aws

Guía del usuario

Amazon CloudWatch Logs

Amazon CloudWatch Logs: Guía del usuario

Table of Contents

¿Qué es Amazon CloudWatch Logs?

Puede usar Amazon CloudWatch Logs para monitorear, almacenar y acceder a sus archivos de registro desde instancias de Amazon Elastic Compute Cloud (Amazon EC2) AWS CloudTrail, Route 53 y otras fuentes.

CloudWatch Logs le permite centralizar los registros de todos los sistemas, aplicaciones y AWS servicios que utilice en un único servicio altamente escalable. A continuación, podrá verlos fácilmente, buscarlos por códigos o patrones de error específicos, filtrarlos en función de campos específicos o archivarlos de forma segura para futuros análisis. CloudWatch Los registros le permiten ver todos sus registros, independientemente de su origen, como un flujo único y coherente de eventos ordenados por tiempo.

CloudWatch Los registros también permiten consultar los registros con un potente lenguaje de consulta, auditar y enmascarar los datos confidenciales de los registros y generar métricas a partir de los registros mediante filtros o un formato de registro integrado.

CloudWatch Los registros admiten dos clases de registros. Los grupos de CloudWatch registros de la clase Logs Standard admiten todas las funciones de CloudWatch Logs. Los grupos de registros de la clase de CloudWatch registros Logs Infrequent Access incurren en cargos de ingesta más bajos y admiten un subconjunto de las capacidades de la clase Estándar. Para obtener más información, consulte Clases de registro.

Características

- Dos clases de registros para mayor flexibilidad: CloudWatch Logs ofrece dos clases de registros, por lo que puede disponer de una opción rentable para los registros a los que accede con poca frecuencia. También dispone de una opción con todas las funciones para los registros que requieren supervisión en tiempo real u otras funciones. Para obtener más información, consulte Clases de registro.
- Consulte sus datos de registro: puede utilizar CloudWatch Logs Insights para buscar y analizar sus datos de registro de forma interactiva. Puede realizar consultas para responder de manera más eficiente y eficaz a los problemas operativos. CloudWatch Logs Insights incluye un lenguaje de consultas diseñado específicamente con unos pocos comandos simples pero potentes. Proporcionamos consultas de ejemplo, descripciones de comandos, autocompletado de consultas y detección de campos de registro para ayudarle a comenzar. Se incluyen ejemplos de consultas

para varios tipos de registros de AWS servicio. Para empezar, consulte Análisis de datos de registro con CloudWatch Logs Insights.

- Detecte y depure con Live Tail: puede utilizar Live Tail para solucionar de forma rápida los incidentes al consultar una lista en streaming de los nuevos eventos de registro a medida que se incorporan. Puede ver, filtrar y destacar los registros incorporados casi en tiempo real, lo que ayuda a detectar y resolver problemas con mayor rapidez. Puede filtrar los registros en función de los términos que especifique y, también, destacar los registros que contienen términos específicos para ayudarlo a encontrar lo que busca con rapidez. Para obtener más información, consulte Use Live Tail para ver los registros casi en tiempo real.
- Supervise los registros de las instancias de Amazon EC2: puede utilizar CloudWatch los registros para supervisar las aplicaciones y los sistemas mediante datos de registro. Por ejemplo, CloudWatch Logs puede realizar un seguimiento del número de errores que se producen en los registros de su aplicación y enviarle una notificación cada vez que la tasa de errores supere el umbral que especifique. CloudWatch Logs utiliza sus datos de registro para la supervisión, por lo que no es necesario cambiar el código. Por ejemplo, puede supervisar los registros de las aplicaciones para detectar términos literales específicos (como NullReferenceException «») o contar el número de veces que aparece un término literal en una posición determinada de los datos de registro (como los códigos de estado «404» en un registro de acceso de Apache). Cuando se encuentra el término que busca, CloudWatch Logs reporta los datos a la CloudWatch métrica que especifique. Los datos de registro están cifrados mientras están en tránsito y cuando están en reposo. Para empezar, consulte Cómo empezar con CloudWatch los registros.
- Supervise los eventos AWS CloudTrail registrados: puede crear alarmas CloudWatch y recibir notificaciones sobre una actividad concreta de la API tal como la capture, CloudTrail y utilizar la notificación para solucionar problemas. Para empezar, consulta Cómo enviar CloudTrail eventos a los CloudWatch registros en la Guía del AWS CloudTrail usuario.
- Auditar y ocultar los datos confidenciales: si tiene datos confidenciales en sus registros, puede ayudar a protegerlos con políticas de protección de datos. Estas políticas le permiten auditar y enmascarar los datos de registro confidenciales. Si habilita la protección de datos, se ocultarán de forma predeterminada los datos confidenciales que coincidan con los identificadores de datos que seleccione. Para obtener más información, consulte Ayude a proteger los datos de registro confidenciales con el enmascaramiento.
- Retención de registros: de forma predeterminada, los registros se conservan de forma indefinida y no vencen nunca. Puede ajustar la política de retención para cada grupo de registros, manteniendo la retención indefinida o seleccionar un periodo de retención de entre 10 años y un día.

- Archivar datos de registro: puede usar CloudWatch los registros para almacenar sus datos de registro en un almacenamiento de alta durabilidad. El agente de CloudWatch registros facilita el envío rápido de datos de registro rotados y no rotados desde un host al servicio de registro. Posteriormente, cuando lo necesite, podrá obtener acceso a los datos de log en su estado original.
- Registrar consultas de DNS de Route 53: puede usar CloudWatch los registros para registrar información sobre las consultas de DNS que recibe Route 53. Para obtener más información, consulte Registro de consultas de DNS en la Guía para desarrolladores de Amazon Route 53.

AWS Servicios relacionados

Los siguientes servicios se utilizan junto con CloudWatch los registros:

- AWS CloudTrailes un servicio web que le permite supervisar las llamadas realizadas a la API de CloudWatch Logs de su cuenta, incluidas las llamadas realizadas por el AWS Management Console, AWS Command Line Interface (AWS CLI) y otros servicios. Cuando el CloudTrail registro está activado, CloudTrail captura las llamadas a la API de su cuenta y envía los archivos de registro al bucket de Amazon S3 que especifique. Cada archivo de registro puede contener uno o varios registros, en función de la cantidad de acciones que se deben realizar para satisfacer una solicitud. Para obtener más información al respecto AWS CloudTrail, consulte ¿Qué es AWS CloudTrail? en la Guía AWS CloudTrail del usuario. Para ver un ejemplo del tipo de datos que se CloudWatch escriben en los archivos de CloudTrail registro, consulteEl registro CloudWatch registra las operaciones de la API y la consola en AWS CloudTrail.
- AWS Identity and Access Management (IAM) es un servicio web que le ayuda a controlar de forma segura el acceso de sus usuarios a los AWS recursos. Utilice IAM para controlar quién puede usar los recursos de AWS (autenticación), así como cuáles de ellos pueden usar y cómo pueden hacerlo (autorización). Para obtener más información, consulte ¿Qué es IAM? en la Guía del usuario de IAM.
- Amazon Kinesis Data Streams es un servicio web que puede utilizar para una entrada y agregación de datos rápida y continua. El tipo de datos utilizado incluye los datos de registros de infraestructura de TI, registros de aplicaciones, redes sociales, fuentes de datos de mercado y datos de secuencias de clics en sitios web. Dado el tiempo de respuesta necesario para la entrada y el procesamiento de datos se realiza en tiempo real, el procesamiento suele ser ligero. Para obtener más información, consulte Qué son los Amazon Kinesis Data Streams en la Guía para desarrolladores de Amazon Kinesis Data Streams.
- AWS Lambda es un servicio web que puede utilizar para la creación de aplicaciones que respondan rápidamente a nueva información. Cargue su código de aplicación como funciones

de Lambda y Lambda ejecuta el código en una infraestructura informática de alta disponibilidad y ejecuta la administración integral de los recursos informáticos, incluido el mantenimiento del servidor y del sistema operativo, el aprovisionamiento de capacidad y el escalado automático, la implementación de parches de seguridad y código, así como el monitoreo y los registros. Lo único que tiene que hacer es suministrar el código en uno de los lenguajes que admite Lambda. Para obtener más información, consulte ¿Qué es? AWS Lambda en la Guía para AWS Lambda desarrolladores.

Precios

Cuando te registres AWS, podrás empezar a usar CloudWatch Logs de forma gratuita mediante la capa AWS gratuita.

Se aplican tarifas estándar a los registros almacenados por otros servicios mediante CloudWatch registros (por ejemplo, registros de flujo de Amazon VPC y registros de Lambda).

Para obtener más información sobre los precios, consulta Amazon CloudWatch Pricing.

Para obtener más información sobre cómo analizar los costos y el uso de CloudWatch los registros y CloudWatch las mejores prácticas sobre cómo reducir los costos, consulta CloudWatch facturación y costos.

Conceptos CloudWatch de Amazon Logs

La terminología y los conceptos fundamentales para su comprensión y uso de CloudWatch los registros se describen a continuación.

Clase de registro

CloudWatch Los registros ofrecen dos clases de grupos de registros. La clase de registro estándar es una opción con todas las funciones para los registros que requieren supervisión en tiempo real o para los registros a los que se accede con frecuencia. La clase de registro de acceso poco frecuente es una opción más económica para los registros a los que se accede con menos frecuencia. Admite un subconjunto de las capacidades de la clase de registro estándar.

Eventos de registro

Un evento de registro es un registro de algunas actividades guardado por la aplicación o el recurso que se está monitorizando. El registro de eventos que CloudWatch Logs entiende

contiene dos propiedades: la marca de tiempo del momento en que ocurrió el evento y el mensaje del evento sin procesar. Los mensajes de evento deben estar cifrados con UTF-8.

Flujos de registro

Un flujo de registro es una secuencia de eventos de registro que comparten la misma fuente. En concreto, un flujo de registro en general, está pensado para representar la secuencia de eventos procedente de la instancia de aplicación o recurso que se monitoriza. Por ejemplo, un flujo de registro puede asociarse a un registro de acceso de Apache en un host específico. Cuando ya no necesite un flujo de registro, puede eliminarlo mediante el comando aws logs delete-log-stream.

Grupos de registro

Los grupos de registros definen grupos de flujos de registro que comparten la misma configuración de retención, monitorización y control de acceso. Cada flujo de registro tiene que pertenecer a un grupo de registros. Por ejemplo, si tiene un flujo de registro diferente para los registros de acceso de Apache de cada host, puede agrupar estos flujos en un solo grupo de registros denominado `MyWebsite.com/Apache/access_log`.

No hay límites en el número de flujos de registros que pueden pertenecer a un grupo de registros.

Filtros de métricas

Puede usar filtros de métricas para extraer observaciones métricas de los eventos ingeridos y transformarlas en puntos de datos de una CloudWatch métrica. Los filtros de métricas se asignan a grupos de registro y todos los filtros asignados a un grupo de registros se aplican a sus flujos de registro.

Configuración de retención

La configuración de retención se puede usar para especificar cuánto tiempo se guardan los eventos de registro en CloudWatch los registros. Los eventos de registro caducados se eliminarán automáticamente. De la misma forma que los filtros de métricas, los ajustes de retención también se asignan a los grupos de registro y la retención asignada a un grupo de registros se aplica a sus flujos de registro.

Facturación y costo de Registros de Amazon CloudWatch

Para obtener información más detallada sobre cómo analizar los costos y el uso de CloudWatch y Registros de CloudWatch, y para conocer las prácticas recomendadas de cómo reducir los costos, consulte CloudWatch billing and cost.

Para obtener más información sobre precios, consulte Precios de Amazon CloudWatch.

Cuando se registra en AWS, puede comenzar a utilizar Registros de CloudWatch de forma gratuita con la capa gratuita de AWS.

Se aplican las tarifas estándar para los registros almacenados por otros servicios que utilicen Registros de CloudWatch (por ejemplo, registros de flujo de Amazon VPC y registros de Lambda).

Clases de registro

CloudWatch Logs ofrece dos clases de grupos de registros:

- La clase de registro CloudWatch Logs Standard es una opción con todas las funciones para los registros que requieren supervisión en tiempo real o para los registros a los que se accede con frecuencia.
- La clase de registro CloudWatch Logs Infrequent Access es una nueva clase de registro que puede utilizar para consolidar sus registros de forma rentable. Esta clase de registro ofrece un subconjunto de funciones de registro que incluyen la administración de CloudWatch registros, el almacenamiento, el análisis de registros entre cuentas y el cifrado, con un precio de ingesta más bajo por GB. La clase de registro de acceso poco frecuente es ideal para consultas ad hoc y after-the-fact análisis forenses de registros a los que se accede con poca frecuencia.

> **Note**
>
> En cuanto a los cargos, las clases de registros de acceso estándar e infrecuente solo difieren en los costos de administración. Los cargos de almacenamiento y CloudWatch los de Logs Insights son los mismos en cada clase de registro.

Para obtener más información sobre CloudWatch los precios de Logs, consulta CloudWatch los precios de Amazon.

> **Important**
>
> Una vez creado un grupo de registros, su clase de registro no se puede cambiar.

Características admitidas

En la siguiente tabla se enumeran las funciones de cada clase de registro.

	Estándar	Acceso poco frecuente	
Ingesta y almacenamiento de registros totalmente gestionados	✓	✓	
Funciones multicuenta	✓	✓	
Cifrado con AWS KMS	✓	✓	
CloudWatch Comandos de consulta de Logs Insights	✓	✓ (La mayoría de los comandos, consulteComandos compatibles con las clases de registro.)	
CloudWatch Registra los campos detectados por Insights	✓		
Ayuda para realizar consultas en lenguaje natural	✓		
CloudWatch Detección de anomalías en los registros	✓		
Compare con el intervalo de tiempo anterior	✓		
Filtros de suscripción	✓		
Exportar a Amazon S3.	✓		
GetLogEventsy operaciones FilterLogEventsde API	✓	No admitido. Utilice CloudWatch Logs Insights	

	Estándar	Acceso poco frecuente	
		para ver los eventos de registro almacenados en grupos de registros de la clase de registro de acceso poco frecuente.	
Filtros métricos	✓		
Ingesta de registros de Container Insights	✓		
Ingesta de registros de Lambda Insights	✓		
Protección de datos confidenciales con enmascaramiento	✓		
Formato de métricas integrado	✓		

Cómo empezar con CloudWatch los registros

Para recopilar registros de sus instancias Amazon EC2 y servidores locales en CloudWatch Logs, utilice el agente unificado. CloudWatch Permite recopilar registros y métricas avanzadas con un solo agente. Ofrece compatibilidad con distintos sistemas operativos, incluidos los servidores que ejecutan Windows Server. Este agente también proporciona un mejor rendimiento.

Si utiliza el CloudWatch agente unificado para recopilar CloudWatch métricas, permite recopilar métricas adicionales del sistema para que los huéspedes las vean mejor. También admite la recopilación de métricas personalizadas mediante `StatsD` o `collectd`.

Para obtener más información, consulte Instalación del CloudWatch agente en la Guía del CloudWatch usuario de Amazon.

El antiguo agente de CloudWatch registros, que solo admite la recopilación de registros de servidores que ejecutan Linux, está obsoleto y ya no es compatible. Para obtener información sobre la migración del antiguo agente de CloudWatch Logs al agente unificado, consulte Crear el archivo de configuración del CloudWatch agente con el asistente.

Contenido

Requisitos previos

Para usar Amazon CloudWatch Logs necesitas una AWS cuenta. Su AWS cuenta le permite usar servicios (por ejemplo, Amazon EC2) para generar registros que puede ver en la CloudWatch consola, una interfaz basada en la web. Además, puede instalar y configurar el AWS Command Line Interface (AWS CLI).

Inscríbase en una Cuenta de AWS

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

Para suscribirte a una Cuenta de AWS

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

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

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

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

Creación de un usuario con acceso administrativo

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

Proteja su Usuario raíz de la cuenta de AWS

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

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

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

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

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

Iniciar sesión como usuario con acceso de administrador

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

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

Concesión de acceso a usuarios adicionales

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

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

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

Configurar la interfaz de línea de comandos

Puede utilizar el AWS CLI para realizar operaciones de CloudWatch registro.

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

Utilice el CloudWatch agente unificado para empezar a utilizar Logs CloudWatch

Para obtener más información sobre el uso del CloudWatch agente unificado para empezar con CloudWatch los registros, consulte Recopilar métricas y registros de instancias de Amazon EC2 y servidores locales con el CloudWatch agente en la Guía del usuario de Amazon CloudWatch . Realice los pasos indicados en esta sección para instalar, configurar e iniciar el agente. Si no utiliza el agente para recopilar también CloudWatch métricas, puede ignorar cualquier sección que haga referencia a las métricas.

Si actualmente utiliza el antiguo agente de CloudWatch Logs y desea migrar al nuevo agente unificado, le recomendamos que utilice el asistente incluido en el nuevo paquete de agentes. Este asistente puede leer el archivo de configuración actual del agente de CloudWatch registros y configurar el CloudWatch agente para que recopile los mismos registros. Para obtener más información sobre el asistente, consulte Creación del archivo de configuración del CloudWatch agente con el asistente en la Guía del CloudWatch usuario de Amazon.

Utilice el CloudWatch agente anterior para empezar a utilizar CloudWatch Logs

> **Important**
>
> CloudWatch incluye un CloudWatch agente unificado que puede recopilar registros y métricas de instancias EC2 y servidores locales. El agente anterior, que se utilizaba solo para registros, quedó obsoleto y ya no es compatible.
> Para obtener información sobre la migración del antiguo agente de solo registros al agente unificado, consulte Crear el archivo de configuración del CloudWatch agente con el asistente.
> En el resto de esta sección, se explica el uso del antiguo agente de CloudWatch Logs para los clientes que aún lo utilizan.

Con el agente de CloudWatch registros, puede publicar datos de registro de instancias de Amazon EC2 que ejecutan Linux o Windows Server y eventos registrados desde. AWS CloudTrail En su lugar, le recomendamos que utilice el agente CloudWatch unificado para publicar los datos de registro. Para obtener más información sobre el nuevo agente, consulte Recopilar métricas y

registros de instancias de Amazon EC2 y servidores locales con el CloudWatch agente en la Guía del usuario de Amazon CloudWatch .

Contenido

CloudWatch Registra los requisitos previos del agente

El agente CloudWatch Logs requiere las versiones 2.7, 3.0 o 3.3 de Python y cualquiera de las siguientes versiones de Linux:

- Amazon Linux versión 2014.03.02 o versiones posteriores. No se admite en Amazon Linux 2.
- Ubuntu Server versión 12.04, 14.04 o 16.04
- CentOS versión 6, 6.3, 6.4, 6.5 o 7.0
- Red Hat Enterprise Linux (RHEL) versión 6.5 o 7.0
- Debian 8.0

Inicio rápido: instale y configure el agente CloudWatch Logs en una instancia EC2 Linux en ejecución

Important

El antiguo agente de registros está obsoleto. CloudWatch incluye un agente unificado que puede recopilar registros y métricas de instancias EC2 y servidores locales. Para obtener más información, consulte Cómo empezar con CloudWatch los registros.
Para obtener información sobre la migración del antiguo agente de CloudWatch Logs al agente unificado, consulte Crear el archivo de configuración del CloudWatch agente con el asistente.
El agente de registros anterior solo admite las versiones 2.6 a 3.5 de Python. Además, el antiguo agente de CloudWatch Logs no es compatible con la versión 2 del Servicio de Metadatos de Instancia (IMDSv2). Si su servidor usa IMDSv2, debe usar el agente unificado más reciente en lugar del antiguo agente de Logs. CloudWatch
En el resto de esta sección, se explica el uso del antiguo agente de CloudWatch Logs para los clientes que aún lo utilizan.

Tip

CloudWatch incluye un nuevo agente unificado que puede recopilar registros y métricas de las instancias EC2 y los servidores locales. Si aún no utiliza el agente de CloudWatch Logs anterior, le recomendamos que utilice el agente unificado CloudWatch más reciente. Para obtener más información, consulte Cómo empezar con CloudWatch los registros.
Además, el agente anterior no admite la versión 2 del servicio de metadatos de instancia (IMDSv2). Si su servidor usa IMDSv2, debe usar el agente unificado más reciente en lugar del antiguo agente de CloudWatch Logs.
En el resto de esta sección se explica el uso del antiguo agente de CloudWatch Logs.

Configure el agente CloudWatch Logs anterior en una instancia EC2 Linux en ejecución

Puede utilizar el instalador del agente de CloudWatch registros en una instancia de EC2 existente para instalar y configurar el agente de CloudWatch registros. Una vez que se haya completado la

instalación, los registros fluyen automáticamente desde la instancia al flujo de registros que crea al instalar el agente. El agente confirma que se ha iniciado y sigue en ejecución hasta que lo desactiva.

Además de usar el agente, también puede publicar datos de registro mediante el AWS CLI SDK de CloudWatch Logs o la API de CloudWatch Logs. AWS CLI Es la más adecuada para publicar datos en la línea de comandos o mediante scripts. El SDK de CloudWatch registros es el más adecuado para publicar datos de registro directamente desde aplicaciones o para crear su propia aplicación de publicación de registros.

Paso 1: Configure su rol o usuario de IAM para Logs CloudWatch

El agente CloudWatch de registros admite funciones y usuarios de IAM. Si la instancia ya tiene un rol de IAM asociado, asegúrese de incluir la política de IAM a continuación. Si aún no dispone de un rol de IAM asignado a su instancia, puede utilizar las credenciales de IAM para los siguientes pasos o bien puede asignar un rol de IAM a dicha instancia. Para obtener más información, consulte Adjuntar un rol de IAM a una instancia.

Para configurar su rol o usuario de IAM para Logs CloudWatch

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. Seleccione Roles (Roles) en el panel de navegación.
3. Para elegir el rol, seleccione el nombre de rol (no seleccione la casilla de verificación junto al nombre).
4. Elija Attach Policies (Asociar políticas), Create Policy (Crear política).

   Se abrirá una nueva pestaña o ventana del navegador.
5. Elija la pestaña JSON y escriba el siguiente documento de política JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
```

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

6. Cuando haya terminado, elija Review policy (Revisar la política). El validador de políticas notifica los errores de sintaxis.
7. En la página Review Policy (Revisar la política), escriba un Name (Nombre) y una Description (Descripción) (opcional) para la política que crea. Revise el Summary (Resumen) de la política para ver los permisos concedidos por su política. A continuación, elija Create policy (Crear política) para guardar su trabajo.
8. Cierre la pestaña o ventana del navegador y vuelva a la página Add permissions (Agregar permisos) para su rol. Elija Refresh (Actualizar) y, a continuación, elija la política nueva para adjuntarla al rol.
9. Elija Attach Policy (Adjuntar política).

Paso 2: Instalar y configurar CloudWatch los registros en una instancia Amazon EC2 existente

El proceso de instalación del agente CloudWatch Logs varía en función de si la instancia de Amazon EC2 ejecuta Amazon Linux, Ubuntu, CentOS o Red Hat. Utilice los pasos adecuados para la versión de Linux en su instancia.

Para instalar y configurar CloudWatch Logs en una instancia de Amazon Linux existente

A partir de la AMI 2014.09 de Amazon Linux, el agente CloudWatch Logs está disponible como una instalación RPM con el paquete awslogs. Las versiones anteriores de Amazon Linux pueden obtener acceso al paquete awslogs mediante la actualización de su instancia con el comando `sudo yum update -y`. Al instalar el paquete awslogs como un RPM en lugar de utilizar el instalador de CloudWatch Logs, la instancia recibirá actualizaciones y parches periódicos de los paquetes AWS sin tener que volver a instalar manualmente el agente de Logs. CloudWatch

> **Warning**
>
> No actualice el agente de CloudWatch Logs mediante el método de instalación RPM si anteriormente utilizó el script de Python para instalar el agente. Si lo hace, podrían producirse problemas de configuración que impidan que el CloudWatch agente de Logs envíe sus registros a CloudWatch.

1. Conéctese con su instancia de Amazon Linux. Para obtener más información, consulte Connect to Your Instance en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre los problemas de conexión, consulte Solución de problemas de conexión a su instancia en la Guía del usuario de Amazon EC2.

2. Actualice la instancia de Amazon Linux para recoger los últimos cambios en los repositorios de paquetes.

   ```
   sudo yum update -y
   ```

3. Instale el paquete `awslogs`. Este es el método recomendado para instalar awslogs en instancias de Amazon Linux.

   ```
   sudo yum install -y awslogs
   ```

4. Edite el archivo `/etc/awslogs/awslogs.conf` a fin de configurar los registros para realizar seguimiento. Para obtener más información sobre la edición de este archivo, consulte CloudWatch Registra la referencia del agente.

5. De forma predeterminada, el archivo `/etc/awslogs/awscli.conf` apunta a la región EE. UU.-este-1. Para enviar los registros a una región diferente, edite el archivo `awscli.conf` y especifique dicha región.

6. Inicie el servicio `awslogs`.

   ```
   sudo service awslogs start
   ```

   Si ejecuta Amazon Linux 2, inicie el servicio `awslogs` con el siguiente comando.

   ```
   sudo systemctl start awslogsd
   ```

7. (Opcional) Compruebe el archivo `/var/log/awslogs.log` para ver si se han registrado errores al iniciar el servicio.

8. (Opcional) Ejecute el siguiente comando para iniciar el servicio `awslogs` en cada arranque del sistema.

   ```
   sudo chkconfig awslogs on
   ```

Si ejecuta Amazon Linux 2, utilice el siguiente comando para iniciar el servicio en cada arranque del sistema.

```
sudo systemctl enable awslogsd.service
```

9. Debería ver el grupo de registros y el flujo de registros recién creados en la CloudWatch consola después de que el agente haya estado ejecutándose durante unos instantes.

   Para obtener más información, consulte Vea los datos de registro enviados a Logs CloudWatch .

Para instalar y configurar CloudWatch los registros en una instancia existente de Ubuntu Server, CentOS o Red Hat

Si utiliza una AMI que ejecuta Ubuntu Server, CentOS o Red Hat, utilice el siguiente procedimiento para instalar manualmente el agente de CloudWatch Logs en la instancia.

1. Conéctese a la instancia EC2. Para obtener más información, consulte Connect to Your Instance en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre los problemas de conexión, consulte Solución de problemas de conexión a su instancia en la Guía del usuario de Amazon EC2.

2. Ejecute el instalador del agente CloudWatch Logs mediante una de estas dos opciones. Puede ejecutarlo directamente desde Internet o descargar los archivos y ejecutarlo de forma independiente.

   > **Note**
   >
   > Si ejecuta CentOS 6.x, Red Hat 6.x o Ubuntu 12.04, utilice los pasos para descargar y ejecutar el instalador independiente. Estos sistemas no admiten la instalación del agente de CloudWatch Logs directamente desde Internet.

   > **Note**
   >
   > En Ubuntu, ejecute `apt-get update` antes de ejecutar los comandos siguientes.

Para ejecutarlo directamente desde Internet, utilice los siguientes comandos y siga las instrucciones:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1
```

Si el comando anterior no funciona, pruebe lo siguiente:

```
sudo python3 ./awslogs-agent-setup.py --region us-east-1
```

Para descargar y ejecutarlo de forma independiente, utilice los siguientes comandos y siga las instrucciones:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/AgentDependencies.tar.gz -O
```

```
tar xvf AgentDependencies.tar.gz -C /tmp/
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1 --dependency-path /tmp/AgentDependencies
```

Puede instalar el agente CloudWatch Logs especificando us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1 Regiones eu-central-1, eu-west-1 o sa-east-1.

> **Note**
>
> Para obtener más información sobre la versión actual y el historial de versiones de `awslogs-agent-setup`, consulte CHANGELOG.txt.

El instalador del agente Logs requiere cierta información durante la configuración. CloudWatch Antes de comenzar, debe saber qué archivo de registros monitorear y su formato de marca temporal. También debe tener preparada la siguiente información.

Elemento	Descripción
AWS ID de clave de acceso	Pulse Intro si utiliza un rol de IAM. De lo contrario, introduzca su ID de clave de AWS acceso.
AWS clave de acceso secreta	Pulse Intro si utiliza un rol de IAM. De lo contrario, introduzca su clave de acceso AWS secreta.
Nombre de región predeterminado	Pulse Intro. La región predeterminada es us-east-2. Puede configurarlo a us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1 o sa-east-1.
Formato de salida predeterminado	Déjelo en blanco y pulse Intro.
Ruta del archivo de registros que cargar	La ubicación del archivo que contiene los datos de registro que se van a enviar. El instalador sugiere una ruta.
Nombre de grupo de registros de destino	El nombre de su grupo de registros. El instalador sugiere un nombre de grupo de registros.
Nombre de flujo de registros de destino	De forma predeterminada, es el nombre del host. El instalador sugiere un nombre de host.
Formato de marca temporal	Especifique el formato de la marca temporal en el archivo de registros especificado. Elija personalizado para especificar su propio formato.
Posición inicial	Cómo se han cargado los datos. Establézcalo en start_of_file para cargar todo en el archivo de datos. Establézcalo en end_of_file para cargar solo los datos recién agregados.

Una vez que haya completado estos pasos, el instalador preguntará si desea configurar otro archivo de registros. Puede ejecutar el proceso tantas veces como desee para cada archivo de registros. Si no tiene más archivos de registros que monitorear, elija N cuando el instalador lo solicite para configurar otro registro. Para obtener más información sobre la configuración en el archivo de configuración del agente, consulte CloudWatch Registra la referencia del agente.

> **Note**
>
> No se admite configurar varias fuentes de registro para enviar datos a un único flujo de registro.

3. Debería ver el grupo de registros y el flujo de registros recién creados en la CloudWatch consola después de que el agente haya estado ejecutándose durante unos instantes.

   Para obtener más información, consulte Vea los datos de registro enviados a Logs CloudWatch .

Inicio rápido: instale y configure el agente de CloudWatch registros en una instancia EC2 de Linux en el momento del lanzamiento

> **Tip**
>
> El antiguo agente de CloudWatch registros que se describe en esta sección está en vías de quedar obsoleto. Le recomendamos encarecidamente que, en su lugar, utilice el nuevo CloudWatch agente unificado, que puede recopilar tanto registros como métricas. Además, el agente CloudWatch Logs anterior requiere Python 3.3 o una versión anterior, y estas versiones no se instalan en las nuevas instancias de EC2 de forma predeterminada. Para obtener más información sobre el CloudWatch agente unificado, consulte Instalación del CloudWatch agente.
> En el resto de esta sección se explica el uso del antiguo agente de CloudWatch Logs.

Instalación del antiguo agente de CloudWatch Logs en una instancia EC2 de Linux en el momento del lanzamiento

Puede usar los datos de usuario de Amazon EC2, una función de Amazon EC2 que permite transferir información paramétrica a la instancia en el momento del lanzamiento, para instalar y

configurar CloudWatch el agente Logs en esa instancia. Para pasar la información de instalación y configuración del agente CloudWatch Logs a Amazon EC2, puede proporcionar el archivo de configuración en una ubicación de red, como un bucket de Amazon S3.

No se admite configurar varias fuentes de registro para enviar datos a un único flujo de registro.

Requisito previo

Cree un archivo de configuración de agente que describa todos los grupos de registro y flujos de registro. Se trata de un archivo de texto que describe los archivos de registros que monitorear, así como los grupos de registro y los flujos de registro para cargarlos. El agente consume este archivo de configuración y comienza a monitorear y a cargar todos los archivos de registros descritos en el mismo. Para obtener más información sobre la configuración en el archivo de configuración del agente, consulte CloudWatch Registra la referencia del agente.

A continuación, se muestra un ejemplo de archivo de configuración del agente para Amazon Linux 2.

```
[general]
state_file = /var/lib/awslogs/state/agent-state

[/var/log/messages]
file = /var/log/messages
log_group_name = /var/log/messages
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

A continuación, se muestra un ejemplo de archivo de configuración del agente para Ubuntu.

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/syslog]
file = /var/log/syslog
log_group_name = /var/log/syslog
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Para configurar su rol de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Policies (Políticas), Create Policy (Crear política).

3. En la página Create Policy (Crear política), en Create Your Own Policy (Crear su propia política), elija Select (Seleccionar). Para obtener más información sobre la creación de políticas personalizadas, consulte Políticas de IAM para Amazon EC2 en la Guía del usuario de Amazon EC2.
4. En la página Review Policy (Revisar políticas), en Policy Name (Nombre de la política), escriba un nombre para la política.
5. En Policy Document (Documento de la política), pegue la siguiente política:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::myawsbucket/*"
            ]
        }
    ]
}
```

6. Elija Create Policy (Crear política).
7. En el panel de navegación, elija Roles (Roles), Create New Role (Crear nuevo rol).
8. En la página Set Role Name (Establecer nombre del rol), escriba un nombre de rol y, a continuación, elija Next Step (Siguiente paso).

9. En la página Select Role Type (Seleccionar tipo de rol), elija Select (Seleccionar) junto a Amazon EC2.
10. En la página Attach Policy (Adjuntar política), en el encabezado de la tabla, elija Policy Type (Tipo de política), Customer Managed (Administrada por el cliente).
11. Seleccione la política de IAM que ha creado y, a continuación, elija Next Step (Siguiente paso).
12. Seleccione Crear rol.

    Para obtener más información sobre los usuarios y políticas, consulte Usuarios y grupos de IAM y Administración de políticas de IAM en la Guía del usuario de IAM.

Para lanzar una nueva instancia y habilitar los registros CloudWatch

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Elija Iniciar instancia.

   Para obtener más información, consulte la Guía del usuario sobre el lanzamiento de una instancia en Amazon EC2.
3. En la página Step 1: Choose an Amazon Machine Image (AMI) (Paso 1: elegir una Amazon Machine Image [AMI]), seleccione el tipo de instancia de Linux que desea lanzar y, a continuación, en la página Step 2: Choose an Instance Type (Paso 2: elegir un tipo de instancia), elija Next: Configure Instance Details (Siguiente: configurar detalles de la instancia).

   Asegúrese de que cloud-init se incluye en la Amazon Machine Image (AMI). Las AMI de Amazon Linux y las AMI para Ubuntu y RHEL ya incluyen cloud-init, pero es posible que CentOS y otras AMI no. AWS Marketplace
4. En la página Step 3: Configure Instance Details (Paso 3: configurar detalles de la instancia), en IAM role (Rol de IAM), seleccione el rol de IAM que creó.
5. En Advanced Details (Detalles avanzados), en User data (Datos de usuario), pegue el siguiente script en el cuadro. A continuación, para actualizar el script, cambie el valor de la opción -c a la ubicación de su archivo de configuración del agente:

```
#!/bin/bash
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py -O
chmod +x ./awslogs-agent-setup.py
./awslogs-agent-setup.py -n -r us-east-1 -c s3://DOC-EXAMPLE-BUCKET1/my-config-file
```

6. Realice los demás cambios en la instancia, revise la configuración de lanzamiento y, a continuación, elija Launch (Lanzar).
7. Debería ver el grupo de registros y el flujo de registros recién creados en la CloudWatch consola después de que el agente haya estado ejecutándose durante unos instantes.

   Para obtener más información, consulte Vea los datos de registro enviados a Logs CloudWatch .

Inicio rápido: habilite sus instancias de Amazon EC2 que ejecutan Windows Server 2016 para enviar CloudWatch registros a Logs mediante el CloudWatch agente Logs

> **Tip**
>
> CloudWatch incluye un nuevo agente unificado que puede recopilar registros y métricas de las instancias EC2 y los servidores locales. Le recomendamos que utilice el agente unificado CloudWatch más reciente. Para obtener más información, consulte Cómo empezar con CloudWatch los registros.
> En el resto de esta sección, se explica el uso del antiguo agente de CloudWatch Logs.

Habilite sus instancias de Amazon EC2 que ejecutan Windows Server 2016 para enviar CloudWatch registros a Logs mediante el agente de Logs anterior CloudWatch

Existen varios métodos que puede utilizar para permitir que las instancias que ejecutan Windows Server 2016 envíen CloudWatch registros a Logs. En los pasos de esta sección se utiliza Systems Manager Run Command. Para obtener información sobre los demás métodos posibles, consulta Enviar registros, eventos y contadores de rendimiento a Amazon CloudWatch.

Pasos

- Descargar el archivo de configuración de muestra
- Configure el archivo JSON para CloudWatch
- Crear un rol de IAM para Systems Manager
- Comprobar los requisitos previos de Systems Manager
- Verifique el acceso a Internet
- Habilite CloudWatch los registros mediante el comando Run de Systems Manager

Descargar el archivo de configuración de muestra

Descargue el siguiente archivo de muestra en su ordenador: `AWS.EC2.Windows.CloudWatch.json`.

Configure el archivo JSON para CloudWatch

Usted determina a qué registros desea enviar CloudWatch especificando sus opciones en un archivo de configuración. El proceso de creación de este archivo y especificación de las opciones puede tardar 30 minutos o más en completarse. Después de que haya completado esta tarea una vez, podrá volver a utilizar el archivo de configuración en todas sus instancias.

Pasos

- Paso 1: Habilitar CloudWatch los registros
- Paso 2: Configurar los ajustes para CloudWatch
- Paso 3: configurar los datos que se van a enviar
- Paso 4: configurar el control de flujo
- Paso 5: guardar el contenido JSON

Paso 1: Habilitar CloudWatch los registros

En la parte superior del archivo JSON, cambie “false” a “true” en `IsEnabled`:

```
"IsEnabled": true,
```

Paso 2: Configurar los ajustes para CloudWatch

Especifique las credenciales, la región, el nombre de un grupo de registros y un espacio de nombres de flujo de registros. Esto permite a la instancia enviar datos de registro a CloudWatch Logs. Para enviar los mismos datos de registro a diferentes ubicaciones, puedes añadir secciones adicionales con identificadores únicos (por ejemplo, «CloudWatchLogs2" y CloudWatchLogs 3") y una región diferente para cada identificador.

Para configurar los ajustes para enviar datos de registro a CloudWatch Logs

1. En el archivo JSON, busque la sección `CloudWatchLogs`.

```
{
```

```
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. Deje los campos `AccessKey` y `SecretKey` en blanco. Configurará las credenciales mediante un rol de IAM.
3. En `Region`, escriba la región a la que desea enviar los datos de registro (por ejemplo, `us-east-2`).
4. En `LogGroup`, escriba el nombre del grupo de registros. Este nombre aparece en la pantalla de grupos de registros de la CloudWatch consola.
5. En `LogStream`, escriba el flujo de registros de destino. Este nombre aparece en la pantalla Grupos de registros > Transmisiones de la CloudWatch consola.

   Si utiliza `{instance_id}`, el nombre del flujo de registro de destino predeterminado es el ID de esta instancia.

   Si especifica un nombre de flujo de registro que aún no existe, CloudWatch Logs lo crea automáticamente. Puede definir el nombre del flujo de registro mediante una cadena literal, las variables predefinidas `{instance_id}`, `{hostname}` y `{ip_address}` o una combinación de ellas.

Paso 3: configurar los datos que se van a enviar

Puede enviar los datos del registro de eventos, los datos del seguimiento de eventos para Windows (ETW) y otros datos de registro a CloudWatch Logs.

Para enviar los datos del registro de eventos de aplicaciones de Windows a Logs CloudWatch

1. En el archivo JSON, busque la sección `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
```

```
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. En `Levels`, especifique el tipo de mensajes que desea cargar. Puede especificar uno de los siguientes valores:

   - **1**: cargar solo mensajes de error.
   - **2**: cargar solo mensajes de advertencia.
   - **4**: cargar solo mensajes de información.

   Puede combinar valores para incluir más de un tipo de mensaje. Por ejemplo, un valor **3** carga mensajes de error (**1**) y mensajes de advertencia (**2**). Un valor **7** carga mensajes de error (**1**), mensajes de advertencia (**2**) y mensajes de información (**4**).

Para enviar los datos del registro de seguridad a CloudWatch Logs

1. En el archivo JSON, busque la sección `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. En `Levels`, escriba **7** para cargar todos los mensajes.

Para enviar los datos del registro de eventos del sistema a CloudWatch Logs

1. En el archivo JSON, busque la sección `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. En `Levels`, especifique el tipo de mensajes que desea cargar. Puede especificar uno de los siguientes valores:

   - **1**: cargar solo mensajes de error.
   - **2**: cargar solo mensajes de advertencia.
   - **4**: cargar solo mensajes de información.

   Puede combinar valores para incluir más de un tipo de mensaje. Por ejemplo, un valor **3** carga mensajes de error (**1**) y mensajes de advertencia (**2**). Un valor **7** carga mensajes de error (**1**), mensajes de advertencia (**2**) y mensajes de información (**4**).

Para enviar otros tipos de datos de registro de eventos a CloudWatch Logs

1. En el archivo JSON, agregue una nueva sección. Cada sección debe tener un único `Id`.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. En `Id`, escriba un nombre para el registro que desea cargar (por ejemplo, **`WindowsBackup`**).
3. En `LogName`, escriba el nombre del registro que se va a cargar. Puede encontrar el nombre del registro como se indica a continuación.

a. Abra el lector de eventos.

b. En el panel de navegación, elija Applications and Services Logs (Registros de aplicaciones y servicios).

c. Navegue hasta el registro y elija Actions (Acciones), Properties (Propiedades).

4. En `Levels`, especifique el tipo de mensajes que desea cargar. Puede especificar uno de los siguientes valores:

   - **1**: cargar solo mensajes de error.
   - **2**: cargar solo mensajes de advertencia.
   - **4**: cargar solo mensajes de información.

   Puede combinar valores para incluir más de un tipo de mensaje. Por ejemplo, un valor **3** carga mensajes de error (**1**) y mensajes de advertencia (**2**). Un valor **7** carga mensajes de error (**1**), mensajes de advertencia (**2**) y mensajes de información (**4**).

Para enviar los datos del seguimiento de eventos de Windows a los registros CloudWatch

ETW (Seguimiento de eventos para Windows) proporciona un mecanismo de registro detallado y eficiente en el que las aplicaciones pueden escribir los registros. Cada ETW se controla mediante un administrador de sesiones que puede iniciar y parar la sesión de registro. Cada sesión tiene un proveedor y uno o varios consumidores.

1. En el archivo JSON, busque la sección `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. En `LogName`, escriba el nombre del registro que se va a cargar.

3. En `Levels`, especifique el tipo de mensajes que desea cargar. Puede especificar uno de los siguientes valores:

- **1**: cargar solo mensajes de error.
- **2**: cargar solo mensajes de advertencia.
- **4**: cargar solo mensajes de información.

Puede combinar valores para incluir más de un tipo de mensaje. Por ejemplo, un valor **3** carga mensajes de error (**1**) y mensajes de advertencia (**2**). Un valor **7** carga mensajes de error (**1**), mensajes de advertencia (**2**) y mensajes de información (**4**).

Para enviar registros personalizados (cualquier archivo de registro basado en texto) a Logs CloudWatch

1. En el archivo JSON, busque la sección `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. En `LogDirectoryPath`, escriba la ruta de la instancia donde se almacenan los registros.
3. En `TimestampFormat`, escriba el formato de marcas temporales que desea utilizar. Para obtener más información sobre los valores admitidos, consulte el tema Cadenas con formato de fecha y hora personalizado en MSDN.

> **Important**
>
> Su archivo de registros de fuente debe tener la marca temporal al principio de cada línea de registro y debe haber un espacio en blanco tras dicha marca.

4. En `Encoding`, escriba la codificación del archivo que desea utilizar (por ejemplo, UTF-8). Para obtener una lista de los valores admitidos, consulte el tema Clase de codificación en MSDN.

   > **Note**
   >
   > Utilice el nombre de codificación, no el nombre de visualización.

5. (Opcional) En `Filter`, escriba el prefijo de los nombres de registros. Deje en blanco este parámetro para monitorear todos los archivos. Para obtener más información sobre los valores admitidos, consulte el tema FileSystemWatcherFilter Propiedades en MSDN.
6. (Opcional) En `CultureName`, escriba la configuración regional en la que se registra la marca temporal. Si `CultureName` está en blanco, se utiliza de forma predeterminada la misma configuración regional que utiliza actualmente su instancia de Windows. Para obtener más información, consulte la columna `Language tag` en la tabla del tema relacionado con el Comportamiento del producto en MSDN.

   > **Note**
   >
   > No se admiten los valores `div`, `div-MV`, `hu` y `hu-HU`.

7. (Opcional) En `TimeZoneKind`, escriba `Local` o `UTC`. Puede configurar este valor para proporcionar información sobre la zona horaria cuando no se incluye ninguna información sobre esta en la marca temporal del registro. Si este parámetro se deja en blanco y la marca horaria no incluye información sobre la zona horaria, CloudWatch Logs utilizará de forma predeterminada la zona horaria local. Se hace caso omiso a este parámetro si la marca temporal ya contiene información sobre la zona horaria.
8. (Opcional) En `LineCount`, escriba el número de líneas del encabezado para identificar el archivo de registros. Por ejemplo, los archivos de registros de IIS tienen encabezados prácticamente idénticos. Puede ingresar **5**, que leería las tres primeras líneas del encabezado del archivo de registros para identificarlo. En los archivos de registros de IIS, la tercera línea es la marca de fecha y hora, pero la marca temporal no es siempre diferente entre los archivos de registros. Por este motivo, recomendamos que incluya al menos una línea de datos de registro reales para identificar de forma única el archivo de registros.

Para enviar datos de registro de IIS a CloudWatch Logs

1. En el archivo JSON, busque la sección `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. En `LogDirectoryPath`, escriba la carpeta donde se almacenan los registros de IIS para un sitio individual (por ejemplo, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

> **Note**
>
> Solo se admite el formato de registro W3C. Los formatos IIS, NCSA y personalizados no se admiten.

3. En `TimestampFormat`, escriba el formato de marcas temporales que desea utilizar. Para obtener más información sobre los valores admitidos, consulte el tema Cadenas con formato de fecha y hora personalizado en MSDN.
4. En `Encoding`, escriba la codificación del archivo que desea utilizar (por ejemplo, UTF-8). Para obtener más información sobre los valores admitidos, consulte el tema relacionado con la Clase de decodificación en MSDN.

> **Note**
>
> Utilice el nombre de codificación, no el nombre de visualización.

5. (Opcional) En `Filter`, escriba el prefijo de los nombres de registros. Deje en blanco este parámetro para monitorear todos los archivos. Para obtener más información sobre los valores admitidos, consulte el tema FileSystemWatcherFilter Propiedades en MSDN.
6. (Opcional) En `CultureName`, escriba la configuración regional en la que se registra la marca temporal. Si `CultureName` está en blanco, se utiliza de forma predeterminada la misma

configuración regional que utiliza actualmente su instancia de Windows. Para obtener más información sobre los valores admitidos, consulte la columna `Language tag` en la tabla del tema relacionado con el Comportamiento del producto en MSDN.

> **Note**
>
> No se admiten los valores `div`, `div-MV`, `hu` y `hu-HU`.

7. (Opcional) En `TimeZoneKind`, escriba `Local` o `UTC`. Puede configurar este valor para proporcionar información sobre la zona horaria cuando no se incluye ninguna información sobre esta en la marca temporal del registro. Si este parámetro se deja en blanco y la marca horaria no incluye información sobre la zona horaria, CloudWatch Logs utilizará de forma predeterminada la zona horaria local. Se hace caso omiso a este parámetro si la marca temporal ya contiene información sobre la zona horaria.
8. (Opcional) En `LineCount`, escriba el número de líneas del encabezado para identificar el archivo de registros. Por ejemplo, los archivos de registros de IIS tienen encabezados prácticamente idénticos. Puede ingresar **5**, que leería las cinco primeras líneas del encabezado del archivo de registros para identificarlo. En los archivos de registros de IIS, la tercera línea es la marca de fecha y hora, pero la marca temporal no es siempre diferente entre los archivos de registros. Por este motivo, recomendamos que incluya al menos una línea de datos de registro reales para identificar de forma única el archivo de registros.

Paso 4: configurar el control de flujo

Cada tipo de datos debe tener un destino correspondiente en la sección `Flows`. Por ejemplo, para enviar el registro personalizado, el registro ETW y el registro del sistema a CloudWatch Logs, agréguelos `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` a la `Flows` sección.

> **Warning**
>
> Si se agrega un paso que no es válido, se bloquea el flujo. Por ejemplo, si agrega un paso de métrica de disco, pero la instancia no tiene ningún disco, se bloquean todos los pasos del flujo.

Puede enviar el mismo archivo de registros a varios destinos. Por ejemplo, para enviar el registro de la aplicación a dos destinos diferentes definidos en la sección `CloudWatchLogs`, agregue `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` a la sección `Flows`.

Para configurar el control de flujo

1. En el archivo `AWS.EC2.Windows.CloudWatch.json`, busque la sección `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. En `Flows`, agregue todos los tipos de datos que desea cargar (por ejemplo, `ApplicationEventLog`) y su destino (por ejemplo, `CloudWatchLogs`).

Paso 5: guardar el contenido JSON

Acaba de editar el archivo JSON. Guárdelo y pegue el contenido del archivo en un editor de texto en otra ventana. Necesitará el contenido del archivo en un paso posterior de este procedimiento.

Crear un rol de IAM para Systems Manager

Cuando utiliza Systems Manager Run Command, se necesita un rol de IAM para las credenciales de instancia. Este rol habilita a Systems Manager a realizar acciones en la instancia. Para obtener más información, consulte Configuración de los roles de seguridad para Systems Manager en la Guía del usuario de AWS Systems Manager . Para obtener información sobre cómo asociar un rol de IAM a una instancia existente, consulte Adjuntar un rol de IAM a una instancia en la Guía del usuario de Amazon EC2.

Comprobar los requisitos previos de Systems Manager

Antes de usar Systems Manager Run Command para configurar la integración con CloudWatch los registros, compruebe que las instancias cumplen los requisitos mínimos. Para obtener más información, consulte Requisitos previos de Systems Manager en la Guía del usuario de AWS Systems Manager .

Verifique el acceso a Internet

Sus instancias Amazon EC2 de Windows Server y las instancias administradas deben tener acceso saliente a Internet para poder enviar datos de registro y eventos a. CloudWatch Para obtener más información acerca de cómo configurar el acceso a Internet, consulte Internet Gateways (Gateways de Internet) en la Guía del usuario de Amazon VPC.

Habilite CloudWatch los registros mediante el comando Run de Systems Manager

Run Command habilita la administración de la configuración de las instancias en diferido. Puede especificar un documento de Systems Manager, especificar parámetros y ejecutar el comando en una o varias instancias. El SSM Agent de la instancia procesa el comando y configura la instancia tal y como se especifica.

Para configurar la integración con CloudWatch los registros mediante Run Command

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Abra la consola de SSM en https://console.aws.amazon.com/systems-manager/.
3. En el panel de navegación, elija Run Command (Ejecutar comando).
4. Elija Run a command (Ejecutar un comando).
5. Para el documento de comandos, elija AWS- ConfigureCloudWatch.
6. Para las instancias de Target, elija las instancias que desee integrar con CloudWatch Logs. Si no ve ninguna instancia en esta lista, puede que no esté configurada para Run Command. Para obtener más información, consulte los requisitos previos de Systems Manager en la Guía del usuario de Amazon EC2.
7. En Status (Estado), elija Enabled (Habilitado).
8. En Properties (Propiedades), copie y pegue el contenido JSON que creó en las tareas anteriores.
9. Complete los demás campos opcionales y elija Run (Ejecutar).

Utilice el siguiente procedimiento para ver los resultados de la ejecución del comando en la consola de Amazon EC2.

Para ver la información de salida del comando en la consola

1. Seleccione un comando.

2. Elija la pestaña Output (Salida).
3. Elija View Output (Ver salida). La página de salida de comandos muestra los resultados de la ejecución de comandos.

Inicio rápido: habilite las instancias de Amazon EC2 que ejecutan Windows Server 2012 y Windows Server 2008 para enviar registros a Logs CloudWatch

> **Tip**
>
> CloudWatch incluye un nuevo agente unificado que puede recopilar registros y métricas de las instancias EC2 y los servidores locales. Le recomendamos que utilice el agente unificado CloudWatch más reciente. Para obtener más información, consulte Cómo empezar con CloudWatch los registros.
> En el resto de esta sección, se explica el uso del antiguo agente de CloudWatch Logs.

Habilite sus instancias de Amazon EC2 que ejecutan Windows Server 2012 y Windows Server 2008 para enviar registros a Logs CloudWatch

Siga estos pasos para permitir que las instancias que ejecutan Windows Server 2012 y Windows Server 2008 envíen CloudWatch registros a Logs.

Descargar el archivo de configuración de muestra

Descargue el siguiente archivo JSON de muestra en su ordenador: `AWS.EC2.Windows.CloudWatch.json`. Lo editará en los siguientes pasos.

Configura el archivo JSON para CloudWatch

Usted determina a qué registros desea enviar CloudWatch especificando sus opciones en el archivo de configuración JSON. El proceso de creación de este archivo y especificación de las opciones puede tardar 30 minutos o más en completarse. Después de que haya completado esta tarea una vez, podrá volver a utilizar el archivo de configuración en todas sus instancias.

Pasos

- Paso 1: Habilitar CloudWatch los registros

- Paso 2: Configurar los ajustes para CloudWatch
- Paso 3: configurar los datos que se van a enviar
- Paso 4: configurar el control de flujo

Paso 1: Habilitar CloudWatch los registros

En la parte superior del archivo JSON, cambie “false” a “true” en `IsEnabled`:

```
"IsEnabled": true,
```

Paso 2: Configurar los ajustes para CloudWatch

Especifique las credenciales, la región, el nombre de un grupo de registros y un espacio de nombres de flujo de registros. Esto permite a la instancia enviar datos de registro a CloudWatch Logs. Para enviar los mismos datos de registro a diferentes ubicaciones, puedes añadir secciones adicionales con identificadores únicos (por ejemplo, «CloudWatchLogs2" y CloudWatchLogs 3") y una región diferente para cada identificador.

Para configurar los ajustes para enviar datos de registro a CloudWatch Logs

1. En el archivo JSON, busque la sección `CloudWatchLogs`.

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. Deje los campos `AccessKey` y `SecretKey` en blanco. Configurará las credenciales mediante un rol de IAM.
3. En `Region`, escriba la región a la que desea enviar los datos de registro (por ejemplo, `us-east-2`).

4. En `LogGroup`, escriba el nombre del grupo de registros. Este nombre aparece en la pantalla de grupos de registros de la CloudWatch consola.
5. En `LogStream`, escriba el flujo de registros de destino. Este nombre aparece en la pantalla Grupos de registros > Transmisiones de la CloudWatch consola.

   Si utiliza `{instance_id}`, el nombre del flujo de registro de destino predeterminado es el ID de esta instancia.

   Si especifica un nombre de flujo de registro que aún no existe, CloudWatch Logs lo crea automáticamente. Puede definir el nombre del flujo de registro mediante una cadena literal, las variables predefinidas `{instance_id}`, `{hostname}` y `{ip_address}` o una combinación de ellas.

Paso 3: configurar los datos que se van a enviar

Puede enviar los datos del registro de eventos, los datos del seguimiento de eventos para Windows (ETW) y otros datos de registro a CloudWatch Logs.

Para enviar los datos del registro de eventos de aplicaciones de Windows a Logs CloudWatch

1. En el archivo JSON, busque la sección `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. En `Levels`, especifique el tipo de mensajes que desea cargar. Puede especificar uno de los siguientes valores:

   - **1**: cargar solo mensajes de error.
   - **2**: cargar solo mensajes de advertencia.
   - **4**: cargar solo mensajes de información.

Puede combinar valores para incluir más de un tipo de mensaje. Por ejemplo, un valor **3** carga mensajes de error (**1**) y mensajes de advertencia (**2**). Un valor **7** carga mensajes de error (**1**), mensajes de advertencia (**2**) y mensajes de información (**4**).

Para enviar los datos del registro de seguridad a CloudWatch Logs

1. En el archivo JSON, busque la sección `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. En `Levels`, escriba **7** para cargar todos los mensajes.

Para enviar los datos del registro de eventos del sistema a CloudWatch Logs

1. En el archivo JSON, busque la sección `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. En `Levels`, especifique el tipo de mensajes que desea cargar. Puede especificar uno de los siguientes valores:

   - **1**: cargar solo mensajes de error.
   - **2**: cargar solo mensajes de advertencia.

- **4**: cargar solo mensajes de información.

Puede combinar valores para incluir más de un tipo de mensaje. Por ejemplo, un valor **3** carga mensajes de error (**1**) y mensajes de advertencia (**2**). Un valor **7** carga mensajes de error (**1**), mensajes de advertencia (**2**) y mensajes de información (**4**).

Para enviar otros tipos de datos de registro de eventos a CloudWatch Logs

1. En el archivo JSON, agregue una nueva sección. Cada sección debe tener un único `Id`.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. En `Id`, escriba un nombre para el registro que desea cargar (por ejemplo, **`WindowsBackup`**).
3. En `LogName`, escriba el nombre del registro que se va a cargar. Puede encontrar el nombre del registro como se indica a continuación.
   a. Abra el lector de eventos.
   b. En el panel de navegación, elija Applications and Services Logs (Registros de aplicaciones y servicios).
   c. Navegue hasta el registro y elija Actions (Acciones), Properties (Propiedades).
4. En `Levels`, especifique el tipo de mensajes que desea cargar. Puede especificar uno de los siguientes valores:
   - **1**: cargar solo mensajes de error.
   - **2**: cargar solo mensajes de advertencia.
   - **4**: cargar solo mensajes de información.

Puede combinar valores para incluir más de un tipo de mensaje. Por ejemplo, un valor **3** carga mensajes de error (**1**) y mensajes de advertencia (**2**). Un valor **7** carga mensajes de error (**1**), mensajes de advertencia (**2**) y mensajes de información (**4**).

Para enviar los datos del seguimiento de eventos de Windows a los registros CloudWatch

ETW (Seguimiento de eventos para Windows) proporciona un mecanismo de registro detallado y eficiente en el que las aplicaciones pueden escribir los registros. Cada ETW se controla mediante un administrador de sesiones que puede iniciar y parar la sesión de registro. Cada sesión tiene un proveedor y uno o varios consumidores.

1. En el archivo JSON, busque la sección `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. En `LogName`, escriba el nombre del registro que se va a cargar.
3. En `Levels`, especifique el tipo de mensajes que desea cargar. Puede especificar uno de los siguientes valores:

   - **1**: cargar solo mensajes de error.
   - **2**: cargar solo mensajes de advertencia.
   - **4**: cargar solo mensajes de información.

   Puede combinar valores para incluir más de un tipo de mensaje. Por ejemplo, un valor **3** carga mensajes de error (**1**) y mensajes de advertencia (**2**). Un valor **7** carga mensajes de error (**1**), mensajes de advertencia (**2**) y mensajes de información (**4**).

Para enviar registros personalizados (cualquier archivo de registro basado en texto) a Logs CloudWatch

1. En el archivo JSON, busque la sección `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. En `LogDirectoryPath`, escriba la ruta de la instancia donde se almacenan los registros.
3. En `TimestampFormat`, escriba el formato de marcas temporales que desea utilizar. Para obtener más información sobre los valores admitidos, consulte el tema Cadenas con formato de fecha y hora personalizado en MSDN.

> **Important**
>
> Su archivo de registros de fuente debe tener la marca temporal al principio de cada línea de registro y debe haber un espacio en blanco tras dicha marca.

4. En `Encoding`, escriba la codificación del archivo que desea utilizar (por ejemplo, UTF-8). Para obtener más información sobre los valores admitidos, consulte el tema relacionado con la Clase de decodificación en MSDN.

> **Note**
>
> Utilice el nombre de codificación, no el nombre de visualización.

5. (Opcional) En `Filter`, escriba el prefijo de los nombres de registros. Deje en blanco este parámetro para monitorear todos los archivos. Para obtener más información sobre los valores admitidos, consulte el tema FileSystemWatcherFilter Propiedades en MSDN.
6. (Opcional) En `CultureName`, escriba la configuración regional en la que se registra la marca temporal. Si `CultureName` está en blanco, se utiliza de forma predeterminada la misma configuración regional que utiliza actualmente su instancia de Windows. Para obtener más información sobre los valores admitidos, consulte la columna `Language tag` en la tabla del tema relacionado con el Comportamiento del producto en MSDN.

   > **Note**
   >
   > No se admiten los valores `div`, `div-MV`, `hu` y `hu-HU`.

7. (Opcional) En `TimeZoneKind`, escriba `Local` o `UTC`. Puede configurar este valor para proporcionar información sobre la zona horaria cuando no se incluye ninguna información sobre esta en la marca temporal del registro. Si este parámetro se deja en blanco y la marca horaria no incluye información sobre la zona horaria, CloudWatch Logs utilizará de forma predeterminada la zona horaria local. Se hace caso omiso a este parámetro si la marca temporal ya contiene información sobre la zona horaria.
8. (Opcional) En `LineCount`, escriba el número de líneas del encabezado para identificar el archivo de registros. Por ejemplo, los archivos de registros de IIS tienen encabezados prácticamente idénticos. Puede ingresar **5**, que leería las tres primeras líneas del encabezado del archivo de registros para identificarlo. En los archivos de registros de IIS, la tercera línea es la marca de fecha y hora, pero la marca temporal no es siempre diferente entre los archivos de registros. Por este motivo, recomendamos que incluya al menos una línea de datos de registro reales para identificar de forma única el archivo de registros.

Para enviar datos de registro de IIS a CloudWatch Logs

1. En el archivo JSON, busque la sección `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
```

```
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. En `LogDirectoryPath`, escriba la carpeta donde se almacenan los registros de IIS para un sitio individual (por ejemplo, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

> **Note**
>
> Solo se admite el formato de registro W3C. Los formatos IIS, NCSA y personalizados no se admiten.

3. En `TimestampFormat`, escriba el formato de marcas temporales que desea utilizar. Para obtener más información sobre los valores admitidos, consulte el tema Cadenas con formato de fecha y hora personalizado en MSDN.
4. En `Encoding`, escriba la codificación del archivo que desea utilizar (por ejemplo, UTF-8). Para obtener más información sobre los valores admitidos, consulte el tema relacionado con la Clase de decodificación en MSDN.

> **Note**
>
> Utilice el nombre de codificación, no el nombre de visualización.

5. (Opcional) En `Filter`, escriba el prefijo de los nombres de registros. Deje en blanco este parámetro para monitorear todos los archivos. Para obtener más información sobre los valores admitidos, consulte el tema FileSystemWatcherFilter Propiedades en MSDN.
6. (Opcional) En `CultureName`, escriba la configuración regional en la que se registra la marca temporal. Si `CultureName` está en blanco, se utiliza de forma predeterminada la misma configuración regional que utiliza actualmente su instancia de Windows. Para obtener más información sobre los valores admitidos, consulte la columna `Language tag` en la tabla del tema relacionado con el Comportamiento del producto en MSDN.

> **Note**
>
> No se admiten los valores `div`, `div-MV`, `hu` y `hu-HU`.

7. (Opcional) En `TimeZoneKind`, escriba `Local` o `UTC`. Puede configurar este valor para proporcionar información sobre la zona horaria cuando no se incluye ninguna información sobre esta en la marca temporal del registro. Si este parámetro se deja en blanco y la marca horaria no incluye información sobre la zona horaria, CloudWatch Logs utilizará de forma predeterminada la zona horaria local. Se hace caso omiso a este parámetro si la marca temporal ya contiene información sobre la zona horaria.
8. (Opcional) En `LineCount`, escriba el número de líneas del encabezado para identificar el archivo de registros. Por ejemplo, los archivos de registros de IIS tienen encabezados prácticamente idénticos. Puede ingresar **5**, que leería las cinco primeras líneas del encabezado del archivo de registros para identificarlo. En los archivos de registros de IIS, la tercera línea es la marca de fecha y hora, pero la marca temporal no es siempre diferente entre los archivos de registros. Por este motivo, recomendamos que incluya al menos una línea de datos de registro reales para identificar de forma única el archivo de registros.

Paso 4: configurar el control de flujo

Cada tipo de datos debe tener un destino correspondiente en la sección `Flows`. Por ejemplo, para enviar el registro personalizado, el registro ETW y el registro del sistema a CloudWatch Logs, agréguelos `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` a la `Flows` sección.

> **Warning**
>
> Si se agrega un paso que no es válido, se bloquea el flujo. Por ejemplo, si agrega un paso de métrica de disco, pero la instancia no tiene ningún disco, se bloquean todos los pasos del flujo.

Puede enviar el mismo archivo de registros a varios destinos. Por ejemplo, para enviar el registro de la aplicación a dos destinos diferentes definidos en la sección `CloudWatchLogs`, agregue `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` a la sección `Flows`.

Para configurar el control de flujo

1. En el archivo `AWS.EC2.Windows.CloudWatch.json`, busque la sección `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. En `Flows`, agregue todos los tipos de datos que desea cargar (por ejemplo, `ApplicationEventLog`) y su destino (por ejemplo, `CloudWatchLogs`).

Acaba de editar el archivo JSON. Lo utilizará en un paso posterior.

Iniciar el agente

Para permitir que una instancia de Amazon EC2 que ejecute Windows Server 2012 o Windows Server 2008 envíe CloudWatch registros a Logs, utilice el servicio EC2Config (. `EC2Config.exe`) La instancia debe tener EC2Config 4.0 o posterior, y puede utilizar este procedimiento. Para obtener más información sobre el uso de una versión anterior de EC2Config, consulte Use EC2Config 3.x o una versión anterior para configurar en la Guía del usuario de CloudWatch Amazon EC2

CloudWatch Para realizar la configuración mediante EC2Config 4.x

1. Compruebe la codificación del archivo `AWS.EC2.Windows.CloudWatch.json` que editó anteriormente en este procedimiento. Solo se admite la codificación UTF-8 sin BOM. A continuación, guarde el archivo en la siguiente carpeta de la instancia con Windows Server 2008-2012 R2: `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`.
2. Inicie o reinicie el agente SSM (`AmazonSSMAgent.exe`) mediante el panel de control de los servicios de Windows o mediante el siguiente comando: PowerShell

```
PS C:\> Restart-Service AmazonSSMAgent
```

Una vez reiniciado, el agente SSM detecta el archivo de configuración y configura la instancia para su integración. CloudWatch Si cambia los parámetros y la configuración del archivo de configuración local, debe reiniciar el SSM Agent para que detecte los cambios. Para deshabilitar la CloudWatch integración en la instancia, cámbielo `IsEnabled` `false` y guarde los cambios en el archivo de configuración.

Inicio rápido: instale el agente CloudWatch de registros con AWS OpsWorks un Chef

Puede instalar el agente de CloudWatch registros y crear flujos de registros mediante AWS OpsWorks AndChef, que es una herramienta de automatización de sistemas e infraestructuras de nube de terceros. Chef utiliza “recetas”, que se escriben para instalar y configurar software en el ordenador, y “libros de recetas”, que son colecciones de recetas, para realizar sus tareas de configuración y distribución de políticas. Para obtener más información, consulte Chef.

Los siguientes ejemplos de recetas muestran cómo monitorear un archivo de registros en cada instancia EC2. Las recetas utilizan el nombre de pila como grupo de registros y el nombre de host de la instancia como el nombre del flujo de registro. Si desea monitorear varios archivos de registros, tendrá que ampliar las recetas para crear varios grupos y flujos de registro.

Paso 1: crear recetas personalizadas

Crea un repositorio para almacenar tus recetas. AWS OpsWorks es compatible con Git y Subversion, o puede almacenar un archivo en Amazon S3. La estructura de su repositorio de libros de recetas se describe en Cookbook Repositories (Repositorios de libros de recetas) en la Guía del usuario de AWS OpsWorks . Los ejemplos que aparecen a continuación suponen que el libro de recetas se llama `logs`. La receta install.rb instala el agente Logs. CloudWatch También puede descargar el ejemplo del libro de cocina (-Cookbooks.zip). CloudWatchLogs

Cree un archivo denominado metadata.rb que contiene el siguiente código:

```
#metadata.rb

name             'logs'
version          '0.0.1'
```

Cree el archivo de configuración de CloudWatch registros:

```
#config.rb
```

```
template "/tmp/cwlogs.cfg" do
  cookbook "logs"
  source "cwlogs.cfg.erb"
  owner "root"
  group "root"
  mode 0644
end
```

Descargue e instale el agente CloudWatch de registros:

```
# install.rb

directory "/opt/aws/cloudwatch" do
  recursive true
end

remote_file "/opt/aws/cloudwatch/awslogs-agent-setup.py" do
  source "https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
setup.py"
  mode "0755"
end

  execute "Install CloudWatch Logs agent" do
  command "/opt/aws/cloudwatch/awslogs-agent-setup.py -n -r region -c /tmp/cwlogs.cfg"
  not_if { system "pgrep -f aws-logs-agent-setup" }
end
```

> **Note**
>
> En el ejemplo anterior, reemplace la *región* con una de las siguientes: us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1 o sa-east-1.

Si se produce algún error en la instalación del agente, asegúrese de que el paquete `python-dev` está instalado. Si no lo está, utilice el siguiente comando e intente de nuevo la instalación del agente:

```
sudo apt-get -y install python-dev
```

Esta receta utiliza un archivo de plantilla cwlogs.cfg.erb que puede modificar para especificar distintos atributos como, por ejemplo, archivos que registrar. Para obtener más información sobre estos atributos, consulte CloudWatch Registra la referencia del agente.

```
[general]
# Path to the AWSLogs agent's state file. Agent uses this file to maintain
# client side state across its executions.
state_file = /var/awslogs/state/agent-state


## Each log file is defined in its own section. The section name doesn't
## matter as long as its unique within this file.
#
#[kern.log]
#
## Path of log file for the agent to monitor and upload.
#
#file = /var/log/kern.log
#
## Name of the destination log group.
#
#log_group_name = kern.log
#
## Name of the destination log stream.
#
#log_stream_name = {instance_id}
#
## Format specifier for timestamp parsing.
#
#datetime_format = %b %d %H:%M:%S
#
#

[<%= node[:opsworks][:stack][:name] %>]
datetime_format = [%Y-%m-%d %H:%M:%S]
log_group_name = <%= node[:opsworks][:stack][:name].gsub(' ','_') %>
file = <%= node[:cwlogs][:logfile] %>
log_stream_name = <%= node[:opsworks][:instance][:hostname] %>
```

La plantilla obtiene el nombre de la pila y el nombre del host haciendo referencia a los atributos correspondientes en la configuración de pila e implementación JSON. El atributo que especifica el

archivo que registrar se define en el archivo de atributos default.rb del libro de recetas cwlogs (logs/attributes/default.rb).

```
default[:cwlogs][:logfile] = '/var/log/aws/opsworks/opsworks-agent.statistics.log'
```

Paso 2: Crea una AWS OpsWorks pila

1. Abre la AWS OpsWorks consola en https://console.aws.amazon.com/opsworks/.
2. En el OpsWorks panel de control, selecciona Añadir pila para crear una AWS OpsWorks pila.
3. En la pantalla Add stack (Agregar pila), elija Chef 11 stack (Pila Chef 11).
4. En Stack name (Nombre de la pila), ingrese un nombre.
5. En Use custom Chef Cookbooks (Utilizar libros de cocina Chef personalizados), elija Yes (Sí).
6. En Repository type (Tipo de repositorio), seleccione el tipo de repositorio que utiliza. Si utiliza el ejemplo anterior, elija Http Archive.
7. En Repository URL (URL del repositorio), ingrese el repositorio donde almacenó el libro de recetas que creó en el paso anterior. Si utiliza el ejemplo anterior, ingrese **`https://s3.amazonaws.com/aws-cloudwatch/downloads/CloudWatchLogs-Cookbooks.zip`**.
8. Elija Add Stack (Agregar pila) para crear la pila.

Paso 3: ampliar el rol de IAM

Para usar CloudWatch los registros con sus AWS OpsWorks instancias, debe ampliar la función de IAM que utilizan sus instancias.

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, elija Policies (Políticas), Create Policy (Crear políticas).
3. En la página Create Policy (Crear políticas), en Create Your Own Policy (Crear su propia política), elija Select (Seleccionar). Para obtener más información sobre la creación de políticas personalizadas, consulte Políticas de IAM para Amazon EC2 en la Guía del usuario de Amazon EC2.
4. En la página Review Policy (Revisar políticas), en Policy Name (Nombre de la política), escriba un nombre para la política.
5. En Policy Document (Documento de la política), pegue la siguiente política:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

6. Elija Create Policy (Crear política).
7. En el panel de navegación, elija Roles y, a continuación, en el panel de contenido, en Nombre del rol, seleccione el nombre del rol de instancia utilizado por su pila. AWS OpsWorks Puede encontrar el utilizado por su pila en la configuración de pila (el valor predeterminado es `aws-opsworks-ec2-role`).

   > **Note**
   >
   > Elija el nombre del rol, no la casilla de verificación.

8. En la pestaña Permissions (Permisos), en Managed Policies (Políticas administradas), seleccione Attach Policy (Adjuntar política).
9. En la página Attach Policy (Adjuntar política), en el encabezado de la tabla (junto a Filter [Filtro] y Search [Buscar]), elija Policy Type (Tipo de política), Customer Managed Policies (Políticas administradas por el cliente).
10. En Customer Managed Policies (Políticas administradas por el cliente), seleccione la política de IAM que ha creado anteriormente y elija Attach Policy (Adjuntar política).

    Para obtener más información sobre los usuarios y políticas, consulte Usuarios y grupos de IAM y Administración de políticas de IAM en la Guía del usuario de IAM.

Paso 4: agregar una capa

1. Abre la AWS OpsWorks consola en https://console.aws.amazon.com/opsworks/.
2. En el panel de navegación, elija Layers (Capas).
3. En el panel de contenido, seleccione una capa y elija Add layer (Agregar capa).
4. En la OpsWorkspestaña, en Tipo de capa, elija Personalizado.
5. En los campos Name (Nombre) y Short name (Nombre breve), ingrese el nombre breve y largo de la capa y, a continuación, elija Add layer (Agregar capa).
6. En la pestaña Recetas, en Custom Chef Recipes, hay varios encabezados (Configurar, Configurar, Implementar, Desplegar y Cerrar) que corresponden a AWS OpsWorks los eventos del ciclo de vida. AWS OpsWorks desencadena estos eventos en estos puntos clave del ciclo de vida de la instancia, que ejecuta las recetas asociadas.

   > **Note**
   >
   > Si los encabezados anteriores no son visibles, en Custom Chef Recipes (Recetas Chef personalizadas), elija edit (editar).

7. Ingrese logs::config, logs::install junto a Setup (Configuración), elija + para agregarlo a la lista y, a continuación, elija Save (Guardar).

   AWS OpsWorks ejecuta esta receta en cada una de las nuevas instancias de esta capa, justo después del arranque de la instancia.

Paso 5: agregar una instancia

La capa solo controla cómo configurar las instancias. Ahora es necesario agregar algunas instancias a la capa e iniciarlas.

1. Abre la AWS OpsWorks consola en https://console.aws.amazon.com/opsworks/.
2. En el panel de navegación, elija Instances (Instancias) y, a continuación, bajo su capa, elija + Instance (+ Instancia).
3. Acepte la configuración predeterminada y elija Add Instance (Agregar instancia) para agregar la instancia a la capa.
4. En la columna Actions (Acciones) de la fila, haga clic en start (comenzar) para comenzar la instancia.

AWS OpsWorks lanza una nueva instancia de EC2 y configura los registros. CloudWatch El estado de la instancia cambia a en línea cuando está listo.

Paso 6: ver sus registros

Debería ver el grupo de registros y el flujo de registros recién creados en la CloudWatch consola después de que el agente haya estado ejecutándose durante unos instantes.

Para obtener más información, consulte Vea los datos de registro enviados a Logs CloudWatch .

Informe el estado del agente de CloudWatch Logs

Utilice el siguiente procedimiento para informar del estado del agente de CloudWatch Logs en su instancia EC2.

Para informar el estado del agente

1. Conéctese a la instancia EC2. Para obtener más información, consulte Connect to Your Instance en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre los problemas de conexión, consulte Solución de problemas de conexión a su instancia en la Guía del usuario de Amazon EC2
2. En el símbolo del sistema, escriba el siguiente comando:

   ```
   sudo service awslogs status
   ```

   Si ejecuta Amazon Linux 2, escriba el siguiente comando:

   ```
   sudo service awslogsd status
   ```
3. Compruebe si hay errores, advertencias o problemas relacionados con el agente de CloudWatch registros en el archivo /var/log/awslogs.log.

Inicie el agente CloudWatch de registros

Si el agente de CloudWatch registros de su instancia EC2 no se inició automáticamente después de la instalación, o si detuvo el agente, puede utilizar el siguiente procedimiento para iniciar el agente.

Para iniciar el agente de

1. Conéctese a la instancia EC2. Para obtener más información, consulte Connect to Your Instance en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre los problemas de conexión, consulte Solución de problemas de conexión a su instancia en la Guía del usuario de Amazon EC2.
2. En el símbolo del sistema, escriba el siguiente comando:

   ```
   sudo service awslogs start
   ```

   Si ejecuta Amazon Linux 2, escriba el siguiente comando:

   ```
   sudo service awslogsd start
   ```

Detenga el agente CloudWatch de registros

Utilice el siguiente procedimiento para detener el agente de CloudWatch registros en su instancia EC2.

Para detener el agente

1. Conéctese a la instancia EC2. Para obtener más información, consulte Connect to Your Instance en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre los problemas de conexión, consulte Solución de problemas de conexión a su instancia en la Guía del usuario de Amazon EC2.
2. En el símbolo del sistema, escriba el siguiente comando:

   ```
   sudo service awslogs stop
   ```

   Si ejecuta Amazon Linux 2, escriba el siguiente comando:

   ```
   sudo service awslogsd stop
   ```

Inicio rápido: utilícelo AWS CloudFormation para empezar a utilizar los registros CloudWatch

AWS CloudFormation le permite describir y aprovisionar sus AWS recursos en formato JSON. Las ventajas de este método incluyen la posibilidad de gestionar un conjunto de AWS recursos como una sola unidad y replicar fácilmente AWS los recursos en todas las regiones.

Al aprovisionar AWS el uso AWS CloudFormation, se crean plantillas que describen los AWS recursos que se van a utilizar. El siguiente ejemplo es un fragmento de plantilla que crea un grupo de registro y un filtro de métricas que cuenta las incidencias de 404 y envía este recuento al grupo de registro.

```
"WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},

"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "WebServerLogGroup"
        },
        "FilterPattern": "[ip, identity, user_id, timestamp, request, status_code =
 404, size, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    }
}
```

Se trata de un ejemplo básico. Puede configurar despliegues de CloudWatch Logs mucho más completos utilizando AWS CloudFormation. Para obtener más información sobre los ejemplos de plantillas, consulte Fragmentos de plantilla de Amazon CloudWatch Logs en la Guía del AWS

CloudFormation usuario. Para obtener más información de introducción, consulte Introducción a AWS CloudFormation en la Guía del usuario de AWS CloudFormation .

Uso CloudWatch de registros con un AWS SDK

AWS Los kits de desarrollo de software (SDK) están disponibles para muchos lenguajes de programación populares. Cada SDK proporciona una API, ejemplos de código y documentación que facilitan a los desarrolladores la creación de aplicaciones en su lenguaje preferido.

Documentación de SDK	Ejemplos de código
AWS SDK for C++	AWS SDK for C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK for Go	AWS SDK for Go ejemplos de código
AWS SDK for Java	AWS SDK for Java ejemplos de código
AWS SDK for JavaScript	AWS SDK for JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK for .NET	AWS SDK for .NET ejemplos de código
AWS SDK for PHP	AWS SDK for PHP ejemplos de código
AWS Tools for PowerShell	Herramientas para ejemplos PowerShell de código
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) ejemplos de código
AWS SDK for Ruby	AWS SDK for Ruby ejemplos de código
AWS SDK para Rust	AWS SDK para Rust ejemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP ejemplos de código
AWS SDK para Swift	AWS SDK para Swift ejemplos de código

Para ver ejemplos específicos de CloudWatch los registros, consulteEjemplos de código para CloudWatch registros que utilizan AWS SDK.

> **Ejemplo de disponibilidad**
>
> ¿No encuentra lo que necesita? Solicite un ejemplo de código a través del enlace de Enviar comentarios que se encuentra al final de esta página.

Análisis de datos de registro con CloudWatch Logs Insights

Con CloudWatch Logs Insights, puede buscar y analizar de forma interactiva sus datos de registro en Amazon CloudWatch Logs. Puede realizar consultas que le ayuden a responder de forma más eficaz a los problemas de funcionamiento. Si se produce un problema, puede utilizar CloudWatch Logs Insights para identificar las posibles causas y validar las soluciones implementadas.

CloudWatch Logs Insights incluye un lenguaje de consultas diseñado específicamente con unos pocos comandos simples pero potentes. CloudWatch Logs Insights proporciona ejemplos de consultas, descripciones de comandos, autocompletado de consultas y detección de campos de registro para ayudarle a empezar. Se incluyen ejemplos de consultas para varios tipos de registros de servicios de AWS .

CloudWatch Logs Insights descubre automáticamente los campos de AWS los registros de servicios como Amazon Route 53 y Amazon VPC AWS Lambda AWS CloudTrail, así como de cualquier aplicación o registro personalizado que emita eventos de registro como JSON.

Puede usar CloudWatch Logs Insights para buscar los datos de registro que se enviaron a CloudWatch Logs el 5 de noviembre de 2018 o después.

> **Important**
>
> CloudWatch Logs Insights no puede acceder a los eventos de registro con marcas de tiempo anteriores a la hora de creación del grupo de registros.

También puedes usar un lenguaje natural para crear consultas de CloudWatch Logs Insights. Para ello, pregunte o describa los datos que busca. Esta función asistida por IA genera una consulta en función de su solicitud y proporciona una line-by-line explicación de cómo funciona la consulta. Para obtener más información, consulte Usar un lenguaje natural para generar y actualizar las consultas de CloudWatch Logs Insights.

Si ha iniciado sesión en una cuenta configurada como una cuenta de supervisión en el marco de la observabilidad CloudWatch multicuenta, puede ejecutar consultas de CloudWatch Logs Insights en grupos de registros de las cuentas de origen vinculadas a esta cuenta de supervisión. Puede ejecutar una consulta que se ejecute en varios grupos de registro ubicados en diferentes cuentas. Para obtener más información, consulta la observabilidad CloudWatch entre cuentas.

Una única solicitud puede consultar hasta 50 grupos de registro. Las consultas expiran después de 60 minutos, si no se han completado. Los resultados de las consultas están disponibles durante 7 días.

Puede guardar las consultas que haya creado. Esto puede ayudarle a ejecutar consultas complejas cuando lo necesite, sin tener que volver a crearlas cada vez que desee ejecutarlas.

CloudWatch Las consultas de Logs Insights incurren en cargos en función de la cantidad de datos que se consulten. Para obtener más información, consulta los CloudWatch precios de Amazon.

> **Important**
>
> Si su equipo de seguridad de red no permite el uso de sockets web, actualmente no puede acceder a la parte de la CloudWatch consola de CloudWatch Logs Insights. Puede utilizar las funciones de consulta CloudWatch de Logs Insights mediante las API. Para obtener más información, consulta StartQueryla referencia de la API CloudWatch de Amazon Logs.

Contenido

Comandos compatibles con las clases de registro

Todos los comandos de consulta de CloudWatch Logs Insights son compatibles con los grupos de registros de la clase de registro estándar. Los grupos de registros de la clase de registro Infrequent Access admiten todos los comandos de consulta excepto `patterndiff`, `unmask`.

Primeros pasos: tutoriales sobre las consultas

En las siguientes secciones se incluyen ejemplos de tutoriales sobre consultas que le ayudarán a empezar a utilizar CloudWatch Logs Insights.

Temas

Tutorial: ejecutar y modificar una consulta de muestra

El siguiente tutorial le ayuda a empezar a utilizar CloudWatch Logs Insights. Ejecute una consulta de muestra y, a continuación, verá cómo modificarla y volverla a ejecutar.

Para ejecutar una consulta, ya debe tener los registros almacenados en CloudWatch Logs. Si ya utiliza CloudWatch los registros y ha configurado grupos de registros y flujos de registros, está listo para empezar. También es posible que ya tenga registros si utiliza servicios como AWS CloudTrail Amazon Route 53 o Amazon VPC y ha configurado los registros de esos servicios para que vayan a CloudWatch Logs. Para obtener más información sobre el envío de CloudWatch registros a Logs, consulteCómo empezar con CloudWatch los registros.

Las consultas en CloudWatch Logs Insights devuelven un conjunto de campos de eventos de registro o el resultado de una agregación matemática u otra operación realizada en los eventos de registro. Este tutorial muestra una consulta que devuelve una lista de eventos de registro.

Ejecutar una consulta de muestra

Para ejecutar una consulta de ejemplo CloudWatch de Logs Insights

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.

2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).

   En la página Información de registros, el editor de consultas contiene una consulta predeterminada que devuelve los 20 eventos de registro más recientes.

3. En el menú desplegable Select log group(s) (Seleccionar grupos de registros), elija uno o varios grupos de registros que va a consultar.

   Si se trata de una cuenta de supervisión en CloudWatch condiciones de observación multicuenta, puede seleccionar grupos de registros en las cuentas de origen, así como en la cuenta de supervisión. Una sola consulta puede ejecutarse en registros de diferentes cuentas a la vez.

   Puede filtrar los grupos de registro por nombre de grupo de registro, ID de cuenta o etiqueta de cuenta.

   Al seleccionar un grupo de registros en la clase de registro estándar, CloudWatch Logs Insights detecta automáticamente los campos de datos del grupo. Para ver estos campos detectados, seleccione el menú Fields (Campos) cerca de la parte superior derecha de la página.

   > **Note**
   >
   > Los campos descubiertos solo son compatibles con los grupos de registros de la clase de registro estándar. Para obtener más información sobre las clases de registro, consulteClases de registro.

4. (Opcional) Utilice el selector de tiempo para seleccionar el periodo de tiempo que desea consultar.

   Puede elegir entre intervalos de 5 a 30 minutos; intervalos de 1, 3 y 12 horas; o un marco temporal personalizado.

5. Elija Run (Ejecutar) para ver los resultados.

   En este tutorial, los resultados incluyen los 20 eventos de registro agregados más recientemente.

   CloudWatch Logs muestra un gráfico de barras con los eventos de registro del grupo de registros a lo largo del tiempo. Este gráfico de barras muestra no solo los eventos de la tabla, sino también la distribución de eventos del grupo de registros que coincide con la consulta y el intervalo de tiempo.

6. Para ver todos los campos de un evento de registro devuelto, elija el icono desplegable triangular a la izquierda del evento numerado.

Modificar la consulta de muestra

En este tutorial, debe modificar la consulta de muestra para mostrar los 50 eventos de registro más recientes.

Si aún no ha ejecutado el tutorial anterior, hágalo ahora. Este tutorial comienza donde finaliza el tutorial anterior.

> **Note**
>
> Algunas consultas de ejemplo que se proporcionan con CloudWatch Logs Insights utilizan `tail` comandos `head` o comandos en lugar de`limit`. Estos comandos están obsoletos y se han sustituido por `limit`. Utilice `limit` en lugar de `head` o `tail` en todas las consultas que escriba.

Para modificar la consulta de ejemplo CloudWatch de Logs Insights

1. En el editor de consultas, cambie 20 a 50 y, a continuación, elija Ejecutar.

   Aparecen los resultados de la nueva consulta. Suponiendo que haya suficientes datos en el grupo de registros en el intervalo de tiempo predeterminado, ahora hay 50 eventos de registro en la lista.
2. (Opcional) Puede guardar las consultas que haya creado. Para guardar esta consulta, elija Save (Guardar). Para obtener más información, consulte Guarde y vuelva a ejecutar las consultas de CloudWatch Logs Insights.

Agregar un comando de filtro a la consulta de muestra

En este tutorial se muestra cómo realizar un cambio más potente en la consulta en el editor de consultas. En este tutorial, se filtran los resultados de la consulta anterior en función de un campo de los eventos de registro recuperados.

Si aún no ha ejecutado los tutoriales anteriores, hágalo ahora. Este tutorial comienza donde finaliza el tutorial anterior.

Para añadir un comando de filtro a la consulta anterior

1. Decida un campo que filtrar. Para ver los campos más comunes que CloudWatch Logs ha detectado en los eventos de registro contenidos en los grupos de registros seleccionados en los últimos 15 minutos y el porcentaje de esos eventos de registro en los que aparece cada campo, seleccione Campos en la parte derecha de la página.

   Para ver los campos contenidos en un evento de registro determinado, elija el icono que aparece a la izquierda de dicha fila.

   El campo awsRegion podría aparecer en su evento de registro, en función de los eventos que se encuentren en sus registros. En el resto de este tutorial, utilizaremos awsRegion como campo de filtro, pero puede utilizar un campo diferente si ese campo no está disponible.
2. En el editor de consultas, coloque el cursor después de 50 y pulse Intro.
3. En la nueva línea, introduzca | (la barra vertical) y un espacio. Los comandos de una consulta de CloudWatch Logs Insights deben estar separados por una barra vertical.
4. Escriba **`filter awsRegion="us-east-1"`**.
5. Elija Ejecutar.

   La consulta se ejecuta de nuevo, y ahora muestra el 50 resultados más recientes que coinciden con el nuevo filtro.

   Si filtra en otro campo diferente y recibe un resultado erróneo, es posible que sea necesario aplicar escape al nombre de campo. Si el nombre de campo incluye caracteres no alfanuméricos, debe volver a poner acentos graves (') antes y después del nombre de campo: por ejemplo, **`` `error-code`="102" ``**).

   Debe utilizar los caracteres graves para los nombres de campo que contengan caracteres no alfanuméricos, pero no para los valores. Los valores siempre van entre comillas (").

CloudWatch Logs Insights incluye potentes capacidades de consulta, que incluyen varios comandos y soporte para expresiones regulares y operaciones matemáticas y estadísticas. Para obtener más información, consulte CloudWatch Sintaxis de consultas de Logs Insights.

Tutorial: ejecutar una consulta con una función de agregación

Puede utilizar las funciones de agregación con el comando `stats` y como argumentos para otras funciones. En este tutorial, ejecuta un comando de consulta que cuenta el número de eventos de

registro que contienen un campo especificado. El comando de consulta devuelve un recuento total agrupado según el valor o los valores del campo especificados. Para obtener más información sobre las funciones de agregación, consulte Operaciones y funciones compatibles en la Guía del usuario de Amazon CloudWatch Logs.

Para ejecutar una consulta con una función de agregación

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. En el menú desplegable Select log group(s) (Seleccionar grupos de registros), elija uno o varios grupos de registros para consultar.

   Si se trata de una cuenta de supervisión en CloudWatch condiciones de observación multicuenta, puede seleccionar grupos de registros en las cuentas de origen, así como en la cuenta de supervisión. Una sola consulta puede ejecutarse en registros de diferentes cuentas a la vez.

   Puede filtrar los grupos de registro por nombre de grupo de registro, ID de cuenta o etiqueta de cuenta.

   Al seleccionar un grupo de CloudWatch registros, Logs Insights detecta automáticamente los campos de datos del grupo de registros si se trata de un grupo de registros de clase estándar. Para ver estos campos detectados, seleccione el menú Fields (Campos) cerca de la parte superior derecha de la página.
4. Elimine la consulta predeterminada en el editor de consultas e ingrese el siguiente comando:

   ```
   stats count(*) by fieldName
   ```
5. Reemplace *fieldName* por un campo descubierto desde el menú Fields (Campos).

   El menú Campos se encuentra en la parte superior derecha de la página y muestra todos los campos detectados que CloudWatch Logs Insights detecta en su grupo de registros.
6. Elija Run (Ejecutar) para ver los resultados de la consulta.

   Los resultados de la consulta muestran el número de registros del grupo de registros que coinciden con el comando de consulta y el recuento total agrupado según el valor o los valores del campo especificados.

Tutorial: Ejecutar una consulta que produce una visualización agrupada por campos de registro

Si ejecuta una consulta que utiliza la función `stats` para agrupar los resultados devueltos según los valores de uno o varios campos de las entradas de registro, se pueden ver los resultados como un gráfico de barras, un gráfico circular, un gráfico de líneas o un gráfico de áreas apiladas. De este modo, podrá consultar de un modo más eficaz las tendencias de los registros.

Para ejecutar una consulta para visualización

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. En el menú desplegable Select log group(s) (Seleccionar grupos de registros), elija uno o varios grupos de registros para consultar.

   Si se trata de una cuenta de supervisión en CloudWatch condiciones de observación multicuenta, puede seleccionar grupos de registros en las cuentas de origen, así como en la cuenta de supervisión. Una sola consulta puede ejecutarse en registros de diferentes cuentas a la vez.

   Puede filtrar los grupos de registro por nombre de grupo de registro, ID de cuenta o etiqueta de cuenta.
4. En el editor de consultas, elimine el contenido actual y escriba la siguiente función `stats`. Después seleccione Run query (Ejecutar consulta).

   ```
   stats count(*) by @logStream
       | limit 100
   ```

   Los resultados muestran el número de eventos del grupo de registros por cada secuencia de registro. Los resultados tienen un límite de 100 filas.
5. Elija la pestaña Visualization (Visualización).
6. Seleccione la flecha situada junto a Line (Línea) y, a continuación, elija Bar (Barra).

   Aparece el gráfico de barras, con una barra por cada secuencia de registro del grupo de registros.

Tutorial: Ejecutar una consulta que produce una visualización de serie temporal

Cuando se ejecuta una consulta que utiliza la función `bin()` para agrupar los resultados devueltos por un periodo de tiempo, puede ver los resultados como un gráfico de líneas, un gráfico de áreas apiladas, un gráfico circular o un gráfico de barras. De este modo, podrá consultar de un modo más eficaz las tendencias de los eventos de registro a lo largo del tiempo.

Para ejecutar una consulta para visualización

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. En el menú desplegable Select log group(s) (Seleccionar grupos de registros), elija uno o varios grupos de registros para consultar.

   Si se trata de una cuenta de supervisión en CloudWatch condiciones de observación multicuenta, puede seleccionar grupos de registros en las cuentas de origen, así como en la cuenta de supervisión. Una sola consulta puede ejecutarse en registros de diferentes cuentas a la vez.

   Puede filtrar los grupos de registro por nombre de grupo de registro, ID de cuenta o etiqueta de cuenta.
4. En el editor de consultas, elimine el contenido actual y escriba la siguiente función `stats`. Después seleccione Run query (Ejecutar consulta).

   ```
   stats count(*) by bin(30s)
   ```

   Los resultados muestran el número de eventos de registro del grupo de CloudWatch registros recibidos por Logs durante cada período de 30 segundos.
5. Elija la pestaña Visualization (Visualización).

   Los resultados se muestran como un gráfico de líneas. Para cambiar a un gráfico de áreas apiladas o a un gráfico de barras, elija la flecha situada junto a Line (Línea) en la parte superior derecha del gráfico.

Registros y campos detectados compatibles

CloudWatch Logs Insights admite distintos tipos de registro. Por cada registro que se envía a un grupo de CloudWatch registros de clase estándar Amazon Logs, CloudWatch Logs Insights genera automáticamente cinco campos de sistema:

- `@message` contiene el evento de registro sin analizar ni procesar. Es el equivalente al `message` campo de InputLogevent.
- `@timestamp` contiene la marca temporal del evento incluida en el campo `timestamp` del evento de registro. Es el equivalente al `timestamp` campo de InputLogevent.
- `@ingestionTime`contiene la hora en que CloudWatch Logs recibió el evento de registro.
- `@logStream` contiene el nombre del flujo de registros al que se añadió el evento de registro. Las transmisiones de registro agrupan los registros a través del mismo proceso que los generó.
- `@log` es un identificador de grupo de registro con el formato *`account-id`*`:`*`log-group-name`*. Puede ser útil en consultas de varios grupos de registro para identificar a qué grupo de registro pertenece un evento determinado.

> **Note**
>
> La detección de campos solo se admite para los grupos de registros de la clase de registro estándar. Para obtener más información sobre las clases de registro, consulteClases de registro.

CloudWatch Logs Insights inserta el símbolo @ al principio de los campos que genera.

En muchos tipos de CloudWatch registros, Logs también descubre automáticamente los campos de registro contenidos en los registros. Estos campos de detección automática se muestran en la siguiente tabla.

Para otros tipos de registros con campos que CloudWatch Logs Insights no descubre automáticamente, puede usar el `parse` comando para extraer y crear campos extraídos para usarlos en esa consulta. Para obtener más información, consulte CloudWatch Sintaxis de consultas de Logs Insights.

Si el nombre de un campo de registro descubierto comienza por el @ carácter, CloudWatch Logs Insights lo muestra con un @ elemento adicional añadido al principio. Por ejemplo, si un nombre de campo de registro es `@example.com`, este nombre de campo se muestra como `@@example.com`.

Tipo de registro	Campos de registro detectados
Registros de flujo de Amazon VPC	`@timestamp` , `@logStream` , `@message`, `accountId` , `endTime`, `interfaceId` , `logStatus` , `startTime` , `version`, `action`, `bytes`, `dstAddr`, `dstPort`, `packets`, `protocol`, `srcAddr`, `srcPort`
Registros de Route 53	`@timestamp` , `@logStream` , `@message`, `edgeLocation` , `ednsClientSubnet` , `hostZoneId` , `protocol`, `queryName` , `queryTimestamp` , `queryType` , `resolverIp` , `responseCode` , `version`
Registros de Lambda	`@timestamp` , `@logStream` , `@message`, `@requestId` , `@duration`, `@billedDuration` , `@type`, `@maxMemoryUsed` , `@memorySize` Si una línea de registro de Lambda contiene un ID de seguimiento de X-Ray, también incluye los siguientes campos: `@xrayTraceId` y `@xraySegmentId` . CloudWatch Logs Insights descubre automáticamente los campos de registro en los registros Lambda, pero solo para el primer fragmento de JSON incrustado en cada evento de registro. Si un evento de registro de Lambda contiene varios fragmentos JSON, puede analizar y extraer los campos de registro con el comando **`parse`**. Para obtener más información, consulte Campos de registros JSON.
CloudTrail registros Registros en formato JSON	Para obtener más información, consulte Campos de registros JSON.
Otros tipos de registros	`@timestamp` , `@ingestionTime` , `@logStream` , `@message`, `@log`.

Campos de registros JSON

Con CloudWatch Logs Insights, se utiliza la notación de puntos para representar los campos JSON. Esta sección contiene un ejemplo de evento JSON y fragmento de código que muestra cómo acceder a los campos JSON mediante la notación de puntos.

Ejemplo de evento JSON


```
{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123",
                    "currentState": {
                        "code": 0,
                        "name": "pending"
                    },
                    "previousState": {
```

```
                        "code": 80,
                        "name": "stopped"
                    }
                }
            ]
        }
    }
}
```

El evento JSON de ejemplo contiene un objeto denominado `userIdentity`. `userIdentity`contiene un campo que se llama `type`. Para representar el valor de `type` usando una notación de puntos, use `userIdentity.type`.

El evento JSON de ejemplo contiene matrices que se aplanan en listas de nombres y valores de campo anidados. Para representar el valor de `instanceId` para el primer elemento de `requestParameters.instancesSet`, utilice `requestParameters.instancesSet.items.0.instanceId`. El número `0` que se coloca antes del campo `instanceID` hace referencia a la posición de los valores para el campo `items`. El siguiente ejemplo contiene un fragmento de código que muestra cómo puede acceder a los campos JSON anidados en un evento de registro JSON.

Ejemplo: consulta

```
fields @timestamp, @message
| filter requestParameters.instancesSet.items.0.instanceId="i-abcde123"
| sort @timestamp desc
```

El fragmento de código muestra una consulta que utiliza la notación de puntos con el comando `filter` para acceder al valor del campo JSON anidado `instanceId`. La consulta se filtra en los mensajes donde el valor de `instanceId` es igual a `"i-abcde123"` y devuelve todos los eventos de registro que contienen el valor especificado.

> **Note**
>
> CloudWatch Logs Insights puede extraer un máximo de 200 campos de eventos de registro de un registro JSON. Para campos adicionales que no se extraen, puede utilizar el comando `parse` para extraer estos campos desde el evento de registro sin analizar en el campo de mensaje. Para obtener más información sobre el `parse` comando, consulte Sintaxis de consultas en la Guía del CloudWatch usuario de Amazon.

CloudWatch Sintaxis de consultas de Logs Insights

Con CloudWatch Logs Insights, utiliza un lenguaje de consulta para consultar sus grupos de registros. La sintaxis de la consulta admite diferentes funciones y operaciones, incluidas, entre otras, funciones generales, operaciones aritméticas y de comparación y expresiones regulares.

Para crear consultas que contengan varios comandos, separe los comandos con el carácter de barra vertical (|).

Para crear consultas que contengan comentarios, defina los comentarios con el carácter numeral (#).

> **Note**
>
> CloudWatch Logs Insights descubre automáticamente los campos de diferentes tipos de registro y genera campos que comienzan con el carácter @. Para obtener más información sobre estos campos, consulta Registros compatibles y campos detectados en la Guía del CloudWatch usuario de Amazon.

En la tabla siguiente se describe cada comando de forma breve. A continuación, hay una descripción más completa de cada comando con ejemplos.

> **Note**
>
> Todos los comandos de consulta de CloudWatch Logs Insights son compatibles con los grupos de registros de la clase de registro estándar. Los grupos de registros de la clase de registro Infrequent Access admiten todos los comandos de consulta excepto `patterndiff`, `unmask`.

`display`	Muestra un campo o campos específicos en los resultados de la consulta.
`fields`	Muestra campos específicos en los resultados de la consulta y admite funciones y operaciones que puede utilizar para modificar los valores de los campos y crear nuevos campos para utilizarlos en la consulta.

filter	Filtra la consulta para devolver solo los eventos de registro que coincidan con una o más condiciones.
pattern	Agrupa automáticamente los datos de registro en patrones. Un patrón es una estructura de texto compartida que se repite en los campos de registro. CloudWatch Logs Insights le proporciona formas de analizar los patrones encontrados en sus eventos de registro. Para obtener más información, consulte Análisis de patrones.
diff	Compara los eventos de registro encontrados en el período de tiempo solicitado con los eventos de registro de un período de tiempo anterior de igual duración, de modo que pueda buscar tendencias y averiguar si algunos eventos de registro son nuevos.
parse	Extrae los datos de un campo de registro para crear un campo extraído que pueda procesar en su consulta. **parse** admite tanto el modo glob con caracteres comodín como con expresiones regulares.
sort	Muestra los eventos de registro devueltos en orden ascendente (`asc`) o descendente (`desc`).
stats	Calcula estadísticas totales mediante valores en los campos de registro.
limit	Especifica un número máximo de eventos de registro que desea que devuelva la consulta. Es ideal con **sort** para devolver los “20 primeros” resultados o los “20 últimos” resultados.
dedup	Elimina los resultados duplicados en función de valores específicos en los campos que especifique.
unmask	Muestra todo el contenido de un evento de registro que tiene parte del contenido enmascarado debido a una política de protección de datos. Para obtener más información sobre la protección de datos en grupos de registro, consulte Ayude a proteger los datos de registro confidenciales con el enmascaramiento.

Otras operaciones y funciones	CloudWatch Logs Insights también admite numerosas funciones y operaciones de comparación, aritmética, de fecha y hora, numéricas, de cadenas, de direcciones IP y generales.

En las siguientes secciones se proporcionan más detalles sobre los comandos de consulta de CloudWatch Logs Insights.

Temas

display

Use `display` para mostrar un campo o campos específicos en los resultados de la consulta.

El comando `display` muestra solo los campos que especifique. Si la consulta contiene varios comandos `display`, los resultados de la consulta muestran solo el campo o los campos especificados en el comando final `display`.

Ejemplo: mostrar un campo

El fragmento de código muestra un ejemplo de una consulta que usa el comando parse para extraer datos de `@message` con el objetivo de crear los campos extraídos `loggingType` y `loggingMessage`. La consulta devuelve todos los eventos de registro en los que los valores de `loggingType` son ERROR. `display` muestra solo los valores de `loggingMessage` en los resultados de la consulta.

```
fields @message
| parse @message "[*] *" as loggingType, loggingMessage
| filter loggingType = "ERROR"
| display loggingMessage
```

> **Tip**
>
> Use `display` solo una vez en una consulta. Si usa `display` más de una vez en una consulta, los resultados de la consulta muestran los campos especificados en la última aparición del comando `display` que se está utilizando.

fields

Use `fields` para mostrar campos específicos en los resultados de la consulta.

Si su consulta tiene varios comandos `fields` y no incluye un comando `display`, los resultados mostrarán todos los campos que se especifican en los comandos `fields`.

Ejemplo: mostrar campos específicos

El ejemplo siguiente muestra una consulta que devuelve 20 eventos de registro y los organiza en orden descendente. Los valores para `@timestamp` y `@message` se muestran en los resultados de la consulta.

```
fields @timestamp, @message
| sort @timestamp desc
| limit 20
```

Use `fields` en lugar de `display` cuando quiera utilizar las diferentes funciones y operaciones que admite `fields` para modificar los valores de los campos y crear nuevos campos que se puedan usar en las consultas.

Puede utilizar el comando `fields` con la palabra clave as para crear campos extraídos que utilicen campos y funciones en los eventos de registro. Por ejemplo, `fields ispresent as isRes` crea un campo extraído denominado `isRes`, y ese campo extraído se puede utilizar en el resto de la consulta.

filter

Use `filter` para obtener eventos de registro que coincidan con una o más condiciones.

Ejemplo: filtrar eventos de registro con una condición

El fragmento de código muestra un ejemplo de una consulta que devuelve todos los eventos de registro en los que el valor de `range` es mayor que 3000. La consulta limita los resultados a 20 eventos de registro y los ordena por `@timestamp` y en orden descendente.

```
fields @timestamp, @message
| filter (range>3000)
| sort @timestamp desc
| limit 20
```

Ejemplo: filtrar eventos de registro con más de una condición

Puede usar las palabras clave `and` y `or` para combinar más de una condición.

El fragmento de código muestra un ejemplo de una consulta que devuelve todos los eventos de registro en los que el valor de `range` es mayor que 3000 y el valor de `accountId` es igual que 123456789012. La consulta limita los resultados a 20 eventos de registro y los ordena por `@timestamp` y en orden descendente.

```
fields @timestamp, @message
| filter (range>3000 and accountId=123456789012)
| sort @timestamp desc
| limit 20
```

Coincidencias y expresiones regulares en el comando de filtro

El comando de filtro admite el uso de expresiones regulares. Puede utilizar los siguientes operadores de comparación (=, `!=`, <, <=, >, >=) y operadores booleanos (`and`, `or` y `not`).

Puede usar la palabra clave `in` para probar si hay suscripción configurada y verificar si hay elementos en una matriz. Para comprobar los elementos de una matriz, coloque los elementos

después de `in`. Puede utilizar los operadores booleanos `not`, con `in`. Puede crear consultas que utilicen `in` para devolver eventos de registro en los que los campos son coincidencias de cadenas. Los campos deben ser cadenas completas. Por ejemplo, el siguiente fragmento de código muestra una consulta que utiliza `in` para devolver eventos de registro donde el campo `logGroup` es la cadena completa `example_group`.

```
fields @timestamp, @message
| filter logGroup in ["example_group"]
```

Puede usar las frases de palabras clave `like` y `not like` para que coincidan con las subcadenas. Puede utilizar el operador de expresión regular =~ para que coincidan con las subcadenas. Para hacer coincidir una subcadena con `like` y `not like`, encierre la subcadena que desea buscar entre comillas dobles o simples. Puede utilizar patrones de expresión regular con `like` y `not like`. Para hacer coincidir una subcadena con el operador de expresiones regulares, encierre la subcadena que desea buscar entre barras diagonales. Los siguientes ejemplos contienen fragmentos de código que muestran cómo se pueden hacer coincidir las subcadenas mediante el comando `filter`.

Ejemplos: hacer coincidir subcadenas

Los siguientes ejemplos devuelven los eventos de registro en que `f1` contiene la palabra Exception (Excepción). Los tres ejemplos distinguen entre mayúsculas y minúsculas.

El primer ejemplo hace coincidir una subcadena con `like`.

```
fields f1, f2, f3
| filter f1 like "Exception"
```

El segundo ejemplo hace coincidir una subcadena con `like` y un patrón de expresiones regulares.

```
fields f1, f2, f3
| filter f1 like /Exception/
```

El tercer ejemplo hace coincidir una subcadena con una expresión regular.

```
fields f1, f2, f3
| filter f1 =~ /Exception/
```

Ejemplo: hacer coincidir subcadenas con comodines

Puede utilizar el símbolo de punto (`.`) como comodín en expresiones regulares para que coincidan con las subcadenas. En el siguiente ejemplo, la consulta devuelve coincidencias en las que el valor de `f1` comienza con la cadena `ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog./
```

Puede colocar un punto antes del símbolo de punto (`.*`) para crear un cuantificador expansivo que devuelva tantas coincidencias como sea posible. Por ejemplo, la siguiente consulta devuelve coincidencias en las que el valor de `f1` no solo comienza con la cadena `ServiceLog`, sino que incluye además la cadena `ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog.*/
```

Las posibles coincidencias pueden tener el siguiente formato:

- `ServiceLogSampleApiLogGroup`
- `SampleApiLogGroupServiceLog`

Ejemplo: excluir subcadenas de coincidencias

En el siguiente ejemplo, se muestra una consulta que devuelve eventos de registro donde `f1` no contiene la palabra Exception (Excepción). El ejemplo distingue mayúsculas de minúsculas.

```
fields f1, f2, f3
| filter f1 not like "Exception"
```

Ejemplo: hacer coincidir subcadenas con patrones que no distinguen mayúsculas de minúsculas

Puede hacer coincidir las subcadenas que no distinguen mayúsculas de minúsculas con `like` y expresiones regulares. Coloque el siguiente parámetro (?i) antes de la subcadena que desea buscar. En el siguiente ejemplo, se muestra una consulta que devuelve eventos de registro donde `f1` contiene la palabra Exception o exception (Excepción o excepción).

```
fields f1, f2, f3
| filter f1 like /(?i)Exception/
```

pattern

Use `pattern` para agrupar automáticamente los datos de registro en patrones.

Un patrón es una estructura de texto compartida que se repite entre los campos de registro. Puede utilizarlos `pattern` para mostrar las tendencias emergentes, supervisar los errores conocidos e identificar las líneas de registro que se producen con frecuencia o son costosas. CloudWatch Logs Insights también proporciona una experiencia de consola que puede utilizar para buscar y analizar más a fondo los patrones de sus eventos de registro. Para obtener más información, consulte Análisis de patrones.

Como el `pattern` comando identifica automáticamente los patrones comunes, puedes usarlo como punto de partida para buscar y analizar tus registros. También puede combinar `pattern` con los comandos `filter`, `parse` o `sort` para identificar patrones en consultas más precisas.

Entrada del comando pattern

El comando `pattern` espera una de las siguientes entradas: el campo `@message`, un campo extraído creado mediante el comando `parse` o una cadena manipulada mediante una o más funciones de cadena.

Salida del comando pattern

El comando `pattern` produce la salida siguiente:

- `@pattern`: una estructura de texto compartida que se repite entre los campos de eventos de registro. Los campos que varían dentro de un patrón, como un ID de solicitud o una marca de tiempo, se representan con `<*>`. Por ejemplo, `[INFO] Request time: <*> ms` es una salida potencial para el mensaje de registro `[INFO] Request time: 327 ms`.
- `@ratio`: proporción de eventos de registro de un periodo de tiempo seleccionado y los grupos de registro especificados que coinciden con un patrón identificado. Por ejemplo, si la mitad de los eventos de registro de los grupos de registro y el periodo de tiempo seleccionados coinciden con el patrón, `@ratio` devuelve `0.50`
- `@sampleCount`: recuento del número de eventos de registro de un periodo de tiempo seleccionado y los grupos de registro especificados que coinciden con un patrón identificado.
- `@severityLabel`: gravedad o nivel del registro, que indica el tipo de información que contiene un registro. Por ejemplo, `Error`, `Warning`, `Info` o `Debug`.

Ejemplos

El siguiente comando identifica los registros con estructuras similares en los grupos de registro especificados durante el intervalo de tiempo seleccionado y los agrupa por patrón y recuento

```
pattern @message
```

El comando `pattern` se puede utilizar en combinación con el comando `filter`

```
filter @message like /ERROR/
| pattern @message
```

El comando `pattern` se puede utilizar con los comandos `parse` y `sort`

```
filter @message like /ERROR/
| parse @message 'Failed to do: *' as cause
| pattern cause
| sort @sampleCount asc
```

diferencia

Compara los eventos de registro encontrados en el período de tiempo solicitado con los eventos de registro de un período de tiempo anterior de igual duración. De esta forma, puede buscar tendencias y determinar si los eventos de registro específicos son nuevos.

Añada un modificador al `diff` comando para especificar el período de tiempo con el que desea comparar:

- `diff`compara los eventos de registro en el rango de tiempo actualmente seleccionado con los eventos de registro del rango de tiempo inmediatamente anterior.
- `diff previousDay`compara los eventos de registro en el intervalo de tiempo actualmente seleccionado con los eventos de registro de la misma hora del día anterior.
- `diff previousWeek`compara los eventos de registro en el intervalo de tiempo actualmente seleccionado con los eventos de registro de la semana anterior a la misma hora.
- `diff previousMonth`compara los eventos de registro en el intervalo de tiempo actualmente seleccionado con los eventos de registro de la misma época del mes anterior.

Para obtener más información, consulte Compara (diferencia) con intervalos de tiempo anteriores.

parse

Use `parse` para extraer datos de un campo de registro y crear un campo extraído que pueda procesar en su consulta. **`parse`** admite tanto el modo glob con caracteres comodín como expresiones regulares. Para obtener información sobre la sintaxis de las expresiones regulares, consulteSintaxis de expresiones regulares (regex) compatibles.

Puede analizar los campos JSON anidados con una expresión regular.

Ejemplo: análisis de un campo JSON anidado

El fragmento de código muestra cómo analizar un evento de registro JSON que se ha aplanado durante la incorporación.

```
{'fieldsA': 'logs', 'fieldsB': [{'fA': 'a1'}, {'fA': 'a2'}]}
```

El fragmento de código muestra una consulta con una expresión regular que extrae los valores de `fieldsA` y `fieldsB` con el objetivo de crear los campos extraídos `fld` y `array`.

```
parse @message "'fieldsA': '*', 'fieldsB': ['*']" as fld, array
```

Grupos de captura con nombres

Cuando se usa **`parse`** con una expresión regular, puede usar grupos de captura con nombre para capturar un patrón en un campo. La sintaxis es `parse @message (?<Name>pattern).`

El siguiente ejemplo usa un grupo de captura en un registro de flujo de la VPC para extraer el ENI en un campo denominado `NetworkInterface`.

```
parse @message /(?<NetworkInterface>eni-.*?) / display @timestamp, NetworkInterface
```

> **Note**
>
> Los eventos de registro JSON se aplanan durante la ingesta. Actualmente, no se admite el análisis de campos JSON anidados con una expresión global. Solo puede analizar eventos de registro JSON que no incluyan más de 200 campos de eventos de registro. Cuando analice los campos JSON anidados, debe dar formato a la expresión regular de la consulta para que coincida con el formato de su evento de registro JSON.

Ejemplos del comando para analizar

Utilice una expresión glob para extraer los campos **`@user`**, **`@method`** y **`@latency`** del campo de registro **`@message`** y devolver la latencia promedio para cada combinación única de **`@method`** y **`@user`**.

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

Utilice una expresión regular para extraer los campos **`@user2`**, **`@method2`** y **`@latency2`** del campo de registro **`@message`** y devolver la latencia promedio para cada combinación única de **`@method2`** y **`@user2`**.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Extrae los campos **`loggingTime`**, **`loggingType`** y **`loggingMessage`**, filtra hasta los eventos de registro que contienen cadenas **`ERROR`** o **`INFO`** y, a continuación, muestra solo los campos **`loggingMessage`** y **`loggingType`** para los eventos que contienen una cadena **`ERROR`**.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

sort

Use `sort` para mostrar eventos de registro en orden ascendente (`asc`) o descendente (`desc`) por un campo especificado. Puede usarlo con el comando `limit` para crear consultas de los “primeros N” o los “últimos N”.

El algoritmo de clasificación es una versión actualizada de la clasificación natural. Si ordena en orden ascendente, se utiliza la siguiente lógica.

- Todos los valores no numéricos aparecen antes que todos los valores numéricos. Los valores numéricos son valores que incluyen únicamente números, no una mezcla de números y otros caracteres.

- Para los valores que no son numéricos, el algoritmo agrupa los caracteres numéricos consecutivos y los caracteres alfabéticos consecutivos en fragmentos separados para compararlos. Ordena las partes no numéricas por sus valores Unicode y ordena las partes numéricas primero por su longitud y después por su valor numérico.

Para obtener más información sobre el orden Unicode, consulte Lista de caracteres Unicode.

Por ejemplo, lo siguiente es el resultado de una ordenación en orden ascendente.

```
!: >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> sorted by unicode order
#
*%04
0# >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Alphanumeric starting with numbers
5A
111A   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  Starts with more digits than 5A, so it sorted
 to be later than 5A
2345_
@ >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> 2345 is compared with @ in the unicode order,
@_
A >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Values starting with letters
A9876fghj
a12345hfh
0 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Number values
01
1
2
3
```

Si ordena en orden descendente, los resultados de la ordenación son los inversos.

Por ejemplo, la siguiente consulta para los registros de flujo de Amazon VPC busca las 15 transferencias principales de paquetes entre hosts.

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

stats

Utilice `stats` para crear visualizaciones de los datos de registro, como gráficos de barras, gráficos de líneas y gráficos de áreas apiladas. Esto le ayuda a identificar de manera más eficiente los

patrones en los datos de registro. CloudWatch Logs Insights genera visualizaciones para las consultas que utilizan la `stats` función y una o más funciones de agregación.

Por ejemplo, la siguiente consulta en un grupo de registro de Route 53 devuelve visualizaciones que muestran la distribución de los registros de Route 53 por hora, por tipo de consulta.

```
stats count(*) by queryType, bin(1h)
```

Todas estas consultas pueden generar gráficos de barras. Si la consulta utiliza la función `bin()` para agrupar los datos en función de un mismo campo a lo largo del tiempo, también puede ver gráficos de líneas y gráficos de áreas apiladas.

La función `bin` admite las siguientes unidades de tiempo y abreviaturas. Para todas las unidades y abreviaturas que incluyan más de un carácter, se admite agregar s para pluralizar. Así que tanto `hr` como `hrs` funcionan para especificar las horas.

- `millisecond` `ms` `msec`
- `second` `s` `sec`
- `minute` `m` `min`
- `hour` `h` `hr`
- `day` `d`
- `week` `w`
- `month` `mo` `mon`
- `quarter` `q` `qtr`
- `year` `y` `yr`

Temas

- Visualización de datos de series temporales
- Visualización de datos de registro agrupados por campos
- Utilice varios comandos de estadísticas en una sola consulta
- Funciones para usar con estadísticas

Visualización de datos de series temporales

Las visualizaciones de series temporales funcionan con las consultas que tienen las siguientes características:

- La consulta contiene una o varias funciones de agregación. Para obtener más información, consulte Aggregation Functions in the Stats Command.
- La consulta utiliza la función `bin()` para agrupar los datos por un campo.

Estas consultas pueden generar gráficos de líneas, de áreas apiladas, de barras y circulares.

Ejemplos

Para ver un tutorial completo, consulte the section called “Tutorial: Ejecutar una consulta que produce una visualización de serie temporal”.

Aquí hay más consultas de ejemplo que funcionan para la visualización de series temporales.

La siguiente consulta genera una visualización de los valores medios del campo `myfield1`, con un punto de datos creado cada cinco minutos. Cada punto de datos es la agregación de las medias de los valores `myfield1` de los registros de los últimos cinco minutos.

```
stats avg(myfield1) by bin(5m)
```

La siguiente consulta genera una visualización de los tres valores basados en diferentes campos, con un punto de datos creado cada cinco minutos. La visualización se genera porque la consulta contiene las funciones de agregación y utiliza `bin()` como campo de agrupación.

```
stats avg(myfield1), min(myfield2), max(myfield3) by bin(5m)
```

Restricciones de los gráficos de líneas y de áreas apiladas

Las consultas que agregan información de entradas de registro pero no utilizan la función `bin()` pueden generar gráficos de barras. Sin embargo, las consultas no pueden generar gráficos de líneas ni gráficos de áreas apiladas. Para obtener más información sobre estos tipos de consultas, visite the section called “Visualización de datos de registro agrupados por campos”.

Visualización de datos de registro agrupados por campos

Puede generar gráficos de barras para consultas que utilizan la función `stats` y una o varias funciones de agregación. Para obtener más información, consulte Aggregation Functions in the Stats Command.

Para ver la visualización, ejecute la consulta. A continuación, elija la pestaña Visualization (Visualización), seleccione la flecha situada junto a Line (Línea) y haga clic en Bar (Barra). Las visualizaciones de los gráficos de barras tienen un límite máximo de 100 barras.

Ejemplos

Para ver un tutorial completo, consulte the section called “Tutorial: Ejecutar una consulta que produce una visualización agrupada por campos de registro”. Los párrafos siguientes incluyen más consultas de ejemplo de visualizaciones por campos.

La siguiente consulta de registro de flujo de VPC busca el número medio de bytes transferidos por sesión en cada dirección de destino.

```
stats avg(bytes) by dstAddr
```

También puede generar un gráfico que contenga varias barras para cada valor resultante. Por ejemplo, la siguiente consulta de registro de flujo de VPC busca el número medio y máximo de bytes transferidos por sesión en cada dirección de destino.

```
stats avg(bytes), max(bytes) by dstAddr
```

En la siguiente consulta, se busca el número de registros de Amazon Route 53 para cada tipo de consulta.

```
stats count(*) by queryType
```

Utilice varios comandos de estadísticas en una sola consulta

Puede usar hasta dos comandos `stats` en una sola consulta. Esto le permite realizar una agregación adicional en el resultado de la primera agregación.

Ejemplo: consulta con dos comandos **`stats`**

Por ejemplo, la siguiente consulta busca primero el volumen de tráfico total en los intervalos de 5 minutos y, a continuación, calcula el volumen de tráfico más alto, más bajo y medio de esos intervalos de 5 minutos.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length)/1024/1024 as logs_mb BY bin(5m)
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb
```

Ejemplo: combine varios comandos de estadísticas con otras funciones, como **`filter`**, **`fields`**, **`bin`**

Puede combinar dos comandos `stats` con otros comandos, como `filter` y `fields`, en una sola consulta. Por ejemplo, la siguiente consulta busca el número de direcciones IP distintas en las sesiones y busca el número de sesiones por plataforma de cliente, filtra esas direcciones IP y, finalmente, busca el promedio de solicitudes de sesión por plataforma del cliente.

```
STATS count_distinct(client_ip) AS session_ips,
      count(*) AS requests BY session_id, client_platform
| FILTER session_ips > 1
| STATS count(*) AS multiple_ip_sessions,
        sum(requests) / count(*) AS avg_session_requests BY client_platform
```

Puede utilizar funciones `bin` y `dateceil` en consultas con varios comandos `stats`. Por ejemplo, la siguiente consulta combina primero los mensajes en bloques de 5 minutos, luego agrega esos bloques de 5 minutos en bloques de 10 minutos y calcula los volúmenes de tráfico más altos, más bajos y promedio dentro de cada bloque de 10 minutos.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length) / 1024 / 1024 AS logs_mb BY BIN(5m) as @t
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb BY dateceil(@t, 10m)
```

Notas y limitaciones

Una consulta puede tener un máximo de dos comandos `stats`. Esta cuota no se puede cambiar.

Si utiliza un comando `sort` o `limit`, debe aparecer después del segundo comando `stats`. Si está antes del segundo comando `stats`, la consulta no es válida.

Cuando una consulta tiene dos comandos `stats`, los resultados parciales de la consulta no comienzan a mostrarse hasta que se completa la primera agregación `stats`.

En el segundo comando `stats` de una consulta única, solo puede hacer referencia a los campos definidos en el primer comando `stats`. Por ejemplo, la siguiente consulta no es válida porque el campo `@message` no estará disponible después de la primera agregación `stats`.

```
FIELDS @message
| STATS SUM(Fault) by Operation
# You can only reference `SUM(Fault)` or Operation at this point
| STATS MAX(strlen(@message)) AS MaxMessageSize # Invalid reference to @message
```

Todos los campos a los que haga referencia después del primer comando `stats` deben definirse en ese primer comando `stats`.

```
STATS sum(x) as sum_x by y, z
| STATS max(sum_x) as max_x by z
# You can only reference `max(sum_x)`, max_x or z at this point
```

Important

La función `bin` siempre utiliza el campo `@timestamp` de forma implícita. Esto significa que no se puede usar `bin` en el segundo comando `stats` sin usar el primer comando `stats` para propagar el campo `timestamp`. Por ejemplo, la siguiente consulta no es válida.

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes BY @logStream
 | STATS avg(ingested_bytes) BY bin(5m) # Invalid reference to @timestamp field
```

En su lugar, defina el campo `@timestamp` en el primer comando `stats` y, a continuación, podrá usarlo con `dateceil` en el segundo comando `stats`, como en el siguiente ejemplo.

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes, max(@timestamp) as @t BY
 @logStream
 | STATS avg(ingested_bytes) BY dateceil(@t, 5m)
```

Funciones para usar con estadísticas

CloudWatch Logs Insights admite funciones de agregación de estadísticas y funciones de no agregación de estadísticas.

Utilice las funciones de agregación de estadísticas en el comando `stats` y como argumentos para otras funciones.

Función	Tipo de resultado	Descripción
`avg(fieldName: NumericLogField)`	number	La media de los valores en el campo especificado.

`count()`
 	
`count(fieldName: LogField)`
	number	Cuenta los eventos de registro. `count()` (o `count(*)`) cuenta todos los eventos devueltos por la consulta, mientras que `count(fieldName)` cuenta todos los registros que incluyen el nombre de campo especificado.

`count_distinct(fieldName: LogField)`	number	Devuelve el número de valores únicos para el campo. Si el campo tiene una cardinalidad muy alta (contiene muchos valores únicos), el valor devuelto por `count_distinct` es solo una aproximación.
`max(fieldName: LogField)`	LogFieldValue	El máximo de los valores para este campo de registro en los registros consultados.
`min(fieldName: LogField)`	LogFieldValue	El mínimo de los valores para este campo de registro en los registros consultados.
`pct(fieldName: LogFieldValue, percent: number)`	LogFieldValue	Un percentil indica el peso relativo de un valor en un conjunto de datos. Por ejemplo, `pct(@duration, 95)` devuelve el valor `@duration` en que el 95 % de los valores de `@duration` son inferiores a este valor y un 5 por ciento son superiores a este valor.

Función	Tipo de resultado	Descripción
`stddev(fieldName: NumericLogField)`	number	El desvío estándar de los valores en el campo especificado.
`sum(fieldName: NumericLogField)`	number	La suma de los valores en el campo especific ado.

Funciones sin agregación de estadísticas

Utilice las funciones de no agregación en el comando `stats` y como argumentos para otras funciones.

Función	Tipo de resultado	Descripción
`earliest(fieldName: LogField)`	LogField	Devuelve el valor de `fieldName` desde el evento de registro que tiene la primera marca temporal en los registros consultados.
`latest(fieldName: LogField)`	LogField	Devuelve el valor de `fieldName` desde el evento de registro que tiene la última marca temporal en los registros consultados.
`sortsFirst(fieldNa me: LogField)`	LogField	Devuelve el valor de `fieldName` que ordena en primer lugar los registros consultados.
`sortsLast(fieldName: LogField)`	LogField	Devuelve el valor de `fieldName` que ordena al final los registros consultados.

límite

Use `limit` para especificar el número de eventos de registro que desea que devuelva la consulta.

Por ejemplo, el siguiente ejemplo devuelve solo los 25 eventos de registro más recientes

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

dedup

Utilice `dedup` para eliminar los resultados duplicados en función de valores específicos en los campos que indique. Puede utilizar `dedup` con uno o más campos. Si especifica un campo con `dedup`, solo se devolverá un evento de registro por cada valor único de ese campo. Si especifica varios campos, se devolverá un evento de registro por cada combinación única de valores para esos campos.

Los resultados duplicados se descartan según el orden de clasificación y solo se conserva el primer resultado del orden de clasificación. Le recomendamos que ordene los resultados antes de someterlos al comando `dedup`. Si los resultados no se ordenan antes de analizarlos con `dedup`, se utiliza el orden de clasificación descendente predeterminado de `@timestamp`.

Los valores nulos no se consideran duplicados para la evaluación. Se conservan los eventos de registro con valores nulos para cualquiera de los campos especificados. Para eliminar campos con valores nulos, utilice **`filter`** mediante la función `isPresent(field)`.

El único comando de consulta que puede utilizar en una consulta después del comando `dedup` es `limit`.

Ejemplo: Vea solo el evento de registro más reciente para cada valor único del campo denominado **`server`**

En el siguiente ejemplo, se muestran los campos `timestamp`, `server`, `severity` y `message` solo para el evento más reciente de cada valor único de `server`.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server
```

Para ver más ejemplos de consultas de CloudWatch Logs Insights, consulteConsultas generales.

unmask

`unmask` se utiliza para mostrar todo el contenido de un evento de registro que tiene parte del contenido enmascarado debido a una política de protección de datos. Para ejecutar este comando, debe tener el permiso `logs:Unmask`.

Para obtener más información sobre la protección de datos en grupos de registro, consulte Ayude a proteger los datos de registro confidenciales con el enmascaramiento.

Funciones booleanas, de comparación, numéricas, de fecha y hora y otras

CloudWatch Logs Insights admite muchas otras operaciones y funciones en las consultas, como se explica en las siguientes secciones.

Temas

Operadores aritméticos

Los operadores aritméticos aceptan tipos de datos numéricos como argumentos y devuelven resultados numéricos. Utilice operadores aritméticos en los comandos `filter` y `fields` y como argumentos para otras funciones.

Operación	Descripción
`a + b`	Suma
`a - b`	Resta
`a * b`	Multiplicación
`a / b`	División
`a ^ b`	Potencia (`2 ^ 3` devuelve 8)
`a % b`	Resto o módulo (`10 % 3` devuelve 1)

Operadores booleanos

Utilice los operadores booleanos **`and`**, **`or`** y **`not`**.

> **Note**
>
> Utilice operadores booleanos solo en funciones que devuelvan el valor TRUE o FALSE.

Operadores de comparación

Los operadores de comparación aceptan todos los tipos de datos como argumentos y devuelven un resultado booleano. Utilice operadores de comparación en el comando `filter` y como argumentos para otras funciones.

Operador	Descripción
=	Igualdad
!=	Desigualdad
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que

Operadores numéricos

Las operaciones numéricas aceptan tipos de datos numéricos como argumentos y devuelven resultados numéricos. Utilice operaciones numéricas en los comandos `filter` y `fields` y como argumentos para otras funciones.

Operación	Tipo de resultado	Descripción
`abs(a: number)`	number	Valor absoluto

Operación	Tipo de resultado	Descripción
`ceil(a: number)`	number	Redondeo a valor máximo (menor número entero que es mayor que el valor de a).
`floor(a: number)`	number	Redondeo a valor mínimo (mayor número entero que es menor que el valor de a).
`greatest(a: number, ...numbers: number[])`	number	Devuelve el valor más alto
`least(a: number, ...numbers: number[])`	number	Devuelve el valor más bajo
`log(a: number)`	number	Registro natural
`sqrt(a: number)`	number	Raíz cuadrada

Funciones DateTime

Funciones DateTime

Utilice funciones datetime en los comandos `fields` y `filter` y como argumentos para otras funciones. Utilice estas funciones para crear buckets de hora para consultas con funciones de agregación. Utilice períodos de tiempo que consten de un número y uno de los siguientes elementos:

- `ms`durante milisegundos
- `s`durante segundos
- `m`durante minutos
- `h`durante horas

Por ejemplo, `10m` es 10 minutos y `1h` es 1 hora.

> **Note**
>
> Utilice la unidad de tiempo más adecuada para su función de fecha y hora. CloudWatch Los registros limitan su solicitud de acuerdo con la unidad de tiempo que elija. Por ejemplo, pone un límite de 60 como valor máximo para cualquier solicitud que utilices. Por lo tanto, si lo especificas`bin(300s)`, CloudWatch Logs en realidad lo implementa como 60 segundos, ya que 60 es el número de segundos que hay en un minuto, por lo que CloudWatch Logs no utilizará un número superior a 60s. Para crear un intervalo de 5 minutos, utilízalo `bin(5m)` en su lugar.
> El límite para `ms` es 1000, el límite para `s` y `m` es 60 y el límite para h es 24.

En la siguiente tabla, se incluye una lista de las distintas funciones datetime que se pueden usar en comandos de consulta. La tabla enumera el tipo de resultado de cada función y contiene una descripción de cada función.

> **Tip**
>
> Al crear un comando de consulta, puede utilizar el selector de intervalos de tiempo para seleccionar un periodo de tiempo que desea consultar. Por ejemplo, puede establecer un periodo entre intervalos de 5 a 30 minutos; intervalos de 1, 3 y 12 horas; o un marco temporal personalizado. También puede establecer periodos de tiempo entre fechas específicas.

Función	Tipo de resultado	Descripción
`bin(period: Period)`	Timestamp	Redondea el valor de `@timestamp` según el periodo de tiempo indicado y, a continuac ión, trunca. Por ejemplo, `bin(5m)` redondea el valor de `@timestamp` a los 5 minutos más cercanos. Puede usarlo para agrupar varias entradas de registro en una consulta. En el siguiente ejemplo, se devuelve el número de excepcion es por hora:

Función	Tipo de resultado	Descripción

		<pre>filter @message like /Exception/
		    | stats count(*) as exceptionCount
		 by bin(1h)
		    | sort exceptionCount desc</pre>
		La función `bin` admite las siguientes unidades de tiempo y abreviaturas. Para todas las unidades y abreviaturas que incluyan más de un carácter, se admite agregar `s` para pluralizar. Así que tanto `hr` como `hrs` funcionan para especificar las horas.
		• `millisecond ms msec`
		• `second s sec`
		• `minute m min`
		• `hour h hr`
		• `day d`
		• `week w`
		• `month mo mon`
		• `quarter q qtr`
		• `year y yr`

`datefloor(timestamp: Timestamp, period: Period)`	Timestamp	Trunca la marca temporal según el periodo indicado. Por ejemplo, `datefloor (@timestamp, 1h)` trunca todos los valores de `@timestamp` en la parte inferior de la hora.
`dateceil(timestamp : Timestamp, period: Period)`	Timestamp	Redondea hacia arriba la marca temporal según el periodo indicado y, a continuación, trunca. Por ejemplo, `dateceil(@timestamp, 1h)` trunca todos los valores de `@timestamp` en la parte superior de la hora.

Función	Tipo de resultado	Descripción
`fromMillis(fieldName: number)`	Timestamp	Interpreta el campo de entrada como el número de milisegundos desde la fecha de inicio de Unix y lo convierte en una marca de tiempo.
`toMillis(fieldName: Timestamp)`	number	Convierte la marca de tiempo que se encontró en el campo con nombre asignado en un número que representa los milisegundos desde la fecha de inicio de Unix. Por ejemplo, `toMillis(@timestamp)` convierte la marca temporal `2022-01-14T13:18:031.000-08:00` a `1642195111000` .

> **Note**
>
> Actualmente, CloudWatch Logs Insights no admite el filtrado de registros con marcas de tiempo legibles por humanos.

Funciones generales

Funciones generales

Utilice funciones generales en los comandos `fields` y `filter` y como argumentos para otras funciones.

Función	Tipo de resultado	Descripción
`ispresent(fieldName: LogField)`	Booleano	Devuelve `true` si el campo existe
`coalesce(fieldName: LogField, ...fieldNames: LogField[])`	LogField	Devuelve el primer valor no nulo de la lista

Funciones de cadena de dirección IP

Funciones de cadena de dirección IP

Utilice funciones de cadena de dirección IP en los comandos `filter` y `fields` y como argumentos para otras funciones.

Función	Tipo de resultado	Descripción
`isValidIp(fieldName: string)`	booleano	Devuelve `true` si el campo es una dirección IPv4 o IPv6 válida.
`isValidIpV4(fieldName: string)`	boolean	Devuelve `true` si el campo es una dirección IPv4 válida.
`isValidIpV6(fieldName: string)`	boolean	Devuelve `true` si el campo es una dirección IPv6 válida.
`isIpInSubnet(fieldName: string, subnet: string)`	boolean	Devuelve `true` si el campo es una dirección IPv4 o IPv6 válida dentro de la subred v4 o v6 especificada. Al especificar la subred, utilice la notación CIDR como `192.0.2.0/24` o `2001:db8::/32` , donde `192.0.2.0` o `2001:db8::` es el inicio del bloque de CIDR.
`isIpv4InSubnet(fieldName: string, subnet: string)`	boolean	Devuelve `true` si el campo es una dirección IPv4 válida dentro de la subred v4 especificada. Al especificar la subred, utilice la notación CIDR como `192.0.2.0/24` , donde `192.0.2.0` es el inicio del bloque de CIDR.
`isIpv6InSubnet(fieldName: string, subnet: string)`	boolean	Devuelve `true` si el campo es una dirección IPv6 válida dentro de la subred v6 especificada. Al especificar la subred, utilice la notación CIDR como `2001:db8::/32` , donde `2001:db8::` es el inicio del bloque de CIDR.

Funciones de cadena

Funciones de cadena

Utilice funciones de cadena en los comandos `fields` y `filter` y como argumentos para otras funciones.

Función	Tipo de resultado	Descripción
`isempty(fieldName: string)`	Número	Devuelve 1 si el campo no se encuentra o es una cadena vacía.
`isblank(fieldName: string)`	Número	Devuelve 1 si el campo no se encuentra, es una cadena vacía o solo contiene espacio en blanco.
`concat(str: string, ...strings: string[])`	cadena	Concatena las cadenas.

`ltrim(str: string)`
 	
`ltrim(str: string, trimChars: string)`
	cadena	Si la función no tiene un segundo argumento de cadena, elimina los espacios en blanco de la izquierda de la cadena. Si la función tiene un segundo argumento de cadena, no elimina espacios en blanco. En su lugar, elimina los caracteres de `trimChars` desde la izquierda de `str`. Por ejemplo, `ltrim("xyZxyfooxyZ","xyZ")` devuelve `"fooxyZ"`.

`rtrim(str: string)`	cadena	Si la función tiene un segundo argumento de cadena, elimina los espacios en blanco de

Función	Tipo de resultado	Descripción
`rtrim(str: string, trimChars: string)`		la derecha de la cadena. Si la función tiene un segundo argumento de cadena, no elimina espacios en blanco. En su lugar, elimina los caracteres de `trimChars` desde la derecha de `str`. Por ejemplo, `rtrim("xyZfooxyxyZ","xyZ")` devuelve `"xyZfoo"`.

`trim(str: string)`
 	
`trim(str: string, trimChars: string)`
	cadena	Si la función no tiene un segundo argumento, elimina espacios en blanco de ambos extremos de la cadena. Si la función tiene un segundo argumento de cadena, no elimina espacios en blanco. En su lugar, elimina los caracteres de `trimChars` desde ambos lados de `str`. Por ejemplo, `trim("xyZxyfooxyxyZ","xyZ")` devuelve `"foo"`.

`strlen(str: string)`	number	Devuelve la longitud de la cadena puntos de código Unicode.
`toupper(str: string)`	cadena	Convierte la cadena en mayúsculas.
`tolower(str: string)`	cadena	Convierte la cadena de caracteres en minúsculas.

Función	Tipo de resultado	Descripción

`substr(str: string, startIndex: number)`
 	
`substr(str: string, startIndex: number, length: number)`
	cadena	Devuelve una subcadena del índice especificado por el argumento numérico al final de la cadena. Si la función tiene un segundo argumento numérico, contiene la longitud de la subcadena que debe recuperarse. Por ejemplo, `substr("xyZfooxyZ",3, 3)` devuelve `"foo"`.

`replace(fieldName: string, searchValue: string, replaceValue: string)`	cadena	Sustituye todas las instancias de `searchValue` en `fieldName: string` por `replaceValue` . Por ejemplo, la función `replace(logGroup,"smoke_test","Smoke")` busca eventos de registro en los que el campo `logGroup` contiene el valor de cadena `smoke_test` y reemplaza el valor por la cadena `Smoke`.
`strcontains(str: string, searchValue: string)`	number	Devuelve 1 si `str` contiene `searchValue` y 0 en los demás casos.

Campos que contienen caracteres especiales

Si un campo contiene caracteres no alfanuméricos distintos del `@` símbolo o el punto (`.`), debe rodearlo con caracteres de comilla invertida (). ` Por ejemplo, el campo de registro `foo-bar` debe

aparecer entre acentos graves (`foo-bar`), porque contiene un carácter no alfanumérico, el guion (-).

Uso de alias y comentarios en las consultas

Cree consultas que contengan alias. Utilice alias para cambiar el nombre de los campos de registro o al extraer valores en campos. Utilice la palabra clave `as` para asignar un alias a un resultado o campo de registro. Puede utilizar más de un alias en una consulta. Puede utilizar alias en los siguientes comandos:

- `fields`
- `parse`
- `sort`
- `stats`

En los siguientes ejemplos, se muestra cómo crear consultas que contienen alias.

Ejemplo

La consulta contiene un alias en el comando `fields`.

```
fields @timestamp, @message, accountId as ID
| sort @timestamp desc
| limit 20
```

La consulta devuelve los valores de los campos `@timestamp`, `@message` y `accountId`. Los resultados se muestran en orden descendente y su número se limita a 20. Los valores de `accountId` aparecen bajo el alias `ID`.

Ejemplo

La consulta contiene alias en los comandos `sort` y `stats`.

```
stats count(*) by duration as time
| sort time desc
```

La consulta cuenta el número de veces que el campo `duration` aparece en el grupo de registros y muestra los resultados en orden descendente. Los valores de `duration` aparecen bajo el alias `time`.

Uso de comentarios

CloudWatch Logs Insights admite comentarios en las consultas. Utilice el carácter de la almohadilla (#) para desactivar los comentarios. Puede utilizar comentarios para que haga caso omiso de determinadas líneas en consultas o consultas de documentos.

Ejemplo: consulta

Cuando se ejecuta la siguiente consulta, se hace caso omiso de la segunda línea.

```
fields @timestamp, @message, accountId
# | filter accountId not like "7983124201998"
| sort @timestamp desc
| limit 20
```

Análisis de patrones

CloudWatch Logs Insights utiliza algoritmos de aprendizaje automático para encontrar patrones cuando consultas tus registros. Un patrón es una estructura de texto compartida que se repite en los campos de registro. Al ver los resultados de una consulta, puede seleccionar la pestaña Patrones para ver los patrones que los CloudWatch registros encontraron a partir de una muestra de sus resultados. Como alternativa, puede añadir el `pattern` comando a la consulta para analizar los patrones de todo el conjunto de eventos de registro coincidentes.

Los patrones son útiles para analizar conjuntos de registros grandes porque, a menudo, una gran cantidad de eventos de registro se pueden comprimir en unos pocos patrones.

Considere el siguiente ejemplo de tres eventos de registro.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

En el ejemplo anterior, los tres eventos de registro siguen un patrón:

```
<*> <*> [INFO] Calling DynamoDB to store for resource id <*>
```

Los campos dentro de un patrón se denominan fichas. Los campos que varían dentro de un patrón, como un identificador de solicitud o una marca de tiempo, son símbolos dinámicos. Cada token dinámico se representa <*> cuando CloudWatch Logs lo muestra.

Algunos ejemplos comunes de tokens dinámicos son los códigos de error, las marcas de tiempo y los identificadores de solicitud. El valor de un token representa un valor concreto de un token dinámico. Por ejemplo, si un token dinámico representa un código de error HTTP, el valor del token podría ser`501`.

La detección de patrones también se utiliza en el detector de anomalías de CloudWatch Logs y en las funciones de comparación. Para obtener más información, consulte Detección de anomalías de registro y Compara (diferencia) con intervalos de tiempo anteriores.

Cómo empezar con el análisis de patrones

La detección de patrones se realiza automáticamente en cualquier consulta de CloudWatch Logs Insights. Las consultas que no incluyen el `pattern` comando registran tanto los eventos como los patrones en los resultados.

Si incluye el `pattern` comando en la consulta, el análisis de patrones se realiza en todo el conjunto de eventos de registro coincidentes. Esto proporciona resultados de patrones más precisos, pero los eventos de registro sin procesar no se devuelven cuando se utiliza el `pattern` comando. Cuando una consulta no incluye`pattern`, los resultados del patrón se basan en los primeros 1000 eventos de registro devueltos o en el valor límite que utilizó en la consulta. Si lo incluye `pattern` en la consulta, los resultados que se muestran en la pestaña Patrones se derivan de todos los eventos de registro que coinciden con la consulta.

Para empezar con el análisis de patrones en CloudWatch Logs Insights

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs, Logs Insights.

   En la página Información de registros, el editor de consultas contiene una consulta predeterminada que devuelve los 20 eventos de registro más recientes.
3. Elimine la `| limit 20` línea del cuadro de consulta para que la consulta tenga el siguiente aspecto:

   ```
   fields @timestamp, @message, @logStream, @log
   | sort @timestamp desc
   ```
4. En el menú desplegable Seleccionar grupos de registros, elija uno o más grupos de registros para realizar la consulta.

5. (Opcional) Utilice el selector de tiempo para seleccionar el periodo de tiempo que desea consultar.

   Puede elegir entre intervalos de 5 y 30 minutos; intervalos de 1 hora, 3 horas y 12 horas; o un período de tiempo personalizado.

6. Seleccione Ejecutar consulta para iniciar la consulta.

   Cuando la consulta termina de ejecutarse, la pestaña Registros muestra una tabla de eventos de registro devueltos por la consulta. Encima de la tabla hay un mensaje sobre el número de registros que coinciden con la consulta, similar a Mostrar 1000 de los 71.101 registros que coinciden.

7. Seleccione la pestaña Patrones.

8. La tabla muestra ahora los patrones encontrados en la consulta. Como la consulta no incluía el `pattern` comando, esta pestaña muestra solo los patrones detectados entre los 1000 eventos de registro que se mostraban en la tabla de la pestaña Registros.

   Para cada patrón, se muestra la siguiente información:

   - El patrón, en el que cada símbolo dinámico se muestra como<*>.
   - El recuento de eventos, que es el número de veces que el patrón apareció en el registro de eventos consultado. Elija el encabezado de la columna Recuento de eventos para ordenar los patrones por frecuencia.
   - La proporción de eventos, que es el porcentaje de eventos del registro consultados que contienen este patrón.
   - El tipo de gravedad, que será uno de los siguientes:
     - ERROR si el patrón contiene la palabra Error.
     - AVISE si el patrón contiene la palabra Advertir pero no contiene error.
     - INFORMACIÓN si el patrón no contiene advertencia ni error.

     Elija el encabezado de la columna de información sobre la gravedad para ordenar los patrones por gravedad.

9. Ahora cambia la consulta. Sustituya la `| sort @timestamp desc` línea de la consulta por`| pattern @message`, de modo que la consulta completa quede de la siguiente manera:

```
fields @timestamp, @message, @logStream, @log
| pattern @message
```

10. Elija Ejecutar consulta.

    Cuando finalice la consulta, no habrá resultados en la pestaña Registros. Sin embargo, es probable que la pestaña Patrones muestre un mayor número de patrones, en función del número total de eventos de registro que se hayan consultado.

11. Independientemente de si los has incluido `pattern` en la consulta, puedes inspeccionar más a fondo los patrones que devuelve la consulta. Para ello, elija el icono de uno de los patrones en la columna Inspeccionar.

    Aparece el panel de inspección de patrones, que muestra lo siguiente:

    - El patrón. Seleccione un token dentro del patrón para analizar los valores de ese token.
    - Un histograma que muestra el número de apariciones del patrón en el intervalo de tiempo consultado. Esto puede ayudarle a identificar tendencias interesantes, como un aumento repentino de la aparición de un patrón.
    - La pestaña Registrar muestras muestra algunos de los eventos de registro que coinciden con el patrón seleccionado.
    - La pestaña Valores del token muestra los valores del token dinámico seleccionado, si ha seleccionado uno.

      > **Note**
      >
      > Se captura un máximo de 10 valores de token para cada token. Es posible que los recuentos de fichas no sean precisos. CloudWatch Logs utiliza un contador probabilístico para generar el recuento de fichas, no el valor absoluto.

    - La pestaña Patrones relacionados muestra otros patrones que se han producido con frecuencia casi al mismo tiempo que el patrón que se está inspeccionando. Por ejemplo, si el patrón de un `ERROR` mensaje solía ir acompañado de otro evento de registro marcado `INFO` con detalles adicionales, ese patrón se muestra aquí.

Detalles sobre el comando pattern

Esta sección contiene más detalles sobre el `pattern` comando y sus usos.

- En el tutorial anterior, eliminamos el `sort` comando cuando lo agregamos `pattern` porque una consulta no es válida si incluye un `pattern` comando después de otro `sort` comando. Es válido tener un `pattern` antes de un`sort`.

  Para obtener más información sobre `pattern` la sintaxis, consultepattern.
- Cuando se utiliza `pattern` en una consulta, `@message` debe ser uno de los campos seleccionados en el `pattern` comando.
- Puede incluir el `filter` comando antes de un `pattern` comando para que solo el conjunto filtrado de eventos de registro se utilice como entrada para el análisis de patrones.
- Para ver los resultados del patrón de un campo concreto, como un campo derivado del `parse` comando, utilice`pattern @fieldname`.
- Las consultas con un resultado que no sea de registro, como las consultas con el `stats` comando, no devuelven resultados de patrones.

Compara (diferencia) con intervalos de tiempo anteriores

Puedes usar CloudWatch Logs Insights para comparar los cambios en tus eventos de registro a lo largo del tiempo. Puede comparar los eventos de registro ingeridos durante un intervalo de tiempo reciente con los registros del período inmediatamente anterior. Como alternativa, puede compararlos con períodos de tiempo anteriores similares. Esto puede ayudarte a determinar si un error en tus registros se ha introducido recientemente o si ya se estaba produciendo, y también puede ayudarte a encontrar otras tendencias.

Las consultas de comparación solo muestran patrones en los resultados, no eventos de registro sin procesar. Los patrones devueltos le ayudarán a ver rápidamente las tendencias y los cambios en los eventos del registro a lo largo del tiempo. Tras realizar una consulta comparativa y obtener los resultados de los patrones, podrá ver ejemplos de eventos de registro sin procesar correspondientes a los patrones que le interesen. Para obtener más información sobre los patrones de registro, consulteAnálisis de patrones.

Cuando ejecuta una consulta de comparación, la consulta se analiza en función de dos períodos de tiempo diferentes: el período de consulta original que haya seleccionado y el período de comparación. El período de comparación siempre tiene la misma duración que el período de consulta original. Los intervalos de tiempo predeterminados para las comparaciones son los siguientes.

- Período anterior: se compara con el período inmediatamente anterior al período de consulta.

- Día anterior: se compara con el período de un día anterior al período de consulta.
- Semana anterior: se compara con el período de una semana anterior al período de consulta.
- Mes anterior: se compara con el período de un mes anterior al período de consulta.

> **Note**
>
> Las consultas que utilizan comparaciones tienen un coste similar al de ejecutar una sola consulta de CloudWatch Logs Insights durante el intervalo de tiempo combinado. Para obtener más información, consulta los CloudWatch precios de Amazon.

Para ejecutar una consulta comparativa

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs, Logs Insights.

   Aparece una consulta predeterminada en el cuadro de consulta.
3. Mantenga la consulta predeterminada o introduzca una consulta diferente.
4. En el menú desplegable Seleccionar grupos de registros, elija uno o más grupos de registros para consultarlos.
5. (Opcional) Utilice el selector de tiempo para seleccionar el periodo de tiempo que desea consultar. La consulta predeterminada corresponde a los datos de registro de la hora anterior.
6. En el selector de intervalo de tiempo, selecciona Comparar. A continuación, elija el período de tiempo anterior con el que desee comparar los registros originales y seleccione Aplicar.
7. Elija Ejecutar consulta.

   Para que la consulta recoja los datos del período de comparación, el `diff` comando se añade a la consulta.
8. Seleccione la pestaña Patrones para ver los resultados.

   En la tabla se muestra la siguiente información:

   - Cada patrón, con partes variables del patrón sustituidas por el símbolo simbólico dinámico<*>. Para obtener más información, consulte Análisis de patrones.
   - El recuento de eventos es el número de eventos registrados con ese patrón en el período de tiempo original, más actual.

- El recuento de eventos diferencial es la diferencia entre el número de eventos de registro coincidentes en el período de tiempo actual y el período de comparación. Una diferencia positiva significa que hay más eventos de este tipo en el período de tiempo actual.
- La descripción de la diferencia resume brevemente el cambio en ese patrón entre el período de tiempo actual y el período de comparación.
- El tipo de gravedad es la gravedad probable de los eventos del registro con este patrón, basada en las palabras que se encuentran en el registro de eventos`FATAL`, como`ERROR`, y`WARN`.

9. Para seguir inspeccionando uno de los patrones de la lista, elija el icono de la columna Inspeccionar correspondiente a uno de los patrones.

   Aparece el panel de inspección de patrones, que muestra lo siguiente:

   - El patrón. Seleccione un token dentro del patrón para analizar los valores de ese token.
   - Un histograma que muestra el número de apariciones del patrón en el intervalo de tiempo consultado. Esto puede ayudarle a identificar tendencias interesantes, como un aumento repentino de la aparición de un patrón.
   - La pestaña Registrar muestras muestra algunos de los eventos de registro que coinciden con el patrón seleccionado.
   - La pestaña Valores del token muestra los valores del token dinámico seleccionado, si ha seleccionado uno.

     > **Note**
     >
     > Se captura un máximo de 10 valores de token para cada token. Es posible que los recuentos de fichas no sean precisos. CloudWatch Logs utiliza un contador probabilístico para generar el recuento de fichas, no el valor absoluto.

   - La pestaña Patrones relacionados muestra otros patrones que se han producido con frecuencia casi al mismo tiempo que el patrón que se está inspeccionando. Por ejemplo, si el patrón de un `ERROR` mensaje solía ir acompañado de otro evento de registro marcado `INFO` con detalles adicionales, ese patrón se muestra aquí.

Consultas de ejemplo

Esta sección contiene una lista de comandos de consulta generales y útiles que puede ejecutar en la CloudWatch consola. Para obtener información sobre cómo ejecutar un comando de consulta, consulte el Tutorial: Ejecutar y modificar una consulta de ejemplo en la Guía del usuario de Amazon CloudWatch Logs.

Para obtener más información sobre la sintaxis de las consultas, consulteCloudWatch Sintaxis de consultas de Logs Insights.

Temas

Consultas generales

Buscar los 25 eventos de registro agregados más recientes.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

Obtener una lista del número de excepciones por hora.

```
filter @message like /Exception/
    | stats count(*) as exceptionCount by bin(1h)
    | sort exceptionCount desc
```

Obtener una lista de los eventos de registro que no son excepciones.

```
fields @message | filter @message not like /Exception/
```

Obtener el evento de registro más reciente para cada valor único del campo **`server`**.

```
fields @timestamp, server, severity, message
| sort @timestamp asc
| dedup server
```

Obtener el evento de registro más reciente para cada valor único del campo **`server`** para cada tipo **`severity`**.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server, severity
```

Consultas de registros de Lambda

Determinar la cantidad de memoria sobreaprovisionada.

```
filter @type = "REPORT"
    | stats max(@memorySize / 1000 / 1000) as provisonedMemoryMB,
        min(@maxMemoryUsed / 1000 / 1000) as smallestMemoryRequestMB,
        avg(@maxMemoryUsed / 1000 / 1000) as avgMemoryUsedMB,
        max(@maxMemoryUsed / 1000 / 1000) as maxMemoryUsedMB,
        provisonedMemoryMB - maxMemoryUsedMB as overProvisionedMB
```

Crear un informe de latencia.

```
filter @type = "REPORT" |
    stats avg(@duration), max(@duration), min(@duration) by bin(5m)
```

Buscar invocaciones de funciones lentas y eliminar las solicitudes duplicadas que puedan surgir de los reintentos o del código del lado del cliente. En esta consulta, **`@duration`** está en milisegundos.

```
fields @timestamp, @requestId, @message, @logStream
| filter @type = "REPORT" and @duration > 1000
```

```
| sort @timestamp desc
| dedup @requestId
| limit 20
```

Consultas de registros de flujo de Amazon VPC

Buscar las 15 primeras transferencias de paquete en hosts:

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

Buscar las 15 primeras transferencias de bytes para los hosts de una subred determinada.

```
filter isIpv4InSubnet(srcAddr, "192.0.2.0/24")
    | stats sum(bytes) as bytesTransferred by dstAddr
    | sort bytesTransferred desc
    | limit 15
```

Buscar las direcciones IP que utilizan UDP como protocolo de transferencia de datos.

```
filter protocol=17 | stats count(*) by srcAddr
```

Buscar las direcciones IP donde los registros de flujo se omitieron durante la ventana de captura.

```
filter logStatus="SKIPDATA"
    | stats count(*) by bin(1h) as t
    | sort t
```

Buscar un registro único para cada conexión, para ayudar a solucionar problemas de conectividad de red.

```
fields @timestamp, srcAddr, dstAddr, srcPort, dstPort, protocol, bytes
| filter logStream = 'vpc-flow-logs' and interfaceId = 'eni-0123456789abcdef0'
| sort @timestamp desc
| dedup srcAddr, dstAddr, srcPort, dstPort, protocol
```

```
| limit 20
```

Consultas de registros de Route 53

Buscar la distribución de registros por hora por tipo de consulta.

```
stats count(*) by queryType, bin(1h)
```

Buscar los 10 solucionadores de DNS con el mayor número de solicitudes.

```
stats count(*) as numRequests by resolverIp
    | sort numRequests desc
    | limit 10
```

Buscar el número de registros por dominio y subdominio donde el servidor no pudo completar la solicitud de DNS.

```
filter responseCode="SERVFAIL" | stats count(*) by queryName
```

Consultas de CloudTrail registros

Buscar el número de entradas de registro para cada servicio, tipo de evento y región de AWS .

```
stats count(*) by eventSource, eventName, awsRegion
```

Busque los hosts de Amazon EC2 que se iniciaron o se detuvieron en una región determinada AWS .

```
filter (eventName="StartInstances" or eventName="StopInstances") and awsRegion="us-
east-2"
```

Busque las AWS regiones, los nombres de usuario y los ARN de los usuarios de IAM recién creados.

```
filter eventName="CreateUser"
    | fields awsRegion, requestParameters.userName, responseElements.user.arn
```

Buscar el número de registros en los que se ha producido una excepción al invocar a la API **`UpdateTrail`**.

```
filter eventName="UpdateTrail" and ispresent(errorCode)
    | stats count(*) by errorCode, errorMessage
```

Buscar entradas de registro en las que se usó TLS 1.0 o 1.1

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by userIdentity.accountId, recipientAccountId,
 eventSource, eventName, awsRegion, tlsDetails.tlsVersion, tlsDetails.cipherSuite,
 userAgent
| sort eventSource, eventName, awsRegion, tlsDetails.tlsVersion
```

Buscar la cantidad de llamadas por servicio que usaron las versiones de TLS 1.0 o 1.1

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by eventSource
| sort numOutdatedTlsCalls desc
```

Consultas para Amazon API Gateway

Buscar los últimos 10 errores de 4XX

```
fields @timestamp, status, ip, path, httpMethod
| filter status>=400 and status<=499
| sort @timestamp desc
| limit 10
```

Identifique las 10 Amazon API Gateway solicitudes que llevan más tiempo ejecutándose en su grupo de registros de Amazon API Gateway acceso

```
fields @timestamp, status, ip, path, httpMethod, responseLatency
| sort responseLatency desc
| limit 10
```

Devuelve la lista de las rutas de API más populares de tu grupo de registros de Amazon API Gateway acceso

```
stats count(*) as requestCount by path
| sort requestCount desc
| limit 10
```

Crea un informe de latencia de integración para tu grupo de registros de Amazon API Gateway acceso

```
filter status=200
| stats avg(integrationLatency), max(integrationLatency),
min(integrationLatency) by bin(1m)
```

Consultas para la puerta de enlace NAT

Si observas costes más altos de lo normal en tu AWS factura, puedes usar CloudWatch Logs Insights para encontrar a los principales contribuyentes. Para obtener más información sobre los siguientes comandos de consulta, consulte ¿Cómo puedo encontrar los principales contribuyentes al tráfico a través de la puerta de enlace NAT en mi VPC? en la página AWS de soporte premium.

> **Note**
>
> En los siguientes comandos de consulta, sustituya “x.x.x.x” por la IP privada de la puerta de enlace NAT y sustituya “y.y” por los dos primeros octetos del rango CIDR de la VPC.

Buscar las instancias que envían más tráfico a través de su puerta de enlace de NAT

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Determinar el tráfico de entrada y salida de las instancias de las puertas de enlace de NAT

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.') or (srcAddr like 'xxx.xx.xx.xx'
 and dstAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Determinar los destinos de Internet con los que las instancias de la VPC se comunican con mayor frecuencia para las cargas y descargas.

Para cargas

```
filter (srcAddr like 'x.x.x.x' and dstAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Para descargas

```
filter (dstAddr like 'x.x.x.x' and srcAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Consultas para registros del servidor Apache

Puede usar CloudWatch Logs Insights para consultar los registros del servidor Apache. Para obtener más información sobre las siguientes consultas, consulte Simplificar los registros del servidor Apache con CloudWatch Logs Insights en el blog AWS Cloud Operations & Migrations.

Buscar los campos más relevantes para que pueda revisar sus registros de acceso y comprobar si hay tráfico en la ruta /admin de su aplicación

```
fields @timestamp, remoteIP, request, status, filename| sort @timestamp desc
| filter filename="/var/www/html/admin"
| limit 20
```

Buscar el número de solicitudes GET únicas que han accedido a su página principal con el código de estado “200” (correcta).

```
fields @timestamp, remoteIP, method, status
| filter status="200" and referrer= http://34.250.27.141/ and method= "GET"
| stats count_distinct(remoteIP) as UniqueVisits
| limit 10
```

Buscar el número de veces que se ha reiniciado el servicio Apache.

```
fields @timestamp, function, process, message
| filter message like "resuming normal operations"
| sort @timestamp desc
| limit 20
```

Consultas para Amazon EventBridge

Obtenga el número de EventBridge eventos agrupados por tipo de detalle del evento

```
fields @timestamp, @message
| stats count(*) as numberOfEvents by `detail-type`
| sort numberOfEvents desc
```

Ejemplos del comando para analizar

Utilice una expresión glob para extraer los campos **`@user`**, **`@method`** y **`@latency`** del campo de registro **`@message`** y devolver la latencia promedio para cada combinación única de **`@method`** y **`@user`**.

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

Utilice una expresión regular para extraer los campos **`@user2`**, **`@method2`** y **`@latency2`** del campo de registro **`@message`** y devolver la latencia promedio para cada combinación única de **`@method2`** y **`@user2`**.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Extrae los campos **`loggingTime`**, **`loggingType`** y **`loggingMessage`**, filtra hasta los eventos de registro que contienen cadenas **`ERROR`** o **`INFO`** y, a continuación, muestra solo los campos **`loggingMessage`** y **`loggingType`** para los eventos que contienen una cadena **`ERROR`**.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
```

```
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

Visualización de los datos de registro en gráficos

Puede usar visualizaciones como gráficos de barras, gráficos de líneas y gráficos de áreas apiladas para identificar de manera más eficiente los patrones en sus datos de registro. CloudWatch Logs Insights genera visualizaciones para las consultas que utilizan la `stats` función y una o más funciones de agregación. Para obtener más información, consulte stats.

Guarde y vuelva a ejecutar las consultas de CloudWatch Logs Insights

Cuando haya creado una consulta, puede guardarla para volver a ejecutarla más adelante. Las consultas se guardan en una estructura de carpetas para que pueda organizarlas. Puede guardar hasta 1000 consultas por región y cuenta.

Para guardar una consulta, debe haber iniciado sesión en un rol que tenga el permiso `logs:PutQueryDefinition`. Para ver una lista de consultas guardadas, debe haber iniciado sesión en un rol que tenga el permiso `logs:DescribeQueryDefinitions`.

Para guardar una consulta

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. En el editor de consultas, cree una consulta.
4. Seleccione Guardar.

   Si no ve el botón Guardar, debe cambiar al nuevo diseño de la consola de CloudWatch Logs. Para ello:

   a. En el panel de navegación, seleccione Grupos de registro.
   b. Elija Try the new design (Probar el nuevo diseño).
   c. En el panel de navegación, seleccione Información y vuelva al paso 3 de este procedimiento.
5. Escriba un nombre para la consulta.

6. (Opcional) Elija una carpeta en la que desee guardar la consulta. Seleccione Create new (Crear nueva) para crear una carpeta. Si crea una carpeta nueva, puede utilizar caracteres de barra (/) en el nombre de la carpeta para definir una estructura de carpetas. Por ejemplo, poner nombre a una carpeta nueva **`folder-level-1/folder-level-2`** crea una carpeta de nivel superior llamada **`folder-level-1`**, con otra carpeta llamada **`folder-level-2`** dentro de esa carpeta. La consulta se guarda en **`folder-level-2`**.
7. (Opcional) Cambie los grupos de registro de la consulta o el texto de la consulta.
8. Seleccione Guardar.

> **Tip**
>
> Puede crear una carpeta para las consultas guardadas con `PutQueryDefinition`. Con el fin de crear una carpeta para las consultas guardadas, utilice una barra diagonal (/) a fin de anteponer el nombre de la consulta deseada con el nombre de la carpeta deseada: <*`folder-name`*>/<*`query-name`*>. Para obtener más información sobre esta acción, consulte PutQueryDefinition.

Para ejecutar una consulta guardada

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. A la derecha, elija Queries (Consultas).
4. Seleccione la consulta de la lista Saved queries (Consultas guardadas) . Aparece en el editor de consultas.
5. Elija Ejecutar.

Para guardar una nueva versión de una consulta guardada

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. A la derecha, elija Queries (Consultas).
4. Seleccione la consulta de la lista Saved queries (Consultas guardadas) . Aparece en el editor de consultas.

5. Modifique la consulta. Si necesita ejecutarla para comprobar su trabajo, elija Run query (Ejecutar consulta).
6. Cuando esté listo para guardar la nueva versión, elija Actions (Acciones), Save as (Guardar como).
7. Escriba un nombre para la consulta.
8. (Opcional) Elija una carpeta en la que desee guardar la consulta. Seleccione Create new (Crear nueva) para crear una carpeta. Si crea una carpeta nueva, puede utilizar caracteres de barra (/) en el nombre de la carpeta para definir una estructura de carpetas. Por ejemplo, poner nombre a una carpeta nueva **`folder-level-1/folder-level-2`** crea una carpeta de nivel superior llamada **`folder-level-1`**, con otra carpeta llamada **`folder-level-2`** dentro de esa carpeta. La consulta se guarda en **`folder-level-2`**.
9. (Opcional) Cambie los grupos de registro de la consulta o el texto de la consulta.
10. Seleccione Guardar.

Para eliminar una consulta, debe haber iniciado sesión en un rol que tenga el permiso `logs:DeleteQueryDefinition`.

Para editar o eliminar una consulta guardada

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. A la derecha, elija Queries (Consultas).
4. Seleccione la consulta de la lista Saved queries (Consultas guardadas) . Aparece en el editor de consultas.
5. Elija Actions (Acciones), Edit (Editar) o Actions (Acciones), Delete (Eliminar).

Agregar consulta al panel o exportar resultados de consultas

Después de ejecutar una consulta, puede añadirla a un CloudWatch panel o copiar los resultados al portapapeles.

Las consultas agregadas a los paneles se ejecutan automáticamente cada vez que carga el panel y cada vez que el panel se actualiza. Estas consultas cuentan para el límite de 30 consultas simultáneas de CloudWatch Logs Insights.

Para añadir resultados de consultas a un panel

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. Elija uno o varios grupos de registros y ejecute una consulta.
4. Elija Add to dashboard (Añadir a panel).
5. Seleccione el panel o elija Create new (Crear nuevo) para crear un nuevo panel para los resultados de la consulta.
6. Seleccione el tipo de widget que desea utilizar para los resultados de la consulta.
7. Escriba un nombre para el widget.
8. Elija Add to dashboard (Añadir a panel).

Para copiar los resultados de la consulta en el portapapeles o descargar los resultados de la consulta

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Logs Insights (Registros de Insights).
3. Elija uno o varios grupos de registros y ejecute una consulta.
4. Elija Export results (Exportar resultados) y, a continuación, elija la opción que desee.

Ver consultas en marcha o historial de consultas

Puede ver las consultas en curso, así como su historial de consultas recientes.

Las consultas que se están ejecutando actualmente incluyen consultas añadidas a un panel. Está limitado a 30 consultas simultáneas de CloudWatch Logs Insights por cuenta, incluidas las consultas añadidas a los paneles de control.

Para ver su historial de consultas recientes

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Registros y, luego, Información de registros.
3. Elija Historial si utiliza el nuevo diseño de la consola CloudWatch Logs. Si está utilizando el diseño antiguo, elija Actions (Acciones), View query history for this account (Ver historial de consultas de esta cuenta).

Aparece una lista de consultas recientes. Puede volver a ejecutar cualquiera de ellas seleccionando la consulta y eligiendo Run (Ejecutar).

En Estado, CloudWatch los registros muestran En curso para todas las consultas que se estén ejecutando actualmente.

Cifre los resultados de la consulta con AWS Key Management Service

De forma predeterminada, CloudWatch Logs cifra los resultados almacenados de sus consultas de CloudWatch Logs Insights mediante el método de cifrado predeterminado del servidor de CloudWatch Logs. En su lugar, puede optar por utilizar una AWS KMS clave para cifrar estos resultados. Si asocias una AWS KMS clave a los resultados de cifrado, CloudWatch Logs utilizará esa clave para cifrar los resultados almacenados de todas las consultas de la cuenta.

Si posteriormente desasocias la clave de los resultados de la consulta, CloudWatch Logs volverá al método de cifrado predeterminado para consultas posteriores. Sin embargo, las consultas que se ejecutaron mientras la clave estaba asociada siguen cifradas con esa clave. CloudWatch Los registros pueden seguir devolviendo esos resultados una vez desasociada la clave de KMS, ya que CloudWatch los registros pueden seguir haciendo referencia a la clave. Sin embargo, si la clave se deshabilita posteriormente, CloudWatch Logs no podrá leer los resultados de la consulta que se cifraron con esa clave.

> **Important**
>
> CloudWatch Logs solo admite claves KMS simétricas. No utilice una clave asimétrica para cifrar los resultados de la consulta. Para obtener más información, consulte Utilización de claves simétricas y asimétricas.

Límites

- Para realizar los siguientes pasos, debe tener los siguientes permisos: `kms:CreateKey`, `kms:GetKeyPolicy` y `kms:PutKeyPolicy`.
- Después de asociar una clave a los resultados de la consulta o desasociarla de ellos, la operación puede tardar hasta cinco minutos en surtir efecto.

- Si revoca el acceso de CloudWatch los registros a una clave asociada o elimina una clave de KMS asociada, los datos cifrados de los CloudWatch registros ya no se podrán recuperar.
- No puedes usar la CloudWatch consola para asociar una clave, debes usar la API AWS CLI o CloudWatch Logs.

Paso 1: Crea una AWS KMS key

Para crear una clave de KMS, utilice el siguiente comando create-key:

```
aws kms create-key
```

La salida contiene la ID de clave y el nombre de recurso de Amazon (ARN) de la clave. A continuación, se muestra un ejemplo de la salida:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Paso 2: establecer permisos en la clave de KMS

De forma predeterminada, todas las claves de KMS son privadas. Solo el propietario del recurso puede utilizarla para cifrar y descifrar datos. Sin embargo, el propietario del recurso puede conceder permisos para que otros usuarios y recursos accedan a la clave. Con este paso, le das permiso al

director del servicio de CloudWatch registros para usar la clave. El principal de este servicio debe estar en la misma AWS región en la que se almacena la clave.

Como práctica recomendada, le recomendamos que restrinja el uso de la clave solo a las AWS cuentas que especifique.

En primer lugar, guarde la política predeterminada para su clave KMS `policy.json` mediante el siguiente get-key-policycomando:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Abra el archivo `policy.json` en un editor de texto y agregue la sección en negrita desde una de las instrucciones siguientes. Separe la instrucción existente de la nueva instrucción con una coma. Estas instrucciones utilizan `Condition` secciones para mejorar la seguridad de la AWS KMS clave. Para obtener más información, consulte AWS KMS claves y contexto de cifrado.

La `Condition` sección de este ejemplo limita el uso de la AWS KMS clave para los resultados de la consulta de CloudWatch Logs Insights en la cuenta especificada.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
```

```
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:logs:region:account_ID:query-result:*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "Your_account_ID"
                }
            }
        }
    ]
}
```

Por último, añada la política actualizada mediante el siguiente put-key-policycomando:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Paso 3: asociar una clave de KMS a los resultados de la consulta

Para asociar la clave de KMS a los resultados de la consulta en la cuenta

Utilice el comando disassociate-kms-key como se indica a continuación:

```
aws logs associate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-
result:*" --kms-key-id "key-arn"
```

Paso 4: desasociar una clave de los resultados de la consulta en la cuenta

Para desasociar la clave KMS asociada a los resultados de la consulta, utilice el siguiente disassociate-kms-keycomando:

```
aws logs disassociate-kms-key --resource-identifier "arn:aws:logs:region:account-
id:query-result:*"
```

Utilice un lenguaje natural para generar y actualizar las consultas de CloudWatch Logs Insights

> **Note**
>
> Esta función está generalmente disponible en EE. UU. Este (Norte de Virginia), EE. UU. Oeste (Oregón) y Asia-Pacífico (Tokio) para CloudWatch los registros.

CloudWatch Logs admite una función de consulta en lenguaje natural para ayudarte a generar y actualizar consultas para CloudWatch Logs Insights y CloudWatch Metrics Insights.

Con esta función, puede hacer preguntas sobre los datos de CloudWatch Logs que busca o describirlos en un lenguaje sencillo. La función de lenguaje natural genera una consulta basada en un mensaje que usted introduce y proporciona una line-by-line explicación de cómo funciona la consulta. También puede actualizar la consulta para investigar más a fondo los datos.

Según el entorno, puede introducir preguntas como «¿Cuáles son las 100 direcciones IP de origen principales por bytes transferidos?» y «Encuentre las 10 solicitudes de función Lambda más lentas».

Para generar una consulta de CloudWatch Logs Insights con esta capacidad, abra el editor de consultas de CloudWatch Logs Insights, seleccione el grupo de registros que desee consultar y elija Generar consulta.

> **Important**
>
> Para utilizar la función de consulta en lenguaje natural, debe utilizar la ReadOnlyAccesspolítica CloudWatchLogsFullAccessCloudWatchLogsReadOnlyAccessAdministratorAccess,, o. También puede incluir la acción `cloudwatch:GenerateQuery` en una política integrada o administrada por el cliente, nueva o existente.

Consultas de ejemplo

Los ejemplos en esta sección describen cómo generar y actualizar consultas mediante la función de lenguaje natural.

> **Note**
>
> Para obtener más información sobre el editor de consultas y la sintaxis de CloudWatch Logs Insights, consulte Sintaxis de consultas de CloudWatch Logs Insights.

Ejemplo: generar una consulta en lenguaje natural

Para generar una consulta en lenguaje natural, introduzca una petición y seleccione Generar nueva consulta. En este ejemplo se muestra una consulta que realiza una búsqueda básica.

Prompt

A continuación, se muestra un ejemplo de un indicador que indica la capacidad de buscar las 10 invocaciones de funciones Lambda más lentas.

```
Find the 10 slowest requests
```

Consultar

El siguiente es un ejemplo de una consulta que la función de lenguaje natural genera según la petición. Observe cómo se muestra la petición en un comentario antes de la consulta. Tras la consulta, puede leer una explicación que describe cómo funciona la consulta.

```
# Find the 10 slowest requests
fields @timestamp, @message, @duration
| sort @duration desc
| limit 10
# This query retrieves the timestamp, message and duration fields from the logs and
 sorts them in descending order by duration to find the 10 slowest requests.
```

> **Note**
>
> Para desactivar el aspecto de la petición y la explicación de cómo funciona la consulta, use el icono de engranaje del editor.

Ejemplo: actualizar una consulta en lenguaje natural

Puede actualizar una consulta al editar la petición inicial y, a continuación, seleccionar Actualizar consulta.

Petición actualizada

El siguiente ejemplo muestra una versión actualizada de la petición anterior. En lugar de una solicitud que busca las 10 invocaciones de funciones de Lambda más lentas, esta solicitud ahora dirige la capacidad de buscar las 20 invocaciones de funciones de Lambda más lentas e incluye otra columna para eventos de registro adicionales.

```
Show top 20 slowest requests instead and display requestId as a column
```

Consulta actualizada

A continuación, se muestra un ejemplo de una consulta actualizada. Observe cómo se muestra la petición actualizada en un comentario antes de la consulta actualizada. Tras la consulta, puede leer una explicación que describe cómo se actualizó la consulta original.

```
# Show top 20 slowest requests instead and display requestId as a column
fields @timestamp, @message, @requestId, @duration
| sort @duration desc
| limit 20
# This query modifies the original query by replacing the @message field with the
 @requestId field and changing the limit from 10 to 20 to return the top 20 log events
 by duration instead of the top 10.
```

Optar por no utilizar sus datos para mejorar el servicio

Los datos de la petición en lenguaje natural que proporciona para entrenar el modelo de IA y generar consultas relevantes se utilizan únicamente para proporcionar y mantener su servicio. Estos datos podrían usarse para mejorar la calidad de Logs Insights. CloudWatch La confianza y privacidad, como así también la seguridad de su contenido, son nuestra máxima prioridad. Para obtener más información, consulte Condiciones del servicio de AWS y Política de IA responsable de AWS.

Puede optar por que su contenido no se utilice para desarrollar o mejorar la calidad de las consultas en lenguaje natural mediante la creación de una política de exclusión de los servicios de IA. Para excluirse de la recopilación de datos para todas las funciones de CloudWatch Logs AI, incluida la

capacidad de generación de consultas, debe crear una política de exclusión para CloudWatch Logs. Para obtener más información, consulte Políticas de exclusión de servicios de IA en la Guía del usuario de AWS Organizations .

Detección de anomalías de registro

Puede crear un detector de anomalías de registro para cada grupo de registros. El detector de anomalías escanea los eventos de registro introducidos en el grupo de registros y encuentra anomalías en los datos del registro. La detección de anomalías utiliza el aprendizaje automático y el reconocimiento de patrones para establecer las líneas base del contenido típico de los registros.

Después de crear un detector de anomalías para un grupo de registros, se entrena utilizando los eventos de registro de las últimas dos semanas en el grupo de registros para el entrenamiento. El período de entrenamiento puede durar hasta 15 minutos. Una vez finalizada la formación, comienza a analizar los registros entrantes para identificar las anomalías, que se muestran en la consola de CloudWatch registros para que las pueda examinar.

CloudWatch El reconocimiento de patrones de registros extrae los patrones de registro al identificar el contenido estático y dinámico de los registros. Los patrones son útiles para analizar conjuntos de registros grandes porque, a menudo, una gran cantidad de eventos de registro se pueden comprimir en unos pocos patrones.

Por ejemplo, consulte el siguiente ejemplo de tres eventos de registro.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

En el ejemplo anterior, los tres eventos de registro siguen un patrón:

```
<*> <*> [INFO] Calling DynamoDB to store for resource id <*>
```

Los campos dentro de un patrón se denominan fichas. Los campos que varían dentro de un patrón, como un identificador de solicitud o una marca de tiempo, se denominan símbolos dinámicos. Los símbolos dinámicos se representan <*> cuando CloudWatch Logs muestra el patrón. Cada valor diferente encontrado para un token dinámico se denomina valor de token.

Algunos ejemplos comunes de tokens dinámicos son los códigos de error, las marcas de tiempo y los ID de solicitud.

La detección de anomalías en los registros utiliza estos patrones para encontrar anomalías.
Tras el período de entrenamiento del modelo de detector de anomalías, los registros se evalúan

comparándolos con las tendencias conocidas. El detector de anomalías marca las fluctuaciones significativas como anomalías.

La creación de detectores de anomalías logarítmicas no conlleva cargos.

Gravedad y prioridad de las anomalías y los patrones

A cada anomalía detectada por un detector de anomalías logarítmicas se le asigna una prioridad. A cada patrón encontrado se le asigna una gravedad.

- La prioridad se calcula automáticamente y se basa tanto en el nivel de gravedad del patrón como en la cantidad de desviación con respecto a los valores esperados. Por ejemplo, si un determinado valor simbólico aumenta repentinamente un 500%, esa anomalía podría designarse como `HIGH` prioritaria aunque su gravedad lo fuera. `NONE`
- La gravedad se basa únicamente en las palabras clave que se encuentran en los patrones`FATAL`, como`ERROR`, y. `WARN` Si no se encuentra ninguna de estas palabras clave, la gravedad del patrón se marca como`NONE`.

Tiempo de visibilidad de la anomalía

Al crear un detector de anomalías, se especifica el período máximo de visibilidad de anomalías para dicho detector. Es el número de días que la anomalía se muestra en la consola y la devuelve la operación de la ListAnomaliesAPI. Una vez transcurrido este período de tiempo, una anomalía, si continúa ocurriendo, se acepta automáticamente como un comportamiento normal y el modelo de detector de anomalías deja de marcarla como una anomalía.

Si no ajusta el tiempo de visibilidad al crear un detector de anomalías, se utilizan 21 días de forma predeterminada.

Suprimir una anomalía

Una vez detectada una anomalía, puede optar por suprimirla temporal o permanentemente. Al suprimir una anomalía, el detector de anomalías deja de marcar la incidencia como una anomalía durante el tiempo que especifique. Al suprimir una anomalía, puede optar por suprimir solo esa anomalía específica o suprimir todas las anomalías relacionadas con el patrón en el que se encontró la anomalía.

Aún puede ver las anomalías suprimidas en la consola. También puede optar por dejar de suprimirlas.

Preguntas frecuentes

¿ AWS Utilizo mis datos para entrenar algoritmos de aprendizaje automático para AWS su uso o para otros clientes?

No. El modelo de detección de anomalías creado por la capacitación se basa en los eventos de registro de un grupo de registros y solo se usa dentro de ese grupo de registros y esa AWS cuenta.

¿Qué tipos de eventos de registro funcionan bien con la detección de anomalías?

La detección de anomalías en los registros es adecuada para: los registros de aplicaciones y otros tipos de registros en los que la mayoría de las entradas de registro se ajustan a los patrones típicos. Los grupos de registros con eventos que contienen un nivel de registro o palabras clave de gravedad, como INFO, ERROR y DEBUG, son especialmente adecuados para la detección de anomalías de registro.

La detección de anomalías en el registro no es adecuada para: Registrar eventos con estructuras JSON extremadamente largas, como los registros. CloudTrail El análisis de patrones analiza solo los primeros 1500 caracteres de una línea de registro, por lo que se omite cualquier carácter que supere ese límite.

Los registros de auditoría o acceso, como los registros de flujo de VPC, también tendrán menos éxito con la detección de anomalías. El objetivo de la detección de anomalías es detectar problemas en las aplicaciones, por lo que es posible que no sea adecuada para las anomalías de acceso o de red.

Para ayudarle a determinar si un detector de anomalías es adecuado para un grupo de registros determinado, utilice el análisis de patrones de CloudWatch registros para encontrar el número de patrones en los eventos de registro del grupo. Si el número de patrones no es superior a unos 300, la detección de anomalías podría funcionar bien. Para obtener más información sobre el análisis de patrones, consulteAnálisis de patrones.

¿Qué se marca como anomalía?

Las siguientes incidencias pueden provocar que un evento de registro se marque como una anomalía:

- Un evento de registro con un patrón que no se había visto antes en el grupo de registros.

- Una variación significativa de un patrón conocido.
- Un valor nuevo para un token dinámico que tiene un conjunto discreto de valores habituales.
- Un cambio importante en el número de apariciones de un valor de un token dinámico.

Si bien todos los elementos anteriores pueden estar marcados como anomalías, no todos significan que la aplicación esté funcionando mal. Por ejemplo, higher-than-usual varios valores de `200` éxito pueden marcarse como anomalías. En casos como este, podría considerar la posibilidad de suprimir estas anomalías que no indiquen problemas.

¿Qué ocurre con los datos confidenciales que se ocultan?

Las partes del registro de eventos que estén enmascaradas como datos confidenciales no se escanean para detectar anomalías. Para obtener más información sobre cómo enmascarar datos confidenciales, consulte Ayudar a proteger los datos de registro confidenciales mediante el enmascaramiento.

Habilite la detección de anomalías en un grupo de registros

Siga estos pasos para usar la CloudWatch consola y crear un detector de anomalías de registro que analice un grupo de registros en busca de anomalías.

También puede crear detectores de anomalías mediante programación. Para obtener más información, consulte. CreateLogAnomalyDetector

Para crear un detector de anomalías en el registro

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. Elija Registros, Registre anomalías.
3. Elija Crear detector de anomalías.
4. Seleccione el grupo de registros para el que desee crear este detector de anomalías.
5. Introduzca un nombre para el detector en Nombre del detector de anomalías.
6. (Opcional) Cambie la frecuencia de evaluación del valor predeterminado de 5 minutos. Establezca este valor de acuerdo con la frecuencia con la que el grupo de registros recibe nuevos registros. Por ejemplo, si el grupo de registros recibe nuevos eventos de registro en lotes cada 10 minutos, podría ser adecuado establecer la frecuencia de evaluación en 15 minutos.
7. (Opcional) Para configurar el detector de anomalías para que busque anomalías únicamente en los eventos de registro que contengan determinadas palabras o cadenas, elija Filtrar patrones.

A continuación, introduzca un patrón en el patrón del filtro de detección de anomalías. Para obtener más información sobre la sintaxis de los patrones,Filtro de sintaxis de patrones para filtros de métricas, filtros de suscripción, eventos de registro de filtros y Live Tail.

(Opcional) Para probar el patrón de filtro, introduzca algunos mensajes de registro en Registrar mensajes de eventos y, a continuación, seleccione Probar patrón.

8. (Opcional) Para cambiar el período de visibilidad de las anomalías con respecto al predeterminado o para asociar una AWS KMS clave a este detector de anomalías, seleccione Configuración avanzada.

   a. Para cambiar el período de visibilidad de las anomalías con respecto al predeterminado, introduzca un nuevo valor en Período máximo de visibilidad de las anomalías (días).

   b. Para asociar una AWS KMS clave a este detector de anomalías, introduzca el ARN en el ARN de la clave KMS. Si asigna una clave, la información de anomalías detectada por este detector se cifra en reposo con la clave. Los usuarios deben tener permisos para utilizar esta clave y para que el detector de anomalías recupere información sobre las anomalías que encuentre.

      También debe asegurarse de que el director del servicio de CloudWatch registros tenga permiso para usar la clave. Para obtener más información, consulte Cifre un detector de anomalías y sus resultados con AWS KMS.

9. Seleccione Activar la detección de anomalías.

   Se crea el detector de anomalías y comienza a entrenar su modelo en función de los eventos de registro que ingiere el grupo de registros. Transcurridos unos 15 minutos, la detección de anomalías se activa y comienza a detectar anomalías y a descubrirlas.

Vea las anomalías que se han encontrado

Tras crear uno o más detectores de anomalías de registro, puede utilizar la CloudWatch consola para ver las anomalías que hayan encontrado.

Puede ver las anomalías mediante programación. Para obtener más información, consulte. ListAnomalies

Para ver las anomalías detectadas por todos sus detectores de anomalías de registro

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. Elija Registros, Registre anomalías.

   Aparece la tabla de anomalías de los registros. El número que aparece en la parte superior, junto a las anomalías de registro, muestra el número de anomalías de registro que aparecen en la tabla. En cada fila de la tabla se muestra la siguiente información:

   - La columna Anomalía muestra un breve resumen de la anomalía. Estos resúmenes los generan los registros. CloudWatch
   - La prioridad de la anomalía. La prioridad se calcula automáticamente en función de la cantidad de cambios en el registro de eventos, palabras clave como `Exception` que se producen en un evento de registro, etc.
   - El patrón de registro en el que se basa la anomalía. Para obtener más información sobre los patrones, consulteDetección de anomalías de registro.
   - La tendencia del registro de anomalías muestra un histograma que muestra el volumen de registros que coinciden con el patrón.
   - La hora de la última detección muestra la última vez que se encontró la anomalía.
   - La hora de la primera detección muestra la primera vez que se encontró la anomalía.
   - El detector de anomalías muestra el nombre del grupo de registros que contiene los eventos de registro relacionados con esta anomalía. Puede elegir este nombre para ver la página de detalles del grupo de registros.
3. Para seguir inspeccionando una anomalía, pulse el botón de radio de la fila correspondiente.

   Aparece el panel de inspección de patrones y muestra lo siguiente:

   - El patrón en el que se basa esta anomalía. Seleccione un token dentro del patrón para analizar los valores de ese token.
   - Un histograma que muestra el número de apariciones de la anomalía en el intervalo de tiempo consultado.
   - La pestaña Registrar muestras muestra algunos de los eventos de registro que forman parte de la anomalía.
   - La pestaña Valores del token muestra los valores del token dinámico seleccionado, si ha seleccionado uno.

> **Note**
>
> Se captura un máximo de 10 valores de token para cada token. Es posible que los recuentos de fichas no sean precisos. CloudWatch Logs utiliza un contador probabilístico para generar el recuento de fichas, no el valor absoluto.

4. Para suprimir una anomalía, pulse el botón de radio de su fila y, a continuación, haga lo siguiente:

   a. Seleccione Acciones y suprima la anomalía.

   b. A continuación, especifique durante cuánto tiempo desea que se suprima la anomalía.

   c. Para suprimir todas las anomalías relacionadas con este patrón, seleccione Suprimir patrón.

   d. Seleccione Suprimir anomalía.

Para ver las anomalías encontradas en un solo grupo de registros

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. Elija Registros, Grupos de registro.
3. Elija el nombre de un grupo de registros y, a continuación, elija la pestaña Detección de anomalías.

   Aparece la tabla de detección de anomalías. El número que aparece en la parte superior, junto a las anomalías de registro, muestra el número de anomalías de registro que aparecen en la tabla. En cada fila de la tabla se muestra la siguiente información:

   - La columna Anomalía muestra un breve resumen de la anomalía. Estos resúmenes los generan los registros. CloudWatch
   - La prioridad de la anomalía. La prioridad se calcula automáticamente en función de la cantidad de cambios en el registro de eventos, palabras clave como `Exception` que se producen en un evento de registro, etc.
   - El patrón de registro en el que se basa la anomalía. Para obtener más información sobre los patrones, consulteDetección de anomalías de registro.
   - La tendencia del registro de anomalías muestra un histograma que muestra el volumen de registros que coinciden con el patrón.
   - La hora de la última detección muestra la última vez que se encontró la anomalía.

- La hora de la primera detección muestra la primera vez que se encontró la anomalía.

4. Para seguir inspeccionando una anomalía, pulse el botón de radio de la fila correspondiente.

   Aparece el panel de inspección de patrones y muestra lo siguiente:

   - El patrón en el que se basa esta anomalía. Seleccione un token dentro del patrón para analizar los valores de ese token.
   - Un histograma que muestra el número de apariciones de la anomalía en el intervalo de tiempo consultado.
   - La pestaña Registrar muestras muestra algunos de los eventos de registro que forman parte de la anomalía.
   - La pestaña Valores del token muestra los valores del token dinámico seleccionado, si ha seleccionado uno.

   > **Note**
   >
   > Se captura un máximo de 10 valores de token para cada token. Es posible que los recuentos de fichas no sean precisos. CloudWatch Logs utiliza un contador probabilístico para generar el recuento de fichas, no el valor absoluto.

5. Para suprimir una anomalía, pulse el botón de radio de su fila y, a continuación, haga lo siguiente:

   a. Seleccione Acciones y suprima la anomalía.
   b. A continuación, especifique durante cuánto tiempo desea que se suprima la anomalía.
   c. Para suprimir todas las anomalías relacionadas con este patrón, seleccione Suprimir patrón.
   d. Seleccione Suprimir anomalía.

Cree alarmas en los detectores de anomalías de registro

Puede crear una alarma para un detector de anomalías de registro de un grupo de registros. Puede especificar que la alarma entre en `ALARM` estado cuando se encuentre un número específico de anomalías en el grupo de registros durante un período de tiempo específico. También puede usar filtros para que la alarma solo cuente las anomalías con prioridades específicas.

Para crear una alarma para un detector de anomalías de registro

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Registros, Registrar anomalías.

   Aparece la tabla de detectores de anomalías de registro.
3. Seleccione el botón de radio del detector de anomalías para el que desee configurar la alarma y elija Crear alarma.

   Aparece el asistente de creación de CloudWatch alarmas. El LogAnomalyDetectorcampo muestra el nombre del detector de anomalías que haya elegido. Aparece AnomalyCountel campo Nombre de la métrica.
4. (Opcional) Para filtrar esta alarma en función de la prioridad de las anomalías, realice una de las siguientes acciones:
   - Para que la alarma contabilice solo las anomalías de alta prioridad, introduzca for. **`HIGH`** LogAnomalyPriority
   - Para que la alarma contabilice únicamente las anomalías de prioridad alta y media, introduzca for. **`MEDIUM`** LogAnomalyPriority

   Para obtener más información sobre los niveles de prioridad, consulte. Gravedad y prioridad de las anomalías y los patrones
5. Elija utilizar un umbral de detección de anomalías estático o métrico para la alarma. Esta selección determina cómo se establece el umbral de alarma. Un umbral estático significa que el umbral de alarma es un número estático y constante que usted elige. Un umbral de detección de anomalías significa que CloudWatch determina un rango de valores habituales y que la alarma se activa si el recuento real supera el umbral de esta banda. No es necesario seleccionar la detección de anomalías para crear una alarma de detección de anomalías de registro. Para obtener más información sobre la detección de anomalías métricas, consulte Uso CloudWatch de la detección de anomalías.
6. Para siempre que ***`your-metric-name`***sea... , elija Mayor, Mayor/Igual, Inferior o Igual o Inferior. En que . . ., especifique un número para el valor del umbral. La alarma pasa a estar activa **`ALARM`** si el detector de anomalías encuentra más alarmas que este número de alarmas durante un período especificado por el período.
7. Elija Configuración adicional. Para Puntos de datos para alarma, especifique el número de periodos de evaluación (puntos de datos) que deben tener el estado `ALARM` para que se active

la alarma. Si estos dos valores coinciden, creerá una alarma que pasará al estado `ALARM` si se infringen muchos periodos consecutivos.

Para crear una alarma M de N, especifique un número menor para el primer valor que el especificado para el segundo valor. Para obtener más información, consulte Evaluación de una alarma.

8. En Tratamiento de datos que faltan, elija cómo debe comportarse la alarma cuando falten algunos puntos de datos. Para obtener más información, consulte Configurar el modo en que CloudWatch las alarmas tratan los datos faltantes.
9. Elija Siguiente.
10. En Notificación, selecciona Añadir notificación y, a continuación, especifica un tema de Amazon SNS para notificarlo cuando la alarma pase al estado `ALARM`0K, `o`INSUFFICIENT_DATA.

    a. (Opcional) Para que la alarma envíe varias notificaciones para el mismo estado de alarma o para estados de alarma diferentes, elija Añadir notificación.

    > **Note**
    >
    > Le recomendamos que configure la alarma para que tome medidas cuando pase al estado de datos insuficientes, además de cuando pase al estado de alarma. Esto se debe a que muchos problemas con la función de Lambda que se conecta al origen de datos pueden provocar que la alarma pase a datos insuficientes.

    b. (Opcional) Para que no envíe notificaciones de Amazon SNS, elija Eliminar.
11. (Opcional) Si desea que la alarma realice acciones para Amazon EC2 Auto Scaling, Amazon EC2 o tickets, AWS Systems Manager o bien, elija el botón correspondiente y especifique el estado y la acción de la alarma.

    > **Note**
    >
    > La alarma solo puede llevar a cabo acciones de Systems Manager cuando su estado es `ALARM`. Para obtener información sobre las acciones de Systems Manager, consulte Configuración CloudWatch para crear OpsItems y Creación de incidentes.

12. Elija Siguiente.
13. En Nombre y descripción, escriba el nombre y la descripción de la alarma y elija Siguiente. El nombre debe contener solo caracteres UTF-8 y no puede contener caracteres de control ASCII.

La descripción puede incluir un formato de rebajas, que solo se muestra en la pestaña Detalles de la alarma de la CloudWatch consola. Markdown puede resultar útil para añadir enlaces a manuales u otros recursos internos.

> **Tip**
>
> El nombre de alarma solo debe contener caracteres UTF-8. No puede contener caracteres de control ASCII.

14. En Obtener vista previa y crear, confirme que la información y las condiciones son las correctas y luego, elija Crear alarma.

Métricas publicadas por los detectores de anomalías de registro

CloudWatch Logs publica la AnomalyCountmétrica en CloudWatch métricas. Esta métrica se publica en el espacio de `AWS/Logs` nombres.

La AnomalyCountmétrica se publica con las siguientes dimensiones:

- LogAnomalyDetector— El nombre del detector de anomalías
- LogAnomalyPriority— El nivel de prioridad de la anomalía

Cifre un detector de anomalías y sus resultados con AWS KMS

Los datos del detector de anomalías siempre se cifran en los registros. CloudWatch De forma predeterminada, CloudWatch Logs utiliza el cifrado del lado del servidor para los datos en reposo. Como alternativa, puede utilizar AWS Key Management Service para este cifrado. Si lo hace, el cifrado se realiza mediante una AWS KMS clave. AWS KMS El uso del cifrado se habilita en el nivel del detector de anomalías, asociando una clave KMS a un detector de anomalías.

> **Important**
>
> CloudWatch Los registros solo admiten claves KMS simétricas. No utilice una clave asimétrica administrada por el cliente para cifrar los datos de los grupos de registros. Para obtener más información, consulte Utilización de claves simétricas y asimétricas.

Límites

- Para realizar los siguientes pasos, debe tener los siguientes permisos: `kms:CreateKey`, `kms:GetKeyPolicy` y `kms:PutKeyPolicy`.
- Tras asociar o desasociar una clave de un detector de anomalías, la operación puede tardar hasta cinco minutos en surtir efecto.
- Si revoca el acceso de CloudWatch los registros a una clave asociada o elimina una clave de KMS asociada, los datos cifrados de los CloudWatch registros ya no se podrán recuperar.

Paso 1: Crea una clave AWS KMS

Para crear una clave de KMS, utilice el siguiente comando create-key:

```
aws kms create-key
```

La salida contiene la ID de clave y el nombre de recurso de Amazon (ARN) de la clave. A continuación, se muestra un ejemplo de la salida:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "key-default-1",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/key-default-1",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Paso 2: establecer permisos en la clave de KMS

De forma predeterminada, todas AWS KMS las claves son privadas. Solo el propietario del recurso puede utilizarla para cifrar y descifrar datos. Sin embargo, el propietario del recurso puede conceder permisos para que otros usuarios y recursos accedan a la clave de KMS. Con este paso, concedes al servicio de CloudWatch registros el permiso principal para usar la clave. El principal de este servicio debe estar en la misma AWS región en la que se almacena la clave de KMS.

Como práctica recomendada, le recomendamos que restrinja el uso de la clave KMS únicamente a las AWS cuentas o detectores de anomalías que especifique.

En primer lugar, guarde la política predeterminada para su clave KMS `policy.json` mediante el siguiente get-key-policycomando:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Abra el archivo `policy.json` en un editor de texto y agregue la sección en negrita desde una de las instrucciones siguientes. Separe la instrucción existente de la nueva instrucción con una coma. Estas instrucciones utilizan `Condition` secciones para mejorar la seguridad de la AWS KMS clave. Para obtener más información, consulte AWS KMS claves y contexto de cifrado.

La `Condition` sección de este ejemplo limita el uso de la AWS KMS clave a la cuenta especificada, pero se puede usar para cualquier detector de anomalías.

```
{
  "Version": "2012-10-17",
  "Id": "key-default-1",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::Your_account_ID:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "logs.REGION.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "kms:EncryptionContext:aws:logs:arn":
 "arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
        }
      }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.REGION.amazonaws.com"
        },
        "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
        ],
        "Resource": "*",
        "Condition": {
            "ArnLike": {
                "kms:EncryptionContext:aws-crypto-ec:aws:logs:arn":
 "arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
            }
        }
    }
  ]
}
```

Por último, añada la política actualizada mediante el siguiente put-key-policycomando:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Paso 3: Asocie una clave KMS a un detector de anomalías

Puede asociar una clave KMS a un detector de anomalías al crearla en la consola o mediante las API AWS CLI .

Paso 4: Desasociar la clave de un detector de anomalías

Una vez que se ha asociado una clave a un detector de anomalías, no se puede actualizar. La única forma de eliminar la clave es eliminar el detector de anomalías y, a continuación, volver a crearlo.

Trabajar con grupos de registros y flujos de registros

Un flujo de registro es una secuencia de eventos de registro que comparten la misma fuente. Cada fuente independiente de CloudWatch registros de Logs constituye un flujo de registros independiente.

Un grupo de registros es un grupo de flujos de registro que comparten la misma configuración de retención, monitoreo y control de acceso. Puede definir grupos de registros y especificar los flujos que deben incluirse en cada uno. No hay límites en el número de flujos de registros que pueden pertenecer a un grupo de registros.

Utilice los procedimientos de esta sección para trabajar con grupos y flujos de registros.

Cree un grupo de CloudWatch registros en Logs

Al instalar el agente de CloudWatch Logs en una instancia de Amazon EC2 siguiendo los pasos de las secciones anteriores de la Guía del usuario de Amazon CloudWatch Logs, el grupo de registros se crea como parte de ese proceso. También puede crear un grupo de registros directamente en la CloudWatch consola.

Para crear un grupo de registros

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, selecione Grupos de registro.
3. Elija Actions (Acciones) y, a continuación, elija Create log group (Crear grupo de registros).
4. Escriba el nombre del grupo de registros y, a continuación, seleccione Crear grupo de registros.

> **Tip**
>
> Puede marcar como favoritos a grupos de registros, así como paneles y alarmas, desde el menú Favorites and recents (Favoritos y recientes) del panel de navegación. En la columna Visitados recientemente, desplácese sobre el grupo de registros que desea marcar como favoritos y elija el símbolo de estrella junto a este.

Enviar registros a un grupo de registros

CloudWatch Logs recibe automáticamente los eventos de registro de varios AWS servicios. También puede enviar otros eventos de registro a CloudWatch Logs mediante uno de los siguientes métodos:

- CloudWatch agente: el CloudWatch agente unificado puede enviar métricas y CloudWatch registros a Logs. Para obtener información sobre la instalación y el uso del CloudWatch agente, consulte Recopilación de métricas y registros de instancias de Amazon EC2 y servidores locales con el CloudWatch agente en la Guía del usuario de Amazon CloudWatch .
- AWS CLIput-log-events—Carga lotes de eventos de registro a Logs. CloudWatch
- Mediante programación: la PutLogEventsAPI le permite cargar lotes de eventos de registro a Logs de forma programática. CloudWatch

Vea los datos de registro enviados a Logs CloudWatch

Puede ver los datos de registro y desplazarse por ellos stream-by-stream según los envíe el agente de CloudWatch registros a CloudWatch Logs. Puede especificar el intervalo de tiempo para los datos de registro que desee ver.

Para ver los datos de registro

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Log groups (Grupos de registro).
3. En Log Groups (Grupos de registro), elija el grupo de registros para ver los flujos.
4. En la lista de grupos de registros, elija el nombre del grupo de registros que desea ver.
5. En la lista de flujos de registros, elija el nombre del flujo de registros que desea ver.
6. Para cambiar la forma en que se muestran los datos de registro, lleve a cabo alguna de las siguientes operaciones:

   - Para expandir un único evento de registro, elija la flecha situada junto a ese evento de registro.
   - Para ampliar todos los eventos de registro y verlos como texto sin formato, por encima de la lista de eventos de registro, elija Text (Texto).
   - Para filtrar los eventos de registro, escriba el filtro de búsqueda que desee en el campo de búsqueda. Para obtener más información, consulte Creación de métricas a partir de eventos de registro mediante filtros.

- Para ver los datos de registro de un intervalo de fechas y horas especificado, junto al filtro de búsqueda, elija la flecha situada al lado de la fecha y hora. Para especificar un intervalo de fechas y horas, elija Absolute (Absoluto). Para elegir un número predefinido de minutos, horas, días o semanas, elija Relative (Relativo). También puede cambiar entre zona horaria UTC y zona horaria local.

Use Live Tail para ver los registros casi en tiempo real

CloudWatch Logs Live Tail le ayuda a solucionar rápidamente los incidentes al ver una lista en streaming de los nuevos eventos de registro a medida que se van incorporando. Puede ver, filtrar y destacar los registros incorporados casi en tiempo real, lo que ayuda a detectar y resolver problemas con mayor rapidez. Puede filtrar los registros en función de los términos que especifique y, también, destacar los registros que contienen términos específicos para ayudarlo a encontrar lo que busca con rapidez.

Las sesiones de Live Tail generan costos según el tiempo de uso de la sesión por minuto. Para obtener más información sobre los precios, consulta la pestaña Logs en Amazon CloudWatch Pricing.

> **Note**
>
> Live Tail solo es compatible con los grupos de registros de la clase de registro estándar. Para obtener más información sobre las clases de registro, consulteClases de registro.

En las siguientes secciones se explica cómo usar Live Tail en la consola. También puedes iniciar una sesión de Live Tail mediante programación. Para obtener más información, consulte StartLiveTail. Para ver ejemplos de SDK, consulta Cómo iniciar una sesión de Live Tail con un AWS SDK.

Inicio de una sesión de Live Tail

La CloudWatch consola se utiliza para iniciar una sesión de Live Tail. El siguiente procedimiento explica cómo iniciar una sesión de Live Tail mediante la opción Live Tail en el panel de navegación izquierdo. También puedes iniciar sesiones de Live Tail desde la página de grupos de registros o la página de información de CloudWatch registros.

> **Note**
>
> Si utiliza políticas de protección de datos para enmascarar los datos confidenciales de un grupo de registro al verlo mediante Live Tail, los datos confidenciales siempre aparecerán enmascarados en la sesión. Para obtener más información sobre cómo enmascarar datos en grupos de registro, consulte Ayude a proteger los datos de registro confidenciales con el enmascaramiento.

Para iniciar una sesión de Live Tail

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Registros y, a continuación, Live Tail.
3. En Seleccionar grupos de registro, seleccione los grupos de registro de los que desea ver los eventos en la sesión de Live Tail. Puede seleccionar hasta 10 grupos de registro.
4. (Opcional) Si seleccionó solo un grupo de registro, puede aplicar más filtros en su sesión de Live Tail si selecciona uno o más flujos de registro para ver los eventos. Para ello, en Seleccionar flujos de registro, seleccione los nombres de los flujos de registro en la lista desplegable. También, puede utilizar la segunda casilla situada en Seleccionar flujos de registro para ingresar un prefijo de nombre de flujo de registro y, a continuación, se seleccionarán todos los flujos de registro con nombres que coincidan con ese prefijo.
5. (Opcional) Para mostrar solo los eventos de registro que contengan determinadas palabras u otras cadenas, escriba la palabra o la cadena en `Add filter patterns`.

   Por ejemplo, para mostrar solo los eventos de registro que incluyan la palabra `Warning`, escriba **`Warning`**. El campo de filtros distingue entre mayúsculas y minúsculas. Puede incluir varios términos y operadores de patrones en este campo:

   - **`error 404`** muestra solo los eventos de registro que incluyen `error` y `404`
   - **`?Error ?error`** muestra los eventos de registro que incluyen `Error` o `error`
   - **`-INFO`** muestra todos los eventos de registro que no incluyen `INFO`
   - **`{ $.eventType = "UpdateTrail" }`** muestra todos los eventos de registro JSON donde el valor del campo del tipo de evento es `UpdateTrail`

   También puede usar una expresión regular (regex) para filtrar:

- **`%ERROR%`** usa expresiones regulares para mostrar todos los eventos de registro que consisten en la palabra clave ERROR
- **`{ $.names = %Steve% }`** usa expresiones regulares para mostrar los eventos de registro JSON en los que Steve está en la propiedad `"name"`
- **`[ w1 = %abc%, w2 ]`** usa expresiones regulares para mostrar eventos de registro delimitados por espacios donde la primera palabra es abc

Para obtener más información sobre la sintaxis de patrones, consulte Sintaxis de patrones y filtros.

6. (Opcional) Para buscar y destacar algunos de los eventos de registro mostrados, escriba un término en Live Tail. Ingrese los términos destacados uno por uno. Si agrega varios términos para destacar, se asignará un color diferente para representar cada uno. Se muestra un indicador a la izquierda de cualquier evento de registro que contenga el término especificado y, también, aparece debajo del propio término al expandir el evento de registro en la ventana principal para ver el evento de registro completo.

   Puede utilizar las funciones de filtrar y destacar para solucionar problemas de forma rápida. Por ejemplo, puede filtrar los eventos para mostrar solo los eventos que contienen `Error` y, a continuación, también destacar los eventos que contienen `404`.

7. Para iniciar la sesión, seleccione Aplicar filtros

   Los eventos de registro que coincidan comenzarán a aparecer en la ventana. También se muestra la siguiente información:

   - El temporizador muestra el tiempo de actividad de la sesión de Live Tail.
   - eventos/seg muestra cuántos eventos de registro incorporados por segundo coinciden con los filtros establecidos.
   - Para evitar que la sesión se desplace demasiado rápido porque muchos eventos coinciden con los filtros, es posible que los CloudWatch registros muestren solo algunos eventos coincidentes. Si esto ocurre, el porcentaje de eventos coincidentes que se verá en la pantalla se muestra en % mostrado.

8. Para pausar el flujo de eventos e investigar lo que se muestra en el momento, haga clic en cualquier parte de la ventana de eventos.
9. Durante la sesión, puede realizar lo siguiente para ver más detalles sobre cada evento de registro.

- Para mostrar el texto completo de un evento de registro en la ventana principal, seleccione la flecha situada junto a ese evento de registro.
- Para mostrar el texto completo de un evento de registro en una ventana lateral, elija el ícono + situado junto a ese evento de registro. El flujo de eventos se detiene y aparece una ventana lateral.

  Mostrar el texto de un evento de registro en una ventana lateral puede resultar útil para comparar su texto con otros eventos en la ventana principal.

10. Para detener la sesión de Live Tail, seleccione Detener.
11. Para reiniciar la sesión, si lo desea, utilice el panel Filtro para modificar los criterios de filtrado y elija Aplicar filtros. A continuación, elija Start (Inicio).

Búsqueda de datos de registro mediante patrones de filtro

Puede buscar los datos de registro con Filtro de sintaxis de patrones para filtros de métricas, filtros de suscripción, eventos de registro de filtros y Live Tail. Puede buscar en todos los flujos de registro de un grupo de registros o, si lo utiliza, también AWS CLI puede buscar flujos de registro específicos. Cuando se ejecuta cada búsqueda, devuelve hasta la primera página de los datos encontrados y un token para recuperar la siguiente página de datos o para continuar con la búsqueda. Si no se devuelve ningún resultado, puede continuar con la búsqueda.

Puede definir el intervalo de tiempo que desea consultar para limitar el alcance de la búsqueda. Podría comenzar por un intervalo mayor para ver las líneas de registro en las que está interesado y, a continuación, acortar el intervalo de tiempo al ámbito para ver los registros en el intervalo de tiempo que desee.

También puede pasar directamente desde las métricas extraídas de los registros a los registros correspondientes.

Si ha iniciado sesión en una cuenta configurada como una cuenta de monitoreo en el marco de la observabilidad CloudWatch multicuenta, puede buscar y filtrar los eventos de registro de las cuentas de origen vinculadas a esta cuenta de monitoreo. Para obtener más información, consulta la observabilidad CloudWatch entre cuentas.

Búsqueda de entradas de registro con la consola

Puede buscar las entradas de registro que cumplan los criterios especificados mediante la consola.

Para buscar los registros mediante la consola

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Log groups (Grupos de registro).
3. En Log Groups (Grupos de registros), elija el nombre del grupo de registros que contiene el flujo de registros que desea buscar.
4. En Log Streams (Flujos de registros), elija el nombre del flujo de registros que desea buscar.
5. En Log events (Eventos de registros), escriba la sintaxis del filtro que se va a utilizar.

Para buscar todas las entradas de registro durante un intervalo de tiempo mediante la consola

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Log groups (Grupos de registro).
3. En Log Groups (Grupos de registros), elija el nombre del grupo de registros que contiene el flujo de registros que desea buscar.
4. Elija Search Log Group (Buscar grupos de registros).
5. En Log events (Eventos de registros), seleccione el intervalo de fecha y hora e ingrese la sintaxis del filtro.

Busque entradas de registro mediante el AWS CLI

Puede buscar entradas de registro que cumplan un criterio específico mediante el AWS CLI.

Para buscar entradas de registro mediante el AWS CLI

En una línea de comandos, ejecute el siguiente filter-log-eventscomando. Utilice `--filter-pattern` para limitar los resultados al patrón de filtros especificado y `--log-stream-names` para limitar los resultados al flujo de registros especificado.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Para buscar entradas de registro en un intervalo de tiempo determinado mediante el AWS CLI

En una línea de comandos, ejecute el siguiente filter-log-eventscomando:

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--start-time 1482197400000] [--end-
time 1482217558365] [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Cambio de métricas a registros

Puede acceder a determinadas entradas de registro desde otras partes de la consola.

Para acceder desde widgets del panel a registros

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Dashboards (Paneles).
3. Elija un panel.
4. En el widget, elija el icono View logs (Ver registros) y, a continuación, elija View logs in this time range (Ver registros en este intervalo de tiempo). Si hay más de un filtro de métricas, seleccione uno de la lista. Si hay más filtros de métricas de los que podemos mostrar en la lista, elija More metric filters (Más filtros de métricas) y seleccione o busque un filtro de métricas.

Para acceder desde métricas hasta registros

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Metrics (Métricas).
3. En el campo de búsqueda en la pestaña All metrics (Todas las métricas), escriba el nombre de la métrica y pulse Intro.
4. Seleccione una o varias métricas de los resultados de la búsqueda.
5. Elija Actions (Acciones), View logs (Ver registros). Si hay más de un filtro de métricas, seleccione uno de la lista. Si hay más filtros de métricas de los que podemos mostrar en la lista, elija More metric filters (Más filtros de métricas) y seleccione o busque un filtro de métricas.

Resolución de problemas

La búsqueda tarda demasiado tiempo en completarse

Si tiene una gran cantidad de datos de registro, la búsqueda podría tardar mucho tiempo en completarse. Para acelerar la búsqueda, puede hacer lo siguiente:

- Si utiliza el AWS CLI, puede limitar la búsqueda solo a las secuencias de registro que le interesen. Por ejemplo, si su grupo de registros tiene 1000 flujos de registro, pero solo quiere ver tres flujos de registro que sabe que son relevantes, puede usar el AWS CLI para limitar la búsqueda solo a los tres flujos de registro del grupo de registros.
- Utilice un intervalo de tiempo más corto, más granular, lo que reduce la cantidad de datos que se van a buscar y acelera la consulta.

Cambie la retención de datos de registro en CloudWatch los registros

De forma predeterminada, los datos de registro se almacenan en CloudWatch los registros de forma indefinida. Sin embargo, puede configurar durante cuánto tiempo almacenar los datos de registro en un grupo de registros. Cualquier dato anterior a la configuración de retención actual se eliminará. Puede cambiar la retención de registro de cada grupo de registros cuando lo desee.

> **Note**
>
> CloudWatch Logs no elimina inmediatamente los eventos del registro cuando alcanzan su configuración de retención. Por lo general, pueden pasar hasta 72 horas antes de que se eliminen los eventos de registro, pero en raras ocasiones puede llevar más tiempo.
> Esto significa que si cambia un grupo de registro para que tenga una configuración de retención más larga cuando contenga eventos de registro que ya hayan expirado, pero que no se hayan eliminado realmente, esos eventos de registro tardarán hasta 72 horas en eliminarse después de que se alcance la fecha de retención nueva. Para asegurarse de que los datos de registro se eliminen de manera permanente, mantenga un grupo de registro en su configuración de retención más baja hasta que hayan transcurrido 72 horas desde el final del periodo de retención anterior o hasta que haya confirmado que se han eliminado los eventos de registro más antiguos.
> Cuando los eventos de registro alcanzan el límite en su configuración de retención, se marcan para su eliminación. Una vez marcados, ya no se contemplan dentro de los costos de almacenamiento de archivos, incluso si tarda un tiempo en eliminarlos. Estos eventos de registro marcados para su eliminación tampoco se incluyen cuando se utiliza una API para recuperar el valor `storedBytes` y ver cuántos bytes almacena un grupo de registro.

Para cambiar la configuración de retención de registros

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Registros, Grupos de registros.
3. Busque el grupo de registros que desea actualizar.
4. En la columna Retención de ese grupo de registros, elija la configuración de retención actual, como Nunca caduque.
5. En la configuración de retención, en Expirar eventos después, elige un valor de retención del registro y, a continuación, selecciona Guardar.

Etiquetar grupos de registros en Amazon CloudWatch Logs

Puede asignar sus propios metadatos a los grupos de registros que cree en Amazon CloudWatch Logs en forma de etiquetas. Una etiqueta es un par clave-valor definido por el usuario para un grupo de registros. El uso de etiquetas es una forma sencilla pero eficaz de gestionar AWS los recursos y organizar los datos, incluidos los datos de facturación.

> **Note**
>
> Puede usar etiquetas para controlar el acceso a los recursos de CloudWatch registros, incluidos los grupos de registros y los destinos. El acceso a los flujos de registro se controla a nivel de grupo de registro, debido a la relación jerárquica que existe entre los grupos de registro y los flujos de registro. A fin de obtener información sobre el uso de etiquetas para controlar el acceso, consulte Control del acceso a recursos de Amazon Web Services mediante etiquetas.

Contenido

Conceptos básicos de etiquetas

Utiliza AWS CloudFormation la API AWS CLI, o CloudWatch Logs, para completar las siguientes tareas:

- Agregar etiquetas a un grupo de registros al crearlo.
- Agregar etiquetas a un grupo de registros existente.
- Enumerar las etiquetas para un grupo de registros.
- Eliminar las etiquetas de un grupo de registros.

Puede utilizar las etiquetas para categorizar los grupos de registros. Por ejemplo, puede clasificarlas en categorías por objetivo, propietario o entorno. Dado que define la clave y el valor de cada etiqueta, puede crear un conjunto de categorías personalizadas para satisfacer sus necesidades específicas. Por ejemplo, podría definir un conjunto de etiquetas que lo ayude a realizar un seguimiento de los grupos de registros por propietario y aplicaciones asociadas. Estos son algunos ejemplos de etiquetas:

- Proyecto: nombre del proyecto
- Propietario: nombre
- Objetivo: pruebas de carga
- Aplicación: nombre de aplicación
- Entorno: producción

Seguimiento de costos mediante el etiquetado

Puedes usar etiquetas para categorizar y hacer un seguimiento de tus AWS costos. Al aplicar etiquetas a AWS los recursos, incluidos los grupos de registros, el informe de asignación de AWS costos incluye el uso y los costos agregados por etiquetas. Puede aplicar etiquetas que representen categorías de negocio (por ejemplo, centros de costos, nombres de aplicación o propietarios) para organizar los costos entre diferentes servicios. Para obtener más información, consulte Utilizar etiquetas de asignación de costos para informes de facturación personalizados en la Guía del usuario de AWS Billing .

Restricciones de las etiquetas

Se aplican las siguientes restricciones a las etiquetas.

Restricciones básicas

- El número máximo de etiquetas por grupo de registro es 50.
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- No se pueden cambiar o editar etiquetas para un grupo de registros eliminado.

Restricciones de clave de etiqueta

- Cada clave de etiqueta debe ser única. Si agrega una etiqueta con una clave que ya está en uso, la nueva etiqueta sobrescribe el par clave-valor existente.
- No puede iniciar una clave de etiqueta con él `aws:` porque este prefijo está reservado para su uso exclusivo. AWS AWS crea etiquetas que comienzan con este prefijo en tu nombre, pero no puedes editarlas ni eliminarlas.
- Las claves de etiqueta deben tener entre 1 y 128 caracteres Unicode de longitud.
- Las claves de etiquetas deben constar de los siguientes caracteres: letras Unicode, números, espacios en blanco y los siguientes caracteres especiales: `_ . / = + - @`.

Restricciones de valor de etiqueta

- Los valores de etiqueta deben tener entre 0 y 255 caracteres Unicode de longitud.
- Los valores de etiqueta pueden estar en blanco. De lo contrario, deben constar de los siguientes caracteres: letras Unicode, números, espacios en blanco y cualquiera de los siguientes caracteres especiales: `_ . / = + - @`.

Etiquetar grupos de registros mediante AWS CLI

Puede agregar, enumerar y eliminar etiquetas mediante la AWS CLI. Para ver ejemplos, consulte la documentación siguiente:

create-log-group

Crea un grupo de registros. Si lo desea, puede agregar etiquetas al crear el grupo de registros.

tag-resource

Asigna una o más etiquetas (pares clave-valor) al recurso de registros especificado. CloudWatch

list-tags-for-resource

Muestra las etiquetas que están asociadas a un CloudWatch recurso de Logs.

untag-resource

Elimina una o más etiquetas del recurso de CloudWatch registros especificado.

Etiquetado de grupos de registros mediante la API de CloudWatch registros

Puede añadir, enumerar y eliminar etiquetas mediante la API de CloudWatch Logs. Para ver ejemplos, consulte la documentación siguiente:

CreateLogGroup

Crea un grupo de registros. Si lo desea, puede agregar etiquetas al crear el grupo de registros.

TagResource

Asigna una o más etiquetas (pares clave-valor) al recurso Logs especificado CloudWatch .

ListTagsForResource

Muestra las etiquetas que están asociadas a un CloudWatch recurso de Logs.

UntagResource

Elimina una o más etiquetas del recurso de CloudWatch registros especificado.

Cifre los datos de registro en los CloudWatch registros mediante AWS Key Management Service

Los datos de los grupos de registros siempre se cifran en CloudWatch los registros. De forma predeterminada, CloudWatch Logs utiliza el cifrado del lado del servidor para los datos de registro en reposo. Como alternativa, puede utilizar AWS Key Management Service para este cifrado. Si lo hace, el cifrado se realiza mediante una AWS KMS clave. El uso del cifrado AWS KMS se habilita a nivel de grupo de registros, mediante la asociación de una clave de KMS a un grupo de registros, ya sea al crear el grupo de registros o después de su existencia.

> **Important**
>
> CloudWatch Los registros ahora admiten el contexto de cifrado, `kms:EncryptionContext:aws:logs:arn` ya que se utilizan como clave y el ARN del grupo de registros como valor de esa clave. Si tiene grupos de registro que ya ha cifrado con una clave de KMS y desea restringir la clave para que se utilice con una sola cuenta y grupo de registro, debe asignar una nueva clave de KMS que incluya una condición en la política de IAM. Para obtener más información, consulte AWS KMS claves y contexto de cifrado.

Después de asociar una clave de KMS con un grupo de registros, todos los datos ingeridos recientemente para el grupo de registros se cifran mediante la clave. Estos datos se almacenan en formato cifrado durante todo el período de retención. CloudWatch Logs descifra estos datos cada vez que se solicitan. CloudWatch Los registros deben tener permisos para la clave KMS siempre que se soliciten datos cifrados.

Si más adelante desasocias una clave KMS de un grupo de CloudWatch registros, Logs cifra los datos recién ingeridos mediante el método de cifrado predeterminado de CloudWatch Logs. Todos los datos ingeridos anteriormente que se cifraron con la clave KMS permanecen cifrados con la clave KMS. CloudWatch Los registros pueden seguir devolviendo esos datos una vez desasociada la clave de KMS, ya que CloudWatch los registros pueden seguir haciendo referencia a la clave. Sin embargo, si la clave se deshabilita posteriormente, CloudWatch Logs no podrá leer los registros que se cifraron con esa clave.

> **Important**
>
> CloudWatch Los registros solo admiten claves KMS simétricas. No utilice una clave asimétrica administrada por el cliente para cifrar los datos de los grupos de registros. Para obtener más información, consulte Utilización de claves simétricas y asimétricas.

Límites

- Para realizar los siguientes pasos, debe tener los siguientes permisos: `kms:CreateKey`, `kms:GetKeyPolicy` y `kms:PutKeyPolicy`.
- Después de asociar o desvincular una clave desde un grupo de registros, puede tardar hasta cinco minutos para que la operación surta efecto.

- Si revoca el acceso de CloudWatch los registros a una clave asociada o elimina una clave de KMS asociada, los datos cifrados de los CloudWatch registros ya no se podrán recuperar.
- No puede asociar una clave de KMS a un grupo de registros mediante la CloudWatch consola.

Paso 1: Crear una AWS KMS clave

Para crear una clave de KMS, utilice el siguiente comando create-key:

```
aws kms create-key
```

La salida contiene la ID de clave y el nombre de recurso de Amazon (ARN) de la clave. A continuación, se muestra un ejemplo de la salida:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Paso 2: establecer permisos en la clave de KMS

De forma predeterminada, todas AWS KMS las claves son privadas. Solo el propietario del recurso puede utilizarla para cifrar y descifrar datos. Sin embargo, el propietario del recurso puede conceder permisos para que otros usuarios y recursos accedan a la clave de KMS. Con este paso, concedes al servicio de CloudWatch registros el permiso principal para usar la clave. El principal de este servicio debe estar en la misma AWS región en la que se almacena la clave de KMS.

Como práctica recomendada, le recomendamos que restrinja el uso de la clave KMS únicamente a las AWS cuentas o grupos de registros que especifique.

En primer lugar, guarde la política predeterminada de su clave KMS `policy.json` mediante el siguiente get-key-policycomando:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./
policy.json
```

Abra el archivo `policy.json` en un editor de texto y agregue la sección en negrita desde una de las instrucciones siguientes. Separe la instrucción existente de la nueva instrucción con una coma. Estas instrucciones utilizan `Condition` secciones para mejorar la seguridad de la AWS KMS clave. Para obtener más información, consulte AWS KMS claves y contexto de cifrado.

La sección `Condition` de este ejemplo restringe la clave a un ARN único de grupo de registros.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "ArnEquals": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:log-group:log-group-name"
                }
            }
        }
    ]
}
```

La sección `Condition` de este ejemplo limita la utilización de la clave de AWS KMS a la cuenta especificada, pero se puede utilizar para cualquier grupo de registro.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:*"
```

```
                }
            }
        }
    ]
}
```

Por último, añada la política actualizada mediante el siguiente put-key-policycomando:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://
policy.json
```

Paso 3: asociar una clave de KMS a un grupo de registros

Puede asociar una clave de KMS a un grupo de registros al crearlo o posteriormente.

Para saber si un grupo de registros ya tiene asociada una clave KMS, utilice el siguiente describe-log-groupscomando:

```
aws logs describe-log-groups --log-group-name-prefix "log-group-name-prefix"
```

Si la salida incluye un campo `kmsKeyId`, el grupo de registro se asocia con la clave mostrada para el valor de ese campo.

Para asociar la clave de KMS a un grupo de registros al crearlo

Utilice el comando create-log-group como se indica a continuación:

```
aws logs create-log-group --log-group-name my-log-group --kms-key-id "key-arn"
```

Para asociar la clave de KMS a un grupo de registros existente

Utilice el comando associate-kms-key como se indica a continuación:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id "key-arn"
```

Paso 4: desasociar una clave de un grupo de registros

Para desasociar la clave de KMS asociada a un grupo de registros, utilice el siguiente disassociate-kms-keycomando:

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

AWS KMS claves y contexto de cifrado

Para mejorar la seguridad de AWS Key Management Service las claves y los grupos de CloudWatch registros cifrados, Logs ahora incluye los ARN de los grupos de registros como parte del contexto de cifrado que se utiliza para cifrar los datos de registro. El contexto de cifrado es un conjunto de pares clave-valor que se utilizan como datos autenticados adicionales. El contexto de cifrado le permite utilizar las condiciones de la política de IAM para limitar el acceso a su AWS KMS clave por AWS cuenta y grupo de registros. Para obtener más información, consulte Contexto de cifrado y Elementos de la política de JSON de IAM: condición.

Recomendamos que utilice diferentes claves de KMS para cada uno de los grupos de registro cifrados.

Si tiene un grupo de registros que cifró anteriormente y ahora desea cambiar el grupo de registros para utilizar una nueva clave de KMS que funcione solo para ese grupo de registros, siga estos pasos.

Para convertir un grupo de registros cifrado a fin de utilizar una clave de KMS con una política que la limite a ese grupo de registros

1. Ingrese el siguiente comando para encontrar el ARN de la clave actual del grupo de registros:

   ```
   aws logs describe-log-groups
   ```

   La salida incluye la siguiente línea. Tome nota del ARN. Tiene que utilizarlo en el paso 7.

   ```
   ...
   "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
   cdef-0123-456789abcdef"
   ...
   ```

2. Ingrese el siguiente comando para crear una nueva clave de KMS:

   ```
   aws kms create-key
   ```

3. Escriba el siguiente comando para guardar la política de la nueva clave en un archivo `policy.json`:

```
aws kms get-key-policy --key-id new-key-id --policy-name default --output text > ./
policy.json
```

4. Utilice un editor de texto para abrir `policy.json` y agregar una expresión `Condition` a la política:

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT-ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn":
 "arn:aws:logs:REGION:ACCOUNT-ID:log-
group:LOG-GROUP-NAME"
                }
            }
        }
    ]
}
```

5. Ingrese el siguiente comando para agregar la política actualizada a la nueva clave de KMS:

```
aws kms put-key-policy --key-id new-key-ARN --policy-name default --policy file://
policy.json
```

6. Ingrese el siguiente comando para asociar la política al grupo de registros:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id new-key-ARN
```

   CloudWatch Ahora, Logs cifra todos los datos nuevos con la nueva clave.

7. Luego, revoque todos los permisos excepto `Decrypt` en la antigua clave. En primer lugar, ingrese el siguiente comando para recuperar la política anterior:

```
aws kms get-key-policy --key-id old-key-ARN --policy-name default --output text
 > ./policy.json
```

8. Utilice un editor de texto para abrir `policy.json` y eliminar todos los valores de la lista `Action`, excepto `kms:Decrypt*`.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt*"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

9. Ingrese el siguiente comando para agregar la política actualizada a la antigua clave:

```
aws kms put-key-policy --key-id old-key-ARN --policy-name default --policy file://
policy.json
```

Ayude a proteger los datos de registro confidenciales con el enmascaramiento

Puedes ayudar a proteger los datos confidenciales que ingiere CloudWatch Logs mediante políticas de protección de datos de grupos de registros. Estas políticas le permiten auditar y enmascarar los datos confidenciales que aparecen en los eventos de registro incorporados por los grupos de registro en su cuenta.

Cuando creas una política de protección de datos, de forma predeterminada, los datos confidenciales que coincidan con los identificadores de datos que has seleccionado se ocultan en todos los puntos de salida, incluidos CloudWatch Logs Insights, los filtros de métricas y los filtros de suscripción. Solo los usuarios que tienen el permiso de IAM de `logs:Unmask` pueden ver los datos desenmascarados.

Puede crear una política de protección de datos para todos los grupos de registro de su cuenta y, también, puede crear políticas de protección de datos para grupos de registro individuales. Al crear una política para toda la cuenta, se aplica tanto a los grupos de registro existentes como a los que se creen en el futuro.

Si crea una política de protección de datos para toda su cuenta y también crea una política para un único grupo de registro, ambas políticas se aplican a ese grupo de registro. Todos los identificadores de datos administrados que se especifican en cualquiera de las políticas se auditan y enmascaran en ese grupo de registros.

> **Note**
>
> El enmascaramiento de datos confidenciales solo está permitido para los grupos de registros de la clase de registro estándar. Si crea una política de protección de datos para todos los grupos de registros de su cuenta, solo se aplicará a los grupos de registros de la clase de

registro estándar. Para obtener más información sobre las clases de registro, consulteClases de registro.

Cada grupo de registro solo puede tener una política de protección de datos a nivel de grupo de registro, pero esa política puede especificar varios identificadores de datos administrados para auditarlos y enmascararlos. El límite de una política de protección de datos es de 30 720 caracteres.

Important

Los datos confidenciales se detectan y enmascaran cuando se introducen en el grupo de registro. Al establecer una política de protección de datos, los eventos de registro que se introducen en el grupo de registro antes de esa hora no se enmascaran.

CloudWatch Los registros admiten muchos identificadores de datos administrados, que ofrecen tipos de datos preconfigurados que puede seleccionar para proteger los datos financieros, la información de salud personal (PHI) y la información de identificación personal (PII). CloudWatch La protección de los datos de los registros le permite aprovechar los modelos de coincidencia de patrones y de aprendizaje automático para detectar datos confidenciales. En el caso de algunos tipos de identificadores de datos gestionados, la detección depende también de encontrar determinadas palabras clave cerca de los datos confidenciales. También puedes usar identificadores de datos personalizados para crear identificadores de datos adaptados a tu caso de uso específico.

Se emite una métrica CloudWatch cuando se detectan datos confidenciales que coinciden con los identificadores de datos que ha seleccionado. Esta es la LogEventsWithFindingsmétrica y se emite en el espacio de nombres AWS/Logs. Puede usar esta métrica para crear CloudWatch alarmas y visualizarla en gráficos y paneles. Las métricas emitidas por la protección de datos son métricas proporcionadas y son gratuitas. Para obtener más información sobre las métricas a las que envía CloudWatch Logs CloudWatch, consulteMonitorización con CloudWatch métricas.

Cada identificador de datos gestionados está diseñado para detectar un tipo específico de datos confidenciales, como números de tarjetas de crédito, claves de acceso AWS secretas o números de pasaporte de un país o región determinados. Al crear una política de protección de datos, puede configurarla para que utilice estos identificadores con el fin de analizar los registros que se introducen en el grupo de registro y tomar medidas cuando se detecten.

CloudWatch La protección de datos de los registros puede detectar las siguientes categorías de datos confidenciales mediante identificadores de datos gestionados:

- Credenciales, como claves privadas o claves de acceso AWS secretas
- Información financiera, como números de tarjetas de crédito
- Información de identificación personal (PII), como licencias de conducir o números de la seguridad social
- Información médica protegida (PHI), como números de seguro médico o identificación médica
- Identificadores de dispositivos, como direcciones IP o direcciones MAC

Para obtener más información sobre los tipos de datos que puede proteger, consulte Tipos de datos que puede proteger.

Contenido

Descripción de las políticas de protección de datos

Temas

¿Qué son las políticas de protección de datos?

CloudWatch Logs utiliza políticas de protección de datos para seleccionar los datos confidenciales que desea escanear y las medidas que desea tomar para protegerlos. Para seleccionar los datos confidenciales de interés, utilice identificadores de datos. CloudWatch Registra la protección de datos y, a continuación, detecta los datos confidenciales mediante el aprendizaje automático y la coincidencia de patrones. En respuesta a los identificadores de datos encontrados, puede definir operaciones de auditoría y desidentificación. Estas operaciones le permiten registrar los datos confidenciales encontrados (o no encontrados) y enmascarar los datos confidenciales cuando se consultan los eventos de registro.

¿Cómo está estructurada la política de protección de datos?

Tal y como se muestra en la siguiente figura, un documento de la política de protección de datos incluye los siguientes elementos:

- Información opcional aplicable a toda la política en la parte superior del documento
- Una declaración que defina la auditoría y desidentifique acciones

Solo se puede definir una política de protección de datos por grupo de CloudWatch registros. La política de protección de datos puede incluir una o varias instrucciones de denegación o anonimización, pero solo una instrucción de auditoría.

Propiedades JSON para la política de protección de datos

Una política de protección de datos requiere la siguiente información básica para su identificación:

- Name: el nombre de la política.
- Description (opcional): la descripción de la política.
- Version: la versión del idioma de la política. La versión actual es 2021-06-01.
- Statement: una lista de instrucciones en la que se especifican las acciones de la política de protección de datos.

```
{
  "Name": "CloudWatchLogs-PersonalInformation-Protection",
  "Description": "Protect basic types of sensitive data",
  "Version": "2021-06-01",
```

```
  "Statement": [
        ...
  ]
}
```

Propiedades JSON de una instrucción de política

Una instrucción de política establece el contexto de detección de la operación de protección de datos.

- Sid (opcional): el identificador de la instrucción.
- DataIdentifier— Los datos confidenciales que CloudWatch Logs debe escanear. Por ejemplo, nombre, dirección o número de teléfono.
- Funcionamiento: las acciones de seguimiento, ya sea auditar o desidentificar. CloudWatch Logs realiza estas acciones cuando encuentra datos confidenciales.

```
{
  ...
  "Statement": [
    {
      "Sid": "audit-policy",
      "DataIdentifier": [
        "arn:aws:dataprotection::aws:data-identifier/Address"
      ],
      "Operation": {
        "Audit": {
          "FindingsDestination": {}
        }
      }
    },
```

Propiedades JSON de una operación de instrucción de política

Una instrucción de política establece una de las siguientes operaciones de protección de datos.

- Audit (Auditoría): emite informes de métricas y hallazgos sin interrumpir el registro. Las cadenas que coinciden incrementan la LogEventsWithFindingsmétrica que CloudWatch Logs publica en el espacio de nombres de AWS/Logs. CloudWatch Puede utilizar estas métricas para crear alarmas.

  Para ver un ejemplo de un informe de resultados, consulte Informes de resultados de auditoría.

Para obtener más información sobre las métricas a las que envía CloudWatch Logs, consulte. CloudWatch Monitorización con CloudWatch métricas

- De-identify (Desidentificar): enmascara los datos confidenciales sin interrumpir el registro.

Permisos de IAM requeridos para crear una política de protección de datos o trabajar con ella

Para poder trabajar con políticas de protección de datos para los grupos de registro, debe tener ciertos permisos, como se muestra en las tablas siguientes. Los permisos son diferentes para las políticas de protección de datos de toda la cuenta y para las políticas de protección de datos que se aplican a un único grupo de registro.

Permisos necesarios para las políticas de protección de datos de la cuenta

> **Note**
>
> Si realiza alguna de estas operaciones dentro de una función de Lambda, el rol de ejecución de Lambda y el límite de permisos también deben incluir los siguientes permisos.

Operación	Se necesita permiso de IAM	Recurso
Crear una política de protección de datos sin destinos de auditoría	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
Crea una política de protección de datos con CloudWatch Logs como destino de la auditoría	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*

Operación	Se necesita permiso de IAM	Recurso
	`logs:PutResourcePolicy`	*
	`logs:DescribeResourcePolicies`	*
	`logs:DescribeLogGroups`	*
Cree una política de protección de datos con Firehose como destino de auditoría	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`firehose:TagDeliveryStream`	`arn:aws:logs:::deliverystream/` *YOUR_DELIVERY_STREAM*
Crear una política de protección de datos con Amazon S3 como destino de auditoría	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`s3:GetBucketPolicy`	`arn:aws:s3:::` *YOUR_BUCKET*
	`s3:PutBucketPolicy`	`arn:aws:s3:::` *YOUR_BUCKET*

Operación	Se necesita permiso de IAM	Recurso
Desenmascarar eventos de registro en un grupo de registro especificado	`logs:Unmask`	`arn:aws:logs:::log-group:*`
Ver una política de protección de datos existente	`logs:GetDataProtectionPolicy`	*
Eliminar una política de protección de datos	`logs:DeleteAccountPolicy`	*
	`logs:DeleteDataProtectionPolicy`	*

Si ya se están enviando registros de auditoría de protección de datos a algún destino, las demás políticas que envíen registros al mismo destino solo necesitan los permisos `logs:PutDataProtectionPolicy` y `logs:CreateLogDelivery`.

Permisos necesarios para las políticas de protección de datos de un único grupo de registro

> **Note**
>
> Si realiza alguna de estas operaciones dentro de una función de Lambda, el rol de ejecución de Lambda y el límite de permisos también deben incluir los siguientes permisos.

Operación	Se necesita permiso de IAM	Recurso
Crear una política de protección de datos sin destinos de auditoría	`logs:PutDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
Cree una política de protección de datos con CloudWatc	`logs:PutDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Operación	Se necesita permiso de IAM	Recurso

h Logs como destino de auditoría	`logs:CreateLogDelivery`
	`logs:PutResourcePolicy`
	`logs:DescribeResourcePolicies`
	`logs:DescribeLogGroups`
		*
		*
		*
		*

Cree una política de protección de datos con Firehose como destino de auditoría	`logs:PutDataProtectionPolicy`
	`logs:CreateLogDelivery`
	`firehose:TagDeliveryStream`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		*
		`arn:aws:logs:::deliverystream/` *YOUR_DELIVERY_STREAM*

Create a data protection policy with Amazon S3 as an audit destination	`logs:PutDataProtectionPolicy`
	`logs:CreateLogDelivery`
	`s3:GetBucketPolicy`
	`s3:PutBucketPolicy`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		*
		`arn:aws:s3:::` *YOUR_BUCKET*
		`arn:aws:s3:::` *YOUR_BUCKET*

Unmask masked log events	`logs:Unmask`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Operación	Se necesita permiso de IAM	Recurso
View an existing data protection policy	`logs:GetDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
Eliminar una política de protección de datos	`logs:DeleteDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Si ya se están enviando registros de auditoría de protección de datos a algún destino, las demás políticas que envíen registros al mismo destino solo necesitan los permisos `logs:PutDataProtectionPolicy` y `logs:CreateLogDelivery`.

Política de protección de datos de ejemplo

La siguiente política de ejemplo permite al usuario crear, visualizar y eliminar las políticas de protección de datos que pueden enviar los resultados de las auditorías a los tres tipos de destinos de auditoría. No permite que el usuario vea los datos desenmascarados.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "YOUR_SID_1",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:DescribeLogGroups",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "YOUR_SID_2",
            "Effect": "Allow",
            "Action": [
                "logs:GetDataProtectionPolicy",
                "logs:DeleteDataProtectionPolicy",
```

```
                "logs:PutDataProtectionPolicy",
                "s3:PutBucketPolicy",
                "firehose:TagDeliveryStream",
                "s3:GetBucketPolicy"
            ],
            "Resource": [
                "arn:aws:firehose:::deliverystream/YOUR_DELIVERY_STREAM",
                "arn:aws:s3:::YOUR_BUCKET",
                "arn:aws:logs:::log-group:YOUR_LOG_GROUP:*"
            ]
        }
    ]
}
```

Creación de una política de protección de datos para toda la cuenta

Puedes usar la consola o los AWS CLI comandos de CloudWatch Logs para crear una política de protección de datos que oculte los datos confidenciales de todos los grupos de registros de tu cuenta. Esto afectará tanto a los grupos de registro actuales como a los que cree en el futuro.

> **Important**
>
> Los datos confidenciales se detectan y enmascaran cuando se introducen en el grupo de registro. Al establecer una política de protección de datos, los eventos de registro que se introducen en el grupo de registro antes de esa hora no se enmascaran.

Temas

- Consola
- AWS CLI

Consola

Para utilizar la consola con el fin de crear una política de protección de datos para toda la cuenta

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Configuración. Se encuentra cerca del final de la lista.
3. Elija la pestaña Logs (Registros).

4. Elija Configurar.
5. En el caso de los identificadores de datos gestionados, seleccione los tipos de datos que desee auditar y enmascarar para todos sus grupos de registros. Puede escribir en el cuadro de selección para buscar los identificadores que desee.

   Le recomendamos que seleccione solo los identificadores de datos que sean relevantes para sus datos de registro y para su empresa. Elegir muchos tipos de datos puede generar falsos positivos.

   Para obtener más información sobre qué tipos de datos puede proteger, consulte Tipos de datos que puede proteger.
6. (Opcional) Si desea auditar y enmascarar otros tipos de datos mediante identificadores de datos personalizados, elija Agregar identificador de datos personalizado. A continuación, introduzca un nombre para el tipo de datos y la expresión regular que desee utilizar para buscar ese tipo de datos en el registro de eventos. Para obtener más información, consulte Identificadores de datos personalizados.

   Una única política de protección de datos puede incluir hasta 10 identificadores de datos personalizados. Cada expresión regular que defina un identificador de datos personalizado debe tener 200 caracteres o menos.
7. (Opcional) Elija uno o más servicios a los que se deben enviar los resultados de la auditoría. Incluso si decide no enviar los resultados de la auditoría a ninguno de estos servicios, los tipos de datos confidenciales que seleccione seguirán ocultos.
8. Seleccione Activate data protection (Activar protección de datos).

AWS CLI

Para usar el AWS CLI para crear una política de protección de datos

1. Utilice un editor de texto para crear un archivo de política llamado `DataProtectionPolicy.json`. Para obtener información sobre la sintaxis de la política, consulte la siguiente sección.
2. Escriba el siguiente comando:

```
aws logs put-account-policy \
--policy-name TEST_POLICY --policy-type "DATA_PROTECTION_POLICY" \
--policy-document file://policy.json \
```

```
--scope "ALL" \
--region us-west-2
```

Sintaxis de la política de protección de datos para AWS CLI nuestras operaciones de API

Al crear una política de protección de datos de JSON para utilizarla en un AWS CLI comando o una operación de API, la política debe incluir dos bloques de JSON:

- El primer bloque debe incluir tanto una matriz `DataIdentifer` como una propiedad `Operation` con una acción `Audit`. La matriz `DataIdentifer` busca los tipos de datos confidenciales que desea enmascarar. Para obtener más información sobre las opciones disponibles, consulte Tipos de datos que puede proteger.

  La propiedad `Operation` con una acción `Audit` es necesaria para encontrar los términos de datos confidenciales. Esta acción `Audit` debe contener un objeto `FindingsDestination`. De forma opcional, puede utilizar ese objeto `FindingsDestination` para enumerar uno o más destinos a los que se deben enviar los informes de resultados de la auditoría. Si especifica destinos como grupos de registros, transmisiones de Amazon Data Firehose y buckets S3, es necesario que ya existan. Para ver un ejemplo de un informe de resultados de auditoría, consulte Informes de resultados de auditoría.
- El segundo bloque debe incluir tanto una matriz `DataIdentifer` como una propiedad `Operation` con una acción `Deidentify`. La matriz `DataIdentifer` debe coincidir exactamente con la matriz `DataIdentifer` del primer bloque de la política.

  La propiedad `Operation` con la acción `Deidentify` es lo que realmente enmascara los datos y debe contener el objeto `"MaskConfig": {}`. El objeto `"MaskConfig": {}` debe estar vacío.

El siguiente es un ejemplo de una política de protección de datos que utiliza únicamente identificadores de datos gestionados. Esta política oculta las direcciones de correo electrónico y las licencias de conducir de los Estados Unidos.

Para obtener información sobre las políticas que especifican identificadores de datos personalizados, consulteUso de identificadores de datos personalizados en la política de protección de datos.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
```

```
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Creación de una política de protección de datos para un único grupo de registro

Puedes usar la consola o los AWS CLI comandos de CloudWatch Logs para crear una política de protección de datos que oculte los datos confidenciales.

Puede asignar una política de protección de datos a cada grupo de registro. Cada política de protección de datos puede auditar varios tipos de información. Cada política de protección de datos puede incluir una declaración de auditoría.

Temas

- Consola
- AWS CLI

Consola

Para utilizar la consola con el fin de crear una política de protección de datos

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Registros, Grupos de registros.
3. Elija el nombre del grupo de registros.
4. Elija Actions (Acciones), Create data protection policy (Crear política de protección de datos).
5. En el caso de los identificadores de datos gestionados, seleccione los tipos de datos que desee auditar y ocultar en este grupo de registros. Puede escribir en el cuadro de selección para buscar los identificadores que desee.

   Le recomendamos que seleccione solo los identificadores de datos que sean relevantes para sus datos de registro y para su empresa. Elegir muchos tipos de datos puede generar falsos positivos.

   Para obtener más información sobre los tipos de datos que puede proteger mediante identificadores de datos gestionados, consulte. Tipos de datos que puede proteger
6. (Opcional) Si desea auditar y enmascarar otros tipos de datos mediante identificadores de datos personalizados, elija Agregar identificador de datos personalizado. A continuación, introduzca un nombre para el tipo de datos y la expresión regular que desee utilizar para buscar ese tipo de datos en el registro de eventos. Para obtener más información, consulte Identificadores de datos personalizados.

   Una única política de protección de datos puede incluir hasta 10 identificadores de datos personalizados. Cada expresión regular que defina un identificador de datos personalizado debe tener 200 caracteres o menos.

7. (Opcional) Elija uno o más servicios a los que se deben enviar los resultados de la auditoría. Incluso si decide no enviar los resultados de la auditoría a ninguno de estos servicios, los tipos de datos confidenciales que seleccione seguirán ocultos.
8. Seleccione Activate data protection (Activar protección de datos).

AWS CLI

Para usar el AWS CLI para crear una política de protección de datos

1. Utilice un editor de texto para crear un archivo de política llamado `DataProtectionPolicy.json`. Para obtener información sobre la sintaxis de la política, consulte la siguiente sección.
2. Escriba el siguiente comando:

```
aws logs put-data-protection-policy --log-group-identifier "my-log-group" --policy-
document file:///Path/DataProtectionPolicy.json --region us-west-2
```

Sintaxis de la política de protección de datos para AWS CLI nuestras operaciones de API

Al crear una política de protección de datos de JSON para utilizarla en un AWS CLI comando o una operación de API, la política debe incluir dos bloques de JSON:

- El primer bloque debe incluir tanto una matriz `DataIdentifer` como una propiedad `Operation` con una acción `Audit`. La matriz `DataIdentifer` busca los tipos de datos confidenciales que desea enmascarar. Para obtener más información sobre las opciones disponibles, consulte Tipos de datos que puede proteger.

  La propiedad `Operation` con una acción `Audit` es necesaria para encontrar los términos de datos confidenciales. Esta acción `Audit` debe contener un objeto `FindingsDestination`. De forma opcional, puede utilizar ese objeto `FindingsDestination` para enumerar uno o más destinos a los que se deben enviar los informes de resultados de la auditoría. Si especifica destinos como grupos de registros, transmisiones de Amazon Data Firehose y buckets S3, es necesario que ya existan. Para ver un ejemplo de un informe de resultados de auditoría, consulte Informes de resultados de auditoría.
- El segundo bloque debe incluir tanto una matriz `DataIdentifer` como una propiedad `Operation` con una acción `Deidentify`. La matriz `DataIdentifer` debe coincidir exactamente con la matriz `DataIdentifer` del primer bloque de la política.

La propiedad `Operation` con la acción `Deidentify` es lo que realmente enmascara los datos y debe contener el objeto `"MaskConfig": {}`. El objeto `"MaskConfig": {}` debe estar vacío.

El siguiente es un ejemplo de una política de protección de datos que enmascara las direcciones de correo electrónico y las licencias de conducir de los Estados Unidos.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
```

```
            }
        }
    ]
}
```

Visualización de datos desenmascarados

Para ver los datos desenmascarados, el usuario debe tener el permiso `logs:Unmask`. Los usuarios con este permiso pueden consultar los datos desenmascarados de las siguientes maneras:

- Al ver los eventos de un flujo de registro, elija Display (Mostrar), Unmask (Desenmascarar).
- Utilice una consulta de CloudWatch Logs Insights que incluya el comando unmask (@message). La siguiente consulta de ejemplo muestra los 20 eventos de registro más recientes del flujo, sin enmascarar:

  ```
  fields @timestamp, @message, unmask(@message)
  | sort @timestamp desc
  | limit 20
  ```

  Para obtener más información sobre CloudWatch los comandos de Logs Insights, consulteCloudWatch Sintaxis de consultas de Logs Insights.
- Utilice una FilterLogEventsoperación GetLogEventso con el `unmask` parámetro.

La CloudWatchLogsFullAccesspolítica incluye el `logs:Unmask` permiso. Para `logs:Unmask` concedérselo a un usuario que no lo tiene CloudWatchLogsFullAccess, puedes adjuntar una política de IAM personalizada a ese usuario. Para obtener más información, consulte Adición de permisos a un usuario (consola).

Informes de resultados de auditoría

Si configuras las políticas de auditoría de protección de datos de CloudWatch Logs para escribir informes de auditoría en CloudWatch Logs, Amazon S3 o Firehose, estos informes de resultados son similares a los del siguiente ejemplo. CloudWatch Logs escribe un informe de resultados para cada evento de registro que contiene datos confidenciales.

```
{
    "auditTimestamp": "2023-01-23T21:11:20Z",
    "resourceArn": "arn:aws:logs:us-west-2:111122223333:log-group:/aws/lambda/
MyLogGroup:*",
```

```
    "dataIdentifiers": [
        {
            "name": "EmailAddress",
            "count": 2,
            "detections": [
                {
                    "start": 13,
                    "end": 26
                },
{
                    "start": 30,
                    "end": 43
                }
            ]
        }
    ]
}
```

Los campos del informe son los siguientes:

- El campo `resourceArn` muestra el grupo de registro en el que se encontraron los datos confidenciales.
- El objeto `dataIdentifiers` muestra información sobre los resultados de un tipo de datos confidenciales que está auditando.
- El campo `name` identifica el tipo de datos confidenciales sobre los que se informa en esta sección.
- El campo `count` muestra el número de veces que este tipo de datos confidenciales aparece en el evento de registro.
- Los campos `start` y `end` muestran en qué parte del evento de registro, por recuento de caracteres, aparece cada resultado de datos confidenciales.

El ejemplo anterior muestra un informe sobre la búsqueda de dos direcciones de correo electrónico en un evento de registro. La primera dirección de correo electrónico comienza en el carácter 13 del evento de registro y termina en el carácter 26. La segunda dirección de correo electrónico va del carácter 30 al 43. Aunque este evento de registro tiene dos direcciones de correo electrónico, el valor de la métrica `LogEventsWithFindings` solo se incrementa en uno, ya que esa métrica cuenta la cantidad de eventos de registro que contienen datos confidenciales, no la cantidad de resultados de datos confidenciales.

Política clave necesaria para enviar los resultados de la auditoría a un depósito protegido por AWS KMS

Para proteger los datos de un bucket de Amazon S3, habilite el cifrado del lado del servidor con las claves administradas de Amazon S3 (SSE-S3) o con el cifrado del lado del servidor con claves de KMS (SSE-KMS). Para obtener más información, consulte Protección de datos mediante cifrado del lado del servidor en la Guía del usuario de Amazon S3.

Si envía los resultados de la auditoría a un bucket que está protegido con SSE-S3, no se requiere ninguna configuración adicional. Amazon S3 se encarga de la clave de cifrado.

Si envía los resultados de la auditoría a un bucket que está protegido con SSE-KMS, debe actualizar la política de claves para la clave de KMS, de manera que la cuenta de entrega de registros pueda escribir en el bucket de S3. Para obtener más información sobre la política de claves necesaria para su uso con SSE-KMS, consulte Amazon S3 la Guía del usuario de Amazon CloudWatch Logs.

Tipos de datos que puede proteger

Esta sección contiene información sobre los tipos de datos que puede proteger en una política de protección de datos de CloudWatch Logs. CloudWatch Los identificadores de datos gestionados por Logs ofrecen tipos de datos preconfigurados para proteger los datos financieros, la información de salud personal (PHI) y la información de identificación personal (PII). También puede usar identificadores de datos personalizados para crear identificadores de datos adaptados a su caso de uso específico.

Contenido

CloudWatch Registra los identificadores de datos gestionados para tipos de datos confidenciales

Esta sección contiene información sobre los tipos de datos que puede proteger mediante identificadores de datos gestionados y qué países y regiones son relevantes para cada uno de esos tipos de datos.

En el caso de algunos tipos de datos confidenciales, CloudWatch Logs Data Protection busca palabras clave próximas a los datos y solo encuentra una coincidencia si encuentra esa palabra clave. Si una palabra clave debe estar cerca de un tipo de datos en particular, normalmente debe estar dentro de los 30 caracteres (ambos incluidos) de los datos.

Si una palabra clave contiene un espacio, la protección de datos de CloudWatch Logs busca automáticamente las variantes de palabras clave a las que no haya espacio o que contengan un guión bajo (_) o un guión (-) en lugar del espacio. En algunos casos, CloudWatch Logs también expande o abrevia una palabra clave para abordar las variaciones comunes de la misma.

En las siguientes tablas, se enumeran los tipos de información sobre credenciales, dispositivos, información financiera, médica y de salud protegida (PHI) que CloudWatch los registros pueden detectar mediante identificadores de datos gestionados. Estos datos se suman a ciertos tipos de datos que también podrían considerarse información de identificación personal (PII).

Identificadores compatibles que son independientes del idioma y la región

Identificador	Categoría
`Address`	Personal
`AwsSecretKey`	Credenciales
`CreditCardExpiration`	Datos financieros
`CreditCardNumber`	Datos financieros
`CreditCardSecurityCode`	Datos financieros
`EmailAddress`	Personal
`IpAddress`	Personal
`LatLong`	Personal
`Name`	Personal
`OpenSshPrivateKey`	Credenciales
`PgpPrivateKey`	Credenciales
`PkcsPrivateKey`	Credenciales
`PuttyPrivateKey`	Credenciales
`VehicleIdentificationNumber`	Personal

Los identificadores de datos dependientes de la región deben incluir el nombre del identificador y, a continuación, un guion y los códigos de dos letras (ISO 3166-1 alpha-2). Por ejemplo, `DriversLicense-US`.

Identificadores compatibles que deben incluir un código de país o región de dos letras

Identificador	Categoría	Países e idiomas
BankAccountNumber	Datos financieros	DE, ES, FR, GB, IT

Identificador	Categoría	Países e idiomas
CepCode	Personal	BR
Cnpj	Personal	BR
CpfCode	Personal	BR
DriversLicense	Personal	AT, AU, BE, BG, CA, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IT, LT, LU, LV, MT, NL, PL, PT, RO, SE, SI, SK, US
DrugEnforcementAgencyNumber	Estado	EE. UU.
ElectoralRollNumber	Personal	GB
HealthInsuranceCardNumber	Estado	UE
HealthInsuranceClaimNumber	Estado	EE. UU.
HealthInsuranceNumber	Estado	FR
HealthcareProcedureCode	Estado	EE. UU.
IndividualTaxIdentificationNumber	Personal	EE. UU.
InseeCode	Personal	FR
MedicareBeneficiaryNumber	Estado	EE. UU.
NationalDrugCode	Estado	EE. UU.
NationalIdentificationNumber	Personal	DE, ES, IT
NationalInsuranceNumber	Personal	GB
NationalProviderId	Estado	EE. UU.

Identificador	Categoría	Países e idiomas
NhsNumber	Estado	GB
NieNumber	Personal	ES
NifNumber	Personal	ES
PassportNumber	Personal	CA, DE, ES, FR, GB, IT, US
PermanentResidenceNumber	Personal	CA
PersonalHealthNumber	Estado	CA
PhoneNumber	Personal	BR, DE, ES, FR, GB, IT, US
PostalCode	Personal	CA
RgNumber	Personal	BR
SocialInsuranceNumber	Personal	CA
Ssn	Personal	ES, US
TaxId	Personal	DE, ES, FR, GB
ZipCode	Personal	EE. UU.

Credenciales

CloudWatch La protección de datos de Logs puede encontrar los siguientes tipos de credenciales.

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones
AWS clave de acceso secreta	`AwsSecretKey`	`aws_secret_access_key` , `credentials` , `secret access key`,	Todos

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones
		`secret key`, `set-awscredential`	
Clave privada de OpenSSH	`OpenSSHPrivateKey`	Ninguna	Todos
Clave privada de PGP	`PgpPrivateKey`	Ninguna	Todos
Clave privada de Pkcs	`PkcsPrivateKey`	Ninguna	Todos
Clave privada PuTTY	`PuttyPrivateKey`	Ninguna	Todos

ARN de identificador de datos para tipos de datos de credenciales

A continuación se enumeran los nombres de recursos de Amazon (ARN) para los identificadores de datos que puede añadir a sus políticas de protección de datos.

ARN de identificadores de datos de credenciales
`arn:aws:dataprotection::aws:data-identifier/AwsSecretKey`
`arn:aws:dataprotection::aws:data-identifier/OpenSshPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PgpPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PkcsPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PuttyPrivateKey`

Identificadores de dispositivos

CloudWatch La protección de datos de Logs puede encontrar los siguientes tipos de identificadores de dispositivos.

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones
Dirección IP	`IpAddress`	Ninguna	Todos

ARN de identificador de datos para tipos de datos de dispositivos

A continuación se enumeran los nombres de recursos de Amazon (ARN) para los identificadores de datos que puede añadir a sus políticas de protección de datos.

ARN de identificador de datos de dispositivos
`arn:aws:dataprotection::aws:data-identifier/IpAddress`

Información financiera

CloudWatch La protección de datos de Logs puede encontrar los siguientes tipos de información financiera.

Si establece una política de protección de datos, CloudWatch Logs busca los identificadores de datos que especifique, independientemente de la geolocalización en la que se encuentre el grupo de registros. La información de la columna Países y regiones de esta tabla indica si se deben agregar códigos de país de dos letras al identificador de datos para detectar las palabras clave adecuadas para esos países y regiones.

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de cuenta bancaria	`BankAccountNumber`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Palabras clave para números de cuentas	Francia, Alemania Italia, España, Reino Unido	Incluye números de cuentas bancarias internaci

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
		bancarias que aparece más adelante en esta sección.		onales (IBAN) que constan de hasta 34 caracteres alfanuméricos, incluidos elementos como el código de país.
Fecha de caducidad de la tarjeta	`CreditCardExpiration`	`exp d`, `exp m`, `exp y`, `expiration` , `expiry`	Todos	

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de tarjetas de crédito	`CreditCardNumber`	`account number`, `american express`, `amex`, `bank card`, `card`, `card number`, `card num`, `cc #`, `ccn`, `check card`, `credit`, `credit card#`, `dankort`, `debit`, `debit card`, `diners club`, `discover`, `electron`, `japanese card bureau`, `jcb`, `mastercard `, `mc`, `pan`, `payment account number`, `payment card number`, `pcn`, `union pay`, `visa`	Todos	La detección requiere que los datos sean una secuencia de 13 a 19 dígitos que cumpla con la fórmula de cheques de Luhn y utilice un prefijo de número de tarjeta estándar para cualquiera de los

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
				siguientes tipos de tarjetas de crédito: American Express, Dankort, Diner's Club, Discover, Electron, Japanese Card Bureau (JCB) UnionPay, Mastercard y Visa.

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Código de verificación de tarjeta de crédito	`CreditCardSecurityCode`	`card id`, `card identification code`, `card identification number` , `card security code`, `card validation code` , `card validation number` , `card verification data` , `card verification value`, `cvc`, `cvc2`, `cvv`, `cvv2`, `elo verification code`	Todos	

Palabras clave para números de cuentas bancarias

Utilice las siguientes palabras clave para detectar números de cuentas bancarias internacionales (IBAN) que constan de hasta 34 caracteres alfanuméricos, incluidos elementos como códigos de país.

País	Palabras clave
Francia	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `code bancaire`, `compte bancaire`, `customer account id`, `customer account number`, `customer bank account id`, `iban`, `numéro de compte`
Alemania	`account code`, `account number`, `accountno#` , `accountnumber#` , `bankleitzahl` , `bban`, `customer account id`, `customer account number`, `customer bank account id`, `geheimzahl` , `iban`, `kartennummer` , `kontonummer` , `kreditkartennummer` , `sepa`

País	Palabras clave
Italia	account code, account number, accountno# , accountnumber# , bban, codice bancario, conto bancario, customer account id, customer account number, customer bank account id, iban, numero di conto
España	account code, account number, accountno# , accountnumber# , bban, código cuenta, código cuenta bancaria, cuenta cliente id, customer account ID, customer account number, customer bank account id, iban, número cuenta bancaria cliente, número cuenta cliente
Reino Unido	account code, account number, accountno# , accountnumber# , bban, customer account ID, customer account number, customer bank account id, iban, sepa
Estados Unidos	bank account, bank acct, checking account, checking acct, deposit account, deposit acct, savings account, savings acct, chequing account, chequing acct

CloudWatch Los registros no indican la aparición de las siguientes secuencias, que los emisores de tarjetas de crédito han reservado para su comprobación pública.

```
122000000000003, 2222405343248877, 2222990905257051, 2223007648726984,
 2223577120017656,
30569309025904, 34343434343434, 3528000700000000, 3530111333300000, 3566002020360505,
 36148900647913,
36700102000000, 371449635398431, 378282246310005, 378734493671000, 38520000023237,
 4012888888881881,
4111111111111111, 4222222222222, 4444333322221111, 4462030000000000, 4484070000000000,
 4911830000000,
4917300800000000, 4917610000000000, 4917610000000000003, 5019717010103742,
 5105105105105100,
5111010030175156, 5185540810000019, 5200828282828210, 5204230080000017,
 5204740009900014, 5420923878724339,
5454545454545454, 5455330760000018, 5506900490000436, 5506900490000444,
 5506900510000234, 5506920809243667,
```

```
5506922400634930, 5506927427317625, 5553042241984105, 5555553753048194,
 5555555555554444, 5610591081018250,
6011000990139424, 6011000400000000, 6011111111111117, 630490017740292441,
 630495060000000000,
6331101999990016, 6759649826438453, 6799990100000000019, and 76009244561.
```

ARN de identificador de datos para tipos de datos financieros

A continuación se enumeran los nombres de recursos de Amazon (ARN) para los identificadores de datos que puede añadir a sus políticas de protección de datos.

ARN de identificadores de datos financieros
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-DE`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-ES`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-FR`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-GB`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-IT`
`arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-US`
`arn:aws:dataprotection::aws:data-identifier/CreditCardExpiration`
`arn:aws:dataprotection::aws:data-identifier/CreditCardNumber`
`arn:aws:dataprotection::aws:data-identifier/CreditCardSecurityCode`

Información médica protegida (PHI)

CloudWatch La protección de datos de los registros puede encontrar los siguientes tipos de información de salud protegida (PHI).

Si establece una política de protección de datos, CloudWatch Logs busca los identificadores de datos que especifique, independientemente de la geolocalización en la que se encuentre el grupo de registros. La información de la columna Países y regiones de esta tabla indica si se deben agregar

códigos de país de dos letras al identificador de datos para detectar las palabras clave adecuadas para esos países y regiones.

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones
Número de registro de la Administración para el Control de Drogas (DEA)	`DrugEnforcementAgencyNumber`	`dea number`, `dea registration`	Estados Unidos
Número de tarjeta de seguro médico (EHIC)	`HealthInsuranceCardNumber`	`assicurazione sanitaria numero`, `carta assicurazione numero`, `carte d’assurance maladie `, `carte européenne d'assurance maladie `, `ceam`, `ehic`, `ehic#`, `finlandehicnumber#  `, `gesundheitskarte `, `hälsokort `, `health card`, `health card number`, `health insurance card`, `health insurance number`, `insurance card number`, `krankenversicherungskarte `, `krankenversicherungsnummer  `, `medical account number`, `numero conto medico`, `numéro d’assurance maladie `, `numéro de carte d’assurance `, `numéro de compte medical`, `número de cuenta médica`, `número`	Unión Europea

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones
		de seguro de salud, número de tarjeta de seguro, sairaanhoitokortin , sairausvakuutuskortti , sairausvakuutusnumero , sjukförsäkringnummer, sjukförsäkringskort , suomiehic-numero , tarjeta de salud, terveyskortti , tessera sanitaria assicurazione numero , versicherungsnummer	
Número de reclamación del seguro médico (HICN)	HealthInsuranceClaimNumber	health insurance claim number, hic no, hic no., hic number, hic#, hicn, hicn#, hicno#	Estados Unidos
Número de seguro médico o identificación médica	HealthInsuranceNumber	carte d'assuré social, carte vitale, insurance card	Francia
Código del sistema de codificación de procedimientos comunes de atención médica (HCPCS)	HealthcareProcedureCode	current procedural terminology , hcpcs, healthcare common procedure coding system	Estados Unidos

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones
Número de beneficiario de Medicare (MBN)	`MedicareBeneficiaryNumber`	`mbi`, `medicare beneficiary`	Estados Unidos
Código nacional de medicamento (NDC)	`NationalDrugCode`	`national drug code`, `ndc`	Estados Unidos
Identificador nacional de proveedores (NPI)	`NationalProviderId`	`hipaa`, `n.p.i.`, `national provider`, `npi`	Estados Unidos
Número del Servicio Nacional de Salud (NHS)	`NhsNumber`	`national health service`, `NHS`	Gran Bretaña
Número médico personal	`PersonalHealthNumber`	`canada healthcare number`, `msp number`, `care number`, `phn`, `soins de santé`	Canadá

ARN de identificador de datos para tipos de datos de información médica protegida (PHI)

A continuación, se enumeran los nombres de recursos de Amazon (ARN) de identificadores de datos que se pueden utilizar en las políticas de protección de datos de información médica protegida (PHI).

ARN de identificadores de datos de PHI
`arn:aws:dataprotection::aws:data-identifier/DrugEnforcementAgencyNumber-US`
`arn:aws:dataprotection::aws:data-identifier/HealthcareProcedureCode-US`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceCardNumber-EU`

ARN de identificadores de datos de PHI
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceClaimNumber-US
arn:aws:dataprotection::aws:data-identifier/HealthInsuranceNumber-FR
arn:aws:dataprotection::aws:data-identifier/MedicareBeneficiaryNumber-US
arn:aws:dataprotection::aws:data-identifier/NationalDrugCode-US
arn:aws:dataprotection::aws:data-identifier/NationalInsuranceNumber-GB
arn:aws:dataprotection::aws:data-identifier/NationalProviderId-US
arn:aws:dataprotection::aws:data-identifier/NhsNumber-GB
arn:aws:dataprotection::aws:data-identifier/PersonalHealthNumber-CA

Información de identificación personal (PII)

CloudWatch La protección de datos de Logs puede encontrar los siguientes tipos de información de identificación personal (PII).

Si estableces una política de protección de datos, CloudWatch Logs busca los identificadores de datos que especifiques, independientemente de la geolocalización en la que se encuentre el grupo de registros. La información de la columna Países y regiones de esta tabla indica si se deben agregar códigos de país de dos letras al identificador de datos para detectar las palabras clave adecuadas para esos países y regiones.

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Fecha de nacimiento	`DateOfBirth`	`dob`, `date of birth`, `birthdate` , `birth date`, `birthday`, `b-day`, `bday`	Cualquiera	La mayoría de los formatos de fecha están admitidos, como todos los dígitos y combinaciones de dígitos y nombres de meses. Los componentes de fecha se pueden separar mediante espacios, barras (/) o

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
				guiones (-).
Código de Endereçamento Postal (CEP)	`CepCode`	`cep`, `código de endereçamento postal`, `codigo de endereçamento postal`	Brasil	
Cadastro Nacional da Pessoa Jurídica (CNPJ)	`Cnpj`	`cadastro nacional da pessoa jurídica`, `cadastro nacional da pessoa juridica`, `cnpj`	Brasil	
Cadastro de Pessoas Físicas (CPF)	`CpfCode`	`Cadastro de pessoas fisicas`, `cadastro de pessoas físicas`, `cadastro de pessoa física`, `cadastro de pessoa fisica`, `cpf`	Brasil	

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de identificación del permiso de conducir	`DriversLicense`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Números de identificación de licencias de conducir que se encuentra más adelante en esta sección.	Muchos países. Para obtener más información, consulte la tabla de Números de identificación de licencias de conducir.	
Número de registro electoral	`ElectoralRollNumber`	`electoral #`, `electoral number`, `electoral roll #`, `electoral roll no.`, `electoral roll number`, `electoral rollno`	Reino Unido	

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Identificación individual del contribuyente	`IndividualTaxIdentificationNumber`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Números de identificación tributaria que se encuentra más adelante en esta sección.	Brasil, Francia, Alemania España, Reino Unido	
Instituto Nacional de Estadística y Estudios Económicos (INSEE)	`InseeCode`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Palabras claves para los números de identificación nacionales que se encuentra más adelante en esta sección.	Francia	

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de identificación nacional	`NationalIdentificationNumber`	Sí. Para obtener más información, consulte la tabla de Palabras claves para los números de identificación nacionales que se encuentra más adelante en esta sección.	Alemania, España, Italia	Esto incluye los identificadores del documento nacional de identidad (DNI) (España), los códigos del Codice Fiscale (Italia) y los números del documento nacional de identidad (Alemania).

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de seguro nacional (NINO)	`NationalInsuranceNumber`	`insurance no.`, `insurance number`, `insurance#` , `national insurance number`, `nationalinsurance#` , `nationalinsurancenumber` , `nin`, `nino`	Reino Unido	–
Número de identidad de extranjero (NIE)	`NieNumber`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Números de identificación tributaria que se encuentra más adelante en esta sección.	España	
Número de identificación fiscal (NIF)	`NifNumber`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Números de identificación tributaria que se encuentra más adelante en esta sección.	España	

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de pasaporte	`PassportNumber`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Palabras clave para números de pasaporte que aparece más adelante en esta sección.	Canadá, Francia, Alemania, Italia, España, Reino Unido, Estados Unidos	
Número de residencia permanente	`PermanentResidenceNumber`	`carte résident permanent` , `numéro carte résident permanent` , `numéro résident permanent` , `permanent resident card`, `permanent resident card number`, `permanent resident no`, `permanent resident no.`, `permanent resident number`, `pr no`, `pr no.`, `pr non`, `pr number`, `résident permanent no.`, `résident permanent non`	Canadá	

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de teléfono	PhoneNumber	Brasil: las palabras clave también incluyen: `cel`, `celular`, `fone`, `móvel`, `número residencial`, `numero residencial`, `telefone` Otras: `cell`, `contact`, `fax`, `fax number`, `mobile`, phone, `phone number`, `tel`, `telephone`, `telephone number`	Brasil, Canadá, Francia, Alemania, Italia, España, Reino Unido, Estados Unidos	Esto incluye los números gratuitos de Estados Unidos y números de fax. Si una palabra clave está cerca de los datos, no es necesario que el número incluya un código de país. Si una palabra clave no está

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
				cerca de los datos, el número debe incluir un código de país.
Postal Code (Código postal)	`PostalCode`	Ninguna	Canadá	
Registro Geral (RG)	`RgNumber`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Números de identificación tributaria que se encuentra más adelante en esta sección.	Brasil	
Número de Seguro Social (SIN)	`SocialInsuranceNumber`	`canadian id`, `numéro d'assurance sociale`, `social insurance number`, `sin`	Canadá	

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de la Seguridad Social (SSN)	`Ssn`	España: `número de la seguridad social`, `social security no.`, `social security no`, `número de la seguridad social`, `social security number`, `socialsecurityno#` , `ssn`, `ssn#` Estados Unidos: `social security`, `ss#`, `ssn`	España, Estados Unidos	
Número de identificación o referencia del contribuyente	`TaxId`	Sí. Se aplican diferentes palabras clave a diferentes países. Para obtener más información, consulte la tabla de Números de identificación tributaria que se encuentra más adelante en esta sección. .	Francia, Alemania España, Reino Unido	Esto incluye TIN (Francia), Steueridentifikationsnummer (Alemania), CIF (España) y TRN, UTR (Reino Unido).

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Código postal	`ZipCode`	`zip code`, `zip+4`	Estados Unidos	Código postal de los Estados Unidos.
Dirección postal	`Address`	Ninguna	Australia, Canadá, Francia, Alemania Italia, España, Reino Unido, Estados Unidos	Aunque no se requiere una palabra clave, para la detección es necesario que la dirección incluya el nombre de una ciudad o lugar y un código postal.
Dirección de correo electrónico	`EmailAddress`	Ninguna	Cualquiera	

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Coordenadas del sistema de posicionamiento global (GPS)	`LatLong`	`coordinate` , `coordinates` , `latlong`, `latitude longitude` , `location`, `position`	Cualquiera	CloudWatch Los registros pueden detectar las coordenadas GPS si las coordenadas de latitud y longitud se almacenan en pares y están en formato de grados decimales (DD), por ejemplo, 41.948614, -87.65531

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
				1. No es compatible con coordenadas en formato de grados y minutos decimales (DDM), por ejemplo 41°56.9168'N 87°39.3187'O, o en formato de grados, minutos y segundos (DMS), por ejemplo 41°56'55.0104"N

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
				87°39'19.1196"O.
Nombre completo	`Name`	Ninguna	Cualquiera	CloudWatch Los registros solo pueden detectar nombres completos. La compatibilidad se limita a los conjuntos de caracteres latinos.

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
Número de identificación de vehículo (VIN)	`VehicleIdentificationNumber`	`Fahrgestellnummer` , `niv`, `numarul de identificare` , `numarul seriei de sasiu`, `serie sasiu`, `numer VIN`, `Número de Identificação do Veículo`, `Número de Identificación de Automóviles` , `numéro d'identification du véhicule`, `vehicle identification number`, `vin`, `VIN numeris`	Cualquiera	CloudWatch Los registros pueden detectar los VIN que constan de una secuencia de 17 caracteres y cumplen con las normas ISO 3779 y 3780. Estos estándares fueron diseñados para su uso en todo el mundo.

Palabras clave de números de identificación del permiso de conducir

Para detectar varios tipos de números de identificación del carné de conducir, CloudWatch Logs requiere que una palabra clave esté cerca de los números. En la siguiente tabla se enumeran las palabras clave que CloudWatch Logs reconoce para países y regiones específicos.

País o región	Palabras clave
Australia	dl# dl:, dl :, dlno# driver licence, driver license, driver permit, drivers lic., drivers licence, driver's licence, drivers license, driver's license, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Austria	führerschein, fuhrerschein, führerschein republik österreich, fuhrerschein republik osterreich
Bélgica	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer, numéro permis conduire, permis de conduire, rijbewijs, rijbewijsnummer
Bulgaria	превозно средство, свидетелство за управление на моторно, свидетелство за управление на мпс, сумпс, шофьорска книжка
Canadá	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit,

País o región	Palabras clave
	drivers permit number, driving licence, driving license, driving permit, permis de conduire
Croacia	vozačka dozvola
Chipre	άδεια οδήγησης
República Checa	číslo licence, číslo licence řidiče, číslo řidičskéh o průkazu, ovladače lic., povolení k jízdě, povolení řidiče, řidiči povolení, řidičský prúkaz, řidičský průkaz
Dinamarca	kørekort, kørekortnummer
Estonia	juhi litsentsi number, juhiloa number, juhiluba, juhiluba number
Finlandia	ajokortin numero, ajokortti, förare lic., körkort, körkort nummer, kuljettaja lic., permis de conduire
Francia	permis de conduire
Alemania	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer
Grecia	δεια οδήγησης, adeia odigisis
Hungría	illesztőprogramok lic, jogosítvány, jogsi, licencszám, vezető engedély, vezetői engedély
Irlanda	ceadúnas tiomána
Italia	patente di guida, patente di guida numero, patente guida, patente guida numero

País o región	Palabras clave
Letonia	autovadītāja apliecība, licences numurs, vadītāja apliecība, vadītāja apliecības numurs, vadītāja atļauja, vadītāja licences numurs, vadītāji lic.
Lituania	vairuotojo pažymėjimas
Luxemburgo	fahrerlaubnis, führerschäin
Malta	liċenzja tas-sewqan
Países Bajos	permis de conduire, rijbewijs, rijbewijsnummer
Polonia	numer licencyjny, prawo jazdy, zezwolenie na prowadzenie
Portugal	carta de condução, carteira de habilitação, carteira de motorist, carteira habilitação, carteira motorist, licença condução, licença de condução, número de licença, número licença, permissão condução, permissão de condução
Rumanía	numărul permisului de conducere, permis de conducere
Eslovaquia	číslo licencie, číslo vodičského preukazu, ovládače lic., povolenia vodičov, povolenie jazdu, povolenie na jazdu, povolenie vodiča, vodičský preukaz
Eslovenia	vozniško dovoljenje

País o región	Palabras clave
España	carnet conducer, el carnet de conducer, licencia conducer, licencia de manejo, número carnet conducer, número de carnet de conducer, número de permiso conducer, número de permiso de conducer, número licencia conducer, número permiso conducer, permiso conducción, permiso conducer, permiso de conducción
Suecia	ajokortin numero, dlno# ajokortti, drivere lic., förare lic., körkort, körkort nummer, körkortsn ummer, kuljettajat lic.
Reino Unido	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Estados Unidos	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit

Palabras clave para números de documentos nacionales de identificación

Para detectar varios tipos de números de identificación nacionales, CloudWatch Logs requiere que una palabra clave esté muy cerca de los números. Esto incluye los identificadores del documento

nacional de identidad (DNI) (España), los códigos del Instituto Nacional de Estadística y Estudios Económicos (INSEE) de Francia, los números del documento nacional de identidad alemán y los números del Registro Geral (RG) (Brasil).

En la siguiente tabla se enumeran las palabras clave que CloudWatch Logs reconoce para países y regiones específicos.

País o región	Palabras clave
Brasil	registro geral, rg
Francia	assurance sociale, carte nationale d’identit é, cni, code sécurité sociale, French social security number, fssn#, insee, insurance number, national id number, nationalid#, numéro d'assurance, sécurité sociale, sécurité sociale non., sécurité sociale numéro, social, social security, social security number, socialsecuritynumber, ss#, ssn, ssn#
Alemania	ausweisnummer, id number, identification number, identity number, insurance number, personal id, personalausweis
Italia	codice fiscal, dati anagrafici, ehic, health card, health insurance card, p. iva, partita i.v.a., personal data, tax code, tessera sanitaria
España	dni, dni#, dninúmero#, documento nacional de identidad, identidad único, identidadúnico#, insurance number, national identification number, national identity, nationalid#, nationali dno#, número nacional identidad, personal identification number, personal identity no, unique identity number, uniqueid#

Palabras clave para números de pasaporte

Para detectar varios tipos de números de pasaporte, CloudWatch Logs requiere que una palabra clave esté cerca de los números. En la siguiente tabla se enumeran las palabras clave que CloudWatch Logs reconoce para países y regiones específicos.

País o región	Palabras clave
Canadá	passeport, passeport#, passport, passport#, passportno, passportno#
Francia	numéro de passeport, passeport, passeport #, passeport #, passeportn °, passeport n °, passeportNon, passeport non
Alemania	ausstellungsdatum, ausstellungsort, geburtsdatum, passport, passports, reisepass, reisepass–nr, reisepassnummer
Italia	italian passport number, numéro passeport, numéro passeport italien, passaporto, passaporto italiana, passaporto numero, passport number, repubblica italiana passaporto
España	españa pasaporte, libreta pasaporte, número pasaporte, pasaporte, passport, passport book, passport no, passport number, spain passport
Reino Unido	passeport #, passeport n °, passeportNon, passeport non, passeportn °, passport #, passport no, passport number, passport#, passportid
Estados Unidos	passport, travel document

Palabras clave para números de identificación y referencia del contribuyente

Para detectar varios tipos de números de identificación y referencia del contribuyente, CloudWatch Logs requiere que una palabra clave esté cerca de los números. En la siguiente tabla se enumeran las palabras clave que CloudWatch Logs reconoce para países y regiones específicos.

País o región	Palabras clave
Brasil	cadastro de pessoa física, cadastro de pessoa fisica, cadastro de pessoas físicas, cadastro de pessoas fisicas, cadastro nacional da pessoa jurídica, cadastro nacional da pessoa juridica, cnpj, cpf
Francia	numéro d'identification fiscale, tax id, tax identification number, tax number, tin, tin#
Alemania	identifikationsnummer, steuer id, steueride ntifikationsnummer, steuernummer, tax id, tax identification number, tax number
España	cif, cif número, cifnúmero#, nie, nif, número de contribuyente, número de identidad de extranjero, número de identificación fiscal, número de impuesto corporativo, personal tax number, tax id, tax identification number, tax number, tin, tin#
Reino Unido	paye, tax id, tax id no., tax id number, tax identification, tax identification#, tax no., tax number, tax reference, tax#, taxid#, temporary reference number, tin, trn, unique tax reference, unique taxpayer reference, utr
Estados Unidos	número de identificación tributaria individual (ITIN)

ARN de identificadores de datos para la información de identificación personal (PII)

A continuación, se enumeran los nombres de recursos de Amazon (ARN) para los identificadores de datos de información de identificación personal (PII) que puede agregar a sus políticas de protección de datos.

ARN de identificador de datos de PII
arn:aws:dataprotection::aws:data-identifier/Address
arn:aws:dataprotection::aws:data-identifier/CepCode-BR
arn:aws:dataprotection::aws:data-identifier/Cnpj-BR
arn:aws:dataprotection::aws:data-identifier/CpfCode-BR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-AT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-AU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-BE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-BG
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CA
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CY
arn:aws:dataprotection::aws:data-identifier/DriversLicense-CZ
arn:aws:dataprotection::aws:data-identifier/DriversLicense-DE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-DK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-EE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-ES
arn:aws:dataprotection::aws:data-identifier/DriversLicense-FI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-FR

ARN de identificador de datos de PII
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GB
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LV
arn:aws:dataprotection::aws:data-identifier/DriversLicense-MT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-NL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-RO
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-US
arn:aws:dataprotection::aws:data-identifier/ElectoralRollNumber-GB
arn:aws:dataprotection::aws:data-identifier/EmailAddress

ARN de identificador de datos de PII
arn:aws:dataprotection::aws:data-identifier/IndividualTaxIdentificationNumber-US
arn:aws:dataprotection::aws:data-identifier/InseeCode-FR
arn:aws:dataprotection::aws:data-identifier/LatLong
arn:aws:dataprotection::aws:data-identifier/Name
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-DE
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-ES
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-IT
arn:aws:dataprotection::aws:data-identifier/NieNumber-ES
arn:aws:dataprotection::aws:data-identifier/NifNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-CA
arn:aws:dataprotection::aws:data-identifier/PassportNumber-DE
arn:aws:dataprotection::aws:data-identifier/PassportNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-FR
arn:aws:dataprotection::aws:data-identifier/PassportNumber-GB
arn:aws:dataprotection::aws:data-identifier/PassportNumber-IT
arn:aws:dataprotection::aws:data-identifier/PassportNumber-US
arn:aws:dataprotection::aws:data-identifier/PermanentResidenceNumber-CA

ARN de identificador de datos de PII
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-BR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-DE
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-ES
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-FR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-GB
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-IT
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-US
arn:aws:dataprotection::aws:data-identifier/PostalCode-CA
arn:aws:dataprotection::aws:data-identifier/RgNumber-BR
arn:aws:dataprotection::aws:data-identifier/SocialInsuranceNumber-CA
arn:aws:dataprotection::aws:data-identifier/Ssn-ES
arn:aws:dataprotection::aws:data-identifier/Ssn-US
arn:aws:dataprotection::aws:data-identifier/TaxId-DE
arn:aws:dataprotection::aws:data-identifier/TaxId-ES
arn:aws:dataprotection::aws:data-identifier/TaxId-FR
arn:aws:dataprotection::aws:data-identifier/TaxId-GB
arn:aws:dataprotection::aws:data-identifier/VehicleIdentificationNumber
arn:aws:dataprotection::aws:data-identifier/ZipCode-US

Identificadores de datos personalizados

Temas

¿Qué son los identificadores de datos personalizados?

Los identificadores de datos personalizados (CDI) le permiten definir las propias expresiones regulares personalizadas que se pueden utilizar en la política de protección de datos. Con los identificadores de datos personalizados, puede centrarse en los casos de uso de la información de identificación personal (PII) específica de la empresa que los identificadores de datos administrados no pueden proporcionar. Por ejemplo, puedes usar un identificador de datos personalizado para buscar identificaciones de empleados específicas de la empresa. Los identificadores de datos personalizados se pueden utilizar junto con los identificadores de datos administrados.

Restricciones de identificadores de datos personalizados

CloudWatch Los identificadores de datos personalizados de los registros tienen las siguientes limitaciones:

- Cada política de protección de datos admite un máximo de 10 identificadores de datos personalizados.
- Los nombres de identificadores de datos personalizados tienen una longitud máxima de 128 caracteres. Se admiten los siguientes caracteres:
  - Alfanumérico: (a-zA-Z0-9)
  - Símbolos: ("_" |"-")
- RegEx tiene una longitud máxima de 200 caracteres. Se admiten los siguientes caracteres:
  - Alfanumérico: (a-zA-Z0-9)
  - Símbolos: ("_" | "#" | "=" | "@" | / | ";" | "," | "-" |)
  - Caracteres reservados de RegEx: ("^" | "$" | "?" | "[" | "]" | "{" | "}" | "|" | "\\" | "*" | "+" | ".")
- Los identificadores de datos personalizados no pueden compartir el mismo nombre que un identificador de datos administrado.

- Los identificadores de datos personalizados se pueden especificar en una política de protección de datos a nivel de cuenta o en políticas de protección de datos a nivel de grupo de registros. Al igual que los identificadores de datos gestionados, los identificadores de datos personalizados definidos en una política a nivel de cuenta funcionan en combinación con los identificadores de datos personalizados definidos en una política a nivel de grupo de registros.

Uso de identificadores de datos personalizados en la consola

Cuando utiliza la CloudWatch consola para crear o editar una política de protección de datos, para especificar un identificador de datos personalizado solo tiene que introducir un nombre y una expresión regular para el identificador de datos. Por ejemplo, puede escribir **`Employee_ID`** como nombre y **`EmployeeID-\d{9}`** como expresión regular. Esta expresión regular detectará y ocultará los eventos del registro con nueve números después`EmployeeID-`. Por ejemplo, `EmployeeID-123456789`

Uso de identificadores de datos personalizados en la política de protección de datos

Si utiliza la AWS API AWS CLI o para especificar un identificador de datos personalizado, debe incluir el nombre del identificador de datos y la expresión regular en la política de JSON utilizada para definir la política de protección de datos. La siguiente política de protección de datos detecta y oculta los eventos de registro que contienen identificaciones de empleados específicas de la empresa.

1. Cree un bloque de `Configuration` en la política de protección de datos.
2. Ingrese un `Name` para el identificador de datos personalizado. Por ejemplo, **`EmployeeId`**.
3. Ingrese un `Regex` para el identificador de datos personalizado. Por ejemplo, **`EmployeeID-\d{9}`**. Esta expresión regular coincidirá con los eventos de registro `EmployeeID-` que contengan nueve dígitos después. `EmployeeID-` Por ejemplo, `EmployeeID-123456789`
4. Consulte el siguiente identificador de datos personalizado en una instrucción de la política.

```
{
    "Name": "example_data_protection_policy",
    "Description": "Example data protection policy with custom data identifiers",
    "Version": "2021-06-01",
    "Configuration": {
      "CustomDataIdentifier": [
        {"Name": "EmployeeId", "Regex": "EmployeeId-\\d{9}"}
      ]
    },
    "Statement": [
```

```
        {
            "Sid": "audit-policy",
            "DataIdentifier": [
                "EmployeeId"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
            "EmployeeId"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {
                    }
                }
            }
        }
    ]
}
```

5. (Opcional) Siga agregando identificadores de datos personalizados adicionales al bloque de `Configuration` según sea necesario. Las políticas de protección de datos admiten actualmente un máximo de 10 identificadores de datos personalizados.

Creación de métricas a partir de eventos de registro mediante filtros

Puede buscar y filtrar los datos de registro que entran en CloudWatch los registros creando uno o más filtros de métricas. Los filtros métricos definen los términos y patrones que se deben buscar en los datos de registro a medida que se envían a CloudWatch los registros. CloudWatch Logs utiliza estos filtros de métricas para convertir los datos de registro en CloudWatch métricas numéricas que se pueden representar gráficamente o activar una alarma.

Al crear una métrica a partir de un filtro de registro, también puede optar por asignar dimensiones y una unidad a la métrica. Si especifica una unidad, asegúrese de especificar la correcta cuando cree el filtro. Cambiar la unidad del filtro más tarde no tendrá ningún efecto.

> **Note**
>
> Los filtros métricos solo se admiten para los grupos de registros de la clase de registro estándar. Para obtener más información sobre las clases de registro, consulteClases de registro.

Puede usar cualquier tipo de CloudWatch estadística, incluidas las estadísticas de percentiles, al ver estas métricas o configurar las alarmas.

> **Note**
>
> Las métricas admiten estadísticas de percentiles solo si ninguno de sus valores es negativo. Si configura el filtro de métricas para que pueda notificar números negativos, las estadísticas de percentiles no estarán disponibles para esa métrica cuando tenga valores de números negativos. Para obtener más información, consulte Percentiles.

Los filtros no pueden filtrar datos retroactivamente. Los filtros solo publican los puntos de datos de métricas para eventos que ocurran después de la creación del filtro. Los resultados filtrados devuelven las primeras 50 líneas, que no se mostrarán si la marca temporal en los resultados filtrados es anterior al momento de la creación de la métrica.

Contenido

Conceptos

Cada filtro de métrica se compone de los siguientes elementos principales:

valor predeterminado

El valor registrado en el filtro de métricas durante un periodo cuando se ingieren registros, pero no se encuentra ningún registro coincidente. Al configurar este valor como 0, garantiza que los datos se registran durante cada periodo, lo que impide métricas “irregulares” con periodos en los que no hay datos coincidentes. Si no se ingieren registros durante un periodo de un minuto, no se notifica ningún valor.

Si asigna dimensiones a una métrica creada por un filtro de métrica, no puede asignar un valor predeterminado a esa métrica.

dimensiones

Las dimensiones son los pares de valor de clave que definen aún más una métrica. Puede asignar dimensiones a la métrica creada a partir de un filtro de métrica. Dado que las dimensiones forman parte del identificador único de una métrica, cada vez que se extrae un par nombre/valor único de sus registros, se crea una nueva variación de esa métrica.

patrón de filtro

Una descripción simbólica de cómo CloudWatch los registros deben interpretar los datos de cada evento de registro. Por ejemplo, una entrada de registro puede contener las marcas temporales, direcciones IP, cadenas, etc. Puede utilizar el patrón para especificar lo que hay que buscar en el archivo de registro.

nombre de métrica

El nombre de la CloudWatch métrica en la que se debe publicar la información de registro supervisada. Por ejemplo, puede publicar en una métrica llamada ErrorCount.

espacio de nombres de métrica

El espacio de nombres de destino de la nueva CloudWatch métrica.

valor de métrica

El valor numérico para publicar en la métrica cada vez que se encuentra un registro coincidente. Por ejemplo, si está contando las incidencias de un término determinado como "Error", el valor será "1" para cada incidencia. Si está contando los bytes transferidos, puede incrementar según el número real de bytes encontrados en el evento de registro.

Sintaxis del patrón de filtro para filtros métricos

> **Note**
>
> En qué se diferencian los filtros métricos y las consultas de CloudWatch Logs
> Los filtros de métricas se diferencian de las consultas de CloudWatch Logs Insights en que se agrega un valor numérico específico a un filtro de métricas cada vez que se encuentra un registro coincidente. Para obtener más información, consulte Configuración de valores de métrica para un filtro de métricas.
> Para obtener información sobre cómo consultar sus grupos de CloudWatch registros con el lenguaje de consulta de Amazon Logs Insights, consulteCloudWatch Sintaxis de consultas de Logs Insights.
> Ejemplos de patrones de filtro genéricos
> Para obtener más información sobre la sintaxis del patrón de filtro genérico aplicable a los filtros de métricas, así como a los filtros de suscripción y a los eventos de registro de filtros, consulte Sintaxis de patrones de filtros para filtros de métricas, filtros de suscripción y eventos de registro de filtros, que incluye los siguientes ejemplos:
>
> - Sintaxis de expresiones regulares (regex) compatibles
> - Coincidencia de términos en eventos de registro no estructurado
> - Comparación de términos en eventos de registro JSON
> - Coincidencia de términos en eventos de registro delimitados por espacios

Los filtros métricos le permiten buscar y filtrar los datos de registro que ingresan en CloudWatch Logs, extraer observaciones métricas de los datos de registro filtrados y transformar los puntos de datos en una métrica de CloudWatch Logs. Usted define los términos y patrones que se deben

buscar en los datos de registro a medida que se envían a CloudWatch Logs. Los filtros de métricas se asignan a grupos de registro y todos los filtros asignados a un grupo de registros se aplican a sus flujos de registro.

Cuando un filtro de métricas coincide con un término, incrementa el recuento de la métrica a un valor numérico especificado. Por ejemplo, puede crear un filtro de métricas cuente cuántas veces aparece la palabra ERROR en los eventos de registro.

Puede asignar unidades de medida y dimensiones a una métrica. Por ejemplo, si crea un filtro de métricas que cuenta las veces que aparece la palabra ERROR en los eventos de registro, puede especificar una dimensión denominada `ErrorCode` para mostrar el número total de eventos de registro que contienen la palabra ERROR y filtrar los datos por códigos de error notificados.

> **Tip**
>
> Al asignar una unidad de medida a una métrica, asegúrese de especificar la correcta. Si cambia la unidad más adelante, es posible que el cambio no surta efecto. Para ver la lista completa de las unidades CloudWatch compatibles, consulta la referencia MetricDatumde la CloudWatch API de Amazon.

Temas

- Configuración de valores de métrica para un filtro de métricas
- Publicar dimensiones con métricas de valores en JSON o eventos de registro delimitados por espacios
- Uso de valores en eventos de registro para aumentar el valor de una métrica

Configuración de valores de métrica para un filtro de métricas

Al crear un filtro de métricas, defina el patrón de filtro y especifique el valor y el valor predeterminado de la métrica. Puede establecer valores de métrica en números, identificadores con nombre o identificadores numéricos. Si no especificas un valor predeterminado, CloudWatch no se mostrarán los datos cuando el filtro de métricas no encuentre ninguna coincidencia. Se recomienda especificar un valor predeterminado, incluso si el valor es 0. Establecer un valor predeterminado ayuda a CloudWatch informar los datos con mayor precisión y CloudWatch evita la agregación de métricas irregulares. CloudWatch agrega e informa los valores de las métricas cada minuto.

Cuando el filtro de métricas encuentra una coincidencia en los eventos de registro, incrementa el recuento de la métrica según el valor de esta. Si el filtro de métricas no encuentra ninguna coincidencia, muestra CloudWatch el valor predeterminado de la métrica. Por ejemplo, su grupo de registro publica dos registros cada minuto, el valor de la métrica es 1 y el valor predeterminado es 0. Si el filtro de métricas encuentra coincidencias en ambos registros en el primer minuto, el valor de la métrica para ese minuto es 2. Si el filtro de métricas no encuentra coincidencias en ninguno de los registros durante el segundo minuto, el valor predeterminado para ese minuto es 0. Si asigna dimensiones a las métricas que generan los filtros de métricas, no puede especificar los valores predeterminados para esas métricas.

También puede configurar un filtro de métricas para incrementar una métrica con un valor extraído de un evento de registro, en lugar de un valor estático. Para obtener más información, consulte Uso de valores en eventos de registro para aumentar el valor de una métrica.

Publicar dimensiones con métricas de valores en JSON o eventos de registro delimitados por espacios

Puede usar la CloudWatch consola o la AWS CLI para crear filtros de métricas que publiquen dimensiones con métricas generadas por JSON y eventos de registro delimitados por espacios. Las dimensiones son pares de nombre/valor y solo están disponibles para patrones de filtro JSON y delimitados por espacios. Puede crear filtros de métricas JSON y delimitados por espacios con hasta tres dimensiones. Para obtener más información sobre las dimensiones y sobre cómo asignarlas a las métricas, consulte las siguientes secciones:

- Dimensiones en la guía del CloudWatch usuario de Amazon
- Ejemplo: extraer campos de un registro de Apache y asignar dimensiones en la Guía del usuario de Amazon CloudWatch Logs

> **Important**
>
> Las dimensiones contienen valores que recopilan cargos igual que las métricas personalizadas. Para evitar cargos inesperados, no especifique campos de alta cardinalidad, como `IPAddress` o `requestID`, como dimensiones.
> Si extrae métricas de eventos de registro, se le cobran como métricas personalizadas. Para evitar que contraiga cargos elevados accidentales, es posible que Amazon desactive su filtro de métrica si genera 1000 pares de nombre/valor diferentes para las dimensiones especificadas en un plazo determinado.

Puede crear alarmas de facturación que le notifiquen los cargos estimados. Para obtener más información, consulte Crear una alarma de facturación para controlar AWS los cargos estimados.

Publicar dimensiones con métricas de eventos de registro JSON

En los ejemplos siguientes, se incluyen fragmentos de código que describen cómo especificar dimensiones en un filtro de métricas JSON.

Example: JSON log event

```
{
  "eventType": "UpdateTrail",
  "sourceIPAddress": "111.111.111.111",
  "arrayKey": [
        "value",
        "another value"
  ],
  "objectList": [
       {"name": "a",
         "id": 1
       },
       {"name": "b",
         "id": 2
       }
  ]

}
```

Note

Si prueba el filtro de métricas de ejemplo con el evento de registro JSON de ejemplo, debe ingresar el registro JSON de ejemplo en una sola línea.

Example: Metric filter

El filtro de métricas incrementa la métrica siempre que un evento de registro JSON contiene las propiedades `eventType` y `"sourceIPAddress"`.

```
{ $.eventType = "*" && $.sourceIPAddress != 123.123.* }
```

Al crear un filtro de métricas JSON, puede especificar cualquiera de las propiedades del filtro de métricas como una dimensión. Por ejemplo, para establecer `eventType` como dimensión, utilice lo siguiente:

```
"eventType" : $.eventType
```

La métrica de ejemplo contiene una dimensión denominada `"eventType"`, y el valor de la dimensión en el evento de registro de muestra es `"UpdateTrail"`.

Publicar dimensiones con métricas de eventos de registro delimitados por espacios

En los ejemplos siguientes, se incluyen fragmentos de código que describen cómo especificar dimensiones en un filtro de métricas delimitado por espacios.

Example: Space-delimited log event

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

Example: Metric filter

```
[ip, server, username, timestamp, request, status_code, bytes > 1000]
```

El filtro de métricas incrementa la métrica cuando un evento de registro delimitado por espacios incluye cualquiera de los campos especificados en el filtro. Por ejemplo, el filtro de métricas

encuentra los siguientes campos y valores en el evento de registro delimitado por espacios de ejemplo.

```
{
   "$bytes": "1534",
   "$status_code": "404",

   "$request": "GET /index.html HTTP/1.0",
   "$timestamp": "10/Oct/2000:13:25:15 -0700",
   "$username": "frank",
   "$server": "Prod",
   "$ip": "127.0.0.1"
}
```

Al crear un filtro de métricas delimitado por espacios, puede especificar cualquiera de los campos del filtro de métricas como una dimensión. Por ejemplo, para establecer `server` como dimensión, utilice lo siguiente:

```
"server" : $server
```

El filtro de métrica de ejemplo tiene una dimensión denominada `server`, y el valor de la dimensión en el evento de registro de muestra es `"Prod"`.

Example: Match terms with AND (&&) and OR (||)

Puede utilizar los operadores lógicos AND (“&&”) y OR (“||”) para crear filtros de métricas delimitados por espacios que contengan condiciones. El siguiente filtro de métricas devuelve eventos de registro en los que la primera palabra de los eventos es ERROR o supercadena de WARN.

```
[w1=ERROR || w1=%WARN%, w2]
```

Uso de valores en eventos de registro para aumentar el valor de una métrica

Puede crear filtros de métricas que publiquen los valores numéricos que se encuentran en los eventos de registro. El procedimiento de esta sección utiliza el siguiente filtro de métricas de ejemplo

para mostrar cómo se puede publicar un valor numérico en un evento de registro JSON en una métrica.

```
{ $.latency = * } metricValue: $.latency
```

Para crear un filtro de métricas que publique un valor en un evento de registro

1. Abre la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Log groups (Grupos de registros).
3. Seleccione o cree un grupo de registros.

   Para obtener información sobre cómo crear un grupo de registros, consulte Crear un grupo de CloudWatch registros en Logs en la Guía del usuario de Amazon CloudWatch Logs.
4. Elija Actions (Acciones) y, a continuación, seleccione Create metric filter (Crear filtro de métrica).
5. En Filter Pattern (Patrón de filtro), ingrese **`{ $.latency = * }`** y, a continuación, elija Next (Siguiente).
6. En Metric Name (Nombre de métrica), ingrese myMetric.
7. En Metric Value (Valor de métrica), escriba **`$.latency`**.
8. (Opcional) En Default Value (Valor predeterminado), ingrese 0 y, a continuación, elija Next (Siguiente).

   Se recomienda especificar un valor predeterminado, incluso si el valor es 0. Establecer un valor predeterminado ayuda a CloudWatch informar los datos con mayor precisión y CloudWatch evita la agregación de métricas irregulares. CloudWatch agrega e informa los valores de las métricas cada minuto.
9. Elija Create metric filter (Crear filtro de métricas).

El filtro de métricas de ejemplo coincide con el término `"latency"` en el evento de registro JSON de muestra y publica un valor numérico de 50 en la métrica myMetric.

```
{
"latency": 50,
"requestType": "GET"
}
```

Creación de filtros de métricas

Los siguientes procedimientos y ejemplos muestran cómo crear filtros de métricas.

Ejemplos

- Crear un filtro de métricas para un grupo de registros
- Ejemplo: recuento de eventos de registro
- Ejemplo: contar incidencias de un término
- Ejemplo: contar códigos HTTP 404
- Ejemplo: contar códigos HTTP 4xx
- Ejemplo: extraer campos desde un registro de Apache y asignar dimensiones

Crear un filtro de métricas para un grupo de registros

Para crear un filtro de métrica para un grupo de registros, siga los siguientes pasos. La métrica no estará visible hasta que haya algunos puntos de datos para ella.

Para crear un filtro de métricas mediante la CloudWatch consola

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros) y, luego, Log groups (Grupos de registros).
3. Elija el nombre del grupo de registros.
4. Elija `Actions` (Acciones) y, a continuación, seleccione Create metric filter (Crear filtro de métrica).
5. En Filter pattern (Patrón de filtro), ingrese un patrón de filtro. Para obtener más información, consulte Filtro de sintaxis de patrones para filtros de métricas, filtros de suscripción, eventos de registro de filtros y Live Tail.
6. (Opcional) Para probar el patrón de filtro, en Test Pattern (Patrón de prueba), ingrese uno o más eventos de registro para probar el patrón. Cada evento de registro debe estar formateado en una línea. Los saltos de línea se utilizan para separar los eventos de registro en el cuadro Mensajes de eventos de registro.
7. Elija Next (Siguiente) y luego ingrese un nombre para el filtro de métricas.
8. En Detalles de la métrica, en Espacio de nombres de métricas, introduzca un nombre para el espacio de CloudWatch nombres en el que se publicará la métrica. Si este espacio de nombres no existe todavía, asegúrese de que la opción Create new (Crear nuevo) esté seleccionada.

9. Para Metric name (Nombre de métrica), ingrese un nombre para la nueva métrica.
10. Para Metric value (Valor de la métrica), si el filtro de métrica cuenta las ocurrencias de las palabras clave en el filtro, ingrese 1. Esto incrementa la métrica en 1 por cada evento de registro que incluye una de las palabras clave.

    También puede ingresar un token, como **`$size`**. Esto incrementa la métrica por el valor del número en el campo `size` por cada evento de registro que contenga un campo `size`.
11. (Opcional) En Unit (Unidad), seleccione una unidad para asignar a la métrica. Si no especifica una unidad, se configura como `None`.
12. (Opcional) Ingrese los nombres y tokens de hasta tres dimensiones para la métrica. Si asigna dimensiones a las métricas que generan los filtros de métricas, no puede asignar valores predeterminados para esas métricas.

    > **Note**
    >
    > Las dimensiones solo se admiten en JSON o en filtros de métricas delimitados por espacios.
13. Elija Create metric filter (Crear filtro de métricas). Puede encontrar el filtro de métricas que ha creado desde el panel de navegación. Elija Logs (Registros) y, a continuación, elija Log groups (Grupo de registros). Elija el nombre del grupo de registro para el que ha creado el filtro de métricas y, a continuación, seleccione la pestaña Metric filters (Filtros de métricas).

Ejemplo: recuento de eventos de registro

El tipo de monitorización de evento de registro más sencillo consiste en contar el número de eventos de registro que se producen. Es posible que desee hacerlo para llevar un recuento de todos los eventos, para crear un monitor de estilo "latido" o simplemente para practicar la creación de filtros de métricas.

En el siguiente ejemplo de CLI, MyAppAccessCount se aplica un filtro de métricas denominado MyApp /access.log al grupo de registros para crear la métrica EventCount en el espacio de CloudWatch nombres MyNamespace. El filtro está configurado para que compare cualquier contenido de eventos de registro y para aumentar la métrica en "1".

Para crear un filtro de métricas mediante la consola CloudWatch

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.

2. En el panel de navegación, seleccione Grupos de registro.
3. Elija el nombre de un grupo de registros.
4. Elija `Actions`, Create metric filter (Crear filtro de métricas).
5. Deje Filter Pattern (Patrón de filtro) y Select Log Data to Test (Seleccionar los datos de registro para probar) en blanco.
6. Elija Next (Siguiente), y, a continuación, en Filter Name (Nombre de filtro), escriba **`EventCount`**.
7. En Metric Details (Detalles de métrica), en Metric Namespace (Espacio de nombres de métrica), escriba **`MyNameSpace`**.
8. En Nombre de métrica, escriba **`MyAppEventCount`**.
9. Confirme que el Metric Value (Valor métrico) es 1. Esto especifica que el recuento se incrementa en 1 para cada evento de registro.
10. En Default Value (Valor predeterminado), escriba 0 y, a continuación, elija Next (Siguiente). Al especificar un valor predeterminado se garantiza que los datos se registren incluso durante los periodos en los que no se producen eventos de registro, lo que impide que haya métricas irregulares en las que a veces no existen datos.
11. Elija Create metric filter (Crear filtro de métricas).

Para crear un filtro métrico mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name EventCount \
  --filter-pattern " " \
  --metric-transformations \
  metricName=MyAppEventCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Puede probar esta nueva política publicando cualesquiera datos de eventos. Deberías ver los puntos de datos publicados en la métrica MyAppAccessEventCount.

Para publicar los datos del evento mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando:

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
```

```
  --log-events \
    timestamp=1394793518000,message="Test event 1" \
    timestamp=1394793518000,message="Test event 2" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Ejemplo: contar incidencias de un término

Los eventos de registro suelen incluir mensajes importantes que desea contar, quizás referentes al éxito o fracaso de las operaciones. Por ejemplo, puede producirse un error y registrarse en un archivo de registro si falla una determinada operación. Es posible que desee monitorizar estas entradas para comprender la evolución de sus errores.

En el ejemplo siguiente, se crea un filtro de métricas para monitorizar el término Error. La política se creó y se agregó al grupo de registros MyApp/message.log. CloudWatch Logs publica un punto de datos ErrorCount en la métrica CloudWatch personalizada del espacio de nombres MyApp/message.log con un valor de «1" para cada evento que contenga un error. Si ningún evento contiene la palabra Error, entonces se publica un valor 0. Al graficar estos datos en la CloudWatch consola, asegúrese de utilizar la estadística de suma.

Después de crear un filtro de métricas, puede ver la métrica en la CloudWatch consola. Cuando seleccione la métrica que desea ver, seleccione el espacio de nombres de métrica que coincida con el nombre del grupo de registros. Para obtener más información, consulte Viewing Available Metrics (Visualización de las métricas disponibles).

Para crear un filtro de métricas mediante la CloudWatch consola

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. Elija el nombre del grupo de registros.
4. Elija Actions (Acciones), Create metric filter (Crear filtro de métricas).
5. En Filter pattern (Patrón de filtro), escriba **`Error`**.

   > **Note**
   >
   > Todas las entradas de Filter Pattern distinguen entre mayúsculas y minúsculas.

6. (Opcional) Para probar el patrón de filtro, en Test Pattern (Patrón de prueba), ingrese uno o más eventos de registro a utilizar para probar el patrón. Cada evento de registro debe estar dentro de

una línea, ya que los saltos de línea se utilizan para separar los eventos de registro en el cuadro de Log event messages (Mensajes de eventos de registro).

7. Elija Next (Siguiente), y, a continuación, en la página Filter Name (Asignar métrica), en Filter Name (Nombre de filtro), escriba **`MyAppErrorCount`**.
8. En Metric Details, en Metric Namespace, escriba. MyNameSpace
9. En Nombre de métrica, escriba ErrorCount.
10. Confirme que el Metric Value (Valor métrico) es 1. Esto especifica que el recuento se incrementa en 1 para cada evento de registro que contenga "Error".
11. En Default Value (Valor predeterminado) escriba 0 y, a continuación, elija Next (Siguiente).
12. Elija Create metric filter (Crear filtro de métricas).

Para crear un filtro métrico mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando:

```
aws logs put-metric-filter \
  --log-group-name MyApp/message.log \
  --filter-name MyAppErrorCount \
  --filter-pattern 'Error' \
  --metric-transformations \
      metricName=ErrorCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Puede probar esta nueva política publicando eventos que contengan la palabra "Error" en el mensaje.

Para publicar eventos mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando. Tenga en cuenta que los patrones distinguen entre mayúsculas y minúsculas.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="This message contains an Error" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Ejemplo: contar códigos HTTP 404

Con CloudWatch los registros, puede controlar cuántas veces sus servidores Apache devuelven una respuesta HTTP 404, que es el código de respuesta de la página no encontrada. Es posible que le interese monitorizar esto para saber con qué frecuencia los visitantes no encuentran el recurso que buscan. Supongamos que los registros se estructuran para incluir la siguiente información para cada evento de registro (visita al sitio):

- Dirección IP del solicitante
- Identidad RFC 1413
- Nombre de usuario
- Timestamp
- Solicitar método con recurso solicitado y protocolo
- Código de respuesta HTTP para solicitud
- Bytes transferidos en solicitud

Un ejemplo de esto podría ser el siguiente:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 404 2326
```

Podría especificar una regla que intente comparar eventos con dicha estructura para errores HTTP 404, tal y como se muestra en el ejemplo siguiente:

Para crear un filtro métrico mediante la CloudWatch consola

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. Elija `Actions`, Create metric filter (Crear filtro de métricas).
4. En Filter pattern (Patrón de filtro), escriba **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`**.
5. (Opcional) Para probar el patrón de filtro, en Test Pattern (Patrón de prueba), ingrese uno o más eventos de registro a utilizar para probar el patrón. Cada evento de registro debe estar dentro de una línea, ya que los saltos de línea se utilizan para separar los eventos de registro en el cuadro de Log event messages (Mensajes de eventos de registro).

6. Elija Next (Siguiente), y, a continuación, para Filter name (Nombre de filtro), escriba HTTP404Errors.
7. En Metric Details (Detalles de métrica), en Metric Namespace (Espacio de nombres de métrica), escriba **`MyNameSpace`**.
8. En Metric name (Nombre de métrica), escriba **`ApacheNotFoundErrorCount`**.
9. Confirme que el Metric Value (Valor métrico) es 1. Esto especifica que el recuento se incrementa en 1 para cada evento de Error 404.
10. En Default Value (Valor predeterminado), escriba 0 y, a continuación, elija Next (Siguiente).
11. Elija Create metric filter (Crear filtro de métricas).

Para crear un filtro métrico mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP404Errors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=404, size]' \
  --metric-transformations \
      metricName=ApacheNotFoundErrorCount,metricNamespace=MyNamespace,metricValue=1
```

En este ejemplo, se han utilizado caracteres literales como los corchetes izquierdo y derecho, las comillas dobles y la cadena de caracteres 404. El patrón tiene que coincidir con todo el mensaje de evento de registro para que el evento de registro se tenga en cuenta para monitorización.

Puede verificar la creación del filtro de métricas a través del comando describe-metric-filters. Debería ver un resultado con un aspecto similar al siguiente:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log

{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
```

```
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

Ahora puede publicar unos cuantos eventos manualmente:

```
aws logs put-log-events \
--log-group-name MyApp/access.log --log-stream-name hostname \
--log-events \
timestamp=1394793518000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 404 2326" \
timestamp=1394793528000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb2.gif HTTP/1.0\" 200 2326"
```

Poco después de colocar estos eventos de registro de ejemplo, puede recuperar la métrica denominada en la CloudWatch consola como ApacheNotFoundErrorCount.

Ejemplo: contar códigos HTTP 4xx

Como en el ejemplo anterior, es posible que desee monitorizar los registros de acceso al servicio web y monitorizar los niveles del código de respuesta HTTP. Por ejemplo, es posible que desee monitorizar todos los errores de nivel HTTP 400. Sin embargo, es posible que no desee especificar un nuevo filtro de métrica para cada código devuelto.

El siguiente ejemplo muestra cómo crear una métrica que incluya todas las respuestas de código HTTP de nivel 400 desde registro de acceso utilizando el formato de registro de acceso de Apache desde el ejemplo Ejemplo: contar códigos HTTP 404.

Para crear un filtro de métricas mediante la CloudWatch consola

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. Elija el nombre del grupo de registros para el servidor Apache.
4. Elija `Actions`, Create metric filter (Crear filtro de métricas).

5. En Filter pattern (Patrón de filtro), ingrese **[ip, id, user, timestamp, request, status_code=4*, size]**.
6. (Opcional) Para probar el patrón de filtro, en Test Pattern (Patrón de prueba), ingrese uno o más eventos de registro a utilizar para probar el patrón. Cada evento de registro debe estar dentro de una línea, ya que los saltos de línea se utilizan para separar los eventos de registro en el cuadro de Log event messages (Mensajes de eventos de registro).
7. Elija Next (Siguiente) y, a continuación, en Filter Name (Nombre de filtro), tipo **HTTP4xxErrors**.
8. En Metric Details (Detalles de métrica), en Metric Namespace (Espacio de nombres de métrica), ingrese **MyNameSpace**.
9. En Metric Name (Nombre de métrica), ingrese HTTP4xxErrors.
10. En Metric Value (Valor de métrica), ingrese 1. Esto especifica que el recuento se incrementa en 1 para cada evento de registro que contenga un error 4xx.
11. En Default Value (Valor predeterminado), escriba 0 y, a continuación, elija Next (Siguiente).
12. Elija Create metric filter (Crear filtro de métricas).

Para crear un filtro métrico mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP4xxErrors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=4*, size]' \
  --metric-transformations \
  metricName=HTTP4xxErrors,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Puede utilizar los siguientes datos en llamadas PutEvents para probar esta regla. Si no elimina la regla de monitorización en el ejemplo anterior, generará dos métricas diferentes.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Ejemplo: extraer campos desde un registro de Apache y asignar dimensiones

A veces, en lugar de contar, se recomienda utilizar valores dentro de eventos de registro individuales para valores de métricas. Este ejemplo muestra cómo puede crear una regla de extracción para crear una métrica que mida los bytes transferidos por un servidor web Apache.

En este ejemplo también se muestra cómo asignar dimensiones a la métrica que se crea.

Para crear un filtro métrico mediante la CloudWatch consola

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. Elija el nombre del grupo de registros para el servidor Apache.
4. Elija `Actions`, Create metric filter (Crear filtro de métricas).
5. En Filter pattern (Patrón de filtro), ingrese **`[ip, id, user, timestamp, request, status_code, size]`**.
6. (Opcional) Para probar el patrón de filtro, en Test Pattern (Patrón de prueba), ingrese uno o más eventos de registro a utilizar para probar el patrón. Cada evento de registro debe estar dentro de una línea, ya que los saltos de línea se utilizan para separar los eventos de registro en el cuadro de Log event messages (Mensajes de eventos de registro).
7. Elija Next (Siguiente) y, a continuación, en Filter Name (Nombre de filtro), tipo **`size`**.
8. En Metric Details (Detalles de métrica), en Metric Namespace (Espacio de nombres de métrica), ingrese **`MyNameSpace`**. Debido a que este es un nuevo espacio de nombres, asegúrese de que la opción Create new (Crear nuevo) esté seleccionada.
9. En Metric name (Nombre de métrica), ingrese **`BytesTransferred`**.
10. En Metric Value (Valor de métrica), ingrese **`$size`**.
11. En Unit (Unidad), seleccione Bytes.
12. Para Dimension Name, escriba **`IP`**.
13. En Dimension Value (Valor de dimensión) escriba **`$ip`** y, a continuación, elija Next (Siguiente).
14. Elija Create metric filter (Crear filtro de métricas).

Para crear este filtro de métricas mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
 --filter-name BytesTransferred \
 --filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
 --metric-transformations  \
 metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size'
```

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
--filter-name BytesTransferred \
--filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
--metric-transformations  \
metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue='$size',unit=Bytes,dimensio
$ip}}'
```

> **Note**
>
> En este comando, utilice este formato para especificar varias dimensiones.
>
> ```
> aws logs put-metric-filter \
> --log-group-name my-log-group-name \
> --filter-name my-filter-name \
> --filter-pattern 'my-filter-pattern' \
> --metric-transformations  \
> metricName=my-metric-name,metricNamespace=my-metric-namespace,metricValue=my-
> token,unit=unit,dimensions='{dimension1=$dim,dimension2=$dim2,dim3=$dim3}'
> ```

Puede usar los siguientes datos en las put-log-event llamadas para probar esta regla. Esto genera dos métricas diferentes si no elimina la regla de monitorización en el ejemplo anterior.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Enumeración de filtros de métricas

Puede enumerar todos los filtros de métricas de un grupo de registros.

Para enumerar los filtros de métricas mediante la CloudWatch consola

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. En el panel de contenido, en la lista de grupos de registros, en la columna Metric Filters, elija el número de filtros.

   La pantalla Log Groups > Filters for muestra todos los filtros de métricas asociados con el grupo de registros.

Para enumerar los filtros de métricas mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

A continuación, se muestra un ejemplo de la salida:

```
{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404,
 size]"
        }
    ]
}
```

Eliminación de un filtro de métricas

Una política se identifica por su nombre y el grupo de registros al que pertenece.

Para eliminar un filtro de métricas mediante la CloudWatch consola

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. En el panel de contenido, en la columna Metric Filter (Filtros de métrica), elija el número de filtros de métrica para el grupo de registros.
4. En la pantalla de Metric Filters (Filtros de métricas), seleccione la casilla de verificación a la derecha del nombre del filtro que desea eliminar. A continuación, elija Eliminar.
5. Cuando se le pida confirmación, seleccione Eliminar.

Para eliminar un filtro de métricas mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando:

```
aws logs delete-metric-filter --log-group-name MyApp/access.log \
 --filter-name MyFilterName
```

Procesamiento en tiempo real de datos de registros con suscripciones

Puede utilizar las suscripciones para obtener acceso a una transmisión en tiempo real de los eventos de registro de CloudWatch Logs y hacer que se entregue a otros servicios, como una transmisión de Amazon Kinesis o una transmisión de Amazon Data Firehose, o AWS Lambda para su procesamiento, análisis o carga personalizados en otros sistemas. Cuando se envían eventos de registro al servicio de recepción, estos están codificados en base64 y comprimidos con el formato gzip.

Para comenzar a suscribirse a eventos de registro, cree el recurso de recepción como, por ejemplo, un flujo de Kinesis Data Streams, donde se enviarán los eventos. Un filtro de suscripción define el patrón de filtrado que se utilizará para filtrar qué eventos de registro se envían a su AWS recurso, así como información sobre a dónde enviar los eventos de registro coincidentes.

Puede crear suscripciones a nivel de cuenta y a nivel de grupo de registros. Cada cuenta puede tener un filtro de suscripción a nivel de cuenta. Cada grupo de registros puede tener asociado hasta dos filtros de suscripción.

> **Note**
>
> Si el servicio de destino devuelve un error que se puede volver a intentar, como una excepción de limitación o una excepción de servicio que se puede volver a intentar (HTTP 5xx, por ejemplo), CloudWatch Logs seguirá reintentando la entrega durante un máximo de 24 horas. CloudWatch Logs no intenta volver a realizar la entrega si el error no se puede volver a intentar, como o. AccessDeniedException ResourceNotFoundException En estos casos, el filtro de suscripción se deshabilita durante un máximo de 10 minutos y, a continuación, CloudWatch Logs vuelve a intentar enviar los registros al destino. Durante este período de inhabilitación, se omiten los registros.

CloudWatch Los registros también generan CloudWatch métricas sobre el reenvío de los eventos de registro a las suscripciones. Para obtener más información, consulte Monitorización con CloudWatch métricas.

También puedes usar una suscripción a CloudWatch Logs para transmitir datos de registro prácticamente en tiempo real a un clúster de Amazon OpenSearch Service. Para obtener más

información, consulta Cómo transmitir datos de CloudWatch registros a Amazon OpenSearch Service.

Las suscripciones solo se admiten para los grupos de registros de la clase de registro estándar. Para obtener más información sobre las clases de registro, consulteClases de registro.

> **Note**
>
> Los filtros de suscripción pueden registrar eventos por lotes para optimizar la transmisión y reducir la cantidad de llamadas realizadas al destino. El procesamiento por lotes no está garantizado, pero se utiliza siempre que sea posible.

Contenido

Conceptos

Cada filtro de suscripción se compone de los siguientes elementos principales:

patrón de filtro

Una descripción simbólica de cómo CloudWatch los registros deben interpretar los datos de cada evento de registro, junto con expresiones de filtrado que restringen lo que se entrega al AWS recurso de destino. Para obtener más información acerca de la sintaxis del patrón de filtro, consulte Filtro de sintaxis de patrones para filtros de métricas, filtros de suscripción, eventos de registro de filtros y Live Tail.

arn de destino

El nombre del recurso de Amazon (ARN) de la transmisión de Kinesis Data Streams, la transmisión de Firehose o la función Lambda que desee utilizar como destino del feed de suscripción.

arn de rol

Una función de IAM que otorga a CloudWatch Logs los permisos necesarios para colocar los datos en el destino elegido. Esta función no es necesaria para los destinos de Lambda porque los CloudWatch registros pueden obtener los permisos necesarios de la configuración de control de acceso de la propia función Lambda.

distribución

El método utilizado para distribuir los datos de registro al destino, cuando el destino es un flujo de Amazon Kinesis Data Streams. De forma predeterminada, los datos de registro se agrupan por flujo de registro. Para obtener una distribución más uniforme, puede agrupar los datos de registro de forma aleatoria.

En el caso de las suscripciones a nivel de grupo de registros, también se incluye el siguiente elemento clave:

nombre de grupo de registro

El grupo de registro al que asociar el filtro de suscripción. Todos los eventos de registros cargados en este grupo de registros estarían sujetos al filtro de suscripción y los que coinciden con el filtro se entregarían al servicio de destino que recibe los eventos de registro coincidentes.

En el caso de las suscripciones a nivel de cuenta, también se incluye el siguiente elemento clave:

criterios de selección

Los criterios utilizados para seleccionar los grupos de registros a los que se ha aplicado el filtro de suscripción a nivel de cuenta. Si no lo especifica, el filtro de suscripción a nivel de cuenta se aplica a todos los grupos de registros de la cuenta. Este campo se utiliza para evitar bucles de registro infinitos. Para obtener más información sobre el problema del bucle logarítmico infinito, consultePrevención de la recursión de registros.

Los criterios de selección tienen un límite de tamaño de 25 KB.

Filtros de suscripción a nivel de grupo de registros

Puede usar un filtro de suscripción con Kinesis Data Streams, Lambda o Firehose. Todos los registros enviados a un servicio de recepción a través de un filtro de suscripción están codificados en base64 y comprimidos con el formato gzip.

Puede buscar los datos de registro con la sintaxis de patrones y filtros.

Ejemplos

- Ejemplo 1: filtros de suscripción con Kinesis Data Streams
- Ejemplo 2: filtros de suscripción con AWS Lambda
- Ejemplo 3: filtros de suscripción con Amazon Data Firehose

Ejemplo 1: filtros de suscripción con Kinesis Data Streams

El siguiente ejemplo asocia un filtro de suscripción a un grupo de registros que contiene eventos. AWS CloudTrail El filtro de suscripción envía todas las actividades registradas realizadas por AWS las credenciales «Root» a una transmisión de Kinesis Data Streams denominada RootAccess «». Para obtener más información sobre cómo enviar AWS CloudTrail eventos a los CloudWatch registros, consulte Enviar CloudTrail eventos a los CloudWatch registros en la Guía del AWS CloudTrail usuario.

> **Note**
>
> Antes de crear el flujo de , calcule el volumen de los datos de registro que se generarán. Asegúrese de crear un flujo de con fragmentos suficientes para gestionar este volumen. Si el flujo no dispone de suficientes fragmentos, se limitará el flujo de registros. Para obtener más información sobre los límites del volumen del flujo, consulte Cuotas y límites.
> La entrega de los registros limitados se vuelve a intentar durante un máximo de 24 horas. Transcurridas 24 horas, las entregas fallidas se descartan.
> Para mitigar el riesgo de limitación, puede seguir estos pasos:
>
> - Especifique `random` para `distribution` cuando cree el filtro de suscripción con PutSubscriptionFilter o put-subscription-filter. De forma predeterminada, la distribución del filtro de flujo es por flujo de registro y esto puede provocar una limitación.
> - Supervisa tu transmisión mediante CloudWatch métricas. Esto lo ayudará a identificar cualquier limitación y a ajustar la configuración en consecuencia. Por ejemplo, la

`DeliveryThrottling` métrica se puede utilizar para hacer un seguimiento del número de eventos de registro por los que se CloudWatch limitó Logs al reenviar datos al destino de la suscripción. Para obtener más información sobre la supervisión, consulte Monitorización con CloudWatch métricas.

- Utilice el modo de capacidad bajo demanda para el flujo en Kinesis Data Streams. El modo bajo demanda se adapta de forma instantánea a sus cargas de trabajo a medida que aumentan o disminuyen. Para obtener más información sobre el modo de capacidad bajo demanda, consulte Modo bajo demanda.
- Restrinja el patrón de filtros de CloudWatch suscripción para que coincida con la capacidad de su transmisión en Kinesis Data Streams. Si envía demasiados datos al flujo, es posible que deba reducir el tamaño del filtro o ajustar sus criterios.

Para crear un filtro de suscripción para Kinesis Data Streams

1. Crear un flujo de de destino utilizando el siguiente comando:

```
$ C:\>  aws kinesis create-stream --stream-name "RootAccess" --shard-count 1
```

2. Espere hasta que el flujo de esté Activo (esto podría tardar un minuto o dos). Puede utilizar el siguiente comando describe-stream de Kinesis Data Streams para comprobar la. StreamDescription StreamStatuspropiedad. Además, anote el valor StreamDescription.streamArn, ya que lo necesitará en un paso posterior:

```
aws kinesis describe-stream --stream-name "RootAccess"
```

   A continuación, se muestra un ejemplo de la salida:

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "RootAccess",
        "StreamARN": "arn:aws:kinesis:us-east-1:123456789012:stream/RootAccess",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "340282366920938463463374607431768211455",
                    "StartingHashKey": "0"
```

```
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "49551135218688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Crea el rol de IAM que otorgará a CloudWatch Logs el permiso para colocar datos en tu transmisión. En primer lugar, tendrá que crear una política de confianza en un archivo (por ejemplo, `~/TrustPolicyForCWL-Kinesis.json`). Utilice un editor de texto para crear esta política. No utilice la consola de IAM para crearla.

   Esta política incluye una clave de contexto de condición global de `aws:SourceArn` para ayudar a prevenir el confuso problema de seguridad adjunto. Para obtener más información, consulte Prevención del suplente confuso.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. Ejecute el comando create-role para crear el rol de IAM y especifique el archivo de política de confianza. Anote el valor de Role.Arn devuelto, ya que también lo necesitará más tarde:

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

   A continuación se muestra un ejemplo de la salida.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
```

```
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. Crea una política de permisos para definir qué acciones puede realizar CloudWatch Logs en tu cuenta. En primer lugar, creará una política de permisos en un archivo (por ejemplo, `~/PermissionsForCWL-Kinesis.json`). Utilice un editor de texto para crear esta política. No utilice la consola de IAM para crearla.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/RootAccess"
    }
  ]
}
```

6. Asocie la política de permisos al rol mediante el siguiente put-role-policycomando:

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. Una vez que la transmisión esté en estado activo y hayas creado el rol de IAM, puedes crear el filtro de suscripción a CloudWatch Logs. El filtro de suscripción inicia de inmediato el flujo de datos de registro en tiempo real desde el grupo de registros elegido a su flujo de :

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RootAccess" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:kinesis:region:123456789012:stream/RootAccess" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

8. Tras configurar el filtro de suscripción, CloudWatch Logs reenvía a la transmisión todos los eventos de registro entrantes que coincidan con el patrón de filtrado. Puede verificar que esto está ocurriendo si toma un iterador de particiones de Kinesis Data Streams y utiliza el comando get-records de Kinesis Data Streams para obtener algunos registros de Kinesis Data Streams:

```
aws kinesis get-shard-iterator --stream-name RootAccess --shard-id
 shardId-000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Tenga en cuenta que es posible que tenga que realizar esta llamada varias veces antes de que Kinesis Data Streams comience a devolver los datos.

Cabe esperar ver una respuesta en una gama de registros. El atributo Datos de un registro de Kinesis Data Streams tiene codificación de base64 y está comprimido con el formato gzip.

Puede examinar los datos sin procesar desde la línea de comando utilizando los siguientes comandos de Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Los datos descodificados y descomprimidos en base64 se formatean como JSON con la siguiente estructura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
}
```

Los elementos clave en la estructura de datos anterior son los siguientes:

owner

El ID de AWS cuenta de los datos de registro originarios.

logGroup

El nombre del grupo de registros de los datos de registro de origen.

logStream

El nombre del flujo de registros de los datos de registro de origen.

subscriptionFilters

La lista de nombres de filtros de suscripción que coincide con los datos de registro de origen.

messageType

Los mensajes de datos utilizarán el tipo "DATA_MESSAGE". A veces, CloudWatch los registros pueden emitir registros de Kinesis Data Streams del tipo «CONTROL_MESSAGE», principalmente para comprobar si se puede acceder al destino.

logEvents

Los datos de registro reales, representados como un conjunto de registros de eventos de registro. La propiedad "id" es un identificador único de cada evento de registro.

Ejemplo 2: filtros de suscripción con AWS Lambda

En este ejemplo, crearás un filtro de suscripción de CloudWatch registros que envía los datos de registro a tu AWS Lambda función.

> **Note**
>
> Antes de crear la función Lambda, calcule el volumen de los datos de registro que se generarán. Asegúrese de crear una función que pueda gestionar este volumen. Si la función no dispone de suficiente volumen, se limitará el flujo de registros. Para obtener más información sobre los límites de Lambda, consulte Límites de AWS Lambda.

Para crear un filtro de suscripción para Lambda

1. Crea la AWS Lambda función.

Asegúrese de haber configurado el rol de ejecución de Lambda. Para obtener más información, consulte Paso 2.2: Crear un rol de IAM (rol de ejecución) en la AWS Lambda Guía del desarrollador.

2. Abra un editor de texto y cree un archivo denominado `helloWorld.js` con el siguiente contenido:

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Comprima el archivo helloWorld.js en un zip y guárdelo con el nombre `helloWorld.zip`.

4. Utilice el siguiente comando, donde el rol es el rol de ejecución de Lambda que configuró en el primer paso:

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs12.x
```

5. CloudWatch Concede a Logs el permiso para ejecutar tu función. Utilice el siguiente comando, sustituyendo la cuenta del marcador por su propia cuenta y el grupo de registros del marcador por el grupo de registros que procesar:

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
```

```
    --source-arn "arn:aws:logs:region:123456789123:log-group:TestLambda:*" \
    --source-account "123456789012"
```

6. Cree un filtro de suscripción utilizando el siguiente comando, sustituyendo la cuenta del marcador por su propia cuenta y el grupo de registros del marcador por el grupo de registros que procesar:

```
aws logs put-subscription-filter \
    --log-group-name myLogGroup \
    --filter-name demo \
    --filter-pattern "" \
    --destination-arn arn:aws:lambda:region:123456789123:function:helloworld
```

7. (Opcional) Probar mediante un evento de registro de ejemplo. En el símbolo del sistema, ejecute el siguiente comando, que pone un mensaje de registro sencillo en el flujo suscrito.

   Para consultar la salida de su función de Lambda, diríjase a la función de Lambda donde verá la salida en /aws/lambda/helloworld:

```
aws logs put-log-events --log-group-name myLogGroup --log-stream-name stream1 --
log-events "[{\"timestamp\":<CURRENT TIMESTAMP MILLIS> , \"message\": \"Simple
 Lambda Test\"}]"
```

   Cabe esperar ver una respuesta en una matriz de Lambda. El atributo Data (Datos) del registro de Lambda tiene codificación de base64 y está comprimido con el formato gzip. La carga útil real que recibe Lambda está en el siguiente formato `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`. Puede examinar los datos sin procesar desde la línea de comandos mediante los siguientes comandos de Unix:

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

   Los datos descodificados y descomprimidos en base64 se formatean como JSON con la siguiente estructura:

```
{
    "owner": "123456789012",
    "logGroup": "CloudTrail",
    "logStream": "123456789012_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
```

```
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ]
}
```


Los elementos clave en la estructura de datos anterior son los siguientes:

owner

El ID de AWS cuenta de los datos de registro originarios.

logGroup

El nombre del grupo de registros de los datos de registro de origen.

logStream

El nombre del flujo de registros de los datos de registro de origen.

subscriptionFilters

La lista de nombres de filtros de suscripción que coincide con los datos de registro de origen.

messageType

Los mensajes de datos utilizarán el tipo "DATA_MESSAGE". A veces, CloudWatch los registros pueden emitir registros Lambda del tipo «CONTROL_MESSAGE», principalmente para comprobar si se puede acceder al destino.

logEvents

Los datos de registro reales, representados como un conjunto de registros de eventos de registro. La propiedad "id" es un identificador único de cada evento de registro.

Ejemplo 3: filtros de suscripción con Amazon Data Firehose

En este ejemplo, crearás una suscripción a CloudWatch Logs que enviará todos los eventos de registro entrantes que coincidan con tus filtros definidos a tu transmisión de entrega de Amazon Data Firehose. Los datos enviados desde CloudWatch los registros a Amazon Data Firehose ya están comprimidos con la compresión gzip de nivel 6, por lo que no es necesario utilizar la compresión en la transmisión de entrega de Firehose. A continuación, puedes usar la función de descompresión de Firehose para descomprimir automáticamente los registros. Para obtener más información, consulte Cómo escribir en Kinesis Data CloudWatch Firehose mediante registros.

> **Note**
>
> Antes de crear la transmisión Firehose, calcule el volumen de datos de registro que se generarán. Asegúrate de crear una transmisión de Firehose que pueda soportar este volumen. Si el flujo no puede gestionar el volumen, se limitará el flujo de registros. Para obtener más información sobre los límites de volumen de transmisión de Firehose, consulte Amazon Data Firehose Data Limits.

Para crear un filtro de suscripción para Firehose

1. Cree un bucket de Amazon Simple Storage Service (Amazon S3). Te recomendamos que utilices un depósito creado específicamente para CloudWatch los registros. Sin embargo, si desea utilizar un bucket existente, vaya al paso 2.

   Ejecute el siguiente comando y sustituya la región del marcador por la región que desee utilizar:

```
aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
 LocationConstraint=region
```

A continuación, se muestra un ejemplo de la salida:

```
{
    "Location": "/my-bucket"
}
```

2. Cree la función de IAM que conceda permiso a Amazon Data Firehose para colocar datos en su bucket de Amazon S3.

   Para obtener más información, consulte Control del acceso con Amazon Data Firehose en la Guía para desarrolladores de Amazon Data Firehose.

   En primer lugar, utilice un editor de texto para crear una política de confianza en un archivo `~/TrustPolicyForFirehose.json` tal como se indica a continuación:

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "firehose.amazonaws.com" },
    "Action": "sts:AssumeRole"
    }
}
```

3. Ejecute el comando create-role para crear el rol de IAM y especifique el archivo de política de confianza. Anote el valor de Role.Arn devuelto, ya que lo necesitará en un paso posterior:

```
aws iam create-role \
 --role-name FirehosetoS3Role \
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
```

```
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Crea una política de permisos para definir qué acciones puede realizar Firehose en tu cuenta. En primer lugar, utilice un editor de texto para crear una política de permisos en un archivo ~/PermissionsForFirehose.json:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::my-bucket",
          "arn:aws:s3:::my-bucket/*" ]
    }
  ]
}
```

5. Asocia la política de permisos al rol mediante el siguiente put-role-policy comando:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Cree una transmisión de entrega de Firehose de destino de la siguiente manera y sustituya los valores de los marcadores de posición de RoLearn y BucketArn por los ARN de rol y bucket que creó:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

Tenga en cuenta que Firehose utiliza automáticamente un prefijo en el formato de hora UTC AAAA/MM/DD/HH para los objetos de Amazon S3 entregados. Puede especificar un prefijo adicional que añadir delante del prefijo de formato de hora. Si el prefijo termina con una barra inclinada (/), aparece como una carpeta en el bucket de Amazon S3.

7. Espere hasta que el flujo se active (esto podría tardar unos minutos). Puede utilizar el describe-delivery-streamcomando Firehose para comprobar el. DeliveryStreamDescription DeliveryStreamStatuspropiedad. Además, tenga en cuenta el DeliveryStreamDescription. DeliveryStreamValor ARN, ya que lo necesitará en un paso posterior:

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
```

```
                }
            }
        ]
    }
}
```

8. Crea el rol de IAM que otorga permiso a CloudWatch Logs para colocar datos en tu transmisión de entrega de Firehose. En primer lugar, utilice un editor de texto para crear una política de confianza en un archivo `~/TrustPolicyForCWL.json`:

   Esta política incluye una clave de contexto de condición global de `aws:SourceArn` para ayudar a prevenir el confuso problema de seguridad adjunto. Para obtener más información, consulte Prevención del suplente confuso.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
     }
  }
}
```

9. Ejecute el comando create-role para crear el rol de IAM y especifique el archivo de política de confianza. Anote el valor de Role.Arn devuelto, ya que lo necesitará en un paso posterior:

```
aws iam create-role \
--role-name CWLtoKinesisFirehoseRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
```

```
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Crea una política de permisos para definir qué acciones puede realizar CloudWatch Logs en tu cuenta. En primer lugar, utilice un editor de texto para crear un archivo de política de permisos (por ejemplo, ~/PermissionsForCWL.json):

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. Asocie la política de permisos al rol mediante el put-role-policy comando:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Una vez que la transmisión de entrega de Amazon Data Firehose esté en estado activo y haya creado el rol de IAM, podrá crear el filtro de suscripción a CloudWatch Logs. El filtro de suscripción inicia inmediatamente el flujo de datos de registro en tiempo real desde el grupo de registros elegido a la transmisión de entrega de Amazon Data Firehose:

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "Destination" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
```

13. Tras configurar el filtro de suscripción, CloudWatch Logs reenviará todos los eventos de registro entrantes que coincidan con el patrón de filtrado a la transmisión de entrega de Amazon Data Firehose. Sus datos comenzarán a aparecer en su Amazon S3 en función del intervalo de tiempo establecido en la transmisión de entrega de Amazon Data Firehose. Una vez que haya transcurrido el tiempo suficiente, puede verificar los datos comprobando su bucket de Amazon S3.

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2015-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-35-40-7cc92023-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
```

```
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2015/10/29/00/my-
delivery-stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2015 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

Los datos en el objeto de Amazon S3 se comprimen con el formato gzip. Puede examinar los datos sin procesar desde la línea de comando utilizando el siguiente comando de Unix:

```
zcat testfile.gz
```

Filtros de suscripción a nivel de cuenta

> **Important**
>
> Existe el riesgo de provocar un bucle recursivo infinito con los filtros de suscripción que, si no se abordan, pueden provocar un gran aumento de la facturación por ingestión. Para mitigar este riesgo, le recomendamos que utilice criterios de selección en los filtros de suscripción a nivel de cuenta para excluir los grupos de registros que ingieren datos de registro de los recursos que forman parte del flujo de trabajo de entrega de suscripciones. Para obtener más información sobre este problema y determinar qué grupos de registros excluir, consulte. Prevención de la recursión de registros

Puede establecer una política de suscripción a nivel de cuenta que incluya un subconjunto de grupos de registros en la cuenta. La política de suscripción de la cuenta puede funcionar con Kinesis Data Streams, Lambda o Firehose. Los registros que se envían a un servicio receptor mediante una política de suscripción a nivel de cuenta están codificados en base64 y comprimidos en formato gzip.

> **Note**
>
> Para ver una lista de todas las políticas de filtrado de suscripciones de tu cuenta, usa el `describe-account-policies` comando con un valor de `SUBSCRIPTION_FILTER_POLICY` para el parámetro. `--policy-type` Para obtener más información, consulte describe-account-policies¶.

Ejemplos

- Ejemplo 1: filtros de suscripción con Kinesis Data Streams
- Ejemplo 2: filtros de suscripción con AWS Lambda
- Ejemplo 3: filtros de suscripción con Amazon Data Firehose

Ejemplo 1: filtros de suscripción con Kinesis Data Streams

Antes de crear una transmisión de datos de Kinesis Data Streams para usarla con una política de suscripción a nivel de cuenta, calcule el volumen de datos de registro que se generará. Asegúrese de crear un flujo de con fragmentos suficientes para gestionar este volumen. Si una transmisión no tiene suficientes fragmentos, se limita. Para obtener más información sobre los límites de volumen de transmisión, consulte Cuotas y límites en la documentación de Kinesis Data Streams.

> **Warning**
>
> Como los eventos de registro de varios grupos de registros se reenvían al destino, existe el riesgo de que se limiten. La entrega de los registros limitados se vuelve a intentar durante un máximo de 24 horas. Transcurridas 24 horas, las entregas fallidas se descartan.
> Para mitigar el riesgo de limitación, puede seguir estos pasos:
>
> - Supervise su transmisión de Kinesis Data Streams CloudWatch con métricas. Esto le ayuda a identificar las limitaciones y a ajustar la configuración en consecuencia. Por ejemplo, la `DeliveryThrottling` métrica registra el número de eventos de registro por los que se ha limitado CloudWatch Logs al reenviar los datos al destino de la suscripción. Para obtener más información, consulte Monitorización con CloudWatch métricas.
> - Utilice el modo de capacidad bajo demanda para el flujo en Kinesis Data Streams. El modo bajo demanda se adapta de forma instantánea a sus cargas de trabajo a medida que aumentan o disminuyen. Para obtener más información, consulte Modo bajo demanda.

- Restrinja el patrón de filtros de suscripción de CloudWatch Logs para que coincida con la capacidad de su transmisión en Kinesis Data Streams. Si envía demasiados datos al flujo, es posible que deba reducir el tamaño del filtro o ajustar sus criterios.

El siguiente ejemplo utiliza una política de suscripción a nivel de cuenta para reenviar todos los eventos de registro a una transmisión en Kinesis Data Streams. El patrón de filtro hace coincidir todos los eventos del registro con el texto `Test` y los reenvía a la transmisión en Kinesis Data Streams.

Para crear una política de suscripción a nivel de cuenta para Kinesis Data Streams

1. Crear un flujo de de destino utilizando el siguiente comando:

```
$ C:\>  aws kinesis create-stream –stream-name "TestStream" –shard-count 1
```

2. Espere unos minutos hasta que la transmisión se active. Puede comprobar si la transmisión está activa mediante el comando describe-stream para comprobar la. StreamDescription StreamStatuspropiedad.

```
aws kinesis describe-stream --stream-name "TestStream"
```

A continuación, se muestra un ejemplo de la salida:

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "TestStream",
        "StreamARN": "arn:aws:kinesis:region:123456789012:stream/TestStream",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "EXAMPLE846346337460743176821145 5",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "EXAMPLE6888184566795038319814587845913527021815726 10"
                }
```

```
            }
        ]
    }
}
```

3. Crea el rol de IAM que otorgará permiso a CloudWatch Logs para incluir datos en tu transmisión. En primer lugar, tendrá que crear una política de confianza en un archivo (por ejemplo, `~/TrustPolicyForCWL-Kinesis.json`). Utilice un editor de texto para crear esta política.

   Esta política incluye una clave de contexto de condición global de `aws:SourceArn` para ayudar a prevenir el confuso problema de seguridad adjunto. Para obtener más información, consulte Prevención del suplente confuso.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
        "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
     }
   }
}
```

4. Ejecute el comando create-role para crear el rol de IAM y especifique el archivo de política de confianza. Anote el valor de Role.Arn devuelto, ya que también lo necesitará más tarde:

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document
 file://~/TrustPolicyForCWL-Kinesis.json
```

   A continuación se muestra un ejemplo de la salida.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
```

```
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "EXAMPLE450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
    }
}
```

5. Crea una política de permisos para definir qué acciones puede realizar CloudWatch Logs en tu cuenta. En primer lugar, creará una política de permisos en un archivo (por ejemplo, `~/PermissionsForCWL-Kinesis.json`). Utilice un editor de texto para crear esta política. No utilices la consola de IAM para crearlo.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/TestStream"
    }
  ]
}
```

6. Asocie la política de permisos al rol mediante el siguiente put-role-policycomando:

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-Policy-For-CWL  --policy-document file://~/PermissionsForCWL-Kinesis.json
```

7. Una vez que la transmisión esté en estado activo y haya creado el rol de IAM, puede crear la política de filtrado de suscripciones de CloudWatch registros. La política inicia inmediatamente el flujo de datos de registro en tiempo real a tu transmisión. En este ejemplo, se transmiten todos los eventos de registro que contienen la cadena`ERROR`, excepto los de los grupos de registros denominados `LogGroupToExclude1` y`LogGroupToExclude2`.

```
aws logs put-account-policy \
    --policy-name "ExamplePolicy" \
```

```
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisRole", "DestinationArn":"arn:aws:kinesis:region:123456789012:stream/
TestStream", "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

8. Tras configurar el filtro de suscripción, CloudWatch Logs reenvía a la transmisión todos los eventos de registro entrantes que coincidan con el patrón de filtro y los criterios de selección.

   El `selection-criteria` campo es opcional, pero es importante para excluir los grupos de registros que pueden provocar una recursión de registro infinita en un filtro de suscripción. Para obtener más información sobre este problema y determinar qué grupos de registros se van a excluir, consultePrevención de la recursión de registros. Actualmente, NOT IN es el único operador compatible`selection-criteria`.

   Puede comprobar el flujo de eventos de registro mediante un iterador de fragmentos de Kinesis Data Streams y el comando Kinesis Data Streams para obtener algunos registros de Kinesis `get-records` Data Streams:

```
aws kinesis get-shard-iterator --stream-name TestStream --shard-id
 shardId-000000000000 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL
+wev+e2P4djJg4L9wmXKvQYoE+rMUiFq
+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f+OIK8zM5My8ID
+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Puede que necesite usar este comando varias veces antes de que Kinesis Data Streams comience a devolver datos.

Cabe esperar ver una respuesta en una gama de registros. El atributo Datos de un registro de Kinesis Data Streams tiene codificación de base64 y está comprimido con el formato gzip. Puede examinar los datos sin procesar desde la línea de comando utilizando los siguientes comandos de Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Los datos descodificados y descomprimidos en base64 se formatean como JSON con la siguiente estructura:

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicy"
    ],
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ],
```

```
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

Los elementos clave de la estructura de datos son los siguientes:

messageType

Los mensajes de datos utilizarán el tipo "DATA_MESSAGE". A veces, CloudWatch los registros pueden emitir registros de Kinesis Data Streams del tipo «CONTROL_MESSAGE», principalmente para comprobar si se puede acceder al destino.

owner

El ID de AWS cuenta de los datos de registro originarios.

logGroup

El nombre del grupo de registros de los datos de registro de origen.

logStream

El nombre del flujo de registros de los datos de registro de origen.

subscriptionFilters

La lista de nombres de filtros de suscripción que coincide con los datos de registro de origen.

logEvents

Los datos de registro reales, representados como un conjunto de registros de eventos de registro. La propiedad "id" es un identificador único de cada evento de registro.

Nivel de política

El nivel en el que se aplicó la política. «ACCOUNT_LEVEL_POLICY» es la política de filtrado de suscripciones a nivel `policyLevel` de cuenta.

Ejemplo 2: filtros de suscripción con AWS Lambda

En este ejemplo, crearás una política de filtrado de suscripciones a nivel de cuenta de CloudWatch Logs que envía los datos de registro a tu AWS Lambda función.

> **Warning**
>
> Antes de crear la función Lambda, calcule el volumen de los datos de registro que se generarán. Asegúrese de crear una función que pueda gestionar este volumen. Si la función no puede gestionar el volumen, el flujo de registros se reducirá. Como los eventos de registro de todos los grupos de registros o de un subconjunto de los grupos de registros de la cuenta se reenvían al destino, existe el riesgo de que se limiten. Para obtener más información sobre los límites de Lambda, consulte Límites de AWS Lambda.

Para crear una política de filtrado de suscripciones a nivel de cuenta para Lambda

1. Cree la función. AWS Lambda

   Asegúrese de haber configurado el rol de ejecución de Lambda. Para obtener más información, consulte Paso 2.2: Crear un rol de IAM (rol de ejecución) en la AWS Lambda Guía del desarrollador.
2. Abra un editor de texto y cree un archivo denominado `helloWorld.js` con el siguiente contenido:

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Comprima el archivo helloWorld.js en un zip y guárdelo con el nombre `helloWorld.zip`.
4. Utilice el siguiente comando, donde el rol es el rol de ejecución de Lambda que configuró en el primer paso:

```
aws lambda create-function \
    --function-name helloworld \
```

```
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs18.x
```

5. CloudWatch Concede a Logs el permiso para ejecutar tu función. Usa el siguiente comando para reemplazar la cuenta de marcador de posición por tu propia cuenta.

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789012:log-group:*" \
    --source-account "123456789012"
```

6. Cree una política de filtrado de suscripciones a nivel de cuenta mediante el siguiente comando y sustituya la cuenta de marcador de posición por la suya propia. En este ejemplo, se transmiten todos los eventos de registro que contienen la cadenaERROR, excepto los de los grupos de registro denominados y. `LogGroupToExclude1` `LogGroupToExclude2`

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyLambda" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:lambda:region:123456789012:function:helloWorld",
 "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Tras configurar el filtro de suscripción, CloudWatch Logs reenvía a la transmisión todos los eventos de registro entrantes que coincidan con el patrón de filtro y los criterios de selección.

El `selection-criteria` campo es opcional, pero es importante para excluir los grupos de registros que pueden provocar una recursión de registro infinita en un filtro de suscripción. Para obtener más información sobre este problema y determinar qué grupos de registros se van a excluir, consultePrevención de la recursión de registros. Actualmente, NOT IN es el único operador compatible`selection-criteria`.

7. (Opcional) Probar mediante un evento de registro de ejemplo. En el símbolo del sistema, ejecute el siguiente comando, que pone un mensaje de registro sencillo en el flujo suscrito.

   Para consultar la salida de su función de Lambda, diríjase a la función de Lambda donde verá la salida en /aws/lambda/helloworld:

```
aws logs put-log-events --log-group-name Example1 --log-stream-name logStream1 --
log-events "[{\"timestamp\":CURRENT TIMESTAMP MILLIS , \"message\": \"Simple Lambda
 Test\"}]"
```

   Cabe esperar ver una respuesta en una matriz de Lambda. El atributo Data (Datos) del registro de Lambda tiene codificación de base64 y está comprimido con el formato gzip. La carga útil real que recibe Lambda está en el siguiente formato `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`. Puede examinar los datos sin procesar desde la línea de comandos mediante los siguientes comandos de Unix:

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

   Los datos descodificados y descomprimidos en base64 se formatean como JSON con la siguiente estructura:

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicyLambda"
    ],
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
```

```
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"Root\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

> **Note**
>
> El filtro de suscripción a nivel de cuenta no se aplicará al grupo de registros de la función Lambda de destino. Esto se hace para evitar una recursión infinita de registros que pueda provocar un aumento en la facturación por ingestión. Para obtener más información sobre este problema, consulte. Prevención de la recursión de registros

Los elementos clave de la estructura de datos son los siguientes:

messageType

Los mensajes de datos utilizarán el tipo "DATA_MESSAGE". A veces, CloudWatch los registros pueden emitir registros de Kinesis Data Streams del tipo «CONTROL_MESSAGE», principalmente para comprobar si se puede acceder al destino.

owner

El ID de AWS cuenta de los datos de registro originarios.

logGroup

El nombre del grupo de registros de los datos de registro de origen.

logStream

El nombre del flujo de registros de los datos de registro de origen.

subscriptionFilters

La lista de nombres de filtros de suscripción que coincide con los datos de registro de origen.

logEvents

Los datos de registro reales, representados como un conjunto de registros de eventos de registro. La propiedad "id" es un identificador único de cada evento de registro.

Nivel de política

El nivel en el que se aplicó la política. «ACCOUNT_LEVEL_POLICY» es la política de filtrado de suscripciones a nivel `policyLevel` de cuenta.

Ejemplo 3: filtros de suscripción con Amazon Data Firehose

En este ejemplo, crearás una política de filtrado de suscripciones a nivel de cuenta de CloudWatch Logs que envíe los eventos de registro entrantes que coincidan con tus filtros definidos a tu transmisión de entrega de Amazon Data Firehose. Los datos enviados desde CloudWatch los registros a Amazon Data Firehose ya están comprimidos con la compresión gzip de nivel 6, por lo que no es necesario utilizar la compresión en la transmisión de entrega de Firehose. A continuación, puedes usar la función de descompresión de Firehose para descomprimir automáticamente los registros. Para obtener más información, consulte Cómo escribir en Kinesis Data CloudWatch Firehose mediante registros.

> **Warning**
>
> Antes de crear la transmisión Firehose, calcule el volumen de datos de registro que se generarán. Asegúrate de crear una transmisión de Firehose que pueda soportar este volumen. Si el flujo no puede gestionar el volumen, se limitará el flujo de registros. Para obtener más información sobre los límites de volumen de transmisión de Firehose, consulte Amazon Data Firehose Data Limits.

Para crear un filtro de suscripción para Firehose

1. Cree un bucket de Amazon Simple Storage Service (Amazon S3). Te recomendamos que utilices un depósito creado específicamente para CloudWatch los registros. Sin embargo, si desea utilizar un bucket existente, vaya al paso 2.

   Ejecute el siguiente comando y sustituya la región del marcador por la región que desee utilizar:

```
aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
 LocationConstraint=region
```

A continuación, se muestra un ejemplo de la salida:

```
{
    "Location": "/my-bucket"
}
```

2. Cree la función de IAM que conceda permiso a Amazon Data Firehose para colocar datos en su bucket de Amazon S3.

   Para obtener más información, consulte Control del acceso con Amazon Data Firehose en la Guía para desarrolladores de Amazon Data Firehose.

   En primer lugar, utilice un editor de texto para crear una política de confianza en un archivo `~/TrustPolicyForFirehose.json` tal como se indica a continuación:

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "firehose.amazonaws.com" },
    "Action": "sts:AssumeRole"
    }
}
```

3. Ejecute el comando create-role para crear el rol de IAM y especifique el archivo de política de confianza. Anote el valor de Role.Arn devuelto, ya que lo necesitará en un paso posterior:

```
aws iam create-role \
 --role-name FirehosetoS3Role \
 --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
```

```
                }
            }
        },
        "RoleId": "EXAMPLE50GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Crea una política de permisos para definir qué acciones puede realizar Firehose en tu cuenta. En primer lugar, utilice un editor de texto para crear una política de permisos en un archivo ~/PermissionsForFirehose.json:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::my-bucket",
          "arn:aws:s3:::my-bucket/*" ]
    }
  ]
}
```

5. Asocia la política de permisos al rol mediante el siguiente put-role-policy comando:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Cree una transmisión de entrega de Firehose de destino de la siguiente manera y sustituya los valores de los marcadores de posición de RoLearn y BucketArn por los ARN de rol y bucket que creó:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

NFireHose utiliza automáticamente un prefijo en formato de hora UTC AAAA/MM/DD/HH para los objetos Amazon S3 entregados. Puede especificar un prefijo adicional que añadir delante del prefijo de formato de hora. Si el prefijo termina con una barra inclinada (/), aparece como una carpeta en el bucket de Amazon S3.

7. Espere unos minutos hasta que la transmisión se active. Puede utilizar el describe-delivery-streamcomando Firehose para comprobar el. DeliveryStreamDescription DeliveryStreamStatuspropiedad. Además, tenga en cuenta el DeliveryStreamDescription. DeliveryStreamValor ARN, ya que lo necesitará en un paso posterior:

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:123456789012:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
```

```
                }
            }
        ]
    }
}
```

8. Crea el rol de IAM que otorga permiso a CloudWatch Logs para colocar datos en tu transmisión de entrega de Firehose. En primer lugar, utilice un editor de texto para crear una política de confianza en un archivo `~/TrustPolicyForCWL.json`:

   Esta política incluye una clave de contexto de condición global de `aws:SourceArn` para ayudar a prevenir el confuso problema de seguridad adjunto. Para obtener más información, consulte Prevención del suplente confuso.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
     }
  }
}
```

9. Ejecute el comando create-role para crear el rol de IAM y especifique el archivo de política de confianza. Anota el valor de Role.Arn devuelto, ya que lo necesitarás en un paso posterior:

```
aws iam create-role \
--role-name CWLtoKinesisFirehoseRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
```

```
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Crea una política de permisos para definir qué acciones puede realizar CloudWatch Logs en tu cuenta. En primer lugar, utilice un editor de texto para crear un archivo de política de permisos (por ejemplo, `~/PermissionsForCWL.json`):

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. Asocie la política de permisos al rol mediante el put-role-policy comando:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-
name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Una vez que la transmisión de entrega de Amazon Data Firehose esté en estado activo y haya creado la función de IAM, podrá crear la política de filtrado de suscripciones a nivel de cuenta de CloudWatch Logs. La política inicia inmediatamente el flujo de datos de registro en tiempo real desde el grupo de registros elegido a su flujo de entrega de Amazon Data Firehose:

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyFirehose" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisFirehoseRole", "DestinationArn":"arn:aws:firehose:us-
east-1:123456789012:deliverystream/delivery-stream-name", "FilterPattern": "Test",
 "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

13. Tras configurar el filtro de suscripción, CloudWatch Logs reenvía los eventos de registro entrantes que coinciden con el patrón de filtrado a la transmisión de entrega de Amazon Data Firehose.

    El `selection-criteria` campo es opcional, pero es importante para excluir de un filtro de suscripción los grupos de registros que pueden provocar una recursión infinita de registros. Para obtener más información sobre este problema y determinar qué grupos de registros se van a excluir, consulte Prevención de la recursión de registros. Actualmente, NOT IN es el único operador compatible `selection-criteria`.

    Sus datos comenzarán a aparecer en su Amazon S3 en función del intervalo de tiempo establecido en la transmisión de entrega de Amazon Data Firehose. Una vez que haya transcurrido el tiempo suficiente, puede verificar los datos comprobando su bucket de Amazon S3.

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2023-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
```

```
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2023/10/29/00/my-delivery-stream-2023-10-29-00-35-40-
EXAMPLE-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "EXAMPLE6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2023/10/29/00/my-
delivery-stream-2023-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2023 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

Los datos en el objeto de Amazon S3 se comprimen con el formato gzip. Puede examinar los datos sin procesar desde la línea de comando utilizando el siguiente comando de Unix:

```
zcat testfile.gz
```

Suscripciones multicuentas y regiones

Puede colaborar con un propietario de otra AWS cuenta y recibir sus eventos de registro en sus AWS recursos, como una transmisión de Amazon Kinesis o Amazon Data Firehose (esto se conoce como intercambio de datos entre cuentas). Por ejemplo, los datos de este registro de eventos se pueden leer desde una transmisión centralizada de Kinesis Data Streams o Firehose para realizar un

procesamiento y análisis personalizados. El procesamiento personalizado resulta especialmente útil al colaborar y analizar datos en muchas cuentas.

Por ejemplo, el grupo de seguridad de información de una empresa podría desear analizar datos de detección de intrusiones en tiempo real o de comportamientos anómala para poder realizar una auditoría de cuentas en todas las divisiones de la empresa recopilando sus registros de producción federada para procesamiento central. Un flujo en tiempo real de datos de eventos en dichas cuentas se puede montar y enviar a los grupos de seguridad de información que pueden utilizar Kinesis Data Streams para adjuntar los datos a sus sistemas de análisis de seguridad existentes.

> **Note**
>
> El grupo de registros y el destino deben estar en la misma AWS región. Sin embargo, el AWS recurso al que apunta el destino puede estar ubicado en una región diferente. En los ejemplos de las siguientes secciones, todos los recursos específicos de la región se crean en EE. UU. Este (Virginia del Norte).

Temas

Uso compartido de datos de registro entre cuentas y regiones mediante Kinesis Data Streams

Al crear una suscripción entre cuentas, puede especificar una única cuenta o una organización para que sea el remitente. Si especifica una organización, este procedimiento permite que todas las cuentas de la organización envíen registros a la cuenta de receptor.

Para compartir los datos de registro entre cuentas, es necesario establecer un emisor y receptor de datos de registro:

- Remitente de datos de registro: obtiene la información de destino del destinatario e informa a CloudWatch Logs de que está listo para enviar sus eventos de registro al destino especificado. En

los procedimientos descritos en el resto de esta sección, se muestra al remitente de los datos de registro con un número de AWS cuenta ficticio de 1111.

Si va a tener varias cuentas de una organización que envíen registros a una cuenta de destinatario, puede crear una política que otorgue a todas las cuentas de la organización el permiso para enviar registros a la cuenta de destinatario. Aún tiene que configurar filtros de suscripción independientes para cada cuenta de remitente.

- Destinatario de los datos de registro: configura un destino que encapsula una transmisión de Kinesis Data Streams y permite a CloudWatch Logs saber que el destinatario quiere recibir los datos de registro. A continuación, el destinatario comparte la información sobre este destino con el remitente. En los procedimientos que se describen en el resto de esta sección, se muestra al destinatario de los datos de registro con un número de AWS cuenta ficticio, el número de cuenta 999.

Para empezar a recibir eventos de registro de usuarios con varias cuentas, el destinatario de los datos de registro crea primero un destino de CloudWatch registros. Cada destino consta de los siguientes elementos fundamentales:

Nombre de destino

El nombre del destino que desea crear.

ARN de destino

El nombre del recurso de Amazon (ARN) del AWS recurso que quieres usar como destino del feed de suscripción.

ARN del rol

Un rol AWS Identity and Access Management (IAM) que otorga a CloudWatch Logs los permisos necesarios para colocar los datos en la transmisión elegida.

Política de acceso

Un documento de política de IAM (en formato JSON, escrito con la gramática de política de IAM) que rige el conjunto de los usuarios a los que se les permite escribir en su destino.

> **Note**
>
> El grupo de registros y el destino deben estar en la misma AWS región. Sin embargo, el recurso de AWS al que apunta el destino puede estar ubicado en una región diferente. En los

ejemplos de las secciones siguientes, todos los recursos específicos de una región se crean en EE. UU. Este (Norte de Virginia).

Temas

Configuración de una nueva suscripción entre cuentas

Siga los pasos de estas secciones para configurar una nueva suscripción de registro entre cuentas.

Temas

Paso 1: crear un destino

Important

Todos los pasos de este procedimiento deben realizarse en la cuenta del destinatario de los datos de registro.

En este ejemplo, la cuenta receptora de los datos de registro tiene un identificador de AWS cuenta de 10000 9999, mientras que el identificador de la AWS cuenta del remitente de los datos de registro es 1111.

En este ejemplo, se crea un destino mediante una transmisión de Kinesis Data Streams RecipientStream llamada y una función que CloudWatch permite a Logs escribir datos en ella.

Cuando se crea el destino, CloudWatch Logs envía un mensaje de prueba al destino en nombre de la cuenta del destinatario. Cuando el filtro de suscripciones se active más adelante, CloudWatch Logs envía los eventos de registro al destino en nombre de la cuenta de origen.

Para crear un destino

1. En la cuenta de destinatario, cree un flujo de destino en Kinesis Data Streams. En el símbolo del sistema, escriba:

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. Espere hasta que el flujo de se active. Puede utilizar el comando aws kinesis describe-stream para comprobar el. StreamDescription StreamStatuspropiedad. Además, tome nota del valor StreamDescription.streamArn porque lo pasará a CloudWatch Logs más adelante:

```
aws kinesis describe-stream --stream-name "RecipientStream"
{
  "StreamDescription": {
    "StreamStatus": "ACTIVE",
    "StreamName": "RecipientStream",
    "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702 18EXAMPLE"
        }
      }
    ]
  }
}
```

   El flujo puede tardar un minuto o dos en mostrarse en el estado activo.

3. Crea el rol de IAM que otorga a CloudWatch Logs el permiso para colocar datos en tu transmisión. En primer lugar, tendrás que crear una política de confianza en un archivo ~/

TrustPolicyFor CWL.json. Utilice un editor de texto para crear este archivo de política; no utilice la consola de IAM.

Esta política incluye una clave de contexto de condición global `aws:SourceArn` que especifica la `sourceAccountId` para ayudar a prevenir el problema de seguridad de suplente confuso. Si aún no conoce el ID de cuenta de origen en la primera llamada, le recomendamos que coloque el ARN de destino en el campo ARN de origen. En las llamadas posteriores, debe configurar el ARN de origen para que sea el ARN de origen real que recopiló desde la primera llamada. Para obtener más información, consulte Prevención del suplente confuso.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
}
```

4. Ejecute el comando aws iam create-role para crear el rol de IAM y especifique el archivo de política de confianza. Toma nota del valor Role.Arn devuelto porque también se pasará a Logs más adelante: CloudWatch

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
```

```
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Crea una política de permisos para definir qué acciones puede realizar CloudWatch Logs en tu cuenta. Primero, usa un editor de texto para crear una política de permisos en un archivo ~/PermissionsFor CWL.json:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Asocie la política de permisos al rol mediante el comando aws iam: put-role-policy

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Una vez que la transmisión esté en estado activo y haya creado el rol de IAM, puede crear el destino de los CloudWatch registros.

   a. Este paso no asocia una política de acceso a su destino y solo es el primer paso de los dos que completan la creación de un destino. Anota lo DestinationArnque se devuelve en la carga útil:

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. Una vez que se haya completado el paso 7a, en la cuenta del destinatario de los datos de registro, asocie una política de acceso con el destino. Esta política debe especificar los registros: la PutSubscriptionFilter acción y otorga permiso a la cuenta del remitente para acceder al destino.

      La política concede permiso a la AWS cuenta que envía los registros. Puede especificar solo esta cuenta en la política o, si la cuenta de remitente es miembro de una organización, la política puede especificar el ID de organización de la organización. De esta forma, puede crear una sola política para permitir que varias cuentas de una organización envíen registros a esta cuenta de destino.

      Utilice un editor de texto para crear un archivo denominado `~/AccessPolicy.json` con una de las siguientes declaraciones de política.

      Este primer ejemplo de política permite a todas las cuentas de la organización que tienen un ID de `o-1234567890` enviar registros a la cuenta de destinatario.

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

```
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

En el siguiente ejemplo, solo se permite que la cuenta del remitente de los datos de registro (111111111111) envíe los registros a la cuenta del destinatario de los datos de registro.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

c. Adjunte la política que creó en el paso anterior al destino.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Esta política de acceso permite a los usuarios de la AWS cuenta con el ID 1111 llamar al destino

con el ARN ***PutSubscriptionFilter****arn:aws:logs:region:55559999:destination:testDestination.* Se PutSubscriptionFilter rechazará cualquier intento de otro usuario de llamar a este destino.

Para validar los privilegios de un usuario con una política de acceso, consulte Uso del validador de políticas en la Guía del usuario de IAM.

Cuando hayas terminado, si los estás utilizando AWS Organizations para tus permisos multicuenta, sigue los pasos que se indican. Paso 2: (solo si se utiliza una organización) crear un rol de IAM Si está concediendo permisos directamente a la otra cuenta en lugar de utilizar Organizations, puede omitir ese paso y continuar con Paso 4: crear un filtro de suscripción.

Paso 2: (solo si se utiliza una organización) crear un rol de IAM

En la sección anterior, si ha creado el destino mediante una política de acceso que otorga permisos a la organización en la que está esa cuenta 111111111111, en lugar de conceder permisos directamente a la cuenta 111111111111, siga los pasos de esta sección. De lo contrario, puede ir directamente a Paso 4: crear un filtro de suscripción.

Los pasos de esta sección crean un rol de IAM, que CloudWatch puede asumir y validar si la cuenta del remitente tiene permiso para crear un filtro de suscripción para el destino del destinatario.

Realice los pasos de esta sección en la cuenta del remitente. El rol debe existir en la cuenta del remitente y usted especifica el ARN de este rol en el filtro de suscripción. En este ejemplo, la cuenta del remitente es 111111111111.

Para crear el rol de IAM necesario para las suscripciones de registros entre cuentas mediante AWS Organizations

1. Cree la siguiente política de confianza en un archivo `/TrustPolicyForCWLSubscriptionFilter.json`. Utilice un editor de texto para crear este archivo de política; no utilice la consola de IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Cree un rol de IAM que utilice la política. Anote el valor `Arn` que devuelve el comando, ya que lo necesitará más tarde en este procedimiento. En este ejemplo, usaremos `CWLtoSubscriptionFilterRole` para el nombre del rol que estamos creando.

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Crea una política de permisos para definir las acciones que CloudWatch Logs puede realizar en tu cuenta.

   a. En primer lugar, utilice un editor de texto para crear la siguiente política de permisos en un archivo denominado: `~/PermissionsForCWLSubscriptionFilter.json`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

   b. Ingrese el siguiente comando para asociar la política de permisos que acaba de crear con el rol que creó en el paso 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Cuando haya terminado, puede proceder a Paso 4: crear un filtro de suscripción.

Paso 3: agregar o validar los permisos de IAM para el destino entre cuentas

Según la lógica de evaluación de políticas AWS multicuenta, para acceder a cualquier recurso multicuenta (como una transmisión de Kinesis o Firehose utilizada como destino para un filtro de suscripciones), debe tener una política basada en la identidad en la cuenta remitente que

proporcione acceso explícito al recurso de destino multicuenta. Para obtener más información sobre la lógica de evaluación de políticas, consulte Lógica de evaluación de políticas entre cuentas.

Puede adjuntar la política basada en la identidad al rol de IAM o al usuario de IAM que utilice para crear el filtro de suscripción. Esta política debe estar presente en la cuenta de envío. Si utiliza la función de administrador para crear el filtro de suscripciones, puede omitir este paso y continuar con Paso 4: crear un filtro de suscripción.

Para agregar o validar los permisos de IAM necesarios para el uso entre cuentas

1. Introduzca el siguiente comando para comprobar qué rol o usuario de IAM se utiliza para ejecutar los comandos de registros de AWS .

   ```
   aws sts get-caller-identity
   ```

   El comando devuelve un resultado similar al siguiente:

   ```
   {
   "UserId": "User ID",
   "Account": "sending account id",
   "Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
   }
   ```

   Anote el valor representado por o. *RoleNameUserName*

2. Inicie sesión AWS Management Console en la cuenta remitente y busque las políticas adjuntas con el rol de IAM o el usuario de IAM que aparecen en el resultado del comando que ingresó en el paso 1.

3. Compruebe que las políticas adjntas a este rol o usuario brindan permisos explícitos para llamar a `logs:PutSubscriptionFilter` en el recurso de destino entre cuentas. En los siguientes ejemplos de política, se muestran los permisos recomendados.

   La siguiente política proporciona permisos para crear un filtro de suscripción en cualquier recurso de destino solo en una sola AWS cuenta, la cuenta: `123456789012`

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

```
            "Sid": "Allow subscription filters on any resource in one specific
 account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

La siguiente política proporciona permisos para crear un filtro de suscripción solo en un recurso de destino específico denominado `sampleDestination` AWS cuenta única`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
}
```

Paso 4: crear un filtro de suscripción

Después de crear un destino, la cuenta del destinatario de los datos de registro puede compartir el ARN de destino (arn:aws:logs:us-east-1:999999999999:destination:testDestination) con otras cuentas de AWS para que puedan enviar eventos de registro al mismo destino. A continuación, los usuarios de estas otras cuentas remitentes crean un filtro de suscripción en sus grupos de registros respectivos frente a este destino. El filtro de suscripción inicia de inmediato el flujo de datos de registro en tiempo real desde el grupo de registros elegido al destino especificado.

> **Note**
>
> Si concede permisos para el filtro de suscripción a toda una organización, tendrá que usar el ARN del rol de IAM en el que creó Paso 2: (solo si se utiliza una organización) crear un rol de IAM.

En el siguiente ejemplo, se crea un filtro de suscripción en una cuenta remitente. El filtro se asocia a un grupo de registros que contiene AWS CloudTrail eventos, de modo que cada actividad registrada con AWS las credenciales «Root» se envía al destino que creó anteriormente. Ese destino encapsula una transmisión llamada "»RecipientStream.

En el resto de los pasos de las siguientes secciones, se supone que ha seguido las instrucciones de la Guía del AWS CloudTrail usuario sobre cómo enviar CloudTrail eventos a los CloudWatch registros y ha creado un grupo de registros que contiene sus CloudTrail eventos. En estos pasos se supone que el nombre de este grupo de registros es `CloudTrail/logs`.

Al introducir el siguiente comando, asegúrese de haber iniciado sesión como usuario de IAM o de utilizar el rol de IAM para el que agregó la política, en Paso 3: agregar o validar los permisos de IAM para el destino entre cuentas.

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

El grupo de registros y el destino deben estar en la misma AWS región. Sin embargo, el destino puede apuntar a un AWS recurso, como una transmisión de Kinesis Data Streams, que se encuentre en una región diferente.

Validación del flujo de eventos de registro

Tras crear el filtro de suscripción, CloudWatch Logs reenvía todos los eventos de registro entrantes que coinciden con el patrón de filtrado a la transmisión que está encapsulada en la transmisión de destino denominada "». RecipientStream El propietario del destino puede comprobar que esto está ocurriendo utilizando el get-shard-iterator comando aws kinesis para capturar un fragmento de Kinesis Data Streams y utilizando el comando aws kinesis get-records para recuperar algunos registros de Kinesis Data Streams:

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Es posible que tenga que volver a ejecutar el comando get-records varias veces antes de que Kinesis Data Streams comience a devolver los datos.

Debería ver una respuesta con una matriz de registros de Kinesis Data Streams. El atributo de datos en un registro de Kinesis Data Streams está comprimido en formato gzip y cifrado en base64. Puede examinar los datos sin procesar desde la línea de comando utilizando el siguiente comando de Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Los datos descodificados y descomprimidos en base64 se formatean como JSON con la siguiente estructura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
```

```
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Los elementos fundamentales de esta estructura de datos son los siguientes:

owner

El AWS ID de cuenta de los datos de registro originarios.

logGroup

El nombre del grupo de registros de los datos de registro de origen.

logStream

El nombre del flujo de registros de los datos de registro de origen.

subscriptionFilters

La lista de nombres de filtros de suscripción que coincide con los datos de registro de origen.

messageType

Los mensajes de datos utilizan el tipo “DATA_MESSAGE”. A veces, CloudWatch los registros pueden emitir registros de Kinesis Data Streams del tipo «CONTROL_MESSAGE», principalmente para comprobar si se puede acceder al destino.

logEvents

Los datos de registro reales, representados como un conjunto de registros de eventos de registro. La propiedad ID es un identificador único de cada evento de registro.

Modificación de la suscripción al destino en tiempo de ejecución

Puede encontrar situaciones en las que tenga que añadir o eliminar la pertenencia de algunos usuarios de un destino de su propiedad. Puede utilizar el comando `put-destination-policy` en su destino con la nueva política de acceso. En el siguiente ejemplo, una cuenta 111111111111 añadida anteriormente deja de enviar más datos de registro y se habilita la cuenta 222222222222.

1. Busca la política que está asociada actualmente con el destino TestDestination y anota lo siguiente: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Actualice la política para reflejar que la cuenta 111111111111 está detenida y que la cuenta 222222222222 está habilitada. Coloca esta política en el archivo ~/ .json: NewAccessPolicy

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Llame PutDestinationPolicypara asociar la política definida en el NewAccessPolicyarchivo.json con el destino:

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Esto finalmente deshabilitará los eventos de registro del ID de cuenta 111111111111. Los eventos de registro del ID de cuenta 222222222222 empiezan a fluir al destino en cuanto el propietario de la cuenta 222222222222 crea un filtro de suscripción.

Actualización de una suscripción entre cuentas existente

Si actualmente tiene una suscripción de registros entre cuenta en la que la cuenta de destino concede permisos solo a cuentas de remitentes específicas y desea actualizar esta suscripción para que la cuenta de destino conceda acceso a todas las cuentas de una organización, siga los pasos de esta sección.

Temas

- Paso 1: actualizar los filtros de suscripción
- Paso 2: actualizar la política de acceso de destino existente

Paso 1: actualizar los filtros de suscripción

> **Note**
>
> Este paso solo es necesario para las suscripciones multicuenta de los registros creados por los servicios enumerados en Habilitar el registro desde AWS los servicios. Si no está trabajando con registros creados por uno de estos grupos de registros, puede ir directo a Paso 2: actualizar la política de acceso de destino existente.

En algunos casos, debe actualizar los filtros de suscripción en todas las cuentas de remitente que envían registros a la cuenta de destino. La actualización añade una función de IAM, que permite CloudWatch asumir y validar que la cuenta del remitente tiene permiso para enviar los registros a la cuenta del destinatario.

Siga los pasos de esta sección para cada cuenta de remitente que desee actualizar para utilizar el ID de organización para los permisos de suscripción entre cuentas.

En los ejemplos de esta sección, dos cuentas, 111111111111 y 222222222222, ya cuentan con filtros de suscripción creados para enviar registros a la cuenta 999999999999. Los valores de filtro de suscripción existentes son los siguientes:

```
## Existing Subscription Filter parameter values
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Si tiene que encontrar los valores de los parámetros de filtro de suscripción actuales, ingrese el siguiente comando.

```
aws logs describe-subscription-filters
    \ --log-group-name "my-log-group-name"
```

Para actualizar un filtro de suscripción para empezar a utilizar ID de organización para permisos de registro entre cuentas

1. Cree la siguiente política de confianza en un archivo ~/TrustPolicyForCWL.json. Utilice un editor de texto para crear este archivo de política; no utilice la consola de IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Cree un rol de IAM que utilice la política. Anote el valor `Arn` del valor `Arn` que devuelve el comando, ya que lo necesitará más tarde en este procedimiento. En este ejemplo, usaremos `CWLtoSubscriptionFilterRole` para el nombre del rol que estamos creando.

```
aws iam create-role
    \ --role-name CWLtoSubscriptionFilterRole
    \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

3. Crea una política de permisos para definir las acciones que CloudWatch Logs puede realizar en tu cuenta.

   a. En primer lugar, utilice un editor de texto para crear la siguiente política de permisos en un archivo denominado: `/PermissionsForCWLSubscriptionFilter.json`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

   b. Ingrese el siguiente comando para asociar la política de permisos que acaba de crear con el rol que creó en el paso 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

4. Ingrese el siguiente comando para actualizar el filtro de suscripción.

```
aws logs put-subscription-filter
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn
 "arn:aws:logs:region:999999999999:destination:testDestination"
    \ --role-arn "arn:aws:iam::111111111111:role/CWLtoSubscriptionFilterRole"
```

Paso 2: actualizar la política de acceso de destino existente

Después de actualizar los filtros de suscripción en todas las cuentas de remitente, puede actualizar la política de acceso de destino en la cuenta de destinatario.

En los ejemplos siguientes, la cuenta de destinatario es 999999999999 y el destino se llama `testDestination`.

La actualización permite que todas las cuentas que forman parte de la organización con ID `o-1234567890` envíen registros a la cuenta de destinatario. Solo las cuentas que tienen filtros de suscripción creados enviarán registros a la cuenta del destinatario.

Para actualizar la política de acceso de destino en la cuenta de destinatario a fin de empezar a utilizar un ID de organización para obtener permisos

1. En la cuenta del destinatario, utilice un editor de texto para crear un archivo `~/AccessPolicy.json` con el siguiente contenido.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
```

```
                }
            }
        }
    ]
}
```

2. Ingrese el siguiente comando para adjuntar la política que acaba de crear al destino existente. Para actualizar un destino para utilizar una política de acceso con un ID de organización en lugar de una política de acceso que enumera los ID de cuenta AWS específicos, incluya el parámetro `force`.

> **Warning**
>
> Si está trabajando con registros enviados por uno de los AWS servicios incluidos en la listaHabilitar el registro desde AWS los servicios, antes de realizar este paso, debe haber actualizado los filtros de suscripción en todas las cuentas de remitentes, tal y como se explica enPaso 1: actualizar los filtros de suscripción.

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Intercambio de datos de registro entre cuentas y regiones mediante Firehose

Para compartir los datos de registro entre cuentas, es necesario establecer un emisor y receptor de datos de registro:

- Remitente de los datos de registro: obtiene la información de destino del destinatario e informa a CloudWatch Logs de que está listo para enviar sus eventos de registro al destino especificado. En los procedimientos descritos en el resto de esta sección, se muestra al remitente de los datos de registro con un número de AWS cuenta ficticio de 1111.
- Destinatario de los datos de registro: configura un destino que encapsula una transmisión de Kinesis Data Streams y permite a CloudWatch Logs saber que el destinatario quiere recibir los datos de registro. A continuación, el destinatario comparte la información sobre este destino con el

remitente. En los procedimientos descritos en el resto de esta sección, se muestra al destinatario de los datos de registro con un número de AWS cuenta ficticio, 222222222222.

El ejemplo de esta sección utiliza una transmisión de entrega de Firehose con almacenamiento en Amazon S3. También puedes configurar las transmisiones de entrega de Firehose con diferentes ajustes. Para obtener más información, consulte Creación de un flujo de entrega de Firehose.

> **Note**
>
> El grupo de registros y el destino deben estar en la misma AWS región. Sin embargo, el recurso de AWS al que apunta el destino puede estar ubicado en una región diferente.

> **Note**
>
> Se admite el filtro de suscripción Firehose para una misma cuenta y el flujo de entrega entre regiones.

Temas

Paso 1: Crear un flujo de entrega de Firehose

> **Important**
>
> Antes de completar los siguientes pasos, debes usar una política de acceso para que Firehose pueda acceder a tu bucket de Amazon S3. Para obtener más información, consulte Controlling Access en la Guía para desarrolladores de Amazon Data Firehose.
> Todos los pasos en esta sección (Paso 1) deben realizarse en la cuenta del destinatario de los datos de registro.

En los ejemplos siguientes, se utiliza Este de EE. UU. (Norte de Virginia). Reemplace esta región por la región correcta para su implementación.

Para crear un flujo de entrega de Firehose para usarlo como destino

1. Cree un bucket de Amazon S3:

```
aws s3api create-bucket --bucket firehose-test-bucket1 --create-bucket-
configuration LocationConstraint=us-east-1
```

2. Crea el rol de IAM que otorga permiso a Firehose para colocar datos en el depósito.

   a. En primer lugar, utilice un editor de texto para crear una política de confianza en un archivo `~/TrustPolicyForFirehose.json`.

```
{ "Statement": { "Effect": "Allow", "Principal": { "Service":
 "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
 { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
```

   b. Cree el rol de IAM y especifique el archivo de política de confianza que acaba de crear.

```
aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json
```

   c. El resultado de este comando debería ser similar a lo siguiente. Haga una nota del nombre del rol y del ARN del rol.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
```

```
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. Crea una política de permisos para definir las acciones que Firehose puede realizar en tu cuenta.

   a. En primer lugar, utilice un editor de texto para crear la siguiente política de permisos en un archivo denominado: `~/PermissionsForFirehose.json`. Según el caso de uso, es posible que tenga que agregar más permisos a este archivo.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::firehose-test-bucket1",
            "arn:aws:s3:::firehose-test-bucket1/*"
        ]
    }]
}
```

   b. Ingrese el siguiente comando para asociar la política de permisos que acaba de crear con el rol de IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Ingresa el siguiente comando para crear el flujo de entrega de Firehose. Sustituya *my-role-arn* y *my-bucket-arn* por los valores correctos para su implementación.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::firehose-test-bucket1"}'
```

El resultado debería tener un aspecto similar al siguiente:

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Paso 2: creación de un destino

> **Important**
>
> Todos los pasos de este procedimiento deben realizarse en la cuenta del destinatario de los datos de registro.

Cuando se crea el destino, CloudWatch Logs envía un mensaje de prueba al destino en nombre de la cuenta del destinatario. Cuando el filtro de suscripciones se active más adelante, CloudWatch Logs envía los eventos de registro al destino en nombre de la cuenta de origen.

Para crear un destino

1. Espera a que se active la transmisión de Firehose en Paso 1: Crear un flujo de entrega de Firehose la que creaste. Puede usar el siguiente comando para comprobar laStreamDescription. StreamStatuspropiedad.

   ```
   aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
   ```

   Además, tome nota de la DeliveryStreamDescription. DeliveryStreamValor ARN, ya que tendrá que usarlo en un paso posterior. Resultado de ejemplo de este comando:

   ```
   {
   ```

```
    "DeliveryStreamDescription": {
        "DeliveryStreamName": "my-delivery-stream",
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:222222222222:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamEncryptionConfiguration": {
            "Status": "DISABLED"
        },
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
```

```
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

El flujo de entrega puede tardar un minuto o dos en mostrarse en el estado activo.

2. Cuando la transmisión de entrega esté activa, crea la función de IAM que concederá a CloudWatch Logs el permiso para colocar datos en tu transmisión de Firehose. En primer lugar, tendrás que crear una política de confianza en un archivo TrustPolicyFor~/ CWL.json. Utilice un editor de texto para crear esta política. Para obtener más información sobre CloudWatch los puntos de enlace de Logs, consulte los puntos de enlace y las cuotas de Amazon CloudWatch Logs.

   Esta política incluye una clave de contexto de condición global `aws:SourceArn` que especifica la `sourceAccountId` para ayudar a prevenir el problema de seguridad de suplente confuso. Si aún no conoce el ID de cuenta de origen en la primera llamada, le recomendamos que coloque el ARN de destino en el campo ARN de origen. En las llamadas posteriores, debe configurar el ARN de origen para que sea el ARN de origen real que recopiló desde la primera llamada. Para obtener más información, consulte Prevención del suplente confuso.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.region.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
     }
}
```

3. Utilice aws iam create-role para crear el rol de IAM y especifique el archivo de política de confianza que acaba de crear.

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

A continuación, se muestra un ejemplo de la salida. Anote el valor de `Role.Arn` devuelto, ya que lo necesitará en un paso posterior.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2021-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Cree una política de permisos para definir qué acciones puede realizar CloudWatch Logs en su cuenta. Primero, usa un editor de texto para crear una política de permisos en un archivo ~/ PermissionsFor CWL.json:

```
{
```

```
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. Asocie la política de permisos con el rol mediante el siguiente comando:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Una vez que la transmisión de entrega de Firehose esté en estado activo y hayas creado la función de IAM, puedes crear el destino de los CloudWatch registros.

   a. Este paso no asociará una política de acceso a su destino y solo es el primer paso de los dos que completan la creación de un destino. Anote el ARN del nuevo destino que se devuelve en la carga, porque lo utilizará como `destination.arn` en un paso posterior.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

   b. Después de completar el paso previo, en la cuenta del destinatario de los datos de registro (222222222222), asocie una política de acceso con el destino.

Esta política permite que la cuenta del remitente de los datos de registro (111111111111) tenga acceso al destino justo en la cuenta del destinatario de los datos de registro (222222222222). Puedes usar un editor de texto para colocar esta política en el archivo ~/ .json: AccessPolicy

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. Esto crea una política que define quién tiene acceso de escritura al destino. Esta política debe especificar los registros: PutSubscriptionFilter acción para acceder al destino. Los usuarios con varias cuentas utilizarán la PutSubscriptionFilteracción para enviar los eventos del registro al destino:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Paso 3: agregar o validar los permisos de IAM para el destino entre cuentas

Según la lógica de evaluación de políticas AWS multicuenta, para acceder a cualquier recurso multicuenta (como una transmisión de Kinesis o Firehose utilizada como destino para un filtro de suscripciones), debe tener una política basada en la identidad en la cuenta remitente que proporcione acceso explícito al recurso de destino multicuenta. Para obtener más información sobre la lógica de evaluación de políticas, consulte Lógica de evaluación de políticas entre cuentas.

Puede adjuntar la política basada en la identidad al rol de IAM o al usuario de IAM que utilice para crear el filtro de suscripción. Esta política debe estar presente en la cuenta de envío. Si utiliza la función de administrador para crear el filtro de suscripciones, puede omitir este paso y continuar con Paso 4: crear un filtro de suscripción.

Para agregar o validar los permisos de IAM necesarios para el uso entre cuentas

1. Introduzca el siguiente comando para comprobar qué rol o usuario de IAM se utiliza para ejecutar los comandos de registros de AWS .

   ```
   aws sts get-caller-identity
   ```

   El comando devuelve un resultado similar al siguiente:

   ```
   {
   "UserId": "User ID",
   "Account": "sending account id",
   "Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
   }
   ```

   Anote el valor representado por o. *RoleNameUserName*

2. Inicie sesión AWS Management Console en la cuenta remitente y busque las políticas adjuntas con el rol de IAM o el usuario de IAM que aparecen en el resultado del comando que ingresó en el paso 1.
3. Compruebe que las políticas adjntas a este rol o usuario brindan permisos explícitos para llamar a `logs:PutSubscriptionFilter` en el recurso de destino entre cuentas. En los siguientes ejemplos de política, se muestran los permisos recomendados.

   La siguiente política proporciona permisos para crear un filtro de suscripción en cualquier recurso de destino solo en una sola AWS cuenta, la cuenta: `123456789012`

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "Allow subscription filters on any resource in one specific
    account",
               "Effect": "Allow",
               "Action": "logs:PutSubscriptionFilter",
   ```

```
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

La siguiente política proporciona permisos para crear un filtro de suscripción solo en un recurso de destino específico denominado `sampleDestination` AWS cuenta única`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
}
```

Paso 4: crear un filtro de suscripción

Cambie a la cuenta de envío, que es 111111111111 en este ejemplo. Ahora creará el filtro de suscripción en la cuenta de envío. En este ejemplo, el filtro está asociado a un grupo de registros que contiene AWS CloudTrail eventos, de modo que cada actividad registrada con AWS las credenciales «Root» se envía al destino que creaste anteriormente. Para obtener más información sobre cómo enviar AWS CloudTrail eventos a los CloudWatch registros, consulte Enviar CloudTrail eventos a los CloudWatch registros en la Guía del AWS CloudTrail usuario.

Al introducir el siguiente comando, asegúrese de haber iniciado sesión como usuario de IAM o de utilizar el rol de IAM para el que agregó la política, en Paso 3: agregar o validar los permisos de IAM para el destino entre cuentas.

```
aws logs put-subscription-filter \
    --log-group-name "aws-cloudtrail-logs-111111111111-300a971e" \
    --filter-name "firehose_test" \
    --filter-pattern "{$.userIdentity.type = AssumedRole}" \
    --destination-arn "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
```

El grupo de registros y el destino deben estar en la misma AWS región. Sin embargo, el destino puede apuntar a un AWS recurso, como un arroyo Firehose, que se encuentra en una región diferente.

Validación del flujo de eventos de registro

Tras crear el filtro de suscripción, CloudWatch Logs reenvía todos los eventos de registro entrantes que coinciden con el patrón de filtrado al flujo de entrega de Firehose. Los datos comienzan a aparecer en tu bucket de Amazon S3 en función del intervalo de tiempo establecido en la transmisión de entrega de Firehose. Una vez que haya transcurrido el tiempo suficiente, puede verificar los datos comprobando su bucket de Amazon S3. Escriba el siguiente comando para comprobar el bucket:

```
aws s3api list-objects --bucket 'firehose-test-bucket1'
```

El resultado de ese comando será similar a lo siguiente:

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2021-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

Puede recuperar un objeto específico del bucket al introducir el siguiente comando. Reemplace el valor de `key` con el valor que encontró en el comando anterior.

```
aws s3api get-object --bucket 'firehose-test-bucket1' --key '2021/02/02/08/my-delivery-stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

Los datos en el objeto de Amazon S3 se comprimen con el formato gzip. Puede examinar los datos sin procesar desde la línea de comando mediante uno de los siguientes comandos:

Linux:

```
zcat testfile.gz
```

macOS:

```
zcat <testfile.gz
```

Modificación de la suscripción al destino en tiempo de ejecución

Puede encontrar situaciones en las que tenga que agregar o eliminar remitentes de registros de un destino de su propiedad. Puedes usar la PutDestinationPolicyacción en tu destino con una nueva política de acceso. En el siguiente ejemplo, una cuenta 111111111111 agregada anteriormente deja de enviar datos de registro y se habilita la cuenta 333333333333.

1. Busca la política que está asociada actualmente con el destino TestDestination y anota lo siguiente: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testFirehoseDestination"

{
    "destinations": [
        {
            "destinationName": "testFirehoseDestination",
            "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
```

```
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. Actualice la política para reflejar que la cuenta 111111111111 está detenida y que la cuenta 333333333333 está habilitada. Coloca esta política en el archivo ~/ .json: NewAccessPolicy

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. Use el siguiente comando para asociar la política definida en el NewAccessPolicyarchivo.json con el destino:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

Esto finalmente deshabilita los eventos de registro del ID de cuenta 111111111111. Los eventos de registro del ID de cuenta 333333333333 empiezan a fluir al destino en cuanto el propietario de la cuenta 333333333333 crea un filtro de suscripción.

Suscripciones a nivel de cuenta multirregional mediante Kinesis Data Streams

Al crear una suscripción entre cuentas, puede especificar una única cuenta o una organización para que sea el remitente. Si especifica una organización, este procedimiento permite que todas las cuentas de la organización envíen registros a la cuenta de receptor.

Para compartir los datos de registro entre cuentas, es necesario establecer un emisor y receptor de datos de registro:

- Remitente de datos de registro: obtiene la información de destino del destinatario e informa a CloudWatch Logs de que está listo para enviar sus eventos de registro al destino especificado. En los procedimientos descritos en el resto de esta sección, se muestra al remitente de los datos de registro con un número de AWS cuenta ficticio de 1111.

  Si va a tener varias cuentas de una organización que envíen registros a una cuenta de destinatario, puede crear una política que otorgue a todas las cuentas de la organización el permiso para enviar registros a la cuenta de destinatario. Aún tiene que configurar filtros de suscripción independientes para cada cuenta de remitente.
- Destinatario de los datos de registro: configura un destino que encapsula una transmisión de Kinesis Data Streams y permite a CloudWatch Logs saber que el destinatario quiere recibir los datos de registro. A continuación, el destinatario comparte la información sobre este destino con el remitente. En los procedimientos que se describen en el resto de esta sección, se muestra al destinatario de los datos de registro con un número de AWS cuenta ficticio, el número de cuenta 999.

Para empezar a recibir eventos de registro de usuarios con varias cuentas, el destinatario de los datos de registro crea primero un destino de CloudWatch registros. Cada destino consta de los siguientes elementos fundamentales:

Nombre de destino

El nombre del destino que desea crear.

ARN de destino

El nombre del recurso de Amazon (ARN) del AWS recurso que quieres usar como destino del feed de suscripción.

ARN del rol

Un rol AWS Identity and Access Management (IAM) que otorga a CloudWatch Logs los permisos necesarios para colocar los datos en la transmisión elegida.

Política de acceso

Un documento de política de IAM (en formato JSON, escrito con la gramática de política de IAM) que rige el conjunto de los usuarios a los que se les permite escribir en su destino.

> **Note**
>
> El grupo de registros y el destino deben estar en la misma AWS región. Sin embargo, el recurso de AWS al que apunta el destino puede estar ubicado en una región diferente. En los ejemplos de las secciones siguientes, todos los recursos específicos de una región se crean en EE. UU. Este (Norte de Virginia).

Temas

Configuración de una nueva suscripción entre cuentas

Siga los pasos de estas secciones para configurar una nueva suscripción de registro entre cuentas.

Temas

Paso 1: crear un destino

> **Important**
>
> Todos los pasos de este procedimiento deben realizarse en la cuenta del destinatario de los datos de registro.

En este ejemplo, la cuenta receptora de los datos de registro tiene un identificador de AWS cuenta de 10000 9999, mientras que el identificador de la AWS cuenta del remitente de los datos de registro es 1111.

En este ejemplo, se crea un destino mediante una transmisión de Kinesis Data Streams RecipientStream llamada y una función que CloudWatch permite a Logs escribir datos en ella.

Cuando se crea el destino, CloudWatch Logs envía un mensaje de prueba al destino en nombre de la cuenta del destinatario. Cuando el filtro de suscripciones se active más adelante, CloudWatch Logs envía los eventos de registro al destino en nombre de la cuenta de origen.

Para crear un destino

1. En la cuenta de destinatario, cree un flujo de destino en Kinesis Data Streams. En el símbolo del sistema, escriba:

   ```
   aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
   ```

2. Espere hasta que el flujo de se active. Puede utilizar el comando aws kinesis describe-stream para comprobar el. StreamDescription StreamStatuspropiedad. Además, tome nota del valor StreamDescription.streamArn porque lo pasará a CloudWatch Logs más adelante:

   ```
   aws kinesis describe-stream --stream-name "RecipientStream"
   {
     "StreamDescription": {
       "StreamStatus": "ACTIVE",
       "StreamName": "RecipientStream",
       "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
       "Shards": [
         {
           "ShardId": "shardId-000000000000",
           "HashKeyRange": {
             "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
   ```

```
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "495511352186888184566795038319814587845913527021 8EXAMPLE"
        }
      }
    ]
  }
}
```

El flujo puede tardar un minuto o dos en mostrarse en el estado activo.

3. Crea el rol de IAM que otorga a CloudWatch Logs el permiso para colocar datos en tu transmisión. En primer lugar, tendrás que crear una política de confianza en un archivo ~/ TrustPolicyFor CWL.json. Utilice un editor de texto para crear este archivo de política; no utilice la consola de IAM.

   Esta política incluye una clave de contexto de condición global `aws:SourceArn` que especifica la `sourceAccountId` para ayudar a prevenir el problema de seguridad de suplente confuso. Si aún no conoce el ID de cuenta de origen en la primera llamada, le recomendamos que coloque el ARN de destino en el campo ARN de origen. En las llamadas posteriores, debe configurar el ARN de origen para que sea el ARN de origen real que recopiló desde la primera llamada. Para obtener más información, consulte Prevención del suplente confuso.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "Action": "sts:AssumeRole"
    }
```

```
}
```

4. Ejecute el comando aws iam create-role para crear el rol de IAM y especifique el archivo de política de confianza. Toma nota del valor Role.Arn devuelto porque también se pasará a Logs más adelante: CloudWatch

```
aws iam create-role \
--role-name CWLtoKinesisRole \
--assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Crea una política de permisos para definir qué acciones puede realizar CloudWatch Logs en tu cuenta. Primero, usa un editor de texto para crear una política de permisos en un archivo ~/ PermissionsFor CWL.json:

```
{
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Asocie la política de permisos al rol mediante el comando aws iam: put-role-policy

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Una vez que la transmisión esté en estado activo y haya creado el rol de IAM, puede crear el destino de los CloudWatch registros.

   a. Este paso no asocia una política de acceso a su destino y solo es el primer paso de los dos que completan la creación de un destino. Anota lo DestinationArnque se devuelve en la carga útil:

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. Una vez que se haya completado el paso 7a, en la cuenta del destinatario de los datos de registro, asocie una política de acceso con el destino. Esta política debe especificar los registros: la PutSubscriptionFilter acción y otorga permiso a la cuenta del remitente para acceder al destino.

      La política concede permiso a la AWS cuenta que envía los registros. Puede especificar solo esta cuenta en la política o, si la cuenta de remitente es miembro de una organización,

la política puede especificar el ID de organización de la organización. De esta forma, puede crear una sola política para permitir que varias cuentas de una organización envíen registros a esta cuenta de destino.

Utilice un editor de texto para crear un archivo denominado `~/AccessPolicy.json` con una de las siguientes declaraciones de política.

Este primer ejemplo de política permite a todas las cuentas de la organización que tienen un ID de `o-1234567890` enviar registros a la cuenta de destinatario.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

En el siguiente ejemplo, solo se permite que la cuenta del remitente de los datos de registro (111111111111) envíe los registros a la cuenta del destinatario de los datos de registro.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
```

```
      "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

c. Adjunte la política que creó en el paso anterior al destino.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Esta política de acceso permite a los usuarios de la AWS cuenta con el ID 1111 llamar al destino con el ARN ***PutSubscriptionFilter****arn:aws:logs:region:55559999:destination:testDestination.* Se PutSubscriptionFilter rechazará cualquier intento de otro usuario de llamar a este destino.

Para validar los privilegios de un usuario con una política de acceso, consulte Uso del validador de políticas en la Guía del usuario de IAM.

Cuando hayas terminado, si los estás utilizando AWS Organizations para tus permisos multicuenta, sigue los pasos que se indican. Paso 2: (solo si se utiliza una organización) crear un rol de IAM Si está concediendo permisos directamente a la otra cuenta en lugar de utilizar Organizations, puede omitir ese paso y continuar con Paso 3: Crea una política de filtrado de suscripciones a nivel de cuenta.

Paso 2: (solo si se utiliza una organización) crear un rol de IAM

En la sección anterior, si ha creado el destino mediante una política de acceso que otorga permisos a la organización en la que está esa cuenta 111111111111, en lugar de conceder permisos directamente a la cuenta 111111111111, siga los pasos de esta sección. De lo contrario, puede ir directamente a Paso 3: Crea una política de filtrado de suscripciones a nivel de cuenta.

Los pasos de esta sección crean un rol de IAM, que CloudWatch puede asumir y validar si la cuenta del remitente tiene permiso para crear un filtro de suscripción para el destino del destinatario.

Realice los pasos de esta sección en la cuenta del remitente. El rol debe existir en la cuenta del remitente y usted especifica el ARN de este rol en el filtro de suscripción. En este ejemplo, la cuenta del remitente es 111111111111.

Para crear el rol de IAM necesario para las suscripciones de registros entre cuentas mediante AWS Organizations

1. Cree la siguiente política de confianza en un archivo `/TrustPolicyForCWLSubscriptionFilter.json`. Utilice un editor de texto para crear este archivo de política; no utilice la consola de IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Cree un rol de IAM que utilice la política. Anote el valor `Arn` que devuelve el comando, ya que lo necesitará más tarde en este procedimiento. En este ejemplo, usaremos `CWLtoSubscriptionFilterRole` para el nombre del rol que estamos creando.

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Crea una política de permisos para definir las acciones que CloudWatch Logs puede realizar en tu cuenta.

   a. En primer lugar, utilice un editor de texto para crear la siguiente política de permisos en un archivo denominado: `~/PermissionsForCWLSubscriptionFilter.json`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Ingrese el siguiente comando para asociar la política de permisos que acaba de crear con el rol que creó en el paso 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Cuando haya terminado, puede proceder a Paso 3: Crea una política de filtrado de suscripciones a nivel de cuenta.

Paso 3: Crea una política de filtrado de suscripciones a nivel de cuenta

Después de crear un destino, la cuenta del destinatario de los datos de registro puede compartir el ARN de destino (arn:aws:logs:us-east-1:999999999999:destination:testDestination) con otras cuentas de AWS para que puedan enviar eventos de registro al mismo destino. A continuación, los usuarios de estas otras cuentas remitentes crean un filtro de suscripción en sus grupos de registros respectivos frente a este destino. El filtro de suscripción inicia de inmediato el flujo de datos de registro en tiempo real desde el grupo de registros elegido al destino especificado.

> **Note**
>
> Si concede permisos para el filtro de suscripción a toda una organización, tendrá que usar el ARN del rol de IAM en el que creó Paso 2: (solo si se utiliza una organización) crear un rol de IAM.

En el siguiente ejemplo, se crea una política de filtrado de suscripciones a nivel de cuenta en una cuenta remitente. El filtro se asocia a la cuenta del remitente 111111111111 para que todos los eventos de registro que coincidan con el filtro y los criterios de selección se entreguen al destino que creó anteriormente. Ese destino encapsula una transmisión llamada "». RecipientStream

El `selection-criteria` campo es opcional, pero es importante para excluir de un filtro de suscripción los grupos de registros que pueden provocar una recursión infinita de registros. Para obtener más información sobre este problema y determinar qué grupos de registros se van a excluir, consultePrevención de la recursión de registros. Actualmente, NOT IN es el único operador compatible`selection-criteria`.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Los grupos de registro de la cuenta del remitente y el destino deben estar en la misma AWS región. Sin embargo, el destino puede apuntar a un AWS recurso, como una transmisión de Kinesis Data Streams, que se encuentre en una región diferente.

Validación del flujo de eventos de registro

Tras crear la política de filtrado de suscripciones a nivel de cuenta, CloudWatch Logs reenvía todos los eventos de registro entrantes que coincidan con el patrón de filtrado y los criterios de selección al flujo encapsulado en el flujo de destino denominado «». RecipientStream El propietario del destino puede comprobar que esto está ocurriendo utilizando el get-shard-iterator comando aws kinesis para capturar un fragmento de Kinesis Data Streams y utilizando el comando aws kinesis get-records para recuperar algunos registros de Kinesis Data Streams:

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
```

```
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Puede que tenga que volver a ejecutar el `get-records` comando varias veces antes de que Kinesis Data Streams comience a devolver datos.

Debería ver una respuesta con una matriz de registros de Kinesis Data Streams. El atributo de datos en un registro de Kinesis Data Streams está comprimido en formato gzip y cifrado en base64. Puede examinar los datos sin procesar desde la línea de comando utilizando el siguiente comando de Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Los datos descodificados y descomprimidos en base64 se formatean como JSON con la siguiente estructura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
```

```
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Los elementos clave de la estructura de datos son los siguientes:

messageType

Los mensajes de datos utilizarán el tipo "DATA_MESSAGE". A veces, CloudWatch los registros pueden emitir registros de Kinesis Data Streams del tipo «CONTROL_MESSAGE», principalmente para comprobar si se puede acceder al destino.

owner

El ID de AWS cuenta de los datos de registro originarios.

logGroup

El nombre del grupo de registros de los datos de registro de origen.

logStream

El nombre del flujo de registros de los datos de registro de origen.

subscriptionFilters

La lista de nombres de filtros de suscripción que coincide con los datos de registro de origen.

logEvents

Los datos de registro reales, representados como un conjunto de registros de eventos de registro. La propiedad "id" es un identificador único de cada evento de registro.

Nivel de política

El nivel en el que se aplicó la política. «ACCOUNT_LEVEL_POLICY» es la política de filtrado de suscripciones a nivel `policyLevel` de cuenta.

Modificación de la suscripción al destino en tiempo de ejecución

Puede encontrar situaciones en las que tenga que añadir o eliminar la pertenencia de algunos usuarios de un destino de su propiedad. Puede utilizar el comando `put-destination-policy`

en su destino con la nueva política de acceso. En el siguiente ejemplo, una cuenta 111111111111 añadida anteriormente deja de enviar más datos de registro y se habilita la cuenta 222222222222.

1. Busca la política que está asociada actualmente con el destino TestDestination y anota lo siguiente: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Actualice la política para reflejar que la cuenta 111111111111 está detenida y que la cuenta 222222222222 está habilitada. Coloca esta política en el archivo ~/ .json: NewAccessPolicy

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Llame PutDestinationPolicypara asociar la política definida en el NewAccessPolicyarchivo.json con el destino:

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Esto finalmente deshabilitará los eventos de registro del ID de cuenta 111111111111. Los eventos de registro del ID de cuenta 222222222222 empiezan a fluir al destino en cuanto el propietario de la cuenta 222222222222 crea un filtro de suscripción.

Actualización de una suscripción entre cuentas existente

Si actualmente tiene una suscripción de registros entre cuenta en la que la cuenta de destino concede permisos solo a cuentas de remitentes específicas y desea actualizar esta suscripción para que la cuenta de destino conceda acceso a todas las cuentas de una organización, siga los pasos de esta sección.

Temas

- Paso 1: actualizar los filtros de suscripción
- Paso 2: actualizar la política de acceso de destino existente

Paso 1: actualizar los filtros de suscripción

> **Note**
>
> Este paso solo es necesario para las suscripciones multicuenta de los registros creados por los servicios enumerados en Habilitar el registro desde AWS los servicios. Si no está trabajando con registros creados por uno de estos grupos de registros, puede ir directo a Paso 2: actualizar la política de acceso de destino existente.

En algunos casos, debe actualizar los filtros de suscripción en todas las cuentas de remitente que envían registros a la cuenta de destino. La actualización añade una función de IAM, que permite CloudWatch asumir y validar que la cuenta del remitente tiene permiso para enviar los registros a la cuenta del destinatario.

Siga los pasos de esta sección para cada cuenta de remitente que desee actualizar para utilizar el ID de organización para los permisos de suscripción entre cuentas.

En los ejemplos de esta sección, dos cuentas, 111111111111 y 222222222222, ya cuentan con filtros de suscripción creados para enviar registros a la cuenta 999999999999. Los valores de filtro de suscripción existentes son los siguientes:

```
## Existing Subscription Filter parameter values
{
    "DestinationArn": "arn:aws:logs:region:999999999999:destination:testDestination",
    "FilterPattern": "{$.userIdentity.type = Root}",
    "Distribution": "Random"
}
```

Si tiene que encontrar los valores de los parámetros de filtro de suscripción actuales, ingrese el siguiente comando.

```
aws logs describe-account-policies \
--policy-type "SUBSCRIPTION_FILTER_POLICY" \
--policy-name "CrossAccountStreamsExamplePolicy"
```

Para actualizar un filtro de suscripción para empezar a utilizar ID de organización para permisos de registro entre cuentas

1. Cree la siguiente política de confianza en un archivo ~/TrustPolicyForCWL.json. Utilice un editor de texto para crear este archivo de política; no utilice la consola de IAM.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. Cree un rol de IAM que utilice la política. Anote el valor Arn del valor Arn que devuelve el comando, ya que lo necesitará más tarde en este procedimiento. En este ejemplo, usaremos CWLtoSubscriptionFilterRole para el nombre del rol que estamos creando.

   ```
   aws iam create-role
   ```

```
    \ --role-name CWLtoSubscriptionFilterRole
    \ --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

3. Crea una política de permisos para definir las acciones que CloudWatch Logs puede realizar en tu cuenta.

   a. En primer lugar, utilice un editor de texto para crear la siguiente política de permisos en un archivo denominado: `/PermissionsForCWLSubscriptionFilter.json`.

   ```
   {
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "logs:PutLogEvents",
               "Resource": "arn:aws:logs:region:111111111111:log-
   group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
           }
       ]
   }
   ```

   b. Ingrese el siguiente comando para asociar la política de permisos que acaba de crear con el rol que creó en el paso 2.

   ```
   aws iam put-role-policy
       --role-name CWLtoSubscriptionFilterRole
       --policy-name Permissions-Policy-For-CWL-Subscription-filter
       --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
   ```

4. Ingresa el siguiente comando para actualizar la política de filtros de suscripciones.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "{$.userIdentity.type = Root}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Paso 2: actualizar la política de acceso de destino existente

Después de actualizar los filtros de suscripción en todas las cuentas de remitente, puede actualizar la política de acceso de destino en la cuenta de destinatario.

En los ejemplos siguientes, la cuenta de destinatario es 999999999999 y el destino se llama `testDestination`.

La actualización permite que todas las cuentas que forman parte de la organización con ID `o-1234567890` envíen registros a la cuenta de destinatario. Solo las cuentas que tienen filtros de suscripción creados enviarán registros a la cuenta del destinatario.

Para actualizar la política de acceso de destino en la cuenta de destinatario a fin de empezar a utilizar un ID de organización para obtener permisos

1. En la cuenta del destinatario, utilice un editor de texto para crear un archivo `~/AccessPolicy.json` con el siguiente contenido.

```
{
"Version" : "2012-10-17",
    "Statement" : [
        {
"Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
"StringEquals" : {
"aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

2. Ingrese el siguiente comando para adjuntar la política que acaba de crear al destino existente. Para actualizar un destino para utilizar una política de acceso con un ID de organización en lugar de una política de acceso que enumera los ID de cuenta AWS específicos, incluya el parámetro `force`.

> **Warning**
>
> Si está trabajando con registros enviados por uno de los AWS servicios incluidos en la listaHabilitar el registro desde AWS los servicios, antes de realizar este paso, debe haber actualizado los filtros de suscripción en todas las cuentas de remitentes, tal y como se explica enPaso 1: actualizar los filtros de suscripción.

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Suscripciones a nivel de cuenta multirregional mediante Firehose

Para compartir los datos de registro entre cuentas, es necesario establecer un emisor y receptor de datos de registro:

- Remitente de los datos de registro: obtiene la información de destino del destinatario e informa a CloudWatch Logs de que está listo para enviar sus eventos de registro al destino especificado. En los procedimientos descritos en el resto de esta sección, se muestra al remitente de los datos de registro con un número de AWS cuenta ficticio de 1111.
- Destinatario de los datos de registro: configura un destino que encapsula una transmisión de Kinesis Data Streams y permite a CloudWatch Logs saber que el destinatario quiere recibir los datos de registro. A continuación, el destinatario comparte la información sobre este destino con el remitente. En los procedimientos descritos en el resto de esta sección, se muestra al destinatario de los datos de registro con un número de AWS cuenta ficticio, 222222222222.

El ejemplo de esta sección utiliza una transmisión de entrega de Firehose con almacenamiento en Amazon S3. También puedes configurar las transmisiones de entrega de Firehose con diferentes ajustes. Para obtener más información, consulte Creación de un flujo de entrega de Firehose.

> **Note**
>
> El grupo de registros y el destino deben estar en la misma AWS región. Sin embargo, el recurso de AWS al que apunta el destino puede estar ubicado en una región diferente.

> **Note**
>
> Se admite el filtro de suscripción Firehose para una misma cuenta y el flujo de entrega entre regiones.

Temas

Paso 1: Crear un flujo de entrega de Firehose

> **Important**
>
> Antes de completar los siguientes pasos, debes usar una política de acceso para que Firehose pueda acceder a tu bucket de Amazon S3. Para obtener más información, consulte Controlling Access en la Guía para desarrolladores de Amazon Data Firehose.
> Todos los pasos en esta sección (Paso 1) deben realizarse en la cuenta del destinatario de los datos de registro.
> En los ejemplos siguientes, se utiliza Este de EE. UU. (Norte de Virginia). Reemplace esta región por la región correcta para su implementación.

Para crear un flujo de entrega de Firehose para usarlo como destino

1. Cree un bucket de Amazon S3:

```
aws s3api create-bucket --bucket firehose-test-bucket1 --create-bucket-
configuration LocationConstraint=us-east-1
```

2. Crea el rol de IAM que otorga permiso a Firehose para colocar datos en el depósito.

   a. En primer lugar, utilice un editor de texto para crear una política de confianza en un archivo ~/TrustPolicyForFirehose.json.

   ```
   { "Statement": { "Effect": "Allow", "Principal": { "Service":
    "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
    { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
   ```

   b. Cree el rol de IAM y especifique el archivo de política de confianza que acaba de crear.

   ```
   aws iam create-role \
       --role-name FirehosetoS3Role \
       --assume-role-policy-document file://~/TrustPolicyForFirehose.json
   ```

   c. El resultado de este comando debería ser similar a lo siguiente. Haga una nota del nombre del rol y del ARN del rol.

   ```
   {
       "Role": {
           "Path": "/",
           "RoleName": "FirehosetoS3Role",
           "RoleId": "AROAR3BXASEKW7K635M53",
           "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
           "CreateDate": "2021-02-02T07:53:10+00:00",
           "AssumeRolePolicyDocument": {
               "Statement": {
                   "Effect": "Allow",
                   "Principal": {
                       "Service": "firehose.amazonaws.com"
                   },
                   "Action": "sts:AssumeRole",
                   "Condition": {
                       "StringEquals": {
                           "sts:ExternalId": "222222222222"
                       }
                   }
               }
           }
   ```

```
    }
}
```

3. Crea una política de permisos para definir las acciones que Firehose puede realizar en tu cuenta.

   a. En primer lugar, utilice un editor de texto para crear la siguiente política de permisos en un archivo denominado: `~/PermissionsForFirehose.json`. Según el caso de uso, es posible que tenga que agregar más permisos a este archivo.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::firehose-test-bucket1",
            "arn:aws:s3:::firehose-test-bucket1/*"
        ]
    }]
}
```

   b. Ingrese el siguiente comando para asociar la política de permisos que acaba de crear con el rol de IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Ingresa el siguiente comando para crear el flujo de entrega de Firehose. Sustituya *my-role-arn* y *my-bucket-arn* por los valores correctos para su implementación.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::firehose-test-bucket1"}'
```

   El resultado debería tener un aspecto similar al siguiente:

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Paso 2: creación de un destino

> **Important**
>
> Todos los pasos de este procedimiento deben realizarse en la cuenta del destinatario de los datos de registro.

Cuando se crea el destino, CloudWatch Logs envía un mensaje de prueba al destino en nombre de la cuenta del destinatario. Cuando el filtro de suscripciones se active más adelante, CloudWatch Logs envía los eventos de registro al destino en nombre de la cuenta de origen.

Para crear un destino

1. Espera a que se active la transmisión de Firehose en Paso 1: Crear un flujo de entrega de Firehose la que creaste. Puede usar el siguiente comando para comprobar laStreamDescription. StreamStatuspropiedad.

   ```
   aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
   ```

   Además, tome nota de la DeliveryStreamDescription. DeliveryStreamValor ARN, ya que tendrá que usarlo en un paso posterior. Resultado de ejemplo de este comando:

   ```
   {
       "DeliveryStreamDescription": {
           "DeliveryStreamName": "my-delivery-stream",
           "DeliveryStreamARN": "arn:aws:firehose:us-
   east-1:222222222222:deliverystream/my-delivery-stream",
           "DeliveryStreamStatus": "ACTIVE",
           "DeliveryStreamEncryptionConfiguration": {
               "Status": "DISABLED"
           },
           "DeliveryStreamType": "DirectPut",
   ```

```
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

El flujo de entrega puede tardar un minuto o dos en mostrarse en el estado activo.

2. Cuando la transmisión de entrega esté activa, crea la función de IAM que concederá a CloudWatch Logs el permiso para colocar datos en tu transmisión de Firehose. En primer lugar, tendrás que crear una política de confianza en un archivo TrustPolicyFor~/ CWL.json. Utilice un editor de texto para crear esta política. Para obtener más información sobre CloudWatch los puntos de enlace de Logs, consulte los puntos de enlace y las cuotas de Amazon CloudWatch Logs.

   Esta política incluye una clave de contexto de condición global `aws:SourceArn` que especifica la `sourceAccountId` para ayudar a prevenir el problema de seguridad de suplente confuso. Si aún no conoce el ID de cuenta de origen en la primera llamada, le recomendamos que coloque el ARN de destino en el campo ARN de origen. En las llamadas posteriores, debe configurar el ARN de origen para que sea el ARN de origen real que recopiló desde la primera llamada. Para obtener más información, consulte Prevención del suplente confuso.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
     }
}
```

3. Utilice aws iam create-role para crear el rol de IAM y especifique el archivo de política de confianza que acaba de crear.

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json
```

A continuación, se muestra un ejemplo de la salida. Anote el valor de `Role.Arn` devuelto, ya que lo necesitará en un paso posterior.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2023-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Cree una política de permisos para definir qué acciones puede realizar CloudWatch Logs en su cuenta. Primero, usa un editor de texto para crear una política de permisos en un archivo ~/PermissionsFor CWL.json:

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
```

```
}
```

5. Asocie la política de permisos con el rol mediante el siguiente comando:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name
 Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

6. Una vez que la transmisión de entrega de Firehose esté en estado activo y hayas creado la función de IAM, puedes crear el destino de los CloudWatch registros.

   a. Este paso no asociará una política de acceso a su destino y solo es el primer paso de los dos que completan la creación de un destino. Anote el ARN del nuevo destino que se devuelve en la carga, porque lo utilizará como `destination.arn` en un paso posterior.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

   b. Después de completar el paso previo, en la cuenta del destinatario de los datos de registro (222222222222), asocie una política de acceso con el destino. Esta política permite que la cuenta del remitente de los datos de registro (111111111111) tenga acceso al destino justo en la cuenta del destinatario de los datos de registro (222222222222). Puedes usar un editor de texto para incluir esta política en el `~/AccessPolicy.json` archivo:

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

```
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. Esto crea una política que define quién tiene acceso de escritura al destino.
Esta política debe especificar las `logs:PutAccountPolicy` acciones `logs:PutSubscriptionFilter` y las acciones para acceder al destino. Los usuarios con varias cuentas utilizarán las `PutAccountPolicy` acciones `PutSubscriptionFilter` y para enviar los eventos de registro al destino.

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Paso 3: Crea una política de filtrado de suscripciones a nivel de cuenta

Cambie a la cuenta de envío, que es 111111111111 en este ejemplo. Ahora crearás la política de filtrado de suscripciones a nivel de cuenta en la cuenta remitente. En este ejemplo, el filtro hace que todos los eventos de registro que contengan la cadena ERROR en todos los grupos de registros excepto en dos se envíen al destino que creó anteriormente.

```
aws logs put-account-policy \
    --policy-name "CrossAccountFirehoseExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"DestinationArn":"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination", "FilterPattern":
 "{$.userIdentity.type = AssumedRole}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Los grupos de registro de la cuenta remitente y el destino deben estar en la misma AWS región. Sin embargo, el destino puede apuntar a un AWS recurso, como un arroyo Firehose, que se encuentra en una región diferente.

Validación del flujo de eventos de registro

Tras crear el filtro de suscripción, CloudWatch Logs reenvía todos los eventos de registro entrantes que coincidan con el patrón de filtro y los criterios de selección al flujo de entrega de Firehose. Los datos comienzan a aparecer en tu bucket de Amazon S3 en función del intervalo de tiempo establecido en la transmisión de entrega de Firehose. Una vez que haya transcurrido el tiempo suficiente, puede verificar los datos comprobando su bucket de Amazon S3. Escriba el siguiente comando para comprobar el bucket:

```
aws s3api list-objects --bucket 'firehose-test-bucket1'
```

El resultado de ese comando será similar a lo siguiente:

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2023-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

Puede recuperar un objeto específico del bucket al introducir el siguiente comando. Reemplace el valor de `key` con el valor que encontró en el comando anterior.

```
aws s3api get-object --bucket 'firehose-test-bucket1' --key '2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

Los datos en el objeto de Amazon S3 se comprimen con el formato gzip. Puede examinar los datos sin procesar desde la línea de comando mediante uno de los siguientes comandos:

Linux:

```
zcat testfile.gz
```

macOS:

```
zcat <testfile.gz
```

Modificación de la suscripción al destino en tiempo de ejecución

Puede encontrar situaciones en las que tenga que agregar o eliminar remitentes de registros de un destino de su propiedad. Puedes usar las `PutAccountPolicy` acciones PutDestinationPolicyy en tu destino con la nueva política de acceso. En el siguiente ejemplo, una cuenta 111111111111 agregada anteriormente deja de enviar datos de registro y se habilita la cuenta 333333333333.

1. Busca la política que está asociada actualmente con el destino TestDestination y anota lo siguiente: AccessPolicy

   ```
   aws logs describe-destinations \
       --destination-name-prefix "testFirehoseDestination"
   ```

   Los datos devueltos podrían tener este aspecto.

   ```
   {
       "destinations": [
           {
               "destinationName": "testFirehoseDestination",
               "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
   my-delivery-stream",
               "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
               "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
   \" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
    \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
   \" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
   east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
               "arn": "arn:aws:logs:us-east-1:
    222222222222:destination:testFirehoseDestination",
               "creationTime": 1612256124430
   ```

```
        }
    ]
}
```

2. Actualice la política para reflejar que la cuenta 111111111111 está detenida y que la cuenta 333333333333 está habilitada. Coloque esta política en el archivo ~/ NewAccessPolicy .json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. Use el siguiente comando para asociar la política definida en el NewAccessPolicyarchivo.json con el destino:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

Esto finalmente deshabilita los eventos de registro del ID de cuenta 111111111111. Los eventos de registro del ID de cuenta 333333333333 empiezan a fluir al destino en cuanto el propietario de la cuenta 333333333333 crea un filtro de suscripción.

Prevención del suplente confuso

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema

de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger los datos de todos los servicios cuyos directores de servicio tengan acceso a los recursos de su cuenta.

Recomendamos utilizar las claves de contexto de condición `aws:SourceOrgPaths`global `aws:SourceArnaws:SourceAccountaws:SourceOrgID`, y las claves de condición global en las políticas de recursos para limitar los permisos que otorgan otro servicio al recurso. Utilice `aws:SourceArn` para asociar solo un recurso al acceso entre servicios. Utilice `aws:SourceAccount` para permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios. Utilice `aws:SourceOrgID` para permitir que cualquier recurso de cuentas dentro de una organización se asocie al uso entre servicios. Utilice `aws:SourceOrgPaths` para asociar cualquier recurso de cuentas dentro de una ruta de AWS Organizations al uso entre servicios. Para obtener más información sobre el uso y la comprensión de las rutas, consulte Comprender la ruta de la AWS Organizations entidad.

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. Si no conoce el ARN completo del recurso o si está especificando varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con caracteres comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un ARN de bucket de Amazon S3, debe utilizar `aws:SourceAccount` y `aws:SourceArn` para limitar los permisos.

Para protegerse contra el problema del suplente confuso a gran escala, utilice la clave de contexto de condición global `aws:SourceOrgID` o `aws:SourceOrgPaths`con el identificador de organización o la ruta de organización del recurso en sus políticas basadas en recursos. Las políticas que incluyan la clave `aws:SourceOrgID` o `aws:SourceOrgPaths` incluirán automáticamente las cuentas correctas y no requerirán una actualización manual cuando se agregan, quitan o mueven cuentas en la organización.

Las políticas documentadas para conceder acceso a CloudWatch los registros para escribir datos Paso 1: crear un destino en Kinesis Data Streams y Firehose Paso 2: creación de un destino y muestran cómo puede utilizar la clave contextual awsSourceArn : global condition para ayudar a evitar el confuso problema de los diputados.

Prevención de la recursión de registros

Con los filtros de suscripción, se corre el riesgo de provocar una recursión infinita de registros, lo que, si no se evita, puede provocar un gran aumento de la facturación por ingestión tanto en CloudWatch los registros como en tu destino. Esto puede ocurrir cuando un filtro de suscripciones está asociado a un grupo de registros que recibe eventos de registro como resultado del flujo de trabajo de entrega de suscripciones. Los registros ingeridos en el grupo de registros se entregarán al destino, lo que provocará que el grupo de registros ingiera más registros que luego se reenviarán de nuevo al destino, lo que creará un bucle de recursión.

Por ejemplo, considere un filtro de suscripción con el destino Firehose, que envía eventos de registro a Amazon S3. Además, también hay una función Lambda que procesa los nuevos eventos enviados a Amazon S3 y produce algunos registros por sí misma. Si el filtro de suscripción se aplica al grupo de registros de la función Lambda, los eventos de registro producidos por la función se reenviarán a Firehose y Amazon S3 en el destino, que luego volverá a invocar la función, lo que provocará que se generen más registros y se reenvíen a Firehose y Amazon S3, se volverá a invocar la función y así sucesivamente. Esto ocurrirá en un bucle infinito, lo que provocará un aumento inesperado en la facturación de registros, Firehose y Amazon S3.

Si la función Lambda está asociada a una VPC con los registros de flujo habilitados para los registros, el grupo de CloudWatch registros de la VPC también puede provocar una recursión de registros.

Le recomendamos que no aplique filtros de suscripción a los grupos de registros que forman parte de su flujo de trabajo de entrega de suscripciones. Para los filtros de suscripción a nivel de cuenta, usa el `selectionCriteria` parámetro de la `PutAccountPolicy` API para excluir estos grupos de registros de la política.

Al excluir los grupos de registros, tenga en cuenta los siguientes AWS servicios que producen registros y pueden formar parte de sus flujos de trabajo de entrega de suscripciones:

- Amazon EC2 con Fargate
- Lambda
- AWS Step Functions
- Registros de flujo de Amazon VPC que están habilitados para Logs CloudWatch

> **Note**
>
> Los eventos de registro generados por el grupo de registros de un destino de Lambda no se reenviarán a la función de Lambda para una política de filtrado de suscripciones a nivel de cuenta. En este caso, no `selectionCriteria` es necesario excluir el uso del grupo de registros de la función Lambda de destino para las políticas de suscripción de cuentas.

Filtro de sintaxis de patrones para filtros de métricas, filtros de suscripción, eventos de registro de filtros y Live Tail

> **Note**
>
> Para obtener información sobre cómo consultar sus grupos de CloudWatch registros con el lenguaje de consulta de Amazon Logs Insights, consulteCloudWatch Sintaxis de consultas de Logs Insights.

Con CloudWatch Logs, puede utilizar filtros de métricas para transformar los datos de registro en métricas procesables, filtros de suscripción para dirigir los eventos de registro a otros AWS servicios, filtrar los eventos de registro para buscar eventos de registro y Live Tail para ver sus registros de forma interactiva en tiempo real a medida que se van incorporando.

Los patrones de filtro conforman la sintaxis que utilizan los filtros de métricas, los filtros de suscripción, filtro de eventos de registro y Live Tail para hacer coincidir los términos de los eventos de registro. Los términos pueden ser palabras, frases exactas o valores numéricos. Las expresiones regulares (regex) se pueden usar para crear patrones de filtro independientes o se pueden incorporar con patrones de filtro JSON y delimitados por el espacio.

Cree patrones de filtro con los términos que desea que coincidan. Los patrones de filtro solo devuelven los eventos de registro que contienen los términos definidos. Puedes probar los patrones de filtros en la consola. CloudWatch

Temas

Sintaxis de expresiones regulares (regex) compatibles

Sintaxis de expresiones regulares compatibles

Cuando utilice expresiones regulares para buscar y filtrar datos de registro, debe rodear las expresiones con `%`.

Los patrones de filtrado con expresiones regulares solo pueden incluir lo siguiente:

- Caracteres alfanuméricos: un carácter alfanumérico es un carácter que puede ser una letra (de la A a la Z o de la a la z) o un dígito (del 0 al 9).
- Caracteres simbólicos compatibles, entre los cuales se incluyen: “_”, “#”, “=”, “@”, “/”, “;”, “,” y “-”. Por ejemplo, se rechazaría `%something!%` porque “!” no es compatible.
- Operadores compatibles, entre los cuales se incluyen: “^”, “$”, “?”, “[”, “]”, “{”, “}”, “|”, “\”, “*”, “+” y “.”.

Los operadores (y) no son compatibles. No puede usar paréntesis para definir un subpatrón.

Los caracteres de varios bytes no son compatibles.

> **Note**
>
> Cuotas
> Hay un máximo de 5 patrones de filtro que contienen expresiones regulares para cada grupo de registro al crear filtros de métricas o filtros de suscripción.
> Hay un límite de 2 expresiones regulares para cada patrón de filtro al crear un patrón de filtro delimitado o JSON para los filtros de métricas y los filtros de suscripción o al filtrar los eventos de registro. o Live Tail.

Uso de operadores compatibles

- ^: ancla la coincidencia al inicio de una cadena. Por ejemplo, `%^[hc]at%` coincide con los términos en inglés “hat” y “cat”, pero solo al principio de una cadena.
- $: ancla la coincidencia al final de una cadena. Por ejemplo, `%[hc]at$%` coincide con los términos en inglés “hat” y “cat”, pero solo al final de una cadena.
- ?: Coincide con cero o más instancias del término anterior. Por ejemplo, `%colou?r%` puede coincidir tanto con el término en inglés “color” como con el término en inglés “colour”.

- `[]`: define una clase de caracteres. Coincide con la lista el rango de caracteres que figuran entre corchetes. Por ejemplo, `%[abc]%` coincide con “a”, “b” o “c”; `%[a-z]%` coincide con cualquier letra minúscula de la “a” a la “z”; y `%[abcx-z]%` coincide con las letras “a”, “b”, “c”, “x”, “y” o “z”.
- `{m, n}`: coincide con el término anterior al menos m y no más de n veces. Por ejemplo, `%a{3,5}%` solo coincide con “aaa”, “aaaa” y “aaaaa”.

> **Note**
>
> Se pueden omitir m o n si decide no definir un mínimo o un máximo.

- `|`: “O” booleano, que coincide con el término que aparece a ambos lados de la barra vertical. Por ejemplo, `%gra|ey%` puede coincidir con los términos en inglés “gray” o “grey”.

> **Note**
>
> Un término es como un carácter único o una clase de caracteres repetidos que utiliza uno de los siguientes operadores: `?`, `*`, `+` o `{n,m}`.

- `\`: carácter de escape, que permite utilizar el significado literal de un operador en lugar de su significado especial. Por ejemplo, `%\[.\]%` coincide con cualquier carácter individual rodeado de “[“ y ”]”, ya que los corchetes están separados, como “[a]”, “[b]”, “[7]”, “[@]”, “[]]” y “[ ]”.

> **Note**
>
> `%10\.10\.0\.1%` es la forma correcta de crear una expresión regular que coincida con la dirección IP 10.10.0.1.

- `*`: Coincide con cero o más instancias del término anterior. Por ejemplo, `%ab*c%` puede coincidir con “ac”, “abc” y “abbbc”; `%ab[0-9]*%` puede coincidir con “ab”, “ab0” y “ab129”.
- `+`: Coincide con una o más instancias del término anterior. Por ejemplo, `%ab+c%` puede coincidir con “abc”, “abbc” y “abbbc”, pero no con “ac”.
- `.`: coincide con cualquier carácter. Por ejemplo, `%.at%` coincide con cualquier cadena de tres caracteres que termine en “at”, incluidos los términos en inglés “hat”, “cat”, “bat”, “4at”, “#at” y “at” (que comienza con un espacio).

> **Note**
>
> Al crear una expresión regular para que coincida con las direcciones IP, es importante evitar el uso del operador `.`. Por ejemplo, `%10.10.0.1%` puede coincidir con “10010,051”, lo que podría no ser el objetivo real de la expresión.

- `\d`, `\D`: coincide con un carácter que sea un dígito o que no lo sea. Por ejemplo, `%\d%` es equivalente a `%[0-9]%`, y `%\D%` es equivalente a `%[^0-9]%`.

> **Note**
>
> El operador en mayúscula indica el inverso de su homólogo en minúscula.

- `\s`, `\S`: coincide con un carácter con espacio en blanco o sin espacio en blanco.

> **Note**
>
> El operador en mayúscula indica el inverso de su homólogo en minúscula. Los espacios en blanco incluyen los caracteres tab (`\t`), space ( ) y newline (`\n`).

- `\w`, `\W`: coincide con un carácter alfanumérico o no alfanumérico. Por ejemplo, `%\w%` es equivalente a `%[a-zA-Z_0-9]%`, y `%\W%` es equivalente a `%[^a-zA-Z_0-9]%`.

> **Note**
>
> El operador en mayúscula indica el inverso de su homólogo en minúscula.

- `\xhh`: coincide con la asignación ASCII de un carácter hexadecimal de dos dígitos. `\x` es la secuencia de escape que indica que los siguientes caracteres representan el valor hexadecimal para ASCII. hh especifica los dos dígitos hexadecimales (0-9 y A-F) que apuntan a un carácter de la tabla ASCII.

> **Note**
>
> Puede utilizar `\xhh` para hacer coincidir caracteres de símbolo que no son compatibles con el patrón de filtro. Por ejemplo, `%\x3A%` coincide con `:`; y `%\x28%` coincide con `(`.

Uso de los patrones de filtro para hacer coincidir los términos con una expresión regular (regex)

Haga coincidir los términos mediante el uso de regex

Puede hacer coincidir los términos de sus eventos de registro mediante el uso de un patrón de expresiones regulares rodeado de % (signos de porcentaje antes y después del patrón de expresiones regulares). El siguiente fragmento de código muestra un ejemplo de un patrón de filtro que devuelve todos los eventos de registro que consisten en la palabra clave AUTHORIZED.

Para ver una lista de las expresiones regulares compatibles, consulte Expresiones regulares compatibles.

```
%AUTHORIZED%
```

Este patrón de filtro devuelve mensajes de eventos de registro, como los siguientes:

- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[SUCCESS 200] AUTHORIZED REQUEST`

El uso de patrones de filtro para hacer coincidir los términos en eventos de registro sin estructura.

Haga coincidir los términos de los eventos de registro no estructurados

En los siguientes ejemplos se incluyen fragmentos de código que muestran cómo puede utilizar patrones de filtro para hacer coincidir los términos de los eventos de registro sin estructura.

> **Note**
>
> Los patrones de filtro distinguen mayúsculas y minúsculas. Incluya frases y términos exactos que incluyan caracteres no alfanuméricos entre comillas dobles (“”).

Example: Match a single term

En el siguiente fragmento de código, se muestra un ejemplo de un patrón de filtro de un solo término que devuelve todos los eventos de registro en los que los mensajes contienen la palabra ERROR.

```
ERROR
```

Este patrón de filtro coincide con mensajes de eventos de registro, como los siguientes:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match multiple terms

El siguiente fragmento de código muestra un ejemplo de un patrón de filtro de varios términos que devuelve todos los eventos de registro en los que los mensajes contienen las palabras ERROR y ARGUMENTS.

```
ERROR ARGUMENTS
```

El filtro devuelve mensajes de eventos de registro, como los siguientes:

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Este patrón de filtro no devuelve los siguientes mensajes de eventos de registro, porque no contienen los dos términos especificados en el patrón de filtro.

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Example: Match optional terms

Puede utilizar la coincidencia de patrones para crear patrones de filtro que devuelvan eventos de registro que contengan términos opcionales. Coloque un signo de interrogación (“?”) antes de los términos que desea hacer coincidir. El siguiente fragmento de código muestra un ejemplo de un patrón de filtro que devuelve todos los eventos de registro en los que los mensajes contienen la palabra ERROR o ARGUMENTOS.

```
?ERROR ?ARGUMENTS
```

Este patrón de filtro coincide con mensajes de eventos de registro, como los siguientes:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

> **Note**
>
> No se puede combinar el signo de interrogación (“?”) con otros patrones de filtrado, como incluir y excluir términos. Si combina “?” con otros patrones de filtrado, se ignorará el signo de interrogación (“?”).
> Por ejemplo, el siguiente patrón de filtrado coincide con todos los eventos que contienen la palabra `REQUEST`, pero se ignora el filtro del signo de interrogación (“?”) y no tiene ningún efecto.
>
> ```
> ?ERROR ?ARGUMENTS REQUEST
> ```
>
> Coincidencias de eventos de registro
>
> - `[INFO] REQUEST FAILED`
> - `[WARN] UNAUTHORIZED REQUEST`
> - `[ERROR] 400 BAD REQUEST`

Example: Match exact phrases

El siguiente fragmento de código muestra un ejemplo de un patrón de filtro que devuelve todos los eventos de registro en los que los mensajes contienen la frase exacta INTERNAL SERVER ERROR.

```
"INTERNAL SERVER ERROR"
```

Este patrón de filtro devuelve el siguiente mensaje de evento de registro:

- `[ERROR 500] INTERNAL SERVER ERROR`

Example: Include and exclude terms

Puede crear patrones de filtro que devuelvan eventos de registro en los que los mensajes incluyen algunos términos y excluyen otros. Coloque un símbolo menos ("-") antes de los términos que desea excluir. El siguiente fragmento de código muestra un ejemplo de un patrón de filtro que devuelve todos los eventos de registro en los que los mensajes incluyen el término ERROR y excluyen el término ARGUMENTS.

```
ERROR -ARGUMENTS
```

Este patrón de filtro devuelve mensajes de eventos de registro, como los siguientes:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Este patrón de filtro no devuelve los siguientes mensajes de eventos de registro, porque contienen la palabra ARGUMENTS.

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match everything

Puede hacer una coincidencia total en los eventos de registro con comillas dobles. El siguiente fragmento de código muestra un ejemplo de un patrón de filtro que devuelve todos los eventos de registro.

```
" "
```

Uso de patrones de filtro para hacer coincidir los términos en eventos de registro JSON

Escribir patrones de filtro para eventos de registro de JSON

En los siguientes ejemplos, se describe cómo escribir la sintaxis de los patrones de filtro que coinciden con los términos JSON que contienen cadenas y valores numéricos.

Writing filter patterns that match strings

Puede crear patrones de filtro para que coincidan con cadenas en eventos de registro JSON. El siguiente fragmento de código muestra un ejemplo de la sintaxis de los patrones de filtro basados en cadenas.

```
{ PropertySelector EqualityOperator String }
```

Coloque los patrones de filtro en los corchetes (“{}”). Los patrones de filtro basados en cadenas deben contener los siguientes elementos:

- Selector de propiedades

  Active los selectores de propiedades con un signo de dólar seguido de un punto (“$.”). Los selectores de propiedades son cadenas alfanuméricas que admiten los caracteres guion (“-”) y guion bajo (“_”). Las cadenas no admiten la notación científica. Los selectores de propiedades apuntan a los nodos de valor en los eventos de registro JSON. Los nodos de valor pueden ser cadenas o números. Coloque matrices después de los selectores de propiedades. Los elementos de las matrices siguen un sistema de numeración basado en cero, lo que significa

que el primer elemento de la matriz es el elemento 0, el segundo elemento es el elemento 1, etc. Incluya elementos entre corchetes (“[]”). Si un selector de propiedades apunta a una matriz u objeto, el patrón del filtro no coincidirá con el formato del registro. Si la propiedad JSON contiene un punto (`"."`), se puede utilizar la notación entre corchetes para seleccionar esa propiedad.

> **Note**
>
> Selector de comodín
> Puede utilizar el comodín JSON para seleccionar cualquier elemento de la matriz o campo de objeto JSON.
> Cuotas
> Solo puede utilizar un selector de caracteres comodín en un selector de propiedades.

- Operador de igualdad

  Establezca operadores de igualdad con uno de los siguientes símbolos: igual (“=”) o no igual (“!=”). Los operadores de igualdad devuelven un valor booleano (verdadero o falso).

- Cadena

  Puede incluir cadenas entre comillas dobles (“”). Las cadenas que contienen tipos distintos de caracteres alfanuméricos y el símbolo de guion bajo deben colocarse entre comillas dobles. Use el asterisco (“*”) como comodín para que coincida con el texto.

> **Note**
>
> Puede usar cualquier expresión regular condicional al crear patrones de filtro para que coincidan con los términos de los eventos de registro de JSON. Para ver una lista de las expresiones regulares compatibles, consulte Expresiones regulares compatibles.

El siguiente fragmento de código incluye un ejemplo de patrones de filtro que muestra cómo puede dar formato a un filtro de métricas para que coincida con un término JSON con una cadena.

```
{ $.eventType = "UpdateTrail" }
```

Writing filter patterns that match numeric values

Puede crear patrones de filtro para que coincidan con los valores numéricos en eventos de registro JSON. El siguiente fragmento de código muestra un ejemplo de la sintaxis de los patrones de filtro que coinciden con valores numéricos.

```
{ PropertySelector NumericOperator Number }
```

Coloque los patrones de filtro en los corchetes (“{}”). Los patrones de filtro que coinciden con los valores numéricos deben tener los siguientes elementos:

- Selector de propiedades

  Active los selectores de propiedades con un signo de dólar seguido de un punto (“$.”). Los selectores de propiedades son cadenas alfanuméricas que admiten los caracteres guion (“-”) y guion bajo (“_”). Las cadenas no admiten la notación científica. Los selectores de propiedades apuntan a los nodos de valor en los eventos de registro JSON. Los nodos de valor pueden ser cadenas o números. Coloque matrices después de los selectores de propiedades. Los elementos de las matrices siguen un sistema de numeración basado en cero, lo que significa que el primer elemento de la matriz es el elemento 0, el segundo elemento es el elemento 1, etc. Incluya elementos entre corchetes (“[]”). Si un selector de propiedades apunta a una matriz u objeto, el patrón del filtro no coincidirá con el formato del registro. Si la propiedad JSON contiene un punto (`"."`), se puede utilizar la notación entre corchetes para seleccionar esa propiedad.

  > **Note**
  >
  > Selector de comodín
  > Puede utilizar el comodín JSON para seleccionar cualquier elemento de la matriz o campo de objeto JSON.
  > Cuotas
  > Solo puede utilizar un selector de caracteres comodín en un selector de propiedades.

- Operador numérico

Establezca operadores numéricos con uno de los siguientes símbolos: mayor que (“>”), menor que (“<”), igual (“=”), no igual (“!=”), mayor o igual que (“>=”) o menor o igual que (“<=”) o menor o igual que (“<=”).

- Número

  Puede utilizar números enteros que contengan símbolos más (“+”) o menos (“-”) y seguir la notación científica. Use el asterisco (“*”) como comodín para que coincida con los números.

El siguiente fragmento de código contiene ejemplos que muestran cómo puede dar formato a los patrones de filtro para que los términos JSON coincidan con valores numéricos.

```
// Filter pattern with greater than symbol
{ $.bandwidth > 75 }
// Filter pattern with less than symbol
{ $.latency < 50 }
// Filter pattern with greater than or equal to symbol
{ $.refreshRate >= 60 }
// Filter pattern with less than or equal to symbol
{ $.responseTime <= 5 }
// Filter pattern with equal sign
{ $.errorCode = 400}
// Filter pattern with not equal sign
{ $.errorCode != 500 }
// Filter pattern with scientific notation and plus symbol
{ $.number[0] = 1e-3 }
// Filter pattern with scientific notation and minus symbol
{ $.number[0] != 1e+3 }
```

Haga coincidir los términos en todos los eventos de registro de JSON con expresiones simples

En los siguientes ejemplos, se incluyen fragmentos de código que muestran cómo los patrones de filtro pueden coincidir con los términos de un evento de registro JSON.

> **Note**
>
> Si prueba un patrón de filtro de ejemplo con el evento de registro JSON de ejemplo, debe ingresar el registro JSON de ejemplo en una sola línea.

Evento de registro JSON

```
{
      "eventType": "UpdateTrail",
      "sourceIPAddress": "111.111.111.111",
      "arrayKey": [
            "value",
            "another value"
      ],
      "objectList": [
           {
             "name": "a",
             "id": 1
           },
           {
             "name": "b",
             "id": 2
           }
      ],
      "SomeObject": null,
      "cluster.name": "c"
}
```

Example: Filter pattern that matches string values

Este patrón de filtro coincide con la cadena `"UpdateTrail"` de la propiedad `"eventType"`.

```
{ $.eventType = "UpdateTrail" }
```

Example: Filter pattern that matches string values (IP address)

Este patrón del filtro contiene un comodín y coincide con la propiedad `"sourceIPAddress"` porque no contiene un número con el prefijo `"123.123."`.

```
{ $.sourceIPAddress != 123.123.* }
```

Example: Filter pattern that matches a specific array element with a string value

Este patrón de filtro coincide con el elemento `"value"` de la matriz `"arrayKey"`.

```
{ $.arrayKey[0] = "value" }
```

Example: Filter pattern that matches a string using regex

Este patrón de filtro coincide con la cadena `"Trail"` de la propiedad `"eventType"`.

```
{ $.eventType = %Trail% }
```

Example: Filter pattern that uses a wildcard to match values of any element in the array using regex

El patrón de filtro contiene expresiones regulares que coinciden con el elemento `"value"` de la matriz `"arrayKey"`.

```
{ $.arrayKey[*] = %val.{2}% }
```

Example: Filter pattern that uses a wildcard to match values of any element with a specific prefix and subnet using regex (IP address)

Este patrón de filtro contiene expresiones regulares que coinciden con el elemento `"111.111.111.111"` de la propiedad `"sourceIPAddress"`.

```
{ $.* = %111\.111\.111\.1[0-9]{1,2}% }
```

> **Note**
>
> Cuotas
>
> Solo puede utilizar un selector de caracteres comodín en un selector de propiedades.

Example: Filter pattern that matches a JSON property with a period (.) in the key

```
{ $.['cluster.name'] = "c" }
```

Example: Filter pattern that matches JSON logs using IS

Puede crear patrones de filtro que coincidan con los campos de los registros JSON con la variable `IS`. La variable `IS` puede coincidir con los campos que contienen los valores `NULL`, `TRUE` o bien `FALSE`. El siguiente patrón de filtro devuelve registros JSON donde el valor de `SomeObject` es `NULL`.

```
{ $.SomeObject IS NULL }
```

Example: Filter pattern that matches JSON logs using NOT EXISTS

Puede crear patrones de filtro con la `NOT EXISTS` variable para devolver registros JSON que no contengan campos específicos en los datos de registro. El siguiente patrón de filtro utiliza `NOT EXISTS` para devolver registros JSON que no contienen el campo `SomeOtherObject`.

```
{ $.SomeOtherObject NOT EXISTS }
```

> **Note**
>
> Las variables `IS NOT` y `EXISTS` no se admiten actualmente.

Uso de expresiones compuestas para coincidir términos en objetos JSON

Puede utilizar los operadores lógicos AND (“&&”) y OR (“||”) en los patrones de filtro para crear expresiones compuestas que coincidan con los eventos de registro en los que se cumplen dos o más condiciones. Las expresiones compuestas admiten el uso de paréntesis (“()”) y el siguiente orden de operaciones estándar: () > && > ||. En los siguientes ejemplos, se incluyen fragmentos de código que muestran cómo puede utilizar los patrones de filtro con expresiones compuestas para hacer coincidir los términos de un objeto JSON.

Objeto JSON

```
{
```

```
    "user": {
        "id": 1,
        "email": "John.Stiles@example.com"
    },
    "users": [
        {
         "id": 2,
         "email": "John.Doe@example.com"
        },
        {
         "id": 3,
         "email": "Jane.Doe@example.com"
        }
    ],
    "actions": [
        "GET",
        "PUT",
        "DELETE"
    ],
    "coordinates": [
        [0, 1, 2],
        [4, 5, 6],
        [7, 8, 9]
    ]
}
```

Example: Expression that matches using AND (&&)

Este patrón de filtro contiene una expresión compuesta que encuentra una coincidencia de `"id"` en `"user"` con un valor numérico de 1 y `"email"` en `"users"` con la cadena `"John.Doe@example.com"`.

```
{ ($.user.id = 1) && ($.users[0].email = "John.Doe@example.com") }
```

Example: Expression that matches using OR (||)

Este patrón de filtro contiene una expresión compuesta que encuentra una coincidencia de `"email"` en `"user"` con la cadena `"John.Stiles@example.com"`.

```
{ $.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch" &&
 $.actions[2] = "nonmatch" }
```

Example: Expression that doesn't match using AND (&&)

Este patrón de filtro contiene una expresión compuesta que no encuentra ninguna coincidencia porque la expresión no coincide con la tercera acción en `"actions"`.

```
{ ($.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = "nonmatch") &&
 $.actions[2] = "nonmatch" }
```

> **Note**
>
> Cuotas
> Solo puede usar un selector de comodín en un selector de propiedades y hasta tres selectores de comodín en un patrón de filtro con expresiones compuestas.

Example: Expression that doesn't match using OR (||)

Este patrón de filtro contiene una expresión compuesta que no encuentra ninguna coincidencia porque la expresión no coincide con la primera propiedad de `"users"` o con la tercera acción en `"actions"`.

```
{ ($.user.id = 2 && $.users[0].email = "nonmatch") || $.actions[2] = "GET" }
```

Uso de la coincidencia de los patrones de filtro para hacer coincidir términos en eventos de registro delimitados por espacios

Escribir los patrones de filtro para todos los eventos de registro delimitados por espacios

Puede crear patrones de filtro para que coincidan con los términos de todos los eventos de registro delimitados por espacios. A continuación, se proporciona un ejemplo de un evento de registro delimitado por espacios y se describe cómo escribir la sintaxis de los patrones de filtro que coinciden con los términos del evento de registro delimitado por espacios.

> **Note**
>
> Puede usar cualquier expresión regular condicional al crear los patrones de filtro que coincidan con los términos de todos los eventos de registro delimitados por espacios. Para ver una lista de las expresiones regulares compatibles, consulte Expresiones regulares compatibles.

Example: Space-delimited log event

En el siguiente fragmento de código, se muestra un evento de registro delimitado por espacios que contiene siete campos: `ip`, `user`, `username`, `timestamp`, `request`, `status_code` y `bytes`.

```
127.0.0.1 Prod frank [10/Oct/2000:13:25:15 -0700] "GET /index.html HTTP/1.0" 404
 1534
```

> **Note**
>
> Los caracteres entre corchetes (“[]”) y comillas dobles (“”) se consideran campos individuales.

Writing filter patterns that match terms in a space-delimited log event

Para crear un patrón de filtro que asigne y extraiga valores de los campos de un evento de registro delimitado por espacios, incluya el patrón de filtro entre corchetes (“[]”) y especifique campos con nombres separados por comas (“,”). El siguiente patrón de filtro analiza siete campos.

```
[ip=%127\.0\.0\.[1-9]%, user, username, timestamp, request =*.html*, status_code =
 4*, bytes]
```

Puede utilizar operadores numéricos (**>**, **<**, **=**, **!=**, **>=** o **<=**) y el asterisco (*) como comodín para asignar las condiciones del patrón del filtro. En el ejemplo, el patrón de filtro `ip` utiliza expresiones regulares que coinciden con el rango de direcciones IP 127.0.0.1 a 127.0.0.9, `request` contiene un comodín que indica que debe extraer un valor con `.html`, y `status_code` contiene un comodín que indica que debe extraer un valor que comience con `4`.

Si no conoce el número de campos que está analizando en un evento de registro delimitado por espacios, puede utilizar puntos suspensivos (...) para hacer referencia a cualquier campo sin nombre. Los puntos suspensivos pueden hacer referencia a tantos campos como sea necesario. En el ejemplo siguiente, se muestra un patrón de filtro con puntos suspensivos que representan los cuatro primeros campos sin nombre que se muestran en el patrón de filtro del ejemplo anterior.

```
[..., request =*.html*, status_code = 4*, bytes]
```

También puede utilizar los operadores lógicos AND (&&) y OR (||) para crear expresiones compuestas. El siguiente patrón de filtro contiene una expresión compuesta que indica que el valor de `status_code` debe ser `404` o `410`.

```
[ip, user, username, timestamp, request =*.html*, status_code = 404 || status_code =
 410, bytes]
```

Coincidir términos en eventos de registro delimitados por espacios mediante el uso de la coincidencia de patrones

Puede utilizar la coincidencia de patrones para crear patrones de filtro delimitados por espacios que coincidan con términos en un orden específico. Especifique el orden de los términos con indicadores. Use w1 para representar su primer término y w2 para el segundo, y así sucesivamente para representar el orden de los términos posteriores. Coloque comas (“,”) entre sus términos. En los siguientes ejemplos, se incluyen fragmentos de código que muestran cómo se puede utilizar la coincidencia de patrones con filtro delimitados por espacios.

> **Note**
>
> Puede usar cualquier expresión regular condicional al crear los patrones de filtro que coincidan con los términos de todos los eventos de registro delimitados por espacios. Para ver una lista de las expresiones regulares compatibles, consulte Expresiones regulares compatibles.

Evento de registro delimitado por el espacio

```
INFO 09/25/2014 12:00:00 GET /service/resource/67 1200
INFO 09/25/2014 12:00:01 POST /service/resource/67/part/111 1310
WARNING 09/25/2014 12:00:02 Invalid user request
ERROR 09/25/2014 12:00:02 Failed to process request
```

Example: Match terms in order

El siguiente patrón de filtro delimitado por espacios devuelve eventos de registro en los que la primera palabra de los eventos de registro es ERROR.

```
[w1=ERROR, w2]
```

> **Note**
>
> Al crear patrones de filtro delimitados por espacios que utilizan la coincidencia de patrones, debe incluir un indicador en blanco después de especificar el orden de los términos. Por ejemplo, si crea un patrón de filtro que devuelve eventos de registro en los

que se encuentra la primera palabra ERROR, incluya un indicador w2 en blanco después del término w1.

Example: Match terms with AND (&&) and OR (||)

Puede utilizar los operadores lógicos AND (“&&”) y OR (“||”) para crear patrones de filtro delimitados por espacios que contengan condiciones. El siguiente patrón de filtro devuelve eventos de registro en los que la primera palabra de los eventos es ERROR o WARNING.

```
[w1=ERROR || w1=WARNING, w2]
```

Example: Exclude terms from matches

Puede crear patrones de filtro delimitados por espacios que devuelvan eventos de registro, a excepción de uno o más términos. Coloque un símbolo de no igual (“!=”) antes de los términos que desea excluir. El siguiente fragmento de código muestra un ejemplo de un patrón de filtro que devuelve eventos de registro donde las primeras palabras no son ERROR ni WARNING.

```
[w1!=ERROR && w1!=WARNING, w2]
```

Example: Match the top level item in a resource URI

En el siguiente fragmento de código, se muestra un ejemplo de un patrón de filtro que coincide con el elemento de nivel superior de un URI de recurso que utiliza la expresión regular.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+$%, response_time]
```

Example: Match the child level item in a resource URI

En el siguiente fragmento de código, se muestra un ejemplo de un patrón de filtro que coincide con el elemento de nivel secundario en un URI de recurso que utiliza la expresión regular.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+/part/[0-9]+$%,
 response_time]
```

Habilitar el registro desde AWS los servicios

Si bien muchos servicios publican registros solo en CloudWatch Logs, algunos AWS servicios pueden publicar registros directamente en Amazon Simple Storage Service o Amazon Data Firehose. Si su principal requisito para los registros es almacenarlos o procesarlos en uno de estos servicios, puede hacer que el servicio que produce los registros los envíe directamente a Amazon S3 o Firehose sin necesidad de configuración adicional.

Incluso cuando los registros se publican directamente en Amazon S3 o Firehose, se aplican cargos. Para obtener más información, consulta Registros vendidos en la pestaña Logs de Amazon CloudWatch Pricing.

Algunos AWS servicios utilizan una infraestructura común para enviar sus registros. Para habilitar el registro desde estos servicios, debe iniciar sesión como usuario con ciertos permisos. Además, debe conceder permisos para AWS permitir el envío de los registros.

Para los servicios que requieren estos permisos, se necesitan dos versiones de los permisos. Los servicios que requieren estos permisos adicionales se indican en la tabla como Compatibles [permisos V1] y Compatibles [permisos V2]. Para obtener información sobre estos permisos necesarios, consulte las secciones que aparecen después de la tabla.

Tipo de registro	CloudWatch Logs	Amazon S3	Firehose
Registros de acceso a Amazon API Gateway	Compatibles [permisos V1]		
AWS AppSync logs	Compatible		
Registros MySQL de Amazon Aurora	Compatible		
Amazon Bedrock Registro de bases de conocimiento	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]
Registros de métricas de calidad de medios de Amazon Chime y registros de mensajes SIP	Compatibles [permisos V1]		
CloudFront: registros de acceso		Compatibles [permisos V1]	

Tipo de registro	CloudWatch Logs	Amazon S3	Firehose
AWS CloudHSM registros de auditoría	Compatible		
CloudWatch Evidentemente, registros de eventos de evaluación	Compatibles [permisos V1]	Compatibles [permisos V1]	
CloudWatch Registros de Internet Monitor		Compatibles [permisos V1]	
CloudTrail registros	Compatible		
AWS CodeBuild logs	Compatible		
Amazon CodeWhisperer registros de eventos	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]
Amazon Cognito logs	Compatibles [permisos V1]		
Registros de Amazon Connect	Compatible		
AWS DataSync logs	Compatible		
Registros ElastiCache de Amazon para Redis	Compatibles [permisos V1]		Compatibles [permisos V1]
AWS Elastic Beanstalk logs	Compatible		
Registros de Amazon Elastic Container Service	Compatible		
Registros de plano de control de Amazon Elastic Kubernetes Service	Compatible		
Amazon EventBridge Registro de tuberías	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
AWS Fargate logs	Compatible		

Tipo de registro	CloudWatch Logs	Amazon S3	Firehose
AWS Fault Injection Service registros de experimentos		Compatibles [permisos V1]	
Amazon FinSpace	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
AWS Global Accelerator registros de flujo		Compatibles [permisos V1]	
AWS Glue registros de trabajos	Compatible		
registros de errores de IAM Identity Center	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]
Registros de chat de Amazon Interactive Video Service	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
AWS IoT logs	Compatible		
AWS IoT FleetWise logs	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
AWS Lambda logs	Compatible		
Registros de Amazon Macie	Compatible		
AWS Mainframe Modernization	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
Registros de Amazon Managed Service for Prometheus	Compatibles [permisos V1]		
Registros de agente de Amazon MSK	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
Registros de Amazon MSK Connect	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]

Tipo de registro	CloudWatch Logs	Amazon S3	Firehose
Registros generales y de auditoría de Amazon MQ	Compatible		
AWS Registros de Network Firewall	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
Registros de acceso de Network Load Balancer		Compatibles [permisos V1]	
OpenSearch registros	Compatible		
Registros OpenSearch de ingestión de Amazon Service	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
AWS OpsWorks logs	Compatible		
Registros ServicePostgre SQL de Amazon Relational Database	Compatible		
AWS RoboMaker registros	Compatible		
Registros de consultas de DNS públicas de Amazon Route 53	Compatible		
Registros de consultas de Amazon Route 53 Resolver	Compatibles [permisos V1]	Compatibles [permisos V1]	
SageMaker Eventos de Amazon	Compatibles [permisos V1]		
Eventos para SageMaker trabajadores de Amazon	Compatibles [permisos V1]		
AWS Registros de VPN de sitio a sitio	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]

Tipo de registro	CloudWatch Logs	Amazon S3	Firehose
Registros de Amazon Simple Notification Service	Compatible		
Registros de la política de protección de datos de Amazon Simple Notification Service	Compatible		
Archivos de fuente de datos de instancia de spot de EC2		Compatibles [permisos V1]	
AWS Step Functions Registros de flujo de trabajo rápido y flujo de trabajo estándar	Compatibles [permisos V1]		
Registros de auditoría y estado de Storage Gateway	Compatibles [permisos V1]		
AWS Transfer Family logs	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
Acceso verificado de AWS logs	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
Registros de flujo de Amazon Virtual Private Cloud	Compatible	Compatibles [permisos V1]	Compatibles [permisos V1]
Registros de acceso de Amazon VPC Lattice	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]
AWS WAF logs	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatible
Amazon WorkMail registros	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]

Registro que requiere permisos adicionales [V1]

Algunos AWS servicios utilizan una infraestructura común para enviar sus CloudWatch registros a Logs, Amazon S3 o Firehose. Para habilitar los servicios de AWS que se muestran en la siguiente tabla a fin de enviar sus registros a estos destinos, debe iniciar sesión como usuario con ciertos permisos.

Además, se deben conceder permisos para AWS permitir el envío de los registros. AWS puede crear esos permisos automáticamente al configurar los registros, o puede crearlos usted mismo antes de configurar el registro. Para la entrega entre cuentas, debe crear usted mismo las políticas de permisos manualmente.

Si decide configurar AWS automáticamente los permisos y las políticas de recursos necesarios cuando usted o alguien de su organización configure por primera vez el envío de registros, el usuario que configure el envío de registros debe tener determinados permisos, como se explica más adelante en esta sección. Como alternativa, puede crear las políticas de recursos usted mismo y, por lo tanto, los usuarios que configuran el envío de registros no necesitan tantos permisos.

En la siguiente tabla, se resumen los tipos de registros y los destinos de registro a los que se aplica la información de esta sección.

En las siguientes secciones, se proporcionan más detalles para cada uno de estos destinos.

Los registros se envían a CloudWatch Logs

> **Important**
>
> Al configurar los tipos de registro de la siguiente lista para que se envíen a CloudWatch Logs, AWS crea o cambia las políticas de recursos asociadas al grupo de registros que recibe los registros, si es necesario. Siga leyendo esta sección para ver los detalles.

Esta sección se aplica cuando los tipos de registros enumerados en la tabla de la sección anterior se envían a CloudWatch Logs:

Permisos de usuario

Para poder configurar el envío de cualquiera de estos tipos de CloudWatch registros a Logs por primera vez, debe iniciar sesión en una cuenta con los siguientes permisos.

- `logs:CreateLogDelivery`
- `logs:PutResourcePolicy`
- `logs:DescribeResourcePolicies`
- `logs:DescribeLogGroups`

> **Note**
>
> Cuando especifique el `logs:PutResourcePolicy` permiso `logs:DescribeLogGroupslogs:DescribeResourcePolicies`, o asegúrese de configurar el ARN de su `Resource` línea para que utilice un `*` comodín, en lugar de especificar solo un nombre de grupo de registros. Por ejemplo, `"Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*"`

Si alguno de estos tipos de registros ya se está enviando a un grupo de CloudWatch registros en Logs, para configurar el envío de otro de estos tipos de registros a ese mismo grupo de registros, solo necesita el `logs:CreateLogDelivery` permiso.

Política de recursos del grupo de registros

El grupo de registros al que se envían los registros debe tener una política de recursos que incluya determinados permisos. Si el grupo de registros actualmente no tiene una política de recursos y el usuario que configura el registro tiene los `logs:PutResourcePolicy` `logs:DescribeLogGroups` permisos y los permisos para el grupo de registros, creará AWS automáticamente la siguiente política para dicho grupo cuando comience a enviar los CloudWatch registros a Logs. `logs:DescribeResourcePolicies`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "delivery.logs.amazonaws.com"
        ]
      },
      "Action": [
```

```
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:0123456789:log-group:my-log-group:log-stream:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

Si el grupo de registros tiene una política de recursos, pero esa política no contiene la instrucción que se muestra en la política anterior, y el usuario que configura el registro tiene los permisos `logs:PutResourcePolicy`, `logs:DescribeResourcePolicies` y `logs:DescribeLogGroups` para el grupo de registros, esa instrucción se anexa a la política de recursos del grupo de registros.

Consideraciones de límite de tamaño de la política de recursos del grupo de registros

Estos servicios deben enumerar cada grupo de registros al que envían registros en la política de recursos, y las políticas de recursos de CloudWatch Logs están limitadas a 5120 caracteres. Un servicio que envía registros a un gran número de grupos de registros puede estar dentro de este límite.

Para mitigarlo, CloudWatch Logs supervisa el tamaño de las políticas de recursos utilizadas por el servicio que envía los registros y, cuando detecta que una política se acerca al límite de tamaño de 5120 caracteres, CloudWatch Logs activa `/aws/vendedlogs/*` automáticamente la política de recursos de ese servicio. Puede comenzar a utilizar grupos de registros con nombres que comiencen por `/aws/vendedlogs/` como los destinos de los registros de estos servicios.

Registros enviados a Amazon S3

Cuando configura los registros para que se envíen a Amazon S3, AWS crea o cambia las políticas de recursos asociadas al depósito de S3 que recibe los registros, si es necesario.

Los registros publicados directamente en Amazon S3 se publican en un bucket existente que especifique. Se crean uno o varios archivos de registro cada cinco minutos en el bucket especificado.

Cuando entrega registros por primera vez a un bucket de Amazon S3, el servicio que entrega registros registra al propietario del bucket para asegurarse de que los registros se entregan solo a un bucket perteneciente a esta cuenta. Como resultado, para cambiar el propietario del bucket de Amazon S3, debe volver a crear o actualizar la suscripción de registro en el servicio de origen.

> **Note**
>
> CloudFront utiliza un modelo de permisos diferente al de los demás servicios que envían los registros vendidos a S3. Para obtener más información, consulte Permisos necesarios para configurar el registro estándar y el acceso a los archivos de registro.
> Además, si utiliza el mismo depósito de S3 para CloudFront acceder a los registros y otra fuente de registros, si habilita la ACL en el depósito, CloudFront también se concederán permisos a todas las demás fuentes de registro que utilicen este depósito.

Permisos de usuario

Para poder configurar el envío de cualquiera de estos tipos de registros a Amazon S3 por primera vez, debe iniciar sesión en una cuenta con los siguientes permisos.

- `logs:CreateLogDelivery`
- `S3:GetBucketPolicy`
- `S3:PutBucketPolicy`

Si alguno de estos tipos de registros ya se envía a un bucket de Amazon S3, para configurar el envío de otro de estos tipos de registros al mismo bucket, solo necesita tener el permiso `logs:CreateLogDelivery`.

Política de recursos de bucket de S3

El bucket de S3 al que se envían los registros debe tener una política de recursos que incluya determinados permisos. Si el bucket no tiene actualmente una política de recursos y el usuario que configura el registro tiene los `S3:PutBucketPolicy` permisos `S3:GetBucketPolicy` y para el bucket, creará AWS automáticamente la siguiente política para él cuando comience a enviar los registros a Amazon S3.

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
                },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::my-bucket",
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
                }
            }
        }
    ]
}
```

En la política anterior, para `aws:SourceAccount`, especifique la lista de ID de cuenta para los que se entregan los registros a este bucket. Para `aws:SourceArn`, especifique la lista de ARN del recurso que genera los registros, en el formulario `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Si el bucket tiene una política de recursos, pero esa política no contiene la instrucción que se muestra en la política anterior, y el usuario que configura el registro tiene los permisos `S3:GetBucketPolicy` y `S3:PutBucketPolicy` para el bucket, esa instrucción se anexa a la política de recursos del bucket.

> **Note**
>
> En algunos casos, es posible que veas `AccessDenied` errores al AWS CloudTrail indicar si no se ha concedido el `s3:ListBucket` permiso a`delivery.logs.amazonaws.com`. Para evitar estos errores en sus CloudTrail registros, debe conceder el `s3:ListBucket` permiso `delivery.logs.amazonaws.com` e incluir `Condition` los parámetros que se muestran con el conjunto de `s3:GetBucketAcl` permisos en la política de bucket anterior. Para simplificar esto, en lugar de crear una nueva `Statement`, puede actualizar directamente `AWSLogDeliveryAclCheck` para que sea `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Uso de cifrado en el lado de servidor del bucket de Amazon S3

Puede proteger los datos de su bucket de Amazon S3 habilitando el cifrado del lado del servidor con claves administradas por Amazon S3 (SSE-S3) o el cifrado del lado del servidor con una clave almacenada en (SSE-KMS). AWS KMS AWS Key Management Service Para obtener más información, consulte Protección de los datos con el cifrado del lado del servidor.

Si elige SSE-S3, no se requiere ninguna configuración adicional. Amazon S3 se encarga de la clave de cifrado.

> **Warning**
>
> Si elige SSE-KMS, debe usar una clave administrada por el cliente, ya que no se admite el uso de una clave administrada en este escenario. AWS Si configura el cifrado con una clave AWS administrada, los registros se entregarán en un formato ilegible.

Cuando utilizas una AWS KMS clave gestionada por el cliente, puedes especificar el nombre de recurso de Amazon (ARN) de la clave gestionada por el cliente al activar el cifrado de buckets. Debe agregar lo siguiente a la política de clave para la clave administrada por el cliente (no a la política del bucket para el bucket de S3), de modo que la cuenta de entrega de registros pueda escribir en el bucket de S3.

Si elige SSE-KMS, debe usar una clave administrada por el cliente, ya que no se admite el uso de una clave AWS administrada en este escenario. Cuando utilizas una AWS KMS clave gestionada por el cliente, puedes especificar el nombre de recurso de Amazon (ARN) de la clave gestionada por el cliente al activar el cifrado de buckets. Debe agregar lo siguiente a la política de clave para la clave administrada por el cliente (no a la política del bucket para el bucket de S3), de modo que la cuenta de entrega de registros pueda escribir en el bucket de S3.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
        }
}
```

Para `aws:SourceAccount`, especifique la lista de ID de cuenta para los que se entregan los registros a este bucket. Para `aws:SourceArn`, especifique la lista de ARN del recurso que genera los registros, en el formulario `arn:aws:logs:`*source-region*`:`*source-account-id*`:*`.

Registros enviados a Firehose

Esta sección se aplica cuando los tipos de troncos enumerados en la tabla de la sección anterior se envían a Firehose:

Permisos de usuario

Para poder configurar el envío de cualquiera de estos tipos de registros a Firehose por primera vez, debes iniciar sesión en una cuenta con los siguientes permisos.

- `logs:CreateLogDelivery`
- `firehose:TagDeliveryStream`
- `iam:CreateServiceLinkedRole`

Si alguno de estos tipos de registros ya se está enviando a Firehose, para configurar el envío de otro de estos tipos de registros a Firehose, solo necesita tener los permisos y. `logs:CreateLogDelivery` `firehose:TagDeliveryStream`

Roles de IAM utilizados para permisos

Como Firehose no usa políticas de recursos, AWS usa roles de IAM al configurar estos registros para enviarlos a Firehose. AWS crea un rol vinculado a un servicio denominado. AWSServiceRoleForLogDelivery Este rol vinculado a un servicio incluye los siguientes permisos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
```

```
    ]
}
```

Esta función vinculada al servicio concede permisos para todas las transmisiones de entrega de Firehose que tengan la `LogDeliveryEnabled` etiqueta establecida en. `true` AWS asigna esta etiqueta al flujo de entrega de destino cuando configuras el registro.

Este rol vinculado a un servicio también tiene una política de confianza que permite que la entidad principal de servicio `delivery.logs.amazonaws.com` asuma el rol vinculado al servicio necesario. Esta política de confianza es la siguiente:

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

Registro que requiere permisos adicionales [V2]

Algunos AWS servicios utilizan un método nuevo para enviar sus registros. Se trata de un método flexible que le permite configurar la entrega de registros desde estos servicios a uno o más de los siguientes destinos: CloudWatch Logs, Amazon S3 o Firehose.

Una entrega de registros funcional consta de tres elementos:

- A`DeliverySource`, que es un objeto lógico que representa los recursos que realmente envían los registros.
- A`DeliveryDestination`, que es un objeto lógico que representa el destino real de la entrega.
- A`Delivery`, que conecta una fuente de entrega con el destino de entrega

Para configurar la entrega de registros entre un AWS servicio compatible y un destino, debe hacer lo siguiente:

- Cree una fuente de entrega con PutDeliverySource.
- Cree un destino de entrega con PutDeliveryDestination.
- Si va a entregar registros entre cuentas, debe utilizarlos PutDeliveryDestinationPolicyen la cuenta de destino para asignar una IAM política al destino. Esta política autoriza la creación de una entrega desde la fuente de entrega de la cuenta A hasta el destino de la entrega en la cuenta B. En el caso de la entrega entre cuentas, debe crear usted mismo las políticas de permisos manualmente.
- Cree una entrega combinando exactamente una fuente de entrega y un destino de entrega, utilizando. CreateDelivery

En las siguientes secciones, se brindan detalles de los permisos que debe tener al iniciar sesión para configurar la entrega de registros en cada tipo de destino, mediante el proceso V2. Estos permisos se pueden conceder a un rol de IAM con el que haya iniciado sesión.

> **Important**
>
> Es su responsabilidad eliminar los recursos de entrega de registros después de eliminar el recurso generador de registros. Para ello, siga estos pasos.
>
> 1. Elimine el `Delivery` mediante la DeleteDeliveryoperación.
> 2. Elimine el `DeliverySource` mediante la DeleteDeliverySourceoperación.
> 3. Si el `DeliveryDestination` elemento asociado al `DeliverySource` que acaba de eliminar se usa solo para este `DeliverySource` propósito específico, puede eliminarlo mediante la DeleteDeliveryDestinationsoperación.

Contenido

Los registros se envían a CloudWatch Logs

Permisos de usuario

Para habilitar el envío de CloudWatch registros a Logs, debe iniciar sesión con los siguientes permisos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
```

```
        {
            "Sid": "AllowUpdatesToResourcePolicyCWL",
            "Effect": "Allow",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:*"
            ]
        }
    ]
}
```

Política de recursos del grupo de registros

El grupo de registros al que se envían los registros debe tener una política de recursos que incluya determinados permisos. Si el grupo de registros actualmente no tiene una política de recursos y el usuario que configura el registro tiene los `logs:PutResourcePolicy` `logs:DescribeLogGroups` permisos y los permisos para el grupo de registros, crea AWS automáticamente la siguiente política para él cuando comience a enviar los CloudWatch registros a Logs. `logs:DescribeResourcePolicies`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "delivery.logs.amazonaws.com"
        ]
      },
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:0123456789:log-group:my-log-group:log-stream:*"
      ],
      "Condition": {
```

```
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

Consideraciones de límite de tamaño de la política de recursos del grupo de registros

Estos servicios deben enumerar cada grupo de registros al que envían registros en la política de recursos, y las políticas de recursos de CloudWatch Logs están limitadas a 5120 caracteres. Un servicio que envía registros a un gran número de grupos de registros podría alcanzar este límite.

Para mitigar esta situación, CloudWatch Logs supervisa el tamaño de las políticas de recursos utilizadas por el servicio que envía los registros y, cuando detecta que una política se acerca al límite de tamaño de 5120 caracteres, CloudWatch Logs activa `/aws/vendedlogs/*` automáticamente la política de recursos de ese servicio. Puede comenzar a utilizar grupos de registros con nombres que comiencen por `/aws/vendedlogs/` como los destinos de los registros de estos servicios.

Registros enviados a Amazon S3

Permisos de usuario

Para habilitar el envío de registros a Amazon S3, debe iniciar sesión con los siguientes permisos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
```

```
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyS3",
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        }
    ]
}
```

El bucket de S3 al que se envían los registros debe tener una política de recursos que incluya determinados permisos. Si el bucket no tiene actualmente una política de recursos y el usuario que configura el registro tiene los `S3:PutBucketPolicy` permisos `S3:GetBucketPolicy` y para el bucket, creará AWS automáticamente la siguiente política para él cuando comience a enviar los registros a Amazon S3.

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

```
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
                },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::my-bucket",
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source*"]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-
source:*"]
                }
            }
        }
    ]
}
```

En la política anterior, para `aws:SourceAccount`, especifique la lista de ID de cuenta para los que se entregan los registros a este bucket. Para `aws:SourceArn`, especifique la lista de ARN del

recurso que genera los registros, en el formulario `arn:aws:logs:source-region:source-account-id:*`.

Si el bucket tiene una política de recursos, pero esa política no contiene la instrucción que se muestra en la política anterior, y el usuario que configura el registro tiene los permisos `S3:GetBucketPolicy` y `S3:PutBucketPolicy` para el bucket, esa instrucción se anexa a la política de recursos del bucket.

> **Note**
>
> En algunos casos, es posible que veas `AccessDenied` errores al AWS CloudTrail indicar si no se ha concedido el `s3:ListBucket` permiso a`delivery.logs.amazonaws.com`. Para evitar estos errores en sus CloudTrail registros, debe conceder el `s3:ListBucket` permiso `delivery.logs.amazonaws.com` e incluir `Condition` los parámetros que se muestran con el conjunto de `s3:GetBucketAcl` permisos en la política de bucket anterior. Para simplificar esto, en lugar de crear una nueva `Statement`, puede actualizar directamente `AWSLogDeliveryAclCheck` para que sea `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Uso de cifrado en el lado de servidor del bucket de Amazon S3

Puede proteger los datos de su bucket de Amazon S3 habilitando el cifrado del lado del servidor con claves administradas por Amazon S3 (SSE-S3) o el cifrado del lado del servidor con una clave almacenada en (SSE-KMS). AWS KMS AWS Key Management Service Para obtener más información, consulte Protección de los datos con el cifrado del lado del servidor.

Si elige SSE-S3, no se requiere ninguna configuración adicional. Amazon S3 se encarga de la clave de cifrado.

> **Warning**
>
> Si elige SSE-KMS, debe usar una clave administrada por el cliente, ya que no se admite el uso de una clave administrada en este escenario. AWS Si configura el cifrado con una clave AWS administrada, los registros se entregarán en un formato ilegible.

Cuando utilizas una AWS KMS clave gestionada por el cliente, puedes especificar el nombre de recurso de Amazon (ARN) de la clave gestionada por el cliente al activar el cifrado de buckets. Debe

agregar lo siguiente a la política de clave para la clave administrada por el cliente (no a la política del bucket para el bucket de S3), de modo que la cuenta de entrega de registros pueda escribir en el bucket de S3.

Si elige SSE-KMS, debe usar una clave administrada por el cliente, ya que no se admite el uso de una clave AWS administrada en este escenario. Cuando utilizas una AWS KMS clave gestionada por el cliente, puedes especificar el nombre de recurso de Amazon (ARN) de la clave gestionada por el cliente al activar el cifrado de buckets. Debe agregar lo siguiente a la política de clave para la clave administrada por el cliente (no a la política del bucket para el bucket de S3), de modo que la cuenta de entrega de registros pueda escribir en el bucket de S3.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source:*"]
        }
        }
}
```

Para `aws:SourceAccount`, especifique la lista de ID de cuenta para los que se entregan los registros a este bucket. Para `aws:SourceArn`, especifique la lista de ARN del recurso que genera los registros, en el formulario `arn:aws:logs:`*source-region*`:`*source-account-id*`:*`.

Registros enviados a Firehose

Permisos de usuario

Para habilitar el envío de registros a Firehose, debes iniciar sesión con los siguientes permisos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "AllowUpdatesToResourcePolicyFH",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": [
                "arn:aws:firehose:region:account-id:deliverystream/*"
            ]
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        }
    ]
}
```

Roles de IAM utilizados para permisos de recursos

Como Firehose no usa políticas de recursos, AWS usa roles de IAM al configurar estos registros para enviarlos a Firehose. AWS crea un rol vinculado a un servicio denominado. AWSServiceRoleForLogDelivery Este rol vinculado a un servicio incluye los siguientes permisos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
```

```
        }
    ]
}
```

Esta función vinculada al servicio concede permisos para todas las transmisiones de entrega de Firehose que tengan la `LogDeliveryEnabled` etiqueta establecida en. `true` AWS asigna esta etiqueta al flujo de entrega de destino cuando configuras el registro.

Este rol vinculado a un servicio también tiene una política de confianza que permite que la entidad principal de servicio `delivery.logs.amazonaws.com` asuma el rol vinculado al servicio necesario. Esta política de confianza es la siguiente:

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

Permisos específicos del servicio

Además de los permisos específicos del destino enumerados en las secciones anteriores, algunos servicios requieren una autorización explícita para que los clientes puedan enviar registros desde sus recursos, como medida de seguridad adicional. Autoriza la `AllowVendedLogDeliveryForResource` acción para los recursos que venden registros dentro de ese servicio. Para estos servicios, utilice la siguiente política y sustituya el *`servicio`* y el *`tipo de recurso`* por los valores adecuados. Para ver los valores específicos de cada servicio para estos campos, consulte la página de documentación de dichos servicios para ver los registros vendidos.

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

```
            "Effect": "Allow",
            "Action": [
                "service:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:service:region:account-id:resource-type/*"
        }
    ]
}
```

Permisos específicos de la consola

Además de los permisos enumerados en las secciones anteriores, si configura la entrega de registros mediante la consola en lugar de las API, también necesitará los siguientes permisos adicionales:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogDeliveryActionsConsoleCWL",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleS3",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleFH",
            "Effect": "Allow",
```

```
            "Action": [
                "firehose:ListDeliveryStreams",
                "firehose:DescribeDeliveryStream"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Se recomienda utilizar las claves de contexto de condición `aws:SourceOrgPaths`global `aws:SourceArnaws:SourceAccountaws:SourceOrgID`, y las claves de contexto de condición global en las políticas de recursos para limitar los permisos que CloudWatch Logs concede a otro servicio al recurso. Utilice `aws:SourceArn` para asociar solo un recurso al acceso entre servicios. Utilice `aws:SourceAccount` para permitir que cualquier recurso de esa cuenta se asocie al uso entre servicios. Utilice `aws:SourceOrgID` para permitir que cualquier recurso de cuentas dentro de una organización se asocie al uso entre servicios. Utilice `aws:SourceOrgPaths` para asociar cualquier recurso de cuentas dentro de una ruta de AWS Organizations al uso entre servicios. Para obtener más información sobre el uso y la comprensión de las rutas, consulte Comprender la ruta de la AWS Organizations entidad.

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. Si no conoce el ARN completo del recurso o si está especificando varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con caracteres comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Si el valor de `aws:SourceArn` no contiene el ID de cuenta, como un ARN de bucket de Amazon S3, debe utilizar `aws:SourceAccount` y `aws:SourceArn` para limitar los permisos.

Para protegerse contra el problema del suplente confuso a gran escala, utilice la clave de contexto de condición global `aws:SourceOrgID` o `aws:SourceOrgPaths`con el identificador de organización o la ruta de organización del recurso en sus políticas basadas en recursos. Las políticas que incluyan la clave `aws:SourceOrgID` o `aws:SourceOrgPaths` incluirán automáticamente las cuentas correctas y no requerirán una actualización manual cuando se agregan, quitan o mueven cuentas en la organización.

Las políticas de las secciones anteriores de esta página muestran cómo se puede utilizar las claves del contexto de condición global `aws:SourceArn` y `aws:SourceAccount` para evitar el problema suplente confuso.

CloudWatch Registra las actualizaciones de las políticas AWS gestionadas

Consulta los detalles sobre las actualizaciones de las políticas AWS administradas para CloudWatch los registros desde que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbase a la fuente RSS de la página del historial de documentos de CloudWatch registro.

Cambio	Descripción	Fecha
AWSServiceRoleForLogDelivery política de funciones vinculadas al servicio: se actualiza a una política existente	CloudWatch Los registros cambiaron los permisos de la política de IAM asociados al rol vinculado al AWSServiceRoleForLogDeliveryservicio. Se realizó el siguiente cambio: • La clave de condición `firehose:ResourceTag/LogDeliveryEnabled": "true"` se cambió a `aws:ResourceTag/Lo`	15 de julio de 2021

Cambio	Descripción	Fecha
	`gDeliveryEnabled": "true"`.	
CloudWatch Los registros empezaron a registrar los cambios	CloudWatch Los registros empezaron a registrar los cambios de sus políticas AWS gestionadas.	10 de junio de 2021

Exportación de datos de registro a Simple Storage Service (Amazon S3)

Exporte los datos de registro desde sus grupos de registro a un bucket de Amazon S3 y utilice estos datos en el procesamiento y análisis personalizados o para cargarlos en otros sistemas. Puede exportar los datos en un bucket en la misma cuenta o en una diferente.

Puede hacer lo siguiente:

- Exporte los datos de registro a depósitos de S3 cifrados por SSE-KMS en () AWS Key Management Service AWS KMS
- Exportación de datos de registro a buckets de S3 que cuentan con bloqueo de objetos de S3 habilitado con un periodo de retención

> **Note**
>
> La exportación a Amazon S3 solo se admite para grupos de registros de la clase de registro estándar. Para obtener más información sobre las clases de registro, consulteClases de registro.

Para iniciar el proceso de exportación, debe crear un bucket de S3 para almacenar los datos de registro exportados. Puede almacenar los archivos exportados en su bucket de S3 y definir reglas de ciclo de vida de Amazon S3 para archivar o eliminar archivos exportados automáticamente.

Puede exportar a buckets de S3 que están cifrados con AES-256 o con SSE-KMS. No se admite la exportación a buckets que están cifrados con DSSE-KMS.

Puede exportar los registros de varios grupos de registro o varios intervalos de tiempo en el mismo bucket de S3. Para separar los datos de registro de cada tarea de exportación, puede especificar un prefijo que se utilizará como prefijo de clave de Amazon S3 para todos los objetos exportados.

> **Note**
>
> No se garantiza la ordenación basada en el tiempo de los fragmentos de datos de registro dentro de un archivo exportado. Puede ordenar los datos del campo de registro exportados

mediante las utilidades de Linux. Por ejemplo, el siguiente comando de utilidad ordena los eventos de todos los archivos de `.gz` una sola carpeta.

```
find . -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
```

El siguiente comando de utilidad ordena los archivos.gz de varias subcarpetas.

```
find ./*/ -type f -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
```

Además, puede utilizar otro comando `stdout` para canalizar la salida ordenada a otro archivo para guardarla.

Los datos de registro pueden tardar hasta 12 horas en estar disponibles para la exportación. El tiempo de espera de las tareas de exportación se agota tras 24 horas. Si se agota el tiempo de espera de las tareas de exportación, reduzca el intervalo de tiempo al crear la tarea de exportación.

Para el análisis casi en tiempo real de datos de registro, consulte Análisis de datos de registro con CloudWatch Logs Insights o Procesamiento en tiempo real de datos de registros con suscripciones en su lugar.

Contenido

Conceptos

Antes de comenzar, conviene familiarizarse con los siguientes conceptos de exportación:

nombre de grupo de registro

El nombre del grupo de registro asociado a la tarea de exportación. Los datos de registro de este grupo de registro se exportarán al bucket de S3 especificado.

desde (marca temporal)

Una marca temporal necesaria expresada como el número de milisegundos desde el 1 de enero de 1970 00:00:00 UTC. Se exportarán todos los eventos de registro del grupo de registros que se hayan ingerido durante este tiempo o después.

hasta (marca temporal)

Una marca temporal necesaria expresada como el número de milisegundos desde el 1 de enero de 1970 00:00:00 UTC. Se exportarán todos los eventos de registro en el grupo de registro recibidos antes de este momento.

bucket de destino

El nombre del bucket de S3 asociado a la tarea de exportación. Este bucket se utiliza para exportar los datos de registro desde el grupo de registro especificado.

prefijo de destino

Un atributo opcional que se utiliza como prefijo de clave de Amazon S3 para todos los objetos exportados. Esto le ayuda a crear una organización similar a carpetas en su bucket.

Exportar datos de registro a Simple Storage Service (Amazon S3) utilizando la consola

En los siguientes ejemplos, utiliza la CloudWatch consola de Amazon para exportar todos los datos de un grupo de CloudWatch registros de Amazon Logs denominado `my-log-group` a un bucket de Amazon S3 denominado`my-exported-logs`.

Se admite la exportación de datos de registro a buckets de S3 cifrados por SSE-KMS. No se admite la exportación a buckets que están cifrados con DSSE-KMS.

Los detalles de cómo configurar la exportación dependen de si el bucket de Amazon S3 al que desea exportar está en la misma cuenta que los registros que se están exportando o en una diferente.

Temas

Exportación en la misma cuenta

Si el bucket de Amazon S3 está en la misma cuenta que los registros que se exportan, siga las instrucciones de esta sección.

Temas

Paso 1: Crear un bucket de Amazon S3

Le recomendamos que utilice un bucket creado específicamente para CloudWatch Logs. Sin embargo, si desea utilizar un bucket existente, puede pasar al paso 2.

> **Note**
>
> El depósito de S3 debe residir en la misma región que los datos de registro que se van a exportar. CloudWatch Los registros no admiten la exportación de datos a depósitos de S3 en una región diferente.

Para crear un bucket de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. De ser necesario, cambie la región. En la barra de navegación, elige la región en la que residen tus CloudWatch registros.
3. Seleccione la opción Crear bucket.
4. En Bucket Name (Nombre del bucket), escriba un nombre para el bucket.
5. En Región, selecciona la región en la que residen CloudWatch los datos de tus registros.
6. Seleccione Crear.

Paso 2: Configurar los permisos de acceso

Para crear la tarea de exportación del paso 5, tendrá que iniciar sesión con el rol de IAM `AmazonS3ReadOnlyAccess` y con los siguientes permisos:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center .
- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Paso 3: definir permisos en un bucket de S3

De forma predeterminada, los buckets y los objetos de S3 son privados. Solo el propietario del recurso, la Cuenta de AWS que creó el bucket, puede tener acceso a ese bucket y a los objetos que contiene. No obstante, el propietario del recurso puede elegir conceder permisos de acceso a otros recursos y usuarios escribiendo una política de acceso.

Cuando establezca la política, le recomendamos que incluya una cadena generada aleatoriamente como prefijo para el bucket, de manera que solo se exporten al bucket los flujos de registros deseados.

> **Important**
>
> Para que las exportaciones a los buckets de S3 sean más seguras, ahora debe especificar la lista de cuentas de origen que pueden exportar datos de registro a su bucket de S3.
> En el siguiente ejemplo, la lista de identificadores de cuenta de la `aws:SourceAccount` clave serían las cuentas desde las que un usuario puede exportar los datos de registro a su bucket de S3. La clave `aws:SourceArn` sería el recurso para el que se está tomando la acción. Puede restringirla a un grupo de registro específico o utilizar un comodín como se muestra en este ejemplo.
> Le recomendamos que también incluya el ID de cuenta de la cuenta en la que se creó el bucket de S3 para permitir la exportación dentro de la misma cuenta.

Para configurar permisos en un bucket de Amazon S3

1. En la consola de Amazon S3, elija el bucket que creó en el paso 1.
2. Elija Permissions (Permisos), Bucket policy (Política de bucket).
3. En el Bucket Policy Editor (Editor de políticas de bucket), agregue la siguiente política. Cambie `my-exported-logs` por el nombre de su bucket de S3. Asegúrese de especificar el punto de conexión de la región correcta, como `us-west-1`, en Entidad principal.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
```

```
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                    ]
            }
          }
      }
    ]
}
```

4. Elija Save para definir la política que acaba de añadir como política de acceso en su bucket. Esta política permite a CloudWatch Logs exportar datos de registro a su bucket de S3. El propietario del bucket tiene permisos completos en todos los objetos exportados.

> **Warning**
>
> Si el bucket existente ya tiene una o más políticas asociadas, añada las instrucciones para que CloudWatch Logs acceda a esa política o políticas. Le recomendamos que evalúe el conjunto de permisos resultante para asegurarse de que sean adecuados para los usuarios que van a tener acceso al bucket.

(Opcional) Paso 4: exportación a un bucket cifrado con SSE-KMS

Este paso solo es necesario si va a exportar a un bucket de S3 que utilice el cifrado del lado del servidor con. AWS KMS keys Este cifrado se conoce como SSE-KMS.

Para exportar a un bucket cifrado con SSE-KMS

1. Abra la AWS KMS consola en https://console.aws.amazon.com/kms.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.
3. En la barra de navegación izquierda, elija Customer managed keys (Claves administradas por el cliente).

   Elija Create Key (Crear clave).
4. En Key type (Tipo de clave), elija Symmetric (Simétrica).
5. En Key usage (Uso de clave), elija Encrypt and decrypt (Cifrar y descifrar) y, a continuación, elija Next (Siguiente).
6. En Add labels (Agregar etiquetas), introduzca un alias para la clave y, si lo desea, una descripción o etiquetas. A continuación, elija Next.
7. En Key administrators (Administradores de claves), seleccione quién puede administrar esta clave y, a continuación, elija Next (Siguiente).
8. En Define key usage permissions (Definir permisos de uso de claves), no realice cambios y seleccione Next (Siguiente).
9. Revise la configuración y seleccione Finish (Finalizar).
10. En la página Customer managed keys (Claves administradas por el cliente), elija el nombre de la clave que acaba de crear.

11. Elija la sección Key policy (Política de claves) y luego Switch to policy view (Cambiar a la vista de política).
12. En la sección Key policy (Política de claves), elija Edit (Editar).
13. Agregue la siguiente declaración a la lista de declaraciones de políticas de claves. Cuando lo haga, sustituya *Region* (Región) por la región de sus registros y *account-ARN* (ARN de cuenta) por el ARN de la cuenta que posee la clave de KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

14. Elija Guardar cambios.

15. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
16. Encuentre el bucket que creó en Paso 1: crear un bucket de S3 y elija el nombre de bucket.
17. Elija la pestaña Propiedades. Luego, en Default Encryption (Cifrado predeterminado), elija Edit (Editar).
18. En Server-side Encryption (Cifrado del lado del servidor), elija Enable (Habilitar).
19. En Encryption type (Tipo de cifrado), elija AWS Key Management Service key (SSE-KMS) (Clave de KMS [SSE-KMS]).
20. Elige una de tus AWS KMS claves y busca la clave que creaste.
21. En Bucket key (Clave de bucket), seleccione Enable (Habilitar).
22. Elija Guardar cambios.

Paso 5: crear una tarea de exportación

En este paso se crea la tarea de exportación para exportar los registros desde un grupo de registros.

Para exportar datos a Amazon S3 mediante la CloudWatch consola

1. Inicie sesión con los permisos necesarios, tal y como se indica en Paso 2: Configurar los permisos de acceso.
2. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación, seleccione Grupos de registro.
4. En la pantalla Log Groups (Grupos de registros) elija el nombre del grupo de registros.
5. En Actions (Acciones), seleccione Export to Amazon S3 (Exportar datos a Amazon S3).
6. En la pantalla Export data to Amazon S3 (Exportar datos a Amazon S3) , debajo de Define data export (Definir datos para exportar), defina el intervalo de tiempo para los datos a exportar mediante From (Desde) y To (Hasta).
7. Si su grupo de registros tiene varios flujos de registro, puede proporcionar un prefijo de flujo de registro para limitar los datos del grupo de registro a un flujo específico. Elija Advanced (Avanzadas) y, a continuación, en Stream prefix (Prefijo del flujo), escriba el prefijo del flujo de registros.
8. En Choose S3 bucket (Elegir bucket de S3), elija la cuenta asociada con el bucket de S3.
9. En S3 bucket name (Nombre del bucket de S3), elija un bucket de S3.
10. En Export data to (Prefijo del bucket de S3), escriba la cadena generada aleatoriamente que especificó en la política del bucket.

11. Elija Export (Exportar) para exportar los datos de registro a Amazon S3.
12. Para ver el estado de los datos de registro exportados a Amazon S3, elija Actions (Acciones) y luego View all exports to Amazon S3 (Ver todas las exportaciones a Amazon S3).

Exportación entre cuentas

Si el bucket de Amazon S3 está en una cuenta diferente a la de los registros que se exportan, siga las instrucciones de esta sección.

Temas

Paso 1: Crear un bucket de Amazon S3

Le recomendamos que utilice un depósito creado específicamente para CloudWatch los registros. Sin embargo, si desea utilizar un bucket existente, puede pasar al paso 2.

> **Note**
>
> El depósito de S3 debe residir en la misma región que los datos de registro que se van a exportar. CloudWatch Los registros no admiten la exportación de datos a depósitos de S3 en una región diferente.

Para crear un bucket de S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. De ser necesario, cambie la región. En la barra de navegación, elige la región en la que residen tus CloudWatch registros.
3. Seleccione la opción Crear bucket.
4. En Bucket Name (Nombre del bucket), escriba un nombre para el bucket.
5. En Región, selecciona la región en la que residen CloudWatch los datos de tus registros.

6. Seleccione Crear.

Paso 2: Configurar los permisos de acceso

En primer lugar, debe crear una nueva política de IAM para permitir que CloudWatch Logs tenga el `s3:PutObject` permiso para el bucket de Amazon S3 de destino en la cuenta de destino.

La política que cree depende de si el bucket de destino utiliza el AWS KMS cifrado.

Para crear una política de IAM a fin de exportar registros a un bucket de Amazon S3

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la izquierda, elija Políticas.
3. Elija Crear política.
4. En la sección Editor de políticas, elija JSON.
5. Si el depósito de destino no usa AWS KMS cifrado, pegue la siguiente política en el editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Si el depósito de destino utiliza AWS KMS cifrado, pegue la siguiente política en el editor.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-exported-logs/*"
    },
    {
      "Effect": "Allow",
```

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

6. Elija Siguiente.
7. Escriba un nombre para la política. Utilizará este nombre para adjuntar la política a su rol de IAM.
8. Elija Crear política para guardar la nueva política.

Para crear la tarea de exportación del paso 5, tendrá que iniciar sesión con el rol de IAM `AmazonS3ReadOnlyAccess`. También debe iniciar sesión con la política de IAM que acaba de crear y los siguientes permisos:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center .
- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.

- (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Paso 3: definir permisos en un bucket de S3

De forma predeterminada, los buckets y los objetos de S3 son privados. Solo el propietario del recurso, la Cuenta de AWS que creó el bucket, puede tener acceso a ese bucket y a los objetos que contiene. No obstante, el propietario del recurso puede elegir conceder permisos de acceso a otros recursos y usuarios escribiendo una política de acceso.

Cuando establezca la política, le recomendamos que incluya una cadena generada aleatoriamente como prefijo para el bucket, de manera que solo se exporten al bucket los flujos de registros deseados.

> **Important**
>
> Para que las exportaciones a los buckets de S3 sean más seguras, ahora debe especificar la lista de cuentas de origen que pueden exportar datos de registro a su bucket de S3.
> En el siguiente ejemplo, la lista de identificadores de cuenta de la `aws:SourceAccount` clave serían las cuentas desde las que un usuario puede exportar los datos de registro a su bucket de S3. La clave `aws:SourceArn` sería el recurso para el que se está tomando la acción. Puede restringirla a un grupo de registro específico o utilizar un comodín como se muestra en este ejemplo.
> Le recomendamos que también incluya el ID de cuenta de la cuenta en la que se creó el bucket de S3 para permitir la exportación dentro de la misma cuenta.

Para configurar permisos en un bucket de Amazon S3

1. En la consola de Amazon S3, elija el bucket que creó en el paso 1.
2. Elija Permissions (Permisos), Bucket policy (Política de bucket).
3. En el Bucket Policy Editor (Editor de políticas de bucket), agregue la siguiente política. Cambie `my-exported-logs` por el nombre de su bucket de S3. Asegúrese de especificar el punto de conexión de la región correcta, como `us-west-1`, en Entidad principal.

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

```
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                    ]
            }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
              "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control"
              }
            }
        }
    ]
}
```

4. Elija Save para definir la política que acaba de añadir como política de acceso en su bucket. Esta política permite a CloudWatch Logs exportar datos de registro a su bucket de S3. El propietario del bucket tiene permisos completos en todos los objetos exportados.

> **Warning**
>
> Si el bucket existente ya tiene una o más políticas asociadas, añada las instrucciones para que CloudWatch Logs acceda a esa política o políticas. Le recomendamos que evalúe el conjunto de permisos resultante para asegurarse de que sean adecuados para los usuarios que van a tener acceso al bucket.

(Opcional) Paso 4: exportación a un bucket cifrado con SSE-KMS

Este paso solo es necesario si va a exportar a un bucket de S3 que utilice el cifrado del lado del servidor con. AWS KMS keys Este cifrado se conoce como SSE-KMS.

Para exportar a un bucket cifrado con SSE-KMS

1. Abra la AWS KMS consola en https://console.aws.amazon.com/kms.
2. Para cambiarla Región de AWS, usa el selector de regiones en la esquina superior derecha de la página.

3. En la barra de navegación izquierda, elija Customer managed keys (Claves administradas por el cliente).

   Elija Create Key (Crear clave).
4. En Key type (Tipo de clave), elija Symmetric (Simétrica).
5. En Key usage (Uso de clave), elija Encrypt and decrypt (Cifrar y descifrar) y, a continuación, elija Next (Siguiente).
6. En Add labels (Agregar etiquetas), introduzca un alias para la clave y, si lo desea, una descripción o etiquetas. A continuación, elija Next.
7. En Key administrators (Administradores de claves), seleccione quién puede administrar esta clave y, a continuación, elija Next (Siguiente).
8. En Define key usage permissions (Definir permisos de uso de claves), no realice cambios y seleccione Next (Siguiente).
9. Revise la configuración y seleccione Finish (Finalizar).
10. En la página Customer managed keys (Claves administradas por el cliente), elija el nombre de la clave que acaba de crear.
11. Elija la sección Key policy (Política de claves) y luego Switch to policy view (Cambiar a la vista de política).
12. En la sección Key policy (Política de claves), elija Edit (Editar).
13. Agregue la siguiente declaración a la lista de declaraciones de políticas de claves. Cuando lo haga, sustituya *Region* (Región) por la región de sus registros y *account-ARN* (ARN de cuenta) por el ARN de la cuenta que posee la clave de KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

14. Elija Guardar cambios.
15. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
16. Encuentre el bucket que creó en Paso 1: crear un bucket de S3 y elija el nombre de bucket.
17. Elija la pestaña Propiedades. Luego, en Default Encryption (Cifrado predeterminado), elija Edit (Editar).
18. En Server-side Encryption (Cifrado del lado del servidor), elija Enable (Habilitar).
19. En Encryption type (Tipo de cifrado), elija AWS Key Management Service key (SSE-KMS) (Clave de KMS [SSE-KMS]).

20. Elige una de tus AWS KMS claves y busca la clave que creaste.
21. En Bucket key (Clave de bucket), seleccione Enable (Habilitar).
22. Elija Guardar cambios.

Paso 5: crear una tarea de exportación

En este paso se crea la tarea de exportación para exportar los registros desde un grupo de registros.

Para exportar datos a Amazon S3 mediante la CloudWatch consola

1. Inicie sesión con los permisos necesarios, tal y como se indica en Paso 2: Configurar los permisos de acceso.
2. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación, seleccione Grupos de registro.
4. En la pantalla Log Groups (Grupos de registros) elija el nombre del grupo de registros.
5. En Actions (Acciones), seleccione Export to Amazon S3 (Exportar datos a Amazon S3).
6. En la pantalla Export data to Amazon S3 (Exportar datos a Amazon S3) , debajo de Define data export (Definir datos para exportar), defina el intervalo de tiempo para los datos a exportar mediante From (Desde) y To (Hasta).
7. Si su grupo de registros tiene varios flujos de registro, puede proporcionar un prefijo de flujo de registro para limitar los datos del grupo de registro a un flujo específico. Elija Advanced (Avanzadas) y, a continuación, en Stream prefix (Prefijo del flujo), escriba el prefijo del flujo de registros.
8. En Choose S3 bucket (Elegir bucket de S3), elija la cuenta asociada con el bucket de S3.
9. En S3 bucket name (Nombre del bucket de S3), elija un bucket de S3.
10. En Export data to (Prefijo del bucket de S3), escriba la cadena generada aleatoriamente que especificó en la política del bucket.
11. Elija Export (Exportar) para exportar los datos de registro a Amazon S3.
12. Para ver el estado de los datos de registro exportados a Amazon S3, elija Actions (Acciones) y luego View all exports to Amazon S3 (Ver todas las exportaciones a Amazon S3).

Exporte los datos de registro a Amazon S3 mediante AWS CLI

En el siguiente ejemplo, utiliza una tarea de exportación para exportar todos los datos de un grupo de CloudWatch registros denominado Logs `my-log-group` a un bucket de Amazon S3 denominado`my-exported-logs`. En este ejemplo se presupone que ya ha creado un grupo denominado `my-log-group`.

Se admite la exportación de datos de registro a buckets de S3 cifrados mediante AWS KMS . No se admite la exportación a buckets que están cifrados con DSSE-KMS.

Los detalles de cómo configurar la exportación dependen de si el bucket de Amazon S3 al que desea exportar está en la misma cuenta que los registros que se están exportando o en una diferente.

Temas

Exportación en la misma cuenta

Si el bucket de Amazon S3 está en la misma cuenta que los registros que se exportan, siga las instrucciones de esta sección.

Temas

Paso 1: crear un bucket de S3

Le recomendamos que utilice un depósito creado específicamente para CloudWatch los registros. Sin embargo, si desea utilizar un bucket existente, puede pasar al paso 2.

> **Note**
>
> El depósito de S3 debe residir en la misma región que los datos de registro que se van a exportar. CloudWatch Los registros no admiten la exportación de datos a depósitos de S3 en una región diferente.

Para crear un bucket de S3 mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando create-bucket, donde `LocationConstraint` es la región en la que se exportan los datos de registro.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "Location": "/my-exported-logs"
}
```

Paso 2: Configurar los permisos de acceso

Para crear la tarea de exportación del paso 5, tendrá que iniciar sesión con el rol de IAM `AmazonS3ReadOnlyAccess` y con los siguientes permisos:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center .

- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Paso 3: definir permisos en un bucket de S3

De forma predeterminada, los buckets y los objetos de S3 son privados. Solo el propietario del recurso y la cuenta que creó el bucket pueden tener acceso a ese bucket y a los objetos que contiene. No obstante, el propietario del recurso puede elegir conceder permisos de acceso a otros recursos y usuarios escribiendo una política de acceso.

> **Important**
>
> Para que las exportaciones a los buckets de S3 sean más seguras, ahora debe especificar la lista de cuentas de origen que pueden exportar datos de registro a su bucket de S3.
> En el siguiente ejemplo, la lista de identificadores de cuenta de la `aws:SourceAccount` clave serían las cuentas desde las que un usuario puede exportar los datos de registro a su bucket de S3. La clave `aws:SourceArn` sería el recurso para el que se está tomando la acción. Puede restringirla a un grupo de registro específico o utilizar un comodín como se muestra en este ejemplo.
> Le recomendamos que también incluya el ID de cuenta de la cuenta en la que se creó el bucket de S3 para permitir la exportación dentro de la misma cuenta.

Para definir permisos en un bucket de S3

1. Cree un archivo denominado `policy.json` y agregue la siguiente política de acceso, cambiando `my-exported-logs` por el nombre de su bucket de S3 y `Principal` por el punto de conexión de la región a la que va a exportar los datos de registro, como `us-west-1`. Utilice un editor de texto para crear este archivo de política. No utilice la consola de IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
```

```
                        ...
                    ]
                }
            }
        }
    ]
}
```

2. Defina la política que acaba de añadir como política de acceso a su bucket mediante el put-bucket-policycomando. Esta política permite a CloudWatch Logs exportar los datos de registro a su bucket de S3. El propietario del bucket tendrá permisos completos en todos los objetos exportados.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

> **Warning**
>
> Si el bucket existente ya tiene una o más políticas asociadas, añada las instrucciones para que CloudWatch Logs acceda a esa política o políticas. Le recomendamos que evalúe el conjunto de permisos resultante para asegurarse de que sean adecuados para los usuarios que van a tener acceso al bucket.

(Opcional) Paso 4: exportación a un bucket cifrado con SSE-KMS

Este paso solo es necesario si va a exportar a un bucket de S3 que utilice el cifrado del lado del servidor con. AWS KMS keys Este cifrado se conoce como SSE-KMS.

Para exportar a un bucket cifrado con SSE-KMS

1. Utilice un editor de texto para crear un archivo denominado `key_policy.json` y agregue la siguiente política de acceso. Al agregar la política, realice los siguientes cambios:

   - Sustituya *`Region`* (Región) por la región de sus registros.
   - Sustituya *`account-ARN`* (ARN de cuenta) por el ARN de la cuenta propietaria de la clave de KMS.

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

```
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

2. Escriba el siguiente comando:

```
aws kms create-key --policy file://key_policy.json
```

A continuación, se muestra un ejemplo de salida de este comando:

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
```

```
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. Utilice un editor de texto para crear un archivo denominado `bucketencryption.json` con los siguientes contenidos.

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Ingrese el siguiente comando, reemplazando *bucket-name* por el nombre del bucket al que exportará los registros.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

Si el comando no devuelve ningún error, el proceso se ha realizado correctamente.

Paso 5: crear una tarea de exportación

Utilice el siguiente comando para crear la política. Después de crearla, la tarea de exportación podría llevar de unos segundos a unas horas, en función del tamaño de los datos que se van a exportar.

Para exportar datos a Amazon S3 mediante AWS CLI

1. Inicie sesión con los permisos necesarios, tal y como se indica en Paso 2: Configurar los permisos de acceso.
2. En una línea de comandos, utilice el siguiente create-export-taskcomando para crear la tarea de exportación.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-
group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --
to 1441494000000 --destination "my-exported-logs" --destination-prefix "export-
task-output"
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Exportación entre cuentas

Si el bucket de Amazon S3 está en una cuenta diferente a la de los registros que se exportan, siga las instrucciones de esta sección.

Temas

- Paso 1: crear un bucket de S3
- Paso 2: Configurar los permisos de acceso
- Paso 3: definir permisos en un bucket de S3
- (Opcional) Paso 4: exportación a un bucket cifrado con SSE-KMS
- Paso 5: crear una tarea de exportación

Paso 1: crear un bucket de S3

Le recomendamos que utilice un depósito creado específicamente para CloudWatch los registros. Sin embargo, si desea utilizar un bucket existente, puede pasar al paso 2.

> **Note**
>
> El depósito de S3 debe residir en la misma región que los datos de registro que se van a exportar. CloudWatch Los registros no admiten la exportación de datos a depósitos de S3 en una región diferente.

Para crear un bucket de S3 mediante el AWS CLI

En el símbolo del sistema, ejecute el siguiente comando create-bucket, donde `LocationConstraint` es la región en la que se exportan los datos de registro.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "Location": "/my-exported-logs"
}
```

Paso 2: Configurar los permisos de acceso

En primer lugar, debe crear una nueva política de IAM para permitir que CloudWatch Logs tenga el `s3:PutObject` permiso para el bucket Amazon S3 de destino.

Para crear la tarea de exportación del paso 5, tendrá que iniciar sesión con el rol de IAM `AmazonS3ReadOnlyAccess` y con otros permisos determinados. Puede crear una política que contenga algunos de estos otros permisos necesarios.

La política que cree depende de si el bucket de destino utiliza AWS KMS cifrado. Si no utiliza el AWS KMS cifrado, cree una política con el siguiente contenido.

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Si el depósito de destino usa AWS KMS cifrado, cree una política con el siguiente contenido.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

Para crear la tarea de exportación del paso 5, debe iniciar sesión con el rol de IAM `AmazonS3ReadOnlyAccess`, la política de IAM que acaba de crear y también con los siguientes permisos:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center .

- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.

- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Paso 3: definir permisos en un bucket de S3

De forma predeterminada, los buckets y los objetos de S3 son privados. Solo el propietario del recurso y la cuenta que creó el bucket pueden tener acceso a ese bucket y a los objetos que contiene. No obstante, el propietario del recurso puede elegir conceder permisos de acceso a otros recursos y usuarios escribiendo una política de acceso.

> **Important**
>
> Para que las exportaciones a los buckets de S3 sean más seguras, ahora debe especificar la lista de cuentas de origen que pueden exportar datos de registro a su bucket de S3.
> En el siguiente ejemplo, la lista de identificadores de cuenta de la `aws:SourceAccount` clave serían las cuentas desde las que un usuario puede exportar los datos de registro a su bucket de S3. La clave `aws:SourceArn` sería el recurso para el que se está tomando la acción. Puede restringirla a un grupo de registro específico o utilizar un comodín como se muestra en este ejemplo.
> Le recomendamos que también incluya el ID de cuenta de la cuenta en la que se creó el bucket de S3 para permitir la exportación dentro de la misma cuenta.

Para definir permisos en un bucket de S3

1. Cree un archivo denominado `policy.json` y agregue la siguiente política de acceso, cambiando `my-exported-logs` por el nombre de su bucket de S3 y `Principal` por el punto de conexión de la región a la que va a exportar los datos de registro, como `us-west-1`. Utilice un editor de texto para crear este archivo de política. No utilice la consola de IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
```

```
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                    ]
            }
        }
      },
      {
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
          },
          "Action": "s3:PutObject",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control"
            }
          }
        }
    ]
}
```

2. Defina la política que acaba de añadir como política de acceso a su bucket mediante el put-bucket-policycomando. Esta política permite a CloudWatch Logs exportar los datos de registro a su bucket de S3. El propietario del bucket tendrá permisos completos en todos los objetos exportados.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

> **Warning**
>
> Si el bucket existente ya tiene una o más políticas asociadas, añada las instrucciones para que CloudWatch Logs acceda a esa política o políticas. Le recomendamos que

evalúe el conjunto de permisos resultante para asegurarse de que sean adecuados para los usuarios que van a tener acceso al bucket.

(Opcional) Paso 4: exportación a un bucket cifrado con SSE-KMS

Este paso solo es necesario si va a exportar a un bucket de S3 que utilice el cifrado del lado del servidor con. AWS KMS keys Este cifrado se conoce como SSE-KMS.

Para exportar a un bucket cifrado con SSE-KMS

1. Utilice un editor de texto para crear un archivo denominado `key_policy.json` y agregue la siguiente política de acceso. Al agregar la política, realice los siguientes cambios:

   - Sustituya *`Region`* (Región) por la región de sus registros.
   - Sustituya *`account-ARN`* (ARN de cuenta) por el ARN de la cuenta propietaria de la clave de KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
```

```
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

2. Escriba el siguiente comando:

```
aws kms create-key --policy file://key_policy.json
```

A continuación, se muestra un ejemplo de salida de este comando:

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
```

```
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. Utilice un editor de texto para crear un archivo denominado `bucketencryption.json` con los siguientes contenidos.

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Ingrese el siguiente comando, reemplazando *`bucket-name`* por el nombre del bucket al que exportará los registros.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

   Si el comando no devuelve ningún error, el proceso se ha realizado correctamente.

Paso 5: crear una tarea de exportación

Utilice el siguiente comando para crear la política. Después de crearla, la tarea de exportación podría llevar de unos segundos a unas horas, en función del tamaño de los datos que se van a exportar.

Para exportar datos a Amazon S3 mediante AWS CLI

1. Inicie sesión con los permisos necesarios, tal y como se indica en Paso 2: Configurar los permisos de acceso.
2. En una línea de comandos, utilice el siguiente create-export-taskcomando para crear la tarea de exportación.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-
group-09-10-2015" --log-group-name "my-log-group" --from 1441490400000 --
to 1441494000000 --destination "my-exported-logs" --destination-prefix "export-
task-output"
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Descripción de tareas de exportación

Después de crear una tarea de exportación, puede obtener el estado actual de la tarea.

Para describir las tareas de exportación mediante el AWS CLI

En una línea de comandos, utilice el siguiente describe-export-taskscomando.

```
aws logs --profile CWLExportUser describe-export-tasks --task-id
 "cda45419-90ea-4db5-9833-aade86253e66"
```

A continuación, se muestra un ejemplo del resultado.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "RUNNING",
         "message": "Started Successfully"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
```

```
      "tTo": 1441494000000
   }]
}
```

Puede utilizar el comando `describe-export-tasks` de tres formas diferentes:

- Sin filtros: enumera todas las tareas de exportación, en orden de creación inverso.
- Filtrar por ID de tarea: muestra la tarea de exportación, si existe, con el ID especificado.
- Filtrar por estado de tarea: muestra las tareas de exportación con el estado especificado.

Por ejemplo, utilice el siguiente comando para filtrar por el estado `FAILED`.

```
aws logs --profile CWLExportUser describe-export-tasks --status-code "FAILED"
```

A continuación, se muestra un ejemplo del resultado.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "completionTime": 1441498600000
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "FAILED",
         "message": "FAILED"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "to": 1441494000000
   }]
}
```

Cancelación de una tarea de exportación

Puede cancelar una tarea de exportación si se encuentra en el estado `PENDING` o `RUNNING`.

Para cancelar una tarea de exportación mediante el AWS CLI

En una línea de comandos, utilice el siguiente cancel-export-taskcomando:

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

Puede usar el describe-export-taskscomando para comprobar que la tarea se ha cancelado correctamente.

Transmisión de datos de CloudWatch registros a Amazon OpenSearch Service

Puedes configurar un grupo de CloudWatch registros de Logs para transmitir los datos que recibe a tu clúster de Amazon OpenSearch Service prácticamente en tiempo real a través de una suscripción a CloudWatch Logs. Para obtener más información, consulte Procesamiento en tiempo real de datos de registros con suscripciones.

> **Note**
>
> La transmisión al OpenSearch servicio solo se admite para los grupos de registros de la clase de registro estándar. Para obtener más información sobre las clases de registro, consulteClases de registro.

En función de la cantidad de datos de registro que se vayan a transmitir, es posible que desee establecer un límite de ejecuciones simultáneas en la función. Para obtener más información, consulte Escalado de funciones de Lambda

> **Note**
>
> La transmisión de grandes cantidades de datos de CloudWatch registros al OpenSearch Servicio puede generar altos cargos por uso. Te recomendamos que crees un presupuesto en la AWS Billing and Cost Management consola. Para obtener más información, consulte Administración de costos con AWS Budgets.

Requisitos previos

Antes de empezar, cree un dominio OpenSearch de servicio. El dominio puede tener acceso público o acceso de VPC, pero no puede modificar el tipo de acceso después de que se cree el dominio. Es posible que desee revisar la configuración del dominio de OpenSearch servicio más adelante y modificar la configuración del clúster en función de la cantidad de datos que procese el clúster. Para obtener instrucciones sobre cómo crear un dominio, consulta Crear dominios OpenSearch de servicio.

Para obtener más información sobre el OpenSearch Servicio, consulta la Guía para desarrolladores OpenSearch de Amazon Service.

Suscriba un grupo de registro a OpenSearch Service

Puede usar la CloudWatch consola para suscribir un grupo de registros al OpenSearch Servicio.

Para suscribir un grupo de registros al OpenSearch Servicio

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, seleccione Grupos de registro.
3. Seleccione el nombre del grupo de registro.
4. Selecciona Acciones, Filtros de suscripción o Crear filtro de suscripción a Amazon OpenSearch Service.
5. Elija si desea transmitir a un clúster de esta cuenta u otra cuenta.
   - Si eligió esta cuenta, seleccione el dominio que creó en el paso anterior.
   - Si eligió otra cuenta, proporcione el ARN del dominio y el punto de enlace.
6. Para el rol de ejecución de IAM de Lambda, elija el rol de IAM que Lambda debe usar al ejecutar las llamadas a. OpenSearch

   El rol de IAM que elija deberá cumplir los siguientes requisitos:

   - Debo tener `lambda.amazonaws.com` en la relación de confianza.
   - Debe incluir la política siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:region:account-id:domain/target-domain-name/
*"
        }
    ]
}
```

- Si el dominio de OpenSearch servicio de destino usa el acceso a la VPC, el rol debe tener la AWSLambdaVPCAccessExecutionRolepolítica adjunta. Esta política gestionada por Amazon concede a Lambda acceso a la VPC del cliente, lo que permite a Lambda escribir en el punto final de la VPC. OpenSearch

7. En Log format (Formato de registro), elija un formato de registro.
8. En Subscription filter pattern (Patrón de filtro de suscripción), escriba los términos o el patrón que desea buscar en los eventos de registro. Esto garantiza que solo envíe a su clúster los datos que le interesan. OpenSearch Para obtener más información, consulte Creación de métricas a partir de eventos de registro mediante filtros.
9. (Opcional) En Select log data to test (Seleccionar datos de registro para probar), seleccione un flujo de registros y, a continuación, elija Test pattern (Patrón de prueba) para verificar que el filtro de búsqueda devuelva los resultados esperados.
10. Elija Start streaming (Comenzar streaming).

Ejemplos de código para CloudWatch registros que utilizan AWS SDK

Los siguientes ejemplos de código muestran cómo usar CloudWatch los registros con un kit de desarrollo de AWS software (SDK).

Las acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Mientras las acciones muestran cómo llamar a las funciones de servicio individuales, es posible ver las acciones en contexto en los escenarios relacionados y en los ejemplos entre servicios.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica llamando a varias funciones dentro del mismo servicio.

Los ejemplos con varios servicios son aplicaciones de muestra que funcionan con varios Servicios de AWS.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de código

- Acciones para los CloudWatch registros que utilizan AWS SDK
  - Úselo AssociateKmsKey con un AWS SDK o CLI
  - Úselo CancelExportTask con un AWS SDK o CLI
  - Úselo CreateExportTask con un AWS SDK o CLI
  - Úselo CreateLogGroup con un AWS SDK o CLI
  - Úselo CreateLogStream con un AWS SDK o CLI
  - Úselo DeleteLogGroup con un AWS SDK o CLI
  - Úselo DeleteSubscriptionFilter con un AWS SDK o CLI
  - Úselo DescribeExportTasks con un AWS SDK o CLI
  - Úselo DescribeLogGroups con un AWS SDK o CLI
  - Úselo DescribeSubscriptionFilters con un AWS SDK o CLI
  - Úselo GetQueryResults con un AWS SDK o CLI
  - Úselo PutSubscriptionFilter con un AWS SDK o CLI

Acciones para los CloudWatch registros que utilizan AWS SDK

Los siguientes ejemplos de código muestran cómo realizar acciones de CloudWatch registro individuales con los AWS SDK. Estos extractos se denominan API de CloudWatch registros y son extractos de código de programas más grandes que deben ejecutarse en su contexto. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Los siguientes ejemplos incluyen solo las acciones que se utilizan con mayor frecuencia. Para obtener una lista completa, consulta la referencia de la API de Amazon CloudWatch Logs.

Ejemplos

- Úselo AssociateKmsKey con un AWS SDK o CLI
- Úselo CancelExportTask con un AWS SDK o CLI
- Úselo CreateExportTask con un AWS SDK o CLI
- Úselo CreateLogGroup con un AWS SDK o CLI
- Úselo CreateLogStream con un AWS SDK o CLI
- Úselo DeleteLogGroup con un AWS SDK o CLI
- Úselo DeleteSubscriptionFilter con un AWS SDK o CLI
- Úselo DescribeExportTasks con un AWS SDK o CLI
- Úselo DescribeLogGroups con un AWS SDK o CLI
- Úselo DescribeSubscriptionFilters con un AWS SDK o CLI
- Úselo GetQueryResults con un AWS SDK o CLI
- Úselo PutSubscriptionFilter con un AWS SDK o CLI
- Úselo StartLiveTail con un AWS SDK o CLI
- Úselo StartQuery con un AWS SDK o CLI

Úselo **`AssociateKmsKey`** con un AWS SDK o CLI

En el siguiente ejemplo de código, se muestra cómo usar `AssociateKmsKey`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to associate an AWS Key Management Service (AWS KMS) key with
    /// an Amazon CloudWatch Logs log group.
    /// </summary>
    public class AssociateKmsKey
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string kmsKeyId = "arn:aws:kms:us-west-2:<account-
number>:key/7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
            string groupName = "cloudwatchlogs-example-loggroup";

            var request = new AssociateKmsKeyRequest
            {
                KmsKeyId = kmsKeyId,
                LogGroupName = groupName,
            };
```

```
            var response = await client.AssociateKmsKeyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully associated KMS key ID:
{kmsKeyId} with log group: {groupName}.");
            }
            else
            {
                Console.WriteLine("Could not make the association between:
{kmsKeyId} and {groupName}.");
            }
        }
    }
```

- Para obtener más información sobre la API, consulta AssociateKmsKeyla Referencia AWS SDK for .NET de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`CancelExportTask`** con un AWS SDK o CLI

En el siguiente ejemplo de código, se muestra cómo usar `CancelExportTask`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    using System;
```

```
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to cancel an Amazon CloudWatch Logs export task.
/// </summary>
public class CancelExportTask
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();
        string taskId = "exampleTaskId";

        var request = new CancelExportTaskRequest
        {
            TaskId = taskId,
        };

        var response = await client.CancelExportTaskAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"{taskId} successfully canceled.");
        }
        else
        {
            Console.WriteLine($"{taskId} could not be canceled.");
        }
    }
}
```

- Para obtener más información sobre la API, consulta CancelExportTaskla Referencia AWS SDK for .NET de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`CreateExportTask`** con un AWS SDK o CLI

En el siguiente ejemplo de código, se muestra cómo usar `CreateExportTask`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Export Task to export the contents of the Amazon
    /// CloudWatch Logs to the specified Amazon Simple Storage Service (Amazon
S3)
    /// bucket.
    /// </summary>
    public class CreateExportTask
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string taskName = "export-task-example";
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string destination = "doc-example-bucket";
            var fromTime = 1437584472382;
```

```
            var toTime = 1437584472833;

            var request = new CreateExportTaskRequest
            {
                From = fromTime,
                To = toTime,
                TaskName = taskName,
                LogGroupName = logGroupName,
                Destination = destination,
            };

            var response = await client.CreateExportTaskAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"The task, {taskName} with ID: " +
                                  $"{response.TaskId} has been created
successfully.");
            }
        }
    }
```

- Para obtener más información sobre la API, consulta CreateExportTaskla Referencia AWS SDK for .NET de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`CreateLogGroup`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `CreateLogGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to create an Amazon CloudWatch Logs log group.
/// </summary>
public class CreateLogGroup
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();

        string logGroupName = "cloudwatchlogs-example-loggroup";

        var request = new CreateLogGroupRequest
        {
            LogGroupName = logGroupName,
        };

        var response = await client.CreateLogGroupAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"Successfully create log group with ID:
{logGroupName}.");
        }
```

```
            else
            {
                Console.WriteLine("Could not create log group.");
            }
        }
    }
```

- Para obtener más información sobre la API, consulta CreateLogGroupla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

El siguiente comando crea un grupo de registro denominado `my-logs`:

```
aws logs create-log-group --log-group-name my-logs
```

- Para obtener más información sobre la API, consulta CreateLogGroupla Referencia de AWS CLI comandos.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import { CreateLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new CreateLogGroupCommand({
    // The name of the log group.
```

```
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obtener más información sobre la API, consulta CreateLogGroupla Referencia AWS SDK for JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`CreateLogStream`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `CreateLogStream`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
```

```
    /// Shows how to create an Amazon CloudWatch Logs stream for a CloudWatch
    /// log group.
    /// </summary>
    public class CreateLogStream
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string logStreamName = "cloudwatchlogs-example-logstream";

            var request = new CreateLogStreamRequest
            {
                LogGroupName = logGroupName,
                LogStreamName = logStreamName,
            };

            var response = await client.CreateLogStreamAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"{logStreamName} successfully created for
{logGroupName}.");
            }
            else
            {
                Console.WriteLine("Could not create stream.");
            }
        }
    }
```

- Para obtener más información sobre la API, consulta CreateLogStreamla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

El siguiente comando crea un flujo de registro denominado `20150601` en el grupo de registro `my-logs`:

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- Para obtener más información sobre la API, consulta CreateLogStreamla Referencia de AWS CLI comandos.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DeleteLogGroup`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DeleteLogGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Uses the Amazon CloudWatch Logs Service to delete an existing
/// CloudWatch Logs log group.
/// </summary>
```

```
    public class DeleteLogGroup
    {
        public static async Task Main()
        {
            var client = new AmazonCloudWatchLogsClient();
            string logGroupName = "cloudwatchlogs-example-loggroup";

            var request = new DeleteLogGroupRequest
            {
                LogGroupName = logGroupName,
            };

            var response = await client.DeleteLogGroupAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully deleted CloudWatch log group,
{logGroupName}.");
            }
        }
    }
```

- Para obtener más información sobre la API, consulta DeleteLogGroupla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

El siguiente comando elimina un grupo de registro denominado `my-logs`:

```
aws logs delete-log-group --log-group-name my-logs
```

- Para obtener más información sobre la API, consulta DeleteLogGroupla Referencia de AWS CLI comandos.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import { DeleteLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obtener más información sobre la API, consulta DeleteLogGroupla Referencia AWS SDK for JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DeleteSubscriptionFilter`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DeleteSubscriptionFilter`.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DeleteSubscriptionFilterRequest.h>
#include <iostream>
```

Elimine el filtro de suscripción.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DeleteSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetLogGroupName(log_group);

        auto outcome = cwl.DeleteSubscriptionFilter(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Failed to delete CloudWatch log subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        } else {
            std::cout << "Successfully deleted CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Para obtener más información sobre la API, consulta DeleteSubscriptionFilterla Referencia AWS SDK for C++ de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DeleteSubscriptionFilterRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <logGroup>

                Where:
                  filter - The name of the subscription filter (for example,
 MyFilter).
                  logGroup - The name of the log group. (for example, testgroup).
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String filter = args[0];
        String logGroup = args[1];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .build();

        deleteSubFilter(logs, filter, logGroup);
        logs.close();
    }

    public static void deleteSubFilter(CloudWatchLogsClient logs, String filter,
 String logGroup) {
        try {
            DeleteSubscriptionFilterRequest request =
 DeleteSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .logGroupName(logGroup)
                    .build();

            logs.deleteSubscriptionFilter(request);
            System.out.printf("Successfully deleted CloudWatch logs subscription
 filter %s", filter);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obtener más información sobre la API, consulta DeleteSubscriptionFilterla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import { DeleteSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteSubscriptionFilterCommand({
    // The name of the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obtener más información sobre la API, consulta DeleteSubscriptionFilterla Referencia AWS SDK for JavaScript de la API.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
```

```
  filterName: "FILTER",
  logGroupName: "LOG_GROUP",
};

cwl.deleteSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener más información sobre la API, consulta DeleteSubscriptionFilterla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun deleteSubFilter(
    filter: String?,
    logGroup: String?,
) {
    val request =
        DeleteSubscriptionFilterRequest {
            filterName = filter
            logGroupName = logGroup
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { logs ->
        logs.deleteSubscriptionFilter(request)
```

```
        println("Successfully deleted CloudWatch logs subscription filter named
 $filter")
    }
}
```

- Para obtener más información sobre la API, consulta DeleteSubscriptionFilterla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DescribeExportTasks`** con un AWS SDK o CLI

En el siguiente ejemplo de código, se muestra cómo usar `DescribeExportTasks`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to retrieve a list of information about Amazon CloudWatch
/// Logs export tasks.
/// </summary>
public class DescribeExportTasks
{
    public static async Task Main()
    {
```

```
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            var request = new DescribeExportTasksRequest
            {
                Limit = 5,
            };

            var response = new DescribeExportTasksResponse();

            do
            {
                response = await client.DescribeExportTasksAsync(request);
                response.ExportTasks.ForEach(t =>
                {
                    Console.WriteLine($"{t.TaskName} with ID: {t.TaskId} has
status: {t.Status}");
                });
            }
            while (response.NextToken is not null);
        }
    }
```

- Para obtener más información sobre la API, consulta DescribeExportTasksla Referencia AWS SDK for .NET de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DescribeLogGroups`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DescribeLogGroups`.

.NET

AWS SDK for .NET

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Retrieves information about existing Amazon CloudWatch Logs log groups
/// and displays the information on the console.
/// </summary>
public class DescribeLogGroups
{
    public static async Task Main()
    {
        // Creates a CloudWatch Logs client using the default
        // user. If you need to work with resources in another
        // AWS Region than the one defined for the default user,
        // pass the AWS Region as a parameter to the client constructor.
        var client = new AmazonCloudWatchLogsClient();

        bool done = false;
        string newToken = null;

        var request = new DescribeLogGroupsRequest
        {
            Limit = 5,
        };

        DescribeLogGroupsResponse response;

        do
        {
            if (newToken is not null)
```

```
                {
                    request.NextToken = newToken;
                }

                response = await client.DescribeLogGroupsAsync(request);

                response.LogGroups.ForEach(lg =>
                {
                    Console.WriteLine($"{lg.LogGroupName} is associated with the
key: {lg.KmsKeyId}.");
                    Console.WriteLine($"Created on:
{lg.CreationTime.Date.Date}");
                    Console.WriteLine($"Date for this group will be stored for:
{lg.RetentionInDays} days.\n");
                });

                if (response.NextToken is null)
                {
                    done = true;
                }
                else
                {
                    newToken = response.NextToken;
                }
            }
            while (!done);
        }
    }
```

- Para obtener más información sobre la API, consulta DescribeLogGroupsla Referencia AWS SDK for .NET de la API.

CLI

AWS CLI

El siguiente comando describe un grupo de registro denominado `my-logs`:

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

Salida:

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- Para obtener más información sobre la API, consulta DescribeLogGroupsla Referencia de AWS CLI comandos.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import {
  paginateDescribeLogGroups,
  CloudWatchLogsClient,
} from "@aws-sdk/client-cloudwatch-logs";

const client = new CloudWatchLogsClient({});

export const main = async () => {
  const paginatedLogGroups = paginateDescribeLogGroups({ client }, {});
  const logGroups = [];

  for await (const page of paginatedLogGroups) {
    if (page.logGroups && page.logGroups.every((lg) => !!lg)) {
```

```
      logGroups.push(...page.logGroups);
    }
  }

  console.log(logGroups);
  return logGroups;
};
```

- Para obtener más información sobre la API, consulta DescribeLogGroupsla Referencia AWS SDK for JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`DescribeSubscriptionFilters`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `DescribeSubscriptionFilters`.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DescribeSubscriptionFiltersRequest.h>
#include <aws/logs/model/DescribeSubscriptionFiltersResult.h>
#include <iostream>
#include <iomanip>
```

Enumere los filtros de suscripción.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DescribeSubscriptionFiltersRequest request;
        request.SetLogGroupName(log_group);
        request.SetLimit(1);

        bool done = false;
        bool header = false;
        while (!done) {
            auto outcome = cwl.DescribeSubscriptionFilters(
                    request);
            if (!outcome.IsSuccess()) {
                std::cout << "Failed to describe CloudWatch subscription filters
"
                    << "for log group " << log_group << ": " <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header) {
                std::cout << std::left << std::setw(32) << "Name" <<
                    std::setw(64) << "FilterPattern" << std::setw(64) <<
                    "DestinationArn" << std::endl;
                header = true;
            }

            const auto &filters = outcome.GetResult().GetSubscriptionFilters();
            for (const auto &filter : filters) {
                std::cout << std::left << std::setw(32) <<
                    filter.GetFilterName() << std::setw(64) <<
                    filter.GetFilterPattern() << std::setw(64) <<
                    filter.GetDestinationArn() << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Para obtener más información sobre la API, consulta DescribeSubscriptionFilters la Referencia AWS SDK for C++ de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersResponse
import software.amazon.awssdk.services.cloudwatchlogs.model.SubscriptionFilter;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeSubscriptionFilters {
    public static void main(String[] args) {

        final String usage = """

                Usage:
                  <logGroup>

                Where:
                  logGroup - A log group name (for example, myloggroup).
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String logGroup = args[0];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        describeFilters(logs, logGroup);
        logs.close();
    }

    public static void describeFilters(CloudWatchLogsClient logs, String
logGroup) {
        try {
            boolean done = false;
            String newToken = null;

            while (!done) {
                DescribeSubscriptionFiltersResponse response;
                if (newToken == null) {
                    DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                            .logGroupName(logGroup)
                            .limit(1).build();

                    response = logs.describeSubscriptionFilters(request);
                } else {
                    DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                            .nextToken(newToken)
                            .logGroupName(logGroup)
                            .limit(1).build();
                    response = logs.describeSubscriptionFilters(request);
                }

                for (SubscriptionFilter filter : response.subscriptionFilters())
{
                    System.out.printf("Retrieved filter with name %s, " +
"pattern %s " + "and destination arn %s",
                            filter.filterName(),
                            filter.filterPattern(),
                            filter.destinationArn());
                }
```

```
                if (response.nextToken() == null) {
                    done = true;
                } else {
                    newToken = response.nextToken();
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.printf("Done");
    }
}
```

- Para obtener más información sobre la API, consulta DescribeSubscriptionFilterslа Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import { DescribeSubscriptionFiltersCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  // This will return a list of all subscription filters in your account
  // matching the log group name.
  const command = new DescribeSubscriptionFiltersCommand({
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
    limit: 1,
  });
```

```
  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obtener más información sobre la API, consulta DescribeSubscriptionFilterslа Referencia AWS SDK for JavaScript de la API.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  logGroupName: "GROUP_NAME",
  limit: 5,
};

cwl.describeSubscriptionFilters(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.subscriptionFilters);
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener más información sobre la API, consulta DescribeSubscriptionFiltersla Referencia AWS SDK for JavaScript de la API.

Kotlin

SDK para Kotlin

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
suspend fun describeFilters(logGroup: String) {
    val request =
        DescribeSubscriptionFiltersRequest {
            logGroupName = logGroup
            limit = 1
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { cwlClient ->
        val response = cwlClient.describeSubscriptionFilters(request)
        response.subscriptionFilters?.forEach { filter ->
            println("Retrieved filter with name  ${filter.filterName} pattern
 ${filter.filterPattern} and destination ${filter.destinationArn}")
        }
    }
}
```

- Para obtener más información sobre la API, consulta DescribeSubscriptionFiltersla referencia sobre el AWS SDK para la API de Kotlin.

Para ver una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consultaUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`GetQueryResults`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `GetQueryResults`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Ejecución de una consulta de gran tamaño

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
```

- Para obtener más información sobre la API, consulta GetQueryResultsla Referencia AWS SDK for JavaScript de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])
```

- Para obtener más información sobre la API, consulta GetQueryResultsla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso CloudWatch de registros con un AWS SDK En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **PutSubscriptionFilter** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `PutSubscriptionFilter`.

C++

SDK para C++

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Incluir los archivos requeridos.

```
#include <aws/core/Aws.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/PutSubscriptionFilterRequest.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Cree el filtro de suscripción.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::PutSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetFilterPattern(filter_pattern);
        request.SetLogGroupName(log_group);
        request.SetDestinationArn(dest_arn);
        auto outcome = cwl.PutSubscriptionFilter(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch logs subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
```

```
        }
```

- Para obtener más información sobre la API, consulta PutSubscriptionFilterla Referencia AWS SDK for C++ de la API.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.PutSubscriptionFilterRequest;

/**
 * Before running this code example, you need to grant permission to CloudWatch
 * Logs the right to execute your Lambda function.
 * To perform this task, you can use this CLI command:
 *
 * aws lambda add-permission --function-name "lamda1" --statement-id "lamda1"
 * --principal "logs.us-west-2.amazonaws.com" --action "lambda:InvokeFunction"
 * --source-arn "arn:aws:logs:us-west-2:111111111111:log-group:testgroup:*"
 * --source-account "111111111111"
 *
 * Make sure you replace the function name with your function name and replace
 * '111111111111' with your account details.
 * For more information, see "Subscription Filters with AWS Lambda" in the
 * Amazon CloudWatch Logs Guide.
 *
 *
 * Also, before running this Java V2 code example,set up your development
 * environment,including your credentials.
```

```
 *
 * For more information,see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 */

public class PutSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <pattern> <logGroup> <functionArn>\s

                Where:
                  filter - A filter name (for example, myfilter).
                  pattern - A filter pattern (for example, ERROR).
                  logGroup - A log group name (testgroup).
                  functionArn - An AWS Lambda function ARN (for example,
 arn:aws:lambda:us-west-2:111111111111:function:lambda1) .
                """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String filter = args[0];
        String pattern = args[1];
        String logGroup = args[2];
        String functionArn = args[3];
        Region region = Region.US_WEST_2;
        CloudWatchLogsClient cwl = CloudWatchLogsClient.builder()
                .region(region)
                .build();

        putSubFilters(cwl, filter, pattern, logGroup, functionArn);
        cwl.close();
    }

    public static void putSubFilters(CloudWatchLogsClient cwl,
            String filter,
            String pattern,
```

```
            String logGroup,
            String functionArn) {

        try {
            PutSubscriptionFilterRequest request =
 PutSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .filterPattern(pattern)
                    .logGroupName(logGroup)
                    .destinationArn(functionArn)
                    .build();

            cwl.putSubscriptionFilter(request);
            System.out.printf(
                    "Successfully created CloudWatch logs subscription filter
 %s",
                    filter);

        } catch (CloudWatchLogsException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obtener más información sobre la API, consulta PutSubscriptionFilterla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import { PutSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";
```

```
const run = async () => {
  const command = new PutSubscriptionFilterCommand({
    // An ARN of a same-account Kinesis stream, Kinesis Firehose
    // delivery stream, or Lambda function.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
SubscriptionFilters.html
    destinationArn: process.env.CLOUDWATCH_LOGS_DESTINATION_ARN,

    // A name for the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,

    // A filter pattern for subscribing to a filtered stream of log events.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
FilterAndPatternSyntax.html
    filterPattern: process.env.CLOUDWATCH_LOGS_FILTER_PATTERN,

    // The name of the log group. Messages in this group matching the filter
 pattern
    // will be sent to the destination ARN.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obtener más información sobre la API, consulta PutSubscriptionFilterla Referencia AWS SDK for JavaScript de la API.

SDK para JavaScript (v2)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  destinationArn: "LAMBDA_FUNCTION_ARN",
  filterName: "FILTER_NAME",
  filterPattern: "ERROR",
  logGroupName: "LOG_GROUP",
};

cwl.putSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Para obtener información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.
- Para obtener más información sobre la API, consulta PutSubscriptionFilterla Referencia AWS SDK for JavaScript de la API.

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`StartLiveTail`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `StartLiveTail`.

.NET

AWS SDK for .NET

Incluir los archivos requeridos.

```
using Amazon;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

Inicie la sesión de Live Tail.

```
            var client = new AmazonCloudWatchLogsClient();
            var request = new StartLiveTailRequest
            {
                LogGroupIdentifiers = logGroupIdentifiers,
                LogStreamNames = logStreamNames,
                LogEventFilterPattern = filterPattern,
            };

            var response = await client.StartLiveTailAsync(request);

            // Catch if request fails
            if (response.HttpStatusCode != System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine("Failed to start live tail session");
                return;
            }
```

Puede controlar los eventos de la sesión de Live Tail de dos maneras:

```
            /* Method 1
            * 1). Asynchronously loop through the event stream
            * 2). Set a timer to dispose the stream and stop the Live Tail
 session at the end.
            */
            var eventStream = response.ResponseStream;
            var task = Task.Run(() =>
            {
                foreach (var item in eventStream)
```

```
                {
                    if (item is LiveTailSessionUpdate liveTailSessionUpdate)
                    {
                        foreach (var sessionResult in
liveTailSessionUpdate.SessionResults)
                        {
                            Console.WriteLine("Message : {0}",
sessionResult.Message);
                        }
                    }
                    if (item is LiveTailSessionStart)
                    {
                        Console.WriteLine("Live Tail session started");
                    }
                    // On-stream exceptions are processed here
                    if (item is CloudWatchLogsEventStreamException)
                    {
                        Console.WriteLine($"ERROR: {item}");
                    }
                }
            });
            // Close the stream to stop the session after a timeout
            if (!task.Wait(TimeSpan.FromSeconds(10))){
                eventStream.Dispose();
                Console.WriteLine("End of line");
            }
```

```
            /* Method 2
            * 1). Add event handlers to each event variable
            * 2). Start processing the stream and wait for a timeout using
AutoResetEvent
            */
            AutoResetEvent endEvent = new AutoResetEvent(false);
            var eventStream = response.ResponseStream;
            using (eventStream) // automatically disposes the stream to stop the
session after execution finishes
            {
                eventStream.SessionStartReceived += (sender, e) =>
                {
                    Console.WriteLine("LiveTail session started");
                };
                eventStream.SessionUpdateReceived += (sender, e) =>
```

```
                {
                    foreach (LiveTailSessionLogEvent logEvent in
 e.EventStreamEvent.SessionResults){
                        Console.WriteLine("Message: {0}", logEvent.Message);
                    }
                };
                // On-stream exceptions are captured here
                eventStream.ExceptionReceived += (sender, e) =>
                {
                    Console.WriteLine($"ERROR:
 {e.EventStreamException.Message}");
                };

                eventStream.StartProcessing();
                // Stream events for this amount of time.
                endEvent.WaitOne(TimeSpan.FromSeconds(10));
                Console.WriteLine("End of line");
            }
```

- Para obtener más información sobre la API, consulte StartLiveTaillla referencia AWS SDK for .NET de la API.

Go

SDK para Go V2

Incluir los archivos requeridos.

```
import (
 "context"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)
```

Gestione los eventos de la sesión de Live Tail.

```
func handleEventStreamAsync(stream *cloudwatchlogs.StartLiveTailEventStream) {
 eventsChan := stream.Events()
 for {
  event := <-eventsChan
  switch e := event.(type) {
  case *types.StartLiveTailResponseStreamMemberSessionStart:
   log.Println("Received SessionStart event")
  case *types.StartLiveTailResponseStreamMemberSessionUpdate:
   for _, logEvent := range e.Value.SessionResults {
    log.Println(*logEvent.Message)
   }
  default:
   // Handle on-stream exceptions
   if err := stream.Err(); err != nil {
    log.Fatalf("Error occured during streaming: %v", err)
   } else if event == nil {
    log.Println("Stream is Closed")
    return
   } else {
    log.Fatalf("Unknown event type: %T", e)
   }
  }
 }
}
```

Inicie la sesión de Live Tail.

```
 cfg, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  panic("configuration error, " + err.Error())
 }
 client := cloudwatchlogs.NewFromConfig(cfg)

 request := &cloudwatchlogs.StartLiveTailInput{
  LogGroupIdentifiers:   logGroupIdentifiers,
  LogStreamNames:        logStreamNames,
  LogEventFilterPattern: logEventFilterPattern,
 }

 response, err := client.StartLiveTail(context.TODO(), request)
 // Handle pre-stream Exceptions
 if err != nil {
```

```
 log.Fatalf("Failed to start streaming: %v", err)
}

// Start a Goroutine to handle events over stream
stream := response.GetStream()
go handleEventStreamAsync(stream)
```

Detenga la sesión de Live Tail una vez transcurrido un periodo de tiempo.

```
// Close the stream (which ends the session) after a timeout
time.Sleep(10 * time.Second)
stream.Close()
log.Println("Event stream closed")
```

- Para obtener más información sobre la API, consulte StartLiveTaillla Referencia AWS SDK for Go de la API.

Java

SDK para Java 2.x

Incluir los archivos requeridos.

```
import io.reactivex.FlowableSubscriber;
import io.reactivex.annotations.NonNull;
import org.reactivestreams.Subscription;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsAsyncClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionStart;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionUpdate;
import software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseHandler;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseStream;
```

```
import java.util.Date;
import java.util.List;
import java.util.concurrent.atomic.AtomicReference;
```

Gestione los eventos de la sesión de Live Tail.

```
   private static StartLiveTailResponseHandler
getStartLiveTailResponseStreamHandler(
           AtomicReference<Subscription> subscriptionAtomicReference) {
       return StartLiveTailResponseHandler.builder()
           .onResponse(r -> System.out.println("Received initial response"))
           .onError(throwable -> {
               CloudWatchLogsException e = (CloudWatchLogsException)
throwable.getCause();
               System.err.println(e.awsErrorDetails().errorMessage());
               System.exit(1);
           })
           .subscriber(() -> new FlowableSubscriber<>() {
               @Override
               public void onSubscribe(@NonNull Subscription s) {
                   subscriptionAtomicReference.set(s);
                   s.request(Long.MAX_VALUE);
               }

               @Override
               public void onNext(StartLiveTailResponseStream event) {
                   if (event instanceof LiveTailSessionStart) {
                       LiveTailSessionStart sessionStart =
(LiveTailSessionStart) event;
                       System.out.println(sessionStart);
                   } else if (event instanceof LiveTailSessionUpdate) {
                       LiveTailSessionUpdate sessionUpdate =
(LiveTailSessionUpdate) event;
                       List<LiveTailSessionLogEvent> logEvents =
sessionUpdate.sessionResults();
                       logEvents.forEach(e -> {
                           long timestamp = e.timestamp();
                           Date date = new Date(timestamp);
                           System.out.println("[" + date + "] " + e.message());
                       });
                   } else {
```

```
                        throw CloudWatchLogsException.builder().message("Unknown
event type").build();
                    }
                }

                @Override
                public void onError(Throwable throwable) {
                    System.out.println(throwable.getMessage());
                    System.exit(1);
                }

                @Override
                public void onComplete() {
                    System.out.println("Completed Streaming Session");
                }
            })
            .build();
    }
```

Inicie la sesión de Live Tail.

```
        CloudWatchLogsAsyncClient cloudWatchLogsAsyncClient =
                CloudWatchLogsAsyncClient.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();

        StartLiveTailRequest request =
                StartLiveTailRequest.builder()
                    .logGroupIdentifiers(logGroupIdentifiers)
                    .logStreamNames(logStreamNames)
                    .logEventFilterPattern(logEventFilterPattern)
                    .build();

        /* Create a reference to store the subscription */
        final AtomicReference<Subscription> subscriptionAtomicReference = new
AtomicReference<>(null);

        cloudWatchLogsAsyncClient.startLiveTail(request,
getStartLiveTailResponseStreamHandler(subscriptionAtomicReference));
```

Detenga la sesión de Live Tail una vez transcurrido un periodo de tiempo.

```
        /* Set a timeout for the session and cancel the subscription. This will:
         * 1). Close the stream
         * 2). Stop the Live Tail session
         */
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        if (subscriptionAtomicReference.get() != null) {
            subscriptionAtomicReference.get().cancel();
            System.out.println("Subscription to stream closed");
        }
```

- Para obtener más información sobre la API, consulte StartLiveTaillla Referencia AWS SDK for Java 2.x de la API.

JavaScript

SDK para JavaScript (v3)

Incluir los archivos requeridos.

```
import { CloudWatchLogsClient, StartLiveTailCommand } from "@aws-sdk/client-
cloudwatch-logs";
```

Gestione los eventos de la sesión de Live Tail.

```
async function handleResponseAsync(response) {
    try {
      for await (const event of response.responseStream) {
        if (event.sessionStart !== undefined) {
          console.log(event.sessionStart);
        } else if (event.sessionUpdate !== undefined) {
          for (const logEvent of event.sessionUpdate.sessionResults) {
            const timestamp = logEvent.timestamp;
            const date = new Date(timestamp);
            console.log("[" + date + "] " + logEvent.message);
          }
```

```
        } else {
            console.error("Unknown event type");
        }
      }
    } catch (err) {
        // On-stream exceptions are captured here
        console.error(err)
    }
}
```

Inicie la sesión de Live Tail.

```
    const client = new CloudWatchLogsClient();

    const command = new StartLiveTailCommand({
        logGroupIdentifiers: logGroupIdentifiers,
        logStreamNames: logStreamNames,
        logEventFilterPattern: filterPattern
    });
    try{
        const response = await client.send(command);
        handleResponseAsync(response);
    } catch (err){
        // Pre-stream exceptions are captured here
        console.log(err);
    }
```

Detenga la sesión de Live Tail una vez transcurrido un periodo de tiempo.

```
    /* Set a timeout to close the client. This will stop the Live Tail session.
 */
    setTimeout(function() {
        console.log("Client timeout");
        client.destroy();
      }, 10000);
```

- Para obtener más información sobre la API, consulte StartLiveTaillla Referencia de AWS SDK for JavaScript la API.

Kotlin

SDK para Kotlin

Incluir los archivos requeridos.

```
import aws.sdk.kotlin.services.cloudwatchlogs.CloudWatchLogsClient
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailRequest
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailResponseStream
import kotlinx.coroutines.flow.takeWhile
```

Inicie la sesión de Live Tail.

```
    val client = CloudWatchLogsClient.fromEnvironment()

    val request = StartLiveTailRequest {
        logGroupIdentifiers = logGroupIdentifiersVal
        logStreamNames = logStreamNamesVal
        logEventFilterPattern = logEventFilterPatternVal
    }

    val startTime = System.currentTimeMillis()

    try {
        client.startLiveTail(request) { response ->
            val stream = response.responseStream
            if (stream != null) {
                /* Set a timeout to unsubcribe from the flow. This will:
                * 1). Close the stream
                * 2). Stop the Live Tail session
                */
                stream.takeWhile { System.currentTimeMillis() - startTime <
10000 }.collect { value ->
                    if (value is StartLiveTailResponseStream.SessionStart) {
                        println(value.asSessionStart())
                    } else if (value is
StartLiveTailResponseStream.SessionUpdate) {
                        for (e in value.asSessionUpdate().sessionResults!!) {
                            println(e)
                        }
                    } else {
                        throw IllegalArgumentException("Unknown event type")
```

```
                    }
                }
            } else {
                throw IllegalArgumentException("No response stream")
            }
        }
    } catch (e: Exception) {
        println("Exception occurred during StartLiveTail: $e")
        System.exit(1)
    }
```

- Para obtener más información sobre la API, consulta StartLiveTaillla referencia sobre el AWS SDK para la API de Kotlin.

Python

SDK para Python (Boto3)

Incluir los archivos requeridos.

```
import boto3
import time
from datetime import datetime
```

Inicie la sesión de Live Tail.

```
    # Initialize the client
    client = boto3.client('logs')

    start_time = time.time()

    try:
        response = client.start_live_tail(
            logGroupIdentifiers=log_group_identifiers,
            logStreamNames=log_streams,
            logEventFilterPattern=filter_pattern
        )
        event_stream = response['responseStream']
        # Handle the events streamed back in the response
        for event in event_stream:
```

```
            # Set a timeout to close the stream.
            # This will end the Live Tail session.
            if (time.time() - start_time >= 10):
                event_stream.close()
                break
            # Handle when session is started
            if 'sessionStart' in event:
                session_start_event = event['sessionStart']
                print(session_start_event)
            # Handle when log event is given in a session update
            elif 'sessionUpdate' in event:
                log_events = event['sessionUpdate']['sessionResults']
                for log_event in log_events:
                    print('[{date}]
{log}'.format(date=datetime.fromtimestamp(log_event['timestamp']/1000),log=log_event['me
            else:
                # On-stream exceptions are captured here
                raise RuntimeError(str(event))
    except Exception as e:
        print(e)
```

- Para obtener más información sobre la API, consulta StartLiveTaillla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso CloudWatch de registros con un AWS SDK En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Úselo **`StartQuery`** con un AWS SDK o CLI

En los siguientes ejemplos de código, se muestra cómo utilizar `StartQuery`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en el siguiente ejemplo de código:

- Ejecución de una consulta de gran tamaño

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
/**
 * Wrapper for the StartQueryCommand. Uses a static query string
 * for consistency.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 * @returns {Promise<{ queryId: string }>}
 */
async _startQuery([startDate, endDate], maxLogs = 10000) {
  try {
    return await this.client.send(
      new StartQueryCommand({
        logGroupNames: this.logGroupNames,
        queryString: "fields @timestamp, @message | sort @timestamp asc",
        startTime: startDate.valueOf(),
        endTime: endDate.valueOf(),
        limit: maxLogs,
      }),
    );
  } catch (err) {
    /** @type {string} */
    const message = err.message;
    if (message.startsWith("Query's end date and time")) {
      // This error indicates that the query's start or end date occur
      // before the log group was created.
      throw new DateOutOfBoundsError(message);
    }

    throw err;
  }
}
```

- Para obtener más información sobre la API, consulta StartQueryla Referencia AWS SDK for JavaScript de la API.

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_groups,
                    startTime=start_time,
                    endTime=end_time,
                    queryString="fields @timestamp, @message | sort @timestamp
asc",
                    limit=self.limit,
```

```
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_groups,
                startTime=start_time,
                endTime=end_time,
                queryString="fields @timestamp, @message | sort @timestamp asc",
```

```
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")
```

- Para obtener más información sobre la API, consulta StartQueryla AWS Referencia de API de SDK for Python (Boto3).

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte. Uso CloudWatch de registros con un AWS SDK En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Escenarios de CloudWatch registros que utilizan AWS SDK

En los siguientes ejemplos de código, se muestra cómo implementar escenarios comunes en CloudWatch registros con AWS SDK. Estos escenarios muestran cómo realizar tareas específicas mediante la llamada a varias funciones dentro de CloudWatch Logs. Cada escenario incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar el código.

Ejemplos

- Usa CloudWatch los registros para ejecutar una consulta grande

Usa CloudWatch los registros para ejecutar una consulta grande

Los siguientes ejemplos de código muestran cómo usar CloudWatch los registros para consultar más de 10 000 registros.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Hay más información. GitHub Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Este es el punto de entrada.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { CloudWatchLogsClient } from "@aws-sdk/client-cloudwatch-logs";
import { CloudWatchQuery } from "./cloud-watch-query.js";

console.log("Starting a recursive query...");

if (!process.env.QUERY_START_DATE || !process.env.QUERY_END_DATE) {
  throw new Error(
    "QUERY_START_DATE and QUERY_END_DATE environment variables are required.",
  );
}

const cloudWatchQuery = new CloudWatchQuery(new CloudWatchLogsClient({}), {
  logGroupNames: ["/workflows/cloudwatch-logs/large-query"],
  dateRange: [
    new Date(parseInt(process.env.QUERY_START_DATE)),
    new Date(parseInt(process.env.QUERY_END_DATE)),
  ],
});

await cloudWatchQuery.run();

console.log(
  `Queries finished in ${cloudWatchQuery.secondsElapsed} seconds.\nTotal logs
 found: ${cloudWatchQuery.results.length}`,
);
```

Esta es una clase que divide las consultas en varios pasos si es necesario.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  StartQueryCommand,
  GetQueryResultsCommand,
} from "@aws-sdk/client-cloudwatch-logs";
import { splitDateRange } from "@aws-doc-sdk-examples/lib/utils/util-date.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

class DateOutOfBoundsError extends Error {}
```

```
export class CloudWatchQuery {
  /**
   * Run a query for all CloudWatch Logs within a certain date range.
   * CloudWatch logs return a max of 10,000 results. This class
   * performs a binary search across all of the logs in the provided
   * date range if a query returns the maximum number of results.
   *
   * @param {import('@aws-sdk/client-cloudwatch-logs').CloudWatchLogsClient}
 client
   * @param {{ logGroupNames: string[], dateRange: [Date, Date], queryConfig:
 { limit: number } }} config
   */
  constructor(client, { logGroupNames, dateRange, queryConfig }) {
    this.client = client;
    /**
     * All log groups are queried.
     */
    this.logGroupNames = logGroupNames;

    /**
     * The inclusive date range that is queried.
     */
    this.dateRange = dateRange;

    /**
     * CloudWatch Logs never returns more than 10,000 logs.
     */
    this.limit = queryConfig?.limit ?? 10000;

    /**
     * @type {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]}
     */
    this.results = [];
  }

  /**
   * Run the query.
   */
  async run() {
    this.secondsElapsed = 0;
    const start = new Date();
    this.results = await this._largeQuery(this.dateRange);
    const end = new Date();
```

```
    this.secondsElapsed = (end - start) / 1000;
    return this.results;
  }

  /**
   * Recursively query for logs.
   * @param {[Date, Date]} dateRange
   * @returns {Promise<import("@aws-sdk/client-cloudwatch-logs").ResultField[]
[]>}
   */
  async _largeQuery(dateRange) {
    const logs = await this._query(dateRange, this.limit);

    console.log(
      `Query date range: ${dateRange
        .map((d) => d.toISOString())
        .join(" to ")}. Found ${logs.length} logs.`,
    );

    if (logs.length < this.limit) {
      return logs;
    }

    const lastLogDate = this._getLastLogDate(logs);
    const offsetLastLogDate = new Date(lastLogDate);
    offsetLastLogDate.setMilliseconds(lastLogDate.getMilliseconds() + 1);
    const subDateRange = [offsetLastLogDate, dateRange[1]];
    const [r1, r2] = splitDateRange(subDateRange);
    const results = await Promise.all([
      this._largeQuery(r1),
      this._largeQuery(r2),
    ]);
    return [logs, ...results].flat();
  }

  /**
   * Find the most recent log in a list of logs.
   * @param {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]} logs
   */
  _getLastLogDate(logs) {
    const timestamps = logs
      .map(
        (log) =>
          log.find((fieldMeta) => fieldMeta.field === "@timestamp")?.value,
```

```
      )
      .filter((t) => !!t)
      .map((t) => `${t}Z`)
      .sort();

    if (!timestamps.length) {
      throw new Error("No timestamp found in logs.");
    }

    return new Date(timestamps[timestamps.length - 1]);
  }

  // snippet-start:[javascript.v3.cloudwatch-logs.actions.GetQueryResults]
  /**
   * Simple wrapper for the GetQueryResultsCommand.
   * @param {string} queryId
   */
  _getQueryResults(queryId) {
    return this.client.send(new GetQueryResultsCommand({ queryId }));
  }
  // snippet-end:[javascript.v3.cloudwatch-logs.actions.GetQueryResults]

  /**
   * Starts a query and waits for it to complete.
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   */
  async _query(dateRange, maxLogs) {
    try {
      const { queryId } = await this._startQuery(dateRange, maxLogs);
      const { results } = await this._waitUntilQueryDone(queryId);
      return results ?? [];
    } catch (err) {
      /**
       * This error is thrown when StartQuery returns an error indicating
       * that the query's start or end date occur before the log group was
       * created.
       */
      if (err instanceof DateOutOfBoundsError) {
        return [];
      } else {
        throw err;
      }
    }
```

```
}

// snippet-start:[javascript.v3.cloudwatch-logs.actions.StartQuery]
/**
 * Wrapper for the StartQueryCommand. Uses a static query string
 * for consistency.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 * @returns {Promise<{ queryId: string }>}
 */
async _startQuery([startDate, endDate], maxLogs = 10000) {
  try {
    return await this.client.send(
      new StartQueryCommand({
        logGroupNames: this.logGroupNames,
        queryString: "fields @timestamp, @message | sort @timestamp asc",
        startTime: startDate.valueOf(),
        endTime: endDate.valueOf(),
        limit: maxLogs,
      }),
    );
  } catch (err) {
    /** @type {string} */
    const message = err.message;
    if (message.startsWith("Query's end date and time")) {
      // This error indicates that the query's start or end date occur
      // before the log group was created.
      throw new DateOutOfBoundsError(message);
    }

    throw err;
  }
}
// snippet-end:[javascript.v3.cloudwatch-logs.actions.StartQuery]

/**
 * Call GetQueryResultsCommand until the query is done.
 * @param {string} queryId
 */
_waitUntilQueryDone(queryId) {
  const getResults = async () => {
    const results = await this._getQueryResults(queryId);
    const queryDone = [
      "Complete",
```

```
        "Failed",
        "Cancelled",
        "Timeout",
        "Unknown",
      ].includes(results.status);

      return { queryDone, results };
    };

    return retry(
      { intervalInMs: 1000, maxRetries: 60, quiet: true },
      async () => {
        const { queryDone, results } = await getResults();
        if (!queryDone) {
          throw new Error("Query not done.");
        }

        return results;
      },
    );
  }
}
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK for JavaScript .
  - GetQueryResults
  - StartQuery

Python

SDK para Python (Boto3)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Este archivo invoca un módulo de ejemplo para gestionar CloudWatch consultas que superen los 10 000 resultados.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import logging
import os
import sys

import boto3
from botocore.config import Config

from cloudwatch_query import CloudWatchQuery
from date_utilities import DateUtilities

# Configure logging at the module level.
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(filename)s:%(lineno)d - %(message)s",
)


class CloudWatchLogsQueryRunner:
    def __init__(self):
        """
        Initializes the CloudWatchLogsQueryRunner class by setting up date
 utilities
        and creating a CloudWatch Logs client with retry configuration.
        """
        self.date_utilities = DateUtilities()
        self.cloudwatch_logs_client = self.create_cloudwatch_logs_client()

    def create_cloudwatch_logs_client(self):
        """
        Creates and returns a CloudWatch Logs client with a specified retry
 configuration.

        :return: A CloudWatch Logs client instance.
        :rtype: boto3.client
        """
        try:
            return boto3.client("logs", config=Config(retries={"max_attempts":
 10}))
        except Exception as e:
            logging.error(f"Failed to create CloudWatch Logs client: {e}")
            sys.exit(1)
```

```
   def fetch_environment_variables(self):
       """
       Fetches and validates required environment variables for query start and
end dates.

       :return: Tuple of query start date and end date as integers.
       :rtype: tuple
       :raises SystemExit: If required environment variables are missing or
invalid.
       """
       try:
           query_start_date = int(os.environ["QUERY_START_DATE"])
           query_end_date = int(os.environ["QUERY_END_DATE"])
       except KeyError:
           logging.error(
               "Both QUERY_START_DATE and QUERY_END_DATE environment variables
are required."
           )
           sys.exit(1)
       except ValueError as e:
           logging.error(f"Error parsing date environment variables: {e}")
           sys.exit(1)

       return query_start_date, query_end_date

   def convert_dates_to_iso8601(self, start_date, end_date):
       """
       Converts UNIX timestamp dates to ISO 8601 format using DateUtilities.

       :param start_date: The start date in UNIX timestamp.
       :type start_date: int
       :param end_date: The end date in UNIX timestamp.
       :type end_date: int
       :return: Start and end dates in ISO 8601 format.
       :rtype: tuple
       """
       start_date_iso8601 =
self.date_utilities.convert_unix_timestamp_to_iso8601(
           start_date
       )
       end_date_iso8601 = self.date_utilities.convert_unix_timestamp_to_iso8601(
           end_date
       )
```

```
        return start_date_iso8601, end_date_iso8601

    def execute_query(
        self,
        start_date_iso8601,
        end_date_iso8601,
        log_group="/workflows/cloudwatch-logs/large-query",
    ):
        """
        Creates a CloudWatchQuery instance and executes the query with provided
 date range.

        :param start_date_iso8601: The start date in ISO 8601 format.
        :type start_date_iso8601: str
        :param end_date_iso8601: The end date in ISO 8601 format.
        :type end_date_iso8601: str
        :param log_group: Log group to search: "/workflows/cloudwatch-logs/large-
query"
        :type log_group: str
        """
        cloudwatch_query = CloudWatchQuery(
            [start_date_iso8601, end_date_iso8601],
        )
        cloudwatch_query.query_logs((start_date_iso8601, end_date_iso8601))
        logging.info("Query executed successfully.")
        logging.info(
            f"Queries completed in {cloudwatch_query.query_duration} seconds.
 Total logs found: {len(cloudwatch_query.query_results)}"
        )


def main():
    """
    Main function to start a recursive CloudWatch logs query.
    Fetches required environment variables, converts dates, and executes the
 query.
    """
    logging.info("Starting a recursive CloudWatch logs query...")
    runner = CloudWatchLogsQueryRunner()
    query_start_date, query_end_date = runner.fetch_environment_variables()
    start_date_iso8601 = DateUtilities.convert_unix_timestamp_to_iso8601(
        query_start_date
    )
```

```
    end_date_iso8601 =
 DateUtilities.convert_unix_timestamp_to_iso8601(query_end_date)
    runner.execute_query(start_date_iso8601, end_date_iso8601)


if __name__ == "__main__":
    main()
```

Este módulo procesa CloudWatch las consultas que superan los 10 000 resultados.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import logging
import time
from datetime import datetime
import threading
import boto3

from date_utilities import DateUtilities


class DateOutOfBoundsError(Exception):
    """Exception raised when the date range for a query is out of bounds."""

    pass


class CloudWatchQuery:
    """
    A class to query AWS CloudWatch logs within a specified date range.

    :ivar date_range: Start and end datetime for the query.
    :vartype date_range: tuple
    :ivar limit: Maximum number of log entries to return.
    :vartype limit: int
    """

    def __init__(self, date_range):
        self.lock = threading.Lock()
        self.log_groups = "/workflows/cloudwatch-logs/large-query"
        self.query_results = []
        self.date_range = date_range
```

```
        self.query_duration = None
        self.datetime_format = "%Y-%m-%d %H:%M:%S.%f"
        self.date_utilities = DateUtilities()
        self.limit = 10000

    def query_logs(self, date_range):
        """
        Executes a CloudWatch logs query for a specified date range and
calculates the execution time of the query.

        :return: A batch of logs retrieved from the CloudWatch logs query.
        :rtype: list
        """
        start_time = datetime.now()

        start_date, end_date = self.date_utilities.normalize_date_range_format(
            date_range, from_format="unix_timestamp", to_format="datetime"
        )

        logging.info(
            f"Original query:"
            f"\n       START:    {start_date}"
            f"\n       END:      {end_date}"
        )
        self.recursive_query((start_date, end_date))
        end_time = datetime.now()
        self.query_duration = (end_time - start_time).total_seconds()

    def recursive_query(self, date_range):
        """
        Processes logs within a given date range, fetching batches of logs
recursively if necessary.

        :param date_range: The date range to fetch logs for, specified as a tuple
(start_timestamp, end_timestamp).
        :type date_range: tuple
        :return: None if the recursive fetching is continued or stops when the
final batch of logs is processed.
                 Although it doesn't explicitly return the query results, this
method accumulates all fetched logs
                 in the `self.query_results` attribute.
        :rtype: None
        """
        batch_of_logs = self.perform_query(date_range)
```

```
        # Add the batch to the accumulated logs
        with self.lock:
            self.query_results.extend(batch_of_logs)
        if len(batch_of_logs) == self.limit:
            logging.info(f"Fetched {self.limit}, checking for more...")
            most_recent_log = self.find_most_recent_log(batch_of_logs)
            most_recent_log_timestamp = next(
                item["value"]
                for item in most_recent_log
                if item["field"] == "@timestamp"
            )
            new_range = (most_recent_log_timestamp, date_range[1])
            midpoint = self.date_utilities.find_middle_time(new_range)

            first_half_thread = threading.Thread(
                target=self.recursive_query,
                args=((most_recent_log_timestamp, midpoint),),
            )
            second_half_thread = threading.Thread(
                target=self.recursive_query, args=((midpoint, date_range[1]),)
            )

            first_half_thread.start()
            second_half_thread.start()

            first_half_thread.join()
            second_half_thread.join()

    def find_most_recent_log(self, logs):
        """
        Search a list of log items and return most recent log entry.
        :param logs: A list of logs to analyze.
        :return: log
        :type :return List containing log item details
        """
        most_recent_log = None
        most_recent_date = "1970-01-01 00:00:00.000"

        for log in logs:
            for item in log:
                if item["field"] == "@timestamp":
                    logging.debug(f"Compared: {item['value']} to
{most_recent_date}")
                    if (
```

```
                    self.date_utilities.compare_dates(
                        item["value"], most_recent_date
                    )
                    == item["value"]
                ):
                    logging.debug(f"New most recent: {item['value']}")
                    most_recent_date = item["value"]
                    most_recent_log = log
        logging.info(f"Most recent log date of batch: {most_recent_date}")
        return most_recent_log

    # snippet-start:[python.example_code.cloudwatch_logs.start_query]
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_groups,
                    startTime=start_time,
                    endTime=end_time,
                    queryString="fields @timestamp, @message | sort @timestamp
asc",
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
```

```
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_groups,
                startTime=start_time,
                endTime=end_time,
                queryString="fields @timestamp, @message | sort @timestamp asc",
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
```

```
            raise DateOutOfBoundsError(f"Resource not found: {e}")

    # snippet-end:[python.example_code.cloudwatch_logs.start_query]

    # snippet-start:[python.example_code.cloudwatch_logs.get_query_results]
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])

    # snippet-end:[python.example_code.cloudwatch_logs.get_query_results]
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para Python (Boto3).
  - GetQueryResults
  - StartQuery

Para obtener una lista completa de guías para desarrolladores del AWS SDK y ejemplos de código, consulteUso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Ejemplos de servicios cruzados de CloudWatch registros que utilizan SDK AWS

En las siguientes aplicaciones de ejemplo, se utilizan AWS los SDK para combinar los CloudWatch registros con otros. Servicios de AWS Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones sobre cómo configurar y ejecutar la aplicación.

Ejemplos

- Usar eventos programados para invocar una función de Lambda

Usar eventos programados para invocar una función de Lambda

Los siguientes ejemplos de código muestran cómo crear una AWS Lambda función invocada por un evento EventBridge programado de Amazon.

Python

SDK para Python (Boto3)

En este ejemplo se muestra cómo registrar una AWS Lambda función como destino de un EventBridge evento programado de Amazon. El controlador Lambda escribe un mensaje descriptivo y los datos completos del evento en Amazon CloudWatch Logs para su posterior recuperación.

- Implementa una función de Lambda.
- Crea un evento EventBridge programado y convierte la función Lambda en el objetivo.
- Otorga permiso para EventBridge invocar la función Lambda.
- Imprime los datos más recientes de CloudWatch los registros para mostrar el resultado de las invocaciones programadas.
- Limpia todos los recursos creados durante la demostración.

Es mejor ver este ejemplo en GitHub. Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- CloudWatch Registros
- EventBridge

- Lambda

Para obtener una lista completa de las guías para desarrolladores del AWS SDK y ejemplos de código, consulte Uso CloudWatch de registros con un AWS SDK. En este tema también se incluye información sobre cómo comenzar a utilizar el SDK y detalles sobre sus versiones anteriores.

Seguridad en Amazon CloudWatch Logs

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

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

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento aplicables WorkSpaces, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos .
- 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 los reglamentos aplicables

Esta documentación le ayuda a entender cómo aplicar el modelo de responsabilidad compartida al utilizar Amazon CloudWatch Logs. Le muestra cómo configurar Amazon CloudWatch Logs para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de CloudWatch Logs.

Contenido

Protección de datos en Amazon CloudWatch Logs

> **Note**
>
> Además de la siguiente información sobre la protección general de datos AWS, CloudWatch Logs también le permite proteger los datos confidenciales de los eventos de registro ocultándolos. Para obtener más información, consulte Ayude a proteger los datos de registro confidenciales con el enmascaramiento.

El modelo de se aplica a protección de datos en Amazon CloudWatch Logs. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información

sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con CloudWatch registros u otros usos de la Servicios de AWS consola, la API o los SDK. AWS CLI AWS Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya la información de las credenciales en la URL para validar la solicitud para ese servidor.

Cifrado en reposo

CloudWatch Logs protege los datos en reposo mediante el cifrado. Todos los grupos de registro están cifrados. De forma predeterminada, el servicio de CloudWatch registros administra las claves de cifrado del lado del servidor.

Si desea administrar las claves utilizadas para cifrar y descifrar los registros, utilice las claves. AWS KMS Para obtener más información, consulte Cifre los datos de registro en los CloudWatch registros mediante AWS Key Management Service.

Cifrado en tránsito

CloudWatch Los registros utilizan el end-to-end cifrado de los datos en tránsito. El servicio CloudWatch de registros administra las claves de cifrado del lado del servidor.

Administración de identidades y accesos para Amazon CloudWatch Logs

El acceso a Amazon CloudWatch Logs requiere credenciales que AWS puedas usar para autenticar tus solicitudes. Esas credenciales deben tener permisos para acceder a los AWS recursos, por ejemplo, para recuperar los datos de CloudWatch Logs sobre sus recursos en la nube. En las siguientes secciones, se proporciona información detallada sobre cómo puede utilizar AWS Identity and Access Management (IAM) y CloudWatch los registros para proteger sus recursos controlando quién puede acceder a ellos:

- Autenticación
- Control de acceso

Autenticación

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center .
- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Control de acceso

Puede tener credenciales válidas para autenticar sus solicitudes, pero a menos que tenga permisos, no podrá crear los recursos de CloudWatch Logs ni acceder a ellos. Por ejemplo, debe disponer de permisos para crear flujos de registro, crear grupos de registro, etc.

En las siguientes secciones, se describe cómo administrar los permisos de los CloudWatch registros. Recomendamos que lea primero la información general.

- Descripción general de la administración de los permisos de acceso a los recursos CloudWatch de Logs
- Uso de políticas basadas en la identidad (políticas de IAM) para los registros CloudWatch
- CloudWatch Referencia de permisos de registro

Descripción general de la administración de los permisos de acceso a los recursos CloudWatch de Logs

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center .
- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones descritas en Creación de un rol para un proveedor de identidad de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Temas

CloudWatch Registra los recursos y las operaciones

En CloudWatch Logs, los recursos principales son los grupos de registros, los flujos de registros y los destinos. CloudWatch Logs no admite subrecursos (otros recursos para usar con el recurso principal).

Estos recursos y subrecursos tienen nombres de recursos de Amazon (ARN) únicos asociados a ellos, tal y como se muestra en la siguiente tabla:

Tipo de recurso	Formato de ARN
Grupo de registros	Se utilizan las dos siguientes opciones. El segundo, con el `:*` al final, es lo que devuelven el comando `describe-log-groups` CLI y la DescribeLogGroupsAPI. arn:aws:logs:*región*:*ID-de-cuenta* :log-group:*nombre_de_grupo_de_registros* arn:aws:logs:*región*:*ID-de-cuenta* :log-group:*nombre_de_grupo_de_registros* :* Utilice la primera versión, sin la versión final`:*`, en las siguientes situaciones: • En el campo `logGroupIdentifier` de entrada de muchas CloudWatch Logs API. • En el `resourceArn` campo del etiquetado de las API • En IAM las políticas, al especificar los permisos para TagResourceUntagResource, y ListTagsForResource. Utilice la segunda versión, con la última`:*`, para hacer referencia al ARN al especificar los permisos en las políticas de IAM para todas las demás acciones de la API.
Flujo de registro	*arn:aws:logs: region: account-id:log-group: log_group_name:log-stream: log-stream-name*
Destino	arn:aws:logs:*región*:*ID-de-cuenta* :destination:*nombre_de_destino*

Para obtener más información sobre los ARN, consulte ARN en la Guía del usuario de IAM. Para obtener información sobre CloudWatch los ARN de registros, consulte Amazon Resource Names (ARN) en. Referencia general de Amazon Web Services Para ver un ejemplo de una política que cubre CloudWatch los registros, consulte. Uso de políticas basadas en la identidad (políticas de IAM) para los registros CloudWatch

CloudWatch Los registros proporcionan un conjunto de operaciones para trabajar con los recursos de los CloudWatch registros. Para ver la lista de las operaciones disponibles, consulte CloudWatch Referencia de permisos de registro.

Titularidad de los recursos

La AWS cuenta es propietaria de los recursos que se crean en la cuenta, independientemente de quién los haya creado. En concreto, el propietario del recurso es la AWS cuenta de la entidad principal (es decir, la cuenta raíz, un usuario o un rol de IAM) que autentica la solicitud de creación de recursos. Los siguientes ejemplos ilustran cómo funciona:

- Si utiliza las credenciales de la cuenta raíz de su AWS cuenta para crear un grupo de registros, su AWS cuenta es la propietaria del recurso de CloudWatch registros.
- Si creas un usuario en tu AWS cuenta y le concedes permisos para crear recursos de CloudWatch Logs, el usuario podrá crear recursos de CloudWatch Logs. Sin embargo, tu AWS cuenta, a la que pertenece el usuario, es propietaria de los recursos de CloudWatch Logs.
- Si crea un rol de IAM en su AWS cuenta con permisos para crear recursos de CloudWatch registros, cualquier persona que pueda asumir el rol podrá crear recursos de CloudWatch registros. Tu AWS cuenta, a la que pertenece el rol, es propietaria de los recursos de CloudWatch Logs.

Administración del acceso a los recursos

Una política de permisos describe quién tiene acceso a qué. En la siguiente sección se explican las opciones disponibles para crear políticas de permisos.

> **Note**
>
> En esta sección se describe el uso de IAM en el contexto de CloudWatch los registros. No se proporciona información detallada sobre el servicio de IAM. Para ver la documentación completa de IAM, consulte What is IAM? (¿Qué es IAM?) en la Guía del usuario de IAM.

Para obtener más información sobre la sintaxis y descripciones de las políticas de IAM, consulte Referencia de políticas de IAM en la Guía del usuario de IAM.

Las políticas asociadas a una identidad de IAM se denominan políticas basadas en la identidad (políticas de IAM) y las políticas asociadas a un recurso se denominan políticas basadas en recursos. CloudWatch Los registros admiten políticas basadas en la identidad y políticas basadas en recursos para los destinos, que se utilizan para habilitar las suscripciones entre cuentas. Para obtener más información, consulte Suscripciones multicuentas y regiones.

Temas

- Permisos de grupo de registro y Información de colaboradores
- Políticas basadas en recursos

Permisos de grupo de registro y Información de colaboradores

Contributor Insights es una función CloudWatch que le permite analizar los datos de los grupos de registros y crear series temporales que muestren los datos de los colaboradores. Puede ver métricas acerca de los colaboradores Top-N, el número total de colaboradores únicos y su uso. Para obtener más información, consulta Uso de Información de colaboradores para analizar datos de alta cardinalidad.

Al conceder a un usuario los `cloudwatch:GetInsightRuleReport` permisos `cloudwatch:PutInsightRule` y, dicho usuario puede crear una regla que evalúe cualquier grupo de registros de los CloudWatch registros y, a continuación, ver los resultados. Los resultados pueden contener datos de colaborador para esos grupos de registro. Asegúrese de conceder estos permisos solo a los usuarios que puedan ver estos datos.

Políticas basadas en recursos

CloudWatch Logs admite políticas basadas en recursos para los destinos, que puedes usar para habilitar las suscripciones entre cuentas. Para obtener más información, consulte Paso 1: crear un destino. Los destinos se pueden crear mediante la PutDestinationAPI y se puede añadir una política de recursos al destino mediante la PutDestinationPolicyAPI. El siguiente ejemplo permite que otra AWS cuenta con el ID de cuenta 111122223333 suscriba sus grupos de registros al destino. `arn:aws:logs:us-east-1:123456789012:destination:testDestination`

```
{
```

```
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111122223333"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-east-1:123456789012:destination:testDestination"
    }
  ]
}
```

Especificar elementos de la política: acciones, efectos y entidades principales

Para cada recurso CloudWatch de Logs, el servicio define un conjunto de operaciones de API. Para conceder permisos para estas operaciones de API, CloudWatch Logs define un conjunto de acciones que puedes especificar en una política. Algunas operaciones de API pueden requerir permisos para más de una acción para poder realizar la operación de API. Para obtener más información sobre los recursos y las operaciones de API, consulte CloudWatch Registra los recursos y las operaciones y CloudWatch Referencia de permisos de registro.

A continuación, se indican los elementos básicos de la política:

- Recurso: use un Nombre de recurso de Amazon (ARN) para identificar el recurso al que se aplica la política. Para obtener más información, consulte CloudWatch Registra los recursos y las operaciones.
- Acción: utilice palabras clave de acción para identificar las operaciones del recurso que desea permitir o denegar. Por ejemplo, el permiso `logs.DescribeLogGroups` concede permiso a los usuarios para realizar la operación `DescribeLogGroups`.
- Efecto: especifique el efecto, permitir o denegar, cuando el usuario solicite la acción específica. Si no concede acceso de forma explícita (permitir) a un recurso, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a un recurso para asegurarse de que un usuario no pueda obtener acceso a él, aunque otra política le conceda acceso.
- Entidad principal: en las políticas basadas en identidades (políticas de IAM), el usuario al que se asocia esta política es la entidad principal implícita. En el caso de las políticas basadas en recursos, debe especificar el usuario, la cuenta, el servicio u otra entidad para la que desea

recibir los permisos (solo se aplica a las políticas basadas en recursos). CloudWatch Logs admite políticas basadas en recursos para los destinos.

Para obtener más información sobre la sintaxis y descripciones de las políticas de IAM, consulte Referencia de la política de IAM de AWS en la Guía del usuario de IAM.

Para ver una tabla que muestra todas las acciones de la API de CloudWatch Logs y los recursos a los que se aplican, consulte. CloudWatch Referencia de permisos de registro

Especificación de las condiciones de una política

Al conceder permisos, puede utilizar el lenguaje de la política de acceso para especificar las condiciones en las que se debe aplicar una política. Por ejemplo, es posible que desee que solo se aplique una política después de una fecha específica. Para obtener más información sobre cómo especificar condiciones en un lenguaje de política, consulte Condition en la Guía del usuario de IAM.

Para expresar condiciones, se usan claves de condición predefinidas. Para obtener una lista de las claves de contexto compatibles con cada AWS servicio y una lista de las claves de política AWS generales, consulte las claves de contexto de las acciones, los recursos y las claves de condición de los AWS servicios y las claves de contexto de condición AWS globales.

> **Note**
>
> Puede usar etiquetas para controlar el acceso a CloudWatch los recursos de los registros, incluidos los grupos de registros y los destinos. El acceso a los flujos de registro se controla a nivel de grupo de registro, debido a la relación jerárquica que existe entre los grupos de registro y los flujos de registro. A fin de obtener información sobre el uso de etiquetas para controlar el acceso, consulte Control del acceso a recursos de Amazon Web Services mediante etiquetas.

Uso de políticas basadas en la identidad (políticas de IAM) para los registros CloudWatch

Este tema contiene ejemplos de políticas basadas en identidades, donde los administradores de cuentas pueden asociar políticas de permisos a identidades de IAM (es decir, a usuarios, grupos y funciones).

> **Important**
>
> Le recomendamos que consulte primero los temas introductorios en los que se explican los conceptos básicos y las opciones disponibles para administrar el acceso a sus CloudWatch recursos de Logs. Para obtener más información, consulte Descripción general de la administración de los permisos de acceso a los recursos CloudWatch de Logs.

Este tema cubre lo siguiente:

- Permisos necesarios para usar la CloudWatch consola
- AWS políticas gestionadas (predefinidas) para CloudWatch los registros
- Ejemplos de políticas administradas por el cliente

A continuación se muestra un ejemplo de una política de permisos:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Esta política tiene una declaración que concede permisos para crear grupos de registros y flujos de registro para cargar eventos de registro a flujos de registro y para mostrar un listado de detalles acerca de los flujos de registro.

El carácter comodín (*) que aparece al final del valor `Resource` significa que la declaración concede permiso para las acciones `logs:CreateLogGroup`, `logs:CreateLogStream`,

`logs:PutLogEvents` y `logs:DescribeLogStreams` en cualquier grupo de registros. Para limitar este permiso a un grupo de registros específico, sustituya el carácter comodín (*) en el ARN del recurso con el ARN de grupo de registros específico. Para obtener más información acerca de las secciones dentro de una declaración de política de IAM, consulte Referencia de los elementos de la política de IAM en la Guía del usuario de IAM. Para ver una lista de todas las acciones de CloudWatch Logs, consulteCloudWatch Referencia de permisos de registro.

Permisos necesarios para usar la CloudWatch consola

Para que un usuario pueda trabajar con los CloudWatch registros de la CloudWatch consola, debe tener un conjunto mínimo de permisos que le permita describir otros AWS recursos de su AWS cuenta. Para usar CloudWatch los registros en la CloudWatch consola, debe tener permisos de los siguientes servicios:

- CloudWatch
- CloudWatch Registros
- OpenSearch Servicio
- IAM
- Kinesis
- Lambda
- Amazon S3

Si crea una política de IAM que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para los usuarios con esa política de IAM. Para garantizar que esos usuarios puedan seguir utilizando la CloudWatch consola, adjunte también la política `CloudWatchReadOnlyAccess` administrada al usuario, tal y como se describe enAWS políticas gestionadas (predefinidas) para CloudWatch los registros.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API CloudWatch Logs AWS CLI o a la misma.

El conjunto completo de permisos necesarios para que un usuario que no utilice la CloudWatch consola para gestionar las suscripciones de registros funcione con la consola es el siguiente:

- Cloudwatch: GetMetricData
- vigilancia en la nube: ListMetrics
- registros: CancelExportTask

- registros: CreateExportTask
- registros: CreateLogGroup
- registros: CreateLogStream
- registros: DeleteLogGroup
- registros: DeleteLogStream
- registros: DeleteMetricFilter
- registros: DeleteQueryDefinition
- registros: DeleteRetentionPolicy
- registros: DeleteSubscriptionFilter
- registros: DescribeExportTasks
- registros: DescribeLogGroups
- registros: DescribeLogStreams
- registros: DescribeMetricFilters
- registros: DescribeQueryDefinitions
- registros: DescribeQueries
- registros: DescribeSubscriptionFilters
- registros: FilterLogEvents
- registros: GetLogEvents
- registros: GetLogGroupFields
- registros: GetLogRecord
- registros: GetQueryResults
- registros: PutMetricFilter
- registros: PutQueryDefinition
- registros: PutRetentionPolicy
- registros: StartQuery
- registros: StopQuery
- registros: PutSubscriptionFilter
- registros: TestMetricFilter

Para un usuario que también utilice la consola para administrar las suscripciones de registro, los siguientes permisos son igualmente necesarios:

- Sí: DescribeElasticsearchDomain
- Sí: ListDomainNames
- objetivo: AttachRolePolicy
- objetivo: CreateRole
- objetivo: GetPolicy
- objetivo: GetPolicyVersion
- objetivo: GetRole
- objetivo: ListAttachedRolePolicies
- objetivo: ListRoles
- cinesia: DescribeStreams
- cinesia: ListStreams
- lambda: AddPermission
- lambda: CreateFunction
- lambda: GetFunctionConfiguration
- lambda: ListAliases
- lambda: ListFunctions
- lambda: ListVersionsByFunction
- lambda: RemovePermission
- s3: ListBuckets

AWS políticas gestionadas (predefinidas) para CloudWatch los registros

AWS aborda muchos casos de uso comunes al proporcionar políticas de IAM independientes que son creadas y administradas por. AWS Las políticas administradas conceden los permisos necesarios para casos de uso comunes, lo que le evita tener que investigar los permisos que se necesitan. Para más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Las siguientes políticas AWS gestionadas, que puede adjuntar a los usuarios y roles de su cuenta, son específicas de los CloudWatch registros:

- CloudWatchLogsFullAccess— Otorga acceso completo a CloudWatch los registros.
- CloudWatchLogsReadOnlyAccess— Otorga acceso de solo lectura a los CloudWatch registros.

CloudWatchLogsFullAccess

La CloudWatchLogsFullAccesspolítica otorga acceso total a CloudWatch los registros. La política incluye el `cloudwatch:GenerateQuery` permiso, de modo que los usuarios con esta política puedan generar una cadena de consulta de CloudWatch Logs Insights a partir de un mensaje en lenguaje natural. El contenido es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

CloudWatchLogsReadOnlyAccess

La CloudWatchLogsReadOnlyAccesspolítica concede acceso de solo lectura a los CloudWatch registros. Incluye el `cloudwatch:GenerateQuery` permiso para que los usuarios con esta política puedan generar una cadena de consulta de CloudWatch Logs Insights a partir de un mensaje en lenguaje natural. El contenido es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
```

```
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudWatchLogsCrossAccountSharingConfiguration

La CloudWatchLogsCrossAccountSharingConfigurationpolítica permite crear, administrar y ver los enlaces de Observability Access Manager para compartir los recursos de CloudWatch Logs entre cuentas. Para obtener más información, consulta la observabilidad CloudWatch entre cuentas.

El contenido es el siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Link",
                "oam:ListLinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
```

```
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

CloudWatch Registra las actualizaciones de las políticas gestionadas AWS

Consulta los detalles sobre las actualizaciones de las políticas AWS administradas para CloudWatch los registros desde que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbase a la fuente RSS de la página del historial de documentos de CloudWatch registro.

Cambio	Descripción	Fecha
CloudWatchLogsFullAccess: actualización de una política existente.	CloudWatch Los registros agregaron un permiso a CloudWatchLogsFullAccess. Se agregó el `cloudwatch:GenerateQuery` permiso para que los usuarios con esta política puedan generar una cadena de consulta de CloudWatch Logs Insights a partir de un mensaje en lenguaje natural.	27 de noviembre de 2023
CloudWatchLogsReadOnlyAccess: actualización de una política existente.	CloudWatch agregó un permiso para CloudWatchLogsReadOnlyAccess. Se agregó el `cloudwatch:GenerateQuery` permiso para que los usuarios	27 de noviembre de 2023

Cambio	Descripción	Fecha
	con esta política puedan generar una cadena de consulta de CloudWatch Logs Insights a partir de un mensaje en lenguaje natural.	
CloudWatchLogsReadOnlyAccess: actualización de una política actual	CloudWatch Los registros agregaron permisos a CloudWatchLogsReadOnlyAccess. Los `logs:StopLiveTail` permisos `logs:StartLiveTail` y se agregaron para que los usuarios con esta política puedan usar la consola para iniciar y detener las sesiones finales de CloudWatch Logs Live. Para obtener más información, consulte Use live tail to view logs in near real time.	6 de junio de 2023
CloudWatchLogsCrossAccountSharingConfiguration: política nueva	CloudWatch Logs agregó una nueva política que te permite administrar los enlaces de observabilidad CloudWatch entre cuentas que comparten grupos de CloudWatch registros de Logs. Para obtener más información, consulta CloudWatch la observabilidad multicuenta	27 de noviembre de 2022

Cambio	Descripción	Fecha
CloudWatchLogsRead OnlyAccess: actualización de una política actual	CloudWatch Registra los permisos añadidos a. CloudWatchLogsRead OnlyAccess Los `oam:ListAttachedLinks` permisos `oam:ListSinks` y se agregaron para que los usuarios con esta política puedan usar la consola para ver los datos compartidos desde las cuentas de origen de CloudWatch forma observable entre cuentas.	27 de noviembre de 2022

Ejemplos de políticas administradas por el cliente

Puede crear sus propias políticas de IAM personalizadas para permitir permisos para las acciones y los recursos de CloudWatch Logs. Puede asociar estas políticas personalizadas a los usuarios o grupos de que requieran esos permisos.

En esta sección, encontrará ejemplos de políticas de usuario que otorgan permisos para diversas acciones de CloudWatch Logs. Estas políticas funcionan cuando utilizas la API de CloudWatch Logs, AWS los SDK o el AWS CLI.

Ejemplos

- Ejemplo 1: permitir el acceso total a los registros CloudWatch
- Ejemplo 2: Permitir el acceso de solo lectura a los registros CloudWatch
- Ejemplo 3: permitir el acceso a un grupo de registro

Ejemplo 1: permitir el acceso total a los registros CloudWatch

La siguiente política permite a un usuario acceder a todas las acciones de los CloudWatch registros.

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

Ejemplo 2: Permitir el acceso de solo lectura a los registros CloudWatch

AWS proporciona una CloudWatchLogsReadOnlyAccesspolítica que permite el acceso de solo lectura a los datos de los registros. CloudWatch Esta política incluye los siguientes permisos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Ejemplo 3: permitir el acceso a un grupo de registro

La siguiente política permite a un usuario leer y escribir eventos de registro en un grupo de registros especificado.

> **Important**
>
> El `:*` al final del nombre del grupo de registro en la línea `Resource` es necesario para indicar que la política aplica a todos los flujos de registro de este grupo de registro. Si omite `:*`, no se aplicará la política.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName:*"
      }
   ]
}
```

Usar el etiquetado y las políticas de IAM para realizar el control en el nivel de grupo de registro

Puede conceder a los usuarios acceso a determinados grupos de registros al mismo tiempo que les impide tener acceso a otros grupos de registros. Para ello, etiquete los grupos de registro y utilice políticas de IAM que hagan referencia a esas etiquetas. Para aplicar etiquetas a un grupo de registro, debe tener el permiso `logs:TagResource` o `logs:TagLogGroup`. Esto se aplica si asigna etiquetas al grupo de registro cuando lo crea o si las asigna más adelante.

Para obtener más información sobre el etiquetado de grupos de registros, consulte Etiquetar grupos de registros en Amazon CloudWatch Logs.

Al etiquetar grupos de registro, puede conceder una política de IAM a un usuario para permitirle el acceso únicamente a los grupos de registro con una etiqueta determinada. Por ejemplo, la siguiente instrucción de política concede acceso únicamente a los grupos de registros que tienen el valor `Team` para la clave de etiqueta `Green`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

Las operaciones StopQueryy las de la StopLiveTailAPI no interactúan con AWS los recursos en el sentido tradicional. No devuelven ningún dato, no colocan ningún dato ni tampoco modifican ningún recurso de ninguna manera. En cambio, solo funcionan en una sesión de seguimiento en vivo determinada o en una consulta de CloudWatch Logs Insights determinada, que no se clasifican como recursos. Por lo tanto, al especificar el campo `Resource` en las políticas de IAM para estas operaciones, debe establecer el valor del campo `Resource` como `*`, como se muestra en el siguiente ejemplo.

```
{
    "Version": "2012-10-17",
    "Statement":
        [ {
            "Effect": "Allow",
            "Action": [
                "logs:StopQuery",
                "logs:StopLiveTail"
            ],
            "Resource": "*"
```

```
            }
        ]
}
```

Para obtener más información acerca del uso de instrucciones de política de IAM, consulte Control del acceso mediante las políticas en la Guía del usuario de IAM.

CloudWatch Referencia de permisos de registro

Puede usar la siguiente tabla como referencia cuando configure Control de acceso y escriba políticas de permisos que vaya a asociar a una identidad de IAM (políticas basadas en identidad). En la tabla se muestra cada operación de la API de CloudWatch Logs y las acciones correspondientes para las que puede conceder permisos para realizar la acción. Las acciones se especifican en el campo `Action` de la política. Para el `Resource` campo, puede especificar el ARN de un grupo de registros o un flujo de registros, o especificar que represente todos los `*` recursos de CloudWatch registros.

Puede utilizar claves AWS de condición completas en sus políticas de CloudWatch registros para expresar las condiciones. Para obtener una lista completa de las claves AWS generales, consulte las claves de contexto de condición AWS globales y de IAM en la Guía del usuario de IAM.

> **Note**
>
> Para especificar una acción, use el prefijo `logs:` seguido del nombre de operación de API. Por ejemplo:`logs:CreateLogGroup`,`logs:CreateLogStream`, o `logs:*` (para todas las acciones de los CloudWatch registros).

CloudWatch Registra las operaciones de la API y los permisos necesarios para las acciones

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
CancelExportTask	`logs:CancelExportTask` Necesario para cancelar una tarea de exportación en ejecución o pendiente.
CreateExportTask	`logs:CreateExportTask` Necesario para exportar datos desde un grupo de registros a un bucket de Amazon S3.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
CreateLogGroup	`logs:CreateLogGroup` Necesario para crear un nuevo grupo de registros.
CreateLogStream	`logs:CreateLogStream` Necesario para crear un nuevo flujo de registros en un grupo de registros.
DeleteDestination	`logs:DeleteDestination` Necesario para eliminar un destino de registro y deshabilita los filtros de suscripción al mismo.
DeleteLogGroup	`logs:DeleteLogGroup` Necesario para eliminar un grupo de registros y todos los eventos de registro asociados.
DeleteLogStream	`logs:DeleteLogStream` Necesario para eliminar un flujo de registros y todos los eventos de registro asociados.
DeleteMetricFilter	`logs:DeleteMetricFilter` Necesario para eliminar un filtro de métricas asociado con un grupo de registros.
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Necesario para eliminar una definición de consulta guardada en CloudWatch Logs Insights.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Necesario para eliminar una política CloudWatch de recursos de Logs.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Necesario para eliminar la política de retención de un grupo de registros.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Necesario para eliminar el filtro de suscripción asociado a un grupo de registros.
DescribeDestinations	`logs:DescribeDestinations` Necesario para ver todos los destinos asociados a la cuenta.
DescribeExportTasks	`logs:DescribeExportTasks` Necesario para ver todas las tareas de exportación asociadas a la cuenta.
DescribeLogGroups	`logs:DescribeLogGroups` Necesario para ver todos los grupos de registro asociados a la cuenta.
DescribeLogStreams	`logs:DescribeLogStreams` Necesario para ver todos los flujos de registro asociados a un grupo de registros.
DescribeMetricFilters	`logs:DescribeMetricFilters` Necesario para ver todas las métricas asociadas a un grupo de registros.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Necesario para ver la lista de definiciones de consultas guardadas en CloudWatch Logs Insights.
DescribeQueries	`logs:DescribeQueries` Necesario para ver la lista de consultas de CloudWatch Logs Insights que están programadas, en ejecución o que se han ejecutado recientemente.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Necesario para ver una lista de políticas de recursos de CloudWatch Logs.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Necesario para ver todos los filtros de suscripci ón asociados con un grupo de registros.
FilterLogEvents	`logs:FilterLogEvents` Necesario para ordenar los eventos de registros por patrón de filtro de grupo de registros.
GetLogEvents	`logs:GetLogEvents` Necesario para recuperar eventos de registro de un flujo de registros.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
GetLogGroupFields	`logs:GetLogGroupFields` Necesario para recuperar la lista de campos que se incluyen en los eventos de registro de un grupo de registros.
GetLogRecord	`logs:GetLogRecord` Necesario para recuperar los detalles de un único evento de registro.
GetQueryResults	`logs:GetQueryResults` Necesario para recuperar los resultados de las consultas de CloudWatch Logs Insights.
ListTagsLogGroup	`logs:ListTagsLogGroup` Necesario para ver las etiquetas asociadas a un grupo de registros.
PutDestination	`logs:PutDestination` Necesario para crear o actualizar un flujo de registros de destino (como, por ejemplo, un flujo de Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Necesario para crear o actualizar una política de acceso asociada a un destino de registro existente.
PutLogEvents	`logs:PutLogEvents` Necesario para cargar un lote de eventos de registro en un flujo de registros.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
PutMetricFilter	`logs:PutMetricFilter` Necesario para crear o actualizar un filtro de métricas y asociarlo a un grupo de registros.
PutQueryDefinition	`logs:PutQueryDefinition` Necesario para guardar una consulta en CloudWatch Logs Insights.
PutResourcePolicy	`logs:PutResourcePolicy` Necesario para crear una política CloudWatch de recursos de Logs.
PutRetentionPolicy	`logs:PutRetentionPolicy` Necesario para establecer el número de días que conservar los eventos de registro (retenció n) en un grupo de registros.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Necesario para crear o actualizar un filtro de suscripción y asociarlo a un grupo de registros.
StartQuery	`logs:StartQuery` Necesario para iniciar las consultas CloudWatc h de Logs Insights.
StopQuery	`logs:StopQuery` Necesario para detener una consulta de CloudWatch Logs Insights que está en curso.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
TagLogGroup	`logs:TagLogGroup` Necesario para añadir o actualizar etiquetas de grupo de registro.
TestMetricFilter	`logs:TestMetricFilter` Necesario para probar un patrón de filtro con respecto a una muestra de mensajes de evento de registro.

Uso de roles vinculados a servicios para Logs CloudWatch

Amazon CloudWatch Logs utiliza funciones AWS Identity and Access Management vinculadas a servicios (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Logs. CloudWatch Los roles vinculados al servicio están predefinidos en CloudWatch Logs e incluyen todos los permisos que el servicio requiere para llamar a otros AWS servicios en tu nombre.

Un rol vinculado a un servicio hace que la configuración de CloudWatch los registros sea más eficiente, ya que no es necesario añadir manualmente los permisos necesarios. CloudWatch Logs define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo CloudWatch Logs puede asumir esas funciones. Los permisos definidos incluyen la política de confianza y la política de permisos. Dicha política de permisos no se puede asociar a ninguna otra entidad de IAM.

Para obtener más información sobre otros servicios que admiten los roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM. Busque los servicios para los que se indique Sí en la columna Roles vinculados a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Permisos de roles vinculados al servicio para Logs CloudWatch

CloudWatch Logs usa el rol vinculado al servicio denominado. AWSServiceRoleForLogDelivery CloudWatch Logs usa esta función vinculada a un servicio para escribir registros directamente en Firehose. Para obtener más información, consulte Habilitar el registro desde AWS los servicios.

El rol vinculado al servicio AWSServiceRoleForLogDelivery confía en los siguientes servicios para asumir el rol:

- `logs.amazonaws.com`

La política de permisos de roles permite a CloudWatch Logs realizar las siguientes acciones en los recursos especificados:

- Acción: `firehose:PutRecord` y `firehose:PutRecordBatch` en todas las transmisiones de Firehose que tengan una etiqueta con una `LogDeliveryEnabled` clave con un valor de. `True` Esta etiqueta se adjunta automáticamente a una transmisión de Firehose al crear una suscripción para entregar los registros a Firehose.

Debe configurar los permisos para permitir que una entidad de IAM cree, edite o elimine un rol vinculado al servicio. Esta entidad puede ser un usuario, un grupo o un rol. Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Crear un rol vinculado a un servicio para Logs CloudWatch

No necesita crear manualmente un rol vinculado a un servicio. Cuando configuras los registros para que se envíen directamente a una transmisión de Firehose en la AWS Management Console, la o la AWS API AWS CLI, CloudWatch Logs crea el rol vinculado al servicio por ti.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando vuelves a configurar los registros para que se envíen directamente a una transmisión de Firehose, CloudWatch Logs vuelve a crear el rol vinculado al servicio para ti.

Edición de un rol vinculado a un servicio para Logs CloudWatch

CloudWatch Los registros no permiten editar AWSServiceRoleForLogDeliveryni ningún otro rol vinculado a un servicio después de crearlo. Dado que varias entidades pueden hacer referencia al rol, no puede cambiar su nombre después de crearlo. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Eliminar un rol vinculado a un servicio para Logs CloudWatch

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se monitorice ni mantenga de forma activa. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

> **Note**
>
> Si el servicio de CloudWatch registros utiliza el rol al intentar eliminar los recursos, es posible que la eliminación no se realice correctamente. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar CloudWatch los recursos de registros utilizados por el rol AWSServiceRoleForLogDeliveryvinculado al servicio

- Deja de enviar registros directamente a las transmisiones de Firehose.

Eliminación manual del rol vinculado a servicios mediante IAM

Utilice la consola de IAM AWS CLI, la o la AWS API para eliminar la función vinculada al AWSServiceRoleForLogDeliveryservicio. Para obtener más información, consulte Eliminar un rol vinculado al servicio

Funciones vinculadas al servicio Regions for Logs CloudWatch compatibles

CloudWatch Logs admite el uso de funciones vinculadas al servicio en todas las AWS regiones en las que el servicio está disponible. Para obtener más información, consulte CloudWatch Regiones y puntos finales de registros.

Validación de conformidad para Amazon CloudWatch Logs

Los auditores externos evalúan la seguridad y la conformidad de Amazon CloudWatch Logs como parte de varios programas de AWS conformidad. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para ver una lista de AWS los servicios incluidos en el ámbito de los programas de conformidad específicos, consulte AWS Servicios incluidos en el ámbito de aplicación por programa de conformidad y . Para obtener información general, consulte Programas de conformidad deAWS.

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

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

- Security and Compliance Quick Start Guides (Guías de inicio rápido de seguridad y conformidad) (Guías de inicio rápido de seguridad y conformidad): Estas guías de implementación analizan las consideraciones en materia de arquitectura y proporcionan los pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Diseño de arquitectura para la seguridad y el cumplimiento de la HIPAA en Amazon Web Services: este documento técnico describe cómo las empresas pueden utilizar AWS para crear aplicaciones compatibles con la HIPAA.
- AWS Recursos de cumplimiento Recursos de : esta colección de libros de trabajo y guías puede aplicarse a su sector y ubicación.
- Evaluación de los recursos con las reglas de la guía paraAWS Config desarrolladores: AWS Configevalúa en qué medida las configuraciones de sus recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar el cumplimiento de los estándares y las mejores prácticas del sector de la seguridad.

Resiliencia en Amazon CloudWatch Logs

La infraestructura global de AWS está conformada por regiones y zonas de disponibilidad de AWS. Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

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

Seguridad de la infraestructura en Amazon CloudWatch Logs

Como servicio gestionado, Amazon CloudWatch Logs 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 SeguridadAWS 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 los CloudWatch registros a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Uso de CloudWatch registros con puntos finales de VPC de interfaz

Si utiliza Amazon Virtual Private Cloud (Amazon VPC) para alojar sus AWS recursos, puede establecer una conexión privada entre su VPC y Logs. CloudWatch Puede utilizar esta conexión para enviar CloudWatch registros a Logs sin enviarlos a través de Internet.

Amazon VPC es un AWS servicio que puede utilizar para lanzar AWS recursos en una red virtual que usted defina. Con una VPC, puede controlar la configuración de la red, como el rango de direcciones IP, las subredes, las tablas de ruteo y las gateways de red. Para conectar la VPC a CloudWatch los registros, debe definir un punto final de VPC de interfaz para los registros. CloudWatch Este tipo de punto de enlace le permite conectar la VPC a los servicios de AWS . El punto final proporciona una conectividad fiable y escalable a CloudWatch Logs sin necesidad de una puerta de enlace a Internet, una instancia de traducción de direcciones de red (NAT) o una conexión VPN. Para obtener más información, consulte Qué es Amazon VPC en la Guía del usuario de Amazon VPC.

Los puntos finales de VPC de interfaz cuentan con una AWS tecnología que permite la comunicación privada entre AWS servicios mediante una interfaz de red elástica con direcciones IP privadas. AWS PrivateLink Para obtener más información, consulte Nuevo: AWS PrivateLink para AWS servicios.

Los siguientes pasos son para usuarios de Amazon VPC. Para obtener más información, consulte Introducción en la Guía del usuario de Amazon VPC.

Disponibilidad

CloudWatch Actualmente, Logs admite puntos finales de VPC en todas AWS las regiones, incluidas las regiones. AWS GovCloud (US)

Creación de un punto final de VPC para registros CloudWatch

Para empezar a usar CloudWatch los registros con la VPC, cree un punto de enlace de VPC de interfaz para los registros. CloudWatch El servicio que debe elegir es com.amazonaws.***región***.logs. No necesita cambiar ninguna configuración de Logs. CloudWatch Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Probar la conexión entre la VPC y los registros CloudWatch

Una vez creado el punto de conexión, puede probar la conexión.

Para probar la conexión entre la VPC CloudWatch y el punto de conexión de Logs

1. Conéctese a una instancia de Amazon EC2 que resida en la VPC. Para obtener más información acerca de la conexión, consulte Conexión con la instancia de Linux o Conexión con la instancia de Windows en la documentación de Amazon EC2.
2. Desde la instancia, úsala AWS CLI para crear una entrada de registro en uno de tus grupos de registros existentes.

   En primer lugar, cree un archivo JSON con un evento de registro. La marca temporal se debe especificar como el número de milisegundos después del 1 de enero de 1970 00:00:00 UTC.

```
[
  {
    "timestamp": 1533854071310,
    "message": "VPC Connection Test"
  }
```

```
]
```

A continuación, utilice el comando `put-log-events` para crear la entrada de registro:

```
aws logs put-log-events --log-group-name LogGroupName --log-stream-
name LogStreamName --log-events file://JSONFileName
```

Si la respuesta al comando incluye `nextSequenceToken`, el comando se ha realizado correctamente y el punto de enlace de la VPC funciona.

Controlar el acceso a su punto final CloudWatch de Logs VPC

Una política de punto de conexión de VPC es una política de recursos de IAM que puede asociar a un punto de conexión cuando crea o modifica el punto de conexión. Si no asocia una política al crear un punto de conexión, se asociará automáticamente una política predeterminada que conceda acceso completo al servicio. Una política de punto de conexión no anula ni sustituye a las políticas de IAM ni las políticas específicas del servicio. Se trata de una política independiente para controlar el acceso desde el punto de conexión al servicio especificado.

Las políticas de punto de conexión deben escribirse en formato JSON.

Para obtener más información, consulte Controlar el acceso a servicios con puntos de conexión de VPC en la Guía del usuario de Amazon VPC.

El siguiente es un ejemplo de una política de punto final para CloudWatch Logs. Esta política permite a los usuarios que se conectan a CloudWatch Logs a través de la VPC crear flujos de registros y enviar CloudWatch registros a Logs, y les impide realizar otras acciones de CloudWatch Logs.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
```

```
  ]
}
```

Para modificar la política de puntos finales de la VPC para los registros CloudWatch

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de navegación, elija Puntos de conexión.
3. Si aún no ha creado el punto de enlace para los CloudWatch registros, elija Crear punto de enlace. A continuación, seleccione com.amazonaws.***región***.logs y elija Create endpoint (Crear punto de enlace).
4. Seleccione el punto de enlace com.amazonaws.***región***.logs y elija la pestaña Policy (Política) en la mitad inferior de la pantalla.
5. Elija Editar política y realice los cambios en la política.

Compatibilidad con las claves de contexto de la VPC

CloudWatch Los registros admiten las claves `aws:SourceVpc` y de `aws:SourceVpce` contexto que pueden limitar el acceso a VPC específicas o puntos finales de VPC específicos. Estas claves funcionan solo cuando el usuario utiliza puntos de enlace de la VPC. Con el fin de obtener más información, consulte Claves disponibles para algunos servicios en la Guía del usuario de IAM.

El registro CloudWatch registra las operaciones de la API y la consola en AWS CloudTrail

Amazon CloudWatch Logs está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en CloudWatch Logs. CloudTrail captura las llamadas a la API realizadas por tu AWS cuenta o en su nombre. Las llamadas capturadas incluyen llamadas desde la CloudWatch consola y llamadas en código a las operaciones de la API CloudWatch Logs. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos de CloudWatch Logs. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puedes determinar la solicitud que se realizó a CloudWatch Logs, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, incluido cómo configurarla y habilitarla, consulta la Guía del AWS CloudTrail usuario.

Temas

CloudWatch Registra la información en CloudTrail

CloudTrail está activado en tu AWS cuenta al crear la cuenta. Cuando la actividad de eventos admitida se CloudWatch registra en los registros, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puedes ver, buscar y descargar los eventos recientes en tu AWS cuenta. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de CloudWatch los registros, crea un registro. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando crea una ruta en la consola, la ruta se aplica a todas AWS las regiones. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede

configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail Integraciones y servicios compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

CloudWatch Los registros permiten registrar las siguientes acciones como eventos en los archivos de CloudTrail registro:

- CancelExportTask
- CreateExportTask
- CreateLogGroup
- CreateLogStream
- DeleteDestination
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- PutDestination
- PutDestinationPolicy
- PutMetricFilter
- PutResourcePolicy
- PutRetentionPolicy
- PutSubscriptionFilter
- StartQuery
- StopQuery
- TestMetricFilter

Solo se inicia sesión en los elementos de solicitud CloudTrail para estas acciones de la API de CloudWatch registros:

- DescribeDestinations
- DescribeExportTasks
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeResourcePolicies
- DescribeSubscriptionFilters
- FilterLogEvents
- GetLogEvents
- GetLogGroupFields
- GetLogRecord
- GetQueryResults

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 las credenciales raíz o del usuario de IAM.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el elemento CloudTrail UserIdentity.

Información de generación de consultas en CloudTrail

CloudTrail También se admite el registro de eventos de la consola del generador de consultas. Actualmente, el generador de consultas es compatible con CloudWatch Logs Insights y CloudWatch Metric Insights. En estos CloudTrail casos, el `eventSource` es`monitoring.amazonaws.com`.

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la GenerateQueryacción en CloudWatch Logs Insights.


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId": "123456789012",
                "userName": "SAMPLE_NAME"
            },
            "attributes": {
                "creationDate": "2020-04-08T21:43:24Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "monitoring.amazonaws.com",
    "eventName": "GenerateQuery",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "exampleUserAgent",
    "requestParameters": {
        "query_ask": "***",
        "query_type": "LogsInsights",
        "logs_insights": {
            "fields": "***",
            "log_group_names": ["yourloggroup"]
        },
        "include_description": true
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
```

```
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Descripción de las entradas de los archivos de registro de

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

La siguiente entrada del archivo de registro muestra que un usuario ha activado la CreateExportTaskacción CloudWatch Registros.

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
```

```
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

CloudWatch Registra la referencia del agente

> **Important**
>
> Esta referencia es para el antiguo agente de CloudWatch Logs obsoleto. Si usa la versión 2 del Servicio de Metadatos de Instancia (IMDSv2), debe usar el nuevo agente unificado. CloudWatch Aunque no utilice IMDSv2, le recomendamos encarecidamente que utilice el CloudWatch agente unificado más reciente en lugar del antiguo agente de registros. Para obtener más información sobre el agente unificado más reciente, consulte Recopilación de métricas y registros de servidores locales y de instancias Amazon EC2 con el agente. CloudWatch
>
> Para obtener información sobre la migración del antiguo agente de CloudWatch Logs al agente unificado, consulte Crear el archivo de configuración del CloudWatch agente con el asistente.

El agente CloudWatch Logs proporciona una forma automatizada de enviar datos de registro a CloudWatch Logs desde instancias de Amazon EC2. El agente incluye los componentes siguientes:

- Un complemento AWS CLI que envía los datos de registro a CloudWatch Logs.
- Un script (daemon) que inicia el proceso para enviar datos a Logs. CloudWatch
- Un trabajo cron que garantiza que el daemon esté siempre en ejecución.

Archivo de configuración del agente

El archivo de configuración del agente de CloudWatch Logs describe la información que necesita el agente de CloudWatch Logs. La sección [general] del archivo de configuración del agente define las configuraciones comunes que se aplican a todos los flujos de registro. La sección [logstream] define la información necesaria para enviar un archivo local a un flujo de registros remoto. Puede tener más de una sección [logstream], pero cada una debe tener un nombre único en el archivo de configuración, por ejemplo, [logstream1], [logstream2], etc. El valor [logstream] junto con la primera línea de datos en el archivo de registro define la identidad del archivo de registro.

```
[general]
state_file = value
```

```
logging_config_file = value
use_gzip_http_content_encoding = [true | false]

[logstream1]
log_group_name = value
log_stream_name = value
datetime_format = value
time_zone = [LOCAL|UTC]
file = value
file_fingerprint_lines = integer | integer-integer
multi_line_start_pattern = regex | {datetime_format}
initial_position = [start_of_file | end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = integer
batch_count = integer
batch_size = integer

[logstream2]
...
```

state_file

Especifica dónde se almacena el archivo de estado.

logging_config_file

(Opcional) Especifica la ubicación del archivo de configuración de registro del agente. Si no especifica aquí un archivo de configuración de registro de agente, se utiliza el archivo de configuración. La ubicación predeterminada del archivo es `/var/awslogs/etc/awslogs.conf` si instaló el agente con un script y es `/etc/awslogs/awslogs.conf` si instaló el agente con rpm. El archivo está en formato de archivo de configuración de Python (https://docs.python.org/2/library/logging.config.html #logging-config-fileformat). Las funciones de registro con los nombres siguientes se pueden personalizar.

```
cwlogs.push
cwlogs.push.reader
cwlogs.push.publisher
cwlogs.push.event
cwlogs.push.batch
cwlogs.push.stream
cwlogs.push.watcher
```

El ejemplo siguiente cambia el nivel de lector y editor a WARNING mientras el valor por defecto es INFO.

```
[loggers]
keys=root,cwlogs,reader,publisher

[handlers]
keys=consoleHandler

[formatters]
keys=simpleFormatter

[logger_root]
level=INFO
handlers=consoleHandler

[logger_cwlogs]
level=INFO
handlers=consoleHandler
qualname=cwlogs.push
propagate=0

[logger_reader]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.reader
propagate=0

[logger_publisher]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.publisher
propagate=0

[handler_consoleHandler]
class=logging.StreamHandler
level=INFO
formatter=simpleFormatter
args=(sys.stderr,)

[formatter_simpleFormatter]
format=%(asctime)s - %(name)s - %(levelname)s - %(process)d - %(threadName)s -
 %(message)s
```

use_gzip_http_content_encoding

Si se establece en true (predeterminado), habilita la codificación de contenido http con gzip para enviar cargas comprimidas a CloudWatch Logs. Esto reduce el uso de la CPU y disminuye NetworkOut y disminuye la latencia de venta. Para deshabilitar esta función, añada use_gzip_http_content_encoding = false a la sección [general] del archivo de configuración del agente de CloudWatch registros y, a continuación, reinicie el agente.

> **Note**
>
> Esta configuración solo está disponible en la versión 1.3.3 o posterior de awscli-cwlogs.

log_group_name

Especifica el grupo de registro de destino. Un grupo de registro se crea automáticamente si no existe todavía. Los nombres de grupo de registros puede tener de 1 a 512 caracteres de longitud. Entre los caracteres permitidos se incluyen a-z, A-Z, 0-9, "_" (carácter de subrayado), "-" (guion), "/" (barra diagonal) y "." (punto).

log_stream_name

Especifica el flujo de registro de destino. Puede usar una cadena literal, variables predefinidas ({instance_id}, {hostname} y {ip_address}), o una combinación de ellas para definir el nombre del flujo de registro. Un flujo de registro se crea automáticamente si no existe todavía.

datetime_format

Especifica cómo se extrae la marca temporal de los registros. La marca temporal se utiliza para recuperar eventos de registro y generar métricas. Se utiliza la hora actual para cada evento de registro si no se proporciona datetime_format. Si el valor de datetime_format proporcionado no es válido para un mensaje de registro determinado, se utiliza la marca temporal (analizada correctamente) del último evento de registro. Si no existen eventos de registro anteriores, se utiliza la hora actual.

Los códigos datetime_format comunes se enumeran a continuación. También puede utilizar cualquier código datetime_format que admita Python, datetime.strptime (). El desfase de la zona horaria (%z) también se admite aunque no se ha admitido hasta python 3.2, [+ -] HHMM sin dos puntos (:). Para obtener más información, consulte strftime() and strptime() Behavior.

%y: año sin siglo como número decimal rellenado con ceros. 00, 01, ..., 99

%Y: año con siglo como número decimal. 1970, 1988, 2001, 2013

%b: mes como nombre abreviado de configuración regional. Ene, Feb, ..., Dic (es_ES);

%B: mes como nombre completo de configuración regional. enero, febrero,..., diciembre (es_ES);

%m: mes como número decimal rellenado con ceros. 01, 02, ..., 12

%d: día del mes como número decimal rellenado con ceros. 01, 02, ..., 31

%H: hora (formato de 24 horas) como número decimal rellenado con ceros. 00, 01, ..., 23

%I: hora (formato de 12 horas) como número decimal rellenado con ceros. 01, 02, ..., 12

%p: equivalente de la configuración regional a AM o PM.

%M: minutos como número decimal rellenado con ceros. 00, 01, ..., 59

%S: segundos como número decimal rellenado con ceros. 00, 01, ..., 59

%f: microsegundos como número decimal, rellenado con ceros a la izquierda. 000000, ..., 999999

%z: desplazamiento UTC en la forma +HHMM o -HHMM. +0000, -0400, +1030

Formatos de ejemplo:

```
Syslog: '%b %d %H:%M:%S', e.g. Jan 23 20:59:29

Log4j: '%d %b %Y %H:%M:%S', e.g. 24 Jan 2014 05:00:00

ISO8601: '%Y-%m-%dT%H:%M:%S%z', e.g. 2014-02-20T05:20:20+0000
```

time_zone

Especifica la zona horaria de la marca temporal de evento de registro. Los dos valores admitidos son UTC y LOCAL. El valor predeterminado es LOCAL, que se utiliza en caso de que la zona horaria no se pueda determinar a partir de datetime_format.

archivo

Especifica los archivos de registro que desea insertar en Logs. CloudWatch File puede apuntar a un archivo específico o a varios archivos (utilizando comodines como /var/log/system.log*). Solo el archivo más reciente se envía a los CloudWatch registros en función de la hora de modificación del archivo. Le recomendamos que utilice comodines para especificar una serie de archivos del mismo tipo, como access_log.2014-06-01-01, access_log.2014-06-01-02, etc., pero no varios tipos de archivos, como por ejemplo access_log_80 y access_log_443. Para especificar varios

tipos de archivos, agregue otra entrada de flujo de registro al archivo de configuración para que cada tipo de archivo de registro vaya a un flujo de registros distinto. Los archivos comprimidos no son compatibles.

file_fingerprint_lines

Especifica el intervalo de líneas para identificar un archivo. Los valores admitidos son un número o dos números delimitados por guion, como, por ejemplo, "1", "2-5". El valor predeterminado es "1" de modo que se utiliza la primera línea para calcular la huella. Las líneas de huellas digitales no se envían a CloudWatch los registros a menos que estén disponibles todas las líneas especificadas.

multi_line_start_pattern

Especifica el patrón para identificar el inicio de un mensaje de registro. Un mensaje de registro consta de una línea que coincide con el patrón y de líneas siguientes que no coinciden con el patrón. Los valores válidos son expresiones regulares o {datetime_format}. Cuando se utiliza {datetime_format}, se debe especifica la opción datetime_format. El valor predeterminado es "^[^ \s]" de modo que cualquier línea que comience con un carácter sin espacios en blanco cierra el mensaje de registro anterior y comienza un nuevo mensaje de registro.

initial_position

Especifica dónde empezar a leer datos (start_of_file o end_of_file). El valor predeterminado es start_of_file. Se utiliza únicamente si no se almacena de forma persistente ningún estado para dicho flujo de registro.

encoding

Especifica la codificación del archivo de registro, de modo que el archivo se pueda leer correctamente. El valor predeterminado es utf_8. Se pueden utilizar aquí las codificaciones compatibles con Python codecs.decode().

> **Warning**
>
> La especificación de una codificación incorrecta podría provocar pérdida de datos porque los caracteres que no se pueden descodificar se sustituirán por otro carácter.

A continuación se muestran las codificaciones comunes:

```
ascii, big5, big5hkscs, cp037, cp424, cp437, cp500, cp720, cp737,
cp775, cp850, cp852, cp855, cp856, cp857, cp858, cp860, cp861, cp862,
```

```
cp863, cp864, cp865, cp866, cp869, cp874, cp875, cp932, cp949, cp950,
cp1006, cp1026, cp1140, cp1250, cp1251, cp1252, cp1253, cp1254, cp1255,
cp1256, cp1257, cp1258, euc_jp, euc_jis_2004, euc_jisx0213, euc_kr,
gb2312, gbk, gb18030, hz, iso2022_jp, iso2022_jp_1, iso2022_jp_2,
iso2022_jp_2004, iso2022_jp_3, iso2022_jp_ext, iso2022_kr, latin_1,
iso8859_2, iso8859_3, iso8859_4, iso8859_5, iso8859_6, iso8859_7,
iso8859_8, iso8859_9, iso8859_10, iso8859_13, iso8859_14, iso8859_15,
iso8859_16, johab, koi8_r, koi8_u, mac_cyrillic, mac_greek, mac_iceland,
mac_latin2, mac_roman, mac_turkish, ptcp154, shift_jis, shift_jis_2004,
shift_jisx0213, utf_32, utf_32_be, utf_32_le, utf_16, utf_16_be,
utf_16_le, utf_7, utf_8, utf_8_sig
```

buffer_duration

Especifica la duración para agrupar en lotes eventos de registro. El valor mínimo es 5000ms y valor predeterminado es 5000ms.

batch_count

Especifica el número máximo de eventos de registro en un lote, hasta 10 000. El valor predeterminado es 10 000.

batch_size

Especifica el tamaño máximo de eventos de registro en un lote, en bytes, hasta 1 048 576 bytes. El valor predeterminado es de 1 048 576 bytes. Este tamaño se calcula como la suma de todos los mensajes de eventos en UTF-8, más 26 bytes para cada evento de registro.

Uso del agente CloudWatch de registros con proxies HTTP

Puede utilizar el agente de CloudWatch registros con proxies HTTP.

> **Note**
>
> Los proxies HTTP son compatibles con la versión 1.3.8 o posterior de awslogs-agent-setup .py.

Para usar el agente CloudWatch Logs con proxies HTTP

1. Realice una de las siguientes acciones siguientes:

a. Para una nueva instalación del agente CloudWatch Logs, ejecute los siguientes comandos:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
sudo python awslogs-agent-setup.py --region us-east-1 --http-proxy http://your/proxy --https-proxy http://your/proxy --no-proxy 169.254.169.254
```

Para mantener el acceso al servicio de metadatos de Amazon EC2 en instancias EC2, utilice --no-proxy 169.254.169.254 (recomendado). Para obtener más información, consulte Metadatos de instancia y datos de usuario en la Guía del usuario de Amazon EC2.

En el valor de `http-proxy` y `https-proxy`, especifique la URL completa.

b. Para una instalación existente del agente CloudWatch Logs, edite /var/awslogs/etc/proxy.conf y añada sus proxies:

```
HTTP_PROXY=
HTTPS_PROXY=
NO_PROXY=
```

2. Reinicie el agente para que los cambios surtan efecto:

```
sudo service awslogs restart
```

Si está utilizando Amazon Linux 2, utilice el comando siguiente para reiniciar el agente:

```
sudo service awslogsd restart
```

CloudWatch Compartimentación de los archivos de configuración del agente Logs

Si utiliza la versión 1.3.8 o posterior awslogs-agent-setup .py con awscli-cwlogs 1.3.3 o posterior, puede importar diferentes configuraciones de transmisión para varios componentes de forma independiente mediante la creación de archivos de configuración adicionales en el directorio /var/awslogs/etc/config/. CloudWatch Cuando se inicia el agente Logs, incluye cualquier configuración de transmisión en estos archivos de configuración adicionales. Las propiedades de configuración

en la sección [general] deben definirse en el archivo de configuración principal (/var/awslogs/etc/awslogs.conf) y se omiten en los archivos de configuración adicionales que se encuentran en /var/awslogs/etc/config/.

Si no dispone de un directorio /var/awslogs/etc/config/ dado que ha instalado el agente con rpm, puede utilizar en su lugar el directorio /etc/awslogs/config/.

Reinicie el agente para que los cambios surtan efecto:

```
sudo service awslogs restart
```

Si está utilizando Amazon Linux 2, utilice el comando siguiente para reiniciar el agente:

```
sudo service awslogsd restart
```

CloudWatch Preguntas frecuentes sobre el agente de registros

¿Qué tipo de rotaciones de archivo se admiten?

Se admiten los siguientes mecanismos de rotación de archivos:

- Cambiar el nombre de los archivos de registro existentes por un sufijo numérico y, a continuación, volver a crear el archivo de registro vacío original. Por ejemplo, el nombre de /var/log/syslog.log se cambia a /var/log/syslog.log.1. Si ya existe /var/log/syslog.log.1 de una rotación anterior, se cambia el nombre a /var/log/syslog.log.2.
- Truncar el archivo de registro original en vigor después de crear una copia. Por ejemplo, /var/log/syslog.log se copia a /var/log/syslog.log.1 y /var/log/syslog.log se trunca. Podría haber pérdida de datos en este caso, por tanto tenga cuidado a la hora de utilizar este mecanismo de rotación de archivo.
- Creación de un nuevo archivo con un patrón común como el antiguo. Por ejemplo, se mantiene /var/log/syslog.log.2014-01-01 y se crea /var/log/syslog.log.2014-01-02.

La huella (ID de origen) del archivo se calcula mediante el hash de la clave del flujo de registro y la primera línea de contenido del archivo. Para omitir este comportamiento, se puede utilizar la opción file_fingerprint_lines. Cuando se produce la rotación de archivos, el nuevo archivo se supone que tiene nuevo contenido y el archivo antiguo no se supone que tenga contenido añadido; el agente envía el nuevo archivo una vez que termine la lectura del antiguo.

¿Cómo puedo determinar la versión del agente que estoy utilizando?

Si ha utilizado un script de configuración para instalar el agente de CloudWatch Logs, puede utilizar el archivo /var/awslogs/bin/awslogs-version.sh para comprobar qué versión del agente está utilizando. Imprime la versión del agente y sus dependencias principales. Si usaste yum para instalar el agente de CloudWatch Logs, puedes usar «yum info awslogs» y «yum info aws-cli-plugin-cloudwatch -logs» para comprobar la versión del agente y el plugin de Logs. CloudWatch

¿Cómo se convierten las entradas de registro a eventos de registro?

Los eventos de registro contienen dos propiedades: la marca temporal de cuando se produjo el evento y el mensaje de registro sin procesar. De forma predeterminada, cualquier línea que comience con un carácter sin espacios en blanco cierra el mensaje de registro anterior si lo hay y comienza un nuevo mensaje de registro. Para anular este comportamiento, se puede usar multi_line_start_pattern y todas las líneas que coincidan con el patrón inician un nuevo mensaje de registro. El patrón podría ser cualquier regex o "{datetime_format}". Por ejemplo, si la primera línea de cada mensaje de registro contiene una marca temporal como “2014-01-02T13:13:01Z”, multi_line_start_pattern se puede establecer en “\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z”. Para simplificar la configuración, la variable "{datetime_format}" variable se puede utilizar si se especifica datetime_format option. Para el mismo ejemplo, si datetime_format se establece en "%Y-%m-%dT%H:%M:%S%z", entonces el patrón multi_line_start_pattern podría ser sencillamente "{datetime_format}".

Se utiliza la hora actual para cada evento de registro si no se proporciona datetime_format. Si el valor de datetime_format proporcionado no es válido para un mensaje de registro determinado, se utiliza la marca temporal (analizada correctamente) del último evento de registro. Si no existen eventos de registro anteriores, se utiliza la hora actual. Se registra un mensaje de advertencia cuando un evento de registro utiliza la hora actual o la hora del evento de registro anterior.

Las marcas temporales se utilizan para recuperar eventos de registro y generar métricas, por lo que si especifica el formato equivocado, los eventos de registro no podrían recuperarse y podrían generar métricas erróneas.

¿Cómo se agrupan en lotes los eventos de registro?

Un lote se completa y se publica cuando cumple alguna de las siguientes condiciones:

1. La cantidad de tiempo de buffer_duration que ha transcurrido desde que se agregó el primer evento de registro.
2. Se ha acumulado un valor inferior a batch_size para eventos de registro, pero al agregar el nuevo evento de registro se supera el valor de batch_size.

3. El número de eventos de registro ha alcanzado el valor batch_count.
4. Los eventos de registro del lote no abarcan más de 24 horas, pero al añadir el nuevo evento de registro se supera la restricción de 24 horas.

¿Qué provocaría la omisión o el truncamiento de las entradas de registro, los eventos de registro o los lotes?

Para seguir la restricción de la operación `PutLogEvents`, los siguientes problemas podrían provocar la omisión de un evento de registro o lote.

> **Note**
>
> El agente de CloudWatch Logs escribe una advertencia en su registro cuando se omiten datos.

1. Si el tamaño de un evento de registro es superior a 256 KB, el evento de registro se omitirá por completo.
2. Si la marca temporal del evento de registro es de más de 2 horas en el futuro, se omitirá el evento de registro.
3. Si la marca temporal del evento de registro es de más de 14 días en el pasado, se omitirá el evento de registro.
4. Si cualquier evento de registro es más antiguo que el periodo de retención del grupo de registro, se omitirá todo el lote.
5. Si el lote de eventos de registro en una solicitud `PutLogEvents` única abarca más de 24 horas, la operación `PutLogEvents` falla.

¿Provoca la parada del agente la pérdida de datos/duplicados?

No siempre y cuando el archivo de estado esté disponible y no se haya producido la rotación de ningún archivo desde la última ejecución. El agente de CloudWatch registros puede empezar desde donde se detuvo y continuar insertando los datos de registro.

¿Puedo señalar a diferentes archivos de registro desde el mismo host o diferentes al mismo flujo de registro?

No se admite configurar varias fuentes de registro para enviar datos a un único flujo de registro.

¿Qué llamadas al API realiza el agente (o qué acciones debo agregar a mi política de IAM)?

El agente de CloudWatch registros requiere las `PutLogEvents` operaciones `CreateLogGroup` `CreateLogStreamDescribeLogStreams`,, y. Si está utilizando el último agente, no es necesario `DescribeLogStreams`. Consulte la política de IAM de ejemplo a continuación.

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

No quiero que el agente de CloudWatch registros cree grupos de registros ni flujos de registros automáticamente. ¿Cómo puedo evitar que el agente vuelva a crear grupos de registro y flujos de registro?

En la política de IAM, puede limitar el agente solo a las siguientes operaciones: `DescribeLogStreams`, `PutLogEvents`.

Antes de revocar los permisos `CreateLogStream` y `CreateLogGroup` del agente, asegúrese de crear los grupos de registro y las secuencias de registro que desee que utilice el agente. El agente de registros no puede crear secuencias de registro en un grupo de registros que haya creado a menos que tenga los permisos `CreateLogStream` y `CreateLogGroup`.

¿Qué registros debería examinar durante la resolución de problemas?

El registro del agente de instalación se encuentra en `/var/log/awslogs-agent-setup.log` y el registro del agente, en `/var/log/awslogs.log`.

Monitorización con CloudWatch métricas

CloudWatch Logs envía métricas a Amazon CloudWatch cada minuto.

CloudWatch Registra las métricas

El espacio de nombres de `AWS/Logs` incluye las siguientes métricas.

Métrica	Descripción
`CallCount`	El número de operaciones de la API especificadas realizadas en su cuenta. `CallCount` es una métrica de uso del servicio de CloudWatch registros . Para obtener más información, consulte CloudWatch Registra las métricas de uso del servicio. Dimensiones válidas: clase, recurso, servicio, tipo Estadísticas válidas: suma Unidades: ninguna
`DeliveryErrors`	El número de eventos de registro en los que CloudWatch Logs recibió un error al reenviar datos al destino de la suscripción. Si el servicio de destino devuelve un error que se puede volver a intentar, como una excepción de limitación o una excepción de servicio que se puede volver a intentar (HTTP 5xx, por ejemplo), CloudWatch Logs seguirá reintentando la entrega durante un máximo de 24 horas. CloudWatc h Logs no intenta volver a realizar la entrega si el error no se puede volver a intentar, como o. `AccessDeniedException` `ResourceN otFoundException` Dimensiones válidas:,,, LogGroupName DestinationType FilterName PolicyLevel Estadísticas válidas: suma

Métrica	Descripción
	Unidades: ninguna
`DeliveryThrottling`	El número de eventos de registro por los que se CloudWatch limitó Logs al reenviar los datos al destino de la suscripción. Si el servicio de destino devuelve un error que se puede volver a intentar, como una excepción de limitación o una excepción de servicio que se puede volver a intentar (HTTP 5xx, por ejemplo), CloudWatch Logs seguirá reintentando la entrega durante un máximo de 24 horas. CloudWatch Logs no intenta volver a realizar la entrega si el error no se puede volver a intentar, como o. `AccessDeniedException` `ResourceNotFoundException` Dimensiones válidas:,,, LogGroupName DestinationType FilterName PolicyLevel Estadísticas válidas: suma Unidades: ninguna
`EMFParsingErrors`	El número de errores de análisis encontrados al procesar los registros de formato de métrica integrada. Estos errores se producen cuando los registros se identifican como un formato de métrica integrada, pero no siguen el formato correcto. Para obtener más información sobre el formato de las métricas integradas, consulte Especificación: formato de métricas integradas. Dimensiones válidas: `LogGroupName` Estadísticas válidas: suma Unidades: ninguna

Métrica	Descripción
`EMFValidationErrors`	El número de errores de validación encontrados al procesar los registros de formato de métrica integrada. Estos errores se producen cuando las definiciones de métricas en los registros de formato de métricas integradas no se adhieren a las especificaciones de `MetricDatum` y al formato de métrica integrada. Para obtener información sobre el formato métrico CloudWatch integrado, consulte Especificación: formato métrico integrado. Para obtener información sobre el tipo de datos`MetricDatum`, consulta MetricDatumla referencia de la CloudWatch API de Amazon. **Note** Ciertos errores de validación pueden provocar que no se publiquen varias métricas dentro de un registro EMF. Por ejemplo, se eliminarán todas las métricas configuradas con un espacio de nombres no válido. Dimensiones válidas: `LogGroupName` Estadísticas válidas: suma Unidades: ninguna
`ErrorCount`	El número de operaciones de la API realizadas en su cuenta que dieron lugar a errores. `ErrorCount` es una métrica de uso del servicio de CloudWatch registros. Para obtener más información, consulte CloudWatch Registra las métricas de uso del servicio. Dimensiones válidas: clase, recurso, servicio, tipo Estadísticas válidas: suma Unidades: ninguna

Métrica	Descripción
`ForwardedBytes`	El volumen de eventos de registro en bytes comprimidos reenviados al destino de la suscripción. Dimensiones válidas: LogGroupName, DestinationType, FilterName Estadísticas válidas: suma Unidades: bytes
`ForwardedLogEvents`	El número de eventos de registro reenviados al destino de la suscripción. Dimensiones válidas: LogGroupName, DestinationType, FilterName, PolicyLevel Estadísticas válidas: suma Unidades: ninguna
`IncomingBytes`	El volumen de eventos de registro en bytes sin comprimir subidos a CloudWatch los registros. Cuando se utiliza con la dimensión `LogGroupName` , es el volumen de eventos de registro en bytes descomprimidos cargados en el grupo de registros. Dimensiones válidas: LogGroupName Estadísticas válidas: suma Unidades: bytes
`IncomingLogEvents`	El número de eventos de registro cargados en los CloudWatch registros . Cuando se utiliza con la dimensión `LogGroupName` , es el número de eventos de registro cargados en el grupo de registros. Dimensiones válidas: LogGroupName Estadísticas válidas: suma Unidades: ninguna

Métrica	Descripción
`LogEvents WithFindings`	El número de eventos de registro que coincidieron con una cadena de datos que está auditando mediante la función de protección de datos de CloudWatch registros. Para obtener más información, consulte Ayude a proteger los datos de registro confidenciales con el enmascaramiento. Dimensiones válidas: None Estadísticas válidas: suma Unidades: ninguna
`ThrottleCount`	El número de operaciones de la API realizadas en su cuenta a las que se aplicó una limitación controlada debido a las cuotas de utilización. `ThrottleCount` es una métrica de uso del servicio de CloudWatch registros. Para obtener más información, consulte CloudWatch Registra las métricas de uso del servicio. Dimensiones válidas: clase, recurso, servicio, tipo Estadísticas válidas: suma Unidades: ninguna

Dimensiones de las métricas CloudWatch de Logs

Las dimensiones que puedes usar con las métricas de CloudWatch Logs se muestran en la siguiente tabla.

Dimensión	Descripción
`LogGroupName`	El nombre del grupo de CloudWatch registros para el que se muestran las métricas.

Dimensión	Descripción
`DestinationType`	El destino de la suscripción para los datos de CloudWatch Logs, que puede ser AWS Lambda Amazon Kinesis Data Streams o Amazon Data Firehose.
`FilterName`	El nombre del filtro de suscripción que reenvía datos desde el grupo de registros al destino. El nombre del filtro de suscripci ón se convierte automáticamente en ASCII y CloudWatch los caracteres no admitidos se sustituyen por un signo de interroga ción (?).

Las dimensiones de las métricas relacionadas con los filtros de suscripción a nivel de cuenta se muestran en la siguiente tabla.

Dimensión	Descripción
`PolicyLevel`	El nivel en el que se aplica la política. Actualmente, el único valor válido para esta dimensión es `AccountPolicy`
`DestinationType`	El destino de la suscripción para los datos de CloudWatch Logs, que puede ser AWS Lambda Amazon Kinesis Data Streams o Amazon Data Firehose.
`FilterName`	El nombre del filtro de suscripción que reenvía datos desde el grupo de registros al destino. El nombre del filtro de suscripci ón se convierte automáticamente en ASCII y CloudWatch los caracteres no admitidos se sustituyen por un signo de interroga ción (?).

CloudWatch Registra las métricas de uso del servicio

CloudWatch Logs envía métricas CloudWatch que rastrean el uso de las operaciones de la API CloudWatch Logs. 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, consulte Integración y métricas de utilización de Service Quotas.

Por ejemplo, puede realizar un seguimiento de la métrica `ThrottleCount` o establecer una alarma en esa métrica. Si el valor de esta métrica aumenta, debe considerar la posibilidad de solicitar un aumento de cuota para la operación de la API que se limita. Para obtener más información sobre las cuotas del servicio de CloudWatch registros, consulte[CloudWatch Cuotas de registros](#).

CloudWatch Logs publica las métricas de uso de la cuota de servicio cada minuto tanto en el espacio de nombres como en el `AWS/Usage` espacio de `AWS/Logs` nombres.

En la siguiente tabla, se muestran las métricas de uso del servicio publicadas por CloudWatch Logs. Estas métricas no tienen una unidad especificada. La estadística más útil para estas métricas es `SUM`, que representa el recuento total de operaciones para el periodo de 1 minuto.

Cada una de estas métricas se publica con valores para todas las dimensiones `Service`, `Class`, `Type` y `Resource`. También se publican con una sola dimensión llamada `Account Metrics`. Utilice la dimensión `Account Metrics` a fin de ver la suma de métricas para todas las operaciones de la API de su cuenta. Utilice las otras dimensiones y especifique el nombre de una operación de la API para la dimensión `Resource` a fin de encontrar las métricas de esa API en particular.

Métricas

Métrica	Descripción
`CallCount`	El número de operaciones especificadas realizadas en su cuenta. `CallCount` se publica en los espacios de nombres `AWS/Usage` y `AWS/Logs`.
`ErrorCount`	El número de operaciones de la API realizadas en su cuenta que dieron lugar a errores. `ErrorCount` se publica solo en `AWS/Logs`.
`ThrottleCount`	El número de operaciones de la API realizadas en su cuenta a las que se aplicó una limitación controlada debido a las cuotas de utilización. `ThrottleCount` se publica solo en `AWS/Logs`.

Dimensiones

Dimensión	Descripción
`Account metrics`	Usa esta dimensión para obtener una suma de la métrica en todas las API de CloudWatch Logs. Si desea ver las métricas de una API en particular, utilice las otras dimensiones enumeradas en esta tabla y especifique el nombre de la API como el valor de `Resource`.
`Service`	El nombre del AWS servicio que contiene el recurso. En el CloudWatch caso de las métricas de uso de los registros, el valor de esta dimensión es`Logs`.
`Class`	La clase de recurso que se está rastreando. CloudWatch Las métricas de uso de la API de registros utilizan esta dimensión con un valor de`None`.
`Type`	El tipo de recurso al que se realiza el seguimiento. Actualmente, cuando la dimensión `Service` es `Logs`, el único valor válido para `Type` es `API`.
`Resource`	El nombre de la operación de la API. Los valores válidos incluyen todos los nombres de operaciones de la API que se enumeran en Actions (Acciones). Por ejemplo, `PutLogEvents`

CloudWatch Cuotas de registros

En las tablas siguientes se muestran las cuotas de servicio predeterminadas, también denominadas límites, para CloudWatch los registros de una AWS cuenta. La mayoría de estas cuotas de servicio, pero no todas, aparecen en el espacio de nombres Amazon CloudWatch Logs de la consola Service Quotas. Para solicitar un aumento para dichas cuotas, consulte el procedimiento más adelante en esta sección.

Recurso	Cuota predeterminada
Políticas a nivel de cuenta	Una política de filtrado de suscripciones a nivel de cuenta por cuenta. Una política de protección de datos a nivel de cuenta por cuenta. Estas cuotas no se pueden cambiar.
Detectores de anomalías	10 detectores de anomalías por cuenta. Esta cuota no se puede cambiar.
Tamaño de lote	El tamaño máximo del lote es 1 048 576 bytes. Este tamaño se calcula como la suma de todos los mensajes de eventos en UTF-8, más 26 bytes para cada evento de registro. Esta cuota no se puede cambiar.
Archivado de datos	Hasta 5 GB de archivado de datos de forma gratuita. Esta cuota no se puede cambiar.
CreateLogGroup	10 transacciones por segundo (TPS/cuenta/región), tras las cuales se limitan las transacciones. Puede solicitar un aumento de cuota.
CreateLogStream	50 transacciones por segundo (TPS/cuenta/región), después de lo cual se limitarán las transacciones. Puede solicitar un aumento de cuota.

Recurso	Cuota predeterminada
Identificadores de datos personalizados	Cada política de protección de datos puede incluir hasta 10 identificadores de datos personalizados. Puede solicitar un aumento de cuota. Cada expresión regular que define un identificador de datos personalizado puede incluir hasta 200 caracteres. Esta cuota no se puede cambiar.
DeleteLogGroup	10 transacciones por segundo (TPS/cuenta/región), tras las cuales se limitan las transacciones. Puede solicitar un aumento de cuota.
DeleteLogStream	15 transacciones por segundo (TPS/cuenta/región), tras las cuales se limitan las transacciones. Puede solicitar un aumento de cuota.
DescribeLogGroups	10 transacciones por segundo (TPS/cuenta/región). Puede solicitar un aumento de cuota.
DescribeLogStreams	25 transacciones por segundo (TPS/cuenta/región). Puede solicitar un aumento de cuota.
Campos de registro detectados	CloudWatch Logs Insights puede detectar un máximo de 1000 campos de eventos de registro en un grupo de registros. Esta cuota no se puede cambiar. Para obtener más información, consulte Registros y campos detectados compatibles.
Campos de registro extraídos en registros JSON	CloudWatch Logs Insights puede extraer un máximo de 200 campos de eventos de registro de un registro JSON. Esta cuota no se puede cambiar. Para obtener más información, consulte Registros y campos detectados compatibles.

Recurso	Cuota predeterminada
Exportar tarea	Una tarea de exportación (activa o pendiente) a la vez, por cuenta. Esta cuota no se puede cambiar.
FilterLogEvents	25 solicitudes por segundo en Este de EE. UU. (Norte de Virginia). 5 solicitudes por segundo en las siguientes regiones: • Asia-Pacífico (Yakarta) • Asia-Pacífico (Osaka) • Europa (Fráncfort) • Oeste de Canadá (Calgary) • Israel (Tel Aviv) 10 solicitudes por segundo en otras regiones. Esta cuota no se puede cambiar.

Recurso	Cuota predeterminada
GetLogEvents	30 solicitudes por segundo en Europa (París). 10 solicitudes por segundo en las siguientes regiones: • Oeste de EE. UU. (Oregón) • Asia-Pacífico (Yakarta) • Asia-Pacífico (Osaka) • Oeste de Canadá (Calgary) • Europa (Irlanda) • Europa (Fráncfort) • Israel (Tel Aviv) 25 solicitudes por segundo en todas las demás regiones. Esta cuota no se puede cambiar. Recomendamos las suscripciones si continuamente está procesando datos nuevos. Si necesita datos históricos, recomendamos exportarlos a Amazon S3.
Datos de entrada	Hasta 5 GB de datos de entrada de forma gratuita. Esta cuota no se puede cambiar.
Sesiones simultáneas de Live Tail.	15 sesiones simultáneas. Puede solicitar un aumento de cuota.
Live Tail: grupos de registro que se buscan en una sesión.	Se permite un máximo de 10 grupos de registro analizados en una sesión de Live Tail. Esta cuota no se puede cambiar.
Tamaño del evento de registro	256 KB (máximo). Esta cuota no se puede cambiar.

Recurso	Cuota predeterminada
Grupos de registro	1 000 000 de grupos de registro por cuenta y región. Puede solicitar un aumento de cuota. No hay cuotas en el número de flujos de registro que pueden pertenecer a un grupo de registros.
Filtros de métricas	100 por grupo de registros. Esta cuota no se puede cambiar.
Métricas de formato métrico integradas	100 métricas por evento de registro y 30 dimensiones por métrica. Para obtener más información sobre el formato métrico integrado, consulte Especificación: formato métrico integrado en la Guía del CloudWatch usuario de Amazon.
PutLogEvents	El tamaño máximo de lote de una PutLogEvents solicitud es de 1 MB. Este tamaño se calcula como la suma de todos los mensajes de eventos en UTF-8, más 26 bytes para cada evento de registro. 5000 transacciones por segundo por cuenta y región Puede solicitar un aumento de la cuota limitada por segundo mediante el servicio. Service Quotas
Tiempo de espera de ejecución de la consulta	Las consultas de CloudWatch Logs Insights caducan después de 60 minutos. Este límite de tiempo no se puede cambiar.
Grupos de registros consultados	Se puede consultar un máximo de 50 grupos de registros en una sola consulta de CloudWatch Logs Insights. Esta cuota no se puede cambiar.

Recurso	Cuota predeterminada
Simultaneidad de consultas	Para los grupos de registros de clase estándar, un máximo de 30 consultas simultáneas de CloudWatch Logs Insights, incluidas las consultas que se hayan agregado a los paneles. Para los grupos de registros de la clase Infrequent Access, un máximo de 5 consultas simultáneas de CloudWatch Logs Insights, incluidas las consultas que se hayan agregado a los paneles. Estas cuotas no se pueden cambiar.
Consultas generadas a partir de lenguaje natural	Hasta cinco solicitudes de consulta simultáneas generadas en lenguaje natural.
Disponibilidad de consultas	Las consultas creadas en la consola están disponibles durante 30 días mediante el comando Historial. Este período de disponibilidad no se puede cambiar. Las definiciones de consulta creadas mediante el uso PutQueryDefinitionno caducan.
Disponibilidad de los resultados de la consulta	Los resultados de una consulta se pueden recuperar durante 7 días. Este tiempo de disponibilidad no se puede cambiar.
Resultados de consultas mostrados en la consola	De forma predeterminada, el número máximo de filas que aparecen en los resultados de la consulta de la consola es 1000. Puede utilizar el comando **`limit`** de una consulta para aumentar este valor hasta las 10 000 filas. Para obtener más información, consulte CloudWatch Sintaxis de consultas de Logs Insights.

Recurso	Cuota predeterminada
Expresiones regulares	Cuando se crean filtros de métricas o filtros de suscripción, es posible especificar hasta 5 patrones de filtrado que incluyan expresiones regulares para cada grupo de registros. Esta cuota no se puede cambiar. Hasta 2 expresiones regulares para cada patrón de filtro, al crear un patrón de filtro delimitado o JSON para filtros métricos y filtros de suscripción o al filtrar eventos de registro.
Políticas de recursos	Políticas de recursos de hasta 10 CloudWatch registros por región y cuenta. Esta cuota no se puede cambiar.
Consultas guardadas	Puede guardar hasta 1000 consultas de CloudWatch Logs Insights por región y cuenta. Esta cuota no se puede cambiar.
Filtros de suscripción	2 por grupo de registro. Esta cuota no se puede cambiar.

Administrar tus cuotas CloudWatch de servicio de registros

CloudWatch Logs se ha integrado con Service Quotas, un AWS servicio que le permite ver y gestionar sus cuotas desde una ubicación central. Para obtener más información, consulte ¿Qué son las Service Quotas? en la Guía del usuario de Service Quotas.

Service Quotas facilita la búsqueda del valor de las cuotas de servicio de CloudWatch Logs.

AWS Management Console

Para ver las cuotas del servicio de CloudWatch registros mediante la consola

1. Abra la consola de Service Quotas en https://console.aws.amazon.com/servicequotas/.
2. En el panel de navegación, elija Servicios de AWS .
3. En la lista de AWS servicios, busca y selecciona Amazon CloudWatch Logs.

En la lista Service Quotas, puede ver el nombre de la Service Quota, el valor aplicado (si está disponible), la cuota predeterminada de AWS y si el valor de cuota es ajustable.

4. Para ver información adicional sobre una cuota de servicio, como, por ejemplo, la descripción, elija el nombre de cuota.
5. (Opcional) Para solicitar un aumento de cuota, seleccione la cuota que desea aumentar, seleccione Solicitar aumento de cuota, escriba o seleccione la información necesaria y seleccione Solicitar.

Para trabajar más con Service Quotas mediante la consola, consulte la Guía del usuario de Service Quotas. Para solicitar un aumento de cuota, consulte Solicitud de un aumento de cuota en la Guía de usuario de Service Quotas.

AWS CLI

Para ver las cuotas del servicio de CloudWatch Logs utilizando el AWS CLI

Ejecute el siguiente comando para ver las cuotas de CloudWatch registros predeterminadas.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code logs \
    --output table
```

Para trabajar más con las cuotas de servicio mediante el AWS CLI, consulte la Referencia de AWS CLI comandos de Service Quotas. Para solicitar un aumento de cuota, consulte el `request-service-quota-increase` comando en la Referencia de comandos de la AWS CLI.

Historial de documentos

La siguiente tabla describe los cambios importantes en cada versión de la Guía del usuario de CloudWatch Logs, a partir de junio de 2018. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
CloudWatch El soporte de Logs Insights para la generación de consultas en lenguaje natural está disponible de forma general	CloudWatch Logs Insights admite el lenguaje natural para generar y actualizar consultas. Para obtener más información, consulte Usar lenguaje natural para generar y actualizar las consultas de CloudWatch Logs Insights.	20 de junio de 2024
CloudWatchLogsReadOnlyAccesspolítica actualizada	CloudWatch Logs agregó el `cloudwatch:GenerateQuery` permiso para CloudWatchLogsReadOnlyAccessque los usuarios con esta política puedan generar una cadena de consulta de CloudWatch Logs Insights a partir de un mensaje en lenguaje natural.	26 de noviembre de 2023
CloudWatchLogsFullAccesspolítica actualizada	CloudWatch Logs agregó el `cloudwatch:GenerateQuery` permiso para CloudWatchLogsFullAccessque los usuarios con esta política puedan generar una cadena de consulta de CloudWatch Logs Insights	26 de noviembre de 2023

	a partir de un mensaje en lenguaje natural.	
CloudWatch Logs añade un análisis de patrones de registro	CloudWatch Ahora, Logs busca patrones en los eventos de registro cada vez que realiza una consulta de CloudWatch Logs Insights. Para obtener más información, consulte Análisis de patrones.	26 de noviembre de 2023
CloudWatch Los registros añaden la detección de anomalías en los registros	Puede crear un detector de anomalías de registro para un grupo de registros. El detector de anomalías escanea los eventos de registro introducidos en el grupo de registros y encuentra anomalías en los datos del registro. Para obtener más información, consulte Detección de anomalías de registro.	26 de noviembre de 2023
CloudWatch Logs añade una función de comparación	Ahora puede usar CloudWatch Logs Insights para comparar los cambios en sus eventos de registro a lo largo del tiempo. Para obtener más información, consulte Comparar (diferenciar) con intervalos de tiempo anteriores.	26 de noviembre de 2023

CloudWatch Logs agrega una nueva clase de registro	CloudWatch Logs admite dos clases de grupos de registros , por lo que puede disponer de una opción rentable para los registros a los que accede con poca frecuencia, y también tiene una opción completa para los registros que requieren supervisión en tiempo real u otras funciones. Para obtener más información, consulte Clases de registro.	26 de noviembre de 2023
CloudWatch Logs Insights admite la generación de consultas en lenguaje natural	CloudWatch Logs Insights admite el lenguaje natural para generar y actualizar consultas. Para obtener más información, consulte Usar lenguaje natural para generar y actualizar las consultas de CloudWatch Logs Insights.	26 de noviembre de 2023
CloudWatch Logs añade compatibilidad con la sintaxis de patrones de filtrado de expresiones regulares para Live Tail	Ahora puede personalizar aun más sus operaciones de búsqueda y coinciden cia para adaptarlas a sus necesidades con expresion es regulares flexibles dentro de los patrones de filtro de Live Tail. Para obtener más información, consulta la sintaxis de los patrones de filtrado en la Guía del usuario de Amazon CloudWatch Logs.	13 de noviembre de 2023

CloudWatch Logs añade compatibilidad con la sintaxis de patrones de filtrado de expresiones regulares para filtros métricos, filtros de suscripción y eventos de registro de filtros	Ahora puede personalizar aun más sus operaciones de búsqueda y coinciden cia para adaptarlas a sus necesidades con expresion es regulares flexibles dentro de los patrones de filtrado. Para obtener más informaci ón, consulta la sintaxis de los patrones de filtrado en la Guía del usuario de Amazon CloudWatch Logs.	5 de septiembre de 2023
CloudWatch Logs Insights añade un comando de patrón	Ahora puede usar un patrón en sus consultas de CloudWatch Logs Insights para agrupar automáticamente sus datos de registro en patrones. Un patrón es una estructura de texto compartida que se repite entre los campos de registro. Para obtener más información, consulta el patrón en la Guía del usuario de Amazon CloudWatch Logs.	17 de julio de 2023

CloudWatch Logs Insights añade un comando de dedup	Ahora puede usar la deduplicación en sus consultas de CloudWatch Logs Insights para eliminar los resultados duplicados en función de valores específicos en los campos que especifique. Para obtener más información, consulta dedup en la Guía del usuario de Amazon CloudWatch Logs.	20 de junio de 2023
Políticas de protección de datos de la cuenta	Ahora puede establecer políticas de protección de datos de la cuenta. Estas políticas para la cuenta pueden auditar y enmascarar la información confidencial de los eventos de registro de todos los grupos de registro de la cuenta. Para obtener más información, consulta Ayuda a proteger los datos de registro confidenciales mediante el enmascaramiento en la Guía del usuario de Amazon CloudWatch Logs.	8 de junio de 2023

Incorporación de la función Live Tail	CloudWatch A Logs se ha añadido la función Live Tail, que te permite escanear los registros a medida que se ingieren para ayudarte a solucionar problemas. Si lo desea, puede filtrar el flujo de eventos de registro que se muestra en función de términos específicos y, también, destacar los eventos de registro que tengan esos términos. Para obtener más información, consulte Use live tail to view logs in near real time.	6 de junio de 2023
CloudWatchLogsRead OnlyAccesspolítica actualizada	CloudWatch Registra los permisos añadidos a CloudWatchLogsRead OnlyAccess. Los `logs:StopLiveTail` permisos `logs:StartLiveTail` y se agregaron para que los usuarios con esta política puedan usar la consola para iniciar y detener las sesiones finales de CloudWatch Logs Live. Para obtener más información, consulte Use live tail to view logs in near real time.	6 de junio de 2023

CloudWatch Publicada Logs Insights	Puede utilizar CloudWatch Logs Insights para buscar y analizar sus datos de registro de forma interactiva. Para obtener más información, consulte Análisis de datos de registro con CloudWatch Logs Insights en la Guía del usuario de Amazon CloudWatch Logs.	27 de noviembre de 2018
Compatibilidad con puntos de conexión de Amazon VPC	Ahora puede establecer una conexión privada entre la VPC y CloudWatch los registros. Para obtener más información, consulte Uso de CloudWatch registros con puntos de enlace de VPC de interfaz en la Guía del usuario de Amazon CloudWatch Logs.	28 de junio de 2018

En la siguiente tabla se describen los cambios importantes en la Guía del usuario de Amazon CloudWatch Logs.

Cambio	Descripción	Fecha de la versión
Puntos de conexión de VPC de tipo interfaz	En algunas regiones, puedes usar un punto de enlace de VPC de interfaz para evitar que el tráfico entre tu Amazon VPC y CloudWatch Logs salga de la red de Amazon. Para obtener más información, consulte Uso de CloudWatch registros con puntos finales de VPC de interfaz.	7 de marzo de 2018
Registros de consultas de DNS de Route 53	Puede usar CloudWatch los registros para almacenar los registros sobre las consultas de DNS recibidas por Route 53. Para obtener más información, consulte ¿Qué es Amazon CloudWatc	7 de septiembre de 2017

Cambio	Descripción	Fecha de la versión
	h Logs? o Registro de consultas de DNS en la Guía para desarrolladores de Amazon Route 53.	
Etiquetar grupos de registros	Puede utilizar las etiquetas para categorizar los grupos de registros. Para obtener más informaci ón, consulte Etiquetar grupos de registros en Amazon CloudWatch Logs.	13 de diciembre de 2016
Mejoras en la consola	Puede navegar desde los gráficos de métricas a los grupos de registros asociados. Para obtener más información, consulte Cambio de métricas a registros.	7 de noviembre de 2016
Mejoras de uso de la consola	Mejora de la experiencia para facilitar la búsqueda, el filtrado y la resolución de problemas. Por ejemplo, ahora puede filtrar los datos de registro en un intervalo de fecha y hora. Para obtener más información, consulte Vea los datos de registro enviados a Logs CloudWatch .	29 de agosto de 2016
Se agregó AWS CloudTrai l soporte para Amazon CloudWatch Logs y nuevas métricas CloudWatch de Logs	Se ha añadido AWS CloudTrail soporte para CloudWatch Logs. Para obtener más informaci ón, consulte El registro CloudWatch registra las operaciones de la API y la consola en AWS CloudTrail.	10 de marzo de 2016
Se agregó soporte para la exportación de CloudWatc h registros a Amazon S3	Se ha añadido soporte para la exportación CloudWatch de datos de Logs a Amazon S3. Para obtener más información, consulte Exportación de datos de registro a Simple Storage Service (Amazon S3).	7 de diciembre de 2015

Cambio	Descripción	Fecha de la versión
Se agregó soporte para eventos AWS CloudTrail registrados en Amazon CloudWatch Logs	Puede crear alarmas CloudWatch y recibir notificaciones sobre una actividad concreta de la API tal como la capture, CloudTrail y utilizar la notificación para solucionar problemas.	10 de noviembre de 2014
Se agregó soporte para Amazon CloudWatch Logs	Puede usar Amazon CloudWatch Logs para monitorear, almacenar y acceder a su sistema, aplicación y archivos de registro personalizados desde instancias de Amazon Elastic Compute Cloud (Amazon EC2) u otras fuentes. A continuación, puede recuperar los datos de registro asociados de CloudWatch Logs mediante la CloudWatch consola de Amazon, los comandos de CloudWatch Logs o el SDK de CloudWatch Logs. AWS CLI Para obtener más información, consulte ¿Qué es Amazon CloudWatch Logs?.	10 de julio de 2014

AWS Glosario

Para obtener la AWS terminología más reciente, consulte el AWS glosario de laGlosario de AWS Referencia.

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