



Guía del usuario de

EventBridge Planificador



EventBridge Planificador: Guía del usuario de

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es el Programador de EventBridge?	1
Características principales del Programador de EventBridge	1
Acceder al programador de EventBridge	2
Configuración	3
Inscríbase en AWS	3
Creación un usuario de IAM	3
Usar políticas administradas por	4
Configurar el rol de ejecución	5
Configurar un destino	9
Sigüientes pasos	12
Introducción	13
Requisitos previos	14
Uso de la consola	14
Uso de AWS CLI	18
Uso de los SDK	18
Sigüientes pasos	20
Tipos de programaciones	21
Programaciones basadas en frecuencia	22
Sintaxis	22
Ejemplos	22
Programaciones basadas en cron	23
Sintaxis	23
Ejemplos	24
Programación única	25
Sintaxis	25
Ejemplos	25
Zonas horarias	26
Horario de verano	26
Administrar una programación	28
Cambiar el estado de la programación	29
Configuración de ventanas de tiempo flexibles	30
Configuración de una DLQ	31
Cree una cola de Amazon SQS.	32
Configurar permisos de rol de ejecución	33

Especificar una política de cola de mensajes fallidos	34
Recuperar el evento de mensajes fallidos	35
Eliminar una programación	38
Eliminación tras la finalización de la programación	38
Eliminación manual	40
Sigüientes pasos	40
Administrar un grupo de programaciones	41
Crear un grupo de programaciones	42
Paso uno: Crear un nuevo grupo de programaciones	42
Asociar una programación	44
Eliminar un grupo de programaciones	45
Recursos relacionados	47
Gestión de destinos	48
Uso de destinos basados en plantillas	49
Amazon SQS SendMessage	50
Lambda Invoke	52
Step Functions StartExecution	54
Uso de destinos universales	56
Acciones no admitidas	57
Ejemplos	58
Agregar atributos de contexto	60
Sigüientes pasos	61
AWS PrivateLink	62
Consideraciones	62
Creación de un punto de conexión de interfaz	62
Creación de una política de punto de conexión	63
Seguridad	65
Administración del acceso	65
Público	66
Autenticación con identidades	66
Administración del acceso con políticas	68
Integración con IAM	69
Uso de políticas basadas en identidades	75
Prevención del suplente confuso	87
Resolución de problemas	89
Protección de datos	91

Cifrado en reposo	92
Cifrado en tránsito	99
Validación de conformidad	100
Resiliencia	100
Seguridad de infraestructuras	101
Monitorización y métricas	102
Monitorización con CloudWatch	102
Términos	103
Dimensiones	104
Acceder a las métricas de	104
Lista de métricas	105
Métricas de uso	111
Monitorización con registros de CloudTrail	115
Información del Programador de EventBridge en CloudTrail	115
Descripción de las entradas de archivos de registro de Programador de EventBridge	116
Cuotas	117
Solución de problemas de cuotas	125
ServiceQuotaExceededException	125
Resolución de problemas	127
Errores de destino	127
Causas habituales:	127
Pasos para la solución de problemas	127
Permisos de rol	129
Causas habituales	129
Síntomas	129
Pasos para la solución de problemas	130
Cuotas de servicio	132
Identificación de problemas de cuota	132
Resolución de problemas de cuota	132
Tiempo de activación y patrones	133
Causas habituales	133
Pasos para la solución de problemas	133
Creación de patrones	134
Problemas comunes	134
Pasos para la solución de problemas	135
¿Se está activando mi destino?	135

Destinos con plantilla frente a destinos universales	136
Entrada de destino universal no válida	136
Síntomas	136
Ejemplos	136
¿Cómo resolverlo	137
Programación de actualizaciones que activen invocaciones inesperadas	138
Habilitación o desactivación de las programaciones únicas	138
Historial de revisión	139
.....	cxliii

¿Qué es el Programador de Amazon EventBridge?

Programador de Amazon EventBridge es un programador sin servidor que le permite crear, ejecutar y administrar tareas desde un único 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 de frecuencia para patrones recurrentes, o configurar invocaciones únicas. Puede configurar intervalos de tiempo flexibles para la entrega, definir límites de reintentos y establecer el tiempo máximo de retención para activadores fallidos.

Temas

- [Características principales del Programador de EventBridge](#)
- [Acceder al programador de EventBridge](#)

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 basados en plantillas: el Programador de EventBridge admite destinos basados en 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 de EventBridge, el SDK del Programador de EventBridge o directamente mediante la API AWS CLI del Programador de EventBridge.

Configuración de Amazon EventBridge Scheduler

Antes de poder utilizar EventBridge Scheduler, debe completar los siguientes pasos.

Temas

- [Inscríbese en AWS](#)
- [Creación un usuario de IAM](#)
- [Usar políticas administradas por](#)
- [Configurar el rol de ejecución](#)
- [Configurar un destino](#)
- [Siguiendo pasos](#)

Inscríbese en AWS

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

Para suscribirse a una Cuenta de AWS

1. Abrir <https://portal.aws.amazon.com/billing/registro>.
2. Siga las instrucciones que se le indiquen.

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

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

Creación un usuario de IAM

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

Elegir una forma de administrar el administrador	Para	Haga esto	También puede
En IAM Identity Center (recomendado)	Usar credenciales a corto plazo para acceder a AWS. Esto se ajusta a las prácticas recomendadas de seguridad. Para obtener información sobre las prácticas recomendadas, consulta 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 configurando el AWS CLI que se utilizará AWS IAM Identity Center en la Guía del AWS Command Line Interface usuario.
En IAM (no recomendado)	Usar credenciales a largo plazo para acceder a AWS.	Siguiendo las instrucciones de Crear un usuario de IAM para acceso de emergencia de la Guía del usuario de IAM.	Configure el acceso programático mediante Administrar 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 AWS recursos. En la mayoría de los casos, para usar EventBridge Scheduler de forma segura,

le recomendamos que cree usuarios, grupos o roles independientes con solo los permisos necesarios para usar EventBridge Scheduler. EventBridge Scheduler admite las siguientes políticas administradas para casos de uso comunes.

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

Puede adjuntar estas políticas administradas a sus entidades principales de IAM del mismo modo que adjuntó la política `AdministratorAccess` en el paso anterior. Para obtener más información sobre la gestión del acceso a EventBridge Scheduler mediante políticas de IAM basadas en la identidad, consulte [the section called “Uso de políticas basadas en identidades”](#)

Configurar el rol de ejecución

Una función de ejecución es una función de IAM que EventBridge Scheduler asume para interactuar con otras personas en su nombre. Servicios de AWS Debe adjuntar políticas de permisos a esta función para conceder a EventBridge Scheduler acceso a los objetivos.

También puede crear una nueva función de ejecución cuando utilice la consola para [crear una nueva programación](#). Si utilizas la consola, EventBridge Scheduler crea un rol en tu nombre con permisos en función del destino que elijas. Cuando EventBridge Scheduler crea un rol para ti, la política de confianza del rol incluye [claves de condición](#) que limitan los directores que pueden asumir el rol en tu nombre. Esto protege contra el posible [problema de seguridad del suplente adjunto](#).

En los siguientes pasos se describe cómo crear un nuevo rol de ejecución y cómo conceder a EventBridge Scheduler acceso para invocar un objetivo. En este tema se describen los permisos para los destinos basados en plantillas más populares. Para obtener información sobre cómo añadir permisos para otros destinos, consulte [the section called “Uso de destinos basados en plantillas”](#).

Para crear un rol de ejecución mediante el AWS CLI

1. Copie la siguiente política JSON de asumir rol y guárdela localmente como `Scheduler-Execution-Role.json`. Esta política de confianza permite a EventBridge Scheduler asumir la función en su nombre.

JSON

```
{
  "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. Desde AWS Command Line Interface (AWS CLI), ingresa 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 a EventBridge Scheduler 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 a EventBridge Scheduler realizar la `sqs:SendMessage` acción en todas las colas de Amazon SQS de su cuenta.

JSON

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

Amazon SNS – Publish

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

JSON

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

Lambda – Invoke

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

JSON

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

```
}
```

- 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": "2022-03-01T19:31:18.620Z",  
    "AttachmentCount": 0,  
    "IsAttachable": true,  
    "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",  
    "DefaultVersionId": "v1",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:policy/PolicyName",  
    "UpdateDate": "2022-03-01T19:31:18.620Z"  
  }  
}
```

- 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 EventBridge programación de Scheduler, necesita al menos un objetivo para que la programación la invoque. Puedes usar un AWS recurso existente o crear uno nuevo.

Los siguientes pasos muestran cómo crear una nueva cola estándar de Amazon SQS con CloudFormation

Para crear una nueva cola de Amazon SQS

1. Copie la siguiente CloudFormation plantilla JSON y guárdela localmente como. `Scheduler-Target-SQS.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"
        ]
      }
    }
  }
}
```

```
}  
}
```

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

```
$ 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"  
}
```

- Ejecute el siguiente comando para ver la información resumida de la CloudFormation pila. 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, utilizarás el valor for QueueARN para configurar la cola como destino de EventBridge Scheduler.

Siguientes pasos

Cuando hayas completado el paso de configuración, usa la guía de [introducción](#) para crear tu primer EventBridge programador de Scheduler e invocar un objetivo.

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 basado en 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](#)
- [Crear una programación mediante la consola Programador de EventBridge](#)
- [Cree una programación utilizando la AWS CLI](#)
- [Cree una programación con los SDK del Programador de EventBridge](#)
- [Siguiendo pasos](#)

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 Consola de administración de AWS 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 Consola de administración de AWS.

2. En la página Programaciones, elija Crear programación.
3. En la página Especificar los detalles de la programación, en la sección Nombre y descripción de la programación, proceda del modo 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 añadir 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 desplegadas.

- 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 expresión cron. Para usar una expresión de frecuencia, elija una programación basada en frecuencia e introduzca un número positivo en Valor y, a continuación, elija una de las opciones desplegadas de Unidad.

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 destino una vez cada hora, la programación se ejecuta 15 minutos después del inicio de cada hora.

5. 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 basados en plantillas y elija una API de destino. Para este ejemplo, elegiremos el destino Amazon SQS **SendMessage** basado en plantilla.
- 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), ingrese 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 basado en 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 Programaciones. En la columna 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 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 basado en 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 basado en 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 basados en plantillas 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 programación en EventBridge Scheduler

En el siguiente tema se describen los distintos tipos de horarios que admite Amazon EventBridge Scheduler, así como la forma en que EventBridge Scheduler 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 frecuencia 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 EventBridge Scheduler 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 EventBridge Scheduler evalúa la programación.

Note

Todos los tipos de horarios del EventBridge Scheduler invocan sus objetivos 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 la 1:00:00 y la 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 EventBridge Scheduler.

Temas

- [Programaciones basadas en frecuencia](#)
- [Programaciones basadas en cron](#)
- [Programación única](#)
- [Zonas horarias en EventBridge Scheduler](#)
- [Horario de verano en Scheduler EventBridge](#)

Programaciones basadas en frecuencia

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 programaciones recurrentes más comunes mediante una programación basada en frecuencia. Por ejemplo, si quieres una programación que invoque su objetivo cada 15 minutos, una vez cada dos horas o una vez cada cinco días, puedes usar una programación basada en tarifas para lograrlo. Para configurar una programación basada en frecuencia, utilice una expresión de frecuencia.

Con las programaciones basadas en frecuencia, se utiliza la propiedad [StartDate](#) para establecer la primera ocurrencia 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 quieres que tu programación invoque su objetivo.

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

Ejemplos

El siguiente ejemplo muestra cómo utilizar las expresiones de tasas con el AWS CLI `create-schedule` comando para configurar una programación basada en tasas. 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 `SqsParameters` basado en plantilla.

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 un programa periódico detallado que se ejecuta a la hora específica que elija. EventBridge El programador permite configurar horarios basados en cronogramas 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 planificaciones basadas en cronogramas cuando necesite una programación de periodicidad personalizada que no sea compatible con ninguna de las expresiones tarifarias de EventBridge Scheduler. 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 day-of-month, mes y un campo opcional day-of-week, 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	, - * /
D ay-of-month	1-31	, - * ? / L W
Mes	1-12 o JAN-DEC	, - * /
D ay-of-week	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 puedes usar el asterisco (*) en Day-of-week los campos Day-of-month y. 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 Day-of-month campo puede escribir 7 y si cualquier día de la semana es aceptable, ¿puede escribir? en el Day-of-week campo.
- El comodín L en los campos Day-of-month o Day-of-week especifica el último día del mes o de la semana.
- El W comodín del Day-of-month campo especifica un día de la semana. En el Day-of-month campo, 3W especifica el día de la semana más cercano al tercer día del mes.
- El comodín # del Day-of-week campo especifica una instancia determinada del día de la semana especificado dentro 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 el carácter «#», solo puede definir una expresión en el day-of-week campo. 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 AWS CLI `create-schedule` comando para configurar una programación basada en cron. En este ejemplo, se crea una programación que se ejecuta a las 10:15 h 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` basado en 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 mediante una fecha válida y una marca de tiempo. EventBridge El programador admite la programación en la hora universal coordinada (UTC) o en la zona horaria que especifique al crear la programación.

Note

Una programación única sigue teniendo en cuenta para la cuota de tu cuenta una vez finalizada la ejecución y el cumplimiento de su objetivo. 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 de registro consta de la fecha y la hora en las que desea que EventBridge Scheduler invoque su horario, como se muestra a continuación.

Sintaxis

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

Al configurar una programación de una sola vez, el EventBridge programador ignora la programación StartDate y la EndDate que usted especifique para la programación.

Ejemplos

En el siguiente ejemplo, se muestra cómo utilizar las expresiones at con el AWS CLI create-schedule comando 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 SqsParameters basado en plantilla.

```
$ 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 EventBridge Scheduler

EventBridge El programador permite configurar programaciones únicas y basadas en cronogramas en cualquier zona horaria que especifiques. EventBridge Scheduler utiliza la [base de datos de zonas horarias](#) mantenida por la Autoridad de Números Asignados de Internet (IANA).

Con el AWS CLI, puede configurar la zona horaria en la que desea que EventBridge Scheduler evalúe su horario mediante el parámetro. `--schedule-expression-timezone` Por ejemplo, el siguiente comando crea una programación basada en cron que invoca un destino `SendMessage` de Amazon SQS basado en plantilla en `America/New_York` todos los días a las 8:30 h.

```
$ 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 Scheduler EventBridge

EventBridge El programador ajusta automáticamente tu horario al horario de verano. Cuando la hora se adelanta en primavera, si una expresión cron cae en una fecha y hora inexistentes, se omite la invocación de la programación. Cuando la hora se atrasa en otoño, la programación se ejecuta solo una vez y no repite su invocación. Las siguientes invocaciones se producen con normalidad en la fecha y hora especificadas.

EventBridge El programador ajusta tu horario en función de la zona horaria que especifiques al crear el horario. Si configura una programación en `America/New_York`, la 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 frecuencia que 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, EventBridge Scheduler seguirá evaluando la expresión de la tasa 24 horas después de la última invocación del horario.

Note

Según las normas y reglamentos locales, algunas zonas horarias no aplican el horario de verano. Si crea un horario en una zona horaria que no respeta el horario de verano, EventBridge Scheduler no ajustará su horario. Los ajustes del horario de verano no se aplican a las programaciones en tiempo 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 h en la zona horaria especificada.

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

Administrar una programación en EventBridge Scheduler

Un cronograma es el recurso principal que se crea, configura y administra con Amazon EventBridge Scheduler.

Cada programación tiene una expresión de programación que determina cuándo y con qué frecuencia se ejecuta la programación. EventBridge Scheduler admite tres tipos de programaciones: tarifarias, temporales y únicas. 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 objetivo es una operación de API a la que EventBridge Scheduler llama en tu nombre cada vez que se ejecuta tu programación. EventBridge Scheduler admite dos tipos de objetivos: los objetivos con plantillas que se refieren a operaciones de API comunes en un grupo principal de servicios, y el parámetro de destino universal (UTP), que puedes 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](#).

Puedes configurar la forma en que tu programa gestiona los errores cuando EventBridge Scheduler no puede entregar correctamente un evento a un destino mediante dos mecanismos principales: una política de reintentos y una cola de espera (DLQ). Una política de reintentos determina el número de veces que el EventBridge programador debe volver a intentar un evento fallido y cuánto tiempo debe mantener un evento sin procesar. Un DLQ es un programador de EventBridge colas estándar de Amazon SQS que se utiliza para enviar 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 DLQ”](#).

En esta sección, encontrará ejemplos de cómo gestionar las programaciones del Scheduler mediante la consola, el EventBridge Scheduler y el Scheduler. AWS CLI EventBridge SDKs

Temas

- [Cambiar el estado de la programación en Scheduler EventBridge](#)
- [Configuración de ventanas de tiempo flexibles en Scheduler EventBridge](#)
- [Configuración de la cola de cartas muertas de una programación en Scheduler EventBridge](#)
- [Eliminar un horario en Scheduler EventBridge](#)
- [Siguiendo pasos](#)

Cambiar el estado de la programación en Scheduler EventBridge

Un EventBridge programa de Scheduler 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 lo utilice `UpdateSchedule`, deberá proporcionar todos los parámetros necesarios. EventBridge Scheduler reemplaza su horario con la información que usted proporciona. 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 basado en plantilla.

Example SDK de 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 en Scheduler EventBridge

Cuando configuras tu programación con una ventana de tiempo flexible, EventBridge Scheduler invoca el objetivo dentro de la ventana de tiempo que hayas 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. Los siguientes ejemplos AWS CLI, junto con los ejemplos del SDK de EventBridge Scheduler, se utilizan UpdateSchedule para establecer un intervalo de tiempo flexible de 15 minutos para un horario 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 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 la cola de cartas muertas de una programación en Scheduler EventBridge

Amazon EventBridge Scheduler admite colas de cartas muertas (DLQ) mediante Amazon Simple Queue Service. Cuando un programa no puede invocar su destino, EventBridge Scheduler envía una carga útil 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, EventBridge Scheduler le mostrará el caso sin efecto, agotando el número máximo de reintentos que haya establecido.

En los temas siguientes se describe cómo puede configurar una cola de Amazon SQS como DLQ para su programación, configurar los permisos que EventBridge Scheduler necesita para entregar mensajes a Amazon SQS y recibir eventos de letra muerta del DLQ.

Temas

- [Cree una cola de Amazon SQS.](#)
- [Configurar permisos de rol de ejecución](#)
- [Especificar una política de cola de mensajes fallidos](#)
- [Recuperar el evento de mensajes fallidos](#)

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

EventBridge El programador no admite el uso de una cola FIFO como DLQ de su agenda.

Utilice el siguiente AWS CLI comando 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 un DLQ para su EventBridge programación del Scheduler. 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 EventBridge Scheduler entregue eventos de letra muerta a Amazon SQS.

Configurar permisos de rol de ejecución

Para permitir que EventBridge Scheduler entregue eventos de letra muerta 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](#).

JSON

```
{
  "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 EventBridge Scheduler para invocar un destino de la API de Amazon SQS.

En la siguiente sección, utilizará la consola del EventBridge programador y especificará un DLQ para su programación.

Especificar una política de cola de mensajes fallidos

Para especificar un DLQ, usa la consola del EventBridge Scheduler o la AWS CLI para actualizar un cronograma existente o crear uno nuevo.

Console

Para especificar una DLQ utilizando la consola

1. Inicie sesión en y, a continuación Consola de administración de AWS, seleccione el siguiente enlace para abrir la sección EventBridge Programador de la consola: EventBridge <https://console.aws.amazon.com/scheduler/home>
2. En la consola del EventBridge Scheduler, crea una nueva programación o elige una programación existente de la lista de programaciones para editarla.
3. En la página Configuración, para hacer Dead-letter cola (DLQ), realice una de las siguientes acciones:
 - Seleccione Seleccione una cola de Amazon SQS en mi AWS cuenta como 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 AWS cuentas como DLQ y, a continuación, introduzca el ARN de cola de su DLQ. Si elige una cola en otra AWS cuenta, la consola del EventBridge Scheduler 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 del DLQ de una programación, elige el nombre de la lista y, a continuación, selecciona la pestaña de Dead-letter colas en la página de detalles de la programación.

AWS CLI

Para actualizar un horario existente mediante el 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 un nuevo horario con un DLQ mediante el 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, utilizarás el AWS CLI para recibir un evento con letra muerta del 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": "AQEBkNKTD0MrWgHKPoITRBwrPoK3eCSZicZwVqCY0BZ
+FfTcORFpopJbtCqj36VbBTLHreM8+qM/m5jcwqS1A1GmIJ0/hYmMgn/
+dwIty9izE7HnpvRhhEyHxbeTZ5V05RbeasYaBdNyi9WLcnAHviDh6MebLXXNWoFyYNSxdwJuG0f/
w3htX6r3dXpXvvFNPGoQb8ihY37+u0gtsbuIwhLtUSmE8rbldEEwiUfi3IJ1zEZpUS77n/k1GWrMrnYg0Gx/
```

```

BuaLz0rFi2F738XI/
Hnh45uv3ca60YwS1ojPQ1LtX2URg1haV5884FYlaRvY8jRlpCZabTkYRTZKSXG5KNgYZnHpmsspii6JNkjitYVFKPo0H91w
  "MD5OfBody": "07adc3fc889d6107d8bb8fda42fe0573",
  "Body": "{\"MessageBody\": \"Hello, world!\", \"QueueUrl\": \"https://sqs.us-
west-2.amazonaws.com/123456789012/does-not-exist\"}",
  "Attributes": {
    "SenderId": "AR0A2DZE3W4CTL5ZR7EIN: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 el evento de letra muerta para ayudarlo a identificar y solucionar los posibles motivos por los que se ha producido un error en la invocación de Target.

- **ERROR_CODE**— Contiene el código de error que EventBridge Scheduler recibe de la API de servicio del objetivo. En el ejemplo anterior, el código de error devuelto por Amazon SQS es `AWS.SimpleQueueService.NonExistentQueue`. Si el programa no puede invocar un destino debido a un problema con el EventBridge Scheduler, en su lugar verás el siguiente código de error: `AWS.Scheduler.InternalServerError`
- **ERROR_MESSAGE**— Contiene el mensaje de error que EventBridge Scheduler recibe de la API de servicio del objetivo. 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 EventBridge Scheduler, en su lugar verás 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 EventBridge Scheduler lo envió a la DLQ después de superar el número máximo de reintentos que configuraste para la programación o `MaximumEventAgeInSeconds` si el evento tiene una antigüedad superior a la edad máxima que configuraste 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.

Note

Si utiliza [objetivos universales](#), tenga en cuenta que EventBridge Scheduler no valida el contenido del Input campo en el momento de crear la programación. Una programación con parámetros de entrada no válidos se creará correctamente, pero fallará en cada invocación. El mensaje DLQ contendrá el código de error y el mensaje del servicio de destino, que pueden ayudarle a identificar el parámetro no válido. Para obtener más información, consulte [Configuraciones de entrada de destino universales no válidas](#).

Eliminar un horario en Scheduler EventBridge

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](#)
- [Eliminación manual](#)

Eliminación tras la finalización de la programación

Configure la eliminación automática una vez finalizada la programación si quiere evitar tener que gestionar individualmente los recursos de la programación en EventBridge Scheduler. 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, EventBridge Scheduler elimina la programación después de 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 ella `EndDate`, EventBridge Scheduler 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.

Consola de administración de AWS

1. Abra la consola del EventBridge Scheduler 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 AWS CLI comando [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 basados en 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 EventBridge programador, consulta la referencia de la API del [EventBridge programador](#).

Administrar un grupo de programaciones en el Programador de EventBridge

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. Por último, eliminará el grupo.

Temas

- [Crear un grupo de programaciones en el Programador de EventBridge](#)

- [Eliminar un grupo de programaciones en el Programador de EventBridge](#)
- [Recursos relacionados](#)

Crear un grupo de programaciones en el Programador de EventBridge

Utilice los grupos de programaciones y el etiquetado para organizar las programaciones que compartan 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`.

Consola de administración de AWS

Para crear un nuevo grupo usando la Consola de administración de AWS


1. Inicie sesión en Consola de administración de AWS 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 Añadir 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](#).

Consola de administración de AWS

Para asociar una programación a un grupo mediante la Consola de administración de AWS

1. Inicie sesión en Consola de administración de AWS 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](#) basado en 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" }'  
\br/>--group-name TestGroup  
--flexible-time-window '{ "Mode": "OFF" }'
```

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

Eliminar un grupo de programaciones en el Programador de EventBridge

A continuación, puede obtener información sobre cómo eliminar un grupo de programaciones mediante la Consola de administración de AWS 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.

Consola de administración de AWS

Para eliminar un grupo desde la Consola de administración de AWS

1. Inicie sesión en Consola de administración de AWS 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 objetivos en EventBridge Scheduler

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

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 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 objetivo universal es un conjunto de parámetros personalizables que le permiten invocar un conjunto más amplio de operaciones de API para muchos servicios. AWS Por ejemplo, puede usar el parámetro de destino universal (UTP) del EventBridge programador para crear una nueva cola de Amazon SQS mediante la operación. [CreateQueue](#)

Para configurar destinos universales o basados en plantillas, 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 basado en 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 objetivos con plantillas en EventBridge Scheduler](#)
- [Uso de objetivos universales en EventBridge Scheduler](#)
- [Agregar atributos de contexto del Programador de EventBridge](#)
- [Sigüientes pasos](#)

Uso de objetivos con plantillas en EventBridge Scheduler

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 basado en 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 del rol 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](#)

Important

EventBridge El programador no admite la entrega entre regiones cuando se usa EventBridge PutEvents como destino con plantilla. Puedes usar este destino para enviar eventos a un bus de eventos en un lugar diferente Cuenta de AWS solo si el bus de eventos coincide con la programación Región de AWS . Para enviar eventos a un autobús

de eventos de una región diferente, cree un horario en la región en la que se encuentra el autobús de eventos de destino.

- Amazon Inspector – [StartAssessmentRun](#)
- Kinesis – [PutRecord](#)
 - Parámetros: [KinesisParameters](#)
- Firehose: [PutRecord](#)
- Lambda: [Invoke](#)
- SageMaker IA: [StartPipelineExecution](#)
 - Parámetros: [SageMakerPipelineParameters](#)
- Amazon SNS – [Publish](#)
- Amazon SQS – [SendMessage](#)
 - Parámetros: [SqsParameters](#)
- Step Functions – [StartExecution](#)

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

Amazon SQS **SendMessage**

Example Política de permisos para el rol de ejecución

JSON

```
{
  "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 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

JSON

```

{
  "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 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

JSON

```
{
  "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 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 objetivos universales en EventBridge Scheduler

Un objetivo universal es un conjunto de parámetros personalizables que permiten invocar un conjunto más amplio de operaciones de API para muchos servicios. 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 objetivo universal para su programación mediante el AWS CLI EventBridge programador o uno de ellos SDKs, debe especificar la siguiente información:

- **RoleArn**— El ARN de la función de ejecución que desea utilizar para el objetivo. 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`.

Note

El *service* valor del ARN de destino universal debe coincidir con el identificador del servicio de AWS SDK del servicio de destino. Este identificador puede diferir del prefijo de punto final del servicio. Por ejemplo, para Amazon Cognito Identity Provider, utilice `cognitoidentityprovider` (no `cognito-idp`). Para encontrar el identificador de servicio correcto, consulte la documentación del AWS SDK del servicio al que desee dirigirse.

- Entrada: un JSON bien formado que especificas con los parámetros de solicitud que EventBridge Scheduler 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

EventBridge El programador no admite acciones de API de solo lectura, como las GET operaciones 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
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 `get` prefijo, EventBridge Scheduler no admite este objetivo. 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 puedes configurar de forma diferente APIs, consulta la referencia de la API de ese servicio. Al igual que `LambdaInvoke`, algunas APIs aceptan parámetros de URI, así como una carga útil 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");
    }
}

```

Agregar atributos de contexto del Programador de EventBridge

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 del EventBridge Scheduler, consulte la referencia de la API del Scheduler. EventBridge](#)

Acceso al Programador de Amazon EventBridge mediante un punto de conexión de interfaz (AWS PrivateLink)

Puede usar AWS PrivateLink para crear una conexión privada entre la VPC y el Programador de Amazon EventBridge. Puede acceder al Programador de EventBridge como si estuviera en la VPC, sin el uso de una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión Direct Connect. Las instancias de la VPC no necesitan direcciones IP públicas para acceder al Programador de EventBridge.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado al Programador de EventBridge.

Para obtener más información, consulte [Acceso a los Servicios de AWS a través de AWS PrivateLink](#) en la Guía de AWS PrivateLink.

Consideraciones para el Programador de EventBridge

Antes de configurar un punto de conexión de interfaz para el Programador de EventBridge, consulte [Consideraciones](#) en la Guía de AWS PrivateLink.

El Programador de EventBridge admite la realización de llamadas a todas las acciones de la API a través del punto de conexión de interfaz.

Creación de un punto de conexión de interfaz para el Programador de EventBridge

Puede crear un punto de conexión de interfaz para el Programador de EventBridge mediante la consola de Amazon VPC o la AWS Command Line Interface (AWS CLI). Para obtener más información, consulte [Creación de un punto de conexión de interfaz](#) en la Guía de AWS PrivateLink.

Cree un punto de conexión de interfaz para el Programador de EventBridge mediante el siguiente nombre de servicio:

```
com.amazonaws.region.scheduler
```

Si habilita DNS privado para el punto de conexión de interfaz, puede realizar solicitudes a la API para el Programador de EventBridge mediante su nombre de DNS predeterminado regional. Por ejemplo, `scheduler.us-east-1.amazonaws.com`.

Creación de una política de puntos de conexión para el punto de conexión de interfaz

Una política de punto de conexión es un recurso de IAM que puede adjuntar al punto de conexión de su interfaz. La política de puntos de conexión predeterminada permite acceso completo al Programador de EventBridge a través del punto de conexión de interfaz. Para controlar el acceso permitido al Programador de EventBridge desde la VPC, adjunte una política de puntos de conexión personalizada al punto de conexión de interfaz.

Una política de punto de conexión especifica la siguiente información:

- Las entidades principales que pueden llevar a cabo acciones (Cuentas de AWS, usuarios de IAM y roles de IAM).
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.

Para obtener más información, consulte [Control del acceso a los servicios con políticas de punto de conexión](#) en la Guía del usuario de AWS PrivateLink.

Ejemplo: Política de punto de conexión de VPC para acciones del Programador de EventBridge

El siguiente es un ejemplo de una política de un punto de conexión personalizado. Cuando se asocia con un punto de conexión de interfaz, se concede acceso a las acciones del Programador de EventBridge mostradas para todas las entidades principales en todos los recursos.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
```

```
        "scheduler:GetSchedule",
        "scheduler:ListSchedules",
        "scheduler:GetScheduleGroup",
        "scheduler:ListScheduleGroups"
    ],
    "Resource": "*"
}
]
```

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 se refiere a estos conceptos 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. Third-party los auditores comprueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los programas de [AWS cumplimiento, los programas](#) 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. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa 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](#)
- [Protección de datos en Amazon EventBridge Scheduler](#)
- [Validación de conformidad para Amazon EventBridge Scheduler](#)
- [Resiliencia en Amazon EventBridge Scheduler](#)
- [Seguridad de infraestructura en Amazon EventBridge Scheduler](#)

Administrar el acceso a Amazon EventBridge Scheduler

AWS Identity and Access Management (IAM) es un sistema Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los AWS recursos. 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](#)
- [Autenticación con identidades](#)
- [Administración del acceso con políticas](#)
- [Cómo funciona EventBridge Scheduler con IAM](#)
- [Uso de políticas basadas en la identidad en Scheduler EventBridge](#)
- [Prevención de errores confusa en Scheduler EventBridge](#)
- [Solución de problemas de identidad y acceso a Amazon EventBridge Scheduler](#)

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte [Solución de problemas de identidad y acceso a Amazon EventBridge Scheduler](#)).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte [Cómo funciona EventBridge Scheduler con IAM](#)).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte [Uso de políticas basadas en la identidad en Scheduler EventBridge](#)).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte [Cómo iniciar sesión en la Cuenta de AWS](#) en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte [AWS Signature Version 4 para solicitudes de API](#) en la Guía del usuario de IAM.

Cuenta de AWS usuario raíz

Al crear una Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz, que tiene acceso completo a todos los Servicios de AWS recursos. Se recomienda encarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de 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 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 empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, se recomienda AWS IAM Identity Center. Para obtener más información, consulte [¿Qué es el Centro de identidades de IAM?](#) en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte [Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte [Casos de uso para usuarios de IAM](#) en la Guía del usuario de IAM.

Roles de IAM

Un [Rol de IAM](#) es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol [cambiando de un rol de usuario a uno de IAM \(consola\)](#) o llamando a una AWS

CLI operación de AWS API. Para obtener más información, consulte [Métodos para asumir un rol](#) en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte [Información general de políticas de JSON](#) en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Identity-based políticas

Identity-based las políticas son documentos de política de permisos de JSON que se adjuntan a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) en la Guía del usuario de IAM.

Identity-based las políticas pueden ser políticas integradas (integradas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte [Selección entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

Resource-based políticas

Resource-based las políticas son documentos de políticas de JSON que se adjuntan a un recurso. Los ejemplos incluyen 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. Debe [especificar una entidad principal](#) en una política basada en recursos.

Resource-based las políticas son políticas en línea que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- Políticas de control de servicios (SCP): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte [Políticas de control de servicios](#) en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCP): definen los permisos máximos disponibles para los recursos de las cuentas. Para obtener más información, consulte [Políticas de control de recursos \(RCP\)](#) en la Guía del usuario de AWS Organizations .
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener 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 se AWS determina 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
Identity-based políticas	Sí
Resource-based políticas	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
Service-linked roles	No

Para obtener una visión general de cómo funcionan el EventBridge programador 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](#).

Identity-based políticas de Scheduler EventBridge

Compatibilidad con las políticas basadas en identidad: sí

Identity-based las políticas son documentos de política de permisos de JSON que puedes adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. 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 la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) 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. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte [Referencia de los elementos de la política de JSON de IAM](#) en la Guía del usuario de IAM.

Identity-based ejemplos de políticas para Scheduler EventBridge

Para ver ejemplos de políticas de EventBridge Scheduler basadas en la identidad, consulte. [Uso de políticas basadas en la identidad en Scheduler EventBridge](#)

Resource-based políticas dentro de Scheduler EventBridge

Admite políticas basadas en recursos: no

Resource-based las políticas son documentos de políticas de JSON que se adjuntan a un recurso. Los 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 basada en 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. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Acciones políticas para Scheduler EventBridge

Compatibilidad con las 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. 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

Compatibilidad con los 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. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). En el caso de las acciones que no admiten permisos por recurso, 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 la identidad en Scheduler EventBridge](#)

Claves de condición de política para Scheduler EventBridge

Compatibilidad con 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` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud. 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 la identidad en Scheduler EventBridge](#)

ACL en Scheduler EventBridge

Compatibilidad con 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

Compatibilidad con ABAC (etiquetas en las políticas): parcial

Attribute-based el control de acceso (ABAC) es una estrategia de autorización que define los permisos en función de unos atributos denominados etiquetas. Puede adjuntar etiquetas a las entidades y AWS los recursos de IAM y, a continuación, diseñar políticas de ABAC para permitir las operaciones cuando la etiqueta del director coincida con la etiqueta del recurso.

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 [Definición de permisos con la autorización de 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

Compatibilidad con credenciales temporales: sí

Las credenciales temporales proporcionan acceso a AWS los recursos a corto plazo y se crean automáticamente cuando se utiliza la federación o se cambia de rol. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte [Credenciales de seguridad temporales en IAM](#) y [Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

Cross-service permisos principales para EventBridge Scheduler

Admite sesiones de acceso directo (FAS): sí

Las sesiones de acceso directo (FAS) utilizan los permisos del operador principal que realiza la llamada Servicio de AWS, junto con los de solicitud, Servicio de AWS para realizar solicitudes a los servicios descendentes. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Sesiones de acceso directo](#).

Funciones de servicio del programador EventBridge

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 obtener más información, consulte [Crear 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.

Service-linked funciones para Scheduler EventBridge

Compatibilidad 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 la función de realizar una acción en su nombre. Service-linked las funciones aparecen en su nombre 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, consulta [Servicios de AWS que funcionan con IAM](#). Busque un servicio en la tabla que incluya una Yes en la columna de Service-linked funciones. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Uso de políticas basadas en la identidad en Scheduler EventBridge

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar los recursos del Scheduler. EventBridge Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan.

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 \(consola\)](#) 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

Identity-based las políticas determinan si alguien puede crear, acceder o eliminar los recursos de EventBridge Scheduler de su cuenta. Estas acciones pueden generar costos 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 tarea](#) en la Guía de 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 CloudFormation. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.
- Utiliza 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 [Validación de políticas con el Analizador de acceso de IAM](#) en la Guía del 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 exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte [Acceso seguro a la API con MFA](#) en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte [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

- `AmazonEventBridgeSchedulerFullAccess`

Otorga permisos para usar todas las acciones del EventBridge programador para las programaciones y los grupos de programaciones.

Para ver los permisos de esta política, consulte [AmazonEventBridgeSchedulerFullAccess](#) en la Referencia de la política administrada de AWS .

- `AmazonEventBridgeSchedulerReadOnlyAccess`

Concede permisos de solo lectura para ver los detalles de las programaciones y grupos de programaciones.

Para ver los permisos de esta política, consulte [AmazonEventBridgeSchedulerReadOnlyAccess](#) en la Referencia de la política administrada de AWS .

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 una nueva programación, puede elegir si desea cifrar sus datos en EventBridge Scheduler mediante una clave gestionada por el cliente o mediante una clave [Clave propiedad de AWS](#) gestionada 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

JSON

```
{
  "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.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "scheduler:CreateSchedule"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:scheduler:us-east-1:123456789012: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"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": "scheduler.us-east-1.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.

JSON

```

{
    "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:

JSON

```

{
  "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:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "scheduler:UpdateSchedule"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:scheduler:us-east-1:123456789012: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"
    ],
    "Condition": {
        "StringLike": {
            "kms:ViaService": "scheduler.us-east-1.amazonaws.com",
            "kms:EncryptionContext:aws:scheduler:schedule:arn":
"arn:aws:scheduler:us-east-1:123456789012: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"
```

JSON

```
{
  "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
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 los programas y los grupos de programas.	10 de noviembre de 2022
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
AmazonEventBridgeSchedulerReadOnlyAccess — Política gestionada actualizada	EventBridge Scheduler actualizó la política AmazonEventBridgeSchedulerReadOnlyAccess administrada para usar acciones comodín (scheduler:List* y scheduler:Get*) para garantizar que la política se mantenga actualizada a medida que se agreguen nuevas acciones de solo lectura.	25 de marzo de 2026

Cambio	Descripción	Fecha
EventBridge El programador comenzó a rastrear los cambios	EventBridge Scheduler comenzó a rastrear los cambios de sus políticas AWS gestionadas.	10 de noviembre de 2022

Prevención de errores confusa en Scheduler EventBridge

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. Cross-service la suplantación de identidad puede producirse cuando un servicio (el servicio de 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](#) las claves de contexto en su función de ejecución de la programación para limitar los permisos que EventBridge Scheduler concede a otro servicio para acceder al recurso. Utiliza `aws:SourceArn` si desea que solo se asocie un recurso al acceso entre servicios. Utiliza `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:

JSON

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

Si utiliza caracteres comodín en el valor de `aws:SourceArn`, debe utilizarlos en `ArnLike` lugar de `StringEquals` como operador de condición. Por ejemplo, la siguiente política de confianza se utiliza `ArnLike` para hacer coincidir cualquier grupo de programación de la cuenta:

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

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "scheduler.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "arn:aws:scheduler:*:123456789012:schedule-group/*"
    },
    "StringEquals": {
      "aws:SourceAccount": "123456789012"
    }
  }
}
```

Solución de problemas de identidad y acceso a Amazon EventBridge Scheduler

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas comunes 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 mi Cuenta de AWS para acceder 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.

Algunos 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 la función 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 EventBridge Scheduler. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir la función 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 AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi Cuenta de AWS para acceder a mis recursos de EventBridge Scheduler

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se 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 obtener 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 \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Protección de datos en Amazon EventBridge Scheduler

El [modelo de](#) se aplica a la 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. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres 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 privacidad de datos](#) y los . Para obtener más información sobre la protección de datos en Europa, consulte el [Centro del Reglamento General de Protección de Datos \(RGPD\)](#).

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure 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:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.

- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte [Cómo trabajar con CloudTrail senderos](#) en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados que contienen Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 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 \(FIPS\) 140-3](#).

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 introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Cifrado inactivo en Scheduler EventBridge

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 AWS son 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 y 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:	Description (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	Soportado	compatible

Tipo de datos:	Description (Descripción)	Clave propiedad de AWS	clave administrada por el cliente
	que se entregue al destino.		
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 la expresión de frecuencia o cron para las programaciones recurrentes y la marca de tiempo para las invocaciones únicas, así como la fecha de inicio, la fecha de finalización y la zona horaria de la programación.	compatible	No compatible
Configuración de destino	El nombre de recurso de Amazon (ARN) y otros detalles de configuración relacionados con el destino.	compatible	No compatible

Tipo de datos:	Description (Descripción)	Clave propiedad de AWS	clave administrada por el cliente
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.

Identity-based permisos para las 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 una programación.

- **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.

Estas acciones se suman a las siguientes:

- **scheduler:***

- **iam:PassRole**: necesario para pasar el rol de ejecución.

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 permitir que EventBridge Scheduler pueda llamar a la AWS KMS API al descifrar los datos.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEventBridgeSchedulerToDecryptDataUsingCMKMS",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-east-1:123456789012: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.

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 qué recursos del EventBridge Scheduler pueden usar tu clave.

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. En el siguiente ejemplo, se 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": "ABCDEFGH11JKLMN0P2Q3",
    "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 en Scheduler EventBridge

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 un programa de cumplimiento 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](#) 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. Para obtener más información sobre su responsabilidad de conformidad al utilizarlos Servicios de AWS, consulte [AWS la documentación de seguridad](#).

Resiliencia en Amazon EventBridge Scheduler

La infraestructura AWS global se basa en zonas Regiones de AWS 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.
- Paquetes de cifrado con perfecto secreto directo (PFS), como el DHE (Ephemeral) o el ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

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 supervisa sus recursos de AWS y las aplicaciones que ejecute 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 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 puede identificar qué usuarios y cuentas llamaron a AWS, la dirección IP de origen de las llamadas y el momento en que se hicieron. Para más información, consulte la [Guía del usuario de AWS CloudTrail](#).

Temas

- [Supervisión de Amazon EventBridge Scheduler con Amazon CloudWatch](#)
- [Registro de llamadas a la API del Programador de Amazon EventBridge mediante AWS CloudTrail](#)

Supervisión de Amazon EventBridge Scheduler con Amazon CloudWatch

Puede supervisar Amazon EventBridge Scheduler con Amazon Scheduler CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. EventBridge Scheduler emite un conjunto de métricas para todos los horarios y un conjunto adicional de métricas para los horarios que tienen una cola de espera (DLQ) asociada. Si [configuras un DLQ](#) para tu programación, EventBridge Scheduler publicará métricas adicionales cuando tu programación agote 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, consulta la [Guía del CloudWatch usuario de Amazon](#).

Best-effort CloudWatch métricas, entrega

CloudWatch las métricas se entregan haciendo el mejor esfuerzo posible. La mayoría de las operaciones del EventBridge Scheduler dan como resultado el envío de un punto de datos a CloudWatch. La integridad y la puntualidad de las métricas no están garantizadas. Es posible que un punto de datos de un minuto determinado se retrase antes de CloudWatch estar disponible o que no se entregue en absoluto. CloudWatch Las métricas te dan una idea de la naturaleza de la actividad prácticamente en tiempo real. No pretenden ser una contabilidad completa de todas las operaciones.

Temas

- [Términos](#)
- [Dimensiones](#)
- [Acceder a las métricas de](#)
- [Lista de métricas](#)
- [EventBridge Métricas de uso del programador](#)

Términos

Namespace

Un espacio de nombres es un contenedor de las CloudWatch métricas de un AWS servicio. Para EventBridge Scheduler, el espacio de nombres es `AWS/Scheduler`

CloudWatch métricas

Una CloudWatch métrica representa un conjunto de puntos de datos ordenados en el tiempo que son específicos de CloudWatch.

Dimensión

Una dimensión es un name/value par que forma parte de la identidad de una métrica.

Unidad

Las estadísticas tienen unidades de medida. En el EventBridge caso del planificador, las unidades incluyen el recuento.

Dimensiones

En esta sección se describen las CloudWatch dimensiones en las que se agrupan las métricas del EventBridge Scheduler. CloudWatch

Dimensión	Description (Descripción)
ScheduleGroup	El grupo de planificaciones que se utilizan para ver las métricas. CloudWatch Si aún no ha creado ningún grupo, EventBridge Scheduler asocia sus horarios al default grupo.

Acceder a las métricas de

En esta sección se describe cómo acceder a las métricas de rendimiento de un EventBridge programa específico del Scheduler. CloudWatch

Pasos para consultar las métricas de rendimiento de una dimensión

1. Abre la [página de métricas](#) en la CloudWatch consola.
2. Usa el selector de AWS regiones para elegir la región para tu 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 CloudWatch métrica para una dimensión. Por ejemplo, InvocationAttemptCounto InvocationDroppedCount, a continuación, elija Búsqueda gráfica.
6. Seleccione la pestaña Métricas graficadas para ver las estadísticas de rendimiento de las métricas del EventBridge Scheduler.

Lista de métricas

En las siguientes tablas se enumeran las métricas de todos los programas del EventBridge Scheduler, así como las métricas adicionales de los horarios para los que ha configurado un DLQ.

Métricas de todas las programaciones

Namespace	Métrica	Unidad	Description (Descripción)
AWS/Scheduler	InvocationAttemptCount	Recuento	Se emite por cada intento de invocación. Usa esta métrica para comprobar si EventBridge Scheduler está intentando invocar tus horarios y para ver cuándo las invocaciones se acercan a las cuotas de tu cuenta.
AWS/Scheduler	TargetErrorCount	Recuento	Se emite cuando el objetivo devuelve una excepción después de que EventBridge Scheduler llame a la API de destino. Utilízela 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

Namespace	Métrica	Unidad	Description (Descripción)
			destino. Úselo para diagnosticar errores de entrega cuando el motivo subyacente sea la limitación de las llamadas realizadas por Scheduler a la API de destino EventBridge
AWS/Scheduler	InvocationThrottleCount	Recuento	Se emite cuando EventBridge Scheduler limita una invocación de destino porque supera las cuotas de servicio establecidas por Scheduler. EventBridge Úselo para determinar cuándo ha superado la cuota del límite de las invocaciones. Para obtener más información acerca de las cuotas de servicio, consulte Cuotas .

Namespace	Métrica	Unidad	Description (Descripción)
AWS/Scheduler	InvocationDroppedCount	Recuento	Se emite cuando EventBridge Scheduler deja de intentar invocar el objetivo una vez agotada la política de reintentos de un programa. Para obtener más información sobre las políticas de reintentos, consulta la referencia de la EventBridge API del RetryPolicy programador.

Métricas de las programaciones con una DLQ

Namespace	Métrica	Unidad	Description (Descripción)
AWS/Scheduler	InvocationsSentToDeadLetterCount	Recuento	Se emiten por cada entrega exitosa a la DLQ de una programación. Utilice esta opción para determinar cuándo se envían los

Namespace	Métrica	Unidad	Description (Descripción)
			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.

Namespace	Métrica	Unidad	Description (Descripción)
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount	Recuento	Se emite cuando el EventBridge programado no puede enviar un evento al DLQ. Utilice estas dos métricas para determinar el motivo por el que EventBridge Scheduler no puede enviar un evento al DLQ y modifique la configuración del DLQ para resolver el problema.
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount_<error_code>	Recuento	<p>El siguiente es un ejemplo de la métrica <code>InvocationsFailedToBeSentToDeadLetterCount_<error_code></code></p>

Namespace	Métrica	Unidad	Description (Descripción)
			> cuando la cola de Amazon SQS que especificó como DLQ no existe: <code>InvocationsFailedToBeSentToDeadLetterQueueCount_ AWS.SimpleQueueService.NonExistentQueue</code>

Namespace	Métrica	Unidad	Description (Descripción)
AWS/Scheduler	InvocationsSentToDeadLetterCount_Truncated_MessageSize Exceeded	Recuento	Se emite cuando la carga útil del evento enviado al DLQ supera el tamaño máximo permitido por Amazon SQS EventBridge y Scheduler trunca la carga útil que especifique en el atributo de un programa. Input

EventBridge Métricas de uso del programador

CloudWatch recopila métricas que rastrean el uso de algunos AWS recursos. Estas métricas corresponden a las cuotas AWS de servicio. El seguimiento de estas métricas puede ayudarlo a administrar sus cuotas de forma proactiva. Para obtener más información acerca de las cuotas de servicio, consulte [Cuotas](#).

Estas métricas están contenidas en el espacio de nombres de `AWS/Usage` y no se `AWS/Scheduler` recopilan cada minuto. CloudWatch publica las métricas en este espacio de nombres con las dimensiones `Service`, `Class`, `Type` y `Resource`.

Dimensiones comunes de las métricas de uso

Dimensión	Description (Descripción)
Service	El nombre del AWS servicio que contiene el recurso. En el Programador de EventBridge caso de las métricas de uso, el valor es <code>Scheduler</code> .
Class	La clase de recurso a la que se realiza el seguimiento. En el Programador de EventBridge caso de las métricas de uso, el valor es <code>None</code> .
Type	El tipo de recurso al que se realiza el seguimiento. En el caso de las métricas de uso de la API, el valor es <code>API</code> . En el caso de las métricas de recuento de recursos, el valor es <code>Resource</code> .
Resource	El recurso específico que se está rastreando. En el caso de las métricas de uso de la API, este es el nombre de la operación de la API. En el caso de las métricas de recuento de recursos, este es el tipo de recurso que se está contando.

Métricas de uso de las API

Las métricas de uso de la API registran el número de operaciones de la API realizadas en tu cuenta. Usa estas métricas para monitorear el volumen de llamadas a la API y administrar tus cuotas de tarifas de API.

El nombre de la métrica es `CallCount`. La estadística más útil para esta métrica es `SUM` la que representa el recuento total de operaciones durante el período de 1 minuto.

Métricas de uso de las API

Métrica	Recurso	Description (Descripción)
<code>CallCount</code>	<code>CreateSchedule</code>	El número de operaciones de <code>CreateSchedule</code> API realizadas en tu cuenta.
<code>CallCount</code>	<code>CreateScheduleGroup</code>	El número de operaciones de <code>CreateScheduleGroup</code> API realizadas en tu cuenta.

Métrica	Recurso	Description (Descripción)
CallCount	DeleteSchedule	El número de operaciones de DeleteSchedule API realizadas en tu cuenta.
CallCount	DeleteScheduleGroup	El número de operaciones de DeleteScheduleGroup API realizadas en tu cuenta.
CallCount	GetSchedule	El número de operaciones de GetSchedule API realizadas en tu cuenta.
CallCount	GetScheduleGroup	El número de operaciones de GetScheduleGroup API realizadas en tu cuenta.
CallCount	ListScheduleGroups	El número de operaciones de ListScheduleGroups API realizadas en tu cuenta.
CallCount	ListSchedules	El número de operaciones de ListSchedules API realizadas en tu cuenta.
CallCount	ListTagsForResource	El número de operaciones de ListTagsForResource API realizadas en tu cuenta.
CallCount	TagResource	El número de operaciones de TagResource API realizadas en tu cuenta.
CallCount	UntagResource	El número de operaciones de UntagResource API realizadas en tu cuenta.
CallCount	UpdateSchedule	El número de operaciones de UpdateSchedule API realizadas en tu cuenta.

Por ejemplo, la CallCount métrica con las siguientes dimensiones indica el número de veces que se ha llamado a la operación de la CreateSchedule API en tu cuenta:

- "Servicio": "Programador"
- «Clase»: «Ninguna»
- "Tipo": "API"

- «Recurso»: "CreateSchedule»

Métricas de recuento de recursos

Las métricas de recuento de recursos registran la cantidad aproximada de recursos de tu cuenta. Usa estas métricas para monitorear cuándo te estás acercando a los límites de tu cuota de servicio, lo que te permitirá solicitar aumentos de cuota antes de que se agote la capacidad.

El nombre de la métrica es `ResourceCount`. La estadística más útil para esta métrica es `Maximum`.

Métricas de recuento de recursos

Métrica	Recurso	Description (Descripción)
<code>ResourceCount</code>	<code>ApproximateSchedule</code>	<p>Número aproximado de horarios en tu cuenta. Usa esta métrica para monitorear cuándo te acercas al límite de tu cuota de horarios y así evitar <code>ServiceQuotaExceededException</code> errores al llamar <code>CreateSchedule</code> .</p> <p>Si el número de programaciones es inferior a 1 millón, esta métrica puede mostrar cero. Para que una alarma te avise cuando te acerques a tu cuota de horarios, te recomendamos que <code>Maximum</code> utilices la estadística con un umbral de 5 millones o superior, ya que la cuota de horarios predeterminada es de 10 millones.</p>
<code>ResourceCount</code>	<code>ApproximateScheduleGroup</code>	<p>Número aproximado de grupos de horarios en tu cuenta. Usa esta métrica para controlar cuándo te acercas al límite de cuota de tus grupos de horarios y así evitar <code>ServiceQuotaExceededException</code> errores al llamar <code>CreateScheduleGroup</code> .</p>

Por ejemplo, la `ResourceCount` métrica con las siguientes dimensiones y la `Maximum` estadística indica el número aproximado de horarios de tu cuenta:

- "Servicio": "Programador"
- «Clase»: «Ninguna»
- «Tipo»: «Recurso»
- «Recurso»: "ApproximateSchedule»

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 se crea la cuenta. 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 eventos recientes en su Cuenta de AWS. Para más información, consulte [Visualización de 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 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 de la API 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 lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario de 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 más información, consulte [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 de registro de CloudTrail pueden contener una o varias entradas de registro. Un evento representa una solicitud específica realizada desde un origen cualquiera 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 Region-specific. Puede solicitar el aumento de la mayoría de las cuotas, pero algunas 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 a las que más se hace referencia en EventBridge Scheduler son el número de programas (por defecto, 10 000 000 por región, ajustable en miles de millones), la tasa de CreateSchedule solicitudes (por defecto, 5000 TPS en las regiones principales, ajustable a decenas de miles) y el límite límite de invocaciones (por defecto, 1000 TPS en las regiones principales, ajustable a decenas de miles). Los tres son ajustables; consulte [Solicitar un aumento de cuota](#) en la Guía del usuario de Service Quotas para obtener más información.

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

Name	Predeterminado	Ajuste	Description (Descripción)
CreateSchedule tasa de solicitud	us-east-1: 5000	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. Esto se puede ajustar a decenas de miles de solicitudes por segundo.
	us-east-2: 5000		
	us-west-2: 5000		
	ap-northeast-1: 5000		
	ap-south-1:5.000		

Name	Predeterminado	Ajuste	Description (Descripción)
	ap-southeast-1:5.000 ap-southeast-2:5.000 eu-central-1:5.000 eu-west-1: 5000 eu-west-2:5.000 sa-east-1:5.000 Cada una de las demás regiones compatibles: 250		
CreateScheduleGroup tasa de solicitudes	Cada región admitida: 10	Sí	Número máximo de CreateScheduleGroup solicitudes por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo.

Name	Predeterminado	Ajuste	Description (Descripción)
DeleteSchedule tasa de solicitudes	us-east-1: 1000 us-east-2: 1000 us-west-2: 1000 ap-northeast-1: 1000 ap-south-1: 1000 ap-southeast-1: 1000 ap-southeast-2: 1000 eu-central-1: 1000 eu-west-1: 1000 eu-west-2: 1000 sa-east-1: 1000 Cada una de las demás regiones compatibles: 250	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. Esto se puede ajustar a decenas de miles de solicitudes por segundo.
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.

Name	Predeterminado	Ajuste	Description (Descripción)
GetSchedule tasa de solicitudes	us-east-1: 1000 us-east-2: 1000 us-west-2: 1000 ap-northeast-1: 1000 ap-south-1: 1000 ap-southeast-1: 1000 ap-southeast-2: 1000 eu-central-1: 1000 eu-west-1: 1000 eu-west-2: 1000 sa-east-1: 1000 Cada una de las demás regiones compatibles: 250	Sí	Número máximo de GetSchedule solicitud es por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo. Esto se puede ajustar a decenas de miles de solicitudes por segundo.
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.

Name	Predeterminado	Ajuste	Description (Descripción)
Limitación de invocaciones en transacciones por segundo	us-east-1: 1000 us-east-2: 1000 us-west-2: 1000 ap-northeast-1: 1000 ap-south-1: 1000 ap-southeast-1: 1000 ap-southeast-2: 1000 eu-central-1: 1000 eu-west-1: 1000 eu-west-2: 1000 sa-east-1: 1000 Cada una de las demás regiones compatibles: 500	Sí	Una invocación es una carga programada que se entrega al objetivo definido. Una vez alcanzado el límite, las invocaciones se limitan; es decir, siguen produciéndose pero se retrasan. Se puede ajustar a decenas de miles de transacciones por segundo.
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.

Name	Predeterminado	Ajuste	Description (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: 10 000 000	Sí	El número máximo de programaciones por región. Esta cuota incluye los programas únicos que han terminado de ejecutarse. Recomendamos configurar sus horarios para que se eliminen automáticamente una vez finalizados mediante ActionAfterCompletion esta función. Se puede ajustar a miles de millones de programaciones.

Name	Predeterminado	Ajuste	Description (Descripción)
TagResource tasa de solicitud	Cada región admitida: 1	Sí	Asigna una o más etiquetas (pares clave-valor) al recurso del Programador especificado.
UntagResource tasa de solicitud	Cada región admitida: 1	Sí	Elimina una o más etiquetas del recurso del Programador especificado.

Name	Predeterminado	Ajuste	Description (Descripción)
UpdateSchedule tasa de solicitud	us-east-1: 1000 us-east-2: 1000 us-west-2: 1000 ap-northeast-1: 1000 ap-south-1: 1000 ap-southeast-1: 1000 ap-southeast-2: 1000 eu-central-1: 1000 eu-west-1: 1000 eu-west-2: 1000 sa-east-1: 1000 Cada una de las demás regiones compatibles: 250	<u>Sí</u>	Número máximo de UpdateSchedule solicitud es por segundo. Al alcanzar esta cuota, el EventBridge programador rechaza las solicitudes de esta operación durante el resto del intervalo. Esto se puede ajustar a decenas de miles de solicitudes por segundo.

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.

Solución de problemas de cuotas en el Programador de EventBridge

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que puedan surgir relativos a las cuotas del Programador de EventBridge.

ServiceQuotaExceededException

Recibo errores de limitación en la frecuencia de las solicitudes `CreateSchedule`, `DeleteSchedule`, `GetSchedule` o `UpdateSchedule`, aunque estoy por debajo del límite de frecuencia predeterminado.

Causa habitual

El 7 de septiembre de 2023, el Programador de EventBridge comenzó a admitir el ARN (nombre de recurso de Amazon) `ScheduleGroup` en lugar del ARN `Schedule` en las políticas de confianza de roles de ejecución. Los clientes que figuran en la lista de permitidos para seguir utilizando los ARN `Schedule` en su política de confianza pueden tener límites de 50 TPS, en lugar de los límites predeterminados de 250 a 1000 TPS (según la región).

Resolución

Póngase en contacto con el [servicio de asistencia](#) para solicitar un límite máximo superior.

Prevención

Modifique las políticas de confianza existentes de una de las siguientes formas:

- Eliminar todo el alcance del rol.
- Definir el alcance del rol para que pueda asumirse mediante el ARN `Schedule` o el ARN `ScheduleGroup`.

Por ejemplo, suponga que tiene la siguiente política de confianza:

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "scheduler.amazonaws.com"
  },
}
```

```
"Action": "sts:AssumeRole",
"Condition": {
  "StringEquals": {
    "aws:SourceArn":
"arn:aws:scheduler:region:account:schedule/schedule_group/schedule"
  }
}
```

Podría actualizar la política de confianza a lo siguiente:

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "scheduler.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:SourceArn": [
        "arn:aws:scheduler:region:account:schedule/schedule_group/schedule",
        "arn:aws:scheduler:region:account:schedule-group/schedule_group"
      ]
    }
  }
}
```

Solución de problemas de Amazon EventBridge Scheduler

Puede utilizar los temas de esta sección para solucionar problemas comunes de Amazon EventBridge Scheduler.

Temas

- [Mi programación produce un error con errores de destino](#)
- [Problemas de permisos de roles de ejecución de programación](#)
- [Descripción y administración de Service Quotas](#)
- [Problemas de tiempo de activación y patrones de programación](#)
- [Creación de patrones de programación y expresiones cron](#)
- [¿Se está activando mi destino?](#)
- [Destinos con plantilla frente a destinos universales](#)
- [Configuraciones de entrada de destino universales no válidas](#)
- [Programación de actualizaciones que activen invocaciones inesperadas](#)
- [Habilitación o desactivación de las programaciones únicas](#)

Mi programación produce un error con errores de destino

Los errores de invocación de Target son uno de los problemas más comunes de Scheduler. EventBridge Estos errores se pueden producir por varias razones:

Causas habituales:

- Parámetros de destino faltantes o incorrectos.
- Problemas de conectividad de red.
- Limitación de las API.
- Configuración de destino incorrecta.

Pasos para la solución de problemas

1. Configurar una Dead-Letter cola (DLQ)
 - Una DLQ le ayuda a capturar y analizar las invocaciones erróneas.

- Las invocaciones erróneas se envían a la DLQ con mensajes de error detallados.
- Para [configurar una DLQ](#), agréguela a la configuración de la programación:

```
{
  "DeadLetterConfig": {
    "Arn": "arn:aws:sqs:region:account-id:MyDLQ"
  }
}
```

Nota: Si su DLQ está encriptado con una clave KMS, asegúrese de que la política de claves permita a EventBridge Scheduler usarlo:

```
{
  "Sid": "Allow EventBridge Scheduler to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "scheduler.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

2. Verificación de los parámetros de la API

- Asegúrese de que todos los parámetros necesarios para las llamadas a la API de destino estén presentes y tengan el formato correcto.
- Compruebe que los valores de los parámetros estén dentro de los rangos permitidos.
- Compruebe que se pueda acceder al punto de conexión de la API desde la VPC si se utilizan puntos de conexión de VPC.

3. Revisión de la configuración de red

- Si las llamadas producen un error debido a problemas transitorios de la red, implemente una lógica de [reintento](#).
- Política de reintentos de ejemplo:

```
{
  "RetryPolicy": {
    "MaximumRetryAttempts": 3,
    "MaximumEventAgeInSeconds": 3600
  }
}
```

4. Comprobación de las configuraciones específicas del destino

- Para los destinos basados en plantillas (como las tareas de ECS), asegúrese de proporcionar invalidaciones mediante el parámetro `Target . Input` de la API de creación de programaciones.
- Compruebe que el servicio de destino sea [compatible](#) y esté configurado correctamente.

Problemas de permisos de roles de ejecución de programación

Los problemas con los permisos de rol de IAM son un motivo común de errores de ejecución de la programación. A continuación, se muestra cómo solucionar estos problemas:

Causas habituales

- Faltan los permisos necesarios para el servicio de destino
- Configuración de rol incorrecta en la programación
- Falta una relación de confianza con EventBridge el servicio Scheduler
- Permisos insuficientes para acceder a los recursos cifrados

Síntomas

- Aumento de la `TargetErrorCount` métrica en CloudWatch
- Las programaciones no se ejecutan sin problemas aparentes en la configuración de la programación

Pasos para la solución de problemas

1. Supervise CloudWatch las métricas

- Compruebe la `TargetErrorCount` métrica CloudWatch.

2. Usa Dead-Letter Queue (DLQ) para confirmar los problemas de permisos

- Configure una DLQ para la programación.
- Si hay problemas de permisos con el destino y la DLQ está configurada correctamente, verá las invocaciones erróneas en la DLQ con mensajes de error relacionados con los permisos.
- Si el DLQ permanece vacío a pesar de que las CloudWatch métricas muestran las ejecuciones fallidas, es probable que se deba a un problema de permisos que impide que EventBridge Scheduler escriba en el propio DLQ.

Note

Asegúrese de que la propia DLQ tenga los permisos correctos. Si está cifrada, asegúrate de que EventBridge Scheduler tenga permiso para usar la clave KMS.

3. Verificación de la relación de confianza

- Asegúrese de que su función de IAM tenga la relación de confianza correcta con EventBridge Scheduler:

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

4. Comprobación de los permisos de los roles de ejecución de la programación

- El rol de ejecución de la programación necesita permisos específicos para invocar diferentes tipos de destinos.
- Permisos de ejemplo para incluir en la política de roles de ejecución de la programación:

```
// For Lambda function targets - add to schedule execution role
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda:region:account-id:function:function-name"
  }]
}

// For SQS queue targets - add to schedule execution role
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "sqs:SendMessage"
    ],
    "Resource": "arn:aws:sqs:region:account-id:queue-name"
  }]
}
```

5. Comprobación del acceso cifrado a los recursos

- Si su objetivo utiliza recursos cifrados (por ejemplo, colas de KMS-encrypted SQS), asegúrese de que su función tenga permisos para usar la clave KMS:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:region:account-id:key/key-id"
    }
  ]
}
```

```
]
}
```

6. Verificación de la configuración del ARN del rol

- Asegúrese de que el ARN del rol en la configuración de la programación sea correcto.
- Compruebe que el rol existe en la misma región Cuenta de AWS y en la que está programado.

Descripción y administración de Service Quotas

Si tienes problemas para crear horarios o recibes invocaciones limitadas, es posible que estés alcanzando los límites de las cuotas de servicio. EventBridge Scheduler tiene cuotas para la cantidad de programaciones, grupos de programaciones y tasas de invocación, que pueden variar según la región.

Identificación de problemas de cuota

Para determinar si está alcanzando los límites de cuota:

1. Supervise las métricas CloudWatch

- Compruebe la métrica `InvocationThrottleCount`. Un aumento en esta métrica indica que está excediendo el límite de frecuencia de invocación.
- Revise la métrica `InvocationAttemptCount` para entender el uso actual.

2. Prestar atención a los mensajes de error específicos

- Al crear o modificar programaciones, una `LimitExceededException` indica que ha alcanzado el número máximo de programaciones o grupos de programaciones.
- Las llamadas a la API que devuelven errores de limitación sugieren que está excediendo la cuota de solicitudes de la API.

Resolución de problemas de cuota

Si determina que está alcanzando los límites de cuota:

1. Revise y optimice las programaciones actuales. Considere la posibilidad de consolidar programas similares o eliminar los que no se utilicen.
2. Para limitar las API, implemente el [reintento con retroceso](#) en las llamadas a la API.

3. Si necesita cuotas más altas, solicite un aumento a través de la consola de Service Quotas. Seleccione EventBridge Scheduler, elige la cuota que necesitas aumentar y envía una solicitud con tu justificación empresarial.

Problemas de tiempo de activación y patrones de programación

A veces, los usuarios se topan con problemas debido a que los horarios no se activan a las horas esperadas. En la mayoría de los casos, esto puede deberse a malentendidos sobre los patrones de programación, a los cambios del horario de verano o a los periodos de tiempo flexibles.

Causas habituales

- Interpretación errónea de las expresiones cron.
- Comportamiento inesperado durante los cambios de horario de verano.
- Confusión sobre periodos de tiempo flexibles.
- Malentendido de las expresiones de frecuencia.

Pasos para la solución de problemas

1. Verificación de las expresiones cron
 - Asegúrese de que la expresión cron tiene el formato correcto.
 - Tenga en cuenta que no puede especificar los campos Día del mes y Día de la semana de forma simultánea en una expresión cron.
2. Consideraciones sobre la zona horaria
 - Seleccione la zona horaria que prefiera al crear la programación.
 - Comprenda cómo afecta el horario de verano a su programación, ya que este ajuste se basa en UTC.

Ejemplo del impacto del horario de verano: si configura un horario para que se ejecute a las 7:00 GMT:

- Durante el invierno: el horario es a las 7:00 GMT (ya que GMT = UTC)
- Durante el verano: el horario sigue funcionando a las 7:00 a.m. UTC, que ahora son las 6:00 a.m. GMT/BST

Si necesita que el horario se ejecute a la misma hora local durante todo el año, asegúrese de seleccionar la zona horaria adecuada al crear el horario y de qué manera el horario de verano puede afectar a esa zona horaria.

3. Descripción de los periodos de tiempo flexibles

- [Las ventanas horarias flexibles](#) permiten a EventBridge Scheduler optimizar las invocaciones.
- Es posible que la programación no se active exactamente al principio del periodo.
- Supervise los tiempos de invocación reales para comprender el comportamiento.

4. Revisión de las expresiones de frecuencia y cron

- Asegúrese de que las expresiones de frecuencia tengan el formato correcto (por ejemplo, `rate(5 minutes)`, `rate(1 hour)`).
- Para las expresiones de frecuencia y cron, tenga en cuenta que las invocaciones de programación no se limitan al segundo 0 de un minuto.
- Las programaciones se pueden activar en el minuto especificado, pero no necesariamente al principio exacto del minuto.

Por ejemplo:

- Una programación con `rate(1 hour)` puede ser a las 14:00:45., 15:00:32, 16:00:18, etc.
- Una programación de cron establecida para `0 * * * ? *` (cada hora) podría ser a las 14:00:15, 15:00:07, 16:00:52, etc.

5. Supervise las métricas CloudWatch

- Use la métrica `InvocationAttemptCount` para verificar si la programación se está activando.
- Compruebe `TargetErrorCount` si las invocaciones producen un error.
- Si ha configurado una Dead-Letter cola, supervise para realizar un seguimiento de `InvocationsSentToDeadLetterCount` las invocaciones fallidas.

Creación de patrones de programación y expresiones cron

Los usuarios suelen tener problemas al crear patrones de programación, especialmente con las expresiones cron. A continuación, se indican algunos problemas comunes y la forma de resolverlos:

Problemas comunes

- Sintaxis de cron incorrecta

- Se está intentando utilizar características de cron no compatibles
- Confusión sobre qué campos se pueden utilizar de forma conjunta

Pasos para la solución de problemas

1. Revisión de la sintaxis de las expresiones cron

- Asegúrese de que la expresión cron siga el [formato](#) correcto: Minutes Hours Day-of-month Month Day-of-week Year.
- Recuerde que EventBridge Scheduler utiliza el estándar cron con un campo de año adicional.

2. Limitaciones de la comprensión

- No se pueden especificar los campos Día del mes y Día de la semana de forma simultánea, como se explica [aquí](#).
- No se admiten las expresiones Cron que conducen a frecuencias superiores a 1 minuto.

3. Uso de la característica de vista previa de la programación

- Al crear o editar una programación, EventBridge Scheduler proporciona una vista previa de los próximos 10 tiempos de ejecución.
- Utilice esta vista previa para comprobar que la programación se ejecutará a las horas previstas.
- Si la vista previa no coincide con las expectativas, revise y ajuste la expresión cron.

¿Se está activando mi destino?

Para confirmar si el destino se está activando:

1. Compruebe las CloudWatch métricas:

- `InvocationAttemptCount` muestra el número de intentos de invocación
- `TargetErrorCount` indica si alguna invocación ha producido un error
- `TargetErrorThrottledCount` muestra si el destino está siendo limitado
- `InvocationDroppedCount` indica si se descartó alguna invocación

2. [Configure una Dead-Letter cola](#) (DLQ) para capturar y analizar cualquier invocación fallida.

Destinos con plantilla frente a destinos universales

Si recibe un mensaje de error como “Se ha proporcionado una solicitud no válida: [servicio] no es un servicio compatible para un destino”, es posible que esté intentando utilizar un servicio no compatible como destino con plantilla.

Para resolverlo:

1. Compruebe si el servicio deseado es compatible como [destino con plantilla](#).
2. Si no es compatible, utilice un [destino universal](#) en su lugar y configúrelo para que realice la llamada a la API adecuada al servicio.

Configuraciones de entrada de destino universales no válidas

Al crear una programación con un [destino universal](#), EventBridge Scheduler valida el formato ARN de destino, pero no valida el contenido del Input campo con la API del servicio descendente. Esto significa que un programa se puede crear correctamente incluso si Input contiene valores que el servicio de destino rechazará en el momento de la invocación.

Los horarios con configuraciones de entrada de destino no válidas se activan en la expresión configurada, pero fallan en cada invocación. Es posible que no descubra el error de configuración hasta que se invoque la programación, lo que puede ocurrir horas o días después de su creación.

Síntomas

- La programación se creó sin errores, pero la TargetErrorCount CloudWatch métrica aumenta con cada invocación.
- Los mensajes de DLQ contienen códigos de error del servicio de destino (por ejemplo, InvalidParameterValueException o ValidationException), no. `AWS.Scheduler.InternalServerError`
- El mensaje ERROR_MESSAGE del DLQ hace referencia a errores específicos de validación de parámetros de entrada.

Ejemplos

Los siguientes ejemplos muestran las configuraciones de entrada no válidas más comunes para un objetivo AWS Lambda universal (`arn:aws:scheduler:::aws-sdk:lambda:invoke`).

Calificadores que no coinciden

Una programación con la siguiente entrada especifica la versión 2 en el campo `FunctionName` y la versión 1 en el `Qualifier` campo:

```
{
  "FunctionName": "MyFunction:2",
  "Qualifier": "1"
}
```

Esta programación se creó correctamente, pero todas las invocaciones fallan. El mensaje de DLQ contiene:

- `ERROR_CODE`: `InvalidParameterValueException`
- `ERROR_MESSAGE`: `The derived qualifier from the function name does not match the specified qualifier.`

Nombre de función no válido

Una programación con la siguiente entrada especifica un valor de solo espacios en blanco para: `FunctionName`

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

El mensaje de DLQ contiene:

- `ERROR_CODE`: `ValidationException`
- `ERROR_MESSAGE`: un error de validación que indica que el nombre de la función no coincide con el patrón requerido.

¿Cómo resolverlo

1. Configure un DLQ. [Configure siempre una cola de letra muerta](#) para los horarios que utilizan objetivos universales. Los atributos (`ERROR_CODE` y `ERROR_MESSAGE`) de los mensajes de DLQ contienen el error específico devuelto por el servicio de destino, que identifica el parámetro de entrada no válido.

2. Valide los parámetros de entrada con la API del servicio de destino. Antes de crear una programación, comprueba que el JSON de tu Input campo contiene valores válidos llamando directamente a la API de destino. Por ejemplo, invoca tu AWS Lambda función con los mismos parámetros mediante la AWS Lambda Invoke API para confirmar que la solicitud se ha realizado correctamente.
3. Realice la prueba con un programa único. Cree una programación única para comprobar que la invocación del destino se ha realizado correctamente antes de configurar una programación recurrente.
4. Revisa la referencia de la API del servicio de destino. Comprueba la referencia de la API del servicio al que te diriges para confirmar los parámetros necesarios, los rangos de valores válidos y las restricciones. Para ello AWS Lambda Invoke, consulta [Invoke](#) en la Guía para AWS Lambda desarrolladores.

Programación de actualizaciones que activen invocaciones inesperadas

Al realizar un cambio en una programación, es posible que las invocaciones no reflejen inmediatamente la programación actualizada. Espere un breve periodo para que los cambios surtan efecto. Por ejemplo, si actualiza una programación cerca de su hora de activación original, es posible que vea una invocación en función de la configuración de la programación original.

Habilitación o desactivación de las programaciones únicas

Al volver a habilitar un programa único después de que haya transcurrido su hora programada original, el programa puede invocar inmediatamente su destino. Esto puede ocurrir incluso si la programación se desactivó antes de su hora de ejecución original.

Por ejemplo:

- Hora actual: 13:15 UTC
- One-time horario creado para: 13:30 UTC
- Programación desactivada antes de las 13:30 UTC
- La programación se ha vuelto a activar a las 14:00 UTC
- Resultado: se puede invocar el destino inmediatamente después de la rehabilitación

Historial de documentos de la Guía del usuario del EventBridge Scheduler

En la siguiente tabla se describen las versiones de la documentación de EventBridge Scheduler.

Cambio	Descripción	Fecha
Política AmazonEventBridgeSchedulerReadOnlyAccess gestionada actualizada	Se actualizó la política AmazonEventBridgeSchedulerReadOnlyAccess gestionada para utilizar acciones comodín (scheduler:List* y scheduler:Get*). <ul style="list-style-type: none"> • the section called “AWS actualizaciones de políticas gestionadas” 	25 de marzo de 2026
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. <ul style="list-style-type: none"> • the section called “Prevención del suplente confuso” 	7 de septiembre de 2023
Eliminación automática de programaciones una vez finalizadas	EventBridge El programador admite la eliminación automática. Al configurar la eliminación automática, EventBridge Scheduler elimina	2 de agosto de 2023

la programación después de la última invocación planificada.

- [the section called “Eliminación tras la finalización de la programación”](#)

[Tema actualizado sobre el uso de destinos universales](#)

Se actualizó la lista de servicios compatibles a los que EventBridge Scheduler puede dirigirse y con los que puede integrarse. 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.

17 de marzo de 2023

- [the section called “Uso de destinos universales”](#)

[Información actualizada sobre las programaciones basadas en frecuencia que no tienen fecha de inicio](#)

Se agregó información sobre cómo EventBridge Scheduler gestiona los horarios basados en tarifas si no se especifica un. [StartDate](#)

17 de marzo de 2023

- [the section called “Programaciones basadas en frecuencia”](#)

[Nuevo tema sobre la gestión de los grupos de planificadores](#)

Se agregó un nuevo capítulo sobre cómo crear grupos de planificadores con Scheduler . EventBridge Utilice este capítulo para aprender a crear un grupo, añadir horarios al grupo, aplicar etiquetas para gestionar y supervisar más fácilmente los recursos del EventBridge Scheduler y, por último, eliminar un grupo.

17 de marzo de 2023

- [Administrar un grupo de programaciones](#)

[Nuevos temas sobre el horario de verano y las zonas horarias](#)

Se agregaron nuevas secciones que describen cómo EventBridge Scheduler gestiona el horario de verano y cómo se pueden crear horarios en diferentes zonas horarias.

17 de noviembre de 2022

- [the section called “Horario de verano”](#)
- [the section called “Zonas horarias”](#)

[Nuevo tema sobre métricas](#)

Se agregó un nuevo tema que describe las métricas en las que EventBridge Scheduler publica. CloudWatch Puede utilizar estas métricas para supervisar los errores de invocación y comprender cómo resolver los problemas relacionados con sus programaciones.

15 de noviembre de 2022

- [the section called “Monitoreo con CloudWatch”](#)

[Versión inicial](#)

Versión inicial de la Guía del usuario del EventBridge Scheduler.

10 de noviembre de 2022

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.