aws

Guía del usuario

EventBridge Programador

EventBridge Programador: Guía del usuario

Table of Contents

¿Qué es el Programador de 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. Altamente escalable, el Programador de EventBridge le permite programar millones de tareas que pueden invocar más de 270 servicios de AWS y más de 6000 operaciones de API. Sin necesidad de aprovisionar y gestionar la infraestructura, ni de integrarse con varios servicios, el Programador de EventBridge le ofrece la posibilidad de ofrecer programaciones a escala y reducir los costes de mantenimiento.

El Programador de EventBridge entrega sus tareas de forma fiable, con mecanismos integrados que ajustan sus programaciones en función de la disponibilidad de los destinos posteriores. Con el Programador de EventBridge, puede crear programadores mediante expresiones cron y rate 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 activadores fallidos.

Temas

Características principales del Programador de EventBridge

El Programador de EventBridge ofrece las siguientes funciones clave que puede utilizar para configurar los destinos y escalar sus programaciones.

- Destinos con plantillas: el Programador de EventBridge admite destinos con plantillas para realizar operaciones de API comunes mediante Amazon SQS, Amazon SNS, Lambda y EventBridge. Con los destinos predefinidos, puede configurar sus programaciones rápidamente mediante la consola del Programador de EventBridge, el SDK del Programador de EventBridge o la AWS CLI.
- Destinos universales: el Programador de EventBridge proporciona un parámetro de destino universal (UTP) que puede utilizar para crear activadores personalizados que se dirijan a más de 270 servicios de AWS y más de 6000 operaciones de API de forma programada. Con UTP, puede configurar sus activadores personalizados mediante la consola del Programador de EventBridge, el SDK de el Programador de EventBridge o la AWS CLI.

- Ventanas temporales flexibles: el Programador de EventBridge admite ventanas temporales flexibles, lo que le permite dispersar sus programaciones y mejorar la fiabilidad de los activadores para casos de uso que no requieren una invocación de destinos programada y precisa.
- Reintentos: el Programador de EventBridge proporciona una entrega de eventos a los destinos al menos una vez, lo que significa que al menos una entrega se realiza correctamente con una respuesta del destino. El Programador de EventBridge le permite establecer el número de reintentos de su programación para una tarea fallida. El Programador de EventBridge reintenta las tareas fallidas con intentos retrasados para mejorar la fiabilidad de la programación y garantizar la disponibilidad de los destinos.

Acceder al programador de EventBridge

Puede utilizar el Programador de EventBridge a través de la consola del Programador de EventBridge, el SDK del Programador de EventBridge o directamente mediante la API AWS CLI del Programador de EventBridge.

Configurar el Programador de Amazon EventBridge

Antes de poder utilizar el Programador de EventBridge, debe llevar a cabo los pasos siguientes.

Temas

Registrarse en AWS

Si no dispone de una Cuenta de AWS, siga los pasos que figuran a continuación para crear una.

Para registrarse en Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones en línea.

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

   Al registrase en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tiene acceso a todos los recursos y Servicios de AWS de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario administrativo y utilice únicamente el usuario raíz para realizar la ejecución tareas que requieren acceso de usuario raíz.

Creación de un usuario de IAM

Para crear un usuario administrador, elija una de las siguientes opciones.

Elegir una forma de administrar el administrador	Para	B	También puede
En Centro de identidades de IAM (Recomendado)	Use credenciales a corto plazo para acceder a AWS. Esto se alinea con las prácticas recomendadas de seguridad. Para obtener información sobre las prácticas recomendadas, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.	Siga las instrucciones en Introducción en la Guía del usuario de AWS IAM Identity Center.	Configure el acceso programático mediante Configuración de la AWS CLI para usar AWS IAM Identity Center en la Guía del usuario de AWS Command Line Interface.
En IAM (No recomendado)	Use credenciales a largo plazo para acceder a AWS.	Siga las instrucciones en Creación del primer grupo de usuarios y usuario de administrador de IAM en la Guía del usuario de IAM.	Configure el acceso programático mediante Administración de las claves de acceso de los usuarios de IAM en la Guía del usuario de IAM.

Usar políticas administradas por

En el paso anterior, configuró un usuario de IAM con las credenciales para acceder a sus recursos de AWS. En la mayoría de los casos, para utilizar el Programador de EventBridge de forma segura, se recomienda crear usuarios, grupos o roles independientes con solo los permisos necesarios para utilizar el Programador de EventBridge. El Programador de EventBridge admite las políticas administradas que se describen a continuación para casos de uso habituales.

- the section called “`AmazonEventBridgeSchedulerFullAccess`”— Otorga acceso completo al Programador de EventBridge mediante la consola y la API.
- the section called “`AmazonEventBridgeSchedulerReadOnlyAccess`”— Concede acceso de solo lectura al Programador de EventBridge.

Puede adjuntar estas políticas gestionadas a sus entidades principales de IAM del mismo modo que adjuntó la política `AdministratorAccess` en el paso anterior. Para obtener más información acerca de cómo administrar el acceso al Programador de EventBridge mediante políticas de IAM basadas en identidades, consulte the section called “Uso de políticas basadas en identidades”.

Configurar el rol de ejecución

Un rol de ejecución es un rol de IAM que asume el Programador de EventBridge para interactuar con Servicios de AWS en su nombre. Adjunte políticas de permisos a este rol para conceder acceso al Programador de EventBridge para invocar destinos.

También puede crear una nueva función de ejecución cuando utilice la consola para crear una nueva programación. Si utiliza la consola, el Programador de EventBridge crea un rol en su nombre con permisos en función del destino que elija. Cuando el Programador de EventBridge crea un rol para usted, la política de confianza del rol incluye claves de condición que limitan las entidades principales que pueden asumir el rol en su nombre. Esto protege contra el posible problema de seguridad del suplente adjunto.

Los pasos que se describen a continuación describen cómo crear un nuevo rol de ejecución y cómo conceder acceso al Programador de EventBridge para invocar un destino. En este tema se describen los permisos para los destinos con plantillas más populares. Para obtener información sobre cómo añadir permisos para otros destinos, consulte the section called “Uso de destinos con plantillas”.

Para crear un rol de ejecución con la AWS CLI

1. Copie la siguiente política JSON de asumir rol y guárdela localmente como `Scheduler-Execution-Role.json`. Esta política de confianza permite que el Programador de EventBridge asuma el rol en nombre de usted.

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

> **Important**
>
> Para configurar un rol de ejecución en un entorno de producción, recomendamos implementar medidas de seguridad adicionales para evitar errores del suplente confuso. Para obtener más información y una política de ejemplo, consulte the section called “Prevención del suplente confuso”.

2. En AWS Command Line Interface (AWS CLI), escriba el siguiente comando para crear un nuevo rol. Sustituya *`SchedulerExecutionRole`* por el nombre que desee asignar a este rol.

```
$ aws iam create-role --role-name SchedulerExecutionRole --assume-role-policy-
document file://Scheduler-Execution-Role.json
```

Si todo va bien, obtendrá el siguiente resultado:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "Scheduler-Execution-Role",
        "RoleId": "BR1L2DZK3K4CTL5ZF9EIL",
        "Arn": "arn:aws:iam::123456789012:role/SchedulerExecutionRole",
```

```
        "CreateDate": "2022-03-10T18:45:01+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "scheduler.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

3. Para crear una nueva política que permita al Programador de EventBridge invocar un destino, elija uno de los siguientes destinos comunes. Copie la política de permisos de JSON y guárdela localmente como un archivo `.json`.

   Amazon SQS – SendMessage

   Lo siguiente permite al Programador de EventBridge activar la acción `sqs:SendMessage` en todas las colas de Amazon SQS de su cuenta.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Action": [
                   "sqs:SendMessage"
               ],
               "Effect": "Allow",
               "Resource": "*"
           }
       ]
   }
   ```

   Amazon SNS – Publish

   Lo siguiente permite a el Programador de EventBridge iniciar la acción `sns:Publish` en todos los temas de Amazon SNS de su cuenta.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sns:Publish"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Lambda – Invoke

Lo siguiente permite a el Programador de EventBridge ejecutar la acción `lambda:InvokeFunction` en todas las funciones de Lambda de su cuenta.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

4. Ejecute el siguiente comando para crear la nueva política de permisos. Sustituya *PolicyName* por el nombre que desee asignar a esta política.

```
$ aws iam create-policy --policy-name PolicyName --policy-document file://
PermissionPolicy.json
```

Si todo va bien, obtendrá el siguiente resultado. Anote el ARN de la política. Utilice este ARN en el siguiente paso para asociar la política a nuestro rol de ejecución.

```
{
    "Policy": {
        "PolicyName": "PolicyName",
        "CreateDate": "2022-03-015T19:31:18.620Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:policy/PolicyName",
        "UpdateDate": "2022-03-015T19:31:18.620Z"
    }
}
```

5. Ejecute el siguiente comando para adjuntar la política a su rol de ejecución. Sustituya *your-policy-arn* por el ARN de la política que creó en el paso anterior. Sustituya *SchedulerExecutionRole* por el nombre de su rol de ejecución.

```
$ aws iam attach-role-policy --policy-arn your-policy-arn --role-
name SchedulerExecutionRole
```

   La operación `attach-role-policy` no devuelve una respuesta en la línea de comandos.

Configurar un destino

Antes de crear una programación del Programador de EventBridge, necesita al menos un destino que la programación pueda invocar. Puede usar un recurso de AWS existente o crear uno nuevo. Los pasos que se describen a continuación muestran cómo crear una cola Amazon SQS estándar con AWS CloudFormation.

Para crear una nueva cola de Amazon SQS

1. Copie la siguiente plantilla AWS CloudFormation JSON y guárdela localmente como `SchedulerTargetSQS.json`.

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "MyQueue": {
```

```
         "Type": "AWS::SQS::Queue",
         "Properties": {
            "QueueName": "MyQueue"
             }
         }
      },
   "Outputs": {
      "QueueName": {
         "Description": "The name of the queue",
         "Value": {
            "Fn::GetAtt": [
               "MyQueue",
               "QueueName"
            ]
         }
      },
      "QueueURL": {
         "Description": "The URL of the queue",
         "Value": {
            "Ref": "MyQueue"
         }
      },
      "QueueARN": {
         "Description": "The ARN of the queue",
         "Value": {
            "Fn::GetAtt": [
               "MyQueue",
               "Arn"
            ]
         }
      }
   }
}
```

2. Desde AWS CLI, ejecute el siguiente comando para crear una pila AWS CloudFormation a partir de la plantilla `Scheduler-Target-SQS.json`.

```
$ aws cloudformation create-stack --stack-name Scheduler-Target-SQS --template-body
 file://Scheduler-Target-SQS.json
```

Si todo va bien, obtendrá el siguiente resultado:

```
{
```

```
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/Scheduler-
Target-SQS/1d2af345-a121-12eb-abc1-012e34567890"
}
```

3. Ejecute el siguiente comando para ver información resumida de la pila AWS CloudFormation. Esta información incluye el estado de la pila y los resultados especificados en la plantilla.

```
$ aws cloudformation describe-stacks --stack-name Scheduler-Target-SQS
```

Si la operación se realiza correctamente, el comando crea la cola de Amazon SQS y devuelve el siguiente resultado:

```
{
    "Stacks": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
Scheduler-Target-SQS/1d2af345-a121-12eb-abc1-012e34567890",
            "StackName": "Scheduler-Target-SQS",
            "CreationTime": "2022-03-17T16:21:29.442000+00:00",
            "RollbackConfiguration": {},
            "StackStatus": "CREATE_COMPLETE",
            "DisableRollback": false,
            "NotificationARNs": [],
            "Outputs": [
                {
                    "OutputKey": "QueueName",
                    "OutputValue": "MyQueue",
                    "Description": "The name of the queue"
                },
                {
                    "OutputKey": "QueueARN",
                    "OutputValue": "arn:aws:sqs:us-west-2:123456789012:MyQueue",
                    "Description": "The ARN of the queue"
                },
                {
                    "OutputKey": "QueueURL",
                    "OutputValue": "https://sqs.us-
west-2.amazonaws.com/123456789012/MyQueue",
                    "Description": "The URL of the queue"
                }
            ],
            "Tags": [],
```

```
            "EnableTerminationProtection": false,
            "DriftInformation": {
                "StackDriftStatus": "NOT_CHECKED"
            }
        }
    ]
}
```

Más adelante en esta guía, utilizará el valor de `QueueARN` para configurar la cola como destino del Programador de EventBridge.

Siguientes pasos

Cuando haya completado el paso de configuración, utilice la guía de Introducción para crear su primer programador del Programador de EventBridge e invocar un destino.

Primeros pasos con el Programador de EventBridge

En este tema se describe la creación de una nueva programación del Programador de EventBridge. Utilice la consola, AWS Command Line Interface (AWS CLI) o los SDK AWS del Programador de EventBridge para crear una programación con un destino de Amazon SQS con una plantilla. A continuación, configurará el registro, configurará los reintentos y establecerá un tiempo máximo de retención para las tareas fallidas. Tras crear la programación, verificará que la programación invoque correctamente el destino y envíe un mensaje a la cola de destino.

> **Note**
>
> Para seguir esta guía, le recomendamos que configure los usuarios de IAM con los permisos mínimos requeridos que se describen en the section called “Uso de políticas basadas en identidades”. Tras crear y configurar un usuario, ejecute el siguiente comando para configurar sus credenciales de acceso. Necesitará su ID de clave de acceso y su clave de acceso secreta para configurar la AWS CLI.
>
> ```
> $ aws configure
> AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
> AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
> Default region name [None]: us-west-2
> Default output format [None]: json
> ```
>
> Para obtener más información sobre las distintas formas de establecer las credenciales, consulte los Ajustes de configuración y prioridad en la Guía del usuario de la AWS Command Line Interface de la versión 2.

Temas

Requisitos previos

Antes de seguir los pasos de esta sección, debe hacer lo siguiente:

- Realizar las tareas que se describen en Configuración

Crear una programación mediante la consola Programador de EventBridge

Para crear un programa nuevo con la consola

1. Inicie sesión en AWS Management Console y, a continuación, seleccione el siguiente enlace para abrir la sección del Programador de EventBridge de la consola de EventBridge: https://us-west-2.console.aws.amazon.com/scheduler/home?region=us-west-2#home

   > **Note**
   >
   > Puede cambiar de Región de AWS mediante el selector de regiones de la AWS Management Console.

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. En Descripción: opcional, introduzca una descripción para su programación. Por ejemplo, **`My first schedule`**.

   c. En Grupo de programaciones, elija un grupo de programaciones de las opciones de la lista desplegable. Si no ha creado ningún grupo de programaciones anteriormente, puede elegir el grupo `default` para su programación. Para crear un nuevo grupo de programaciones, seleccione el enlace para crear su propia programación en la descripción de la consola. Los grupos de programaciones se utilizan para agregar etiquetas a grupos de programaciones.

4. En la sección Patrón de programación, realice lo siguiente:

a. En Incidencia, elija una de las siguientes opciones de patrón. Las opciones de configuración cambian en función del patrón que seleccione.

   - Programación única: una programación única invoca solo una vez un destino en la fecha y hora que especifique.

     En Fecha y hora, introduzca una fecha válida en formato `YYYY/MM/DD`. A continuación, especifique una marca de tiempo en el formato `hh:mm` de 24 horas. Por último, elija una zona horaria de las opciones desplegables.

   - Programación recurrente: una programación recurrente invoca un destino a una velocidad que especifique mediante una expresión de frecuencia o de cron.

     Elija una programación basada en cron para configurar una programación mediante una expresióncron. O utilice una expresión de frecuencia, elija una programación basada en frecuencia e introduzca un número positivo para el Valor y, a continuación, elija una Unidad de las opciones desplegables.

   Para obtener más información sobre el uso de expresiones de frecuencia y cron, consulte Tipos de programaciones.

b. En Intervalo de tiempo flexible, elija Apagado para desactivar la opción o elegir uno de los periodos de tiempo predefinidos de la lista desplegable. 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.

> **Note**
>
> La característica de intervalo de tiempo flexible no está disponible con las programaciones de una sola vez.

Si eligió Programación recurrente en el paso anterior, en la sección Periodo de tiempo, especifique una zona horaria y, si lo desea, establezca una fecha y hora de inicio y una fecha y hora de finalización para la programación. Una programación periódica sin fecha de inicio comenzará en cuanto se cree y esté disponible. Una programación periódica sin fecha de finalización seguirá invocando su destino indefinidamente.

6. Elija Siguiente.

7. En la página Seleccionar destino, haga lo siguiente:

   a. Seleccione Destinos con plantilla y elija una API de destino. Para este ejemplo, elegiremos el destino con plantilla de Amazon SQS **`SendMessage`**.

   b. En la sección SendMessage, para la cola de SQS, elija un ARN de cola de Amazon SQS existente, por ejemplo `arn:aws:sqs:`*`us-west-2`*`:`*`123456789012`*`:`*`TestQueue`*, de la lista desplegable. Para crear una nueva cola, elija Crear nueva cola SQS para ir a la consola Amazon SQS. Cuando termine de crear una cola, vuelva a la consola del Programador de EventBridge y actualice el menú desplegable. Aparece el nuevo ARN de la cola y se puede seleccionar.

   c. En Destino, introduzca la carga útil que quiere que el Programador de EventBridge entregue al destino. En este ejemplo, enviaremos el siguiente mensaje a la cola de destino: **`Hello, it's EventBridge Scheduler.`**

8. Seleccione Siguiente y, en la página Configuración (opcional), haga lo siguiente:

9.
   a. En la sección Estado de la programación, en Habilitar programación, active o desactive la característica con el interruptor. De forma predeterminada, el programador de EventBridge habilita su programación.

   b. En la sección Acción tras la finalización de la programación, configure la acción que realiza el Programador de EventBridge una vez finalizada la programación:

      - Seleccione ELIMINAR si desea que la programación se elimine automáticamente. Para programaciones únicas, esto ocurre después de que la programación invoca al destino una vez. En el caso de las programaciones recurrentes, esto ocurre después de la última invocación planificada de la programación. Para obtener más información acerca de la eliminación automática, consulte the section called “Eliminación tras la finalización de la programación”.
      - Seleccione NINGUNO o no elija ningún valor si no desea que el Programador de EventBridge realice ninguna acción una vez finalizada la programación.

   c. En la sección Política de reintentos y cola de mensajes fallidos (DLQ), en Política de reintentos, active Reintento para configurar una política de reintentos para su programación. Con las políticas de reintentos, si un programa no puede invocar su destino, 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.

   d. En Antigüedad máxima del evento, opcional, introduzca las horas y minutos máximos que el Programador de EventBridge debe mantener un evento sin procesar.

> **Note**
>
> El valor máximo es de 24 horas.

e. En Cantidad máxima de reintentos, ingrese el número máximo de veces que el Programador de EventBridge reintenta la programación si el destino devuelve un error.

> **Note**
>
> El valor máximo es 185 reintentos.

f. En Cola de mensajes fallidos (DLQ), elija una de las siguientes opciones:

- Ninguna: elija esta opción si no desea configurar una DLQ.
- Seleccionar una cola de Amazon SQS en mi cuenta de AWS como DLQ: elija esta opción, seleccione un ARN de cola de la lista desplegable y configure una DLQ Cuenta de AWS igual al que está creando la programación.
- Especificar una cola de Amazon SQS en otra cuenta AWS como DLQ: elija esta opción e introduzca el ARN de la cola configurada como DLQ, si la cola está en otra Cuenta de AWS. Debe introducir el ARN exacto de la cola para poder utilizar esta opción.

g. En la sección Cifrado, elija Personalizar la configuración de cifrado (avanzada) para utilizar una clave KMS administrada por el cliente para cifrar su entrada de destino. Si elige esta opción, introduzca un ARN de clave de KMS existente o elija Crear una clave KMS de AWS para navegar hasta la consola de AWS KMS. Para obtener más información sobre cómo el Programador de EventBridge cifra los datos en reposo, consulte the section called “Cifrado en reposo”.

h. Para Permisos, elija Usar el rol existente y, a continuación, seleccione el rol que creó durante el procedimiento de configuración en la lista desplegable. También puede elegir Ir a la consola de IAM para crear un nuevo rol.

   Si desea que el Programador de EventBridge cree una nueva función de ejecución, elija Crear nuevo rol para esta programación. A continuación, ingrese un nombre para el Nombre de rol. Si elige esta opción, el Programador de EventBridge añade al rol los permisos necesarios para el destino creado con la plantilla.

10. Elija 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. Seleccione Crear programación para terminar de crear la nueva programación. Puede ver una lista de sus programaciones nuevas y existentes en la página de Programaciones. En la columna de Estado, verifique que su programación nueva se encuentre Habilitada.
13. Para comprobar que su programación invoca el destino de Amazon SQS, abra la consola de Amazon SQS y haga lo siguiente:

    a. Elija la cola de destino de la lista de Colas.

    b. Seleccione Send and receive messages (Enviar y recibir mensajes).

    c. En la página Enviar y recibir mensajes, en Recibir mensajes, seleccione Sondeo de mensajes para recuperar los mensajes de prueba que su agenda envió a la cola de destino.

Cree una programación utilizando la AWS CLI

En el siguiente ejemplo, se muestra cómo utilizar el comando AWS CLI `create-schedule` para crear una programación del Programador de EventBridge con un destino de Amazon SQS con una plantilla. Reemplace los valores de los marcadores de posición de los siguientes parámetros por su información:

- --name: introduzca un nombre para el programa.
- RoleArn: introduzca el ARN del rol de ejecución que desee asociar al cronograma.
- Arn: introduzca el ARN del destino. En este caso, el destino es una cola de Amazon SQS.
- Entrada: introduzca un mensaje que el Programador de EventBridge envíe a la cola de destino.

```
$ aws scheduler create-schedule --name sqs-templated-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Cree una programación con los SDK del Programador de EventBridge

En el siguiente ejemplo, utilice los SDK del Programador de EventBridge para crear una programación del Programador de EventBridge con un destino de Amazon SQS con plantilla.

Example SDK de Python

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "Message for scheduleArn: '<aws.scheduler.schedule-arn>', scheduledTime:
 '<aws.scheduler.scheduled-time>'"
}

scheduler.create_schedule(
    Name="sqs-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)
```

Example SDK de Java

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();
```

```
        Target sqsTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("<QUEUE_ARN>")
                .input("Message for scheduleArn: '<aws.scheduler.schedule-arn>',
 scheduledTime: '<aws.scheduler.scheduled-time>'")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(sqsTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
                .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and an Amazon SQS
 templated target");
    }
}
```

Siguientes pasos

- Para obtener más información sobre la gestión de la programación mediante la consola, AWS CLI o el SDK del Programador de EventBridge, consulte Administrar una programación.
- Para obtener más información sobre cómo configurar los destinos con plantilla y obtener información sobre cómo utilizar el parámetro de destino universal, consulte Gestión de destinos.
- Para obtener más información sobre los tipos de datos y las operaciones de la API del Programador de EventBridge, consulte la Referencia de la API del Programador de EventBridge.

Tipos de programaciones en el Programador de EventBridge

En el siguiente tema se describen los distintos tipos de programación que admite el Programador de Amazon EventBridge, así como la forma en la que el Programador de EventBridge gestiona el horario de verano y la programación en diferentes zonas horarias. Al configurar su programación, puede elegir entre tres tipos de programación: basada en frecuencia, cronológica y única.

Tanto las programaciones basadas en frecuencias como las basadas en cron son programaciones recurrentes. Para configurar cada tipo de programación periódica, utilice una expresión de programación para el tipo de programación que desee configurar y especificando una zona horaria en la que el Programador de EventBridge evalúe la expresión.

Una programación única es una programación que invoca solo una vez un destino. Para configurar una programación única, especifique la hora, la fecha y la zona horaria en la que el Programador de EventBridge evalúa la programación.

> **Note**
>
> Todos los tipos de programación del Programador de EventBridge invocan sus destinos con una precisión de 60 segundos. Esto significa que si configura su programación para que se ejecute a la `1:00`, invocará la API de destino entre `1:00:00` y `1:00:59`.

Utilice las siguientes secciones para obtener información sobre cómo configurar las expresiones de programación para cada tipo de programación recurrente y cómo configurar una programación única en el Programador de EventBridge.

Temas

Programaciones basadas en frecuencias

Una programación basada en frecuencia comienza después de la fecha de inicio que usted especifique para su programa y se ejecuta a una frecuencia regular que usted defina hasta la fecha de finalización de la programación. Puede configurar los casos de uso de la programación recurrente más comunes mediante una programación basada en frecuencias. Por ejemplo, si quiere una programación que invoque su destino cada 15 minutos, una vez cada dos horas o una vez cada cinco días, puede usar una programación basada en frecuencias para lograrlo. Para configurar una programación basada en frecuencia, utilice una expresión de frecuencia.

Con las programaciones basadas en frecuencias, utilice la propiedad `StartDate` para establecer la primera aparición de la programación. Si no proporciona una `StartDate` para una programación basada en frecuencia, su programación empezará a invocar el destino inmediatamente.

Las expresiones de frecuencia tienen dos campos obligatorios separados por un espacio en blanco, como se muestra a continuación.

Sintaxis

```
rate(value unit)
```

valor

Un número positivo.

unidad

La unidad de tiempo en la que quiere que su programación invoque su destino.

Entradas válidas: `minutes` | `hours` | `days`

Ejemplos

El siguiente ejemplo muestra cómo utilizar las expresiones de frecuencia con el comando AWS CLI `create-schedule` para configurar una programación basada en frecuencias. En este ejemplo, se crea una programación que se ejecuta cada cinco minutos y envía un mensaje a una cola de Amazon SQS, utilizando el tipo de destino con plantilla `SqsParameters`.

Como este ejemplo no establece un valor para el parámetro `--start-date`, la programación comienza a invocar su destino inmediatamente después de crearlo y activarlo.

```
$ aws scheduler create-schedule --schedule-expression 'rate(5 minutes)' --
name schedule-name \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Programaciones basadas en cron

Una expresión cron crea una programación recurrente detallada que se ejecuta en el momento específico que elija. El Programador de EventBridge permite configurar programaciones basadas en cron en la hora universal coordinada (UTC) o en la zona horaria que especifique al crear la programación. Con las programaciones basadas en cron, tiene más control sobre cuándo y con qué frecuencia se ejecuta su programación. Utilice programaciones basadas en cron cuando necesite una programación de recurrencia personalizada que no sea compatible con ninguna de las expresiones de frecuencia del Programador de EventBridge. Por ejemplo, puede crear una programación basada en cron que se ejecute a las 8:00 h. PST el primer lunes de cada mes. Una programación basada en cron se configura mediante una expresión cron.

Una expresión cron consta de cinco campos obligatorios separados por espacios en blanco: minutos, horas, día del mes, mes, día de la semana y un campo opcional, el año, como se muestra a continuación.

Sintaxis

```
cron(minutes hours day-of-month month day-of-week year)
```

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.

Ejemplos

El siguiente ejemplo muestra cómo utilizar las expresiones cron con el comando AWS CLI `create-schedule` para configurar una programación basada en cron. En este ejemplo, se crea una programación que se ejecuta a las 10:15 a. m. UTC+0 del último viernes de cada mes durante los

años 2022 a 2023 y envía un mensaje a una cola de Amazon SQS, utilizando el tipo de destino `SqsParameters` con plantilla.

```
$ aws scheduler create-schedule --schedule-expression "cron(15 10 ? * 6L 2022-2023)" --
name schedule-name \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Programación única

Una programación única invocará un destino solo una vez en la fecha y hora que especifique utilizando una fecha válida y una marca de tiempo. El Programador de EventBridge admite la programación en la hora universal coordinada (UTC) o en la zona horaria que especifique al crear la programación.

> **Note**
>
> Una programación única todavía cuenta para la cuota de su cuenta después de que haya terminado de ejecutarse e invocar su destino. Le recomendamos que elimine sus programaciones únicas una vez que hayan terminado de ejecutarse.

Para configurar una programación única, utilice una expresión at. Una expresión at consta de la fecha y la hora en las que desea que el Programador de EventBridge invoque su programación, tal y como se muestra a continuación.

Sintaxis

```
at(yyyy-mm-ddThh:mm:ss)
```

Al configurar una programación única, el Programador de EventBridge ignora la programación `StartDate` y `EndDate` que usted especifique para ella.

Ejemplos

En el siguiente ejemplo, se muestra cómo utilizar las expresiones at con el comando AWS CLI `create-schedule` para configurar una programación única. En este ejemplo, se crea una programación que se ejecuta una vez a las 13:00, hora peninsular española, del 20 de noviembre

de 2022 y envía un mensaje a una cola de Amazon SQS utilizando el tipo de destino con plantilla `SqsParameters`.

```
$ aws scheduler create-schedule --schedule-expression "at(2022-11-20T13:00:00)" --
name schedule-name \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--schedule-expression-timezone "America/Los_Angeles"
--flexible-time-window '{ "Mode": "OFF"}'
```

Zonas horarias en el Programador de EventBridge

El Programador de EventBridge permite configurar programaciones únicas y basadas en cron en cualquier zona horaria que especifique. El Programador de EventBridge utiliza la base de datos de zonas horarias mantenida por Internet Assigned Numbers Authority (IANA).

Con la AWS CLI, puede establecer la zona horaria en la que quiere que el Programador de EventBridge evalúe su programación mediante el parámetro `--schedule-expression-timezone`. Por ejemplo, el siguiente comando crea una programación basada en crono que invoca un destino `SendMessage` de Amazon SQS con plantilla en America/New_York todos los días a las 8:30 a.m.

```
$ aws scheduler create-schedule --schedule-expression "cron(30 8 * * ? *)" --name
 schedule-in-est \
   --target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "This schedule runs
 in the America/New_York time zone." }' \
   --schedule-expression-timezone "America/New_York"
   --flexible-time-window '{ "Mode": "OFF"}'
```

Horario de verano en el Programador de EventBridge

El Programador de EventBridge ajusta automáticamente la programación al horario de verano. Cuando la hora se adelanta en primavera, si una expresión cron coincide con una fecha y hora inexistentes, se omite la invocación de la programación. Cuando el tiempo retrocede en otoño, su programación se ejecuta solo una vez y no repite su invocación. Las siguientes invocaciones se producen normalmente en la fecha y hora especificadas.

El Programador de EventBridge ajusta la programación en función de la zona horaria que especifique al crear la programación. Si configura una programación en America/New_York, su programación

se ajusta cuando cambia la hora en esa zona horaria, mientras que una programación en America/Los_Angeles se ajusta tres horas después, cuando cambia la hora en la costa oeste.

En el caso de las programaciones basadas en frecuencias que se utilizan `days` como unidad, por ejemplo `rate(1 days)`, `days` representa una duración de 24 horas en el reloj. Esto significa que cuando el horario de verano hace que un día se reduzca a 23 horas o se amplíe a 25 horas, el Programador de EventBridge seguirá evaluando la expresión de frecuencia 24 horas después de la última invocación de la programación.

> **Note**
>
> Según las normas y reglamentos locales, algunas zonas horarias no respetan el horario de verano. Si crea una programación en una zona horaria que no respeta el horario de verano, el Programador de EventBridge no ajustará su programación. Los ajustes del horario de verano no se aplican a las programaciones en horario universal coordinado (UTC).

Ejemplo

Considere un escenario en el que se crea una programación utilizando la siguiente expresión cron en America/Los_Angeles: `cron(30 2 * * ? *)`. Esta programación se ejecuta todos los días a las 2:30 a.m. en la zona horaria especificada.

- Avance: cuando el tiempo se adelanta en primavera entre las 1:59 y las 3:00, el Programador de EventBridge omite la invocación de la programación de ese día y vuelve a ejecutar la programación con normalidad al día siguiente.
- Alternativa: cuando el tiempo retrocede en otoño, de las 2:59 a las 2:00 a. m., el Programador de EventBridge ejecuta la programación solo una vez a las 2:30 a.m. antes de que se produzca el cambio, pero no repite la invocación de la programación nuevamente a las 2:30 a.m. después del cambio de hora.

Administrar una programación

Una programación es el recurso principal que se crea, configura y administra con el Programador de Amazon EventBridge.

Cada programación tiene una expresión de programación que determina cuándo y con qué frecuencia se ejecuta la programación. El Programador de EventBridge admite tres tipos de programaciones: frecuencia, cron y de una vez. Para obtener más información acerca de los diferentes tipos de programación, consulte Tipos de programaciones.

Al crear una programación, se configura un destino para que la programación la invoque. Un destino es una operación de API a la que el Programador de EventBridge llama en su nombre cada vez que se ejecuta su programación. El Programador de EventBridge admite dos tipos de destinos: los destinos con plantilla llaman a operaciones de API comunes en un grupo principal de servicios y el parámetro de destino universal (UTP), que puede usar para llamar a más de 6000 operaciones en más de 270 servicios. Para obtener más información acerca de la configuración de destinos, consulte Gestión de destinos.

Puede configurar el modo en que su programación gestiona los errores cuando el Programador de EventBridge no puede entregar un evento correctamente a un destino mediante dos mecanismos principales: una política de reintentos y una cola de mensajes fallidos (DLQ). Una política de reintentos determina el número de veces que el Programador de EventBridge debe volver a intentar un evento fallido y cuánto tiempo debe mantener un evento sin procesar. Una DLQ es una cola estándar de Amazon SQS que el Programador de EventBridge utiliza para entregar los eventos fallidos una vez agotada la política de reintentos. Puede utilizar una DLQ para solucionar problemas relacionados con su programación o con su destino final. Para obtener más información acerca de ello, consulte the section called “Configuración de una cola de mensajes fallidos”.

En esta sección, encontrará ejemplos de cómo gestionar las programaciones del Programador de EventBridge mediante la consola, la AWS CLI y los SDK del Programador de EventBridge.

Temas

- Siguientes pasos

Cambiar el estado de la programación

Una programación del Programador de EventBridge tiene dos estados: activado y desactivado. El siguiente ejemplo se utiliza `UpdateSchedule` para deshabilitar una programación que se activa cada cinco minutos e invoca un destino Lambda.

Cuando utilice `UpdateSchedule`, deberá proporcionar todos los parámetros necesarios. El Programador de EventBridge reemplaza su programación por la información que proporcione. Si no especifica este parámetro establecido anteriormente, se utiliza de forma predeterminada `null`.

Example AWS CLI

```
$ aws scheduler update-schedule --name lambda-universal --schedule-expression 'rate(5
 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"arn:aws:scheduler:::aws-sdk:lambda:invoke"
 "Input": "{\"FunctionName\":\"arn:aws:lambda:REGION:123456789012:function:HelloWorld
\",\"InvocationType\":\"Event\",\"Payload\":\"{\\\"message\\\":\\\"testing function\\
\"}\"}" }' \
--flexible-time-window '{ "Mode": "OFF"}' \
--state DISABLED
```

```
{
    "ScheduleArn": "arn:aws:scheduler:us-west-2:123456789012:schedule/default/lambda-
universal"
}
```

En el siguiente ejemplo, se utiliza el SDK de Python y la operación `UpdateSchedule` para deshabilitar una programación orientada a Amazon SQS mediante un destino con plantilla.

Example SDK de Python

```
import boto3
scheduler = boto3.client('scheduler')

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "{}"}
```

```
flex_window = { "Mode": "OFF" }

scheduler.update_schedule(Name="your-schedule",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window,
    State='DISABLED')
```

Configuración de ventanas de tiempo flexibles

Cuando configura su programación con un intervalo de tiempo flexible, el Programador de EventBridge invoca el destino dentro del intervalo de tiempo que haya establecido. Esto resulta útil en los casos en que no se requiere una invocación precisa y programada de los destinos. Establecer un intervalo de tiempo flexible mejora la fiabilidad de la programación al dispersar las invocaciones de los destinos.

Por ejemplo, si configura un intervalo de tiempo flexible de 15 minutos para una programación que se ejecuta cada hora, invoca al destino 15 minutos después de la hora programada. Las siguientes AWS CLI, y los ejemplos del SDK del Programador de EventBridge, utilizan `UpdateSchedule` para establecer un intervalo de tiempo flexible de 15 minutos para una programación que se ejecuta una vez cada hora.

> **Note**
>
> Debe especificar si desea establecer un intervalo de tiempo flexible o no. Si no desea establecer esta opción, especifique `OFF`. Si establece el valor en `FLEXIBLE`, debe especificar un intervalo de tiempo máximo durante el que se programará la ejecución.

Example AWS CLI

```
$ aws scheduler update-schedule --name lambda-universal --schedule-expression 'rate(1
 hour)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"arn:aws:scheduler:::aws-sdk:lambda:invoke"
 "Input": "{\"FunctionName\":\"arn:aws:lambda:REGION:123456789012:function:HelloWorld
\",\"InvocationType\":\"Event\",\"Payload\":\"{\\\"message\\\":\\\"testing function\\
\"}\"}" }' \
--flexible-time-window '{ "Mode": "FLEXIBLE", "MaximumWindowInMinutes": 15} \
```

```
{
    "ScheduleArn": "arn:aws:scheduler:us-west-2:123456789012:schedule/lambda-universal"
}
```

Example SDK de Python

```
import boto3
scheduler = boto3.client('scheduler')

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "{}"}

flex_window = { "Mode": "FLEXIBLE", "MaximumWindowInMinutes": 15}

scheduler.update_schedule(Name="your-schedule",
    ScheduleExpression="rate(1 hour)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)
```

Configuración de una cola de mensajes fallidos para una programación

El Programador de Amazon EventBridge admite colas de mensajes fallidos (DLQ) mediante Amazon Simple Queue Service. Cuando un programa no puede invocar su destino, el Programador de EventBridge envía una carga JSON que contiene los detalles de la invocación y cualquier respuesta recibida del destino a una cola estándar de Amazon SQS que especifique.

En el siguiente tema, se hace referencia a este JSON como un evento de mensajes fallidos. Un evento de mensajes fallidos le permite solucionar problemas relacionados con su programación o sus destinos. Si configura una política de reintentos para su programación, el Programador de EventBridge proporciona el evento de mensajes fallidos, agotando el número máximo de reintentos que haya establecido.

En los siguientes temas, se describe cómo puede configurar una cola de Amazon SQS como DLQ según su programación, configurar los permisos que el Programador de EventBridge necesita para entregar mensajes a Amazon SQS y recibir eventos de mensajes fallidos de la DLQ.

Temas

Cree una cola de Amazon SQS.

Antes de configurar una DLQ para su programación, debe crear una cola Amazon SQS estándar. Para obtener instrucciones sobre cómo crear una cola mediante la consola de Amazon SQS, consulte Creación de una cola de Amazon SQS en la Guía para desarrolladores de Amazon Simple Queue Service.

> **Note**
>
> El Programador de EventBridge no admite el uso de una cola FIFO como DLQ de su programación.

Utilice el siguiente comando AWS CLI para crear una cola estándar.

```
$ aws sqs create-queue --queue-name queue-name
```

Si se realiza correctamente, verá `QueueURL` en el resultado.

```
{
    "QueueUrl": "https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-dlq-test"
}
```

Una vez creada la cola, anote el ARN de la cola. Necesitará el ARN cuando especifique una DLQ para su programación del Programador de EventBridge. Puede encontrar el ARN de la cola en la consola de Amazon SQS o mediante el comando `get-queue-attributes`AWS CLI.

```
$ aws sqs get-queue-attributes --queue-url your-dlq-url --attribute-names QueueArn
```

Si se realiza correctamente, verá el ARN de la cola en la salida.

```
{
    "Attributes": {
        "QueueArn": "arn:aws:sqs:us-west-2:123456789012:scheduler-dlq-test"
    }
}
```

En la siguiente sección, añadirá los permisos necesarios a su función de ejecución de la programación para permitir que Programador de EventBridge entregue eventos de mensajes fallidos a Amazon SQS.

Configurar permisos de rol de ejecución

Para permitir que el Programador de EventBridge entregue eventos de mensajes fallidos a Amazon SQS, su función de ejecución de la programación necesita la siguiente política de permisos. Para obtener más información sobre cómo adjuntar una nueva política de permisos a su rol de ejecución programado, consulte Configuración del rol de ejecución.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:SendMessage"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Es posible que su función de ejecución programada ya tenga los permisos necesarios adjuntos si utiliza el Programador de EventBridge para invocar un destino de la API de Amazon SQS.

En la siguiente sección, utilizará la consola del Programador de EventBridge y especificará una DLQ para la programación.

Especificar una política de cola de mensajes fallidos

Para especificar una DLQ, utilice la consola el Programador de EventBridge o la AWS CLI para actualizar un programa existente o crear uno nuevo.

Console

Para especificar una DQL utilizando la consola

1. Inicie sesión en AWS Management Console y, a continuación, seleccione el siguiente enlace para abrir la sección del Programador de EventBridge de la consola EventBridge: https://console.aws.amazon.com/scheduler/home
2. En la consola del Programador de EventBridge, cree una nueva programación o elija una programación existente de la lista de programaciones para editarla.
3. En la página de Configuración, para cola de mensajes fallidos (DLQ), realice una de las siguientes acciones:
   - Seleccione Seleccionar una cola de Amazon SQS en mi cuenta AWS como una DLQ y, a continuación, elija el ARN de cola para su DLQ en la lista desplegable.
   - Elija Especificar una cola de Amazon SQS en otras cuentas AWS como DLQ y, a continuación, introduzca el ARN de cola de su DLQ. Si elige una cola en otra cuenta AWS, la consola del Programador de EventBridge no podrá mostrar los ARN de la cola en una lista desplegable.
4. Revise sus selecciones y, a continuación, seleccione Crear programación o Guardar programación para terminar de configurar una DLQ.
5. (Opcional) Para ver los detalles de la DLQ de una programación, elija el nombre de la lista y, a continuación, seleccione la pestaña cola de mensajes fallidos en la página de detalles de la programación.

AWS CLI

Para actualizar una programación existente usando la AWS CLI

- Use el comando `update-schedule` para actualizar su agenda. Especifique la cola de Amazon SQS que creó anteriormente como DLQ. Especifique el ARN del rol de IAM al que ha adjuntado los permisos de Amazon SQS necesarios como rol de ejecución. Reemplace todos los demás valores de marcador de posición con su información.

```
$ aws scheduler update-schedule --name existing-schedule \
    --schedule-expression 'rate(5 minutes)' \
    --target '{"DeadLetterConfig": {"Arn": "DLQ_ARN"}, "RoleArn": "ROLE_ARN",
 "Arn":"QUEUE_ARN", "Input": "Hello world!" }' \
    --flexible-time-window '{ "Mode": "OFF"}'
```

Para crear una nuev programación con un DLQ utilizando la AWS CLI

- Para crear una programación, use el comando `create-schedule`. Sustituya el texto de todos los valores marcadores de posición por sus propios valores.

```
$ aws scheduler create-schedule --name new-schedule \
    --schedule-expression 'rate(5 minutes)' \
    --target '{"DeadLetterConfig": {"Arn": "DLQ_ARN"}, "RoleArn": "ROLE_ARN",
 "Arn":"QUEUE_ARN", "Input": "Hello world!" }' \
    --flexible-time-window '{ "Mode": "OFF"}'
```

En la siguiente sección, utilizará la AWS CLI para recibir un evento de mensajes fallidos de DLQ.

Recuperar el evento de mensajes fallidos

Utilice el comando `receive-message`, tal y como se muestra a continuación, para recuperar un evento de mensajes fallidos de DLQ. Puede establecer el número de mensajes que se van a recuperar mediante el atributo `--max-number-of-messages`.

```
$ aws sqs receive-message --queue-url your-dlq-url --attribute-names All --message-
attribute-names All --max-number-of-messages 1
```

Si la operación se realiza correctamente, verá un resultado similar al siguiente.

```
{
    "Messages": [
        {
            "MessageId": "2aeg3510-fe3a-4f5a-ab6a-6906560eaf7e",
            "ReceiptHandle": "AQEBkNKTdOMrWgHKPoITRBwrPoK3eCSZIcZwVqCY0BZ
+FfTcORFpopJbtCqj36VbBTlHreM8+qM/m5jcwqSlAlGmIJO/hYmMgn/
+dwIty9izE7HnpvRhhEyHxbeTZ5V05RbeasYaBdNyi9WLcnAHviDh6MebLXXNWoFyYNsxdwJuG0f/
w3htX6r3dxpXvvFNPGoQb8ihY37+u0gtsbuIwhLtUSmE8rbldEEwiUfi3IJ1zEZpUS77n/k1GWrMrnYg0Gx/
```

```
BuaLzOrFi2F738XI/
Hnh45uv3ca6OYwS1ojPQ1LtX2URg1haV5884FYlaRvY8jRlpCZabTkYRTZKSXG5KNgYZnHpmsspii6JNkjitYVFKPo0H91w
            "MD5OfBody": "07adc3fc889d6107d8bb8fda42fe0573",
            "Body": "{\"MessageBody\":\"Hello, world!\",\"QueueUrl\":\"https://sqs.us-
west-2.amazonaws.com/123456789012/does-not-exist\"}",
            "Attributes": {
                "SenderId": "AROA2DZE3W4CTL5ZR7EIN:ff00212d8c453aaaae644bc6846d4723",
                "ApproximateFirstReceiveTimestamp": "1652499058144",
                "ApproximateReceiveCount": "2",
                "SentTimestamp": "1652490733042"
            },
            "MD5OfMessageAttributes": "f72c1d78100860e00403d849831d4895",
            "MessageAttributes": {
                "ERROR_CODE": {
                    "StringValue": "AWS.SimpleQueueService.NonExistentQueue",
                    "DataType": "String"
                },
                "ERROR_MESSAGE": {
                    "StringValue": "The specified queue does not exist for this wsdl
 version.",
                    "DataType": "String"
                },
                "EXECUTION_ID": {
                    "StringValue": "ad06616e51cdf74a",
                    "DataType": "String"
                },
                "EXHAUSTED_RETRY_CONDITION": {
                    "StringValue": "MaximumEventAgeInSeconds",
                    "DataType": "String"
                }
                "IS_PAYLOAD_TRUNCATED": {
                    "StringValue": "false",
                    "DataType": "String"
                },
                "RETRY_ATTEMPTS": {
                    "StringValue": "0",
                    "DataType": "String"
                },
                "SCHEDULED_TIME": {
                    "StringValue": "2022-05-14T01:12:00Z",
                    "DataType": "String"
                },
                "SCHEDULE_ARN": {
```

```
                    "StringValue": "arn:aws:scheduler:us-west-2:123456789012:schedule/
DLQ-test",
                    "DataType": "String"
                },
                "TARGET_ARN": {
                    "StringValue": "arn:aws:scheduler:::aws-sdk:sqs:sendMessage",
                    "DataType": "String"
                }
            }
        }
    ]
}
```

Tenga en cuenta los siguientes atributos en caso de en evento de mensajes fallidos para ayudarle a identificar y solucionar los posibles motivos por los que la innovación del destino ha fallado.

- **ERROR_CODE**— Contiene el código de error que el Programador de EventBridge recibe de la API de servicio del destino. En el ejemplo anterior, el código de error devuelto por Amazon SQS es `AWS.SimpleQueueService.NonExistentQueue`. Si la programación no puede invocar un destino debido a un problema con el Programador de EventBridge, en su lugar verá el siguiente código de error: `AWS.Scheduler.InternalServerError`.
- **ERROR_MESSAGE**— Contiene el mensaje de error que el Programador de EventBridge recibe de la API de servicio del destino. En el ejemplo anterior, el mensaje de error devuelto por Amazon SQS es `The specified queue does not exist for this wsdl version`. Si la programación falla debido a un problema con el Programador de EventBridge, en su lugar verá el siguiente mensaje de error: `Unexpected error occurred while processing the request`.
- **TARGET_ARN**— El ARN del destino que invoca su programación, en el siguiente formato ARN de servicio: `arn:aws:scheduler:::aws-sdk:`*`service`*`:`*`apiAction`*.
- **EXHAUSTED_RETRY_CONDITION**— Indica por qué se envió el evento a la DLQ. Este atributo estará presente si el error de la API de destino es un error que se puede volver a intentar y no un error permanente. El atributo puede contener los valores `MaximumRetryAttempts` si el Programador de EventBridge lo envió a la DLQ después de superar el máximo de reintentos que configuró para la programación o `MaximumEventAgeInSeconds` si el evento tiene una antigüedad superior a la antigüedad máxima que configuró en la programación y sigue sin entregarse.

En el ejemplo anterior, podemos determinar, basándonos en el código de error y el mensaje de error, que la cola de destino que especificamos para la programación no existe.

Eliminar una programación

Puede eliminar una programación configurando la eliminación automática o eliminando manualmente una programación individual. Utilice los siguientes temas para obtener información sobre cómo eliminar una programación mediante ambos métodos y por qué podría elegir un método en lugar del otro.

Temas

Eliminación tras la finalización de la programación

Configure la eliminación automática una vez finalizada la programación si quiere evitar tener que gestionar de forma individual los recursos de la programación en el Programador de EventBridge. En las aplicaciones en las que crea miles de programaciones a la vez y necesita flexibilidad para escalar verticalmente el número de programaciones a pedido, la eliminación automática puede garantizar que no alcance la cuota de su cuenta para la cantidad de programaciones de una región específica.

Al configurar la eliminación automática de una programación, el Programador de EventBridge elimina la programación tras su última invocación de destino. En el caso de las programaciones únicas, esto ocurre después de que la programación haya invocado su destino una vez. En el caso de las programaciones recurrentes que configura con expresiones de frecuencia o cron, la programación se elimina después de su última invocación. La última invocación de una programación recurrente es la invocación que se produce más cerca de la `EndDate` que especifique. Si configura una programación con eliminación automática pero no especifica un valor para `EndDate`, el Programador de EventBridge no eliminará automáticamente la programación.

Puede configurar la eliminación automática al crear una programación por primera vez o actualizar las preferencias de una programación existente. Los siguientes pasos describen cómo configurar la eliminación automática de una programación existente.

AWS Management Console

1. Abra la consola del Programador de EventBridge en https://console.aws.amazon.com/scheduler/.
2. En la lista de programaciones, seleccione la programación que desee editar y, a continuación, elija Editar.

3. En la lista de navegación de la izquierda, elija Configuración..
4. En la sección Acción tras finalizar la programación, seleccione ELIMINAR en la lista desplegable y guarde los cambios.

AWS CLI

1. Abra una nueva ventana del símbolo del sistema.
2. Utilice el comando AWS CLI update-schedule para actualizar una programación existente, tal y como se muestra a continuación. El comando establece `--action-after-completion` en `DELETE`. En este ejemplo se supone que ha definido la configuración de destino de forma local en un archivo JSON. Para actualizar una programación, debe proporcionar el destino, así como cualquier otro parámetro de la programación que desee configurar para la programación existente.

   Se trata de una programación recurrente con una frecuencia de una invocación por hora. Por lo tanto, debe especificar una fecha de finalización al configurar el parámetro `--action-after-completion`.

   ```
   $ aws scheduler update-schedule --name schedule-name
    \
   --action-after-completion 'DELETE' \
   --schedule-expression 'rate(1 hour)' \
   --end-date '2024-01-01T00:00:00'
   --target file://target-configuration.json \
   --flexible-time-window '{ "Mode": "OFF"}' \
   ```

Eliminación manual

Cuando ya no necesite una programación, puede eliminarla mediante la operación `DeleteSchedule`.

Example AWS CLI

```
$ aws scheduler delete-schedule --name your-schedule
```

Example SDK de Python

```
import boto3
```

```
scheduler = boto3.client('scheduler')

scheduler.delete_schedule(Name="your-schedule")
```

Siguientes pasos

- Para obtener más información sobre cómo configurar destinos con plantillas para Lambda y Step Functions, y para obtener información sobre el uso del parámetro de destino universal, consulte Gestión de destinos.
- Para obtener más información sobre los tipos de datos y las operaciones de la API del Programador de EventBridge, consulte la Referencia de la API del Programador de EventBridge.

Administrar un grupo de programaciones

Un grupo de programaciones es un recurso del Programador de Amazon EventBridge que se utiliza para organizar las programaciones.

Su Cuenta de AWS viene con un grupo de programaciones `default`. Puede asociar una nueva programación al grupo `default` o a los grupos de programaciones que cree y administre. Puede crear hasta 500 grupos de programaciones en su Cuenta de AWS. Con el Programador de EventBridge, puede organizar grupos de programaciones, en lugar de programaciones individuales, mediante la aplicación de etiquetas.

Una etiqueta es una etiqueta compuesta por una clave que distingue entre mayúsculas y minúsculas y un valor que usted define. Puede crear etiquetas para categorizar programas según criterios como propósito, propietario o entorno. Por ejemplo, puede identificar el entorno al que pertenecen sus programaciones con la siguiente etiqueta: `environment:`*`production`*.

> **Important**
>
> No agregue información de identificación personal (PII) ni otra información confidencial en las etiquetas. Las etiquetas son accesibles para muchos servicios de AWS, incluida la facturación. Las etiquetas no se han diseñado para usarse con información privada o confidencial.

Un grupo de programaciones tiene dos estados posibles: ACTIVE y DELETING.

Cuando crea un grupo por primera vez, es `ACTIVE` de forma predeterminada. Puede añadir programaciones a un grupo `ACTIVE`. Al eliminar un grupo, el estado cambia a `DELETING` hasta que el Programador de EventBridge finalice la eliminación de las programaciones asociadas. Una vez que Programador de EventBridge elimine las programaciones del grupo, el grupo dejará de estar disponible en su cuenta.

Use los siguientes temas para crear un grupo de programaciones y aplicarle una etiqueta. También asociará una programación al grupo y, por último, eliminará el grupo.

Temas

- Recursos relacionados

Crear un grupo de programaciones

Utilice los grupos de programaciones y el etiquetado para organizar las programaciones que comparten un propósito común o que pertenecen al mismo entorno. En los siguientes pasos, creará un nuevo grupo de programaciones y lo etiquetará con una etiqueta. A continuación, asocie una nueva programación a ese grupo.

> **Note**
>
> Una vez que haya creado un grupo, no podrá eliminar una programación de ese grupo ni asociarla a un grupo diferente. Solo puede asociar una programación a un grupo cuando la crea por primera vez.

Paso uno: Crear un nuevo grupo de programaciones

En los siguientes temas se describe cómo crear un nuevo grupo de programaciones y etiquetarlo con la siguiente etiqueta: `environment:development`.

AWS Management Console

Para crear un nuevo grupo usando la AWS Management Console

1. Inicie sesión en AWS Management Console y abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Grupos de programaciones.
3. En la página de Grupos de programaciones, elija Crear grupo de programaciones.
4. En la sección Detalles del grupo de programaciones, en Nombre, introduzca un nombre para el grupo. Por ejemplo, **`TestGroup`**.
5. En la sección Etiquetas, haga lo siguiente:
   a. Elija Add new tag (Agregar nueva etiqueta).
   b. En Clave, introduzca el nombre que desee asignar a esta clave. En este tutorial, para etiquetar el entorno al que pertenece este grupo de programaciones, introduzca **`environment`**.

c. En Valor (opcional), introduzca el valor que desee asignar a esta clave. Para este tutorial, introduzca el valor **`development`** para la clave de entorno.

> **Note**
>
> Puede añadir etiquetas adicionales al grupo una vez que lo haya creado.

6. Seleccione Crear grupo de programaciones para finalizar. El nuevo grupo aparece en la lista Grupos de programaciones.
7. (Opcional) Para editar un grupo o administrar sus etiquetas, active la casilla del nuevo grupo y elija Editar.

> **Note**
>
> El grupo de programaciones `default` no se puede editar.

AWS CLI

Para crear un nuevo grupo usando la AWS CLI

1. Abra una nueva ventana del símbolo del sistema.
2. En AWS Command Line Interface (AWS CLI), escriba el siguiente comando `create-schedule-group` para crear un nuevo rol. Este comando crea un grupo con una etiqueta: `environment:development`. Puede usar esta etiqueta o un sistema de etiquetado similar para etiquetar sus grupos de programaciones según el entorno al que pertenecen.

   Sustituya el nombre del programa y la clave y el valor de la etiqueta por su información.

```
$ aws scheduler create-schedule-group --name TestGroup --tags
 Key=environment,Value=development
```

De forma predeterminada, el nuevo grupo se encuentra en el estado `ACTIVE`. Ahora puede asociar nuevas programaciones al nuevo grupo que creó.

Paso dos: asociar una programación al grupo

Siga los pasos siguientes para asociar una nueva programación al grupo que creó en el paso anterior.

AWS Management Console

Para asociar una programación a un grupo mediante la AWS Management Console

1. Inicie sesión en AWS Management Console y abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Programaciones en el panel de navegación izquierdo.
3. En la tabla Programaciones, seleccione Crear programación para crear una nueva programación.
4. En la página Especificar los detalles de la programación, en Grupo de programaciones, seleccione el nombre del nuevo grupo en la lista desplegable. Por ejemplo, seleccione `TestGroup`.
5. Especifique un patrón de programación, un destino y una configuración y, a continuación, revise su selección en la página Revisar y guardar la programación. Para obtener más información sobre la configuración de una programación nueva, consulte Introducción.
6. Para finalizar y guardar la programación, seleccione Guardar programación.

AWS CLI

Para asociar una programación a un grupo mediante la AWS CLI

1. Abra una nueva ventana del símbolo del sistema.
2. Desde AWS Command Line Interface (AWS CLI), introduzca el siguiente comando `create-schedule`. Esto crea una programación y la asocia al grupo del paso anterior, denominado `sqs-test-schedule`. Este programa utiliza el tipo de destino Amazon SQS con plantilla para invocar la operación `SendMessage`. Sustituya el nombre del programa, el destino y el nombre del grupo por su información.

```
$ aws scheduler create-schedule --name sqs-test-schedule --schedule-expression
 'rate(5 minutes)' \
```

```
--target '{"RoleArn": "ROLE_ARN", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }'
 \
--group-name TestGroup
--flexible-time-window '{ "Mode": "OFF"}'
```

Su nueva programación ahora está asociada al grupo de programaciones `TestGroup`.

Eliminar un grupo de programaciones

A continuación, puede obtener información sobre cómo eliminar un grupo de programaciones mediante la AWS Management Console y la AWS Command Line Interface. Al eliminar un grupo, permanece en ese estado `DELETING` hasta que el Programador de EventBridge elimine todas las programaciones del grupo. Una vez que Programador de EventBridge elimine las programaciones del grupo, el grupo dejará de estar disponible en su cuenta.

> **Note**
>
> Una vez que haya creado un grupo, no podrá eliminar una programación de ese grupo ni asociarla a un grupo diferente. Solo puede asociar una programación a un grupo cuando la crea por primera vez.

AWS Management Console

Para eliminar un grupo desde la AWS Management Console

1. Inicie sesión en AWS Management Console y abra la consola de Amazon EventBridge en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Grupos de programaciones en el panel de navegación izquierdo.
3. En la página Grupos de programaciones, en la lista de grupos existentes en la Región de AWS actual, localice el grupo que desee eliminar. Si no ve el grupo que busca, elija otra Región de AWS.

> **Note**
>
> No puede eliminar ni editar el grupo predeterminado.

4. Seleccione la casilla de verificación correspondiente al grupo que desea eliminar.
5. Elija Eliminar.
6. En el cuadro de diálogo Eliminar grupo de programaciones, introduzca el nombre del grupo para confirmar su elección y, a continuación, seleccione Eliminar.
7. En la lista Grupos de programaciones, la columna Estado cambia para indicar que su grupo se está eliminando en este momento. El grupo permanece en este estado hasta que el Programador de EventBridge elimine todas las programaciones asociadas al grupo.
8. Para actualizar la lista y confirmar que el grupo se ha eliminado, pulse el icono Actualizar.

AWS CLI

Para eliminar un grupo desde la AWS CLI

1. Abra una nueva ventana del símbolo del sistema.
2. Desde AWS Command Line Interface (AWS CLI), introduzca el siguiente comando `delete-schedule-group` para eliminar el grupo de programaciones. Sustituya el valor de `--name` por su información.

```
$ aws scheduler delete-schedule-group --name TestGroup
```

Si se realiza correctamente, esta operación AWS CLI no devuelve ninguna respuesta.

3. Para comprobar que el grupo está en ese estado `DELETING`, ejecute el siguiente comando `get-schedule-group`.

```
$ aws scheduler get-schedule-group --name TestGroup
```

Si se ejecuta correctamente, verá un resultado similar al siguiente:

```
{
    "Arn": "arn:aws::scheduler:us-west-2:123456789012:schedule-group/TestGroup",
    "CreationDate": "2023-01-01T09:00:00.000000-07:00",
    "LastModificationDate": "2023-01-01T09:00:00.000000-07:00",
    "Name": "TestGroup",
    "State": "DELETING"
}
```

El Programador de EventBridge elimina el grupo después de eliminar las programaciones asociadas al grupo. Si vuelve a ejecutar `get-schedule-group`, recibirá la siguiente respuesta `ResourceNotFoundException`:

```
An error occurred (ResourceNotFoundException) when calling the GetScheduleGroup
 operation: Schedule group TestGroup does not exist.
```

Recursos relacionados

Para obtener más información sobre los grupos de programaciones, consulte los siguientes recursos:

- Operación CreateScheduleGroup en la referencia de la API del Programador de EventBridge.
- Operación DeleteScheduleGroup en la referencia de la API del Programador de EventBridge.

Gestión de destinos

En los temas siguientes se describe cómo utilizar los destinos universales y con plantillas con el Programador de EventBridge y se proporciona una lista de los servicios AWS compatibles que puede configurar mediante el parámetro de destino universal de el Programador de EventBridge.

Los destinos con plantillas son un conjunto de operaciones de API comunes en un grupo de servicios de AWS principales, como Amazon SQS, Lambda y Step Functions. Por ejemplo, puede dirigirse a la operación de API Invocar de Lambda proporcionando la función ARN, o a la operación de Amazon SQS `SendMessage` con el ARN de cola del destino.

El destino universal es un conjunto de parámetros personalizables que le permiten invocar un conjunto más amplio de operaciones de API para muchos servicios de AWS. Por ejemplo, puede utilizar el parámetro de destino universal (UTP) de el Programador de EventBridge para crear una nueva cola de Amazon SQS mediante la operación `CreateQueue`.

Para configurar destinos universales o con plantilla, su programa debe tener permiso para llamar a la operación de API que configure como destino. Para ello, adjunte los permisos necesarios al rol de ejecución de la programación. Por ejemplo, para centrarse en la operación `SendMessage` de Amazon SQS, se debe conceder permiso al rol de ejecución para realizar la acción `sqs:SendMessage`. En la mayoría de los casos, puede añadir los permisos necesarios mediante las políticas administradas de AWS que admite el servicio de destino. Sin embargo, también puede crear sus propias políticas administradas por el cliente o agregar permisos integrados a una política existente asociada a la función de ejecución. En los siguientes temas se muestran ejemplos de cómo añadir permisos tanto para los tipos de destino con plantilla como para los universales.

Para obtener más información acerca de cómo configurar un rol de ejecución para un cronograma, consulte the section called “Configurar el rol de ejecución”.

Temas

Uso de destinos con plantillas

Los objetivos con plantillas son un conjunto de operaciones de API comunes en un grupo de AWS servicios principales, como Amazon SQS, Lambda y Step Functions. Por ejemplo, puede segmentar la operación `Invoke` de Lambda proporcionando la función ARN o la operación `SendMessage` de Amazon SQS mediante el ARN de cola. Para configurar un destino con plantilla, también debe conceder permisos a la función de ejecución del programa para que realice la operación de API de destino.

Para configurar un destino con plantilla mediante programación mediante el SDK del EventBridge programador AWS CLI o uno de ellos, debe especificar el ARN de la función de ejecución, el ARN del recurso de destino, una entrada opcional que desee que EventBridge Scheduler entregue al destino y, para algunos destinos con plantilla, un conjunto único de parámetros con opciones de configuración adicionales para ese destino. Cuando especificas el ARN de un recurso de destino con plantilla, EventBridge Scheduler asume automáticamente que quieres llamar a la operación de API compatible para ese servicio. Si quieres que EventBridge Scheduler se dirija a una operación de API diferente para el servicio, debes configurar el destino como un objetivo universal.

La siguiente es una lista completa de todos los destinos con plantillas compatibles con EventBridge Scheduler y, si corresponde, del conjunto único de parámetros asociados a cada objetivo. Elija el enlace de cada conjunto de parámetros para ver los campos obligatorios y opcionales en la referencia de la API del EventBridge programador.

- CodeBuild – `StartBuild`
- CodePipeline – `StartPipelineExecution`
- Amazon ECS – `RunTask`
  - Parámetros: `EcsParameters`
- EventBridge – `PutEvents`
  - Parámetros: `EventBridgeParameters`
- Amazon Inspector – `StartAssessmentRun`
- Kinesis – `PutRecord`
  - Parámetros: `KinesisParameters`
- Firehose — **`PutRecord`**
- Lambda: `Invoke`
- SageMaker – `StartPipelineExecution`

  - Parámetros: SageMakerPipelineParameters
- Amazon SNS – Publish
- Amazon SQS – SendMessage
  - Parámetros: SqsParameters
- Step Functions – StartExecution

Use los siguientes ejemplos para aprender a configurar diferentes destinos con plantillas y los permisos de IAM necesarios para cada destino descrito.

Amazon SQS **SendMessage**

Example Política de permisos para el rol de ejecución

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sqs:SendMessage"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Example AWS CLI

```
$ aws scheduler create-schedule --name sqs-templated --schedule-expression 'rate(5
 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"QUEUE_ARN", "Input": "Message for scheduleArn:
 '<aws.scheduler.schedule-arn>', scheduledTime: '<aws.scheduler.scheduled-time>'" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Example SDK de Python

```
import boto3
scheduler = boto3.client('scheduler')
```

```
flex_window = { "Mode": "OFF" }

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "Message for scheduleArn: '<aws.scheduler.schedule-arn>', scheduledTime:
 '<aws.scheduler.scheduled-time>'"
}

scheduler.create_schedule(
    Name="sqs-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)
```

Example SDK de Java

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target sqsTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("<QUEUE_ARN>")
                .input("Message for scheduleArn: '<aws.scheduler.schedule-arn>',
 scheduledTime: '<aws.scheduler.scheduled-time>'")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(sqsTarget)
```

```
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
                .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and an Amazon SQS
 templated target");
    }
}
```

Lambda **Invoke**

Example Política de permisos para el rol de ejecución

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Example AWS CLI

```
$ aws scheduler create-schedule --name lambda-templated-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"FUNCTION_ARN", "Input": "{ \"Payload\":
 \"TEST_PAYLOAD\" }" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Example SDK de Python

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }
```

```
lambda_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<LAMBDA_ARN>",
    "Input": "{ 'Payload': 'TEST_PAYLOAD' }"}
}

scheduler.create_schedule(
    Name="lambda-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=lambda_templated,
    FlexibleTimeWindow=flex_window)
```

Example SDK de Java

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target lambdaTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("<Lambda ARN>")
                .input("{ 'Payload': 'TEST_PAYLOAD' }")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE_NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(lambdaTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
```

```
                .clientToken("<Token GUID>")
                .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and Lambda templated
 target");
    }
}
```

Step Functions **`StartExecution`**

Example Política de permisos para el rol de ejecución

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "states:StartExecution"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Example AWS CLI

```
$ aws scheduler create-schedule --name sfn-templated-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"STATE_MACHINE_ARN", "Input": "{ \"Payload\":
 \"TEST_PAYLOAD\" }" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Example SDK de Python

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sfn_templated= {
```

```
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<STATE_MACHINE_ARN>",
    "Input": "{ 'Payload': 'TEST_PAYLOAD' }"
}

scheduler.create_schedule(Name="sfn-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=sfn_templated,
    FlexibleTimeWindow=flex_window)
```

Example SDK de Java

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target stepFunctionsTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("<STATE_MACHINE_ARN>")
                .input("{ 'Payload': 'TEST_PAYLOAD' }")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE_NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(stepFunctionsTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
                .clientToken("<Token GUID>")
                .build();
```

```
        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and Step Function
 templated target");
    }
}
```

Uso de destinos universales

Un destino universal es un conjunto de parámetros personalizables que le permiten invocar un conjunto más amplio de operaciones de API para muchos servicios de AWS. Por ejemplo, puede utilizar el parámetro de destino universal (UTP) para crear una nueva cola de Amazon SQS mediante la operación `CreateQueue`.

Para configurar un destino universal para su programación mediante la AWS CLI o uno de los SDK de el Programador de EventBridge, debe especificar la siguiente información:

- RoleArn: el ARN del rol de ejecución que desea usar para el destino. La función de ejecución que especifique debe tener los permisos para llamar a la operación de API a la que desea que se dirija su programación.
- Arn: el ARN del servicio completo, incluida la operación de API a la que desea dirigirse, en el siguiente formato: `arn:aws:scheduler:::aws-sdk:`*`service`*`:`*`apiAction`*.

  Por ejemplo, para Amazon SQS, el nombre del servicio que especifique es `arn:aws:scheduler:::aws-sdk:`**`sqs:sendMessage`**.
- Entrada: un JSON bien formado que se especifica con los parámetros de solicitud que el Programador de EventBridge envía a la API de destino. Los parámetros y la forma del JSON que establezca en `Input` vienen determinados por la API de servicio que invoque su programación. Para encontrar esta información, consulte la referencia de la API del servicio al que quiere dirigirse.

Acciones no admitidas

El Programador de EventBridge no admite acciones de API de solo lectura, como las operaciones `GET` comunes, que comiencen con la siguiente lista de prefijos:

```
get
describe
list
```

```
poll
receive
search
scan
query
select
read
lookup
discover
validate
batchGet
batchDescribe
batchRead
transactGet
adminGet
adminList
testMigration
retrieve
testConnection
translateDocument
isAuthorized
isAuthorizedWithToken
invokeModel
```

Por ejemplo, el ARN del servicio para la acción de la API `GetQueueUrl` sería el siguiente: `arn:aws:scheduler:::aws-sdk:sqs:getQueueURL`. Como la acción de la API comienza con el prefijo `get`, el Programador de EventBridge no admite este destino. Del mismo modo, la acción `ListBrokers` de Amazon MQ no se admite como destino porque la operación comienza con el prefijo `list`.

Ejemplos que utilizan el destino universal

Los parámetros que pase en el campo `Input` de programación dependen de los parámetros de solicitud que acepte la API de servicio que desea invocar. Por ejemplo, para apuntar a Lambda `Invoke`, puede configurar los parámetros que se enumeran en la Referencia de la API AWS Lambda. Esto incluye la carga JSON opcional que puede pasar a una función de Lambda.

Para determinar los parámetros que puede configurar para las distintas API, consulte la referencia de API de ese servicio. Al igual que Lambda `Invoke`, algunas API aceptan parámetros de URI, así como una carga del cuerpo de la solicitud. En esos casos, debe especificar los parámetros de la ruta del URI y la carga de JSON en su programación `Input`.

Los siguientes ejemplos muestran cómo utilizar el destino universal para invocar operaciones de API comunes con Lambda, Amazon SQS y Step Functions.

Example Lambda

```
$ aws scheduler create-schedule --name lambda-universal-schedule --schedule-expression
 'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"arn:aws:scheduler:::aws-sdk:lambda:invoke"
 "Input": "{\"FunctionName\":\"arn:aws:lambda:REGION:123456789012:function:HelloWorld
\",\"InvocationType\":\"Event\",\"Payload\":\"{\\\"message\\\":\\\"testing function\\
\"}\"}" }' \
--flexible-time-window '{ "Mode": "OFF"}'
```

Example Amazon SQS

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sqs_universal= {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "arn:aws:scheduler:::aws-sdk:sqs:sendMessage",
    "Input": "{\"MessageBody\":\"My message\",\"QueueUrl\":\"<QUEUE_URL>\"}"}
}

scheduler.create_schedule(
    Name="sqs-sdk-test",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_universal,
    FlexibleTimeWindow=flex_window)
```

Example Step Functions

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;


public class MySchedulerApp {
```

```
    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
                .region(Region.US_WEST_2)
                .build();

        Target stepFunctionsUniversalTarget = Target.builder()
                .roleArn("<ROLE_ARN>")
                .arn("arn:aws:scheduler:::aws-sdk:sfn:startExecution")
                .input("{\"Input\":\"{}\",\"StateMachineArn\":\"<STATE_MACHINE_ARN>
\"}")
                .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
                .name("<SCHEDULE_NAME>")
                .scheduleExpression("rate(10 minutes)")
                .target(stepFunctionsUniversalTarget)
                .flexibleTimeWindow(FlexibleTimeWindow.builder()
                        .mode(FlexibleTimeWindowMode.OFF)
                        .build())
                .clientToken("<Token GUID>")
                .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and Step Function
 universal target");
    }
}
```

Añadir atributos de contexto

Utilice las siguientes palabras clave en la carga útil que pase al destino para recopilar metadatos sobre la programación. Programador de Amazon EventBridge reemplaza cada palabra clave por su valor respectivo cuando su programación invoca el destino.

- **<aws.scheduler.schedule-arn>**— El ARN de la programación.
- **<aws.scheduler.scheduled-time>**— La hora que especificó para que la programación invocara su destino, por ejemplo, `2022-03-22T18:59:43Z`.
- **<aws.scheduler.execution-id>**— El identificador único que el Programador de EventBridge asigna a cada intento de invocación de un destino, por ejemplo `d32c5kddcf5bb8c3`.

- **<aws.scheduler.attempt-number>**— Un contador que identifica el número de intento de la invocación actual, por ejemplo, 1.

En este ejemplo, se muestra la creación de una programación que se active cada cinco minutos e invoque la operación `SendMessage` de Amazon SQS como destino universal. El cuerpo del mensaje incluye el valor de `schedule-time`.

Example AWS CLI

```
$ aws scheduler create-schedule --name your-schedule \
    --schedule-expression 'rate(5 minutes)' \
    --target '{"RoleArn": "ROLE_ARN", \
        "Arn": "arn:aws:scheduler:::aws-sdk:sqs:sendMessage", \
        "Input": "{\"MessageBody\":\"<aws.scheduler.scheduled-time>\",\"QueueUrl\":
\"https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-cli-test\"}"}' \
    --flexible-time-window '{ "Mode": "OFF"}'
```

Example SDK de Python

```
import boto3
scheduler = boto3.client('scheduler')

sqs_universal= {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "arn:aws:scheduler:::aws-sdk:sqs:sendMessage",
    "Input": "{\"MessageBody\":\"<aws.scheduler.scheduled-time>\",\"QueueUrl\":
\"https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-cli-test\"}"
}

flex_window = { "Mode": "OFF" }


scheduler.update_schedule(Name="your-schedule",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_universal,
    FlexibleTimeWindow=flex_window)
```

Siguientes pasos

Para obtener más información sobre los tipos de datos y las operaciones de la API de el Programador de EventBridge, consulte la Referencia de la API del Programador de EventBridge.

Seguridad en Amazon EventBridge Scheduler

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de conformidad que se aplican a Amazon EventBridge Scheduler, consulte AWS Servicios dentro del alcance por programa de conformidad AWS .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a entender cómo aplicar el modelo de responsabilidad compartida al utilizar EventBridge Scheduler. Los siguientes temas muestran cómo configurar EventBridge Scheduler para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger los recursos de EventBridge Scheduler.

Temas

Administrar el acceso a Amazon EventBridge Scheduler

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los recursos. AWS Los administradores de

IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar EventBridge los recursos del Scheduler. El IAM es un servicio Servicio de AWS que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realices en Scheduler. EventBridge

Usuario del servicio: si utiliza el servicio EventBridge Scheduler para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más funciones del EventBridge programador para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una función del EventBridge Scheduler, consulte. Solución de problemas de identidad y acceso a Amazon EventBridge Scheduler

Administrador de servicios: si está a cargo de los recursos de EventBridge Scheduler en su empresa, probablemente tenga acceso completo a EventBridge Scheduler. Su trabajo consiste en determinar a qué funciones y recursos del EventBridge Scheduler deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con EventBridge Scheduler, consulte. Cómo funciona EventBridge Scheduler con IAM

Administrador de IAM: si es administrador de IAM, puede que desee obtener más información sobre cómo redactar políticas para administrar el acceso a Scheduler. EventBridge Para ver ejemplos de políticas de EventBridge Scheduler basadas en la identidad que puede utilizar en IAM, consulte. Uso de políticas basadas en identidades

Autenticación con identidades

La autenticación es la forma de iniciar sesión con sus AWS credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, asumes un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Uso de la autenticación multifactor (MFA) en AWS en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios empresarial, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para más información, consulte ¿Qué es IAM Identity Center? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. El Centro de identidades de IAM correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder sus identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, en algunos casos Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los

permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar un AWS rol a una instancia EC2 y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifique el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para más información sobre Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations .
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades

del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona EventBridge Scheduler con IAM

Antes de usar IAM para administrar el acceso a EventBridge Scheduler, infórmese sobre las funciones de IAM disponibles para su uso con Scheduler. EventBridge

Funciones de IAM que puede utilizar con Amazon EventBridge Scheduler

Característica de IAM	EventBridge Soporte de Scheduler
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACL	No
ABAC (etiquetas en políticas)	Parcial
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí

Característica de IAM	EventBridge Soporte de Scheduler
Roles vinculados al servicio	No

Para obtener una visión general de cómo funcionan EventBridge Scheduler y otros AWS servicios con la mayoría de las funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas de Scheduler basadas en la identidad EventBridge

Compatibilidad con las políticas basadas en identidades	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para Scheduler EventBridge

Para ver ejemplos de políticas de EventBridge Scheduler basadas en la identidad, consulte. Uso de políticas basadas en identidades

Políticas basadas en recursos dentro de Scheduler EventBridge

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones políticas para EventBridge Scheduler

Admite acciones de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones del EventBridge programador, consulte Acciones definidas por Amazon EventBridge Scheduler en la Referencia de autorización de servicio.

Las acciones políticas del EventBridge Scheduler utilizan el siguiente prefijo antes de la acción:

```
scheduler
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "scheduler:action1",
      "scheduler:action2"
]
```

Puede utilizar caracteres comodín (*) para especificar varias acciones. Por ejemplo, para especificar todas las acciones que comiencen con la palabra `List`, incluya la siguiente acción:

```
"Action": [
   "scheduler:List*"
]
```

Recursos de políticas para Scheduler EventBridge

Admite recursos de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de los tipos de recursos de EventBridge Scheduler y sus ARN, consulte Recursos definidos por Amazon EventBridge Scheduler en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte Acciones definidas por Amazon EventBridge Scheduler.

Para ver ejemplos de políticas de EventBridge Scheduler basadas en la identidad, consulte. Uso de políticas basadas en identidades

Claves de condición de política para Scheduler EventBridge

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación lógica `AND`. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de las claves de condición del EventBridge programador, consulte Claves de condición de Amazon EventBridge Scheduler en la Referencia de autorización de servicio. Para

saber con qué acciones y recursos puede utilizar una clave de condición, consulte Acciones definidas por Amazon EventBridge Scheduler.

Para ver ejemplos de políticas de EventBridge Scheduler basadas en la identidad, consulte. Uso de políticas basadas en identidades

ACL en Scheduler EventBridge

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con programador EventBridge

Admite ABAC (etiquetas en las políticas)	Parcial

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con Scheduler EventBridge

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobrel cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales entre servicios para EventBridge Scheduler

Admite Forward access sessions (FAS)	Sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en él AWS, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las

solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Funciones de servicio para EventBridge Scheduler

Compatible con roles de servicio	Sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad del EventBridge Scheduler. Edite las funciones de servicio solo cuando el EventBridge Programador le indique cómo hacerlo.

Funciones vinculadas al servicio para Scheduler EventBridge

Compatible con roles vinculados al servicio	No

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a servicios. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Uso de políticas basadas en identidades

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar los recursos del EventBridge Scheduler. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede agregar las políticas de IAM a los roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por EventBridge Scheduler, incluido el formato de los ARN de cada uno de los tipos de recursos, consulte Acciones, recursos y claves de condición de Amazon EventBridge Scheduler en la Referencia de autorización de servicios.

Temas

- Prácticas recomendadas sobre las políticas
- EventBridge Permisos del programador
- AWS políticas gestionadas para EventBridge Scheduler
- Políticas administradas por el cliente para Scheduler EventBridge
- AWS actualizaciones de políticas gestionadas

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear, acceder o eliminar los recursos de EventBridge Scheduler de su cuenta. Estas acciones pueden generar costes adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos

definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía del usuario de IAM.

- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para más información, consulte Elementos de política JSON de IAM: condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

EventBridge Permisos del programador

Para que un director de IAM (usuario, grupo o rol) pueda crear horarios en el EventBridge Scheduler y acceder a los recursos del EventBridge Scheduler a través de la consola o la API, el director debe tener un conjunto de permisos agregado a su política de permisos. Puede configurar estos permisos

en función de la función de trabajo de la entidad principal. Por ejemplo, un usuario o rol que solo usa la consola de EventBridge Scheduler para ver una lista de los horarios existentes no necesita tener los permisos necesarios para llamar a la operación de la API. `CreateSchedule` Le recomendamos que adapte sus permisos basados en la identidad para proporcionar solo el acceso con menos privilegios.

La siguiente lista muestra los recursos de EventBridge Scheduler y sus correspondientes acciones compatibles.

- Programación
  - `scheduler:ListSchedules`
  - `scheduler:GetSchedule`
  - `scheduler:CreateSchedule`
  - `scheduler:UpdateSchedule`
  - `scheduler:DeleteSchedule`
- Grupo de programaciones
  - `scheduler:ListScheduleGroups`
  - `scheduler:GetScheduleGroup`
  - `scheduler:CreateScheduleGroup`
  - `scheduler:DeleteScheduleGroup`
  - `scheduler:ListTagsForResource`
  - `scheduler:TagResource`
  - `scheduler:UntagResource`

Puede usar los permisos de EventBridge Scheduler para crear sus propias políticas administradas por el cliente y utilizarlas con EventBridge Scheduler. También puede usar las políticas AWS administradas que se describen en la siguiente sección para conceder los permisos necesarios para casos de uso comunes sin tener que administrar sus propias políticas.

AWS políticas gestionadas para EventBridge Scheduler

AWS aborda muchos casos de uso comunes al proporcionar políticas de IAM independientes que AWS crean y administran. 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 por 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 Scheduler: EventBridge

- the section called "AmazonEventBridgeSchedulerFullAccess"— Otorga acceso completo a EventBridge Scheduler mediante la consola y la API.
- the section called "AmazonEventBridgeSchedulerReadOnlyAccess"— Otorga acceso de solo lectura al Scheduler. EventBridge

AmazonEventBridgeSchedulerFullAccess

La política AmazonEventBridgeSchedulerFullAccess gestionada otorga permisos para usar todas las acciones del EventBridge programador para los horarios y grupos de horarios.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "scheduler:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonEventBridgeSchedulerReadOnlyAccess

La política administrada por AmazonEventBridgeSchedulerReadOnlyAccess concede permisos de solo lectura para ver los detalles de sus programaciones y grupos de programaciones.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "scheduler:ListSchedules",
                "scheduler:ListScheduleGroups",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Políticas administradas por el cliente para Scheduler EventBridge

Utilice los siguientes ejemplos para crear sus propias políticas administradas por el cliente para EventBridge Scheduler. Las políticas administradas por el cliente le permiten conceder permisos únicamente para las acciones y los recursos necesarios para las aplicaciones y los usuarios de su equipo en función de la función de trabajo de la entidad principal.

Temas

- Ejemplo: CreateSchedule
- Ejemplo: GetSchedule
- Ejemplo: UpdateSchedule
- Ejemplo: DeleteScheduleGroup

Ejemplo: **`CreateSchedule`**

Al crear un nuevo programa, puede elegir si desea cifrar sus datos en EventBridge Scheduler mediante una clave gestionada por el cliente o mediante una clave Clave propiedad de AWSgestionada por el cliente.

La siguiente política permite a la entidad principal crear una programación y aplicar el cifrado mediante una Clave propiedad de AWS. Con una Clave propiedad de AWS, AWS gestiona los recursos en AWS Key Management Service (AWS KMS) por ti, por lo que no necesitas permisos adicionales para interactuar con ellos. AWS KMS

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:CreateSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

Use la siguiente política para permitir que un director cree un cronograma y use una clave administrada por el AWS KMS cliente para el cifrado. Para usar una clave administrada por el cliente, el director debe tener permiso para acceder a los AWS KMS recursos de su cuenta. Esta política otorga acceso a una única clave KMS específica que se utilizará para cifrar los datos en EventBridge Scheduler. Como alternativa, puede usar un carácter comodín (*) para conceder acceso a todas las claves de una cuenta o a un subconjunto que coincida con un patrón de nombres determinado.

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

```
            [
                "scheduler:CreateSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
            ]
        },
        {
            "Action":
            [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:kms:us-west-2:123456789012:key/my-key-id"
            ],
            "Conditions": {
                "StringLike": {
                    "kms:ViaService": "scheduler.amazonaws.com",
                    "kms:EncryptionContext:aws:scheduler:schedule:arn":
 "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
            }
        }
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

Ejemplo: **`GetSchedule`**

Utilice la siguiente política para permitir que la entidad principal obtenga información sobre una programación.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:GetSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
            ]
        }
    ]
}
```

Ejemplo: **`UpdateSchedule`**

Utilice las siguientes políticas para permitir que una entidad principal actualice una programación llamando a la acción `scheduler:UpdateSchedule`. Del mismo modo`CreateSchedule`, la política depende de si el programa utiliza una clave administrada por el cliente AWS KMS Clave propiedad de AWS o una clave administrada por el cliente para el cifrado. Para un programa configurado con una Clave propiedad de AWS, utilice la siguiente política:

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:UpdateSchedule"
            ],
            "Effect": "Allow",
```

```
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

Para una programación configurada con una clave administrada por el cliente, utilice la siguiente política. Esta política incluye permisos adicionales que permiten a un director acceder a AWS KMS los recursos de su cuenta:

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Action":
            [
                "scheduler:UpdateSchedule"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name
        },
        {
            "Action":
            [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
```

```
                "kms:Decrypt"
            ],
            "Effect": "Allow",
            "Resource":
            [
                "arn:aws:kms:us-west-2:123456789012:key/my-key-id"
            ],
            "Conditions": {
                "StringLike": {
                    "kms:ViaService": "scheduler.amazonaws.com",
                    "kms:EncryptionContext:aws:scheduler:schedule:arn":
 "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
            }
        }
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

Ejemplo: **`DeleteScheduleGroup`**

Use la siguiente política para permitir que una entidad principal elimine un grupo de programaciones. Cuando elimina un grupo, también elimina las programaciones asociadas a ese grupo. La entidad principal que elimine el grupo debe tener permiso para eliminar también las programaciones asociadas a ese grupo. Esta política otorga un permiso a la entidad principal para ejecutar la acción `scheduler:DeleteScheduleGroup` en los grupos de programaciones especificados, así como en todas las programaciones del grupo:

> **Note**
>
> EventBridge El programador no admite la especificación de permisos a nivel de recursos para programaciones individuales. Por ejemplo, la siguiente declaración no es válida y no debe incluirse en la política:

```
    "Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-
    group/my-schedule-name"
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "scheduler:DeleteSchedule",
            "Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/*"
        },
        {
            "Effect": "Allow",
            "Action": "scheduler:DeleteScheduleGroup",
            "Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        }
    ]
}
```

AWS actualizaciones de políticas gestionadas

Cambio	Descripción	Fecha
the section called “AmazonEventBridgeSchedulerFullAccess” – Nueva política administrada	EventBridge Scheduler añade compatibilidad con una nueva política gestionada que otorga a los usuarios acceso total a todos los recursos, incluidos	10 de noviembre de 2022

Cambio	Descripción	Fecha
	los programas y los grupos de programas.	
the section called "AmazonEventBridgeSchedulerReadOnlyAccess" – Nueva política administrada	EventBridge Scheduler añade compatibilidad con una nueva política administrada que otorga a los usuarios acceso de solo lectura a todos los recursos, incluidos los horarios y los grupos de horarios.	10 de noviembre de 2022
EventBridge Scheduler comenzó a rastrear los cambios	EventBridge Scheduler comenzó a rastrear los cambios de sus políticas AWS gestionadas.	10 de noviembre de 2022

Prevención del suplente confuso

El problema de la sustitución confusa 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 de identidad entre servicios puede provocar el confuso problema de un diputado. 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.

Le recomendamos que utilice las claves de contexto de condición `aws:SourceAccount`global `aws:SourceArn`y las claves de contexto en su función de ejecución programada para limitar los permisos que EventBridge Scheduler concede a otro servicio para acceder al 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 de la sustitución confusa es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. La siguiente condición se aplica a un grupo de programación individual: `arn:aws:scheduler:*:123456789012:schedule-group/your-schedule-group`

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:scheduler:*:123456789012:schedule-group/*`.

El valor de `aws:SourceArn` debe ser el ARN del grupo de programación del EventBridge Scheduler al que desea limitar esta condición.

> **Important**
>
> No limite la declaración `aws:SourceArn` a una programación específica ni a un prefijo de nombre de programación. El ARN que especifique debe ser un grupo de programaciones.

El siguiente ejemplo muestra cómo puede utilizar las claves de contexto de condición global `aws:SourceArn` y `aws:SourceAccount` en su política de confianza de rol de ejecución para evitar el problema del suplente confuso:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "scheduler.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012",
                    "aws:SourceArn": "arn:aws:scheduler:us-
west-2:123456789012:schedule-group/your-schedule-group"
                }
            }
        }
    ]
```

```
}
```

Solución de problemas de identidad y acceso a Amazon EventBridge Scheduler

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas habituales que pueden surgir al trabajar con EventBridge Scheduler e IAM.

Temas

- No estoy autorizado a realizar ninguna acción en Scheduler EventBridge
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de EventBridge Scheduler

No estoy autorizado a realizar ninguna acción en Scheduler EventBridge

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `scheduler:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 scheduler:GetWidget on resource: my-example-widget
```

En este caso, la política de Mateo se debe actualizar para permitirle acceder al recurso *`my-example-widget`* mediante la acción `scheduler:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibes un mensaje de error que indica que no estás autorizado a realizar la `iam:PassRole` acción, debes actualizar tus políticas para que puedas transferir una función a EventBridge Scheduler.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

El siguiente ejemplo de error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en el Scheduler. EventBridge Sin embargo, la acción requiere que el servicio cuente con permisos que concede un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de EventBridge Scheduler

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si EventBridge Scheduler admite estas funciones, consulte. Cómo funciona EventBridge Scheduler con IAM
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.

- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (federación de identidades) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Protección de datos en Amazon EventBridge Scheduler

El modelo de se aplica a protección de datos en Amazon EventBridge Scheduler. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan 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 AWS Shared Responsibility Model and 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 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 con EventBridge Scheduler u otro tipo de herramienta Servicios de AWS mediante 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 la información de las credenciales en la URL para validar la solicitud para ese servidor.

Temas

Cifrado en reposo

En esta sección se describe cómo Amazon EventBridge Scheduler cifra y descifra los datos en reposo. Los datos en reposo son datos almacenados en EventBridge Scheduler y en los componentes subyacentes del servicio. EventBridge Scheduler se integra con AWS Key Management Service (AWS KMS) para cifrar y descifrar sus datos mediante un. AWS KMS key EventBridge Scheduler admite dos tipos de claves KMS: y claves administradas por Claves propiedad de AWSel cliente.

> **Note**
>
> EventBridge Scheduler solo admite el uso de claves KMS de cifrado simétrico.

Claves propiedad de AWSson claves KMS que un AWS servicio posee y administra para su uso en varias AWS cuentas. Si bien los usos del Claves propiedad de AWS EventBridge Programador no se almacenan en su AWS cuenta, el EventBridge Programador los usa para proteger sus datos y recursos. De forma predeterminada, EventBridge Scheduler cifra y descifra todos los datos con una clave propia. AWS No es necesario administrar su Clave propiedad de AWS ni su política de acceso. Cuando EventBridge Scheduler los utiliza Claves propiedad de AWS para proteger sus datos, no se le cobrará ningún cargo y su uso no se incluirá en las cuotas de su AWS KMS cuenta.

Las claves administradas por el cliente son claves de KMS almacenadas en su AWS cuenta que usted crea, posee y administra. Si su caso de uso específico requiere que controle y audite las claves

de cifrado que protegen sus datos en EventBridge Scheduler, puede usar una clave administrada por el cliente. Si elige una clave administrada por el cliente, debe administrar su política de claves. Las claves administradas por el cliente tienen una tarifa mensual y una tarifa por uso excesivo del nivel gratuito. El uso de una clave administrada por el cliente también cuenta como parte de su cuota de AWS KMS. Para obtener más información acerca de los precios, consulte Precios de AWS Key Management Service.

Temas

- Artefactos de cifrado
- Administrar las claves de KMS
- CloudTrail ejemplo de evento

Artefactos de cifrado

En la siguiente tabla se describen los distintos tipos de datos que EventBridge Scheduler cifra en reposo y qué tipo de clave de KMS admite para cada categoría.

Tipo de datos	Descripción	Clave propiedad de AWS	clave administrada por el cliente
Carga (hasta 256 KB)	Los datos que se especifican en el parámetro `TargetInput` de la programación al configurar la programación para que se entregue al destino.	Compatible	Soportado
Identificador y estado	El nombre exclusivo y el estado (activar o desactivar) de la programación.	Compatible	No compatible
Configuración de programación	La expresión de programación, como	Compatible	No compatible

Tipo de datos	Descripción	Clave propiedad de AWS	clave administrada por el cliente
	la expresión de frecuencia o cron para las programac iones recurrentes y la marca de tiempo para las invocacio nes únicas, así como la fecha de inicio, la fecha de finalización y la zona horaria de la programación.		
Configuración de destino	El nombre de recurso de Amazon (ARN) y otros detalles de configuración relacionados con el destino.	Compatible	No compatible
Configuración del comportamiento de invocación y error	La configuración flexible del intervalo de tiempo, la política de reintentos de la programación y los detalles de las colas de mensajes fallidos que se utilizan en las entregas fallidas.	Compatible	No compatible

EventBridge Scheduler usa las claves administradas por el cliente solo al cifrar y descifrar la carga útil de destino, como se describe en la tabla anterior. Si eliges usar una clave gestionada por el cliente, EventBridge Scheduler cifra y descifra la carga dos veces: una con la predeterminada y otra con la clave gestionada por Clave propiedad de AWS el cliente que especifiques. Para todos los

demás tipos de datos, EventBridge Scheduler solo usa la predeterminada Clave propiedad de AWS para proteger los datos en reposo.

Utilice la siguiente the section called “Administrar las claves de KMS” sección para obtener información sobre cómo debe gestionar sus recursos de IAM y sus políticas clave para poder utilizar una clave gestionada por el cliente con EventBridge Scheduler.

Administrar las claves de KMS

Si lo desea, puede proporcionar una clave gestionada por el cliente para cifrar y descifrar la carga útil que su programa entrega a su destino. EventBridge El programador cifra y descifra tu carga útil de hasta 256 KB de datos. El uso de una clave administrada por el cliente genera una tarifa mensual y una tarifa superior al nivel gratuito. El uso de una clave administrada por el cliente cuenta como parte de su cuota de AWS KMS. Para obtener más información acerca de los precios, consulte Precios de AWS Key Management Service

EventBridge El programador utiliza los permisos de IAM asociados al principal, lo que crea un cronograma para cifrar los datos. Esto significa que debe adjuntar los permisos AWS KMS relacionados necesarios al usuario, o rol, que llama a la API de Scheduler. EventBridge Además, EventBridge Scheduler utiliza políticas basadas en recursos para descifrar los datos. Esto significa que la función de ejecución asociada a su programación también debe tener los permisos AWS KMS correspondientes necesarios para llamar a la AWS KMS API al descifrar los datos.

> **Note**
>
> EventBridge Scheduler no admite el uso de concesiones para permisos temporales.

Consulta la siguiente sección para obtener información sobre cómo gestionar tu política de AWS KMS claves y los permisos de IAM necesarios para utilizar una clave gestionada por el cliente en EventBridge Scheduler.

Temas

- Añadir permisos de IAM
- Administrar la política de claves

Añadir permisos de IAM

Para usar una clave administrada por el cliente, debe añadir los siguientes permisos a la entidad principal de IAM basada en la identidad que crea una programación, así como al rol de ejecución que asocia al cronograma.

Permisos basados en identidades para claves administradas por el cliente

Debe añadir AWS KMS las siguientes acciones a la política de permisos asociada a cualquier entidad principal (usuarios, grupos o funciones) que llame a la API de EventBridge Scheduler al crear un cronograma.

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

                # Required to pass the execution role
                "iam:PassRole",

                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
    ]
}
```

- **`kms:DescribeKey`**— Necesaria para validar que la clave que ha proporcionado es una clave KMS de cifrado simétrico.
- **`kms:GenerateDataKey`**— Necesaria para generar la clave de datos que EventBridge Scheduler utiliza para realizar el cifrado del lado del cliente.
- **`kms:Decrypt`**— Es necesario descifrar la clave de datos cifrados que EventBridge Scheduler almacena junto con sus datos cifrados.

Permisos de función de ejecución para las claves administradas por el cliente

Debe añadir la siguiente acción a la política de permisos de las funciones de ejecución de su programa para que EventBridge Scheduler pueda llamar a la AWS KMS API al descifrar sus datos.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
      "Sid" : "Allow EventBridge Scheduler to decrypt data using a customer managed
 key",
      "Effect" : "Allow",
      "Action" : [
        "kms:Decrypt"

      ],
      "Resource": "arn:aws:kms:your-region:123456789012:key/your-key-id"
    }
  ]
}
```

- **`kms:Decrypt`**— Es necesario descifrar la clave de datos cifrados que EventBridge Scheduler almacena junto con sus datos cifrados.

Si utiliza la consola de EventBridge Scheduler para crear una nueva función de ejecución al crear una nueva programación, EventBridge Scheduler adjuntará automáticamente el permiso necesario a su función de ejecución. Sin embargo, si elige un rol de ejecución existente, debe añadir los permisos necesarios al rol para poder utilizar las claves administradas por el cliente.

Administrar la política de claves

Al crear una clave gestionada por el cliente utilizando AWS KMS, de forma predeterminada, la clave tiene la siguiente política clave para proporcionar acceso a las funciones de ejecución de los horarios.

```
{
    "Id": "key-policy-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Provide required IAM Permissions",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

Si lo desea, puede limitar el alcance de su política de claves para proporcionar acceso únicamente al rol de ejecución. Puede hacerlo si desea utilizar la clave gestionada por el cliente únicamente con los recursos del EventBridge Scheduler. Usa el siguiente ejemplo de política clave para limitar los recursos del EventBridge programador que pueden usar tu clave.

```
{
    "Id": "key-policy-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Provide required IAM Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::695325144837:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/schedule-execution-role"
            },
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudTrail ejemplo de evento

AWS CloudTrail captura todos los eventos de llamadas a la API. Esto incluye las llamadas a la API siempre que EventBridge Scheduler utilice la clave gestionada por el cliente para descifrar los datos. El siguiente ejemplo muestra una entrada de CloudTrail evento que demuestra que EventBridge Scheduler utiliza la `kms:Decrypt` acción mediante una clave gestionada por el cliente.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
      "type": "AssumedRole",
      "principalId": "ABCDEABCD1AB12ABABAB0:70abcd123a123a12345a1aa12aa1bc12",
      "arn": "arn:aws:sts::123456789012:assumed-role/execution-
role/70abcd123a123a12345a1aa12aa1bc12",
      "accountId": "123456789012",
      "accessKeyId": "ABCDEFGHI1JKLMNOP2Q3",
      "sessionContext": {
          "sessionIssuer": {
              "type": "Role",
              "principalId": "ABCDEABCD1AB12ABABAB0",
              "arn": "arn:aws:iam::123456789012:role/execution-role",
              "accountId": "123456789012",
              "userName": "execution-role"
          },
          "webIdFederationData": {},
          "attributes": {
              "creationDate": "2022-10-31T21:03:15Z",
              "mfaAuthenticated": "false"
          }
      }
    },
    "eventTime": "2022-10-31T21:03:15Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-north-1",
    "sourceIPAddress": "13.50.87.173",
    "userAgent": "aws-sdk-java/2.17.295 Linux/4.14.291-218.527.amzn2.x86_64 OpenJDK_64-
Bit_Server_VM/11.0.17+9-LTS Java/11.0.17 kotlin/1.3.72-release-468 (1.3.72) vendor/
Amazon.com_Inc. md/internal exec-env/AWS_ECS_FARGATE io/sync http/Apache cfg/retry-
mode/standard AwsCrypto/2.4.0",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-west-2:123456789012:key/2321abab-2110-12ab-a123-
a2b34c5abc67",
```

```
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "aws:scheduler:schedule:arn": "arn:aws:scheduler:us-
west-2:123456789012:schedule/default/execution-role"
        }
    },
    "responseElements": null,
    "requestID": "request-id",
    "eventID": "event-id",
    "readOnly": true,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:123456789012:key/2321abab-2110-12ab-a123-
a2b34c5abc67"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_256_GCM_SHA384",
      "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
  }
}
```

Cifrado en tránsito

EventBridge Scheduler cifra los datos en tránsito a medida que viajan por la red. Transport Layer Security (TLS) cifra tus datos cuando llamas a cualquier operación de la API de EventBridge Scheduler, así como cuando EventBridge Scheduler llama a cualquier API de destino cuando invoca tu programación. De forma predeterminada, EventBridge Scheduler usa TLS 1.2 al cifrar los datos en tránsito. No es necesario configurar el cifrado en tránsito y no se puede elegir una versión de TLS diferente cuando se utiliza Scheduler. EventBridge

Uso de la API de EventBridge Scheduler: cuando realizas una operación de API, por ejemplo`CreateSchedule`, EventBridge Scheduler cifra toda la solicitud HTTP, incluidos el cuerpo y los encabezados de la solicitud. EventBridge El programador también cifra todo el objeto de respuesta que recibes de nuestras API.

Uso de las API de destino: cuando EventBridge Scheduler invoca tu programación, llama a la API de destino que especificaste al crear la programación. Al enviar un evento a un destino, EventBridge Scheduler cifra toda la solicitud, incluidos el cuerpo de la solicitud y todos los encabezados, así como la respuesta que recibe del destino.

Validación de conformidad para Amazon EventBridge Scheduler

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de cumplimiento Servicios de AWS de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en AWS la seguridad y el cumplimiento.
- Diseño de arquitectura para garantizar la seguridad y el cumplimiento de la HIPAA en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar AWS las empresas para crear aplicaciones aptas para la HIPAA.

  > **Note**
  >
  > No Servicios de AWS todas cumplen los requisitos de la HIPAA. Para más información, consulte la Referencia de servicios compatibles con HIPAA.

- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde la perspectiva del cumplimiento. Las guías resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y orientan los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas de Seguridad del Sector de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).

- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulte la Referencia de controles de Security Hub.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS consumo para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en Amazon EventBridge Scheduler

La infraestructura AWS global se basa Regiones de AWS en zonas de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. 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 uno o varios centros de datos.

Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global. AWS

Además de la infraestructura AWS global, EventBridge Scheduler ofrece varias funciones para ayudarlo a satisfacer sus necesidades de respaldo y resiliencia de datos.

Seguridad de infraestructura en Amazon EventBridge Scheduler

Como servicio gestionado, Amazon EventBridge Scheduler está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a EventBridge Scheduler 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 principal de seguridad de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Monitorización y métricas del Programador de Amazon EventBridge

La supervisión es una parte importante para mantener la fiabilidad, la disponibilidad y el rendimiento del Programador de Amazon EventBridge y sus otras soluciones de AWS. AWS proporciona las siguientes herramientas de supervisión para observar el Programador EventBridge, informar cuando algo anda mal y tomar acciones automáticas cuando sea apropiado:

- Amazon CloudWatch monitorea los recursos de AWS y las aplicaciones que ejecuta en AWS en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Para obtener más información, consulte la Guía del usuario de Amazon CloudWatch.
- AWS CloudTrail captura llamadas a la API y eventos relacionados efectuados por su cuenta de AWS o en su nombre, y entrega los archivos de registro al bucket de Amazon S3 que se haya especificado. También pueden identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que se hicieron. Para obtener más información, consulte la Guía del usuario de AWS CloudTrail.

Temas

Monitorización del Programador de Amazon EventBridge con Amazon CloudWatch

Puede supervisar el Programador de Amazon EventBridge mediante Amazon CloudWatch, que recopila y procesa los datos sin procesar y los convierte en métricas legibles y casi en tiempo real. El Programador de EventBridge emite un conjunto de métricas para todas las programaciones y un conjunto adicional de métricas para las programaciones que tienen una cola de mensajes fallidos (DLQ) asociada. Si configura una DLQ para su programación, el Programador de EventBridge publica métricas adicionales cuando su programación agota su política de reintentos.

Estas estadísticas se conservarán durante 15 meses, lo que le permitirá acceder a información histórica y dispondrá de una mejor perspectiva acerca de por qué un programa está fallando y podrá solucionar problemas subyacentes. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulte la Guía del usuario de Amazon CloudWatch.

Temas

Términos

Espacio de nombres

Los espacios de nombres son contenedores para métricas de CloudWatch de servicios de AWS. Para el Programador de EventBridge, el espacio de nombres es `AWS/Scheduler`.

Métricas de CloudWatch

Las métricas representan una serie de puntos de datos ordenados por tiempo que se publican en CloudWatch.

Dimensión

Una dimensión es un par de nombre-valor que forma parte de la identidad de una métrica.

Unidad

Las estadísticas tienen unidades de medida. En el caso del Programador de EventBridge, las unidades incluyen Recuento.

Dimensiones

En esta sección, se describe la agrupación de dimensiones de CloudWatch para las métricas del Programador de EventBridge en CloudWatch.

Dimensión	Descripción
ScheduleGroup	El grupo de programaciones para las que desea ver las métricas mediante CloudWatch. Si aún no ha creado ningún grupo, el Programador de EventBridge asocia sus programaciones al grupo `default`.

Acceder a las métricas de

En esta sección se describe cómo acceder a las métricas de rendimiento de CloudWatch para una programación específica del Programador de EventBridge.

Pasos para consultar las métricas de rendimiento de una dimensión

1. Abra la página de métricas en la consola de CloudWatch.
2. En el selector de regiones de AWS, seleccione la región para su programación
3. Elija el espacio de nombres del Programador.
4. En la pestaña Todas las métricas, elija una dimensión, por ejemplo, Planificar métricas de grupos. Para ver las métricas de todas las programaciones que ha creado en la región seleccionada, seleccione Métricas de la cuenta.
5. Elija una métrica de CloudWatch para la dimensión. Por ejemplo, InvocationAttemtCount o InvocationDroppedCount y, a continuación, seleccione Búsqueda de gráficos.
6. Seleccione la pestaña Métricas diagramadas para ver las estadísticas de rendimiento de las métricas del Programador de EventBridge.

Lista de métricas

En las tablas siguientes se muestran las métricas de todas las programaciones del Programador de EventBridge, así como métricas adicionales de las programaciones para los que se ha configurado un DLQ.

Métricas de todas las programaciones

Espacio de nombres	Métrica	Unidad	Descripción
AWS/Scheduler	InvocationAttemptCount	Recuento	Se emite por cada intento de invocación. Use esta métrica para comprobar que el Programador de EventBridge está intentando invocar sus programaciones y para ver cuándo las invocaciones se acercan a las cuotas de su cuenta.
AWS/Scheduler	TargetErrorCount	Recuento	Se emite cuando el destino devuelve una excepción después de que el Programador de EventBridge llame a la API de destino.

Espacio de nombres	Métrica	Unidad	Descripción
			Utilícela para comprobar cuándo se produce un error en la entrega a un destino.
AWS/Scheduler	TargetErrorThrottledCount	Recuento	Se emite cuando la invocación del destino falla debido a la limitación de la API por parte del destino. Úselo para diagnosticar errores de entrega cuando el motivo subyacente sea la API de destino que limita las llamadas realizadas por el Programador de EventBridge

Espacio de nombres	Métrica	Unidad	Descripción
AWS/Scheduler	InvocationThrottleCount	Recuento	Se emite cuando el Programador de EventBridge limita una invocación de destino porque supera las cuotas de servicio establecidas por el Programador de EventBridge. Úselo para determinar cuándo ha superado sus cuotas del Programador de EventBridge. Para obtener más información acerca de las cuotas de servicio, consulte Cuotas.

Espacio de nombres	Métrica	Unidad	Descripción
AWS/Scheduler	InvocationDroppedCount	Recuento	Se emite cuando el Programador de EventBridge deja de intentar invocar el destino una vez agotada la política de reintentos de una programación. Para obtener más información sobre las políticas de reintentos, consulte RetryPolicy en la Referencia de la API del Programador de EventBridge.

Métricas de las programaciones con una DLQ

Espacio de nombres	Métrica	Unidad	Descripción
AWS/Scheduler	InvocationsSentToDeadLetterCount	Recuento	Se emiten por cada

Espacio de nombres	Métrica	Unidad	Descripción
			entrega exitosa a la DLQ de una programación. Utilice esta opción para determinar cuándo se envían los eventos a una DLQ y, a continuación, compruebe el evento enviado a la DLQ de la programación para obtener información adicional que le ayude a determinar la causa del error.

Espacio de nombres	Métrica	Unidad	Descripción
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount	Recuento	Se emite cuando el Programador de EventBridge no puede enviar un evento a la DLQ. Utilice estas dos métricas para determinar el motivo por el que el Programador de EventBridge no puede enviar un evento a la DLQ y modifique la configuración de la DLQ para resolver el problema. El siguiente es un ejemplo de la métrica InvocationsFailedToBeSentToDeadLetterCount_<error_code
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount_<error_code>	Recuento	

Espacio de nombres	Métrica	Unidad	Descripción
			> cuando la cola de Amazon SQS que especificó como DLQ no existe: `InvocationsFailedToBeSentToDeadLetterCount_` **AWS.SimpleQueueService.NonExistentQueue**

Espacio de nombres	Métrica	Unidad	Descripción
AWS/Scheduler	InvocationsSentToDeadLetterCount_Truncated_MessageSizeExceeded	Recuento	Se emite cuando la carga del evento enviado a la DLQ supera el tamaño máximo permitido por Amazon SQS y el Programador de EventBridge trunca la carga útil que especifique en el atributo Input de una programación.

Registro de llamadas a la API del Programador de Amazon EventBridge mediante AWS CloudTrail

El Programador de Amazon EventBridge está integrado con AWS CloudTrail, un servicio que registra las acciones de los usuarios, los roles o los servicios de AWS en el Programador de EventBridge. CloudTrail captura las llamadas a la API del Programador de EventBridge como eventos. Las llamadas capturadas incluyen las llamadas realizadas desde la consola del Programador de EventBridge, así como las llamadas de código realizadas a las operaciones de API de Programador de EventBridge. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los del Programador de EventBridge. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el

Historial de eventos. Mediante la información recopilada por CloudTrail, puede determinar la solicitud que se realizó al Programador de EventBridge, la dirección IP desde la que se realizó, quién la realizó y cuándo, etc.

Para obtener más información acerca de CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

Información del Programador de EventBridge en CloudTrail

CloudTrail se habilita en su Cuenta de AWS cuando la crea. Cuando se produce una actividad en el Programador de EventBridge, la actividad se registra en un evento de CloudTrail junto con otros eventos de servicios de AWS en el Historial de eventos. Puede ver, buscar y descargar los últimos eventos de la Cuenta de AWS. Para obtener más información, consulte Ver eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de eventos de Cuenta de AWS, incluidos los del Programador de EventBridge, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail
- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de registro de CloudTrail de varias regiones y Recibir archivos de registro de CloudTrail de varias cuentas

CloudTrail registra todas las acciones del Programador de EventBridge, que se documentan en la Referencia de la API del Programador de Amazon EventBridge. Por ejemplo, las llamadas a las acciones `CreateSchedule`, `UpdateSchedule` y `DeleteSchedule` generan entradas en los archivos de registros de CloudTrail.

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 credenciales de usuario AWS Identity and Access Management (IAM) o credenciales de usuario raíz.
- 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.

Para obtener más información, consulte el elemento userIdentity de CloudTrail.

Descripción de las entradas de archivos de registro de Programador de EventBridge

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registros en un bucket de Amazon S3 que especifique. Los archivos log de CloudTrail pueden contener una o varias entradas de log. Un evento representa una solicitud específica realizada desde un origen y contiene información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. Los archivos de registro de CloudTrail no rastrean el orden en la pila de las llamadas públicas a la API, por lo que estas no aparecen en ningún orden específico.

Cuotas para Amazon EventBridge Scheduler

Tu AWS cuenta tiene cuotas predeterminadas, antes denominadas límites, para cada AWS servicio. A menos que se indique lo contrario, cada cuota es específica de la región de . Puede solicitar aumentos para algunas cuotas y otras no se pueden aumentar.

Para ver las cuotas de EventBridge Scheduler, abra la consola Service Quotas. En el panel de navegación, selecciona AWS servicios y, a continuación, EventBridge Scheduler.

Para solicitar un aumento de cuota, consulte Solicitud de un aumento de cuota en la Guía de usuario de Service Quotas. Si la cuota aún no se encuentra disponible en Service Quotas, utilice el formulario de aumento del límite.

> **Note**
>
> Las cuotas`CreateSchedule`, `UpdateScheduleGetSchedule`, y de `DeleteSchedule` transacciones por segundo (TPS) de EventBridge Scheduler se pueden ajustar hasta miles de TPS. La cuota de limitación de invocaciones se puede ajustar hasta decenas de miles de TPS.

Tu AWS cuenta tiene las siguientes cuotas relacionadas con Scheduler. EventBridge

Nombre	Valor predeterminado	Ajusta	Descripción
CreateSchedule tasa de solicitud	Cada región admitida: 50	Sí	Número máximo de CreateSchedule solicitud es por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.
CreateScheduleGroup tasa de solicitudes	Cada región admitida: 10	Sí	Número máximo de CreateScheduleGroup

Nombre	Valor predeterminado	Ajusta	Descripción
			solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.
DeleteSchedule tasa de solicitudes	Cada región admitida: 50	Sí	Número máximo de DeleteSchedule solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.
DeleteScheduleGroup tasa de solicitudes	Cada región admitida: 10	Sí	Número máximo de DeleteScheduleGroup solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.
GetSchedule tasa de solicitudes	Cada región admitida: 50	Sí	Número máximo de GetSchedule solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.

Nombre	Valor predeterminado	Ajusta	Descripción
GetScheduleGroup tasa de solicitudes	Cada región admitida: 10	Sí	Número máximo de GetScheduleGroup solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.
Límite de la limitación de invocaciones en transacciones por segundo	Cada región admitida: 500	Sí	Una invocación es una carga programada que se entrega al objetivo definido. Después de alcanzar el límite, las invocaciones se limitan; es decir, siguen produciéndose pero se retrasan.
ListScheduleGroups tasa de solicitudes	Cada región admitida: 10	Sí	Número máximo de ListScheduleGroups solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.

Nombre	Valor predeterminado	Ajusta	Descripción
ListSchedules tasa de solicitudes	Cada región admitida: 50	Sí	Número máximo de ListSchedules solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.
ListTagsForResource tasa de solicitudes	Cada región admitida: 10	Sí	Lista las etiquetas asociadas al recurso del Programador.
Número de grupos de programas	Cada región admitida: 500	Sí	Número máximo de grupos de programación por región.
Número de programas	Cada región admitida: 1 000 000	Sí	El número máximo de horarios por región. Esta cuota incluye los programas únicos que han terminado de ejecutarse. Le recomendamos que elimine los programas únicos una vez que hayan terminado de ejecutarse y hayan invocado un objetivo.
TagResource tasa de solicitud	Cada región admitida: 1	Sí	Asigna una o más etiquetas (pares clave-valor) al recurso del Programador especificado.

Nombre	Valor predeterminado	Ajusta	Descripción
UntagResource tasa de solicitud	Cada región admitida: 1	Sí	Elimina una o más etiquetas del recurso del Programador especificado.
UpdateSchedule tasa de solicitud	Cada región admitida: 50	Sí	Número máximo de UpdateSchedule solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.

Para obtener más información sobre las cuotas y los puntos de enlace de servicio de EventBridge Scheduler, consulte los puntos de enlace y las cuotas de Amazon EventBridge Scheduler en la AWS guía de referencia general.

Historial de documentos para la Guía del usuario del Programador de EventBridge

En la tabla siguiente se detallan las versiones de la documentación del Programador de EventBridge.

Cambio	Descripción	Fecha
Cambios en el rol de ejecución y prevención adjunta confusa	Esta actualización describe los cambios en la forma en que se aplica el rol de ejecución a un recurso de un grupo de programaciones al implementar una política de permisos confusa en la política de permisos del rol. • the section called “Prevención del suplente confuso”	7 de septiembre de 2023
Eliminación automática de programaciones una vez finalizadas	El Programador de EventBridge admite la eliminación automática. Al configurar la eliminación automática, el Programador de EventBridge elimina la programación después de la última invocación planificada. • the section called “Eliminación tras la finalización de la programación”	2 de agosto de 2023
Tema actualizado sobre el uso de destinos universales	Se ha actualizado la lista de servicios compatibles a los que el Programador de EventBridge puede dirigirse y con los que se puede integrar.	17 de marzo de 2023

	Esta actualización también incluye una lista de operaciones de API GET no compatibles e incluye mejoras en los ejemplos de destinos universales, así como otras mejoras menores en toda la guía. • the section called “Uso de destinos universales”	
Información actualizada sobre las programaciones basadas en frecuencias que no tienen fecha de inicio	Se agregó información sobre cómo el Programador de EventBridge gestiona las programaciones basadas en frecuencias si no se especifica una `StartDate`. • the section called “Programaciones basadas en frecuencias”	17 de marzo de 2023
Nuevo tema sobre la gestión de los grupos de planificadores	Se ha añadido un nuevo capítulo sobre cómo crear grupos de planificadores con el Programador de EventBridge. Utilice este capítulo para aprender a crear un grupo, añadir programaciones al grupo, aplicar etiquetas para gestionar y supervisar más fácilmente los recursos del Programador de EventBridge y, por último, eliminar un grupo. • Administrar un grupo de programaciones	17 de marzo de 2023

Nuevos temas sobre el horario de verano y las zonas horarias	Se han añadido nuevas secciones que describen cómo el Programador de EventBridge gestiona el horario de verano y cómo se pueden crear programaciones en diferentes zonas horarias. • the section called “Horario de verano” • the section called “Zonas horarias”	17 de noviembre de 2022
Nuevo tema sobre métricas	Se ha añadido un tema nuevo que describe las métricas que el Programador de EventBridge publica en CloudWatch. Puede utilizar estas métricas para supervisar los errores de invocación y comprender cómo resolver los problemas relacionados con sus programaciones. • the section called “Monitoreo con CloudWatch”	15 de noviembre de 2022
Versión inicial	Guía de usuario del Programador de EventBridge.	10 de noviembre de 2022

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.