipo de recurso.

Tipo de evento de datos (consola)	resources.type value	Las API de datos registradas en CloudTrail
Bus de eventos	`AWS::Events::Event Bus`	• DescribeEventBus
Regla del bus de eventos	`AWS::Events::Rule`	• DeleteRule • DescribeRule • DisableRule • EnableRule • ListRuleNamesByTarget • ListRules • ListTargetsByRule • PutRule • PutTargets • RemoveTargets • TestEventPattern
Tubo	`AWS::Pipes::Pipe`	• CreatePipe • DeletePipe • DescribePipe

Tipo de evento de datos (consola)	resources.type value	Las API de datos registradas en CloudTrail
		• ListPipes • StartPipe • StopPipe • UpdatePipe

Puede configurar selectores de eventos avanzados para filtrar según los campos `eventName`, `readOnly` y `resources.ARN` y así registrar solo los eventos que son importantes para usted. Para obtener más información acerca de estos campos, consulte AdvancedFieldSelector en la Referencia de la API de AWS CloudTrail .

EventBridge eventos de gestión en CloudTrail

Los eventos de administración proporcionan información sobre las operaciones de administración que se llevan a cabo en los recursos de su empresa Cuenta de AWS. Se denominan también operaciones del plano de control. De forma predeterminada, CloudTrail registra los eventos de administración.

Amazon EventBridge registra todas las operaciones del plano de EventBridge control como eventos de administración. Para obtener una lista de las operaciones del plano de Amazon EventBridge control en las que se EventBridge registra CloudTrail, consulte la referencia de la Amazon EventBridge API.

EventBridge ejemplos de eventos

Un evento representa una solicitud única de cualquier fuente e incluye información sobre la operación de API solicitada, la fecha y la hora de la operación, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que los eventos no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra un CloudTrail evento que demuestra la `PutRule` operación.

```
{
  "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"
}
```


Para obtener información sobre el contenido de los CloudTrail registros, consulte el contenido de los CloudTrail registros en la Guía del AWS CloudTrail usuario.

CloudTrail entradas de registro de las acciones realizadas por EventBridge Pipes

EventBridge Pipes asume la función de IAM proporcionada al leer eventos de las fuentes, invocar enriquecimientos o invocar objetivos. En el caso de las CloudTrail entradas relacionadas con

las acciones realizadas en su cuenta en todos los enriquecimientos, objetivos y fuentes de Amazon SQS, Kinesis y DynamoDB, se incluirán los campos y. `sourceIPAddress` `invokedBy` `pipes.amazonaws.com`

Ejemplo de entrada de CloudTrail registro para todos los enriquecimientos, objetivos y fuentes de Amazon SQS, Kinesis y 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"
}
```

Para todas las demás fuentes, el `sourceIPAddress` campo de las entradas de CloudTrail registro tendrá una dirección IP dinámica y no se debe confiar en él para ninguna integración o categorización de eventos. Además, estas entradas no tendrán el campo `invokedBy`.

Ejemplo de entrada de CloudTrail registro para todas las demás fuentes

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    ...
  },
  "eventTime": ",,,",
  "eventName": "...",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Python-httplib2/0.8 (gzip)",
}
```

Validación de conformidad en Amazon EventBridge

Auditores externos como SOC, PCI, FedRAMP e HIPAA evalúan la seguridad y la conformidad de los servicios de AWS como parte de varios programas de conformidad de AWS.

Para obtener una lista de los servicios de AWS en el ámbito de programas de conformidad específicos, consulte AWS Services in Scope by Compliance Program (Servicios en el ámbito de programas de conformidad). Para obtener información general, consulte Programas de conformidad de AWS.

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

Su responsabilidad de conformidad al utilizar EventBridge se determina en función de la sensibilidad de los datos, los objetivos de cumplimiento de su empresa y la legislación y los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- Guías de inicio rápido de seguridad y conformidad: consideraciones sobre arquitectura y pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre arquitectura para seguridad y conformidad de HIPAA: se describe cómo las empresas pueden utilizar AWS para crear aplicaciones conformes con la ley HIPAA.
- Recursos de conformidad de AWS: colección de manuales y guías.
- Evaluación de recursos con reglas en la Guía para desarrolladores de AWS Config: información sobre el modo en que AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- AWS Security Hub: un servicio que ofrece una vista integral de su estado de seguridad en AWS que lo ayuda a comprobar la conformidad con las normas del sector de seguridad y las prácticas recomendadas.

Resiliencia de Amazon EventBridge

La infraestructura global de AWS se compone de regiones y zonas de disponibilidad de AWS. AWS Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las regiones y zonas de disponibilidad de AWS, consulte la Infraestructura global de AWS.

Seguridad de la infraestructura de Amazon EventBridge

Como se trata de un servicio administrado, Amazon EventBridge se encuentra protegido por la seguridad de red global de AWS. Para obtener información sobre los servicios de seguridad de AWS y cómo AWS protege la infraestructura, consulte Seguridad en la nube de AWS. Para diseñar su entorno de AWS con las prácticas recomendadas de seguridad de infraestructura, consulte Protección de la infraestructura en Portal de seguridad de AWS Well-Architected Framework.

Puede utilizar llamadas a la API publicadas en AWS para acceder a EventBridge a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Puede llamar a estas operaciones de la API desde cualquier ubicación de red y puede usar políticas de acceso basadas en recursos en EventBridge, que pueden incluir restricciones en función de la dirección IP de origen. También puede utilizar políticas de EventBridge para controlar el acceso desde puntos de conexión específicos de Amazon Virtual Private Cloud (Amazon VPC) o VPC específicas. Este proceso aísla con eficacia el acceso de red a un recurso de EventBridge determinado únicamente desde la VPC específica de la red de AWS.

Configuración y análisis de vulnerabilidades en Amazon EventBridge

La configuración y los controles de TI son una responsabilidad compartida entre AWS y usted, nuestro cliente. Para obtener más información, consulte el modelo de responsabilidad compartida de AWS.

Supervisión de Amazon EventBridge

EventBridge envía métricas a Amazon CloudWatch cada minuto para todo, desde el número de eventos coincidentes hasta el número de veces que una regla invoca un objetivo.

En el siguiente vídeo se analiza el EventBridge comportamiento de supervisión y auditoría mediante CloudWatch: Supervisión y auditoría de eventos

Temas

- EventBridge métricas
- Dimensiones de las EventBridge métricas

EventBridge métricas

El espacio de nombres de `AWS/Events` incluye las siguientes métricas.

En el caso de las métricas que utilizan el recuento como unidad, las estadísticas más útiles SampleCount suelen ser las más útiles.

Las métricas que especifican solo la `RuleName` dimensión se refieren al bus de eventos predeterminado. Las métricas que especifican tanto `EventBusName` las `RuleName` dimensiones como las hacen referencia a un bus de eventos personalizado.

Métrica	Descripción	Dimensiones	Unidades
`DeadLetterInvocations`	El número de veces que no se invoca el destino de una regla en respuesta a un evento. Esto incluye las invocaciones que harían que se activara la misma regla de nuevo, lo que provocaría un bucle infinito.	RuleName	Recuento
`Events`	El número de eventos asociados ingeridos por EventBridge.	EventSourceName	Recuento
`FailedInvocations`	El número de invocaciones fallidas permanentemente. No incluye las invocaciones que se	RuleName	Recuento

Métrica	Descripción	Dimensiones	Unidades
	reintentaron o que se realizaron correctam ente tras un reintento. Tampoco incluye las invocaciones fallidas que se cuentan en `DeadLetterInvocations` . **Note** EventBridge solo envía esta métrica a CloudWatch si no es cero.		
`Invocations`	El número de veces que una regla invoca un destino en respuesta a un evento. Incluye las invocaciones que se realizaron correcta e incorrectamente, pero no incluye los intentos limitados y los reintentos hasta que fallan permanentemente. No incluye `DeadLetterInvocations` . **Note** EventBridge solo envía esta métrica a CloudWatch si no es cero.	Ninguna, RuleName	Recuento
`InvocationAttempts`	Número de veces que se EventBridge intentó invocar un objetivo.	Ninguna	Recuento
`InvocationsCreated`	El número total de invocaciones creadas en respuesta a cada evento. Esta métrica se suele utilizar para supervisar la utilización del límite máximo de invocaciones en la cuota de servicio de transacciones por segundoEventBridge .	Ninguna	Recuento

Métrica	Descripción	Dimensiones	Unidades
`InvocationsFailedToBeSentToDlq`	Número de invocaciones que no se pudieron mover a una cola de mensajes fallidos. Pueden producirse errores de cola de mensajes fallidos debido a errores de permisos, recursos no disponibles o límites de tamaño. **Note** EventBridge solo envía esta métrica a CloudWatch si no es cero.	RuleName	Recuento
`IngestionToInvocationCompleteLatency`	El tiempo transcurrido desde la ingesta del evento hasta la finalización del primer intento de invocación exitoso.	EventBusName, Ninguno, RuleName	Milisegundos
`IngestionToInvocationStartLatency`	El tiempo necesario para procesar los eventos, medido desde el momento en que se ingiere un evento EventBridge hasta la primera invocación de un objetivo.	EventBusName, Ninguno, RuleName	Milisegundos
`InvocationsSentToDlq`	Número de invocaciones que se movieron a una cola de mensajes fallidos. **Note** EventBridge solo envía esta métrica a CloudWatch si no es cero.	RuleName	Recuento
`MatchedEvents`	Si se especifica EventBusName o EventSourceName se especifica, el número de eventos que coincidieron con cualquier regla. Si RuleName se especifica, el número de eventos que coincidieron con una regla específica.	EventBusName, EventSourceName, RuleName	Recuento

Métrica	Descripción	Dimensiones	Unidades
`RetryInvocationAttempts`	Número de veces que se ha vuelto a intentar invocar el destino. **Note** EventBridge solo envía esta métrica a CloudWatch si no es cero.	Ninguna	Recuento
`SuccessfulInvocationAttempts`	Número de veces que el destino se invocó correctamente.	Ninguna	Recuento
`ThrottledRules`	El número de veces que se limitó la ejecución de la regla. Es posible que la invocación de esas reglas se retrase. Para obtener más información, consulte la limitación de invocaciones en transacciones por segundo en ???.	EventBusName, Ninguno, RuleName	Recuento
`TriggeredRules`	El número de reglas que se han ejecutado y han coincidido con cualquier evento. No verás esta métrica CloudWatch hasta que se active una regla.	EventBusName, Ninguna, RuleName	Recuento

EventBridge PutEvents métricas

El espacio de nombres de `AWS/Events` incluye las siguientes métricas pertenecientes a las solicitudes API de `PutEvents`.

En el caso de las métricas que utilizan el recuento como unidad, las estadísticas más útiles SampleCount suelen ser las más útiles.

Métrica	Descripción	Dimensiones	Unidades
PutEventsApproximateCallCount	Número aproximado de solicitudes de PutEvents recibidas.	Ninguna	Recuento
PutEventsApproximateFailedCount	Número aproximado de solicitudes de PutEvents fallidas.	Ninguna	Recuento
PutEventsApproximateSuccessCount	Número aproximado de solicitudes de PutEvents realizadas correctamente.	Ninguna	Recuento
PutEventsApproximateThrottledCount	Número de solicitudes de PutEvents rechazadas debido a una limitación.	Ninguna	Recuento
PutEventsEntriesCount	El número de entradas de eventos incluidas en una solicitud de PutEvents.	Ninguna	Recuento
PutEventsFailedEntriesCount	El número de entradas de eventos incluidas en una solicitud de PutEvents que no se ingirieron.	Ninguna	Recuento
PutEventsLatency	El tiempo empleado por solicitud de PutEvents.	Ninguna	Milisegundos
PutEventsRequestSize	El tamaño de la solicitud de PutEvents.	Ninguna	Bytes

EventBridge PutPartnerEvents métricas

El espacio de nombres de `AWS/Events` incluye las siguientes métricas pertenecientes a las solicitudes API de `PutPartnerEvents`.

> **Note**
>
> EventBridge solo incluye métricas relacionadas con las PutPartnerEventssolicitudes en las cuentas de los socios de SaaS que envían eventos. Para obtener más información, consulte ???.

En el caso de las métricas que utilizan Count como unidad, Suma y SampleCount suele ser la estadística más útil.

Métrica	Descripción	Dimensiones	Unidades
`PutPartnerEventsApproximateCallCount`	Número aproximado de solicitudes de `PutPartnerEvents` recibidas.	Ninguna	Recuento
`PutPartnerEventsApproximateFailedCount`	Número aproximado de solicitudes de `PutPartnerEvents` fallidas.	Ninguna	Recuento
`PutPartnerEventsApproximateThrottledCount`	Número de solicitudes de `PutPartnerEvents` rechazadas debido a una limitación.	Ninguna	Recuento
`PutPartnerEventsAp`	Número aproximado de solicitudes de `PutPartnerEvents` realizadas correctamente.	Ninguna	Recuento

Métrica	Descripción	Dimensiones	Unidades
proximateSuccessCount			
PutPartnerEventsEntriesCount	El número de entradas de eventos incluidas en una solicitud de PutPartnerEvents .	Ninguna	Recuento
PutPartnerEventsFailedEntriesCount	El número de entradas de eventos incluidas en una solicitud de PutPartnerEvents que no se ingirieron.	Ninguna	Recuento
PutPartnerEventsLatency	El tiempo empleado por solicitud de PutPartnerEvents .	Ninguna	Milisegundos

Dimensiones de las EventBridge métricas

EventBridge Las métricas tienen dimensiones o atributos que se pueden ordenar, que se muestran a continuación.

Dimensión	Descripción
EventBusName	Filtra las métricas disponibles por nombre de bus de eventos.
EventSourceName	Filtra las métricas disponibles por nombre de origen de eventos de socios.
RuleName	Filtra las métricas disponibles por nombre de regla.

Solución de problemas de Amazon EventBridge

Puedes seguir los pasos de esta sección para solucionar los problemas de Amazon EventBridge.

Temas

Mi regla se ejecutó pero no se invocó mi función de Lambda

Una de las razones por las que es posible que la función de Lambda no se ejecute es si no tiene los permisos adecuados.

Para comprobar los permisos de la función de Lambda

1. Con el AWS CLI, ejecuta el siguiente comando con tu función y tu AWS región:

```
aws lambda get-policy --function-name MyFunction --region us-east-1
```

Debería ver la siguiente salida.

```
{
    "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. Si ve el siguiente mensaje.

```
A client error (ResourceNotFoundException) occurred when calling the GetPolicy
 operation: The resource you requested does not exist.
```

O si ve la salida, pero no puede localizar events.amazonaws.com como entidad de confianza en la política, ejecute el siguiente comando:

```
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. Si la salida contiene un campo `SourceAccount`, debe eliminarlo. Un `SourceAccount` ajuste EventBridge impide que se pueda invocar la función.

> **Note**
>
> Si la política es incorrecta, puede editar la regla en la EventBridge consola quitándola y volviéndola a añadir a la regla. A continuación, la EventBridge consola establece los permisos correctos en el destino.
> Si utiliza un alias o versión de Lambda específica, agregue el parámetro `--qualifier` en los comandos `aws lambda get-policy` y `aws lambda add-permission`, como se muestra en el siguiente comando.
>
> ```
> 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
> ```

Acabo de crear o modificar una regla, pero no coincidió con un evento de prueba

Al realizar un cambio en una regla o en sus destinos, los eventos entrantes podrían no comenzar o parar de inmediato la asignación a reglas nuevas o actualizadas. Espere un breve periodo para que los cambios surtan efecto.

Si los eventos siguen sin coincidir tras un breve periodo de tiempo, comprueba las CloudWatch métricas `TriggeredRules` y `FailedInvocations` comprueba tu regla. `Invocations` Para obtener más información sobre estas métricas, consulta Monitoring Amazon EventBridge.

Si la regla pretende coincidir con un evento de un AWS servicio, realiza una de las siguientes acciones:

- Use la acción `TestEventPattern` para comprobar si el patrón de eventos de la regla coincide con un evento de prueba. Para obtener más información, consulta TestEventPatternla referencia de la EventBridge API de Amazon.
- Usa el Sandbox de la EventBridge consola.

Mi regla no se ejecutó en el momento que especifiqué en la `ScheduleExpression`

Asegúrese de que ha establecido la programación para la regla en la zona horaria UTC+0. Si la `ScheduleExpression` es correcta siga, a continuación, los pasos indicados en Acabo de crear o modificar una regla, pero no coincidió con un evento de prueba.

Mi regla no se ejecutó a la hora esperada

EventBridge ejecuta las reglas en un minuto a partir de la hora de inicio que establezcas. La cuenta atrás hasta la hora de ejecución comienza en cuanto se crea la regla.

> **Note**
>
> Las reglas programadas tienen un tipo de entrega de `guaranteed`, lo que significa que los eventos se desencadenarán para cada hora prevista al menos una vez.

Puede utilizar una expresión cron para invocar destinos a una hora especificada. Para crear una regla que se ejecute cada cuatro horas en el minuto 0, siga uno de estos procedimientos:

- En la EventBridge consola, se utiliza la expresión `0 0/4 * * ? *` cron.
- Al usar el AWS CLI, se usa la expresión`cron(0 0/4 * * ? *)`.

Por ejemplo, para crear una regla denominada `TestRule` que se ejecute cada 4 horas mediante el AWS CLI, utilice el siguiente comando.

```
aws events put-rule --name TestRule --schedule-expression 'cron(0 0/4 * * ? *)'
```

Para ejecutar una regla cada cinco minutos, se utiliza la siguiente expresión de cron.

```
aws events put-rule --name TestRule --schedule-expression 'cron(0/5 * * * ? *)'
```

La resolución más precisa para una EventBridge regla que utiliza una expresión cron es de un minuto. La regla programada se ejecuta dentro de ese minuto, pero no específicamente en el segundo 0.

Como EventBridge los servicios de destino están distribuidos, puede haber un retraso de varios segundos entre el momento en que se ejecuta la regla programada y el momento en que el servicio de destino realiza la acción en el recurso de destino.

Mi regla coincide con las llamadas a la API del servicio AWS global, pero no se ejecutó

AWS Los servicios globales, como IAM y Amazon Route 53, solo están disponibles en la región EE. UU. Este (Virginia del Norte), por lo que los eventos de las llamadas a la AWS API desde servicios globales solo están disponibles en esa región. Para obtener más información, consulte Eventos de los AWS servicios.

El rol de IAM asociado a mi regla se ignora cuando se ejecuta la regla

EventBridge solo usa roles de IAM para las reglas que envían eventos a las transmisiones de Kinesis. Para reglas que invocan funciones de Lambda o temas de Amazon SNS, debe proporcionar permisos basados en recursos.

Asegúrese de que sus AWS STS puntos de enlace regionales estén habilitados para EventBridge poder usarlos cuando asuma la función de IAM que proporcionó. Para obtener más información, consulte Activación y desactivación AWS STS en una AWS región en la Guía del usuario de IAM.

Mi regla tiene un patrón de eventos que se supone que coincide con un recurso, pero ningún evento coincide

La mayoría de los servicios AWS tratan los dos puntos (:)) o una barra (/) como el mismo carácter en los nombres de recursos de Amazon (ARN)., pero EventBridge utilizan una coincidencia exacta en los patrones de eventos y las reglas. Asegúrese de usar los caracteres de ARN correctos cuando cree patrones de eventos, de modo que se ajusten a la sintaxis de ARN del evento de la correspondencia.

Algunos eventos, como los eventos de llamada a la AWS API desde CloudTrail, no tienen ningún elemento en el campo de recursos.

La entrega de mi evento al destino sufrió un retraso

EventBridge intenta enviar un evento a un objetivo durante un máximo de 24 horas, excepto en situaciones en las que los recursos de destino son limitados. El primer intento se realiza en cuanto el evento llega en el flujo de transmisión. Si el servicio de destino tiene problemas, EventBridge reprograma automáticamente otra entrega. Si han pasado 24 horas desde la llegada del evento, EventBridge deja de intentar realizar el evento y publica la `FailedInvocations` métrica en él. CloudWatch Le recomendamos que configure una DLQ para almacenar los eventos que no se hayan podido entregar correctamente a un destino. Para obtener más información, consulte Uso de colas con letra muerta para procesar los eventos no entregados.

Algunos eventos no se entregaron en mi destino

Si el objetivo de una EventBridge regla está restringido durante un tiempo prolongado, es EventBridge posible que no vuelva a intentar la entrega. Por ejemplo, si el destino no está aprovisionado para gestionar el tráfico de eventos entrante y el servicio de destino limita las solicitudes que se realizan en tu nombre, es posible EventBridge que no vuelva EventBridge a intentar la entrega.

Mi regla se ejecutó más de una vez en respuesta a un único evento

En casos excepcionales, la misma regla se puede ejecutar más de una vez para un solo evento o tiempo programado, o el mismo destino se puede invocar más de una vez para una regla activada determinada.

Prevención de bucles infinitos

En EventBridge, es posible crear una regla que dé lugar a bucles infinitos, en los que la regla se ejecute repetidamente. Si tiene una regla que provoca un bucle infinito, reescríbala para que las acciones que lleve a cabo la regla no coincidan con la misma regla.

Por ejemplo, una regla que detecta que las ACL han cambiado en un bucket de Amazon S3 y, a continuación, ejecuta un software para cambiarlas a un nuevo estado provoca un bucle infinito. Una forma de resolverlo es reescribir la regla para que solo coincida con las ACL que estén en mal estado.

Un bucle infinito puede generar cargos superiores a los esperados rápidamente. Le recomendamos que utilice la función de presupuestos, que le avisa cuando los cargos superan el límite especificado. Para obtener más información, consulte Gestión de costos con presupuestos.

Mis eventos no se entregan en la cola de Amazon SQS de destino

Si su cola de Amazon SQS está cifrada, debe crear una clave de KMS gestionada por el cliente e incluir la siguiente sección de permisos en su política de claves de KMS. Para obtener más información, consulte Configuración de AWS KMS permisos.

```
{
  "Sid": "Allow EventBridge to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

Mi regla se ejecuta, pero no veo ningún mensaje publicado en mi tema de Amazon SNS

Escenario 1

Necesita permiso para que los mensajes se publiquen en su tema de Amazon SNS. Usa el siguiente comando con el AWS CLI, sustituyendo us-east-1 por tu región y usando el ARN del tema.

```
aws sns get-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic"
```

Para tener el permiso correcto, los atributos de su política son similares a los siguientes.

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

Si no aparece `events.amazonaws.com` con el permiso `Publish` en su política, primero copie la política actual y añada la siguiente declaración a la lista de declaraciones.

```
{\"Sid\":\"Allow_Publish_Events\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}
```

A continuación, defina los atributos del tema mediante AWS CLI el comando siguiente.

```
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**
>
> Si la política es incorrecta, también puedes editar la regla en la EventBridge consola quitándola y volviéndola a añadir a la regla. EventBridge establece los permisos correctos en el destino.

Escenario 2

Si su tema de SNS está cifrado, debe incluir la siguiente sección en su política de claves de KMS.

```
{
  "Sid": "Allow EventBridge to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

Mi tema de Amazon SNS sigue teniendo permisos EventBridge incluso después de haber eliminado la regla asociada al tema de Amazon SNS

Cuando crea una regla con Amazon SNS como destino, EventBridge añade permiso a su tema de Amazon SNS en su nombre. Si eliminas la regla poco después de crearla, es EventBridge posible que no elimines el permiso de tu tema de Amazon SNS. Si esto ocurre, puede eliminar el permiso desde el tema utilizando el comando `aws sns set-topic-attributes`. Para obtener más información acerca de los permisos basados en recursos para enviar eventos, consulte Uso de políticas basadas en recursos para Amazon EventBridge.

¿Con qué claves de condición de IAM puedo usar? EventBridge

EventBridge admite las claves AWS de condición generales (consulte las claves de IAM y de contexto de AWS STS condición en la Guía del usuario de IAM), además de las claves que se enumeran en. Usar condiciones de las políticas de IAM para control de acceso preciso

¿Cómo puedo saber cuándo se infringen EventBridge las reglas?

Puede usar la siguiente alarma para avisarle cuando se infrinjan sus EventBridge reglas.

Para crear una alarma que avise cuando se infrinjan las reglas

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.

2. Seleccione Crear alarma. En el panel CloudWatch Métricas por categoría, elija Events Metrics.
3. En la lista de métricas, selecciona FailedInvocations.
4. Encima del gráfico, seleccione Statistic, Sum.
5. En Período, seleccione un valor; por ejemplo, 5 minutos. Seleccione Siguiente.
6. En Umbral de alarma, en Nombre, escriba un nombre exclusivo para la alarma, por ejemplo myFailedRules. En Descripción, escriba una descripción de la alarma; por ejemplo, Reglas que no proporcionan eventos a los destinos.
7. En es, seleccione >= y 1. En para, escriba 10.
8. En Acciones, en Siempre que esta alarma, seleccione El estado es ALARMA.
9. En Enviar notificación a, seleccione un tema de Amazon SNS existente o cree uno nuevo. Para crear un nuevo tema, seleccione Lista nueva. Escriba un nombre para el nuevo tema de Amazon SNS, por ejemplo:. myFailedRules
10. En Lista de correos electrónicos, escriba una lista separada por comas con las direcciones de correo electrónico a las que se van a enviar notificaciones cuando la alarma cambie al estado ALARMA.
11. Seleccione Crear alarma.

Cuotas de Amazon EventBridge

Existen cuotas para la mayoría de los aspectos de EventBridge.

Temas

- Cuotas de EventBridge
- Cuotas de PutPartnerEvents por región
- Cuotas del registro de esquemas de EventBridge
- Cuotas de EventBridge Pipes

> **Note**
>
> Para obtener una lista de las cuotas del Programador de EventBridge, consulte Cuotas del Programador de EventBridge en la Guía del usuario del Programador de EventBridge.

Cuotas de EventBridge

EventBridge tiene las siguientes cuotas.

La consola de Service Quotas proporciona información sobre las cuotas de EventBridge. Además de ver las cuotas predeterminadas, puede utilizar la consola de Service Quotas para solicitar aumentos de cuota para cuotas ajustables.

Nombre	Valor predeterminado	Ajusta	Descripción
Destinos de la API	Cada región admitida: 3 000	Sí	Número máximo de destinos de la API por cuenta y por región.
Conexiones	Cada región admitida: 3000	Sí	Número máximo de conexiones por cuenta y por región.

Nombre	Valor predeterminado	Ajusta	Descripción
Limitación CreateEndpoint en transacciones por segundo	Cada región admitida: 5 por segundo	No	Número máximo de solicitudes de por segundo para la API CreateEndpoint. Las solicitudes adicionales se limitan.
Límitación DeleteEndpoint en transacciones por segundo	Cada región admitida: 5 por segundo	No	Número máximo de solicitudes por segundo para la API DeleteEndpoint. Las solicitudes adicionales se limitan.
puntos de conexión	Cada región admitida: 100	Sí	Número máximo de puntos de conexión por cuenta y por región.
Tamaño de la política de bus de eventos	Cada región admitida: 10 240	Sí	Tamaño máximo de la política, en caracteres. Este tamaño de política aumenta cada vez que concede acceso a otra cuenta. Puede ver su política actual y su tamaño mediante la API DescribEventBus.
Buses de eventos	Cada región admitida: 100	Sí	Número máximo de buses de eventos por cuenta.
Tamaño del patrón de eventos	Cada región admitida: 2 048	Sí	Tamaño máximo de un patrón de eventos, en caracteres.

Nombre	Valor predeterminado	Ajusta	Descripción
Limitación de invocaciones en transacciones por segundo	us-east-1: 18 750 por segundo us-east-2: 4500 por segundo us-west-2: 2250 por segundo us-west-2: 18 750 por segundo eu-south-1: 750 por segundo ap-northeast-1: 2250 por segundo ap-northeast-3: 750 por segundo ap-southeast-1: 2250 por segundo ap-southeast-2: 2250 por segundo ap-southeast-3: 750 por segundo eu-central-1: 4500 por segundo eu-south-1: 750 por segundo	Sí	Una invocación es un evento que coincide con una regla y que se envía a los destinos de la regla. Una vez alcanzado el límite, las invocaciones se limitan; es decir, siguen produciéndose pero se retrasan.

Nombre	Valor predeterminado	Ajusta	Descripción
	us-west-1: 18 750 por segundo us-west-2: 2250 por segundo Cada una de las demás regiones compatibles: 1100 por segundo		
Número de regalas	af-south-1: 100 eu-south-1: 100 Cada una de las demás regiones compatibles: 300	Sí	Número máximo de reglas que una cuenta puede tener por bus de eventos

Nombre	Valor predeterminado	Ajusta	Descripción
Limitación de PutEvents en transacciones por segundo	us-east-1: 10 000 por segundo us-east-2: 2400 por segundo us-west-1: 1200 por segundo us-west-2: 10 000 por segundo eu-south-1: 400 por segundo ap-northeast-1: 1200 por segundo ap-northeast-3: 400 por segundo ap-southeast-1: 1200 por segundo ap-southeast-2: 1200 por segundo ap-southeast-3: 400 por segundo eu-central-1: 2400 por segundo eu-south-1: 400 por segundo	Sí	Número máximo de solicitudes por segundo para la API PutEvents. Las solicitudes adicionales se limitan.

Nombre	Valor predeterminado	Ajusta	Descripción
	us-west-1: 10 000 por segundo us-west-2: 1200 por segundo Cada una de las demás regiones compatibles: 600 por segundo		
Tasa de invocaciones por destino de la API	Cada región admitida: 300 por segundo	Sí	Número máximo de invocaciones por segundo que se enviarán a cada punto de conexión de destino de la API por cuenta y región. Una vez que se alcance la cuota, se limitarán las futuras invocaciones a ese punto de conexión de la API. Las invocaciones seguirán produciéndose, pero se retrasarán.
Destinos por regla	Cada región admitida: 5	No	Número máximo de destinos que pueden asociarse a una regla

Nombre	Valor predeterminado	Ajusta	Descripción
Limitación en transacciones por segundo	Cada región admitida: 50 por segundo	Sí	Número máximo de solicitudes por segundo para todas las operaciones de la API de EventBridge, excepto PutEvents. Las solicitudes adicionales se limitan
Limitación de UpdateEndpoint en transacciones por segundo	Cada región admitida: 5 por segundo	No	Número máximo de solicitudes por segundo para la API UpdateEndpoint. Las solicitudes adicionales se limitan.

Además, EventBridge tiene las siguientes cuotas que no se administran a través de la consola de Service Quotas.

Nombre	Predeterminado	Descripción
Buses de eventos	Cada región admitida: 100	Número máximo de buses de eventos por cuenta.
Tamaño de la política de bus de eventos	Cada región admitida: 10 240	Tamaño máximo de la política, en caracteres. Este tamaño de política aumenta cada vez que concede acceso a otra cuenta. Puede ver su política actual y su tamaño mediante la API `DescribeEventBus` .
Tamaño del patrón de eventos	Cada región admitida: 2 048	Tamaño máximo de un patrón de eventos, en caracteres.

Nombre	Predeterminado	Descripción
		Se puede ajustar hasta 4 096 caracteres. Si necesita un rendimiento máximo, póngase en contacto con el servicio de asistencia.
Reglas que contienen caracteres comodín	Cada región admitida: 30 reglas por bus de eventos	Número máximo de reglas, por bus de eventos y por cuenta, que pueden contener filtros de eventos que incluyan caracteres comodín. Esta cuota no se puede ajustar. Para obtener más información sobre el uso de caracteres comodín en los patrones de eventos, consulte ???.
Niveles de detección de esquemas	Cada región admitida: 255 niveles	Número máximo de niveles; la detección de esquemas inferirá los eventos anidados. Se ignora cualquier evento que supere 255 niveles.

Cuotas de PutPartnerEvents por región

Si necesita un rendimiento máximo, póngase en contacto con el servicio de asistencia.

Regiones	Transacciones por segundo
• AWS GovCloud (Oeste de EE. UU.) • AWS GovCloud (Este de EE. UU.) • Este de EE. UU. (Norte de Virginia) • Este de EE. UU. (Ohio) • Oeste de EE.UU. (Norte de California) • Oeste de EE. UU. (Oregón) • África (Ciudad del Cabo)	PutPartnerEvents tiene un límite flexible de 1 400 solicitudes de rendimiento por segundo y de 3 600 solicitudes de ráfagas por segundo de forma predeterminada en todas las regiones.

Regiones	Transacciones por segundo
• Asia-Pacífico (Hong Kong) • Asia-Pacífico (Bombay) • Asia-Pacífico (Osaka) • Asia-Pacífico (Seúl) • Asia-Pacífico (Singapur) • Asia-Pacífico (Sídney) • Asia-Pacífico (Tokio) • Canadá (Centro) • Europa (Fráncfort) • Europa (Irlanda) • Europa (Londres) • Europa (Milán) • Europa (París) • Europa (Estocolmo) • Europa (Milán) • América del Sur (São Paulo) • China (Ningxia) • China (Pekín)	

Cuotas del registro de esquemas de EventBridge

El registro de esquemas de EventBridge tiene las siguientes cuotas.

La consola de Service Quotas proporciona información sobre las cuotas de EventBridge. Además de ver las cuotas predeterminadas, puede utilizar la consola de Service Quotas para solicitar aumentos de cuota para cuotas ajustables.

Nombre	Valor predeterminado	Ajusta	Descripción
DiscoveredSchemas	Cada región admitida: 200	Sí	Número máximo de esquemas para un registro de esquemas detectado que puede crear en la región actual
Discoverers	Cada región admitida: 10	Sí	Número máximo de programas de detección que puede crear en la región actual.
Registries	Cada región admitida: 10	Sí	Número máximo de registros que puede crear en la región actual.
SchemaVersions	Cada región admitida: 100	Sí	Número máximo de versiones por esquema que puede crear en la región actual.
Schemas	Cada región admitida: 100	Sí	Número máximo de esquemas por registro que puede crear en la región actual. (Excepto el registro de esquemas detectados)

Cuotas de EventBridge Pipes

EventBridge Pipes tiene las siguientes cuotas. Si necesita un rendimiento máximo, póngase en contacto con el servicio de asistencia.

Resource	Regiones	Límite predeterminado
Ejecuciones de canalización simultáneas por cuenta	• AWS GovCloud (EE. UU. Oeste) • AWS GovCloud (Este de EE. UU.) • China (Ningxia) • China (Pekín) • Asia-Pacífico (Osaka) • África (Ciudad del Cabo) • Europa (Milán) • Este de EE. UU. (Ohio) • Europa (Fráncfort) • Oeste de EE. UU. (Norte de California) • Europa (Londres) • Asia-Pacífico (Sídney) • Asia-Pacífico (Tokio) • Asia-Pacífico (Singapur) • Canadá (Centro) • Europa (París) • Europa (Estocolmo) • América del Sur (São Paulo) • Asia-Pacífico (Seúl) • Asia-Pacífico (Bombay) • Asia-Pacífico (Hong Kong) • Medio Oriente (Baréin) • China (Ningxia) • China (Pekín) • Asia-Pacífico (Osaka) • África (Ciudad del Cabo)	1 000

Resource	Regiones	Límite predeterminado
	• Europa (Milán)	
Ejecuciones de canalización simultáneas por cuenta	• Este de EE. UU. (Norte de Virginia) • Oeste de EE. UU. (Oregón) • Europa (Irlanda)	3 000
Canalizaciones por cuenta	Todos	1 000

EventBridge Etiquetas de Amazon

Una etiqueta es una etiqueta de atributo personalizada que tú o AWS asignas a un AWS recurso. En EventBridge, puede asignar etiquetas a los buses de reglas y eventos. Cada recurso puede tener un máximo de 50 etiquetas.

Las etiquetas se utilizan para identificar y organizar AWS los recursos. Muchos AWS servicios admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados. Por ejemplo, puede asignar la misma etiqueta a una EventBridge regla que a una instancia de EC2.

Una etiqueta consta de dos partes:

- Una clave de etiqueta, por ejemplo, `CostCenter`, `Environment`, o `Project`.
  - Las claves de etiqueta distinguen entre mayúsculas y minúsculas.
  - La longitud máxima de la clave de etiqueta es de 128 caracteres Unicode en UTF-8.
  - Para cada recurso, cada clave de etiqueta debe ser única.
  - Los caracteres permitidos son letras, números y espacios representables en UTF-8, además de los siguientes caracteres: . : + = @ _ / - (guion).
  - El `aws:` prefijo está prohibido para las etiquetas porque está reservado para AWS su uso. Las claves y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas que tengan este prefijo no cuentan para el límite de etiquetas por recurso.
- Un campo de valor de etiqueta opcional, por ejemplo, `111122223333` o `Production`.
  - Cada clave de etiqueta solo puede tener un valor.
  - Los valores de la etiqueta distinguen entre mayúsculas y minúsculas.
  - Omitir el valor de etiqueta es lo mismo que utilizar una cadena vacía.
  - La longitud máxima del valor de etiqueta es de 256 caracteres Unicode en UTF-8.
  - Los caracteres permitidos son letras, números y espacios representables en UTF-8, además de los siguientes caracteres: . : + = @ _ / - (guion).

> **Tip**
>
> Como práctica recomendada, decida una estrategia de uso de mayúsculas y minúsculas en las etiquetas e implemente esa estrategia sistemáticamente en todos los tipos de recursos.

Por ejemplo, decida si se va a utilizar `Costcenter`, `costcenter` o `CostCenter` y, a continuación, utilice la misma convención para todas las etiquetas.

Puedes usar la EventBridge consola, la EventBridge API o la AWS CLI para añadir, editar o eliminar etiquetas. Para más información, consulte los siguientes temas:

- TagResourceUntagResource, y ListTagsForResourceen la referencia de la EventBridge API de Amazon
- tag-resource, untag-resource y en la Referencia list-tags-for-resourceAWS CLI
- Uso del editor de etiquetas en la Guía del usuario de Resource Groups

Historial de documentos

En la siguiente tabla se describen los cambios importantes en cada versión de la Guía del EventBridge usuario de Amazon, a partir de julio de 2019. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha de lanzamiento
Políticas AWS gestionadas actualizadas.	AWS GovCloud (US) Regions únicamente `AmazonEventBridgeFullAccess` y `AmazonEventBridgeSchemasFullAccess` las políticas no lo incluyen `iam:CreateServiceLinkedRole` , ya que no se utiliza. • the section called “Actualizaciones de políticas”	9 de mayo de 2024
Genere AWS CloudFormation plantillas a partir de autobuses y reglas de eventos.	Ahora puedes generar AWS CloudFormation plantillas a partir de tus autobuses y reglas de EventBridge eventos de Amazon existentes. • Generar una plantilla de AWS CloudFormation a partir de un bus de eventos de Amazon EventBridge	18 de noviembre de 2022
Se lanzó la documentación de EventBridge Pipes.	Ahora puede crear canalizaciones para conectar las fuentes con destinos, con filtros y enriquecimientos opcionales. • Canalizaciones	1 de diciembre de 2022
Genera AWS CloudFormation plantillas a partir de autobuses y reglas de eventos.	Ahora puedes generar AWS CloudFormation plantillas a partir de tus autobuses y reglas de EventBridge eventos de Amazon existentes. • Generar una plantilla de AWS CloudFormation a partir de un bus de eventos de Amazon EventBridge	18 de noviembre de 2022

Cambio	Descripción	Fecha de lanzamiento
Se agregó la AmazonEventBridgePipesFullAccess política.	Proporciona acceso completo a Amazon EventBridge Pipes. • EventBridge Políticas gestionadas específicas de Pipes	1 de diciembre de 2022
Se agregó la AmazonEventBridgePipesReadOnlyAccess política.	Proporciona acceso de solo lectura a Amazon EventBridge Pipes. • EventBridge Políticas gestionadas específicas de Pipes	1 de diciembre de 2022
Se agregó la política AmazonEventBridgePipesOperatorAccess .	Proporciona acceso de solo lectura y mediante operador (es decir, la capacidad de detener y empezar a ejecutar Pipes) a Amazon EventBridge Pipes. • EventBridge Políticas gestionadas específicas de Pipes	1 de diciembre de 2022
Se actualizó la política CloudWatchEventsFullAccess .	Se actualizó para que coincidiera con `AmazonEventBridgeFullAccess` . • AmazonEventBridgeFullAccess política	1 de diciembre de 2022
Se actualizó la CloudWatchEventsReadOnlyAccess política.	Se actualizó para que coincidiera con `AmazonEventBridgeReadOnlyAccess` . • AmazonEventBridgeReadOnlyAccess política	1 de diciembre de 2022

Cambio	Descripción	Fecha de lanzamiento
Se ha actualizado el filtrado de contenido en los patrones de eventos.	Ahora puede usar las opciones de filtrado `suffix`, `equals-ignore-case` y `$or` para crear patrones de eventos. • Filtrado de contenido en los patrones de EventBridge eventos de Amazon	14 de noviembre de 2022
Se actualizó la AmazonEventBridgeFullAccess política.	Se agregaron los permisos necesarios para usar EventBridge Schema Registry y EventBridge Scheduler. • AmazonEventBridgeFullAccess política	10 de noviembre de 2022
Se actualizó la AmazonEventBridgeReadOnlyAccess política.	Ahora puede ver la información del registro de EventBridge esquemas y del EventBridge programador. • AmazonEventBridgeReadOnlyAccess política	10 de noviembre de 2022
Se ha actualizado el filtrado de contenido en los patrones de eventos.	Ahora puede usar las opciones de filtrado `suffix`, `equals-ignore-case` y `$or` para crear patrones de eventos. • Filtrado de contenido en los patrones de EventBridge eventos de Amazon	14 de noviembre de 2022
Se actualizó la AmazonEventBridgeFullAccess política.	Se agregaron los permisos necesarios para usar EventBridge Schema Registry y EventBridge Scheduler. • AmazonEventBridgeFullAccess política	10 de noviembre de 2022

Cambio	Descripción	Fecha de lanzamiento
Se actualizó la AmazonEventBridgeReadOnlyAccess política.	Ahora puede ver la información del registro de EventBridge esquemas y del EventBridge programador. • AmazonEventBridgeReadOnlyAccess política	10 de noviembre de 2022
Se actualizó la AmazonEventBridgeReadOnlyAccess política.	Ahora puede ver la información del punto de conexión. • AmazonEventBridgeReadOnlyAccess política	7 de abril de 2022
Se ha agregado compatibilidad para puntos de conexión globales.	Amazon EventBridge ahora admite el uso de puntos de enlace globales para ayudar a que su aplicación sea tolerante a los errores regionales sin coste adicional. Para obtener más información, consulte lo siguiente: • Tolerancia de las aplicaciones a los errores regionales con puntos de conexión globales y replicación de eventos • `CreateEndpoint`	7 de abril de 2022
Se ha agregado compatibilidad para archivos y reproducciones de eventos.	Amazon EventBridge ahora admite el uso de archivos para almacenar eventos y las repeticiones de eventos para reproducir los eventos de un archivo. Para obtener más información, consulte lo siguiente: • Archivar eventos de Amazon EventBridge . • `CreateArchive` • `StartReplay`	5 de noviembre de 2020

Cambio	Descripción	Fecha de lanzamiento
Se ha añadido compatibilidad para colas de mensajes fallidos y política de reintentos para los destinos.	Amazon EventBridge ahora admite el uso de colas de letra muerta y la definición de una política de reintentos para los objetivos. Para obtener más información, consulte lo siguiente: • Uso de colas con letra muerta para procesar los eventos no entregados. • `PutTargets`	12 de octubre de 2020
Se ha agregado compatibilidad para los esquemas en formato JSONSchema Draft4.	Amazon EventBridge ahora admite esquemas en formato JSONSchema Draft 4. Ahora también puede exportar esquemas mediante la API. EventBridge Para obtener más información, consulte lo siguiente. • EventBridge Esquemas de Amazon • `Export`en la referencia de la API EventBridge de Schema Registry.	28 de septiembre de 2020
Políticas basadas en recursos para el registro de esquemas EventBridge	El Amazon EventBridge Schema Registry ahora admite políticas basadas en recursos. Para obtener más información, consulte lo siguiente. • Políticas basadas en recursos para los esquemas de Amazon EventBridge • `Policy`en la referencia de la API de EventBridge Schema Registry • RegistryPolicy Tipo de recurso en la guía AWS CloudFormation del usuario	30 de abril de 2020

Cambio	Descripción	Fecha de lanzamiento
Etiquetas para los buses de eventos	Esta versión le permite crear y administrar etiquetas para buses de eventos. Puede agregar etiquetas al crear un bus de eventos y agregar o administrar etiquetas existentes llamando a la API relacionada. Para obtener más información, consulte lo siguiente. • EventBridge Etiquetas de Amazon • Políticas basadas en etiquetas • `TagResource` • `UntagResource` • `ListTagsForResource`	24 de febrero de 2020
Se han aumentado las cuotas de servicios	Amazon EventBridge ha aumentado las cuotas de invocaciones y para`PutEvents` . Las cuotas varían según la región y pueden aumentarse si fuese necesario.	11 de febrero de 2020

Cambio	Descripción	Fecha de lanzamiento
Se ha agregado un nuevo tema sobre la transformación de la entrada de destino y se ha agregado un enlace a los eventos de escalado automático.	Documentación mejorada sobre el transformador de entrada. • Transformación EventBridge de entradas de Amazon • Utilizar el transformador de entrada para extraer datos de un evento y especificar esos datos en el destino • Tutorial: Uso del transformador de entrada para personalizar lo que EventBridge transfiere al destino de eventos Se ha agregado un enlace a los eventos de escalado automático. • Eventos de Application Auto Scaling y EventBridge • Eventos de los AWS servicios	20 de diciembre de 2019
Filtrado basado en el contenido		19 de diciembre de 2019
Se han agregado enlaces a ejemplos de eventos de Amazon Augmented AI.	Se ha añadido un enlace al tema Amazon Augmented AI en la Guía para SageMaker desarrolladores de Amazon que ofrece ejemplos de eventos de Amazon Augmented AI. Para obtener más información, consulte lo siguiente. • Usar eventos en Amazon Augmented AI • Eventos de los AWS servicios	13 de diciembre de 2019

Cambio	Descripción	Fecha de lanzamiento
Se han agregado enlaces a ejemplos de eventos de Amazon Chime.	Se ha agregado un enlace al tema de Amazon Chime que proporciona eventos de ejemplo para ese servicio. Para obtener más información, consulte lo siguiente. • Automatizar Amazon Chime con EventBridge • Eventos de los AWS servicios	12 de diciembre de 2019
EventBridge Esquemas de Amazon	Ahora puede gestionar esquemas y generar enlaces de código para eventos en Amazon. EventBridge Para obtener más información, consulte lo siguiente. • EventBridge Esquemas de Amazon • EventBridge Referencia de la API de esquemas • EventSchemas Referencia de tipo de recurso en AWS CloudFormation	1 de diciembre de 2019
AWS CloudFormation soporte para autobuses de eventos	AWS CloudFormation ahora es compatible con el EventBus recurso. También admite el EventBusName parámetro tanto en el recurso como en EventBusPolicy el recurso Rule. Para obtener más información, consulte Amazon EventBridge Resource Type Reference.	7 de octubre de 2019
Nuevo servicio	Versión inicial de Amazon EventBridge.	11 de julio de 2019

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.