aws

Guía del usuario de

Amazon CloudWatch Logs

Amazon CloudWatch Logs: Guía del usuario de

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 Los registros le permiten 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 características necesarias para los registros que requieren supervisión en tiempo real u otras características. 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.

- Cree índices de campo para que las consultas sean más eficientes: se pueden crear índices de campo de los campos en los eventos de registros. Cuando, a continuación, se utiliza un índice de campos en una consulta de CloudWatch Logs Insights, la consulta intenta omitir el procesamiento de los eventos de registro que se sabe que no incluyen el campo indexado. Esta consulta reduce el volumen de digitalización de las consultas, lo que permite devolver resultados con mayor rapidez. Para empezar, consulte Creación de índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de análisis.
- 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 Solución de problemas con CloudWatch Logs Live Tail.
- Supervisa los registros de EC2 las instancias de Amazon: puedes usar CloudWatch Logs para monitorear aplicaciones y sistemas mediante datos de registro. Por ejemplo, CloudWatch Logs puede hacer un seguimiento del número de errores que se producen en los registros de tu aplicación y enviarte una notificación siempre que la tasa de errores supere el umbral que especifiques. 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 registro, manteniendo la retención indefinida o seleccionar un periodo de retención de entre 10 años y un día.
- Protección contra la eliminación: una medida de seguridad que evita la eliminación accidental de los grupos de registros y sus flujos de registros. Cuando está habilitada en un grupo de registros, la protección contra la eliminación bloquea todas las operaciones de eliminación hasta que se deshabilite explícitamente. De forma predeterminada, la protección contra la eliminación no está habilitada. Esta función opcional ayuda a proteger los datos operativos y de cumplimiento críticos para que no se eliminen accidentalmente, como los grupos de registros que contienen datos de auditoría y los registros de las aplicaciones de producción para la solución de problemas y el análisis.
- Archivar los datos de registro: puede utilizar CloudWatch los registros para almacenar los 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.
- Centralice los registros en todas las cuentas y regiones: puede utilizar la centralización de CloudWatch registros para definir reglas de centralización entre cuentas y regiones que repliquen los datos de registro ingeridos en entornos de varias cuentas y regiones en una región y una cuenta centrales. Se puede configurar una región de respaldo dentro de la cuenta central para aumentar la resiliencia, definir la configuración de cifrado para los grupos de registros recién creados en la cuenta central y crear filtros de métricas y de suscripción para registrar los eventos de registro de cuentas y regiones de origen específicas con capacidades de filtrado mejoradas.

Servicios relacionados AWS

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 Consola de administración

de AWS, 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 Logs ofrece dos clases de grupos de registros. La clase de registro Estándar es una opción que tiene todas las características necesarias para los registros que requieren supervisión en tiempo real o los registros a los que se accede con frecuencia. La clase de registro Acceso poco frecuente es una opción más económica para los registros a los que se accede con menos frecuencia. Además, es compatible con un subconjunto de las capacidades de clase 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 de registro que CloudWatch Logs entiende contiene dos propiedades: la marca de tiempo en que se produjo 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 registro 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 registro. 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 registro denominado `MyWebsite.com/Apache/access_log`.

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

Filtros de métricas

Puede utilizar filtros de métrica para extraer las observaciones de métricas de eventos introducidos y transformarlas en puntos de datos en una métrica de CloudWatch . Los filtros de métricas se asignan a grupos de registro y todos los filtros asignados a un grupo de registro 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 registro se aplica a sus flujos de registro.

Protección contra eliminación

La protección contra la eliminación es una medida de seguridad que evita la eliminación accidental de los grupos de registros y sus flujos de registros. Cuando está habilitada en un grupo de registros, la protección contra la eliminación bloquea todas las operaciones de eliminación hasta que se deshabilite explícitamente. De forma predeterminada, la protección contra la eliminación no está habilitada. Esta función opcional ayuda a proteger los datos operativos y de cumplimiento críticos para que no se eliminen accidentalmente, como los grupos de registros que contienen datos de auditoría y los registros de las aplicaciones de producción para la solución de problemas y el análisis.

Facturación y coste de Amazon CloudWatch Logs

Para obtener información detallada sobre cómo analizar los costos y el uso de CloudWatch Logs y CloudWatch las mejores prácticas sobre cómo reducir los costos, consulte CloudWatch facturación y costos.

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

Cuando te registres AWS, podrás empezar a usar CloudWatch Logs de forma gratuita utilizando 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).

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 Estándar y Acceso poco frecuente 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**
>
> Después de crear un grupo de registro, la clase de registro no se puede cambiar.

Características admitidas

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

Característica	Standard	Acceso poco frecuente	
Incorporación y almacenamiento de registros totalmente gestionados	Sí ✓	Sí ✓	
Características entre cuentas	Sí ✓	Sí ✓	
Cifrado con AWS KMS	Sí ✓	Sí ✓	
CloudWatch Comandos de consulta de Logs Insights	Sí ✓	Sí ✓ (a mayoría de los comandos, consulte Comandos de lenguaje de consulta de Información de registros compatibles con las clases de registro)	
CloudWatch Registra los campos descubiertos por Insights	Sí ✓	Sí ✓	
Facetas	Sí ✓	No	
¿Uso de CloudWatch Pipeline para transformar los registros	Sí ✓	Sí ✓	
Exportación a Amazon S3	Sí ✓	Sí ✓	
Integración de tablas S3	Sí ✓	Sí ✓	
Consultas programadas	Sí ✓	Sí ✓	

Característica	Standard	Acceso poco frecuente	
Uso de OpenSearch PPL o OpenSearch SQL para realizar consultas en CloudWatch Logs Insights;	Sí ✓	Sí ✓	
Asistencia para consultas en lenguaje natural	Sí ✓	No	
CloudWatch Registra la detección de anomalías	Sí ✓	No	
Live Tail	Sí ✓	No	
Indexación de campos	Sí ✓	No	
Comparación con el intervalo de tiempo anterior	Sí ✓	No	
Filtros de suscripción	Sí ✓	No	
GetLogEventsy operaciones FilterLogEventsde API	Sí ✓	No admitido. Utilice CloudWatch Logs Insights para ver los eventos de registro almacenados en grupos de registros de la clase de registro de acceso poco frecuente.	
Filtros de métricas	Sí ✓	No	

Característica	Standard	Acceso poco frecuente	
Incorporación de registros de Información de contenedores	Sí ✓	No	
Incorporación de registros de Lambda Insights	Sí ✓	No	
Protección de datos confidenciales con enmascaramiento	Sí ✓	Sí ✓	
Formato de métricas integrado	Sí ✓	No	

> **Note**
>
> Además de estas dos clases de registro, hay una clase de registro `Delivery`. Utilice la clase de `Delivery` registro solo para entregar AWS Lambda registros para almacenarlos en Amazon S3 o Amazon Data Firehose. Los eventos de registro de los grupos de registros de la clase Delivery se guardan CloudWatch Logs durante dos días. Este período de retención es fijo y no se puede cambiar. Esta clase de registro no ofrece CloudWatch Logs funciones completas, como las consultas de CloudWatch Logs Insights.

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

Para empezar AWS, necesitas un Cuenta de AWS. Para obtener información sobre cómo crear un Cuenta de AWS, consulte Cómo empezar con un Cuenta de AWS en la Guía de AWS Account Management referencia.

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 las instancias de EC2 y los 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 (IMDSv2) del Servicio de Metadatos de Instancia. Si su servidor lo 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 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 es compatible con la versión 2 (IMDSv2) del Servicio de Metadatos de Instancia. Si su servidor lo usa IMDSv2, debe usar el agente unificado más reciente en lugar del antiguo agente 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 Conexión a una instancia en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre problemas de conexión, consulte Solucionar problemas de conexión con la 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 de nueva creación y el flujo de registros en la consola de CloudWatch después de que el agente haya estado en ejecución durante unos minutos.

   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 Conexión a una instancia en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre problemas de conexión, consulte Solucionar problemas de conexión con la instancia en la Guía del usuario de Amazon EC2.
2. Ejecute el instalador del agente de CloudWatch registros 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	Description (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 registro de destino	El nombre de su grupo de registro. El instalador sugiere un nombre de grupo de registro.
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 de nueva creación y el flujo de registros en la consola de CloudWatch después de que el agente haya estado en ejecución durante unos minutos.

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

Inicio rápido: instale y configure el agente CloudWatch Logs 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 acerca de 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:

   JSON

```
{
    "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:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

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 Logs CloudWatch

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

   Para obtener más información, consulte Lanzamiento de una instancia en la Guía del usuario de 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). Amazon Linux AMIs, y AMIs para Ubuntu y RHEL, ya incluyen cloud-init, pero es posible que CentOS y otros AMIs 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://amzn-s3-demo-bucket/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 las 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 nuevo agente unificado de CloudWatch. 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.

Steps

- 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.

Steps

- 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 registro 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 una única IDs (por ejemplo, «CloudWatchLogs2" y CloudWatchLogs 3") y una región diferente para cada ID.

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 registro. Este nombre aparecerá en la pantalla Log Groups (Grupos de registros) en la consola de CloudWatch.
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 las 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 adjuntar 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 Puertas de enlace 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 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 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 nuevo agente unificado de CloudWatch. 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.

Steps

- 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 registro 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 una única IDs (por ejemplo, «CloudWatchLogs2" y CloudWatchLogs 3") y una región diferente para cada ID.

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 registro. Este nombre aparecerá en la pantalla Log Groups (Grupos de registros) en la consola de CloudWatch.
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 las 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 EC2 Config (. `EC2Config.exe`) La instancia debe tener EC2 Config 4.0 o una versión posterior, y puedes usar este procedimiento.

Para configurar CloudWatch mediante EC2 Config 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.

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 Conexión a una instancia en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre problemas de conexión, consulte Solucionar el problema de conexión con la 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 el archivo/var/log/awslogs.log para ver si hay errores, advertencias o problemas con el agente de CloudWatch Logs.

Inicie el agente de CloudWatch 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 Conexión a una instancia en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre problemas de conexión, consulte Solucionar problemas de conexión con la 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 Conexión a una instancia en la Guía del usuario de Amazon EC2.

   Para obtener más información sobre problemas de conexión, consulte Solucionar problemas de conexión con la 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
   ```

CloudWatch Registra la referencia del agente

> **Important**
>
> Esta sección es una referencia para quienes utilizan el obsoleto agente CloudWatch Logs. Si utiliza la versión 2 (IMDSv2) del Servicio de Metadatos de Instancia, debe usar el nuevo CloudWatch agente unificado. Sin embargo, aunque no lo utilice IMDSv2, le recomendamos encarecidamente que utilice el CloudWatch agente unificado más reciente en lugar del obsoleto agente CloudWatch Logs. Para obtener información sobre el CloudWatch 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 agente de CloudWatch registros obsoleto al agente unificado, cree 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.pylogging-config-fileformatthon.org/2/library/logging.config.html#). 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 Logs. CloudWatch 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 registro 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 El archivo puede apuntar a un archivo específico o a varios archivos (utilizando caracteres comodín 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:

   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 de 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
```

Compartimentar CloudWatch 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 Cuando se CloudWatch inicia el agente de registros, incluye cualquier configuración de transmisión en estos archivos de configuración adicionales. Las propiedades de configuración de la sección [general] deben definirse en el archivo de configuración principal (/var/awslogs/etc/awslogs.conf) and are ignored in any additional configuration files found in /var/awslogs/etc/config/.

Si no tiene un directorio/var/awslogs/etc/config/porque instaló el agente con rpm, puede usar el directorio/etc/awslogs/config/en su lugar.

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,/var/log/syslog.log is renamed /var/log/syslog.log.1. If /var/log/syslog.log.1 already exists from a previous rotation, it is renamed /var/log/syslog.log.2.
- Truncar el archivo de registro original en vigor después de crear una copia. Por ejemplo,/var/log/syslog.log is copied to /var/log/syslog.log.1 and /var/log/syslog.log está truncado. 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 crea/var/log/syslog.log.2014-01-01 remains and /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/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 necesita permiso para realizar `CreateLogGroup` `CreateLogStreamDescribeLogStreams`,`DescribeLogGrooupd`, `PutLogEvents` y `PutRetentionPolicy` acciones. Si está utilizando el último agente, no es necesario `DescribeLogStreams`. Consulte la política de IAM de ejemplo a continuación.

JSON

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

```
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams",
      "logs:DescribeLogGroups",
      "logs:PutRetentionPolicy"
    ],
    "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 registro 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`.

Inicio rápido: se utiliza CloudFormation para empezar a utilizar Logs 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 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 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 .

Ingesta de registros a través de puntos finales HTTP

Amazon CloudWatch Logs proporciona puntos de enlace HTTP que le permiten enviar registros directamente a CloudWatch Logs mediante simples solicitudes HTTP POST. Estos puntos de enlace admiten tanto la autenticación mediante SigV4 como la autenticación mediante token portador.

> **Important**
>
> Recomendamos utilizar la autenticación SigV4 para todas las cargas de trabajo de producción en las que sea posible la integración del SDK AWS . SigV4 utiliza credenciales a corto plazo y proporciona la mejor postura de seguridad. La autenticación con token portador (clave de API) está pensada para situaciones en las que SigV4 no es factible, como los reenviadores de registros de terceros que no admiten la integración con el SDK. AWS Para obtener más información, consulte Alternativas a las claves de acceso a largo plazo en la Guía del usuario de IAM.

CloudWatch Logs admite los siguientes puntos finales de ingesta de HTTP:

Punto de conexión	Ruta	Contenido-Tipo	Formato
OpenTelemetry Logs	`/v1/logs`	`application/json` o `application/x-protobuf`	OTTP, JSON o Protobug
HLC Logs	`/services/collector/event`	`application/json`	Formato HLC
ND-JSON Logs	`/ingest/bulk`	`application/json` o `application/x-ndjson`	JSON delimitado por saltos de línea
Structured JSON Logs	`/ingest/json`	`application/json`	Objeto o matriz JSON

Comportamiento común

Todos los puntos finales de ingestión de HTTP comparten el siguiente comportamiento:

Autenticación

Todos los puntos finales admiten la autenticación mediante SigV4 y la autenticación mediante token portador:

- SigV4 (recomendado): AWS firma estándar, versión 4. Utilice SigV4 siempre que su aplicación o infraestructura sea compatible con el AWS SDK o pueda firmar solicitudes. SigV4 utiliza credenciales de corta duración y es el método de autenticación más seguro.
- Token de portador: usa el `Authorization: Bearer <ACWL token>` encabezado.
  - El token debe ser un token portador de la ACWL válido. Para obtener instrucciones de configuración, consulte. the section called “Autenticación mediante token portador”
  - Requiere los permisos `logs:PutLogEvents` e `logs:CallWithBearerToken` IAM.

Grupo de registros y flujo de registros

- Se proporciona mediante encabezados: `x-aws-log-group` y `x-aws-log-stream`
- Los parámetros de consulta también `?logGroup=<name>&logStream=<name>` se admiten en todos los puntos finales excepto en OTLP.
- No puede utilizar tanto los parámetros de consulta como los encabezados para el mismo parámetro.
- Se requieren tanto el grupo de registros como el flujo de registros.

Respuesta

- Éxito: `HTTP 200` con cuerpo `{}`
- Errores de validación: `HTTP 400`
- Fallos de autenticación: `HTTP 401`

Configuración de la autenticación por token de portador

Antes de poder enviar los registros mediante la autenticación por token portador con cualquiera de los puntos de enlace de ingestión de HTTP, debe:

- Cree un usuario de IAM con permisos de registro CloudWatch
- Genere credenciales específicas del servicio (token de portador)
- Cree un grupo de registros y un flujo de registros
- Habilite la autenticación por token portador en el grupo de registros

> **Important**
>
> Siempre que sea posible, recomendamos utilizar la autenticación SigV4 con credenciales a corto plazo para todas las cargas de trabajo. SigV4 proporciona la postura de seguridad más sólida. Restrinja el uso de claves de API (símbolos portadores) a situaciones en las que la autenticación basada en credenciales a corto plazo no sea factible. Cuando esté preparado para incorporar los CloudWatch registros en aplicaciones con mayores requisitos de seguridad, debería cambiar a credenciales de corta duración. Para obtener más información, consulte Alternativas a las claves de acceso a largo plazo en la Guía del usuario de IAM.

Opción 1: Inicio rápido a utilizar la consola AWS

La consola AWS de administración proporciona un flujo de trabajo simplificado para generar claves de API para el acceso a los puntos de conexión HTTP.

Para configurar el acceso al punto final HTTP mediante la consola

1. Inicie sesión en la consola AWS de administración.
2. Vaya a CloudWatch> Configuración > Registros.
3. En la sección Claves de API, selecciona Generar clave de API.
4. Para Expiración de la clave de API, realice una de las siguientes acciones:
   - Selecciona una duración de caducidad de la clave de API de 1, 5, 30, 90 o 365 días.
   - Elija Duración personalizada para especificar una fecha de expiración personalizada de la clave de API.
   - Selecciona Nunca caduca (no se recomienda).
5. Seleccione Generar clave de API.

   La consola automáticamente:

- Crea un nuevo usuario de IAM con los permisos adecuados
- Adjunta la política de CloudWatchLogsAPIKeyacceso gestionado (incluye `logs:PutLogEvents` y `logs:CallWithBearerToken` permisos)
- Genera credenciales específicas del servicio (clave de API)

6. Copie y guarde de forma segura las credenciales mostradas:

- ID de clave de API (ID de credencial específica del servicio)
- Secreto de clave de API (token de portador)

> **Important**
>
> Guarde el secreto de la clave de la API inmediatamente. no la puede recuperar en otro momento. Si la pierdes, tendrás que generar una nueva clave de API.

7. Crea el grupo de registros y el flujo de registros donde se almacenarán tus registros:

```
# Create the log group
aws logs create-log-group \
    --log-group-name /aws/hlc-logs/my-application \
    --region us-east-1

# Create the log stream
aws logs create-log-stream \
    --log-group-name /aws/hlc-logs/my-application \
    --log-stream-name application-stream-001 \
    --region us-east-1
```

8. Habilite la autenticación por token de portador en el grupo de registros:

```
aws logs put-bearer-token-authentication \
    --log-group-identifier /aws/hlc-logs/my-application \
    --bearer-token-authentication-enabled \
    --region us-east-1
```

Verifique la configuración:

```
aws logs describe-log-groups \
    --log-group-name-prefix /aws/hlc-logs/my-application \
```

```
    --region us-east-1
```

Permisos incluidos: el usuario de IAM creado automáticamente tendrá los siguientes permisos:

- `logs:PutLogEvents`— Enviar los eventos del registro a CloudWatch los registros
- `logs:CallWithBearerToken`— Autenticarse con un token de portador
- `kms:Describe*`,`kms:GenerateDataKey*`, `kms:Decrypt` — Acceda a grupos de registros cifrados con KMS (con la condición de que se limite al servicio de registros)

Opción 2: configuración manual

Si prefiere tener más control sobre la configuración de IAM o necesita personalizar los permisos, puede configurar el acceso al punto final HTTP de forma manual.

Paso 1: Crear un usuario de IAM

Cree un usuario de IAM que se utilizará para la ingesta de registros:

1. Inicie sesión en la consola de AWS administración y vaya a IAM.
2. En el panel de navegación izquierdo, elija Usuarios.
3. Seleccione la opción Crear un usuario.
4. Introduzca un nombre de usuario (por ejemplo,`cloudwatch-logs-hlc-user`).
5. Elija Siguiente.
6. Adjunte una de las siguientes políticas de IAM:

   Opción A: utilizar la política gestionada (recomendada)

   Adjunte la política de CloudWatchLogsAPIKeyacceso gestionado.

   Opción B: crear una política personalizada

   Cree y adjunte la siguiente política de IAM:

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

```
            "Sid": "LogsAPIs",
            "Effect": "Allow",
            "Action": [
                "logs:CallWithBearerToken",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Sid": "KMSAPIs",
            "Effect": "Allow",
            "Action": [
                "kms:Describe*",
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "logs.*.amazonaws.com"
                    ]
                }
            },
            "Resource": "arn:aws:kms:*:*:key/*"
        }
    ]
}
```

7. Seleccione Siguiente y, a continuación, Crear usuario.

> **Note**
>
> Los permisos de KMS son necesarios si planea enviar registros a grupos de registros cifrados con KMS. La condición restringe el acceso al KMS únicamente a las claves utilizadas a través CloudWatch del servicio de registros.

Paso 2: Generar credenciales específicas del servicio (clave de API)

Genere la clave CloudWatch de API de Logs mediante la CreateServiceSpecificCredentialAPI. También puede utilizar el comando create-service-specific-credentialCLI. En cuanto a la antigüedad

de las credenciales, puede especificar un valor entre 1 y 36600 días. Si no especifica una antigüedad de la credencial, la clave de API no caducará.

Para generar una clave de API con una caducidad de 30 días:

```
aws iam create-service-specific-credential \
    --user-name cloudwatch-logs-hlc-user \
    --service-name logs.amazonaws.com \
    --credential-age-days 30
```

La respuesta es un ServiceSpecificCredentialobjeto. El `ServiceCredentialSecret` valor es tu clave de API de CloudWatch Logs (token portador).

> **Important**
>
> Guarde el valor `ServiceCredentialSecret` de forma segura, ya que no podrá recuperarlo más tarde. Si lo pierdes, tendrás que generar una nueva clave de API.

Paso 3: Crear un grupo de registros y un flujo de registros

Cree el grupo de registros y el flujo de registros donde se almacenarán sus registros:

```
# Create the log group
aws logs create-log-group \
    --log-group-name /aws/hlc-logs/my-application \
    --region us-east-1

# Create the log stream
aws logs create-log-stream \
    --log-group-name /aws/hlc-logs/my-application \
    --log-stream-name application-stream-001 \
    --region us-east-1
```

Paso 4: Habilite la autenticación por token portador

Habilite la autenticación por token portador en el grupo de registros:

```
aws logs put-bearer-token-authentication \
```

```
    --log-group-identifier /aws/hlc-logs/my-application \
    --bearer-token-authentication-enabled \
    --region us-east-1
```

Verifique la configuración:

```
aws logs describe-log-groups \
    --log-group-name-prefix /aws/hlc-logs/my-application \
    --region us-east-1
```

Controle los permisos para generar y usar las claves de API CloudWatch de Logs

La generación y el uso de las claves de la API de CloudWatch Logs se controlan mediante claves de acción y condición tanto en los servicios de CloudWatch Logs como de IAM.

Controlar la generación de claves de API de CloudWatch Logs

La CreateServiceSpecificCredential acción iam: controla la generación de una clave específica del servicio (como una clave de API de CloudWatch Logs). Puede limitar esta acción a los usuarios de IAM como una forma de restringir el número de usuarios para los que se puede generar una clave.

Puede utilizar las siguientes claves de condición para imponer condiciones al permiso de la acción `iam:CreateServiceSpecificCredential`:

- iam: ServiceSpecificCredentialAgeDays — Permite especificar, en esta condición, el tiempo de caducidad de la clave en días. Por ejemplo, puede utilizar esta clave de condición para permitir únicamente la creación de claves de API que caduquen en 90 días.
- iam: ServiceSpecificCredentialServiceName — Permite especificar, en la condición, el nombre de un servicio. Por ejemplo, puedes usar esta clave de condición para permitir solo la creación de claves de API para CloudWatch los registros y no para otros servicios.

Controlar el uso de las claves de API de CloudWatch Logs

La `logs:CallWithBearerToken` acción controla el uso de una clave de API de CloudWatch Logs. Para evitar que una identidad utilice las claves de la API de CloudWatch Logs, adjunte una política que deniegue la `logs:CallWithBearerToken` acción al usuario de IAM asociado a la clave.

Ejemplos de políticas

Impida que una identidad genere y utilice las claves de la API CloudWatch de Logs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyCWLAPIKeys",
            "Effect": "Deny",
            "Action": [
                "iam:CreateServiceSpecificCredential",
                "logs:CallWithBearerToken"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Warning**
>
> Esta política impedirá la creación de credenciales para todos los AWS servicios que admiten la creación de credenciales específicas de un servicio. Para obtener más información, consulte Credenciales específicas del servicio para los usuarios de IAM.

Impida que una identidad utilice las claves de la API de CloudWatch Logs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "logs:CallWithBearerToken",
            "Resource": "*"
        }
    ]
}
```

Permita la creación de claves de CloudWatch registro solo si caducan en un plazo de 90 días

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceSpecificCredential",
            "Resource": "arn:aws:iam::123456789012:user/username",
            "Condition": {
                "StringEquals": {
                    "iam:ServiceSpecificCredentialServiceName": "logs.amazonaws.com"
                },
                "NumericLessThanEquals": {
                    "iam:ServiceSpecificCredentialAgeDays": "90"
                }
            }
        }
    ]
}
```

Claves de API rotativas

La rotación regular de las claves de API reduce el riesgo de acceso no autorizado. Te recomendamos establecer un programa de rotación que se ajuste a las políticas de seguridad de tu organización.

Proceso de rotación

Para rotar una clave de API sin interrumpir la entrega de registros, siga este procedimiento:

1. Cree una nueva credencial (secundaria) para el usuario de IAM:

   ```
   aws iam create-service-specific-credential \
       --user-name cloudwatch-logs-hlc-user \
       --service-name logs.amazonaws.com \
       --credential-age-days 90
   ```

2. (Opcional) Guarde la nueva credencial AWS Secrets Manager para recuperarla de forma segura y rotarla automáticamente.
3. Importe la nueva credencial al portal de su proveedor o actualice la configuración de la aplicación para usar la nueva clave de API.

4. Defina la credencial original como inactiva:

```
aws iam update-service-specific-credential \
    --user-name cloudwatch-logs-hlc-user \
    --service-specific-credential-id ACCA1234EXAMPLE1234 \
    --status Inactive
```

5. Compruebe que la entrega de registros no se vea afectada supervisando la `IncomingBytes` métrica de su grupo de registros. CloudWatch Para obtener más información, consulte Supervisión con CloudWatch métricas.
6. Tras confirmar que la entrega se ha realizado correctamente con la nueva clave, elimine la credencial anterior:

```
aws iam delete-service-specific-credential \
    --service-specific-credential-id ACCA1234EXAMPLE1234
```

Supervisión del vencimiento de la clave

Para comprobar la fecha de creación y el estado de tus claves de API existentes, usa el list-service-specific-credentialscomando:

```
aws iam list-service-specific-credentials \
    --user-name cloudwatch-logs-hlc-user \
    --service-name logs.amazonaws.com
```

La respuesta incluye `CreateDate` y `Status` para cada credencial. Utilice esta información para identificar las claves que están a punto de caducar o que han estado activas durante más tiempo del permitido por su política de rotación.

Responder a una clave de API comprometida

Si sospechas que una clave de API se ha visto comprometida, sigue los siguientes pasos de inmediato:

1. Desactiva la clave inmediatamente para evitar que se siga utilizando sin autorización:

```
aws iam update-service-specific-credential \
    --user-name cloudwatch-logs-hlc-user \
```

```
    --service-specific-credential-id ACCA1234EXAMPLE1234 \
    --status Inactive
```

2. Revise CloudTrail los registros para determinar el alcance del acceso no autorizado. Consulte the section called “Registrar el uso de las claves de API con CloudTrail” para saber cómo habilitar la auditoría del uso de las claves de API.
3. Cree una clave de reemplazo siguiendo el proceso de rotación descrito enthe section called “Proceso de rotación”.
4. Elimine la clave dañada una vez que se haya realizado la sustitución:

```
aws iam delete-service-specific-credential \
    --service-specific-credential-id ACCA1234EXAMPLE1234
```

5. Adjunta una política de denegación si necesitas bloquear inmediatamente todo acceso al token portador para el usuario de IAM mientras investigas:

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "logs:CallWithBearerToken",
        "Resource": "*"
    }
}
```

> **Note**
>
> Para llevar a cabo estas acciones a través de la API, debes autenticarte con AWS credenciales y no con una clave de API de CloudWatch Logs.

También puedes utilizar las siguientes operaciones de la API de IAM para gestionar las claves comprometidas:

- ResetServiceSpecificCredential— Restablezca la clave para generar una nueva contraseña sin eliminar la credencial. La clave no debe haber caducado.

Prácticas recomendadas de seguridad para las claves de API

Sigue estas prácticas recomendadas para proteger tus claves de API de CloudWatch Logs:

- Nunca insertes claves de API en el código fuente. No codifique las claves de API en el código de la aplicación ni las consigne en los sistemas de control de versiones. Si una clave se guarda accidentalmente en un repositorio público, es posible que el escaneo AWS automático la señale y, por lo tanto, debes rotarla inmediatamente.
- Usa un administrador de secretos. Almacene las claves de API en una solución de administración de secretos equivalente AWS Secrets Manager o en una solución equivalente. Esto permite el control de acceso centralizado, el registro de auditorías y la rotación automática.
- Establezca una caducidad para todas las claves. Especifique siempre un `--credential-age-days` valor al crear las claves de API. Para garantizar una vida útil máxima de las claves en toda su organización, utilice la clave de condición de `iam:ServiceSpecificCredentialAgeDays` IAM. Para ver un ejemplo, consulta the section called “Permita la creación de claves de CloudWatch registro solo si caducan en un plazo de 90 días”.
- Aplica permisos con privilegios mínimos. Limite los permisos del usuario de IAM únicamente a los grupos de registros y las acciones necesarias. Utilice la política de CloudWatchLogsAPIKeyacceso gestionado como punto de partida y restrinja aún más según sea necesario.
- Habilite CloudTrail el registro. Audite el uso de las claves de la API habilitando CloudTrail los eventos de datos para`AWS::Logs::LogGroupAuthorization`. Consulte the section called “Registrar el uso de las claves de API con CloudTrail”.
- Supervise con IAM Access Analyzer. Utilice IAM Access Analyzer para identificar las credenciales no utilizadas y las políticas excesivamente permisivas asociadas a los usuarios de IAM clave de la API.
- Gire las claves con regularidad. Establezca un programa de rotación y siga el proceso descrito enthe section called “Claves de API rotativas”.

Registrar el uso de las claves de API con CloudTrail

Se puede utilizar AWS CloudTrail para registrar eventos de datos para el uso de la clave de la API de CloudWatch Logs. CloudWatch Los registros emiten eventos de `AWS::Logs::LogGroupAuthorization` datos para `CallWithBearerToken` las llamadas, lo que te permite auditar cuándo y cómo se utilizan las claves de API para enviar los registros.

Para habilitar el CloudTrail registro para el uso de las claves CloudWatch de la API de Logs:

> **Note**
>
> El bucket de S3 que especifique para la ruta debe tener una política de bucket que permita CloudTrail escribir archivos de registro en él. Para obtener más información, consulte la política de bucket de Amazon S3 para CloudTrail.

1. Cree una ruta:

```
aws cloudtrail create-trail \
    --name cloudwatch-logs-api-key-audit \
    --s3-bucket-name my-cloudtrail-bucket \
    --region us-east-1
```

2. Configure selectores de eventos avanzados para capturar CloudWatch los eventos de autorización de grupos de registros:

```
aws cloudtrail put-event-selectors \
    --region us-east-1 \
    --trail-name cloudwatch-logs-api-key-audit \
    --advanced-event-selectors '[{
        "Name": "CloudWatch Logs API key authorization events",
        "FieldSelectors": [
            { "Field": "eventCategory", "Equals": ["Data"] },
            { "Field": "resources.type", "Equals":
 ["AWS::Logs::LogGroupAuthorization"] }
        ]
    }]'
```

3. Inicie el registro de rutas:

```
aws cloudtrail start-logging \
    --name cloudwatch-logs-api-key-audit \
    --region us-east-1
```

Envío de registros mediante el punto final OTLP (OpenTelemetry registros)

El punto final de OpenTelemetry Logs (`/v1/logs`) acepta datos de registro OpenTelemetry del Protocolo (OTLP) codificados en JSON o Protobuf. Para obtener información detallada sobre el punto final de OTLP, incluida la configuración y el uso, consulte Enviar métricas y seguimientos a with. CloudWatch OpenTelemetry

Si utiliza la autenticación mediante token de portador, complete los pasos de configuración the section called “Autenticación mediante token portador” antes de continuar.

Formato de las solicitudes

- Método: `POST`
- Tipo de contenido: o `application/json` `application/x-protobuf`
- Grupo de registros: solo `x-aws-log-group` encabezado (no se admite el parámetro de consulta)
- Flujo de registro: `x-aws-log-stream` encabezado

Ejemplo de solicitud

```
curl -X POST "https://logs.<region>.amazonaws.com/v1/logs" \
  -H "Authorization: Bearer ACWL<token>" \
  -H "Content-Type: application/json" \
  -H "x-aws-log-group: MyLogGroup" \
  -H "x-aws-log-stream: MyLogStream" \
  -d '{
  "resourceLogs": [
    {
      "resource": {
        "attributes": [
          {
            "key": "service.name",
            "value": { "stringValue": "my-service" }
          }
        ]
      },
      "scopeLogs": [
        {
```

```
          "scope": {
            "name": "my-library",
            "version": "1.0.0"
          },
          "logRecords": [
            {
              "timeUnixNano": "1741900000000000000",
              "severityNumber": 9,
              "severityText": "INFO",
              "body": {
                "stringValue": "User logged in successfully"
              },
              "attributes": [
                {
                  "key": "user.id",
                  "value": { "stringValue": "12345" }
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}'
```

Respuestas

Éxito (se aceptan todos los eventos):

```
HTTP 200 OK
{}
```

Éxito parcial (algunos eventos rechazados):

```
{
  "partialSuccess": {
    "rejectedLogRecords": 5,
    "errorMessage": "{\"tooOldLogEventCount\": 3, \"tooNewLogEventCount\": 1,
 \"expiredLogEventCount\": 1}"
  }
}
```

Cuando la solicitud es Content-Type`application/x-protobuf`, la respuesta se devuelve como un mensaje `ExportLogsServiceResponse` protobuf serializado con los mismos campos.

Comportamientos específicos de OTLP

Los siguientes comportamientos son específicos del punto final de OTLP y no están presentes en los otros puntos de enlace de ingestión de HTTP:

- Encabezado Retry-After: se incluye en las respuestas 503 y 429 para indicar cuándo el cliente debe volver a intentarlo.

Envío de registros mediante el punto final de HLC (registros de HLC)

El punto final de HLC Logs (`/services/collector/event`) se basa en el formato HTTP Log Collector (HLC).

Si utiliza la autenticación mediante token de portador, complete los pasos de configuración antes de continuar. the section called “Autenticación mediante token portador”

Modos de entrada

Cada evento es un objeto JSON con un `"event"` campo obligatorio. Campos de metadatos opcionales:`"time"`,`"host"`,`"source"`,`"sourcetype"`,`"index"`.

Evento único:

```
{"event":"Hello world!","time":1486683865.0}
```

Matriz de eventos JSON:

```
[
  {"event":"msg1","time":1486683865.0},
  {"event":"msg2","time":1486683866.0}
]
```

Eventos concatenados o por lotes (sin contenedor de matrices):

```
{"event":"msg1","time":1486683865.0}{"event":"msg2","time":1486683866.0}
```

Campo de evento (obligatorio)

El `"event"` campo es obligatorio. Su valor puede ser de cualquier tipo de JSON:

```
{"event":"a string message"}
{"event":{"message":"structured data","severity":"INFO"}}
{"event":42}
{"event":true}
```

Los objetos sin un `"event"` campo se omiten silenciosamente:

```
{"message":"this is skipped — no event field"}
```

Campo de tiempo (opcional)

El `"time"` campo está expresado en segundos de época (no en milisegundos), con decimales opcionales para una precisión inferior a un segundo.

Formato	Ejemplo	Interpretado como
Flotante	`"time":1486683865.500`	1486683865500 ms
Entero	`"time":1486683865`	1486683865000 ms
Cadena (flotante)	`"time":"1486683865.500"`	1486683865500 ms
Cadena (entero)	`"time":"1486683865"`	1486683865000 ms
Missing (Ausente)	(sin campo de tiempo)	Hora actual del servidor
Invalid (No válido)	`"time":"invalid"`	Hora actual del servidor

Contenido-Tipo

Solo `application/json` se acepta.

Tipos de valores JSON aceptados

Tipo de nivel superior	Comportamiento
Objeto con `"event"`	Aceptada
Objeto sin `"event"`	Omitido
Matriz de objetos	Cada elemento procesado individualmente
Objetos concatenados	Cada objeto se procesa de forma individual
Primitivo (cadena, número, booleano, nulo)	Omitido

Formato de punto de conexión

La URL del punto final del HLC sigue este formato:

```
https://logs.<region>.amazonaws.com/services/collector/event?
logGroup=<name>&logStream=<name>[&entityName=<name>&entityEnvironment=<environment>]
```

Parámetros necesarios:

- `<region>`— AWS Región (por ejemplo,`us-east-1`,`eu-west-1`)
- `logGroup`— Nombre del grupo de registros codificado en una URL
- `logStream`— Nombre del flujo de registro codificado en una URL

Parámetros opcionales:

Si lo desea, puede asociar sus eventos de registro a una `Service` entidad mediante la inclusión de los siguientes parámetros de consulta. Como los registros enviados a través del punto final del HLC son telemetría personalizada, no se asocian automáticamente a una entidad. Al proporcionar estos parámetros, CloudWatch Logs crea una entidad con `KeyAttributes.Type` set en `Service` y la asocia a sus eventos de registro. Esto permite que la función relacionada con Explore CloudWatch correlacione estos registros con otros datos de telemetría (métricas, trazas y registros) del mismo

servicio, lo que facilita la resolución de problemas y la supervisión de las aplicaciones en diferentes tipos de señales. Para obtener más información sobre las entidades y la telemetría relacionada, consulte Añadir información relacionada a la telemetría personalizada.

- `entityName`— El nombre de la entidad de servicio que se va a asociar al registro de eventos. Este valor se almacena como entidad `KeyAttributes.Name` (por ejemplo, `my-application oapi.myservice.com`).
- `entityEnvironment`— El entorno en el que se aloja el servicio o al que pertenece. Este valor se almacena como entidad `KeyAttributes.Environment` (por ejemplo,`production`,`ec2:default`, `oeks:my-cluster/default`).

Formato de las solicitudes

Envíe los registros mediante HTTP POST con los siguientes encabezados y cuerpo:

Encabezados:

- `Authorization: Bearer <your-bearer-token>`
- `Content-Type: application/json`

Formato de cuerpo:

El cuerpo de la solicitud debe estar en formato JSON con una serie de eventos:

```
{
    "event": [
        {
            "time": 1730141374.001,
            "event": "Application started successfully",
            "host": "web-server-1",
            "source": "application.log",
            "severity": "info"
        },
        {
            "time": 1730141374.457,
            "event": "User login successful",
            "host": "web-server-1",
            "source": "auth.log",
            "user": "john.doe"
        }
```

```
    ]
}
```

Descripciones de los campos:

- `time`— Marca temporal de época de Unix en segundos, con decimal opcional para una precisión inferior a un segundo (opcional)
- `event`— El mensaje de registro o los datos del evento (obligatorio)
- `host`— Nombre de host o identificador de origen (opcional)
- `source`— Identificador de la fuente del registro (opcional)

Se pueden incluir campos personalizados adicionales según sea necesario.

Ejemplo de solicitud

```
curl -X POST "https://logs.<region>.amazonaws.com/services/collector/event?
logGroup=MyLogGroup&logStream=MyStream" \
  -H "Authorization: Bearer ACWL<token>" \
  -H "Content-Type: application/json" \
  -d '{"event":{"message":"User logged
 in","user_id":"u-123"},"time":1486683865.0,"host":"web-01","source":"auth-service"}'
```

Prácticas recomendadas

Agrupación de eventos por lotes

Para un mejor rendimiento y eficiencia:

- Batch varios eventos en una sola solicitud cuando sea posible
- Tamaño de lote recomendado: de 10 a 100 eventos por solicitud
- Tamaño máximo de la solicitud: 1 MB

Gestión de errores

Implemente una gestión de errores adecuada en su aplicación. Códigos de estado HTTP comunes:

- `200 OK`— Los registros se ingirieron correctamente
- `400 Bad Request`— Formato o parámetros de solicitud no válidos

- `401 Unauthorized`— Token de portador no válido o caducado
- `403 Forbidden`— Permisos insuficientes
- `404 Not Found`— El grupo de registros o la transmisión no existen
- `429 Too Many Requests`— Se ha superado el límite de velocidad
- `500 Internal Server Error`— Error de servicio (reintento con un retraso exponencial)

Limitaciones

- Tamaño máximo del evento: 256 KB por evento
- Tamaño máximo de solicitud: 1 MB
- Número máximo de eventos por solicitud: 10 000
- Los nombres de los grupos de registros deben seguir las convenciones CloudWatch de nomenclatura de los registros
- La autenticación por token de portador debe estar habilitada en el grupo de registros si se utiliza la autenticación por token de portador.

Envío de registros mediante el punto final NDJSON (registros ND-JSON)

El punto final de registros de ND-JSON (`/ingest/bulk`) acepta registros en formato NDJSON (JSON delimitado por nueva línea). Cada línea contiene exactamente un valor JSON, separado por caracteres de nueva línea.

Si utiliza la autenticación mediante token de portador, complete los pasos de configuración the section called “Autenticación mediante token portador” antes de continuar.

Formato de las solicitudes

Envía un valor JSON por línea, separado por `\n` (LF) o `\r\n` (CRLF). Las líneas vacías se ignoran silenciosamente.

```
{"timestamp":1771007942000,"message":"event one","level":"INFO"}
{"timestamp":1771007943000,"message":"event two","level":"ERROR"}
{"timestamp":1771007944000,"message":"event three","level":"DEBUG"}
```

Ambas `application/json` `application/x-ndjson` se aceptan como tipo de contenido.

Tipos de valores JSON aceptados

Según la especificación NDJSON (RFC 8259), se acepta cualquier valor JSON válido en cada línea.

Objetos JSON (los más comunes):

```
{"timestamp":1771007942000,"message":"User logged in","service":"auth"}
{"timestamp":1771007943000,"error":"Connection timeout","service":"api"}
```

Matrices JSON (agrupadas en eventos individuales):

```
[{"timestamp":1000,"message":"a"},{"timestamp":2000,"message":"b"}]
```

Esta única línea produce 2 eventos. Cada elemento de la matriz se convierte en un evento de registro independiente.

Valores primitivos:

```
"a plain string log message"
42
true
null
```

Cada primitivo se convierte en su propio evento con la marca de tiempo actual del servidor.

Tipos mixtos:

```
{"timestamp":1771007942000,"message":"structured event"}
"unstructured string message"
42
{"timestamp":1771007943000,"error":"something failed"}
```

Las 4 líneas se aceptan como eventos válidos.

Contenido de la línea	Comportamiento
Objeto JSON	Aceptado, se extrae la marca de tiempo si está presente

Contenido de la línea	Comportamiento
matriz JSON	Aplanado: cada elemento se convierte en un evento independiente
Matriz vacía `[]`	Aceptado, produce 0 eventos
Cadena JSON	Aceptado como mensaje de evento
Número JSON	Aceptado como mensaje de evento
Booleano JSON	Aceptado como mensaje de evento
JSON nulo	Aceptado como mensaje de evento
JSON no válido	Omitido (contado, el procesamiento continúa)
Línea vacía	Ignorada (no se cuenta como omitida)

Campo de fecha y hora

El `"timestamp"` campo está expresado en milisegundos de época (no segundos).

Formato	Ejemplo	Interpretado como
Numérico (milis)	`"timestamp":1771007942000`	1771007942000 ms
Missing (Ausente)	(sin campo de marca de tiempo)	Hora actual del servidor
No numérico	`"timestamp":"invalid"`	Hora actual del servidor

Formato	Ejemplo	Interpretado como
Línea que no es de objetos	`"hello"`, `42`, `true`	Hora actual del servidor

Líneas no válidas

Las líneas que no son JSON válidas se omiten y se cuentan de forma silenciosa. El procesamiento continúa con la siguiente línea.

```
{"message":"valid event"}
this is not valid json
{"message":"another valid event"}
```

Resultado: 2 eventos ingeridos y 1 omitido. Devuelve `HTTP 200`.

Si todas las líneas no son válidas, devuelve `HTTP 400` con. `"All events were invalid"`

Ejemplo de solicitud

```
curl -X POST "https://logs.<region>.amazonaws.com/ingest/bulk?
logGroup=MyLogGroup&logStream=MyStream" \
  -H "Authorization: Bearer ACWL<token>" \
  -H "Content-Type: application/x-ndjson" \
  -d '{"timestamp":1771007942000,"message":"User logged in","level":"INFO"}
{"timestamp":1771007943000,"message":"Query took 42ms","level":"DEBUG"}
{"timestamp":1771007944000,"error":"Connection refused","level":"ERROR"}'
```

Respuestas

Éxito (se aceptan todos los eventos):

```
HTTP 200 OK
{}
```

Éxito parcial (algunos eventos rechazados):

```
{
```

```
  "partialSuccess": {
    "rejectedLogRecords": 5,
    "errorMessage": "{\"tooOldLogEventCount\": 3, \"tooNewLogEventCount\": 1,
 \"expiredLogEventCount\": 1}"
  }
}
```

El `rejectedLogRecords` campo es el número total de eventos rechazados. El `errorMessage` campo contiene un desglose codificado en JSON por motivo de rechazo:

- `tooOldLogEventCount`— Eventos con marcas de tiempo anteriores al período de retención
- `tooNewLogEventCount`— Eventos con marcas de tiempo demasiado lejanas en el futuro
- `expiredLogEventCount`— Eventos que caducaron durante el procesamiento

Prácticas recomendadas

Agrupación de eventos por lotes

Para un mejor rendimiento y eficiencia:

- Batch varios eventos en una sola solicitud cuando sea posible
- Tamaño de lote recomendado: de 10 a 100 eventos por solicitud
- Tamaño máximo de la solicitud: 1 MB

Gestión de errores

Implemente una gestión de errores adecuada en su aplicación. Códigos de estado HTTP comunes:

- `200 OK`— Los registros se ingirieron correctamente
- `400 Bad Request`— Formato o parámetros de solicitud no válidos
- `401 Unauthorized`— Token de portador no válido o caducado
- `403 Forbidden`— Permisos insuficientes
- `404 Not Found`— El grupo de registros o la transmisión no existen
- `429 Too Many Requests`— Se ha superado el límite de velocidad
- `500 Internal Server Error`— Error de servicio (reintento con un retraso exponencial)

Limitaciones

- Tamaño máximo del evento: 256 KB por evento
- Tamaño máximo de solicitud: 1 MB
- Número máximo de eventos por solicitud: 10 000
- Los nombres de los grupos de registros deben seguir las convenciones CloudWatch de nomenclatura de los registros
- La autenticación por token de portador debe estar habilitada en el grupo de registros si se utiliza la autenticación por token de portador.

Envío de registros mediante el punto final de Structured JSON (Structured JSON Logs)

El punto final de Structured JSON Logs (`/ingest/json`) acepta JSON estándar, ya sea un único objeto JSON o una matriz de objetos JSON. Este punto final está diseñado para datos de registro estructurados en los que cada evento es un objeto JSON.

Si utiliza la autenticación por token de portador, complete los pasos de configuración the section called “Autenticación mediante token portador” antes de continuar.

Formato de las solicitudes

Solo `application/json` se acepta como tipo de contenido.

Objeto JSON único:

```
{"timestamp":1771007942000,"message":"single event","level":"INFO"}
```

Matriz de objetos JSON:

```
[
  {"timestamp":1771007942000,"message":"event one","level":"INFO"},
  {"timestamp":1771007943000,"message":"event two","level":"ERROR"}
]
```

Tipos de valores JSON aceptados

Este punto final es estricto: solo los objetos JSON se aceptan como eventos.

Input	Comportamiento
Objeto JSON único	Se acepta como un solo evento
Matriz de objetos JSON	Cada objeto se convierte en un evento independiente
Matriz vacía `[]`	Aceptado, produce 0 eventos
No es un objeto en la matriz (cadena, número, etc.)	Omitido
Primitiva de nivel superior (`"hello"`,) `42`	Omitido
Objetos concatenados `{...}{...}`	Solo se analizó el primer objeto

Ejemplo: matriz con tipos mixtos:

```
[
  {"timestamp":1771007942000,"message":"valid object"},
  "just a string",
  42,
  {"timestamp":1771007943000,"message":"another valid object"}
]
```

Resultado: 2 eventos ingeridos (los objetos) y 2 omitidos (la cadena y el número).

Campo de fecha y hora

El `"timestamp"` campo está expresado en milisegundos de época, igual que el punto final NDJSON.

Formato	Ejemplo	Interpretado como
Numérico (milis)	`"timestamp":1771007942000`	1771007942000 ms
Missing (Ausente)	(sin campo de marca de tiempo)	Hora actual del servidor
No numérico	`"timestamp":"invalid"`	Hora actual del servidor

Ejemplo de solicitud

```
curl -X POST "https://logs.<region>.amazonaws.com/ingest/json?
logGroup=MyLogGroup&logStream=MyStream" \
  -H "Authorization: Bearer ACWL<token>" \
  -H "Content-Type: application/json" \
  -d '[{"timestamp":1771007942000,"message":"User logged in","user_id":"u-123"},
{"timestamp":1771007943000,"message":"Order placed","order_id":"o-456"}]'
```

Respuestas

Éxito (se aceptan todos los eventos):

```
HTTP 200 OK
{}
```

Éxito parcial (algunos eventos rechazados):

```
{
  "partialSuccess": {
    "rejectedLogRecords": 5,
    "errorMessage": "{\"tooOldLogEventCount\": 3, \"tooNewLogEventCount\": 1,
 \"expiredLogEventCount\": 1}"
  }
}
```

El `rejectedLogRecords` campo es el número total de eventos rechazados. El `errorMessage` campo contiene un desglose codificado en JSON por motivo de rechazo:

- `tooOldLogEventCount`— Eventos con marcas de tiempo anteriores al período de retención

- `tooNewLogEventCount`— Eventos con marcas de tiempo demasiado lejanas en el futuro
- `expiredLogEventCount`— Eventos que caducaron durante el procesamiento

Prácticas recomendadas

Agrupación de eventos por lotes

Para un mejor rendimiento y eficiencia:

- Batch varios eventos en una sola solicitud cuando sea posible
- Tamaño de lote recomendado: de 10 a 100 eventos por solicitud
- Tamaño máximo de la solicitud: 1 MB

Gestión de errores

Implemente una gestión de errores adecuada en su aplicación. Códigos de estado HTTP comunes:

- `200 OK`— Los registros se ingirieron correctamente
- `400 Bad Request`— Formato o parámetros de solicitud no válidos
- `401 Unauthorized`— Token de portador no válido o caducado
- `403 Forbidden`— Permisos insuficientes
- `404 Not Found`— El grupo de registros o la transmisión no existen
- `429 Too Many Requests`— Se ha superado el límite de velocidad
- `500 Internal Server Error`— Error de servicio (reintento con un retraso exponencial)

Limitaciones

- Tamaño máximo del evento: 256 KB por evento
- Tamaño máximo de solicitud: 1 MB
- Número máximo de eventos por solicitud: 10 000
- Los nombres de los grupos de registros deben seguir las convenciones CloudWatch de nomenclatura de los registros
- La autenticación por token de portador debe estar habilitada en el grupo de registros si se utiliza la autenticación por token de portador.

Comparación de los puntos finales de ingestión de HTTP

Característica	Registros de HLC	Registros de ND-JSON	Registros JSON estructurados	OpenTelemetry Registros
Ruta	`/services/collector/event`	`/ingest/bulk`	`/ingest/json`	`/v1/logs`
Contenido-Tipo	`application/json`	`application/json o application/x-ndjson`	`application/json`	`application/json o application/x-protobuf`
Campo de fecha y hora	`"time"` (segundos)	`"timestamp"` (milisegundos)	`"timestamp"` (milisegundos)	`"timeUnixNano"` (nanosegundos)
Campos obligatorios	`"event"`	Ninguno	Ninguno	Estructura OTP () `"resourceLogs"`
Respuesta de éxito parcial	No	Sí	Sí	Sí
Compatibilidad con parámetros de consulta	Sí	Sí	Sí	No (solo encabezados)
Metadatos de entidad	Sí	Sí	Sí	No
Acepta primitivas	No	Sí	No	No
Análisis basado en líneas	No	Sí	No	No
Soporte para Protobug	No	No	No	Sí

Característica	Registros de HLC	Registros de ND-JSON	Registros JSON estructurados	OpenTelemetry Registros
Encabezado Retry-After	No	No	No	Sí

Elegir un punto final

- ¿Utiliza el formato HLC? Utilice los registros de HLC. Sus cargas útiles de HLC actuales funcionan con cambios mínimos.
- ¿Registros de streaming? line-by-line Utilice los registros de ND-JSON. Ideal para canalizaciones de registro que emiten un evento por línea. El más flexible: acepta cualquier tipo de valor JSON.
- ¿Envía cargas útiles JSON estructuradas? Utilice registros JSON estructurados. Ideal para aplicaciones que producen matrices o objetos JSON bien formados.
- ¿Ya lo estás usando? OpenTelemetry Usa OpenTelemetry registros. Acepta el formato OTLP, JSON o Protobuf y admite respuestas de éxito parcial con semántica de reintentos.

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 para C++	AWS SDK para C++ ejemplos de código
AWS CLI	AWS CLI ejemplos de código
AWS SDK para Go	AWS SDK para Go ejemplos de código
AWS SDK para Java	AWS SDK para Java ejemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript ejemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin ejemplos de código
AWS SDK para .NET	AWS SDK para .NET ejemplos de código
AWS SDK para PHP	AWS SDK para PHP ejemplos de código
Herramientas de AWS para PowerShell	Herramientas de AWS para PowerShell ejemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) ejemplos de código
AWS SDK para Ruby	AWS SDK para 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 SDKs.

> **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.

Administración de registros:

CloudWatch Logs proporciona funciones avanzadas de administración de registros que le ayudan a organizar, transformar y analizar los datos de registro de manera más eficaz. Estas funciones incluyen la centralización de datos entre cuentas y regiones, el descubrimiento automático de datos y la administración de esquemas, la transformación de los registros durante la ingestión y el análisis mejorado con facetas para la exploración interactiva de los registros.

Temas

- Descubrimiento y administración de fuentes de datos
- Funciones habilitadas por las fuentes de datos
- Orígenes de datos admitidos

Descubrimiento y administración de fuentes de datos

CloudWatch Logs descubre y clasifica automáticamente los datos de registro por fuente y tipo de datos, lo que facilita la comprensión y la administración de los registros a escala. Esta función proporciona la detección de esquemas para fuentes AWS vendidas, como Amazon VPC Flow Logs y Route 53 CloudTrail, así como herramientas de seguridad de terceros.

La consola de administración de registros proporciona una vista de alto nivel de los registros organizados por fuente y tipo de datos, en lugar de limitarse a grupos de registros. Esta organización le ayuda a:

- Consulta los registros clasificados por AWS servicios, fuentes de terceros (como Okta o CrowdStrike) y fuentes personalizadas
- Comprenda automáticamente el esquema y la estructura de sus datos de registro
- Cree políticas de indexación de campos basadas en los campos de esquema descubiertos
- Administre los registros de manera más eficiente en diferentes fuentes de datos
- Consulte los registros por diferentes fuentes de datos

Al habilitar el registro de CloudWatch registros para AWS los servicios compatibles, CloudWatch Logs aplica automáticamente el esquema correspondiente a sus registros. Esta aplicación de esquema automático ayuda a mantener la coherencia y proporciona información inmediata sobre la estructura de los registros.

¿Qué son las fuentes de datos de CloudWatch registros?

CloudWatch Las fuentes de datos de los registros son una función que proporciona una nueva forma de organizar y clasificar los datos de los registros en función de la fuente que los genera. Si bien CloudWatch Logs suele utilizar grupos de registros para organizar los registros, Data Sources ofrece una capa adicional de organización que agrupa los registros por servicio de origen y tipo de registro.

Cómo funcionan las fuentes de datos

Las fuentes de datos proporcionan una organización de registros basada en servicios y un descubrimiento simplificado en toda su AWS infraestructura. Puede localizar fácilmente los registros de servicios específicos y filtrarlos por tipo de registro sin necesidad de conocer los nombres o estructuras individuales de los grupos de registros.

Para las fuentes de terceros y, opcionalmente, para las fuentes de registros de aplicaciones, las fuentes de datos funcionan con CloudWatch canalizaciones para clasificar los registros. Al configurar una canalización para ingerir y transformar los registros, se especifica el nombre y el tipo de la fuente de datos. CloudWatch A continuación, Logs clasifica automáticamente todos los registros que procesa la canalización. Para obtener más información, consulta CloudWatch las canalizaciones en la Guía del CloudWatch usuario de Amazon.

Las fuentes de datos clasifican los registros mediante dos identificadores clave:

- Nombre de la fuente de datos: el AWS servicio, la fuente de terceros o la aplicación que genera los registros (por ejemplo, Route 53, Amazon VPC CloudTrail, Okta SSO o Falcon). CrowdStrike
- Tipo de fuente de datos: el tipo específico de registro generado por ese servicio.

Un esquema define la estructura de los datos de registro, incluidos los campos que están presentes y la forma en que se organiza la información. Una sola fuente de datos puede producir varios tipos de registros con diferentes esquemas y propósitos. Por ejemplo, la fuente de AWS CloudTrail datos tiene dos tipos: eventos de administración (que rastrean las operaciones del plano de control, como la creación o eliminación de recursos) y eventos de datos (que rastrean las operaciones del plano de datos, como el acceso a objetos desde S3). Cada tipo tiene un esquema diferente porque capturan distintos tipos de información.

Cómo comenzar

CloudWatch Los registros clasifican los registros en fuentes de datos según su origen. El método depende del tipo de registros con los que trabajes:

Servicio de AWS registros

Los registros compatibles Servicios de AWS se agrupan automáticamente por fuente de datos sin necesidad de realizar ninguna configuración. CloudWatch Logs reconoce estos registros y aplica el nombre y el tipo de fuente de datos adecuados en función del servicio de origen.

Third-party registros

Third-party los registros requieren canalizaciones para la categorización de las fuentes de datos. Cuando configura una canalización para ingerir registros de fuentes de terceros compatibles, como Microsoft Office 365, Okta o Palo Alto Networks CrowdStrike, especifica el nombre y el tipo de la fuente de datos en la configuración de la canalización. CloudWatch Logs clasifica automáticamente todos los registros que procesa la canalización utilizando esos identificadores.

Opcionalmente, Pipelines puede transformar los registros de terceros al formato Open Cybersecurity Schema Framework (OCSF) para un análisis estandarizado de los eventos de seguridad. Cuando la transformación de OCSF está habilitada, el nombre y el tipo de la fuente de datos se determinan automáticamente en función del mapeo del esquema de OCSF. Sin la transformación OCSF, debe especificar el nombre y el tipo de la fuente de datos en la configuración de la canalización.

Registros de aplicaciones

En el caso de los registros de aplicaciones personalizados, puede clasificarlos por fuente de datos mediante uno de estos métodos:

- Etiquetas de grupos de registros: añada etiquetas a sus grupos de registros utilizando las claves **`cw:datasource:name`** y especificando el nombre y **`cw:datasource:type`** el tipo de la fuente de datos, respectivamente, para todos los registros ingeridos en el grupo de registros. Los valores de las etiquetas pueden tener un máximo de 64 caracteres y solo pueden contener letras minúsculas, números y caracteres de subrayado. Deben empezar por una letra o un número y no pueden contener guiones dobles (__).
- Configuración de canalización: configure la información de la fuente de datos mediante canalizaciones de procesamiento de registros al ingerir los registros de las aplicaciones.

> **Note**
>
> Los nombres de las fuentes de datos no pueden empezar por «aws» o «amazon» para evitar conflictos con los registros de AWS servicio.

Campos del sistema

CloudWatch Los registros agregan automáticamente tres campos del sistema a los registros clasificados por fuente de datos. Estos campos sirven como facetas predeterminadas:

- `@data_source_name`- Contiene el nombre de la fuente de datos o «Desconocido» si no se ha determinado
- `@data_source_type`- Contiene el tipo de fuente de datos o «Desconocido» si no se ha determinado
- `@data_format`- Indica el formato de los datos de registro

Si no se puede determinar el nombre o el tipo de la fuente de datos, estos campos se configuran como «Desconocido». Las fuentes de datos con valores «desconocidos» siguen visibles en las facetas y en la tabla de fuentes de datos situada en la sección «Administración de registros» de la consola, lo que permite identificar los registros no clasificados y del grupo de registros del que provienen.

El `@data_format` campo puede contener uno de los siguientes valores:

- `Default`- Registros ingeridos sin modificación.
- `Custom`- Los registros se procesan a través de procesadores de canalización o los registros se ingieren en un grupo de registros con etiquetas de fuentes name/type de datos.
- `OCSF-<version>`- Los registros se procesan con procesadores OCSF (Open Cybersecurity Schema Framework) en canalizaciones.
- `AWS-OTEL-LOG-V<version>`- OpenTelemetry registros ingeridos a través del punto final OTLP CloudWatch .
- `AWS-OTEL-TRACE-V<version>`- los OpenTelemetry rastros ingeridos a través del punto final de la OTLP CloudWatch .

Estos campos del sistema le permiten filtrar y consultar los registros en función de su origen y formato, lo que facilita el trabajo con registros de diferentes orígenes y procesos de procesamiento.

Acceso a orígenes de datos

Consola

En la consola de CloudWatch registros, se utiliza la pestaña Administración de registros para acceder a las fuentes de datos. CloudWatch Logs consolida automáticamente los datos de registro por fuentes y tipos de datos, y descubre continuamente los datos recién ingresados. Desde la lista de fuentes de datos, puede crear canalizaciones y definir índices y facetas de campos.

AWS CLI

Usa el siguiente comando para enumerar las distintas fuentes de datos y tipos de registros de tu cuenta:

```
aws logs list-aggregate-log-group-summaries --group-by DATA_SOURCE_NAME_AND_TYPE
```

Relación con los grupos de registros

Las fuentes de datos complementan, no sustituyen, a los grupos de registros. Los registros se siguen almacenando en grupos de registros como antes, pero ahora también se etiquetan automáticamente con la información de la fuente de datos. Esta organización dual le permite:

- Utilice grupos de registros para obtener políticas detalladas de retención y control de acceso
- Utilice las fuentes de datos para el descubrimiento y el análisis de registros basados en servicios
- Consulte los registros mediante cualquiera de los dos métodos organizativos en función de sus necesidades

Las fuentes de datos facilitan el trabajo con los registros a escala, ya que proporcionan una visión centrada en el servicio de los datos de registro en toda la infraestructura. AWS

Funciones habilitadas por las fuentes de datos

Las fuentes de datos permiten capacidades avanzadas de procesamiento y análisis de registros mediante el descubrimiento de campo y estructuras de datos consistentes.

- Facetas: las facetas son campos de registro indexados que proporcionan filtrado y análisis interactivos sin necesidad de escribir consultas. CloudWatch Los registros crean automáticamente

facetas para el nombre y el tipo de la fuente de datos, y puede crear políticas de facetas en los campos detectados para acelerar la solución de problemas. Las facetas muestran las distribuciones de valores y los recuentos en CloudWatch Logs Insights, lo que facilita la identificación de patrones mediante una exploración basada en el criterio de selección.

- Canalizaciones: cree canalizaciones de transformación que se apliquen a todos los registros de un nombre y tipo de fuente de datos específicos. Esto le permite definir reglas de procesamiento coherentes para los registros de la misma fuente.
- Detección de campos: CloudWatch los registros descubren automáticamente los campos y sus tipos de datos para cada combinación de nombre y tipo de fuente de datos en función de los procesadores de canalización. En el AWS caso de los registros gestionados, las estructuras de campos están predefinidas. Para los registros de aplicaciones, recomendamos mantener formatos de registro consistentes para maximizar la compatibilidad con las herramientas de análisis, como las tablas de Amazon S3, que requieren estructuras de campo bien definidas.

Puede ver la lista completa de campos y sus tipos para cualquier fuente de datos mediante la `GetLogFields` API:

```
aws logs get-log-fields --data-source-name <name>  --data-source-type <type>
```

Esta detección y coherencia de los campos permiten realizar análisis e integraciones avanzados, ya que las herramientas externas pueden trabajar con estructuras de campo predecibles al procesar los datos de registro.

Orígenes de datos admitidos

CloudWatch Logs admite una amplia gama de fuentes de datos, incluidos AWS servicios, herramientas de TI y seguridad de terceros y fuentes de aplicaciones personalizadas. Esta página proporciona un catálogo completo de todas las fuentes de datos compatibles. Para obtener información sobre cómo se descubren y administran las fuentes de datos, consultethe section called “Descubrimiento y administración de fuentes de datos”.

Temas

- compatible Servicios de AWS para las fuentes de datos
- Fuentes de terceros compatibles para las fuentes de datos
- Orígenes personalizados

compatible Servicios de AWS para las fuentes de datos

En la siguiente tabla se enumeran las Servicios de AWS que los CloudWatch registros clasifican automáticamente como fuentes de datos:

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
amazon_api_gateway	access
amazon_bedrock_agentcore	browser_usage
amazon_bedrock_agentcore	code_interpreter_application
amazon_bedrock_agentcore	code_interpreter_usage
amazon_bedrock_agentcore	gateway_application
amazon_bedrock_agentcore	identity_workload_application
amazon_bedrock_agentcore	memory_application
amazon_bedrock_agentcore	online_evaluation_config
amazon_bedrock_agentcore	runtime_application
amazon_bedrock_agentcore	runtime_usage
amazon_bedrock_agents	application
amazon_bedrock_agents	event
amazon_bedrock_knowledge_bases	application
amazon_cloudfront	access
amazon_cloudfront	connection
amazon_cloudwatch	rum_app_monitor
amazon_cognito	user_pool

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
amazon_ec2	verified_access
amazon_eks	api_server
amazon_eks	audit
amazon_eks	authenticator
amazon_eks	controller_manager
amazon_eks	scheduler
amazon_elasticache	cluster
amazon_eventbridge	eventbus_error
amazon_eventbridge	eventbus_info
amazon_eventbridge	pipes_execution
amazon_interactive_video_service	chat
amazon_managed_prometheus	scraper
amazon_managed_prometheus	workspace
amazon_msk	broker
amazon_msk	connect
amazon_opensearch_service	pipeline
amazon_q_business	events
amazon_q_business	sync_job
amazon_q_connect	events
amazon_route53	global_resolver_query

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
amazon_route53	hosted_zones
amazon_route53	profiles_resolver_query
amazon_route53	resolver_query
amazon_sagemaker	workteam_activity
amazon_ses	ingress_endpoints
amazon_ses	rule_sets
amazon_ses	traffic_policy
amazon_vpc	flow
amazon_vpc	route_server_peer
amazon_vpc_lattice	access
amazon_vpc_lattice	resource_access
amazon_workmail	access_control
amazon_workmail	authentication
amazon_workmail	personal_access
amazon_workmail	workmail_access
amazon_workmail	workmail_availability
aws_b2b_data_interchange	execution
aws_backup	data_access
aws_backup	hypervisor
aws_clean_rooms	analysis

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
aws_client_vpn	connection
aws_client_vpn	event
aws_cloudtrail	data
aws_cloudtrail	management
aws_elemental_mediapackage	egress_access
aws_elemental_mediapackage	ingress_access
aws_elemental_mediatailor	ad_decision
aws_elemental_mediatailor	manifest
aws_elemental_mediatailor	transcode
aws_entity_resolution	id_mapping_workflow
aws_entity_resolution	matching_workflow
aws_iot_fleetwise	error
aws_mainframe_modernization	batch_job
aws_mainframe_modernization	config
aws_mainframe_modernization	console
aws_mainframe_modernization	dataset_import
aws_network_firewall	alert
aws_network_firewall	flow
aws_network_firewall	tls
aws_nlb	access

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
aws_pcs	job_completion
aws_pcs	scheduler
aws_security_hub	compliance_finding
aws_security_hub	data_security_finding
aws_security_hub	detection_finding
aws_security_hub	vulnerability_finding
aws_security_hub_cspm	asff_finding
aws_shield	protection_flow
aws_step_functions	express
aws_step_functions	standard
aws_transfer_family	server
aws_waf	access

Fuentes de terceros compatibles para las fuentes de datos

En la siguiente tabla se enumeran las fuentes de terceros que los CloudWatch registros clasifican automáticamente como fuentes de datos cuando se ingieren a través de canalizaciones:

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
akamai_datastream_2	base_event
akamai_datastream_2	dns_activity
akamai_datastream_2	http_activity

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
cisco_meraki	api_activity
cisco_meraki	detection_finding
cisco_meraki	network_activity
cisco_umbrella	data_security_finding
cisco_umbrella	dns_activity
cisco_umbrella	entity_management
cisco_umbrella	network_activity
crowdstrike_falcon	detection_finding
crowdstrike_falcon	process_activity
drupal_core	application_lifecycle
drupal_core	authentication
drupal_core	entity_management
drupal_core	http_activity
entrust_idaas	authentication
entrust_idaas	entity_management
f5_bigip	http_activity
f5_bigip	network_activity
github_auditlogs	account_change
github_auditlogs	api_activity
github_auditlogs	entity_management

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
microsoft_entraid	account_change
microsoft_entraid	authentication
microsoft_entraid	entity_management
microsoft_entraid	user_access_management
microsoft_office365	account_change
microsoft_office365	application_lifecycle
microsoft_office365	authentication
microsoft_office365	compliance_finding
microsoft_office365	detection_finding
microsoft_office365	email_activity
microsoft_office365	file_hosting_activity
microsoft_office365	group_management
microsoft_office365	incident_finding
microsoft_office365	user_access_management
microsoft_office365	vulnerability_finding
microsoft_office365	web_resources_activity
microsoft_windows	account_change
microsoft_windows	authentication
microsoft_windows	entity_management
microsoft_windows	event_log_activity

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
microsoft_windows	file_system_activity
microsoft_windows	group_management
microsoft_windows	kernel_activity
okta_auth0	api_activity
okta_auth0	authentication
okta_sso	api_activity
okta_sso	authentication
okta_sso	detection_finding
okta_sso	entity_management
onelogin_identity	account_change
onelogin_identity	authentication
onelogin_identity	entity_management
paloaltonetworks_nextgenerationfirewall	authentication
paloaltonetworks_nextgenerationfirewall	detection_finding
paloaltonetworks_nextgenerationfirewall	network_activity
paloaltonetworks_nextgenerationfirewall	process_activity
pingidentity_pingone	account_change

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
pingidentity_pingone	authentication
pingidentity_pingone	entity_management
sentinelone_endpointsecurity	dns_activity
sentinelone_endpointsecurity	file_system_activity
sentinelone_endpointsecurity	http_activity
sentinelone_endpointsecurity	process_activity
servicenow_cmdb	api_activity
servicenow_cmdb	datastore_activity
servicenow_cmdb	entity_management
wiz_cnapp	api_activity
wiz_cnapp	authentication
wiz_cnapp	compliance_finding
wiz_cnapp	detection_finding
wiz_cnapp	vulnerability_finding
zeek	authentication
zeek	base_event
zeek	detection_finding
zeek	dhcp_activity
zeek	dns_activity
zeek	email_activity

Nombre de la fuente de datos (campo @data_source_name)	Tipo de fuente de datos (campo @data_source_type)
zeek	ftp_activity
zeek	http_activity
zeek	network_activity
zeek	rdp_activity
zeek	smb_activity
zeek	software_inventory_info
zeek	ssh_activity
zeek	tunnel_activity
zscaler_internetaccess	authentication
zscaler_internetaccess	dns_activity
zscaler_internetaccess	http_activity
zscaler_internetaccess	network_activity

Fuentes adicionales de terceros a través de AWS CSPM de Security Hub

Los hallazgos de seguridad adicionales de terceros están disponibles a través de AWS la integración de Security Hub CSPM. Los siguientes socios envían los resultados a Security Hub CSPM, que luego están disponibles como fuentes de datos en CloudWatch los registros. Para obtener información detallada sobre estas integraciones, consulte las integraciones de Third-party productos con Security Hub CSPM en la Guía del usuario de AWS Security Hub.

Partner	Integración
3CoreSec — NTA	Envía los resultados a través de Security Hub (CSPM)

Partner	Integración
Alert Logic: gestión de amenazas sin SIEM	Envía los resultados a través de Security Hub (CSPM)
Aqua Security: plataforma de seguridad nativa de la nube	Envía los resultados a través de Security Hub (CSPM)
Aqua Security — Kube-bench	Envía los resultados a través de Security Hub (CSPM)
Armadura: armadura en cualquier lugar	Envía los resultados a través de Security Hub (CSPM)
AttackIQ	Envía los resultados a través de Security Hub (CSPM)
Barracuda Networks: guardián de la seguridad en la nube	Envía los resultados a través de Security Hub (CSPM)
BigID — BigID Enterprise	Envía los resultados a través de Security Hub (CSPM)
Hexágono azul	Envía los resultados a través de Security Hub (CSPM)
Punto de control: CloudGuard IaaS	Envía los resultados a través de Security Hub (CSPM)
Check Point: CloudGuard gestión de la postura	Envía los resultados a través de Security Hub (CSPM)
Claridad — Domo	Envía los resultados a través de Security Hub (CSPM)
Seguridad del almacenamiento en la nube: antivirus para Amazon S3	Envía los resultados a través de Security Hub (CSPM)
Contrast Security: Contrast Assess	Envía los resultados a través de Security Hub (CSPM)

Partner	Integración
CrowdStrike — Halcón CrowdStrike	Envía los resultados a través de Security Hub (CSPM)
CyberArk — Análisis de amenazas privilegiado	Envía los resultados a través de Security Hub (CSPM)
Teorema de los datos	Envía los resultados a través de Security Hub (CSPM)
Drata	Envía los resultados a través de Security Hub (CSPM)
Forcepoint — CASB	Envía los resultados a través de Security Hub (CSPM)
Forcepoint: puerta de enlace de seguridad en la nube	Envía los resultados a través de Security Hub (CSPM)
Forcepoint — DLP	Envía los resultados a través de Security Hub (CSPM)
Forcepoint — NGFW	Envía los resultados a través de Security Hub (CSPM)
Fuga	Envía los resultados a través de Security Hub (CSPM)
Guardicore — Centra	Envía los resultados a través de Security Hub (CSPM)
HackerOne — Inteligencia sobre vulnerabilidades	Envía los resultados a través de Security Hub (CSPM)
JFrog — Radiografía	Envía los resultados a través de Security Hub (CSPM)
Juniper Networks: firewall vSRX de próxima generación	Envía los resultados a través de Security Hub (CSPM)

Partner	Integración
k9 Security: analizador de acceso	Envía los resultados a través de Security Hub (CSPM)
Encajes	Envía los resultados a través de Security Hub (CSPM)
McAfee — MVISION CHNAPP	Envía los resultados a través de Security Hub (CSPM)
NETSCOUT: investigador cibernético	Envía los resultados a través de Security Hub (CSPM)
Orca: plataforma de seguridad en la nube	Envía los resultados a través de Security Hub (CSPM)
Palo Alto Networks: Prisma Cloud Compute	Envía los resultados a través de Security Hub (CSPM)
Palo Alto Networks: Prisma Cloud Enterprise	Envía los resultados a través de Security Hub (CSPM)
Plerion: plataforma de seguridad en la nube	Envía los resultados a través de Security Hub (CSPM)
Merodeador	Envía los resultados a través de Security Hub (CSPM)
Qualys: gestión de vulnerabilidades	Envía los resultados a través de Security Hub (CSPM)
Rapid7 — InsightVM	Envía los resultados a través de Security Hub (CSPM)
SentinelOne	Envía los resultados a través de Security Hub (CSPM)
Snyk	Envía los resultados a través de Security Hub (CSPM)

Partner	Integración
Sonrai Security — Sonrai Dig	Envía los resultados a través de Security Hub (CSPM)
Sophos: protección de servidores	Envía los resultados a través de Security Hub (CSPM)
StackRox — Seguridad de Kubernetes	Envía los resultados a través de Security Hub (CSPM)
Sumo Logic: análisis de datos de máquinas	Envía los resultados a través de Security Hub (CSPM)
Symantec: protección de la carga de trabajo en la nube	Envía los resultados a través de Security Hub (CSPM)
Tenable.io	Envía los resultados a través de Security Hub (CSPM)
Trend Micro – Cloud One	Envía los resultados a través de Security Hub (CSPM)
Vectra — Cognito Detect	Envía los resultados a través de Security Hub (CSPM)
Wiz	Envía los resultados a través de Security Hub (CSPM)
Caveonix — Caveonix Cloud	Envía y recibe los resultados a través de Security Hub (CSPM)
Cloud Custodian	Envía y recibe los resultados a través de Security Hub (CSPM)

Partner	Integración
DisruptOps	Envía y recibe los resultados a través de Security Hub (CSPM)
Kion	Envía y recibe los resultados a través de Security Hub (CSPM)
Turbot	Envía y recibe los resultados a través de Security Hub (CSPM)

> **Note**
>
> Esta lista refleja las integraciones de los socios de Security Hub que envían las conclusiones en el momento de redactar este artículo. Dado que AWS Security Hub añade nuevas integraciones de socios con regularidad, consulte las integraciones de Third-party productos con Security Hub CSPM en la Guía del usuario de AWS Security Hub para obtener la lista más actualizada de los socios disponibles.

Orígenes personalizados

Para los registros de aplicaciones personalizados, puede definir su propia categorización de fuentes de datos mediante uno de los siguientes métodos:

- Etiquetas de grupos de registros: agregue etiquetas a sus grupos de registros mediante las claves **`cw:datasource:name`** y **`cw:datasource:type`** especifique el nombre y el tipo de la fuente de datos para todos los registros incluidos en el grupo de registros.
- Configuración de canalización: configure la información de la fuente de datos mediante canalizaciones de procesamiento de registros al ingerir los registros de las aplicaciones.

> **Note**
>
> Los nombres de las fuentes de datos personalizadas no pueden empezar por «aws» o «amazon» para evitar conflictos con los registros de AWS servicio.

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. Se pueden realizar consultas que le ayuden a responder de forma más eficaz a los problemas de funcionamiento. Además de realizar consultas mediante grupos de registros, puede realizar consultas mediante facetas, fuentes de datos y tipos de datos. Si se produce un problema, puede utilizar CloudWatch Logs Insights para identificar las posibles causas y validar las soluciones implementadas. Está limitado a 100 CloudWatch Logs Insights QL simultáneos por cuenta, incluidas las consultas añadidas a los paneles. Además, puede ejecutar 15 consultas simultáneas para OpenSearch Service PPL o Service SQL. OpenSearch

CloudWatch Logs Insights admite tres lenguajes de consulta que puede usar para sus consultas:

- Un lenguaje de consulta de información de registros (Logs Insights QL) de creación específica con algunos comandos sencillos pero eficaces.
- OpenSearch Lenguaje de procesamiento canalizado (PPL) de Service. OpenSearch El PPL le permite analizar sus registros mediante un conjunto de comandos delimitados por canalizaciones (|).

  Con OpenSearch PPL, puede recuperar, consultar y analizar datos mediante comandos que se agrupan entre sí, lo que facilita la comprensión y la redacción de consultas complejas. La sintaxis permite encadenar comandos para transformar y procesar datos. Con PPL, se pueden filtrar y agregar datos, así como utilizar un amplio conjunto de funciones matemáticas, de cadenas, de fecha, condicionales y de otro tipo para el análisis.
- OpenSearch Lenguaje de consulta estructurado (SQL) de servicio. Con las consultas OpenSearch SQL, puede analizar sus registros de forma declarativa. Se pueden usar comandos como SELECT, FROM, WHERE, GROUP BY, HAVING y varios otros comandos y funciones disponibles en SQL. Se pueden ejecutar JOIN en grupos de registros, correlacionar datos entre registros mediante subconsultas y utilizar el amplio conjunto de funciones JSON, matemáticas, de cadena, condicionales y otras funciones de SQL para realizar análisis eficaces de los registros.

  Cuando utilice comandos de SQL o PPL, asegúrese de incluir los campos con caracteres especiales (no alfabéticos ni numéricos) entre acentos graves para consultarlos correctamente. Por ejemplo, incluya `@message`, `Operation.Export` y `Test::Field` entre acentos graves. No es necesario incluir los campos con nombres exclusivamente alfabéticos entre comillas simples.

CloudWatch Logs Insights ofrece las siguientes funciones que están disponibles para su uso con cualquiera de los lenguajes de consulta.

- Descubrimiento automático de campos de registro en registros de AWS servicios como Amazon Route 53 y Amazon VPC AWS Lambda AWS CloudTrail, y de cualquier aplicación o registro personalizado que emita eventos de registro como JSON.
- Creación de índices de campos para reducir los costos y acelerar los resultados, en especial para consultas de un gran número de grupos de registros o eventos de registro. Después de crear índices de campos que son comunes en sus eventos de registro, puede usarlos en una consulta. La consulta omite el procesamiento de los eventos de registro que se sabe que no incluyen el campo indexado y procesa menos datos.

  > **Note**
  >
  > El comando `filterIndex` solo está disponible en lenguaje de consulta de Información de registros.

- Detección y análisis de patrones en los eventos de registro. Un patrón es una estructura de texto compartida que se repite entre los campos de registro. Al ver los resultados de una consulta, puede elegir la pestaña Patrones para ver los patrones que los CloudWatch registros encontraron a partir de una muestra de sus resultados.
- Guardar consultas, ver el historial de consultas, volver a ejecutar las consultas guardadas y utilizar las consultas guardadas con parámetros.
- Añadir consultas a los paneles.
- Cifrar los resultados de las consultas con. AWS Key Management Service
- La generación de consultas mediante lenguaje natural le permite utilizar un lenguaje natural para crear consultas de CloudWatch Logs Insights. Puede hacer preguntas o describir los datos que busca y, después, la IA genera una consulta según una petición y proporciona una explicación línea por línea sobre cómo funciona la consulta.
- Usa facetas para agrupar, filtrar y explorar tus registros de forma interactiva.
- Los registros circundantes le permiten ver las líneas de registro antes y después de cualquier registro de registro específico en los resultados de la consulta para obtener un contexto instantáneo sobre los eventos de registro críticos. Puede configurar el rango para ver 5, 10, 20, 50 o 100 líneas de registro antes y después de un registro seleccionado, y buscar palabras clave específicas en los registros circundantes.

Las siguientes funciones de CloudWatch Logs Insights solo se admiten cuando utiliza Logs Insights QL.

- Consultas de comparación que comparan los eventos de registro de un grupo de registro con los eventos de registro de un período anterior.

> **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.

Si ha iniciado sesión en una cuenta configurada como una cuenta de monitoreo 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 monitoreo. Puede ejecutar una consulta que se ejecute en varios grupos de registro ubicados en diferentes cuentas. Para obtener más información, consulte Observabilidad entre cuentas de CloudWatch .

Al crear consultas con Logs Insights QL, también puede usar un lenguaje natural para crear consultas de CloudWatch Logs Insights. Para ello, pregunte o describa los datos que busca. Esta AI-assisted capacidad genera una consulta en función de su solicitud y proporciona una explicación línea por línea de cómo funciona la consulta. Para obtener más información, consulte Usar un lenguaje natural para generar y actualizar las consultas de CloudWatch Logs Insights.

Las consultas que utilizan cualquiera de los lenguajes de consultas compatibles expiran después de 60 minutos, si no se han completado. Los resultados de las consultas están disponibles durante siete días.

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

Puedes 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**
>
> 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

Idiomas de consulta compatibles

En las secciones siguientes se enumeran los comandos compatibles con cada lenguaje de consulta. También se describen el formato de sintaxis y se proporcionan ejemplos de consultas.

Temas

CloudWatch Lenguaje de consulta de Logs Insights (Logs Insights QL)

En esta sección se incluye la documentación completa de los comandos y funciones de lenguaje de consulta de Información de registros. También se incluyen ejemplos de consultas para este lenguaje.

Para obtener información sobre otros lenguajes de consulta que puede usar, consulte OpenSearch Service PPL, OpenSearch Service SQL y CloudWatch Metrics Insights.

Temas

CloudWatch Registra la sintaxis de consulta del lenguaje Insights

En esta sección, se proporcionan detalles sobre el lenguaje de consulta de Información 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.

> **Important**
>
> Para evitar generar cargos excesivos al ejecutar consultas extensas, tenga en cuenta las siguientes prácticas recomendadas:
>
> - Seleccione solo los grupos de registro necesarios para cada consulta.
> - Especifique siempre el intervalo de tiempo más breve posible para sus consultas.
> - Cuando utilice la consola para ejecutar consultas, cancele todas las consultas antes de cerrar la página de la consola de CloudWatch Logs Insights. De lo contrario, las consultas seguirán ejecutándose hasta que se completen.
> - Cuando añada un widget de CloudWatch Logs Insights a un panel, asegúrese de que el panel no se actualice con una frecuencia elevada, ya que cada actualización inicia una nueva consulta.

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 registros 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 Logs Insights QL se admiten en los grupos de registro de la clase de registro Estándar. Los grupos de registro de la clase de registro de acceso poco frecuente admiten todos los comandos de consulta de Logs Insights QL, excepto `pattern`, `diff` y `unmask`.

`anomaly`	Identifica patrones poco comunes en sus datos de registro mediante machine learning.
`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.
`filterIndex`	Hace que una consulta intente escanear solo los grupos de registros que están indexados en el campo mencionado en un índice de campo y

	que también contienen un valor para ese índice de campo. Esto reduce el volumen analizado al intentar analizar solo los eventos de registro de estos grupos de registros que contienen el valor especificado en la consulta para este índice de campo. Este comando no es compatible con los grupos de registro de la clase de registro de acceso poco frecuente.
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 del patrón.
diff	Permite comparar 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`).
SOURCE	Incluirlo SOURCE en una consulta es una forma útil de especificar una gran cantidad de grupos de registros para incluirlos en una consulta en función del prefijo del nombre del grupo de registros, los identific adores de cuenta, la clase del grupo de registros, las fuentes de datos o las etiquetas de los grupos de registros. Este comando solo se admite cuando se crea una consulta en la consola AWS CLI o mediante programación, no en la consola. CloudWatch
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.
unnest	Aplana una lista tomada como entrada para generar varios registros con un único registro para cada elemento de la lista.
lookup	Enriquece los eventos del registro con datos de una tabla de consulta haciendo coincidir los valores de los campos. Utilice las tablas de consulta para añadir datos de referencia, como detalles de usuario, nombres de aplicaciones o información de productos, a los resultados de la consulta.
join	Combina los eventos de registro de un grupo de registros de origen con los eventos de otro grupo de registros o el resultado de una consulta en función de un campo coincidente. Utilice el comando join para correlaci onar los eventos de registro relacionados entre distintas fuentes utilizand o claves comunes entre ellas, como los identificadores de solicitud o los ID de transacción coincidentes.
subqueries	Una subconsulta es una consulta anidada de Logs Insights que se puede utilizar como entrada para otra consulta. Las subconsultas se pueden usar para derivar conjuntos de resultados intermedios que luego son consumidos por los comandos posteriores.
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

Comandos de lenguaje de consulta de Información de registros compatibles con las clases de registro

Todos los comandos de consulta de Logs Insights QL se admiten en los grupos de registro de la clase de registro Estándar. Los grupos de registro de la clase de registro de acceso poco frecuente admiten todos los comandos de consulta excepto `pattern`, `diff`, `filterIndex` y `unmask`.

anomalía

Se debe utilizar `anomaly` para identificar de manera automática los patrones inusuales y posibles problemas en sus datos de registro mediante machine learning.

El comando `anomaly` amplía la funcionalidad `pattern` existente y aprovecha los análisis avanzados para ayudar a identificar posibles anomalías en los datos de registro. Se puede utilizar `anomaly` para reducir el tiempo que se tarda en identificar y resolver los problemas operativos al mostrar de manera automática patrones o comportamientos inusuales en sus registros.

El comando `anomaly` funciona con el comando `pattern` para identificar primero los patrones de registro y, a continuación, detectar las anomalías en esos patrones. También puede combinar `anomaly` con los comandos `filter` y `sort` o para concentrar la detección de anomalías en subconjuntos específicos de los datos.

Entrada de comandos de anomalías

El comando `anomaly` se suele utilizar después del comando `pattern` para analizar los patrones identificados en los datos de registro. El comando no exige la presencia de parámetros adicionales y analiza el resultado de los comandos anteriores de la consulta.

Tipos de anomalías identificadas

El comando `anomaly` identifica cinco tipos distintos de anomalías:

- Anomalías de frecuencia de patrones: frecuencias inusuales de patrones de registro específicos, como cuando una aplicación comienza a generar más mensajes de error de lo habitual.
- Nuevas anomalías en los patrones: patrones de registro nunca antes vistos que pueden indicar la aparición de nuevos tipos de errores o mensajes en los registros.
- Anomalías en la variación de los token: cambios inesperados en el contenido de los mensajes de registro que pueden indicar variaciones inusuales en los formatos de registro esperados.
- Anomalías numéricas en los tokens: cambios inusuales en los valores numéricos de los registros que pueden ayudar a detectar posibles problemas de rendimiento o variaciones inesperadas en las métricas.
- Anomalías en los códigos de error HTTP: patrones relacionados con las respuestas a los errores HTTP, útiles en especial a la hora de supervisar aplicaciones web y API.

Salida de comandos de anomalías

El comando `anomaly` conserva todos los campos de los datos de entrada y añade los resultados de la detección de anomalías para ayudar a identificar patrones inusuales en los datos de registro.

Ejemplos

El siguiente comando identifica los patrones en los datos de registro y, a continuación, detecta las anomalías en esos patrones:

```
fields @timestamp, @message
| pattern @message
| anomaly
```

El comando `anomaly` se puede utilizar junto con el filtrado para centrarse en tipos de registro específicos:

```
fields @timestamp, @message
| filter @type = "REPORT"
| pattern @message
| anomaly
```

El comando `anomaly` se puede combinar con la ordenación para organizar los resultados:

```
fields @timestamp, @message
| filter @type = "ERROR"
| pattern @message
| anomaly
| sort @timestamp desc
```

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
```

Se debe utilizar `fields` en lugar de `display` cuando se quieran 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
```

Campos indexados y comando de filtro

Si ha creado índices de campos para un grupo de registros, puede aprovechar esos índices de campo para aumentar la eficacia de las consultas de `filter` y reducir el volumen digitalizado. Supongamos que se ha creado un índice de campo para `requestId`. Luego, cualquier consulta de CloudWatch Logs Insights sobre ese grupo de registros que `filter requestId IN [`*`value`*`, `*`value`*`, ...]` incluya `filter requestId =` *`value`* o intente omitir el procesamiento de eventos de registro que se sepa que no incluyen el campo indexado. Al intentar analizar solo los eventos de

registro que se sabe que contienen ese campo indexado, se puede reducir el volumen de análisis y la consulta es más rápida.

Para obtener más información sobre los índices de campo y cómo crearlos, consulte Creación de índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de análisis.

> **Important**
>
> Solo las consultas que incluyan `filter` *`fieldName`* `=...` y `filter` *`fieldName`* `IN...` se verán beneficiadas con las mejores del índice de campo. Las consultas que incluyen `filter` *`fieldName`* `like` no emplean índices y siempre analizan todos los eventos de registros en los grupos de registros selectos.

Ejemplo: búsqueda de eventos de registro relacionados con un identificador de solicitud determinado mediante índices

En este ejemplo se supone que ha creado un índice de campo en `requestId`. En el caso de los grupos de registros que utilizan este índice de campos, la consulta aprovechará los índices de campo para intentar analizar la menor cantidad posible de eventos de registro con los que buscar eventos con `requestId` y un valor de `123456`

```
fields @timestamp, @message
| filter requestId = "1234656"
| 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/
```

filterIndex

Se usa `filterIndex` para devolver solo datos indexados, mediante el forzado de una consulta a analizar solo los grupos de registros que están indexados en un campo que se especifique en la

consulta. Para los grupos de registros que están indexados en este campo, se optimiza aún más la consulta al omitir los grupos de registros que no tienen ningún evento de registro que contenga el campo especificado en la consulta del campo indexado. Se reduce aún más el volumen analizado al intentar analizar solo los eventos de registro de estos grupos de registros que coincidan con el valor especificado en la consulta para este índice de campos. Para obtener más información sobre los índices de campo y cómo crearlos, consulte Creación de índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de análisis.

El uso de `filterIndex` con campos indexados puede ayudar a consultar grupos de registros que incluyen petabytes de datos de registro de manera eficiente, ya que limita el espacio de búsqueda real a los grupos de registros y los eventos de registro que tienen índices de campo.

Por ejemplo, supongamos que se ha creado un índice de campos para `IPaddress` en algunos de los grupos de registro de su cuenta. A continuación, se puede crear la siguiente consulta y elegir consultar todos los grupos de registros de la cuenta para buscar eventos de registro que incluyan el valor `198.51.100.0` del campo `IPaddress`.

```
fields @timestamp, @message
| filterIndex IPaddress = "198.51.100.0"
| limit 20
```

El comando `filterIndex` hace que esta consulta intente omitir todos los grupos de registros que no están indexados para `IPaddress`. Además, dentro de los grupos de registros que están indexados, la consulta omite los eventos de registro que tienen un campo `IPaddress`, pero que no consideran a `198.51.100.0` como el valor de ese campo.

Utilice el operador `IN` para ampliar los resultados a cualquiera de los múltiples valores de los campos indexados. El siguiente ejemplo busca eventos de registro que incluyen el valor `198.51.100.0` o `198.51.100.1` en el campo `IPaddress`.

```
fields @timestamp, @message
| filterIndex IPaddress in ["198.51.100.0", "198.51.100.1"]
| limit 20
```

CloudWatch Los registros proporcionan índices de campos predeterminados para todos los grupos de registros de la clase de registros estándar. Los índices de campo predeterminados están disponibles automáticamente para los siguientes campos:

- `@logStream`

- `@aws.region`
- `@aws.account`
- `@source.log`
- `@data_source_name`
- `@data_source_type`
- `@data_format`
- `traceId`
- `severityText`
- `attributes.session.id`

CloudWatch Los registros también proporcionan índices de campo predeterminados para determinadas combinaciones de nombres y tipos de fuentes de datos. Los índices de campo predeterminados están disponibles automáticamente para las siguientes combinaciones de nombre y tipo de fuente de datos:

Nombre y tipo de fuente de datos	Índices de campo predeterminados

`amazon_vpc.flow`	`action`
	`logStatus`
	`region`
	`flowDirection`
	`type`

`amazon_route53.resolver_query`	`query_type`
	`transport`
	`rcode`

`aws_waf.access`	`action` `httpRequest.country`
`aws_cloudtrail.data`	`eventSource`

Nombre y tipo de fuente de datos	Índices de campo predeterminados

`aws_cloudtrail.management`	`eventName`
	`awsRegion`
	`userAgent`
	`errorCode`
	`eventType`
	`managementEvent`
	`readOnly`
	`eventCategory`
	`requestId`

Los índices de campos predeterminados se suman a cualquier índice de campo personalizado que defina en su política. Los índices de campo predeterminados no se incluyen en la cuota de índices de campo.

filterIndex comparado con el filtro

Para ilustrar la diferencia entre `filterIndex` y `filter`, se pueden tener en cuenta las siguientes consultas de ejemplo. Supongamos que se ha creado un índice de campos para `IPaddress`, para cuatro de los grupos de registros, pero no para un quinto grupo de registros. La siguiente consulta mediante `filterIndex` omitirá la exploración del grupo de registros que no tiene el campo indexado. Para cada grupo de registro indexado, se intenta analizar solo los eventos de registro que tienen el campo indexado y, además, solo devuelve los resultados de una vez creado el índice de campos.

```
fields @timestamp, @message
| filterIndex IPaddress = "198.51.100.0"
| limit 20
```

Por el contrario, si se utiliza `filter` en lugar de `filterIndex` para una consulta de los mismos cinco grupos de registros, la consulta intentará analizar no solo los eventos de registro que contienen

el valor de los grupos de registros indexados, sino que también analizará el quinto grupo de registros que no esté indexado y analizará todos los eventos de registro de ese quinto grupo de registros.

```
fields @timestamp, @message
| filter IPaddress = "198.51.100.0"
| limit 20
```

SOURCE

Incluirlo `SOURCE` en una consulta es una forma útil de especificar las fuentes de and/or datos de los grupos de registros que se van a incluir en una consulta cuando se utiliza la API AWS CLI o para crear una consulta. El `SOURCE` comando solo se admite en la API AWS CLI and, no en la CloudWatch consola. Cuando usa la CloudWatch consola para iniciar una consulta, usa la interfaz de la consola para especificar los grupos de registros.

Consulte los grupos de registros

A fin de utilizar `SOURCE` para especificar los grupos de registros que se van a consultar, se pueden utilizar las siguientes palabras clave:

- `namePrefix` ejecuta la consulta en grupos de registros que tienen nombres que comienzan por la cadena que se especifique. Si se omite, se consultarán todos los grupos de registros.

  Puede incluir hasta cinco prefijos en la lista.
- `accountIdentifier`ejecuta la consulta en los grupos de registros de la AWS cuenta especificada. Esto solo funciona cuando se ejecuta la consulta en una cuenta de supervisión. Si se omite, la opción predeterminada es consultar todas las cuentas de origen vinculadas y la cuenta de supervisión actual. Para obtener más información sobre la observabilidad entre cuentas, consulta la observabilidad CloudWatch entre cuentas.

  Se pueden incluir hasta 20 identificadores de cuenta en la lista.
- `logGroupClass` ejecuta la consulta en grupos de registros que se encuentran en la clase de registro especificada, ya sea de acceso estándar o de acceso poco frecuente. Si se omite esto, se usa la clase de registro predeterminada, que es Estándar. Para obtener más información acerca de las clases de registros, consulte Clases de registro.

Como se puede especificar un gran número de grupos de registros para consultarlos de esta manera, se recomienda que utilice `SOURCE` únicamente en consultas que aprovechen los índices de campos que se hayan creado. Para obtener más información acerca de la indexación de campos

en grupos de registro, consulte Creación de índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de análisis

En el siguiente ejemplo se seleccionan todos los grupos de registro de la cuenta. Si se trata de una cuenta de supervisión, entonces se seleccionarán todos los grupos de registros de la supervisión y todas las cuentas de origen. Si el número total de grupos de registros supera los 10 000, aparecerá un error que pedirá que se reduzca el número de grupos de registros mediante un método de selección de grupos de registros diferente.

```
SOURCE logGroups()
```

En el siguiente ejemplo, se seleccionan los grupos de registros de la cuenta de origen `111122223333`. Si inicias una consulta en una cuenta de supervisión en el marco de la observabilidad CloudWatch multicuenta, los grupos de registros de todas las cuentas de origen y de la cuenta de supervisión se seleccionan de forma predeterminada.

```
SOURCE logGroups(accountIdentifier:['111122223333'])
```

En el siguiente ejemplo, se seleccionan los grupos de registros en función de los prefijos de los nombres.

```
SOURCE logGroups(namePrefix: ['namePrefix1', 'namePrefix2'])
```

En el siguiente ejemplo se seleccionan todos los grupos de registro de la clase de registro de acceso poco frecuente. Si no incluye el identificador `class`, la consulta se aplica únicamente a los grupos de registro en la clase de registro Estándar, que es la predeterminada.

```
SOURCE logGroups(class: ['INFREQUENT_ACCESS'])
```

En el siguiente ejemplo, se seleccionan los grupos de registros de la cuenta 111122223333 que comienzan con prefijos de nombre específicos y pertenecen a la clase de registro Estándar. La clase no se menciona en el comando porque Estándar es el valor predeterminado de la clase de registro.

```
SOURCE logGroups(accountIdentifier:['111122223333'], namePrefix: ['namePrefix1',
 'namePrefix2'])
```

El último ejemplo muestra cómo utilizar el SOURCE comando con el `start-query` AWS CLI comando.

```
aws logs start-query
--region us-east-1
--start-time 1729728200
--end-time 1729728215
--query-string "SOURCE logGroups(namePrefix: ['Query']) | fields @message | limit 5"
```

Consulte las fuentes de datos

`SOURCE`Para especificar las fuentes de datos que se van a consultar, puede utilizar la `dataSource` palabra clave. Puede incluir hasta diez fuentes de datos en la lista.

En el siguiente ejemplo, se selecciona la fuente de `amazon_vpc.flow` datos.

```
SOURCE dataSource(['amazon_vpc.flow'])
```

En el siguiente ejemplo, se selecciona la fuente de `amazon_vpc.flow` datos y se limitan los grupos de registros en función de un prefijo de nombre de grupo de registros.

```
SOURCE dataSource(['amazon_vpc.flow']) logGroups(namePrefix: ['namePrefix1'])
```

Consulte grupos de registros por etiquetas

`SOURCE`Para filtrar grupos de registros por sus etiquetas, utilice la `logGroupTags` función. Especifique las etiquetas como una lista de filtros de etiquetas, cada uno con una `key` `values` matriz opcional.

- Se combinan varios filtros de etiquetas con claves diferentes con la lógica AND.
- Los valores múltiples del mismo filtro de etiquetas se combinan con la lógica OR.
- Se utiliza `*` para hacer coincidir caracteres comodín. Por ejemplo, `payment*` coincide con los valores que comienzan por. `payment`
- Se usa `!` como prefijo para la negación. Por ejemplo, `!production` coincide con valores que no lo son. `production`
- Puede incluir hasta cinco filtros de etiquetas, cada uno con un máximo de cinco valores.

En el siguiente ejemplo, se seleccionan todos los grupos de registros con los que se ha etiquetado`team=team1 OR team=team2`.

```
SOURCE logGroupTags([{"key":"team", "values":["team1", "team2"]}])
```

```
| fields @message, @timestamp
```

En el siguiente ejemplo, se seleccionan los grupos de registros en los que la `service` etiqueta comienza y la `environment` etiqueta no`production`. `payment`

```
SOURCE logGroupTags([{"key":"service", "values":["payment*"]}, {"key":"environment",
 "values":["!production"]}])
| fields @message, @timestamp
```

El siguiente ejemplo combina el filtrado de etiquetas con un filtro de prefijo de nombre.

```
SOURCE logGroups(namePrefix: ['/aws/lambda']) logGroupTags([{"key":"environment",
 "values":["production"]}])
| fields @message, @timestamp
```

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. Se puede utilizar `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 del patrón.

Dado que el comando `pattern` identifica automáticamente los patrones comunes, puede usarlo como punto de partida para buscar y analizar sus 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.

Si CloudWatch Logs no puede deducir el tipo de datos que representa un token dinámico, lo muestra como <Token- *`number`* > e *`number`* indica en qué parte del patrón aparece este token, en comparación con los demás tokens dinámicos.

Entre los ejemplos más comunes de tokens dinámicos se incluyen los códigos de error, las direcciones IP, las marcas de tiempo y los ID de solicitud.

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 tokens. Si CloudWatch Logs puede determinar el tipo de datos que representa un token dinámico, mostrará el token como. <*string*-*number*> *string*Es una descripción del tipo de datos que representa el token. *number*Muestra en qué parte del patrón aparece este token, en comparación con los otros tokens dinámicos.

  CloudWatch Los registros asignan a la cadena una parte del nombre en función del análisis del contenido de los eventos del registro que lo contienen.

  Si CloudWatch Logs no puede deducir el tipo de datos que representa un token dinámico, lo muestra como <Token- *number* > e *number* indica en qué parte del patrón aparece este token, en comparación con los demás tokens dinámicos.

  Por ejemplo, `[INFO] Request time: <Time-1> 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

Permite comparar los eventos de registro que se encuentren 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.

Agregue un modificador al comando `diff` para especificar el período con el que desee comparar la información:

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

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

parse

Se utiliza `parse` para extraer datos de un campo de registro y crear campos extraídos que pueda procesar en la consulta. El `parse` comando admite tres modos: expresiones globales, expresiones regulares y logfmt.

Si `fieldName` se omite, se `@message` utiliza de forma predeterminada. Puede analizar cualquier campo con nombre especificando el nombre del campo como primer argumento.

Si un evento de registro no coincide con el patrón especificado, se seguirá viendo en los resultados, pero sin los campos extraídos.

Modo global

Utilice caracteres comodín (*) como marcadores de posición para los valores que desee extraer y asígnelos a los campos con nombre asignado. `as`

Sintaxis

```
parse fieldName "pattern" as alias1, alias2
```

El número de * caracteres comodín debe ser igual al número de alias.

Ejemplos

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

```
parse @logStream "*/*/*/*" as env, service, instance, shard
| stats count(*) by env, service
```

Análisis encadenado

Extraiga un campo y, a continuación, analice aún más el campo extraído.

```
parse @message "url=*" as url
| parse url "/api/*/users/*" as apiVersion, userId
| display apiVersion, userId
```

Modo regex

Utilice una expresión regular con grupos de captura con nombre para extraer campos. Para obtener información acerca de la sintaxis de la expresión regular, consulte Sintaxis de expresiones regulares (regex) compatibles.

Sintaxis

```
parse fieldName /regex/
```

Utilice grupos de captura con nombre (`?<name>...`) para definir los campos extraídos.

Ejemplos

Utilice grupos de captura con nombre para extraer campos

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

Utilice un grupo de captura con nombre para extraer el ENI de un registro de flujo de VPC

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

Modo Logfmt

Se utiliza `parse logfmt` para analizar las líneas de registro con formato logfmt en pares clave-valor. Logfmt es un formato de registro estructurado en el que cada línea contiene pares separados por espacios. `key=value`

Sintaxis

```
parse fieldName logfmt as alias
```

El resultado es un mapa al que se accede con notación de puntos (por ejemplo,,). `lf.level` `lf.msg`

Ejemplos

```
parse @message logfmt as lf
| filter lf.level = "error"
| display lf.msg, lf.duration
```

```
parse @message logfmt as lf
| stats count(*) by lf.host
```

campos relevantes

Se utiliza `relevantfields` para identificar automáticamente qué campos de los datos de registro son más relevantes para una condición determinada. El comando compara los registros de referencia con el subconjunto que coincide con su condición y devuelve los campos clasificados según su puntuación de relevancia.

Sintaxis

```
relevantfields [field1, field2, ...] where condition
```

La lista de campos es opcional. Si se omite, se analizan todos los campos.

El comando devuelve los siguientes campos de salida:

- `@fieldName`— Nombre del campo
- `@relevanceScore`— Una puntuación que indica la relevancia en una escala de 0 a 1
- `@topRelevanceContributors`— En el caso de los campos categóricos, muestra las entradas con los cambios de frecuencia más altos.
- `@conditionalMedian`— En el caso de los campos numéricos, la mediana de los valores de los registros que cumplen la condición
- `@baselineMedian`— En el caso de los campos numéricos, la mediana de los valores de los registros que no cumplen la condición

Ejemplos

Analice todos los campos para detectar respuestas lentas de la API

```
relevantfields where Time > 400
```

Centra el análisis en campos específicos

```
relevantfields Controller, Path, Service where Time > 400
```

Identifique los impulsores de los tiempos de espera de Lambda

```
relevantfields where Duration > 5000
```

expandir

Se utiliza `expand` para tomar un campo que contiene una matriz JSON y crear un evento de registro independiente para cada elemento de la matriz. Todos los demás campos del evento de registro original se duplican en cada evento nuevo.

Sintaxis

```
expand fieldName
```

Ejemplo

Si un evento de registro contiene `items = ["apple","banana","cherry"]` y`host = "web-01"`, a continuación, `expand items` produce tres eventos de registro: `{items: "apple", host: "web-01"}``{items: "banana", host: "web-01"}`, y`{items: "cherry", host: "web-01"}`.

```
expand items
| stats count(*) by items, host
```

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, la información siguiente es el resultado de una clasificación 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 clasificación se invertirán.

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

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 varios `stats` comandos en una sola consulta. Esto le permite realizar agregaciones adicionales en el resultado de la primera agregación. El número máximo de `stats` comandos permitido en una consulta depende de la clase de registro del grupo de registros.

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 varios `stats` comandos con otros comandos, por ejemplo, `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

El número máximo de `stats` comandos de una consulta depende de la clase de registro:

- Clase de registro estándar: máximo de 10 `stats` comandos por consulta
- Clase de registro de acceso poco frecuente: máximo de 2 `stats` comandos por consulta

Estas cuotas no se pueden cambiar.

Si utiliza un `limit` comando `sort` o, debe aparecer después del último `stats` comando. Si es anterior al último `stats` comando, la consulta no es válida.

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

En `stats` los comandos siguientes de una sola consulta, solo puede hacer referencia a los campos definidos en el `stats` comando anterior. 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 de un `stats` comando deben definirse en ese `stats` comando.

```
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 un `stats` comando posterior sin usar el `stats` comando anterior para propagar el `timestamp` campo. 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 `@timestamp` campo en el `stats` comando anterior y, a continuación, podrá utilizarlo con él `dateceil` en un `stats` comando posterior, 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	Description (Descripción)
`avg(fieldName: NumericLogField)`	número	La media de los valores en el campo especific ado.

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

`count_distinct(fie ldName: LogField)`	número	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)`	LogFieldV alue	El máximo de los valores para este campo de registro en los registros consultados.
`min(fieldName: LogField)`	LogFieldV alue	El mínimo de los valores para este campo de registro en los registros consultados.
`pct(fieldName: LogFieldValue, percent: number)`	LogFieldV alue	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.
`stddev(fieldName: NumericLogField)`	número	El desvío estándar de los valores en el campo especificado.
`sum(fieldName: NumericLogField)`	número	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	Description (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(fieldName: 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. Si lo omite`limit`, la consulta devolverá de forma predeterminada hasta 10 000 eventos de registro. Puede especificar un `limit` valor de hasta 100.000.

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`.

Cuando se utiliza `dedup` en una consulta, la consola muestra un mensaje como Mostrar X de Y registros, donde X es el número de resultados deduplicados e Y es el número total de registros coincidentes antes de la deduplicación. Esto indica que se eliminaron los registros duplicados y no significa que falten datos.

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.

unnest

Se debe utilizar `unnest` para aplanar una lista tomada como entrada y generar varios registros con un único registro para cada elemento de la lista. En función de la cantidad de elementos que

contenga un campo, este comando descarta el registro actual y genera nuevos registros. Cada registro incluye el `unnested_field`, que representa un elemento. Todos los demás campos provienen del registro original.

La entrada para `unnest` es `LIST`, que proviene de la función `jsonParse`. Para obtener más información, consulte Tipos de estructura. Cualquier otro tipo, como `MAP`, `String` y `numbers`, se trata como una lista con un elemento en `unnest`.

Estructura de comandos

En el ejemplo siguiente, se describe el formato de este comando.

```
unnest field into unnested_field
```

Consulta de ejemplo

En el siguiente ejemplo, se analiza una cadena de objeto JSON y se amplía una lista de eventos de campo.

```
fields jsonParse(@message) as json_message
| unnest json_message.events into event
| display event.name
```

El evento de registro de esta consulta de ejemplo podría ser una cadena JSON de la siguiente manera:

```
{
   "events": [
        {
            "name": "exception"
        },
        {
            "name": "user action"
        }
   ]
}
```

En este caso, la consulta de ejemplo genera dos registros en el resultado de la consulta, uno con `event.name` como `exception` y otro con `event.name` como acción del usuario

Consulta de ejemplo

En el siguiente ejemplo, se aplana una lista y, a continuación, se filtran los elementos.

```
fields jsonParse(@message) as js
| unnest js.accounts into account
| filter account.type = "internal"
```

Consulta de ejemplo

En el siguiente ejemplo, se aplana una lista para su agregación.

```
fields jsonParse(trimmedData) as accounts
| unnest accounts into account
| stats sum(account.droppedSpans) as n by account.accountId
| sort n desc
| limit 10
```

lookup

Se utiliza `lookup` para enriquecer los resultados de la consulta con datos de referencia de una tabla de consulta. Una tabla de consulta contiene datos CSV que subes a Amazon CloudWatch Logs. Cuando se ejecuta una consulta, el `lookup` comando compara un campo del registro de eventos con un campo de la tabla de consulta y agrega los campos de salida especificados a los resultados.

Utilice tablas de consulta para situaciones de enriquecimiento de datos, como asignar los identificadores de usuario a los detalles del usuario, los códigos de producto a la información del producto o los códigos de error a las descripciones de los errores.

Creación y administración de tablas de consulta

Antes de poder utilizar el `lookup` comando en una consulta, debe crear una tabla de consulta. Puede crear y gestionar tablas de búsqueda desde la CloudWatch consola o mediante la API de Amazon CloudWatch Logs.

Para crear una tabla de consulta (consola)

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Configuración y, a continuación, elija la pestaña Registros.
3. Desplázate hasta Tablas de búsqueda y selecciona Administrar.
4. Seleccione Crear tabla de consulta.

5. Introduzca un nombre para la tabla de consulta. El nombre solo puede contener caracteres alfanuméricos, guiones y guiones bajos.
6. (Opcional) Introduzca una descripción.
7. Cargue un archivo CSV. El archivo debe incluir una fila de encabezado con los nombres de las columnas, debe estar UTF-8 codificado y no debe superar los 10 MB.
8. (Opcional) Especifique una AWS KMS clave para cifrar los datos de la tabla.
9. Seleccione Crear.

Después de crear una tabla de consulta, puede verla en el editor de consultas de CloudWatch Logs Insights. Seleccione la pestaña Tablas de búsqueda para buscar las tablas disponibles y sus campos.

Para actualizar una tabla de consulta, selecciónela y elija Acciones, Actualizar. Cargue un nuevo archivo CSV para reemplazar todo el contenido existente. Para eliminar una tabla de consulta, selecciona Acciones, Eliminar.

> **Note**
>
> Puede crear hasta 100 tablas de búsqueda por cuenta y por cuenta Región de AWS. Los archivos CSV pueden tener un tamaño máximo de 10 MB. También puedes gestionar las tablas de búsqueda mediante la API de Amazon CloudWatch Logs. Para obtener más información, consulta CreateLookupTablela referencia de la API CloudWatch de Amazon Logs.

> **Note**
>
> Si la tabla de consulta está cifrada con una clave de KMS, la persona que llama debe tener el `kms:Decrypt` permiso sobre la clave (la clave de KMS utilizada para cifrar la tabla de consulta) para utilizar la `StartQuery` API con una consulta que haga referencia a esa tabla de consulta. Para obtener más información, consulte Cifre las tablas de búsqueda en CloudWatch los registros mediante AWS Key Management Service.

Sintaxis de consulta para la búsqueda

Estructura de comandos

A continuación se muestra el formato de este comando.

```
lookup table match-fields output-mode output-field[,...]
```

El comando utiliza los siguientes argumentos:

- *table*— El nombre de la tabla de consulta que se va a utilizar.
- *match-fields*— Especifique uno o más campos para hacer coincidir los eventos del registro con la tabla de consulta. Puede utilizar cualquiera de los siguientes formularios:
  - *lookup-field* `as` *log-field* `[,...]`— Se utiliza `as` cuando el nombre de la columna de la tabla de consulta es diferente del nombre del campo de eventos del registro. Por ejemplo, `ip_address as srcAddr` hace coincidir la `ip_address` columna de la tabla de consulta con el `srcAddr` campo del registro de eventos.
  - *lookup-field* `[,...]`— Si el nombre del campo de eventos de registro es el mismo que el nombre de la columna de la tabla de consulta, puede omitir `as` y especificar el nombre del campo directamente. Por ejemplo, hace `department, role` coincidir ambas columnas con los campos de eventos de registro con los mismos nombres.

  Cuando se especifican varios campos coincidentes, una fila de la tabla de consulta debe coincidir con todos los campos para obtener un resultado (lógica AND).
- *output-mode*— Especifica cómo se añaden los campos de salida a los resultados. Utilice una de las siguientes:
  - `OUTPUT`— Añade los campos de salida a los resultados. Si ya existe un campo con el mismo nombre en el registro de eventos, se sobrescribe con el valor de la tabla de consulta. Si no se encuentra ninguna coincidencia, el campo se establece en nulo.
  - `OUTPUTNEW`— Añade los campos de salida a los resultados solo si el campo aún no existe en el registro de eventos. Si el campo ya tiene un valor, se mantiene el valor original. Si no se encuentra ninguna coincidencia, el campo permanece sin cambios.
- *output-field*— Uno o más campos de la tabla de búsqueda para añadirlos a los resultados.

Ejemplo: enriquece los eventos del registro con los detalles del usuario

Supongamos que tiene un grupo de registros con eventos que contienen un `id` campo y una tabla de consulta `user_data` con las columnas `idname`,`email`, y`department`. La siguiente consulta enriquece cada evento de registro con el nombre, el correo electrónico y el departamento del usuario de la tabla de búsqueda.

```
fields action, status, name, email, department
| lookup user_data id OUTPUT name, email, department
```

Ejemplo: utilice la búsqueda con agregación

Puede utilizar los campos de salida de la búsqueda con funciones de agregación. La siguiente consulta enriquece los eventos del registro con los detalles del usuario y, a continuación, cuenta los eventos agrupados por dirección de correo electrónico.

```
fields user_id, action, username, email, department
| lookup user_data user_id OUTPUT username, email, department
| stats count(*) by email
```

Ejemplo: utilice la búsqueda con un filtro

Puede filtrar los resultados en función de los campos devueltos por la búsqueda. La siguiente consulta enriquece el registro de eventos y, a continuación, filtra para mostrar solo los eventos de un departamento específico.

```
fields user_id, action
| lookup user_data user_id OUTPUT username, email, department
| filter department = "Engineering"
```

Ejemplo: utilice OUTPUTNEW para enriquecer sin sobrescribir

Si sus eventos de registro ya contienen un `hostname` campo, pero a veces está vacío, utilícelo `OUTPUTNEW` para rellenar los valores faltantes sin sobrescribir los existentes.

```
fields srcAddr, hostname
| lookup known_hosts ip_address as srcAddr OUTPUTNEW hostname, region
```

Ejemplo: utilice la búsqueda con varios campos coincidentes

Puede hacer coincidir en más de un campo. La siguiente consulta coincide con la tabla de `dstPort` búsqueda `srcAddr` y con ella para identificar los servicios de red conocidos.

```
fields @timestamp, srcAddr, dstAddr, dstPort
| lookup network_services ip_address as srcAddr, port as dstPort OUTPUT service_name,
 owner
```

```
| filter ispresent(service_name)
```

Ejemplo: utilice la búsqueda con nombres de campo coincidentes

Si los nombres de los campos de eventos de registro coinciden exactamente con los nombres de las columnas de la tabla de búsqueda, puede omitir la `as` palabra clave. La siguiente consulta hace coincidir ambos `department` `role` campos directamente con los de la tabla de consulta.

```
fields @timestamp, department, role
| lookup employees department, role OUTPUT office, manager
```

unirse

Combina los eventos de registro de un grupo de registros de origen con los eventos de otro grupo de registros o el resultado de una consulta en función de un campo coincidente.

Utilice el `join` comando para correlacionar los eventos de registro relacionados en distintas fuentes, como los grupos de registros, utilizando claves comunes a todos ellos, como los identificadores de solicitud o los ID de transacción coincidentes.

Sintaxis

```
join type=<join_type> left=<left_alias> right=<right_alias>
    where <left_alias>.<field>=<right_alias>.<field>
    (SOURCE <right_log_group>)
```

Parameters

- `<right_log_group>`— La fuente de datos secundaria a la que unirse.
- `<left_alias>`y `<right_alias>` — Alias para distinguir los campos de las fuentes de datos izquierda (primaria) y derecha (secundaria).
- `where <field>`— Especifica el campo utilizado como clave de combinación. El campo debe existir en ambas fuentes de datos.
- `type=<join_type>`(opcional): especifica el tipo de unión. Los valores válidos son los siguientes:
  - `inner`(predeterminado): devuelve solo los registros coincidentes
  - `left`— Devuelve todos los registros de la fuente de datos principal y los registros coincidentes de la fuente de datos secundaria

Ejemplos

Example Ejemplo 1: correlacionar las solicitudes de API Gateway con los registros de ejecución de Lambda

En este ejemplo, se muestra cómo unir los registros de acceso de API Gateway con los registros de funciones de Lambda para correlacionar las solicitudes entrantes con su procesamiento de backend. Esto resulta útil para solucionar problemas de flujos de solicitudes de extremo a extremo e identificar qué invocaciones de Lambda corresponden a solicitudes de API específicas.

```
filter status >= 500
| join type=inner left=api right=lambda
    where api.requestId=lambda.requestId
    (SOURCE '/aws/lambda/my-function')
| fields api.requestId, api.status, api.latency, lambda.duration, lambda.memoryUsed
| sort api.latency desc
```

Esta consulta:

1. Consulta los registros de acceso y los filtros de API Gateway para detectar errores en el servidor (estado >= 500)
2. Se une a los registros de funciones Lambda mediante el `requestId` campo que aparece en ambas fuentes de registro
3. Utiliza alias (`apiylambda`) para distinguir los campos de cada fuente
4. Devuelve información combinada que muestra la latencia de la API junto con la duración de la ejecución de Lambda y el uso de memoria
5. Ordena los resultados por latencia de la API para identificar las solicitudes más lentas

Example Ejemplo 2: Realice un seguimiento de las transacciones distribuidas en los microservicios

Al depurar problemas en una arquitectura de microservicios, a menudo es necesario rastrear una transacción en varios servicios. En este ejemplo, se muestra cómo unir registros de dos servicios diferentes mediante un identificador de transacción común.

```
filter eventType = "ORDER_CREATED"
| join type=left left=order right=payment
    where order.transactionId=payment.transactionId
```

```
    (SOURCE '/aws/lambda/payment-service')
| filter payment.eventType = "PAYMENT_PROCESSED" or !ispresent(payment.eventType)
| fields order.transactionId, order.orderId, order.customerId,
    payment.paymentStatus, payment.amount
| filter payment.paymentStatus != "SUCCESS" or !ispresent(payment.paymentStatus)
```

Esta consulta:

1. Comienza con los eventos de creación de pedidos desde el servicio de pedidos
2. Utiliza `left join` a para incluir todos los pedidos, incluso los que no tienen registros de pago coincidentes
3. Se une a los eventos de procesamiento de pagos mediante el `transactionId` campo compartido
4. Filtra los resultados finales para mostrar solo los pedidos con pagos faltantes o con registros de pago faltantes

La combinación de la izquierda es importante en este caso porque permite ver los pedidos que se crearon pero que nunca tuvieron el evento de pago correspondiente, lo que podría indicar un fallo del sistema.

Comportamiento

- Primero se procesa la fuente de datos principal (lado izquierdo).
- La fuente de datos secundaria se evalúa y compara mediante la clave de unión especificada.
- La coincidencia se realiza mediante la comparación de igualdad en el campo de unión.
- En el caso de las uniones por la izquierda, los registros no coincidentes de la fuente de datos principal se conservan con valores nulos para los campos secundarios.

Notas y limitaciones

- Solo se admiten las condiciones de igualdad (=).
- Solo se admite un comando de unión por consulta.
- Las claves de unión deben existir en ambas fuentes de datos y ser de tipos compatibles.
- Las consultas que utilizan la combinación pueden escanear más datos e incurrir en costes más altos.

- El número de valores clave únicos en la fuente de datos secundaria está limitado a 50 000 para garantizar el rendimiento de las consultas.
- No se admiten las subconsultas del lado derecho de la unión.

Comandos relacionados

- campos
- filtro
- analizar
- stats
- ordenar
- limit

subqueries

Una subconsulta es una consulta anidada de Logs Insights que se puede utilizar como entrada para otra consulta. Las subconsultas se pueden usar para derivar conjuntos de resultados intermedios que luego son consumidos por los comandos posteriores.

Sintaxis

Subconsulta en el filtro

```
filter <field> in (
    <subquery>
)
```

Parameters

- `<subquery>`— Una consulta de Logs Insights válida que devuelve un conjunto de resultados. La subconsulta debe generar campos a los que haga referencia la consulta externa.

Ejemplos

Example Ejemplo 1: busque solicitudes que hayan encontrado errores en los servicios descendentes

En este ejemplo, se muestra cómo usar una subconsulta para identificar las solicitudes en el servicio principal que provocaron errores en un servicio descendente. Esto resulta útil para solucionar errores en cascada en sistemas distribuidos.

```
filter requestId in (
    SOURCE '/aws/lambda/database-service'
    | filter errorType = "DatabaseConnectionTimeout"
    | fields requestId
)
| fields @timestamp, requestId, endpoint, userId, responseTime
| sort @timestamp desc
```

Esta consulta:

1. La subconsulta busca todos los `requestId` valores del servicio de base de datos que ha sufrido tiempos de espera de conexión
2. La consulta externa filtra los registros del servicio principal para mostrar solo las solicitudes que coinciden con los identificadores de solicitud propensos a errores
3. Los resultados muestran el contexto completo de las solicitudes que fallaron en fases posteriores, incluidos los puntos finales y los usuarios afectados

Este patrón le ayuda a comprender el impacto inicial de los errores descendentes.

Example Ejemplo 2: Identifique las solicitudes que suelen fallar para realizar una investigación específica

En este ejemplo, se muestra el uso de una subconsulta con agregación para buscar solicitudes que fallan repetidamente, lo que suele indicar problemas sistemáticos en lugar de errores transitorios.

```
filter requestId in (
    SOURCE '/aws/lambda/payment-processor'
    | filter status = "FAILED"
    | stats count(*) as failureCount by requestId
    | filter failureCount > 3
    | fields requestId
)
| fields @timestamp, requestId, customerId, amount, failureReason
| sort @timestamp asc
```

Esta consulta:

1. La subconsulta agrega los intentos de pago fallidos e identifica los ID de solicitud que fallaron más de 3 veces
2. La consulta externa recupera todos los eventos de registro de esos ID de solicitud problemáticos
3. Los resultados se ordenan cronológicamente para mostrar la progresión de los reintentos

Esto ayuda a distinguir entre los errores transitorios (un solo incidente) y los problemas persistentes (varios errores) que requieren una investigación más profunda.

Comportamiento

- Las subconsultas se ejecutan independientemente de la consulta externa.
- Los resultados se materializan antes de que la consulta externa los consuma.
- Solo los campos seleccionados explícitamente en la subconsulta están disponibles para la consulta externa.

Notas y limitaciones

- Las subconsultas deben devolver los campos a los que hace referencia la consulta externa.
- No se admiten las subconsultas anidadas.
- Las subconsultas pueden aumentar el tiempo y el coste de ejecución de las consultas.
- No se admiten las subconsultas correlacionadas.
- La ejecución de consultas internas está limitada a 30 segundos.

Comandos relacionados

- campos
- filtro
- analizar
- stats
- ordenar
- limit

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	Description (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	Description (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	Description (Descripción)
`abs(a: number)`	número	Valor absoluto

Operación	Tipo de resultado	Description (Descripción)
`ceil(a: number)`	número	Redondeo a valor máximo (menor número entero que es mayor que el valor de a).
`floor(a: number)`	número	Redondeo a valor mínimo (mayor número entero que es menor que el valor de a).
`greatest(a: number, ...numbers: number[])`	número	Devuelve el valor más alto
`least(a: number, ...numbers: number[])`	número	Devuelve el valor más bajo
`log(a: number)`	número	Registro natural
`round(a: number [, d: number])`	número	Redondea el valor dea. Con un argumento, redondea al entero más cercano. Con dos argumentos, redondea a d decimales.
`sqrt(a: number)`	número	Raíz cuadrada
`haversine(lat1: number, lon1: number, lat2: number, lon2: number)`	número	Calcula la distancia del gran círculo en kilómetros entre dos puntos geográficos especific ados mediante la latitud y la longitud en grados.

Tipos de estructura

Un mapa o una lista es un tipo de estructura de CloudWatch Logs Insights que permite acceder a los atributos de las consultas y utilizarlos.

Ejemplo: obtención de un mapa o una lista

Se usa `jsonParse` para analizar un campo que es una cadena json y convertirlo en un mapa o una lista.

```
fields jsonParse(@message) as json_message
```

Ejemplo: acceso a los atributos

Utilice el operador de acceso por puntos (map.attribute) para acceder a los elementos de un mapa. Si un atributo de un mapa contiene caracteres especiales, utilice acentos graves para incluir el nombre del atributo (map.attributes.`special.char`).

```
fields jsonParse(@message) as json_message
| stats count() by json_message.status_code
```

Utilice el operador de acceso entre corchetes (list[index]) para recuperar un elemento en una posición específica de la lista.

```
fields jsonParse(@message) as json_message
| filter json_message.users[1].action = "PutData"
```

Coloque los caracteres especiales entre acentos graves (``) cuando haya caracteres especiales en el nombre de la clave.

```
fields jsonParse(@message) as json_message
| filter json_message.`user.id` = "123"
```

Ejemplo: resultados vacíos

Los mapas y las listas se consideran nulos para las funciones de cadena, número, y fecha y hora.

```
fields jsonParse(@message) as json_message
| display toupper(json_message)
```

La comparación del mapa y la lista con cualquier otro campo da como resultado `false`.

> **Note**
>
> No se admite el uso de mapas y listas en `dedup`, `pattern`,`sort` y `stats`.

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 tengan un número y uno de los siguientes valores:

- `ms` para milisegundos
- `s` para segundos
- `m` para minutos
- `h` para 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 la solicitud en función de la unidad de tiempo que elija. Por ejemplo, pone un límite de 60 como valor máximo para cualquier solicitud que utilice `s`. 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 en un minuto, por lo que CloudWatch Logs no utilizará un número superior a `60s`. Para crear un bucket de 5 minutos, utilice `bin(5m)` en su lugar.
> El límite de `ms` es 1000, el límite de `s` y `m` es 60 y el límite de `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	Description (Descripción)

`bin(period: Period)`	Timestamp	Redondea el valor de `@timestamp` según el periodo de tiempo indicado y, a continuación, 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 excepciones por hora:
		
		`filter @message like /Exception/`
		`    | stats count(*) as exceptionCount`
		`by bin(1h)`
		`    | sort exceptionCount desc`
		
		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`

Función	Tipo de resultado	Description (Descripción)
`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.
`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)`	número	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 `2022-01-14T13:18:0 31.000-08:00` a `1642195111000` .

Función	Tipo de resultado	Description (Descripción)

now( )	número	Devuelve la hora en que se inició el procesamiento de la consulta, en segundos por época. Esta función no toma argumentos.
		
		Puede usarlo para filtrar los resultados de la consulta según la hora actual.
		
		Por ejemplo, la siguiente consulta devuelve todos los errores 4xx de las últimas dos horas:
		
		`parse @message "Status Code: *;" as statusCode\n`
		`| filter statusCode >= 400 and statusCode <= 499 \n`
		`| filter toMillis(@timestamp) >= (now() * 1000 - 7200000)`
		
		El ejemplo siguiente devuelve todas las entradas de registro de las últimas cinco horas que contienen la palabra `error` o `failure`
		
		`fields @timestamp, @message`
		`| filter @message like /(?i)(error|failure)/`
		`| filter toMillis(@timestamp) >= (now() * 1000 - 18000000)`

> **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	Description (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
`case(cond1, val1, cond2, val2, ..., [default])`	LogField	Evalúa las condiciones en orden y devuelve el valor de la primera condición verdadera . Si no se cumple ninguna condición y se proporciona un valor predeterminado, devuelve el valor predeterminado. Admite hasta 10 sucursales.

Funciones JSON

Funciones JSON

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

Función	Tipo de resultado	Description (Descripción)
`jsonParse(fieldName: string)`	Mapa | Lista | Vacío	Devuelve un mapa o una lista cuando la entrada es una

Función	Tipo de resultado	Description (Descripción)
		representación en cadena de un objeto JSON o una matriz JSON. Devuelve un valor vacío si la entrada no es una de las representaciones.
`jsonStringify(fieldName: Map | List)`	Cadena	Devuelve una cadena JSON de un mapa o de una lista de datos.

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	Description (Descripción)
`isValidIp(fieldName: string)`	booleano	Devuelve `true` si el campo es una dirección IPv4 o IPv6 válida.
`isValidIpV4(fieldName: string)`	booleano	Devuelve `true` si el campo es una dirección IPv4 válida.
`isValidIpV6(fieldName: string)`	booleano	Devuelve `true` si el campo es una dirección IPv6 válida.
`isIpInSubnet(fieldName: string, subnet: string)`	booleano	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.

Función	Tipo de resultado	Description (Descripción)
`isIpv4InSubnet(fieldName: string, subnet: string)`	booleano	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)`	booleano	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	Description (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.

Función	Tipo de resultado	Description (Descripción)

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

Función	Tipo de resultado	Description (Descripción)

`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("xyZ xyfooxyxyZ","xyZ")` devuelve `"foo"`.

`strlen(str: string)`	número	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.

`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"`.

Función	Tipo de resultado	Description (Descripción)
`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`.
`regex_replace(fieldName: string, pattern: string, replacement: string)`	cadena	Sustituye todas las subcadenas que coincidan `pattern` con `replacement` la expresión regular por. Utiliza la sintaxis de expresiones regulares RE2.
`strcontains(str: string, searchValue: string)`	número	Devuelve 1 si `str` contiene `searchValue` y 0 en los demás casos.
`startsWith(str: string, searchValue: string)`	número	Devuelve 1 si `str` empieza por `searchValue` y 0 en caso contrario.
`endsWith(str: string, searchValue: string)`	número	Devuelve 1 si `str` termina en `searchValue` y 0 en caso contrario.
`urlencode(str: string)`	cadena	URL-encodes la cadena.

Función	Tipo de resultado	Description (Descripción)
`urldecode(str: string)`	cadena	URL-decodes la cuerda.
`base64encode(str: string)`	cadena	Base64-encodes la cuerda.
`base64decode(str: string)`	cadena	Base64-decodes la cuerda.

Campos que contienen caracteres especiales

Si un campo contiene caracteres no alfanuméricos distintos al símbolo `@` o al punto (`.`), debe rodearlo con caracteres de comillas invertidas (`` ` ``). 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 registro 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
```

Introducción al lenguaje de consulta de Información de registros: tutoriales de consultas

En las secciones siguientes se incluyen tutoriales de consultas de ejemplo para ayudarlo a familiarizarse con Logs Insights QL.

Temas

- Tutorial: ejecutar y modificar una consulta de muestra
- Tutorial: ejecutar una consulta con una función de agregación

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

Tutorial: ejecutar y modificar una consulta de muestra

El siguiente tutorial le ayuda a empezar a utilizar CloudWatch Logs Insights. Se ejecute una consulta de muestra en Logs Insights QL 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 Registros y, luego, Información de registros.

   En la página Información de registros, el editor de consultas contiene una consulta predeterminada en Logs Insights QL que devuelve los 20 eventos de registro más recientes.
3. En el menú desplegable Select log group(s) (Seleccionar grupos de registro), elija uno o varios grupos de registro que va a consultar.

   Si se trata de una cuenta de monitorización 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 detectados solo son compatibles con los grupos de registro de la clase de registro Estándar. Para obtener más información acerca de las clases de registros, consulte Clases 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 registro 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 registro 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 CloudWatch las consultas de 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. Seleccione 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 (“”).

Logs Insights QL incluye potentes capacidades de consulta, incluidos varios comandos y compatibilidad con expresiones regulares, operaciones matemáticas y operaciones estadísticas. Para obtener más información, consulte CloudWatch Registra la sintaxis de consulta del lenguaje 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 Registros y, luego, Información de registros.
3. Confirme que esté seleccionada la pestaña Logs Insights QL.
4. En el menú desplegable Select log group(s) (Seleccionar grupos de registro), elija uno o varios grupos de registro que va a consultar.

   Si se trata de una cuenta de monitorización 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.
5. Elimine la consulta predeterminada en el editor de consultas e ingrese el siguiente comando:

   ```
   stats count(*) by fieldName
   ```
6. *fieldName*Sustitúyalos por un campo descubierto del menú 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.
7. Elija Run (Ejecutar) para ver los resultados de la consulta.

   Los resultados de la consulta muestran el número de registros del grupo de registro 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 registro), elija uno o varios grupos de registro que va a consultar.

   Si se trata de una cuenta de monitorización 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 registro 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 registro.

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 Registros y, luego, Información de registros.

3. Confirme que esté seleccionada la pestaña Logs Insights QL.
4. En el menú desplegable Select log group(s) (Seleccionar grupos de registro), elija uno o varios grupos de registro que va a consultar.

   Si se trata de una cuenta de monitorización 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.
5. 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.
6. 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.

Consultas de ejemplo

Esta sección contiene una lista de comandos de consulta generales y útiles que puede ejecutar en la CloudWatchconsola. 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 la consulta, consulte CloudWatch Registra la sintaxis de consulta del lenguaje Insights.

Temas

- Consultas generales
- Consultas de registros de Lambda

- Consultas de registros de flujo de Amazon VPC
- Consultas de registros de Route 53
- Consultas de CloudTrail registros
- Consultas para Amazon API Gateway
- Consultas para la puerta de enlace NAT
- Consultas para registros del servidor Apache
- Consultas para Amazon EventBridge
- Ejemplos del comando para analizar

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

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

```
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
```

Devolver la lista de las rutas de API más populares de su grupo de registro de acceso de Amazon API Gateway

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

Crear un informe de latencia de integración para su grupo de registro de acceso de Amazon API Gateway

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

Consultas para la puerta de enlace NAT

Si observa costos más altos de lo normal en su AWS factura, puede 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
```

Comparación (diferencia) con intervalos de tiempo anteriores

Puede usar CloudWatch Logs Insights con Logs Insights QL para comparar los cambios en sus eventos de registro a lo largo del tiempo. Puede comparar los eventos de registro incorporados durante un intervalo de tiempo reciente con los registros del período inmediatamente anterior. Como alternativa, puede compararlos con períodos anteriores similares. Esto puede ayudarle a determinar si un error en sus registros se introdujo recientemente o si ya se estaba produciendo; también puede ayudarle a buscar otras tendencias.

Las consultas de comparación solo muestran patrones en los resultados y no eventos de registro sin procesar. Los patrones devueltos le ayudarán a detectar 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 el uso de patrones de registro, consulte Análisis del patrón.

Cuando ejecute una consulta de comparación, esta se analiza en función de dos períodos de tiempo diferentes: el período de consulta original que seleccionó 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 de las comparaciones son los siguientes.

- Período anterior: se compara con el período inmediatamente anterior al período de consulta.
- Período anterior: se compara con el período de un día anterior al período de consulta.
- Período anterior: se compara con el período de una semana anterior al período de consulta.
- Período 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, consulte Precios de Amazon CloudWatch.

Ejecución de una consulta de comparación

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.

   Aparece una consulta predeterminada en el cuadro de consulta.
3. Confirme que esté seleccionada la pestaña Logs Insights QL.
4. Mantenga la consulta predeterminada o introduzca una consulta diferente.
5. En el menú desplegable Seleccionar grupos de registro, elija uno o varios grupos de registro que va a consultar.
6. (Opcional) Utilice el selector de tiempo para seleccionar el periodo de tiempo que desea consultar. La consulta predeterminada corresponde a la hora anterior del registro de datos.
7. En el selector de intervalos de tiempo, seleccione Comparar. A continuación, elija el período de tiempo anterior con el que desee comparar los registros originales y seleccione Aplicar.
8. Elija Ejecutar consulta.

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

   La tabla muestra la siguiente información:

- Cada Patrón, con partes variables del patrón sustituidas por el símbolo dinámico del token <*string*-*number*>. *string*Es una descripción del tipo de datos que representa el token. *number*Muestra en qué parte del patrón aparece este token, en comparación con los otros tokens dinámicos. Para obtener más información, consulte Análisis del patrón.
- Recuento de eventos es el número de eventos de registro que tienen ese patrón en el período de tiempo original, más el período actual.
- Diferencia en el recuento de eventos 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.
- Descripción de la diferencia resume brevemente el cambio en ese patrón entre el período de tiempo actual y el período de comparación.
- Tipo de gravedad es la gravedad probable de los eventos de registro con este patrón, basada en las palabras que se encontraron en el registro de eventos, como `FATAL`, `ERROR` y `WARN`.

10. Para seguir analizando alguno de los patrones de la lista, elija el icono de la columna Inspeccionar correspondiente a uno de los patrones.

    Aparecerá el panel 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 veces que aparece el 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 Muestras de registro 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 es que seleccionó uno.

    > **Note**
    >
    > Se captura un máximo de 10 valores de token por 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 está analizando. Por ejemplo, si el patrón

de un mensaje de `ERROR` solía ir acompañado de otro evento de registro marcado como `INFO` con detalles adicionales, ese patrón se muestra aquí.

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.

OpenSearch Lenguaje de procesamiento canalizado (PPL)

Esta sección contiene una introducción básica a la consulta de CloudWatch registros mediante PPL. OpenSearch Con el PPL se pueden recuperar, consultar y analizar datos mediante comandos que están canalizados juntos, lo que facilita la comprensión y la redacción de consultas complejas. Su sintaxis se basa en las canalizaciones de Unix y permite encadenar comandos para transformar y procesar datos. Con el PPL, se pueden filtrar y agregar datos, así como utilizar un amplio conjunto de funciones matemáticas, de cadenas, de fecha, condicionales y de otro tipo para el análisis.

Incluirlo `SOURCE` en una consulta de PPL es una forma útil de especificar los grupos de registros, los índices de campos y las fuentes de datos que se van a incluir en una consulta cuando se utiliza la API AWS CLI o la API para crear una consulta. El `SOURCE` comando solo se admite en la API AWS CLI and, no en la CloudWatch consola. Cuando utiliza la CloudWatch consola para iniciar una consulta, utiliza la interfaz de la consola para especificar los grupos de registros y el nombre y el tipo de la fuente de datos.

Se usa `aws:fieldIndex` para devolver solo datos indexados, mediante el forzado de una consulta a analizar solo los grupos de registros que están indexados en un campo que se especifique en la consulta. Los grupos de registros relevantes se seleccionan automáticamente en función de los campos especificados en el `filterIndex` comando. Esto reduce el volumen escaneado, ya que se omiten los grupos de registro que no tienen ningún evento de registro que contenga el campo especificado en la consulta y se escanean únicamente los grupos de registro que coincidan con el valor especificado en la consulta para este índice de campos. Se utiliza `aws:fieldIndex` para especificar el nombre del campo, junto con el nombre y el valor del campo en el comando source para consultar únicamente los datos indexados que contienen el campo y el valor especificados. Para obtener más información, consulte Creación de índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de análisis

Puede usar el OpenSearch PPL para consultas de grupos de registros de la clase de registro estándar.

> **Note**
>
> Para obtener información sobre todos los comandos de consulta de OpenSearch PPL compatibles con CloudWatch los registros e información detallada sobre la sintaxis y las restricciones, consulte los comandos de PPL compatibles en la Guía para desarrolladores de OpenSearch servicios.
> Para obtener información sobre otros lenguajes de consulta que puede utilizar, consulte CloudWatch Logs Insights, OpenSearch Service SQL y CloudWatch Metrics Insights

Comando o función	Consulta de ejemplo	Description (Descripción)
`fields`	`fields field1, field2`	Muestra un conjunto de campos que necesitan ser proyectados.
`join`	`LEFT JOIN left=l, right=r on l.id = r.id` `` `join_right_lg` `` `| fields l.field_1, r.field_2`	Une dos conjuntos de datos.
`where`	`where field1="success" | where field2 != "i-023fe0a90929d8822" | fields field3, field4, field5,field6 | head 1000`	Filtra los datos en función de las condiciones que especifique.
`aws:field Index`	`source = [` `` `aws:fieldIndex` `` `="region",` `` `region` `` `= "us-west-2"] | where status = 200 | head 10`	Devuelve solo datos indexados, al obligar a una consulta a escanear solo los grupos de registros que

Comando o función	Consulta de ejemplo	Description (Descripción)
		están indexados en un campo que usted especifique en la consulta.
stats	`stats count(), count(field1), min(field1), max(field1), avg(field1) by field2 | head 1000`	Realiza agregaciones y cálculos.
parse	`parse field1 ".*/(?<field2>[^/]+$)" | where field2 = "requestId" | fields field1, field2 | head 1000`	Extrae un patrón de expresión regular (regex) de una cadena y muestra el patrón extraído. El patrón extraído se puede utilizar además para crear nuevos campos o filtrar datos.
sort	`stats count(), count(field1), min(field1) as field1Alias, max(`field1`), avg(`field1`) by field2 | sort -field1Alias | head 1000`	Ordena los resultados mostrados por un nombre de campo. Use ordenar: FieldName para ordenar en orden descendente.

Comando o función	Consulta de ejemplo	Description (Descripción)
`eval`	`eval field2 = field1 * 2 | fields field1, field2 | head 20`	Modifica o procesa el valor de un campo y lo almacena en un campo diferente . Esto resulta útil para modificar matemátic amente una columna, aplicar funciones de cadena a una columna o aplicar funciones de fecha a una columna.
`rename`	`rename field2 as field1 | fields field1;`	Cambia el nombre de uno o más campos del resultado de la búsqueda.
`head`	`fields `@message` | head 20`	Limita los resultados de la consulta a las N primeras filas.
`top`	`top 2 field1 by field2`	Busca los valores más frecuentes de un campo.

Comando o función	Consulta de ejemplo	Description (Descripción)
dedup	`dedup field1 | fields field1, field2, field3`	Elimina entradas duplicadas en función de los campos que especifique.
rare	`rare field1 by field2`	Busca los valores menos frecuentes de todos los campos de la lista de campos.
subquery	`where field_1 IN [search source= ``subquery_lg`` | fields field_2] | fields id, field_1`	Realiza consultas anidadas y complejas en las instrucciones de PPL.
trendline	`trendline sma(2, field1) as field1Alias`	Calcula los promedios móviles de los campos.

Comando o función	Consulta de ejemplo	Description (Descripción)
eventStats	`eventstats sum(field1) by field2`	Enriquece los datos del evento con estadísticas resumidas calculadas. Analiza los campos específicos de sus eventos, calcula varias medidas estadísticas y, a continuación, agrega estos resultados a cada evento original como campos nuevos.
expand	`eval tags_array_string = json_extract(`@message`, '$.tags')\| eval tags_array = json_array(json_extract(tags_string, '$[0]'), json_extract(tags_string, '$[1]'))| expand tags_array as color_tags`	Divide un campo que contiene varios valores en filas independientes, creando una nueva fila para cada valor del campo especificado.

Comando o función	Consulta de ejemplo	Description (Descripción)
fillnull	`fields `@timestamp`, error_code, status_code | fillnull using status_code = "UNKNOWN", error_code = "UNKNOWN"`	Rellena los campos nulos con el valor que proporcione. Se puede usar en uno o más campos.
flatten	`eval metadata_struct = json_object('size', json_extract(metadata_string, '$.size'), 'color', json_extract(metadata_string, '$.color')) | flatten metadata_struct as (meta_size, meta_color)`	Aplana un campo. El campo debe ser de este tipo: `struct<?,?>` o `array<struct<?,?>>` .
cidrmatch	`where cidrmatch(ip, '2003:db8::/32') | fields ip`	Verifica si la dirección IP especificada está dentro del rango de CIDR dado.
fieldsummary	`where field1 != 200 | fieldsummary includefields= field1 nulls=true`	Calcula las estadísticas básicas de cada campo (recuento, recuento distinto, mínimo, máximo, promedio, stddev y media).

Comando o función	Consulta de ejemplo	Description (Descripción)
`grok`	`grok email '.+@%{HOSTNAME:host}' | fields email, host`	Analiza un campo de texto con un patrón grok y agrega los resultados al resultado de la búsqueda.
Funciones de cadena	`eval field1Len = LENGTH(field1) | fields field1Len`	Built-in funciones en PPL que pueden manipular y transformar cadenas y datos de texto dentro de las consultas de PPL. Por ejemplo, convertir mayúsculas y minúscula s, combinar cadenas, extraer partes y limpiar el texto.

Comando o función	Consulta de ejemplo	Description (Descripción)
Date-Time funciones	`eval newDate = ADDDATE(DATE('2020-08-26'), 1) | fields newDate`	Built-in funciones para gestionar y transformar los datos de fecha y hora en las consultas PPL. Por ejemplo, date_add, date_format, datediff, date-sub, timestamp add, timestamp diff, current_timezone, utc_timestamp y current_date.
Funciones de condiciones	`eval field2 = isnull(field1) | fields field2, field1, field3`	Built-in funciones que comprueban condiciones de campo específicas y evalúan las expresiones de forma condicional. Por ejemplo, si field1 es nulo, devuelve el field2.

Comando o función	Consulta de ejemplo	Description (Descripción)
Funciones matemáticas	`eval field2 = ACOS(field1) | fields field1`	Built-in funciones para realizar cálculos matemáticos y transform aciones en consultas PPL. Por ejemplo, abs (valor absoluto), round (redondea números), sqrt (raíz cuadrada) , pow (cálculo de potencia) y ceil (redondea al entero más cercano).
CryptoGraphic funciones	`eval crypto = MD5(field)| head 1000`	Para calcular el hash de un campo determinado

Comando o función	Consulta de ejemplo	Description (Descripción)
Funciones JSON	`eval valid_json = json('[1,2,3,{"f1":1,"f2":[5,6]},4]') | fields valid_json`	Built-in funciones para el manejo de JSON, incluidas las matrices, la extracción y la validación. Por ejemplo, json_object, json_array, to_json_string, json_array_length, json_extract, json_keys y json_valid.

Alcance de la consulta

Incluir SOURCE en una consulta es una forma útil de especificar los grupos de registros que se van a incluir en una consulta cuando se utiliza la API AWS CLI o para crear una consulta. El comando SOURCE solo se admite en la API AWS CLI and, no en la CloudWatch consola. Cuando utiliza la CloudWatch consola para iniciar una consulta, utiliza la interfaz de la consola para especificar los grupos de registros y el nombre y el tipo de la fuente de datos.

El comando source de PPL ahora admite varias formas de especificarlos:

1. Grupo de registro
2. Índices de campo: nuevos
3. Fuente y tipo de datos: nuevos

Grupo de registros

La selección de fuentes de grupos de registros se puede utilizar cuando los clientes saben en qué grupos de registros exactos deben buscarse

```
source = [lg:`/aws/lambda/my-function`] | where status = 200 | head 10
```

Índices de campo

La selección de fuentes basada en índices de campos reduce la cantidad de datos consultados al limitar los resultados solo a los datos indexados cuando filtra los campos de destino que se han indexado. Los grupos de registros relevantes se seleccionan automáticamente en función de los campos especificados en el comando. `filterIndex` Para obtener más información sobre los índices de campos y cómo crearlos, consulte Crear índices de campos para mejorar el rendimiento de las consultas y reducir el volumen de digitalización.

Se usa `aws:fieldIndex` para devolver solo datos indexados, mediante el forzado de una consulta a analizar solo los grupos de registros que están indexados en un campo que se especifique en la consulta. Para los grupos de registros que están indexados en este campo, se optimiza aún más la consulta al omitir los grupos de registros que no tienen ningún evento de registro que contenga el campo especificado en la consulta del campo indexado. Se reduce aún más el volumen analizado al intentar analizar solo los eventos de registro de estos grupos de registros que coincidan con el valor especificado en la consulta para este índice de campos. Para obtener más información sobre los índices de campo y cómo crearlos, consulte Crear índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de digitalización.

En PPL, `aws:fieldIndex` se usa para especificar qué pares de valores clave deben tratarse como índices. La sintaxis es la siguiente

```
source = [`aws:fieldIndex`="region", `region` = "us-west-2"] | where status = 200 |
 head 10
```

donde,

1. `` `aws:fieldIndex`="region" `` identifica la región como índice de campo.
   a. Nota: En lugar de =, los clientes pueden usar IN para especificar varios índices (ejemplo siguiente)
2. `` `region`="us-west-2" `` identifica la condición de filtro que se va a aplicar

a. Nota: En lugar de =, los clientes pueden usar IN para especificar varios valores (ejemplo siguiente)

Los clientes pueden especificar varios índices de campos de la siguiente manera

```
source = [`aws:fieldIndex` IN ("status", "region"), `status` = 200, `region` IN ("us-
west-2", "us-east-1")] | head 10
```

Fuente y tipo de datos

La selección de fuentes basadas en fuentes y tipos de datos se puede utilizar cuando los clientes saben qué fuentes de datos exactas deben consultarse. Esta consulta se ejecuta en uno o más grupos de registros que contienen la fuente y el tipo de datos especificados.

```
source = [ds:`data_source.type`] | where status = 200 | head 10
```

PPL compatible para consultas de fuentes de datos

Para respaldar el caso de uso de la consulta de fuentes de datos en PPL, puede utilizar la cláusula de selección dinámica de fuentes. Con esta sintaxis, puede consultar las fuentes de datos especificándolas en el comando de búsqueda. Puede especificar hasta 10 fuentes de datos.

Sintaxis

```
source=[ds:`DataSource1.Type1`, ds:`DataSource2.Type2`, ...ds:`DataSourcen.Typen`]
```

Consulta de ejemplo

```
search source=[ds:`DataSource1.Type1`, ds:`DataSource2.Type2`] | fields field1, field2
```

Ejemplo combinado

Los clientes pueden especificar todos los operadores de selección de fuentes en cualquier orden y el resultado sería la intersección de todas las condiciones aplicadas.

Por ejemplo, aws/lambda /my-function-1 puede contener varias fuentes y tipos de datos, incluida una amplia variedad de índices. Cuando se ejecutó la siguiente consulta, los resultados devueltos solo

contenían eventos de origen y tipo DataSource1.Type1 y que coincidieran con el criterio de «estado» = 200.

```
search source=[
    ds:`DataSource1.Type1`,
    lg:`/aws/lambda/my-function-1`,
    `aws:fieldIndex` IN ("status"), `status` = 200
]
```

Restricciones

Cuando se utiliza OpenSearch PPL para realizar consultas en Logs Insights, se aplican las siguientes restricciones. CloudWatch

- No puede usar comandos de unión o subconsulta con las consultas de fuentes de datos.

OpenSearch Lenguaje de consulta estructurado (SQL)

Esta sección contiene una introducción básica a la consulta de CloudWatch registros mediante OpenSearch SQL. Ofrece una opción que es conocida si se está acostumbrado a trabajar con bases de datos relacionales. OpenSearch SQL ofrece un subconjunto de funciones de SQL, lo que lo convierte en una buena opción para realizar consultas ad hoc y tareas de análisis de datos. Con OpenSearch SQL, puede usar comandos como SELECT, FROM, WHERE, GROUP BY, HAVING y varios otros comandos y funciones de SQL. Se puede ejecutar JOIN entre grupos de registros, correlacionar datos entre grupos de registros mediante subconsultas y utilizar el amplio conjunto de funciones JSON, matemáticas, de cadena, condicionales y otras funciones de SQL para realizar análisis eficaces de los datos de registro y de seguridad.

Se usa `filterIndex` para devolver solo datos indexados, mediante el forzado de una consulta a analizar solo los grupos de registros que están indexados en un campo que se especifique en la consulta. Reduzca el volumen escaneado omitiendo los grupos de registro que no tienen eventos de registro que contengan el campo especificado en la consulta y escaneando únicamente los grupos de registro que coincidan con el valor especificado en la consulta para este índice de campos. Se utiliza `filterIndex` para especificar el nombre del campo, junto con el nombre y el valor del campo para consultar únicamente los datos indexados que contienen el campo y el valor especificados.

Puede usar OpenSearch SQL para consultas de grupos de registros de la clase de registro estándar. SQL también admite consultas mediante el nombre y el tipo de fuente de datos.

> **Note**
>
> En la siguiente tabla se enumeran los comandos y funciones de SQL compatibles con CloudWatch los registros. Para obtener información sobre todos los comandos de OpenSearch SQL, incluida la sintaxis, consulte los comandos de SQL compatibles en la Guía para desarrolladores de OpenSearch servicios.
> Para obtener información sobre otros lenguajes de consulta que puede utilizar, consulte CloudWatch Logs Insights, OpenSearch Service PPL y CloudWatch Metrics Insights.

Comandos SQL compatibles

> **Note**
>
> En la columna de comandos de la consulta de muestra, reemplace *<logGroup>* según sea necesario en función del origen de datos que esté consultando.

Comando o función	Consulta de ejemplo	Description (Descripción)
SELECT	`SELECT \`@message\`, Operation FROM \`LogGroupA\``	Muestra los valores proyectados.
FROM	`SELECT \`@message\`, Operation FROM \`LogGroupA\``	Built-in cláusula que especific a las tablas o vistas de origen desde las que se van a recuperar los datos y admite varios tipos de uniones y subconsultas.

Comando o función	Consulta de ejemplo	Description (Descripción)
WHERE	`SELECT * FROM `LogGroupA` WHERE Operation = 'x'`	Filtra los eventos del registro en función de los criterios de campo proporcionados.
filterIndex	`SELECT * FROM `filterIndex('region' = 'us-east-1')` WHERE status = 200 LIMIT 10;`	Devuelve únicamente los datos indexados, obligando a la consulta a analizar únicamente los grupos de registros que están indexados en un campo que se especifique en la consulta.
GROUP BY	`SELECT `@logStream`, COUNT(*) as log_count FROM `LogGroupA` GROUP BY `@logStream``	Los grupos registran los eventos según la categoría y encuentran el promedio en función de las estadísticas.

Comando o función	Consulta de ejemplo	Description (Descripción)
HAVING	`SELECT `@logStream`, COUNT(*) as log_count FROM `LogGroupA` GROUP BY `@logStream` HAVING log_count > 100`	Filtra los resultados en función de las condiciones de agrupación.
ORDER BY	`SELECT * FROM `LogGroupA` ORDER BY `@timestamp` DESC`	Ordena los resultados en función de los campos de la cláusula ORDER BY. Puede especificar un orden tanto ascendente como descenden te.
JOIN	`SELECT A.`@message`, B.`@timestamp`FROM `LogGroupA` as A INNER JOIN `LogGroupB` as B ON A.`requestId` = B.`requestId``	Une los resultados de dos tablas en función de los campos en común. Debe especificarse Inner JOIN o Left Outer Join
LIMIT	`Select * from `LogGroupA` limit 10`	Limita los resultados de la consulta a las N primeras filas.

Comando o función	Consulta de ejemplo	Description (Descripción)
Funciones de cadena	`SELECT upper(Operation) , lower(Operation), Operation FROM `LogGroupA``	Built-in funciones de SQL que pueden manipular y transformar datos de texto y cadenas de texto en consultas SQL. Por ejemplo, convertir mayúsculas y minúsculas, combinar cadenas, extraer partes y limpiar el texto.
Funciones de datos	`SELECT current_date() as today, date_add(current_date(), 30) as thirty_days_later, last_day(current_date()) as month_end FROM `LogGroupA``	Built-in funciones para gestionar y transformar datos de fecha y hora en consultas SQL. Por ejemplo, date_add, date_format, datediff y current_date.

Comando o función	Consulta de ejemplo	Description (Descripción)
Funciones condicionales	`SELECT Operation, IF(Error > 0, 'High', 'Low') as error_category FROM `LogGroup A`;`	Built-in funciones que realizan acciones en función de condiciones específicas o que evalúan las expresiones de forma condicion al. Por ejemplo, CASE e IF.
Funciones de agregación	`SELECT AVG(bytes) as bytesWritten FROM `LogGroupA``	Built-in funciones que realizan cálculos en varias filas para generar un único valor resumido. Por ejemplo, SUM, COUNT, AVG, MAX y MIN.

Comando o función	Consulta de ejemplo	Description (Descripción)
Funciones JSON	`SELECT get_json_object(json_column, '$.name') as name FROM `LogGroupA``	Built-in funciones para analizar, extraer, modificar y consultar JSON-form atted datos en consultas SQL (por ejemplo, from_json , to_json, get_json_object, json_tuple) que permiten manipular las estructuras JSON en los conjuntos de datos.

Comando o función	Consulta de ejemplo	Description (Descripción)
Funciones de matriz	`SELECT scores, size(scores) as length, array_contains(scores, 90) as has_90 FROM `LogGroupA`;`	Built-in funciones para trabajar con columnas de tipo matriz en consultas SQL, lo que permite realizar operaciones como acceder a los datos de una matriz, modificarlos y analizarlos (p. ej., size, explode, array_contains).

Comando o función	Consulta de ejemplo	Description (Descripción)
Funciones de ventana	`SELECT field1, field2, RANK() OVER (ORDER BY field2 DESC) as field2Rank FROM` `` `LogGroupA` ``;	Built-in funciones que realizan cálculos en un conjunto específico de filas relaciona das con la fila actual (ventana), lo que permite realizar operaciones como la clasifica ción, los totales acumulados y las medias móviles. Por ejemplo, ROW_NUMBE R, RANK, LAG y LEAD

Comando o función	Consulta de ejemplo	Description (Descripción)
Funciones de conversión	`SELECT CAST('123' AS INT) as converted_number, CAST(123 AS STRING) as converted_string FROM `LogGroupA``	Built-in funciones para convertir datos de un tipo a otro dentro de las consultas SQL, lo que permite la transform ación de los tipos de datos y las conversiones de formato. Por ejemplo, CAST, TO_DATE, TO_TIMEST AMP y BINARY.
Funciones de predicados	`SELECT scores, size(scores) as length, array_contains(scores, 90) as has_90 FROM `LogGroupA`;`	Built-in funciones que evalúan las condicion es y devuelven valores booleanos (true/ false) en función de criterios o patrones específicos. Por ejemplo, IN, LIKE, BETWEEN, IS NULL y EXISTS.

Comando o función	Consulta de ejemplo	Description (Descripción)
Selección de varios grupos de registros	`SELECT lg1.field1, lg1.field2 from` `logGroups( logGroupIdentifier: ['LogGroup1', 'LogGroup2'])` `as lg1 where lg1.field3= "Success"`	Permite la especificación de varios grupos de registros en una sentencia SELECT
Seleccione varias fuentes de datos	`SELECT ds1.field1, ds1.field2 from` `dataSource(['DataSource1', 'DataSour ce2'])` `as ds1 where ds1.field3= "Success"`	Le permite especificar varias fuentes de datos en una sentencia SELECT

SQL compatible para consultas de varios grupos de registros

Para respaldar el caso de uso de la consulta de varios grupos de registros en SQL, puede usar el comando `logGroups`. Con esta sintaxis, puede consultar varios grupos de registros al especificarlos en el comando FROM.

Sintaxis:

```
`logGroups(
    logGroupIdentifier: ['LogGroup1','LogGroup2', ...'LogGroupn']
)
```

En esta sintaxis, puede especificar hasta 50 grupos de registros en el parámetro `logGroupIdentifier`. Para hacer referencia a los grupos de registros de una cuenta de supervisión, utilice ARN en lugar de nombres `LogGroup`.

Consulta de ejemplo:

```
SELECT LG1.Column1, LG1.Column2 from `logGroups(
    logGroupIdentifier: ['LogGroup1', 'LogGroup2']
)` as LG1 WHERE LG1.Column1 = 'ABC'
```

NO se admite la siguiente sintaxis, que incluye varios grupos de registros después de la FROM sentencia, al consultar CloudWatch los registros.

```
SELECT Column1, Column2 FROM 'LogGroup1', 'LogGroup2', ...'LogGroupn'
WHERE Column1 = 'ABC'
```

SQL compatible para consultas de fuentes de datos

Para respaldar el caso de uso de la consulta de fuentes de datos en SQL, puede usar el comando DataSource. Con esta sintaxis, puede consultar las fuentes de datos especificándolas en el comando. FROM Puede especificar hasta 10 fuentes de datos.

Sintaxis

```
`dataSource(
    ['DataSource1', 'DataSource2', ...'DataSourcen']
)`
```

Ejemplo de consulta

```
SELECT DS1.Column1, DS1.Column2 from `dataSource(
    ['DataSource1', 'DataSource2']
)` as DS1 WHERE DS1.Column1 = 'ABC'
```

Alcance de la consulta

En la API AWS CLI y, puede especificar qué registros consultar mediante el grupo de registros, la fuente y el tipo de datos y los índices de campo.

Grupo de registros

La selección de fuentes de los grupos de registros se puede utilizar cuando los clientes saben en qué grupos de registros exactos deben buscarse

```
SELECT * FROM `logGroups(logGroupIdentifier: ['/aws/lambda/my-function'])`;
```

Fuente y tipo de datos

Los clientes pueden consultar sus registros mediante el nombre y el tipo de fuente de datos.

La selección de fuentes basada en la fuente y el tipo de datos se puede utilizar cuando los clientes saben qué fuentes de datos exactas deben consultarse. Esta consulta se ejecuta en uno o más grupos de registros que contienen la fuente y el tipo de datos especificados.

Para respaldar el caso de uso de la consulta de fuentes de datos en SQL, puede usar el comando DataSource. Con esta sintaxis, puede consultar las fuentes de datos especificándolas en el comando FROM. Puede especificar hasta 10 fuentes de datos.

Sintaxis:

```
`dataSource(
    ['DataSource1.Type1', 'DataSource2.Type2', ...'DataSourcen.Typen']
)`
```

Consulta de ejemplo:

```
SELECT DS1.Column1, DS1.Column2 from `dataSource(
    ['DataSource1.Type1', 'DataSource2.Type2']
)` as DS1 WHERE DS1.Column1 = 'ABC'
```

Para obtener más información sobre las consultas por fuentes de datos, consulteUtilice facetas para agrupar y explorar los registros.

Ejemplo combinado

Los clientes pueden especificar todos los operadores de selección de fuentes dentro de las comillas invertidas en cualquier orden y los resultados se basarán en la intersección de todas las condiciones aplicadas.

Por ejemplo, aws/lambda /my-function-1 puede contener varios tipos y fuentes de datos, incluidos una amplia variedad de índices. Cuando se ejecutó la siguiente consulta, los resultados devueltos solo contenían eventos de origen y tipo DataSource1.Type1 y que coincidieran con el criterio de «estado» = 200.

```
SELECT * FROM `
   logGroups(logGroupIdentifier: ['/aws/lambda/my-function'])
   filterIndex('status' = 200)
   dataSource(['DataSource1.Type1'])
`;
```

Índices de campo

La selección Index-based de fuentes de campo identifica automáticamente los grupos de registros relevantes cuando los filtros se centran en los campos indexados, lo que reduce el volumen de digitalización y el tiempo de ejecución de las consultas.

Se usa `filterIndex` para devolver solo datos indexados, mediante el forzado de una consulta a analizar solo los grupos de registros que están indexados en un campo que se especifique en la consulta. Para los grupos de registros que están indexados en este campo, se optimiza aún más la consulta al omitir los grupos de registros que no tienen ningún evento de registro que contenga el campo especificado en la consulta del campo indexado. Se reduce aún más el volumen analizado al intentar analizar solo los eventos de registro de estos grupos de registros que coincidan con el valor especificado en la consulta para este índice de campos. Para obtener más información sobre los índices de campo y cómo crearlos, consulte Crear índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de digitalización.

En SQL, FilterIndex se usa para especificar qué pares de valores clave deben tratarse como índices. La sintaxis es la siguiente

```
SELECT * FROM `filterIndex('region' = 'us-east-1')`;
```

donde,

1. FilterIndex (...) especifica y trata los valores clave que contienen como índices de campo. Cada par de valores clave está separado por una coma (ejemplo siguiente)
2. 'region' = 'us-east-1' especifica la condición real que debe aplicarse
   a. Nota: En lugar de =, los clientes pueden usar IN para especificar varios valores (ejemplo siguiente)

El uso de varios FilterIndex combinaría las condiciones utilizando «AND». En el ejemplo, se consultarían los registros que coincidan con el estado = 200 y la región en us-east-1 o us-west-2.

```
SELECT * FROM `filterIndex('status' = 200, 'region' IN ['us-east-1', 'us-west-2'])`;
```

Restricciones

Las siguientes restricciones se aplican cuando se utiliza SQL para realizar consultas en Logs Insights. OpenSearch CloudWatch

- Solo se puede usar una JOIN en una sentencia SELECT.
- No puede usar JOIN ni subconsultas con consultas de fuentes de datos.
- Solo se admite un nivel de subconsultas anidadas.
- No se admiten consultas de varias sentencias separadas por punto y coma (;).
- No se admiten consultas que contengan nombres de campo idénticos pero que solo difieran en mayúsculas y minúsculas (como field1 y FIELD1).

  Por ejemplo, no se admite la siguiente consulta.

  ```
  Select AWSAccountId, AwsAccountId from LogGroup
  ```

  Sin embargo, se admite la siguiente consulta porque el nombre del campo (`@logStream`) es idéntico en ambos grupos de registros:

  ```
  Select a.`@logStream`, b.`@logStream` from Table A INNER Join Table B on a.id = b.id
  ```

- Las funciones y expresiones deben funcionar con los nombres de los campos y formar parte de una sentencia SELECT con un grupo de registros especificado en la cláusula FROM.

  Por ejemplo, no se admite esta consulta:

  ```
  SELECT cos(10) FROM LogGroup
  ```

  Se admite esta consulta:

  ```
  SELECT cos(field1) FROM LogGroup
  ```

- Cuando utilice comandos SQL o PPL, encierre determinados campos entre comillas invertidas para consultarlos correctamente. Las comillas simples son necesarias para los campos con caracteres especiales (no alfabéticos ni numéricos). Por ejemplo, incluya `@message`, `Operation.Export` y `Test::Field` entre acentos graves. No es necesario incluir los campos con nombres exclusivamente alfabéticos entre comillas simples.

  Ejemplo de consulta con campos sencillos:

  ```
  SELECT SessionToken, Operation, StartTime  FROM `LogGroup-A`
  LIMIT 1000;
  ```

Consulta similar con comillas invertidas agregadas:

```
SELECT `@SessionToken`, `@Operation`, `@StartTime`  FROM `LogGroup-A` LIMIT 1000;
```

Utilice un lenguaje natural para generar y actualizar las consultas de CloudWatch Logs Insights

CloudWatch Logs admite una función de consulta en lenguaje natural que le ayuda a generar y actualizar consultas para CloudWatch Logs Insights, OpenSearch Service PPL, OpenSearch Service SQL y CloudWatch Metrics Insights.

Con esta función, puede hacer preguntas sobre los datos de CloudWatch Logs que busca o describirlos en un lenguaje sencillo. Esta función de lenguaje natural genera una consulta según una petición presentada y proporciona una explicación línea por línea sobre cómo funciona la consulta. También puede actualizar la consulta para investigar más a fondo los datos.

Según el entorno, puede introducir peticiones como “¿Cuáles son las 100 direcciones IP principales de origen por bytes transferidos?” y “Busque las 10 solicitudes de función de Lambda más lentas”.

> **Note**
>
> La característica de consulta en lenguaje natural es un servicio regional. En algunas regiones, esta característica realiza llamadas interregionales a regiones de Estados Unidos para procesar las solicitudes de consulta. Para obtener más información, consulte Amazon CloudWatch amplía el soporte regional para el resumen de resultados de consultas en lenguaje natural y la generación de consultas.

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 iniciar sesión con las políticas CloudWatchLogsFullAccess, CloudWatchLogsReadOnlyAccessAdministratorAccess, o de ReadOnlyAccessIAM, o tener el `cloudwatch:GenerateQuery` permiso correspondiente.

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.

Ejemplos: generar una consulta en lenguaje natural

Para generar una consulta en lenguaje natural, introduzca una petición y seleccione Generar nueva consulta. En estos ejemplos, se muestran las consultas que realizan una búsqueda básica.

Petición

A continuación, se muestra un ejemplo de una petición que indica la función de buscar las 10 invocaciones más lentas de la función de Lambda.

```
Find the 10 slowest requests
```

Consultar

La siguiente es la consulta que utiliza el lenguaje de consultas CloudWatch Logs Insights que la capacidad de lenguaje natural generó en función de la solicitud. 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.

Petición

Para generar una consulta OpenSearch SQL, seleccione la pestaña OpenSearch SQL y, a continuación, abra el cuadro de diálogo del generador de consultas para introducir la solicitud en lenguaje natural. A continuación, se muestra un ejemplo de un mensaje que utiliza la función de lenguaje natural para generar una consulta OpenSearch SQL.

```
Give me the number of errors and exceptions per hour
```

Consultar

La siguiente es la consulta de SQL generada por esa petición, que se puede utilizar para encontrar el número de errores y excepciones agregados por hora:

```
SELECT DATE_FORMAT(`@timestamp`, 'yyyy-MM-dd HH') AS hour,
       COUNT(*) AS error_count
FROM `/aws/lambda/CloudWatchOdysseyQueryGen`
WHERE `@message` LIKE '%error%'
  OR `@message` LIKE '%exception%'
GROUP BY DATE_FORMAT(`@timestamp`, 'yyyy-MM-dd HH')
ORDER BY hour
```

Petición

Para generar una consulta OpenSearch PPL, seleccione la pestaña OpenSearch PPL y, a continuación, abra el cuadro de diálogo del generador de consultas para introducir la solicitud en lenguaje natural. A continuación, se muestra un ejemplo de un mensaje que utiliza la función de lenguaje natural para generar una consulta OpenSearch PPL.

```
Give me all unique exception messages
```

Consultar

La siguiente es la consulta de PPL generada por esa petición, que puede utilizar para buscar los mensajes de excepción únicos en sus registros:

```
dedup @message
| fields @message
```

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 petición que busca las 10 invocaciones de funciones de Lambda más lentas, esta petición ahora indica 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 la consulta actualizada que utiliza el lenguaje de consultas CloudWatch Logs Insights. 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.
```

Desactivación del uso de los 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 pueden usarse para mejorar la calidad de CloudWatch Logs Insights. 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 .

Registros y campos detectados compatibles

CloudWatch Logs Insights admite diferentes tipos de registros. Por cada registro que se envía a un grupo de CloudWatch registros de clase estándar en Amazon Logs, CloudWatch Logs Insights genera automáticamente cinco campos del 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.
- `@entity` contiene JSON aplanada relacionada con entidades para la característica Telemetría relacionada con Explore.

  Por ejemplo, esta JSON puede representar una entidad.

```
{
```

```
  "Entity": {
    "KeyAttributes": {
      "Type": "Service",
      "Name": "PetClinic"
    },
    "Attributes": {
      "PlatformType": "AWS::EC2",
      "EC2.InstanceId": "i-1234567890123"
    }
  }
}
```

Para esta entidad, los campos del sistema extraídos serían los siguientes:

```
@entity.KeyAttributes.Type = Service
@entity.KeyAttributes.Name = PetClinic
@entity.Attributes.PlatformType = AWS::EC2
@entity.Attributes.EC2.InstanceId = i-1234567890123
```

> **Note**
>
> La detección de campos solo se admite para los grupos de registro de la clase de registro Estándar. Para obtener más información acerca de las clases de registros, consulte Clases 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 Registra la sintaxis de consulta del lenguaje 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`.

> **Note**
>
> Excepto en el caso de `@message`, `@timestamp` o `@log`, se pueden crear índices de campos para los campos descubiertos. Para obtener más información sobre los índices de campo, consulte Creación de índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de análisis.

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 X-Ray seguimiento, 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.

Tipo de registro	Campos de registro detectados
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.

Creación de índices de campo para mejorar el rendimiento de las consultas y reducir el volumen de análisis

Se pueden crear índices de campo de los campos de los eventos de registro para realizar búsquedas eficientes y basadas en la igualdad. Cuando, a continuación, utilizas un índice de campos en una consulta de CloudWatch Logs Insights, la consulta intenta omitir el procesamiento de los eventos de registro que se sabe que no incluyen el campo indexado. Esto reduce el volumen de análisis de las consultas que emplean índices de campo, lo que permite devolver resultados con mayor rapidez. Esto puede ayudar a buscar rápidamente petabytes del total de registros en miles de grupos de registros y a centrarse más rápidamente en los registros pertinentes. Los campos adecuados para indexar son aquellos que se necesitan consultar con frecuencia. Los campos que tienen una alta cardinalidad de valores también son buenos candidatos para los índices de campo, ya que una consulta que utilice estos índices de campo se completará más rápido, dado que limita los eventos de registro que coinciden con el valor objetivo.

Supongamos que se ha creado un índice de campo para `requestId`. Luego, cualquier consulta de CloudWatch Logs Insights sobre ese grupo de registros que incluya `requestId = value` o `requestId IN [value, value, ...]` intente procesar solo los eventos de registro que se sabe que contienen ese campo indexado y el valor consultado, y que CloudWatch Logs haya detectado un valor para ese campo en el pasado.

También se pueden aprovechar los índices de campos para crear consultas eficientes de un mayor número de grupos de registros. Si utiliza el comando `filterIndex` en la consulta en lugar del comando `filter`, la consulta se ejecutará en grupos de registros seleccionados en los eventos de registro que tengan índices de campo. Estas consultas pueden analizar hasta 10 000 grupos de registros, que se eligen al especificar hasta cinco prefijos de nombre de grupos de registros. Si se trata de una cuenta de monitorización mediante la observación CloudWatch multicuenta, puede elegir

todas las cuentas de origen o especificar cuentas de origen individuales para seleccionar los grupos de registros».

Los campos indexados distinguen entre mayúsculas y minúsculas. Por ejemplo, un índice de campo de `RequestId` no coincidirá con un evento de registro que contenga `requestId`.

Los índices de campos solo se admiten para los formatos de registro estructurado de JSON y los registros de servicio.

CloudWatch Los registros proporcionan índices de campo predeterminados para todos los grupos de registros de la clase de registros estándar. Los índices de campo predeterminados están disponibles automáticamente para los siguientes campos:

- `@logStream`
- `@aws.region`
- `@aws.account`
- `@source.log`
- `@data_source_name`
- `@data_source_type`
- `@data_format`
- `traceId`
- `severityText`
- `attributes.session.id`

CloudWatch Los registros también proporcionan índices de campo predeterminados para determinadas combinaciones de nombres y tipos de fuentes de datos. Los índices de campo predeterminados están disponibles automáticamente para las siguientes combinaciones de nombre y tipo de fuente de datos:

Nombre y tipo de fuente de datos	Índices de campo predeterminados

`amazon_vpc.flow`	`action`
	`logStatus`
	`region`

Nombre y tipo de fuente de datos	Índices de campo predeterminados

	`flowDirection`
	
	`type`

`amazon_route53.resolver_query`	`query_type`
	
	`transport`
	
	`rcode`

`aws_waf.access`	`action`
	
	`httpRequest.country`

`aws_cloudtrail.data`
 	
`aws_cloudtrail.management`
	`eventSource`
	
	`eventName`
	
	`awsRegion`
	
	`userAgent`
	
	`errorCode`
	
	`eventType`
	
	`managementEvent`
	
	`readOnly`
	
	`eventCategory`
	
	`requestId`

Los índices de campos predeterminados se suman a cualquier índice de campo personalizado que defina en su política. Los índices de campo predeterminados no se incluyen en la cuota de índices de campo.

CloudWatch Los registros indexan solo los eventos de registro ingeridos después de crear una política de indexación. No indexa los eventos de registro ingeridos antes de que se creara la política.

Tras crear un índice de campos, cada evento de registro coincidente permanece indexado durante 30 días a partir del momento de la ingesta del evento de registro.

> **Note**
>
> Si se crea una política de indexación de campos en una cuenta de supervisión, esa política no se utilizará para los grupos de registros de las cuentas de origen vinculadas. Una política de índice de campos solo se aplica a la cuenta en la que se creó.

En el resto de los temas de esta sección se explica cómo crear índices de campo. Para obtener información sobre cómo hacer referencia a los índices de campo en las consultas, consulte filterIndex y filter.

Temas

Sintaxis y cuotas del índice de campos

Los índices de campos se crean mediante la creación de políticas de índices de campos. Se pueden crear políticas de indexación de cuenta que se apliquen a toda la cuenta y también se pueden crear políticas que se apliquen solo a un grupo de registros. Para las políticas de indexación para toda la cuenta, se puede tener una que se aplique a todos los grupos de registro de la cuenta. También se pueden crear políticas de indexación de cuenta que se apliquen a un subconjunto de grupos de registros de la cuenta, seleccionados por los prefijos de sus nombres de grupos de registro. Si se tienen varias políticas a nivel de cuenta en la misma cuenta, los prefijos de los nombres de los grupos de registros de estas políticas no pueden superponerse. Del mismo modo, puede crear políticas de indexación a nivel de cuenta que se apliquen a una combinación de nombre y tipo de fuente de datos específica. Solo se puede crear una política de cuenta por combinación de nombre y tipo de fuente de datos.

Las políticas de indexación de campos a nivel de grupo de registros anulan las políticas de índice de campos a nivel de cuenta, que se aplican al grupo de registros en su conjunto (por ejemplo, las

políticas a nivel de cuenta sin criterios de selección o con criterios de selección basados en el prefijo del nombre del grupo de registros). Account-level Las políticas que coincidan a nivel de eventos de registro (por ejemplo, para una combinación de nombre y tipo de fuente de datos determinada) se aplicarán además de las políticas que coincidan con el grupo de registro en su conjunto. Si crea una política de indexación a nivel de grupo de registros, ese grupo de registros no utilizará políticas a nivel de cuenta que coincidan a nivel de grupo de registros.

Las coincidencias de los eventos de registro con los nombres de los índices de campo distinguen mayúsculas de minúsculas. Por ejemplo, un índice de campo de `RequestId` no coincidirá con un evento de registro que contenga `requestId`.

Puede tener hasta 40 políticas de indexación a nivel de cuenta, de las cuales 20 pueden usar criterios de selección de prefijos de nombres de grupos de registros y 20 pueden usar criterios de selección basados en la fuente de datos. Si se tienen varias políticas de indexación a nivel de cuenta filtradas para incluir prefijos de nombres de grupos de registros, ninguna de ellas podrá utilizar prefijos de nombres de grupos de registros iguales o superpuestos. Por ejemplo, si se tiene una política filtrada para registrar los grupos que comienzan por `my-log`, no se puede tener otra política de indexación de campos filtrada a `my-logpprod` o `my-logging`. Del mismo modo, si tiene varias políticas de indexación a nivel de cuenta filtradas por combinaciones de nombre y tipo de fuente de datos, ninguna de ellas podrá utilizar el mismo nombre y tipo de fuente de datos. Por ejemplo, si tiene una política filtrada por el nombre `amazon_vpc` y el tipo de fuente de datos, no `flow` podrá crear otra política con esta combinación.

Si tiene una política de indexación a nivel de cuenta que no tiene prefijos de nombre y se aplica a todos los grupos de registros, no se puede crear ninguna otra política de indexación a nivel de cuenta con filtros de prefijos de nombres de grupos de registros; puede crear políticas de índice a nivel de cuenta que utilicen filtros de nombre y tipo de fuente de datos.

Cada política de indexación dispone de las siguientes cuotas y restricciones:

- Se pueden incluir hasta 20 campos en la política.
- Cada nombre de campo puede incluir hasta 100 caracteres.
- Para crear un índice de un campo personalizado en sus grupos de registros que comience por `@`, se debe especificar el campo con un `@` extra al principio del nombre del campo. Por ejemplo, si los eventos de registro incluyen un campo denominado `@userId`, se debe especificar `@@userId` a fin de crear un índice para este campo.

En el caso de las políticas de indexación a nivel de cuenta con criterios de selección basados en el nombre y el tipo de la fuente de datos, se aplica una restricción adicional: todos los campos deben ser tipos de datos primitivos; las primitivas anidadas solo se admiten en el caso de las estructuras.

Campos generados y campos reservados

CloudWatch Logs Insights genera automáticamente los campos del sistema en cada evento de registro. Estos campos generados llevan el prefijo `@`. Si se desea obtener más información sobre los campos generados, consulte Registros y campos detectados compatibles.

De estos campos generados, se admiten los siguientes para su uso como índices de campos:

- `@logStream`
- `@ingestionTime`
- `@requestId`
- `@type`
- `@initDuration`
- `@duration`
- `@billedDuration`
- `@memorySize`
- `@maxMemoryUsed`
- `@xrayTraceId`
- `@xraySegmentId`

Para indexar estos campos generados, no es necesario añadir un `@` extra cuando se los especifica, como ocurre con los campos personalizados que comienzan por `@`. Por ejemplo, para crear un índice de campos `@logStream`, basta con especificar `@logStream` como índice de campo.

CloudWatch Logs proporciona índices de campos predeterminados para todos los grupos de registros de la clase de registro estándar. Los índices de campo predeterminados están disponibles automáticamente para los siguientes campos:

- `@logStream`
- `@aws.region`
- `@aws.account`

- `@source.log`
- `@data_source_name`
- `@data_source_type`
- `@data_format`
- `traceId`
- `severityText`
- `attributes.session.id`

CloudWatch Los registros también proporcionan índices de campo predeterminados para determinadas combinaciones de nombres y tipos de fuentes de datos. Los índices de campo predeterminados están disponibles automáticamente para las siguientes combinaciones de nombre y tipo de fuente de datos:

Nombre y tipo de fuente de datos	Índices de campo predeterminados

`amazon_vpc.flow`	`action`
	`logStatus`
	`region`
	`flowDirection`
	`type`

`amazon_route53.resolver_query`	`query_type`
	`transport`
	`rcode`

`aws_waf.access`	`action` `httpRequest.country`
`aws_cloudtrail.data` `aws_cloudtrail.management`	`eventSource` `eventName`

Nombre y tipo de fuente de datos	Índices de campo predeterminados

	`awsRegion`
	`userAgent`
	`errorCode`
	`eventType`
	`managementEvent`
	`readOnly`
	`eventCategory`
	`requestId`

Los índices de campos predeterminados se suman a cualquier índice de campo personalizado que defina en su política. Los índices de campo predeterminados no se incluyen en la cuota de índices de campo.

Campos secundarios y campos de matriz en los registros JSON

Se pueden indexar campos que sean secundarios anidados o campos de matriz en los registros JSON.

Por ejemplo, se puede crear un índice del campo `accessKeyId` secundario dentro del campo `userIdentity` de este registro:

```
{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EXAMPLE_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "11112222",
        "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",
                "currentState": {
                    "code": 0,
                    "name": "pending"
                },
                "previousState": {
                    "code": 80,
                    "name": "stopped"
                }
            }]
        }
    }
}
```

Si se desea crear este campo, utilice la notación de puntos (`userIdentity.accessKeyId`) tanto al crear el índice de campos como al especificarlo en una consulta. La consulta tendría el siguiente aspecto:

```
fields @timestamp, @message
| filterIndex userIdentity.accessKeyId = "11112222"
```

En el caso del ejemplo anterior, el campo `instanceId` está en una matriz dentro de `requestParameters.instancesSet.items`. Para representar este campo tanto al crear el índice de campos como al realizar consultas, consúltelo como `requestParameters.instancesSet.items.0.instanceId`. El 0 se refiere a la posición de ese campo en la matriz.

Por lo tanto, una consulta para este campo podría ser la siguiente:

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

Creación de una política de indexación de campos a nivel de cuenta

Se utilizan los pasos de esta sección para crear una política de indexación de campos que se aplique a todos los grupos de registros de la cuenta o a varios grupos de registros que tengan nombres de grupos de registros que comiencen por la misma cadena.

Creación de una política de indexación de campos a nivel de cuenta

1. Abra la CloudWatch consola en. https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación izquierdo, elija Configuración y, a continuación, la pestaña Registros.
3. En la sección Políticas de indexación a nivel de cuenta, seleccione Administrar.
4. Elija Crear política de índice.
5. En Nombre de la política, introduzca un nombre para la nueva política.
6. En Seleccione el ámbito de la política, realice una de las siguientes acciones:
   - Seleccione Todos los grupos de registros estándar para que la política de indexación se aplique a todos los grupos de registros de clase Estándar de la cuenta.
   - Elija Grupos de registros por coincidencia de prefijos para aplicar la política a un subconjunto de grupos de registros que tengan todos nombres que comiencen por la misma cadena. A continuación, introduzca el prefijo de estos grupos de registros en Introduzca un nombre de prefijo.

     Después de introducir el prefijo, puede elegir Vista previa de los grupos de registros coincidentes con los prefijos para confirmar que el prefijo coincide con los grupos de registros que esperaba.

     Elija Registrar datos por fuente de datos para aplicar la política a una combinación de nombre y tipo de fuente de datos específica. A continuación, puede seleccionar la fuente de datos y el tipo de datos en el menú desplegable.

     Tras seleccionar el nombre y el tipo de la fuente de datos, puede seleccionar Obtener campos para rellenar la sección Configurar índices y facetas de los campos con información relevante, como los campos disponibles, los grupos de registros incluidos y los índices de campos predeterminados y personalizados.
7. En el caso de Configuración de un campo de indexación personalizado, elija Agregación de ruta de campo para introducir el primer campo que se va a indexar.

A continuación, introduzca la cadena que desee utilizar como valor del nombre del campo o seleccione un campo en el menú desplegable. Debe coincidir exactamente con las mayúsculas y minúsculas que aparecen en los eventos de registro. Por ejemplo, si sus eventos de registro incluyen `requestId`, se debe ingresar `requestId` aquí. `RequestId`, `requestID` y `request Id` no coincidirían.

Si se desea indexar un campo de registro personalizado que comience por el carácter `@`, se debe incluir un carácter `@` adicional al introducir la cadena de índice. Por ejemplo, si se tiene un campo de registro personalizado `@emailname`, se debe introducir `@@emailname` en el cuadro Agregar ruta de campo.

También puede crear índices para los `@logStream` campos `@ingestionTime` y que CloudWatch Logs genera automáticamente. Si lo hace, no tiene que añadir un `@` extra al especificarlos.

8. (Opcional) Además de especificar la ruta del campo, puede seleccionar Establecer como una faceta para crear el campo como una faceta.
9. Repita el paso anterior para agregar hasta 20 índices de campo.
10. Cuando haya terminado, seleccione Create (Crear).

Creación de una política de indexación de campos a nivel de grupo de registros

Se deben seguir los pasos de esta sección para crear una política de indexación de campos que se aplique a un único grupo de registros.

Creación de una política de indexación de campos a nivel de grupo de registros

1. Abra la CloudWatch consola en. https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación izquierdo, elija Logs (Registros), Log groups (Grupos de registros).
3. Elija el nombre del grupo de registro.
4. Seleccione la pestaña Índices de campo.
5. Elija Administración de índices de campos para este grupo de registros
6. En Administración de índices de campos a nivel de grupo de registros, elija Agregación de ruta de campo para introducir el primer campo que se va a indexar.

A continuación, introduzca la cadena que desee utilizar como valor del nombre del campo. Debe coincidir exactamente con las mayúsculas y minúsculas que aparecen en los eventos de registro. Por ejemplo, si sus eventos de registro incluyen `requestId`, se debe ingresar `requestId` aquí. `RequestId`, `requestID` y `request Id` no coincidirían.

Si se desea indexar un campo de registro personalizado que comience por el carácter `@`, se debe incluir un carácter `@` adicional al introducir la cadena de índice. Por ejemplo, si se tiene un campo de registro personalizado `@emailname`, se debe introducir `@@emailname` en el cuadro Agregar ruta de campo.

También puede crear índices para los `@logStream` campos `@ingestionTime` y que CloudWatch Logs genera automáticamente. Si lo hace, no tiene que añadir un `@` extra al especificarlos.

7. (Opcional) Además de especificar la ruta del campo, puede seleccionar Establecer como una faceta para crear el campo como una faceta.
8. Repita el paso anterior para agregar hasta 20 índices de campo.
9. Cuando haya terminado, elija Save.

Selección de grupos de registros al crear una consulta

En esta sección se explican las diversas formas en que se pueden seleccionar grupos de registros para incluirlos en una consulta.

Selección de grupos de registros para una consulta en la consola

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. Seleccione el idioma de consulta que desea utilizar en esta consulta. Puede elegir entre: Logs Insights QL, OpenSearchPPL o OpenSearch SQL.
4. Hay tres formas de seleccionar grupos de registro para la consulta:
   - Utilice el cuadro de nombre del grupo de registros. Este es el método de selección predeterminada. Puede introducir hasta 50 nombres de grupos de registro con este método. Si se trata de una cuenta de supervisión en condiciones de observación CloudWatch 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.

- Utilice la sección Criterios de grupos de registros. En esta sección, se pueden elegir grupos de registros en función del prefijo de los nombres de los grupos de registros. Se pueden incluir hasta cinco prefijos en una consulta. Se seleccionarán los grupos de registros que tengan estos prefijos en sus nombres. Como alternativa, la opción Todos los grupos de registro selecciona todos los grupos de registro de la cuenta.
- Si se trata de una cuenta de monitorización mediante la observación CloudWatch multicuenta, puede seleccionar Todas las cuentas en el menú desplegable de cuentas para seleccionar los grupos de registros de todas las cuentas vinculadas. Como alternativa, puede seleccionar de manera individual qué cuentas deben incluirse en esta consulta.

Si las opciones coinciden con más de 10 000 grupos de registros, aparecerá un error que pedirá que se restrinja la selección.

5. La clase de registro predeterminada para una consulta es Estándar. Puede usar Clase de registro para cambiarla a Infrequent access.

Usando el AWS CLI

Para realizar este tipo de selecciones al iniciar una consulta desde la línea de comandos, se puede usar el comando `source` en la consulta. Para obtener más información y ejemplos, consulta SOURCE.

Efectos de eliminar una política de indexación de campos

Si se elimina una política de indexación de campos que ha estado en vigor durante un tiempo, ocurre lo siguiente:

- Hasta 30 días después de la eliminación de la política, las consultas pueden seguir beneficiándose de los eventos de registro indexado.
- Si se elimina una política de indexación a nivel de grupo de registros y ya existe una política a nivel de cuenta que se aplicaría a ese grupo de registros, la política a nivel de cuenta finalmente se aplicará a ese grupo de registros.

Utilice facetas para agrupar y explorar los registros

Las facetas son útiles para analizar los registros, ya que permiten filtrar y agrupar los datos de forma interactiva sin ejecutar consultas. Una faceta es un campo de los registros (por ejemplo, `ServiceName` o`StatusCode`) que permite filtrar, agregar y analizar los distintos grupos de registros. Puede ver la lista de campos facetados en la consola de CloudWatch Logs Insights, junto con el recuento de eventos de registro para cada valor de faceta en función del intervalo de tiempo seleccionado. A medida que selecciona distintas facetas y valores, los valores y recuentos de las facetas se actualizan en tiempo real, lo que le permite explorar sus registros de forma interactiva.

Cada faceta muestra los valores y recuentos disponibles, que se extraen automáticamente de los campos de los registros en función del intervalo de tiempo y el alcance de la consulta seleccionados, y se conservan durante 30 días. Los recuentos de facetas que se muestran son aproximados. Puede usar las facetas predeterminadas, como el nombre o el tipo de fuente de datos, para explorar sus registros o crear facetas personalizadas en cualquiera de los campos de sus registros. El nombre de la fuente de datos es un servicio o aplicación de AWS que genera los registros (por ejemplo, Route 53, Amazon VPC o CloudTrail) y el tipo de fuente de datos es el tipo específico de registro generado por ese servicio. Las facetas predeterminadas las crea CloudWatch e incluyen`@aws.region`, `@data_source_name`@`data_source_type`, y. `@data_format` Para obtener más información, consulte Administración de registros:. Las facetas solo están disponibles para los registros que se ingieren en la cuenta. Si ha configurado la observabilidad entre cuentas, la cuenta de supervisión no podrá ver las facetas en función de los registros de las cuentas de origen.

Para crear facetas adicionales, seleccione los campos de sus registros que sean relevantes para la solución de problemas y configúrelos mediante las políticas de indexación. Para las facetas personalizadas, recomendamos crearlas en campos de baja cardinalidad (campos con menos de 100 valores únicos por día, como Status y). ApplicationName Las facetas con más de 100 valores únicos por día se clasifican como de cardinalidad alta y los valores de estas facetas no se muestran. Seleccione una o más facetas y elija ejecutar consultas en sus registros.

Para empezar con las facetas de CloudWatch Logs Insights:

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. (Opcional) Utilice el selector de intervalo de tiempo para seleccionar el período de tiempo que desee analizar. Para el intervalo de tiempo seleccionado, las facetas y los valores disponibles se muestran en el panel.

4. Seleccione las facetas para explorar sus datos y ver actualizaciones en tiempo real de las distribuciones de valores entre las facetas.

   No se muestran las facetas con más de 100 valores únicos. Para consultar valores específicos, utilice filtros en la consulta.

Para ejecutar una consulta basada en facetas

1. Seleccione uno o más valores en todas las facetas.
2. El recuento de eventos se actualizará en función de las facetas y los valores seleccionados.
3. A medida que se seleccionan los valores de las facetas, el ámbito de la consulta se actualiza para reflejar la selección.
4. Tras seleccionar los valores de las facetas, elija ejecutar para ejecutar la consulta.
5. El número máximo de valores únicos admitidos por faceta es 100. Por ejemplo, si hay más de 100 valores para una faceta, todos los recuentos se muestran como «-», lo que indica que se desconocen los valores.

Para guardar una consulta basada en facetas

1. Cree la consulta con uno o más valores de faceta.
2. El resto de los pasos son los mismos que para guardar una consulta de Logs Insights. Consulte Guardar consultas de CloudWatch Logs Insights.
3. Las consultas guardadas están disponibles en la sección Consultas guardadas. Al recuperar una consulta guardada, incluirá automáticamente las facetas y los valores utilizados en la consulta, lo que facilitará el análisis de los registros.

Para crear una faceta a nivel de cuenta

1. Para crear facetas, primero debe crear el campo como un índice y configurarlo como una faceta. En el panel de navegación, seleccione Configuración, Registros y Políticas de índice a nivel de cuenta. Como alternativa, puede seleccionar Administrar facetas en el panel de facetas.
2. Elija Crear nueva política de indexación. Para obtener más información sobre la creación de políticas de índice, consulteCreación de una política de indexación de campos a nivel de cuenta.

3. Para crear una faceta, active Definir como faceta para el campo seleccionado en la página de creación de políticas de indexación.

Gestión de facetas mediante API

La gestión de facetas se puede realizar mediante la política de indexación de campos. Consulte `field index`las API para obtener más información.

API de índice de campos

No.	Name	Description (Descripción)
1	PutIndexPolicy	Crea o actualiza una política de indexación de campos para el grupo de registros específico
2	PutAccountPolicy	Crea una política de protección de datos a nivel de cuenta, una política de filtro de suscripci ones, una política de índice de campos, una política de transformadores o una política de extracción de métricas que se aplica a todos los grupos de registros o a un subconjunto de grupos de registros de la cuenta
3	DeleteIndexPolicy	Elimina una política de indexación de campos a nivel de grupo de registros que se aplicó a un único grupo de registros
4	DeleteAccountPolicy	Elimina una política de cuentas de Logs CloudWatch

Ver los registros circundantes en CloudWatch Logs Insights

Puede ver los eventos de registro que se produjeron antes y después de un registro de registro específico devuelto por su consulta de CloudWatch Logs Insights. Los registros circundantes le ayudan a comprender el contexto en torno a los errores, las advertencias u otros eventos importantes.

Cómo funcionan los registros circundantes

Al ejecutar una consulta de CloudWatch Logs Insights, cada registro de registro de los resultados incluye una opción de registros circundantes. Al elegir esta opción, CloudWatch Logs Insights muestra los eventos de registro adicionales del mismo flujo de registro que se produjeron inmediatamente antes y después del registro seleccionado. Estos eventos aparecen aunque no coincidan con los filtros de consulta originales.

Puede configurar el número de líneas de registro circundantes que se van a mostrar. Elija entre 5, 10, 20, 50 o 100 líneas de registro por encima y por debajo del registro seleccionado.

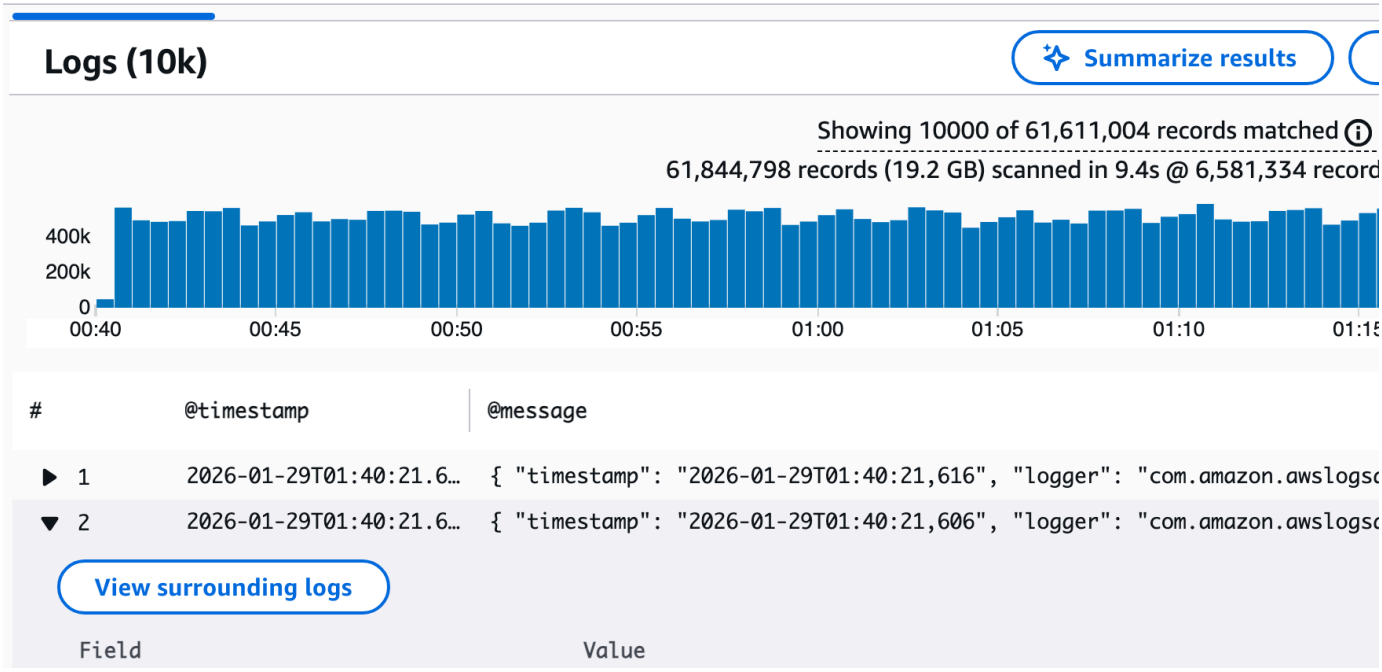
Los registros circundantes son útiles en los siguientes escenarios:

- Depurar errores mediante la visualización de los eventos que provocaron un error
- Investigando las advertencias o los tiempos de espera en sistemas distribuidos
- Comprender la secuencia de eventos en su aplicación
- Analizar los problemas en entornos de registros de gran volumen

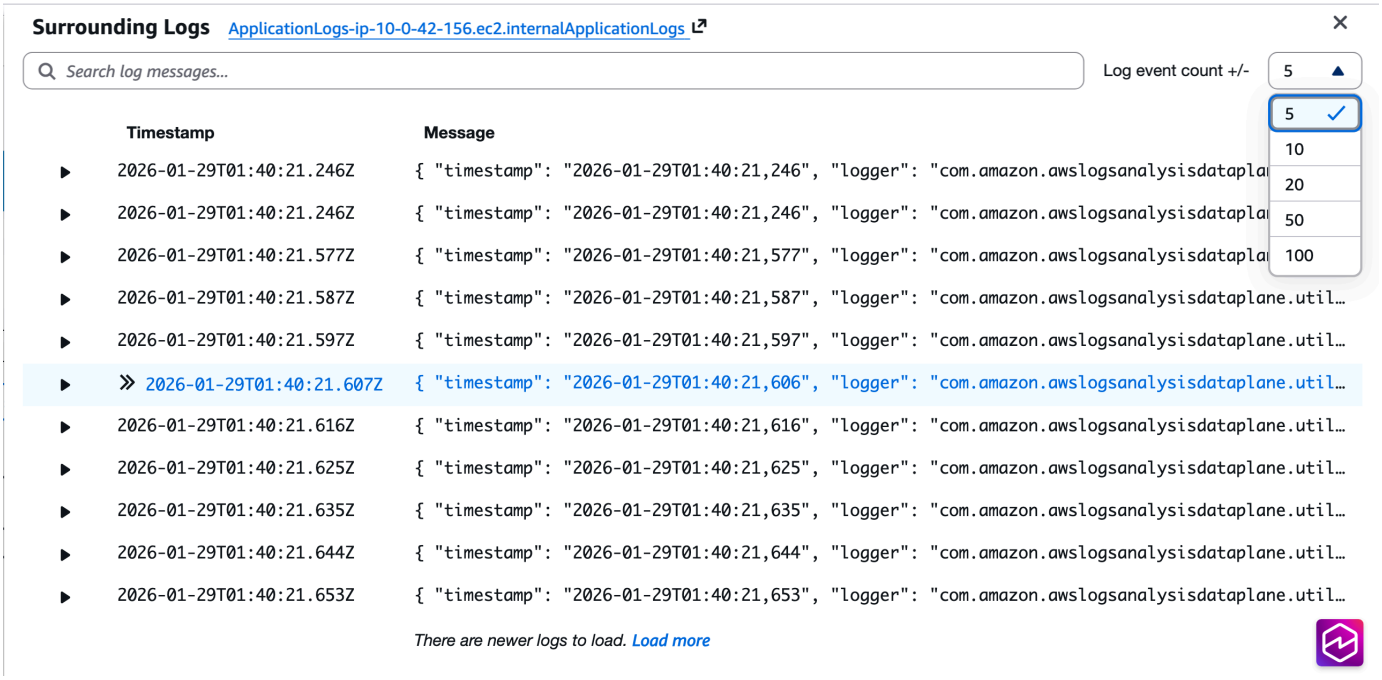
Vea los registros circundantes

Para ver los registros circundantes

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. Ejecute la consulta.
4. En los resultados de la consulta, busque el registro de registro que desee investigar.
5. Elija Registros circundantes.



6. En el menú desplegable Número de eventos, elija el número de eventos de registro que desee mostrar por encima y por debajo del registro seleccionado. Puede elegir 5, 10, 20, 50 o 100.

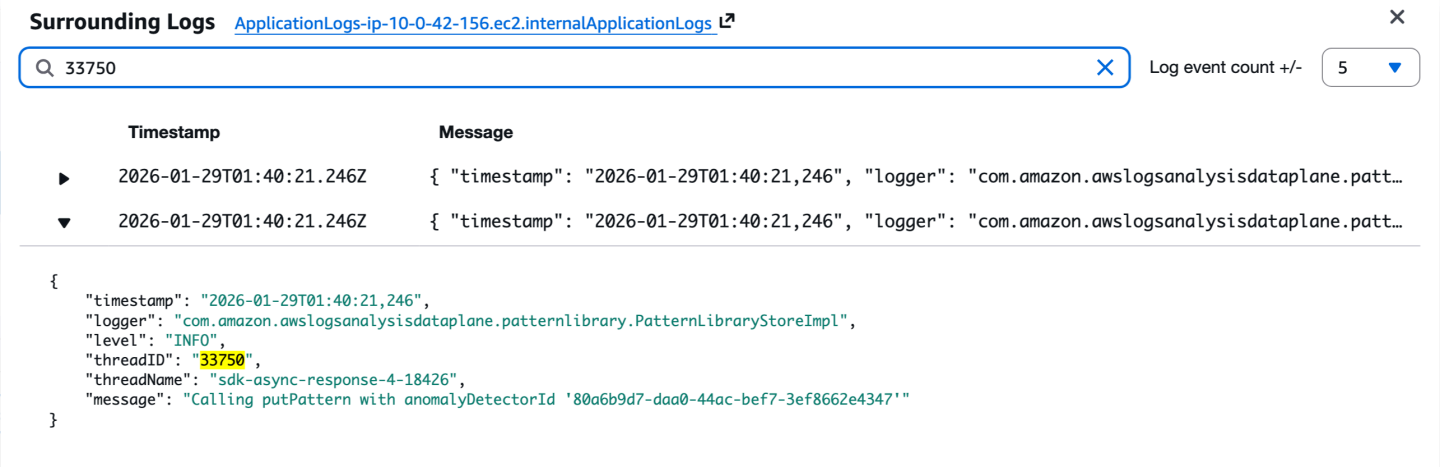


Busque en los registros circundantes

Después de abrir el panel de registros circundante, puedes buscar palabras clave específicas para localizar el contexto relevante.

Para buscar en los registros circundantes

1. En el panel de registros adyacente, introduce el término de búsqueda en el cuadro de búsqueda.
2. Revisa las líneas de registro coincidentes resaltadas.
3. Usa los controles de navegación para moverte entre los partidos.



Análisis del patrón

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 entre los campos de registro. Al ver los resultados de una consulta, puedes elegir la pestaña Patrones para ver los patrones que encontró CloudWatch Logs a partir de una muestra de tus resultados. Como alternativa, puede añadir el comando `pattern` 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.

Estudie 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:

```
<Time-1> [INFO] Calling DynamoDB to store for resource id <ID-2>
```

Los campos dentro de un patrón se denominan tokens. Los campos que varían dentro de un patrón, como un ID de solicitud o una marca de tiempo, son tokens dinámicos. Cada token dinámico está representado por <*string*-*number*>. *string*Es una descripción del tipo de datos que representa el token. *number*Muestra en qué parte del patrón aparece este token, en comparación con los otros tokens dinámicos.

Entre los ejemplos más comunes de tokens dinámicos se incluyen los códigos de error, las marcas de tiempo y los ID 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, entonces 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 en registros y Comparación (diferencia) con intervalos de tiempo anteriores.

Introducción al análisis de patrones

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

Si incluye el comando `pattern` 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 recuerde que los eventos de registro sin procesar no se devuelven cuando se utiliza el comando `pattern`. 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 hizo coincidir 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 Registros y luego, Información de registros.

   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 línea `| limit 20` 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 registro, elija uno o varios grupos de registro que va a consultar.
5. (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.
6. Seleccione Ejecutar consulta para iniciar la consulta.

   Cuando la consulta termina de ejecutarse, la pestaña Registros muestra una tabla con los eventos de registro que ha devuelto la consulta. Encima de la tabla hay un mensaje que indica el número de registros que coinciden con la consulta, similar a Mostrando 10 000 de 71 101 registros coincidentes.
7. Seleccione la pestaña Patrones.
8. La tabla muestra ahora los patrones que se encontraron en la consulta. Como la consulta no incluía el comando `pattern`, en esta pestaña solo se muestran los patrones detectados entre los 10 000 eventos de registro que se muestran en la tabla en la pestaña Registros.

   En cada patrón se muestra la siguiente información:

   - El Patrón, en el que cada token dinámico se muestra como <*string*-*number*>. *string*Es una descripción del tipo de datos que representa el token. *number*Muestra en qué parte del patrón aparece este token, en comparación con los otros tokens dinámicos.
   - El Recuento de eventos, que es el número de veces que el patrón apareció en el registro de eventos consultado. Elija el encabezado de 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.
     - ADVERTENCIA si el patrón contiene la palabra Advertencia, pero no contiene Error.

- INFORMACIÓN si el patrón no contiene las palabras Advertencia ni Error.

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

9. A continuación, cambie la consulta. Sustituya la línea `| sort @timestamp desc` 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 ha incluido `pattern` en la consulta, puede inspeccionar con más detalle los patrones que devuelve la consulta. Para ello, elija el icono de uno de los patrones en la columna Inspeccionar.

Aparecerá el panel 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 veces que aparece el 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 Muestras de registro 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 es que seleccionó uno.

> **Note**
>
> Se captura un máximo de 10 valores de token por 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 está analizando. Por ejemplo, si el patrón de un mensaje de `ERROR` solía ir acompañado de otro evento de registro marcado como `INFO` con detalles adicionales, ese patrón se muestra aquí.

Detalles sobre el comando pattern

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

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

  Para obtener más detalles acerca de la sintaxis `pattern`, consulte pattern.
- Cuando se utiliza `pattern` en una consulta, `@message` debe ser uno de los campos seleccionados en el comando `pattern`.
- Puede incluir el comando `filter` antes de un comando `pattern` 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 comando `parse`, utilice `pattern @fieldname`.
- Las consultas con un resultado que no sea de registro, como las consultas realizadas con el comando `stats`, no devuelven resultados de patrones.

Guarde y vuelva a ejecutar CloudWatch las consultas de 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 mantenerlas organizadas. Puede guardar hasta 1000 consultas por región y cuenta.

Las consultas se guardan en un Region-specific nivel, no en un nivel específico del usuario. Si crea y guarda una consulta, los demás usuarios con acceso a los CloudWatch registros de la misma región podrán ver todas las consultas guardadas y sus estructuras de carpetas en la región.

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`.

> **Note**
>
> Puede crear y guardar consultas con parámetros: plantillas reutilizables con marcadores de posición con nombre. En lugar de guardar varias variantes de la misma consulta con valores diferentes, cree una plantilla y proporcione valores de parámetros diferentes al ejecutarla. Actualmente, esta funcionalidad solo se admite para consultas que utilizan el lenguaje de consultas de Logs Insights. Para obtener más información, consulte Uso de consultas guardadas con parámetros.

Console

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 Save.
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. (Opcional) Para usar parámetros en la consulta, siga estos pasos adicionales:
   a. Añada parámetros a la consulta. Sustituya los valores estáticos por marcadores de posición utilizando la `{{parameter}}` sintaxis (corchetes dobles antes y después del nombre del parámetro).

      Ejemplo: consulta original con valores estáticos:

```
fields @timestamp, @message
| filter level = "Error"
| filter applicationName = "OrderService"
```

Consulta actualizada con parámetros:

```
fields @timestamp, @message
| filter level = {{logLevel}}
| filter applicationName = {{applicationName}}
```

b. Defina los parámetros utilizados en la consulta. Para cada parámetro marcador de posición, especifique:

- Nombre: debe coincidir exactamente con el nombre del marcador de posición (por ejemplo,`logLevel`,`applicationName`).
- Valor predeterminado (opcional): el valor que se utilizará si no se proporciona ningún valor de parámetro.
- Descripción (opcional): explica el propósito del parámetro.

c. Las consultas con parámetros se pueden ejecutar utilizando el nombre de la consulta con un $ prefijo y pasando los nombres de los parámetros como pares clave-valor. Consulte Para ejecutar una consulta guardada para obtener más información.

9. Seleccione Save.

AWS CLI

Para guardar una consulta, utilice`put-query-definition`:

```
aws logs put-query-definition \
  --name "ErrorsByLevel" \
  --query-string "fields @timestamp, @message | filter level = \"ERROR\"" \
  --log-group-names "/aws/lambda/my-function" \
  --region us-east-1
```

(Opcional) Para guardar una consulta con parámetros, añada la `--parameters` opción y utilice `{{parameterName}}` marcadores de posición en la cadena de consulta:

```
aws logs put-query-definition \
  --name "ErrorsByLevel" \
```

```
  --query-string "fields @timestamp, @message | filter level = {{logLevel}} | filter
 applicationName = {{applicationName}}" \
  --parameters '[{"name":"logLevel","defaultValue":"ERROR","description":"Log level
 to filter"},
{"name":"applicationName","defaultValue":"OrderService","description":"Application
 name to filter"}]' \
  --log-group-names "/aws/lambda/my-function" \
  --region us-east-1
```

Para guardar una consulta en una carpeta, añada al nombre de la consulta la ruta de la carpeta:

```
aws logs put-query-definition \
  --name "my-folder/ErrorsByLevel" \
  --query-string "fields @timestamp, @message | filter level = {{logLevel}}" \
  --parameters '[{"name":"logLevel","defaultValue":"ERROR","description":"Log level
 to filter"}]' \
  --log-group-names "/aws/lambda/my-function" \
  --region us-east-1
```

API

Para guardar una consulta, llama a PutQueryDefinition:

```
{
  "name": "ErrorsByLevel",
  "queryString": "fields @timestamp, @message | filter level = \"ERROR\"",
  "logGroupNames": ["/aws/lambda/my-function"]
}
```

(Opcional) Para guardar una consulta con parámetros, incluye el `parameters` campo y utiliza `{{parameterName}}` marcadores de posición en la cadena de consulta:

```
{
  "name": "ErrorsByLevel",
  "queryString": "fields @timestamp, @message | filter level = {{logLevel}} | filter
 applicationName = {{applicationName}}",
  "logGroupNames": ["/aws/lambda/my-function"],
  "parameters": [
    {
      "name": "logLevel",
      "defaultValue": "ERROR",
```

```
      "description": "Log level to filter"
    },
    {
      "name": "applicationName",
      "defaultValue": "OrderService",
      "description": "Application name to filter"
    }
  ]
}
```

> **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.

Console

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 en la lista de consultas guardadas. El texto de la consulta aparece en el editor de consultas.
5. (Opcional) Para usar una consulta con parámetros:
   a. Seleccione el icono + situado junto al nombre de la consulta en el panel lateral de consultas guardadas.
   b. La consulta con los parámetros aparece en el editor de consultas. Por ejemplo, si selecciona el icono + situado junto a`ErrorsByLevel`, el editor de consultas aparecerá con: `$ErrorsByLevel(level=, applicationName=)`

c. Proporcione los valores de los parámetros (level, ApplicationName) y ejecute la consulta. Por ejemplo: `$ErrorsByLevel(level= "ERROR", applicationName= "OrderService")`

6. Seleccione Ejecutar.

AWS CLI

Para ejecutar una consulta guardada con parámetros

`start-query`Utilícela con la `$QueryName()` siguiente sintaxis:

```
aws logs start-query \
  --log-group-names "/aws/lambda/my-function" \
  --start-time 1707566400 --end-time 1707570000 \
  --query-string '$ErrorsByLevel(level= "ERROR", applicationName= "OrderService")' \
  --region us-east-1
```

API

Para ejecutar una consulta guardada con parámetros

Llame StartQuerycon la `$QueryName()` sintaxis del `queryString` campo:

```
{
  "logGroupNames": ["/aws/lambda/my-function"],
  "startTime": 1707566400,
  "endTime": 1707570000,
  "queryString": "$ErrorsByLevel(level=\"ERROR\", applicationName= \"OrderService
\")"
}
```

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 Save.

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).

Uso de consultas guardadas con parámetros

Las consultas guardadas con parámetros son plantillas de consulta reutilizables con marcadores de posición con nombre. En lugar de mantener varias copias de consultas prácticamente idénticas, puede guardar una plantilla y proporcionar valores de parámetros diferentes al ejecutar la consulta. Los parámetros solo se admiten en el lenguaje de consultas de CloudWatch Logs Insights.

Cómo funciona

Al guardar una consulta, los marcadores de posición identifican los valores que puede proporcionar en el momento de la ejecución de la consulta. Los marcadores de posición utilizan la `{{parameterName}}` sintaxis. A continuación se muestra un ejemplo de una consulta guardada denominada `ErrorsByLevel` con dos parámetros `logLevel` y`applicationName`.

```
fields @timestamp, @message
| filter level = {{logLevel}}
| filter applicationName = {{applicationName}}
```

Para ejecutar una consulta guardada, puede invocarla utilizando el nombre de la consulta con el prefijo $ y pasando los valores de los parámetros. El motor de consultas CloudWatch de Logs Insights reemplaza cada marcador de posición. Si un parámetro contiene valores predeterminados, esos valores se utilizan si no se proporcionan otros valores.

```
# Run query by using query name and passing parameter values explicitly
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService")

# Run query without specifying parameter values - default values are used in this case.
$ErrorsByLevel()
```

Los nombres de consulta guardados que contengan espacios o caracteres especiales deben estar entre comillas invertidas:

```
$`Errors By Level`(logLevel = "WARN")
```

Ejemplos de consultas guardadas con parámetros

Añadir un límite de resultados como parámetro

Nombre de consulta: `ErrorsByLevel` con parámetros `logLevel` (predeterminado:`"ERROR"`), `applicationName` (predeterminado:`"OrderService"`) y `maxResults` (predeterminado:`50`)

```
fields @timestamp, @message, @logStream
| filter level = {{logLevel}}
| filter applicationName = {{applicationName}}
| sort @timestamp desc
| limit {{maxResults}}
```

```
# Run the query using the query name and passing parameter values
```

```
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService", maxResults = 100)
```

Uso de varias consultas guardadas con parámetros

En el ejemplo siguiente se utiliza `ErrorsByLevel` una segunda consulta guardada `RecentN` que se define como `sort @timestamp desc | limit {{count}}` (con parámetro`count`, predeterminado`20`). El motor de consultas de CloudWatch Logs Insights expande cada consulta antes de ejecutarla.

```
# Using multiple queries with parameters in sequence
$ErrorsByLevel(logLevel = "WARN", applicationName = "OrderService")
| $RecentN(count = 10)

# Each of the queries is expanded, resulting in the following query when it is run.
fields @timestamp, @message
| filter level = "WARN"
| filter applicationName = "OrderService"
| sort @timestamp desc
| limit 10
```

Cuotas y gestión de errores

> **Note**
>
> Cada consulta guardada puede tener un máximo de 20 parámetros.

La cadena de consulta expandida no puede superar los 10 000 caracteres. Los nombres de los parámetros deben empezar por una letra o un carácter de subrayado. Una consulta guardada no puede hacer referencia a otra consulta guardada (no se admiten las invocaciones anidadas).

Errores comunes

Error	Causa
Los parámetros solo se admiten en el lenguaje de consultas CWLI	Los parámetros solo se admiten en el lenguaje de consultas CloudWatch de Logs Insights.

Error	Causa
No se encontraron los parámetros necesarios en QueryString	El nombre de un parámetro no `--parameters` coincide con el nombre de un parámetro `{{placeholder}}` en la cadena de consulta.
El recuento de parámetros supera el máximo de 20	Actualmente, las consultas guardadas solo admiten 20 parámetros.
Nombre de parámetro duplicado	La definición de consulta contiene parámetros duplicado `sparameters` .

> **Note**
>
> Para crear o actualizar una consulta guardada con parámetros, necesita el `logs:PutQueryDefinition` permiso. Para ejecutar una, necesita `logs:StartQuery` y`logs:DescribeQueryDefinitions`.

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 100 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 registro 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 registro 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 100 consultas simultáneas de CloudWatch Logs Insights por cuenta, incluidas las consultas añadidas a los paneles de control. Además, puede ejecutar 15 consultas simultáneas para OpenSearch Service PPL o Service SQL. OpenSearch

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 la clave de KMS como `policy.json` con el siguiente comando get-key-policy:

```
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.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333: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:us-east-1:111122223333:query-
result:*"
                },
                "StringEquals": {
                "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Por último, agregue la política actualizada con el siguiente comando put-key-policy:

```
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 de KMS asociada a los resultados de las consultas, utilice el siguiente comando disassociate-kms-key:

```
aws logs disassociate-kms-key --resource-identifier "arn:aws:logs:region:account-
id:query-result:*"
```

Genera un resumen en lenguaje natural a partir de CloudWatch los resultados de la consulta de Logs Insights

Analizar los datos de registro es fundamental para comprender el comportamiento de las aplicaciones, pero interpretar grandes volúmenes de entradas de registro puede llevar mucho tiempo. CloudWatch Logs Insights ahora ofrece una función de resumen en lenguaje natural que transforma los resultados de consultas complejas en resúmenes claros y concisos. Esta capacidad ayuda a identificar rápido los problemas y a obtener información útil a partir de los datos de registro.

Funcionamiento

CloudWatch Logs Insights puede generar un resumen legible para las personas a partir de los resultados de su consulta mediante Amazon Bedrock. La función es compatible con todos los lenguajes de consulta de CloudWatch Logs Insights y proporciona información clara y procesable a partir de sus datos de registro.

Disponibilidad regional y procesamiento de datos

> **Important**
>
> Al utilizar esta característica, es posible que los resultados de la consulta se procesen en una Región de AWS diferente. Por ejemplo, si se ejecuta una consulta en Este de EE. UU. (Norte de Virginia), el resumen podría tener lugar en Oeste de EE. UU. (Oregón).

En la siguiente tabla se muestra el posible procesamiento Región de AWS para las diferentes geografías en las que está disponible la función de resultados de la consulta:

Geografía de CloudWatch registros admitida	Posible región de procesamiento
Estados Unidos (EE. UU.)	Región Este de EE. UU. (Norte de Virginia) US East (Ohio) Region Región oeste de EE. UU (Oregón)
Europa	Europe (Frankfurt) Region

Geografía de CloudWatch registros admitida	Posible región de procesamiento
	Región de Europa (Irlanda) Región de Europa (París) Región Europa (Estocolmo) Región de Europa (Londres)
Asia Pacífico	Región Este de EE. UU. (Norte de Virginia) US East (Ohio) Region Región oeste de EE. UU (Oregón)
América del Sur	Región Este de EE. UU. (Norte de Virginia) US East (Ohio) Region Región oeste de EE. UU (Oregón)

Introducción

Creación de un resumen en lenguaje natural

1. Ejecute su consulta CloudWatch de Logs Insights.
2. Una vez completada la consulta, seleccione Resumen de los resultados.

Permisos

Se debe disponer de una de las siguientes:

- Permiso de `CloudWatchLogsFullAccess`
- Permiso de `CloudWatchLogsReadOnlyAccess`
- Política de IAM personalizada que incluye las acciones `cloudwatch:GenerateQueryResultsSummary`, `logs:GetQueryResults`, `logs:DescribeQueries` y `logs:FilterLogEvents`

Privacidad de datos

Los resultados de su consulta se procesan de forma segura y no se utilizan para entrenar o mejorar CloudWatch Logs Insights o Amazon Bedrock. Si opta por enviar comentarios sobre el resumen de los resultados de la consulta mediante los botones de comentarios, sus comentarios indican su nivel de satisfacción con la capacidad proporcionada en CloudWatch Logs Insights.

Automatizar el análisis de registros con consultas programadas

Las consultas programadas le permiten automatizar la ejecución de las consultas de CloudWatch Logs Insights de forma regular. En lugar de ejecutar consultas manualmente para analizar los datos de registro, puede configurar las consultas programadas para que se ejecuten automáticamente y entreguen los resultados a destinos como los buckets de Amazon S3 o los buses de EventBridge eventos de Amazon. Esta automatización es ideal para generar informes periódicos, monitorear tendencias o activar procesos posteriores en función de los resultados del análisis de registros.

Las consultas programadas son compatibles con los tres lenguajes de consulta disponibles en CloudWatch Logs Insights:

- Lenguaje de consulta de Logs Insights (Logs Insights QL)
- OpenSearch Lenguaje de procesamiento por canalización de servicios (PPL)
- OpenSearch Lenguaje de consulta estructurado de servicios (SQL)

Contenido

Comprensión de los conceptos de consultas programadas

Antes de crear consultas programadas, comprenda estos conceptos clave que afectan a la forma en que se ejecutan las consultas y a dónde se entregan los resultados.

Separación de funciones de IAM

Las consultas programadas requieren dos funciones de IAM independientes: una para ejecutar las consultas y otra para entregar los resultados a Amazon S3 o Amazon EventBridge Event Buses.

Entender por qué existe esta separación le ayuda a configurar los permisos correctamente y a utilizar las ventajas operativas y de seguridad que proporciona.

La arquitectura de dos funciones divide las responsabilidades entre el acceso a los datos y la entrega de los datos. La función de ejecución de consultas accede a los datos de registro y ejecuta las consultas, mientras que la función de entrega de destino escribe los resultados en el destino elegido. Esta separación sigue el principio del privilegio mínimo: cada función solo tiene los permisos que necesita para su función específica.

Función de ejecución de consultas

Permite a CloudWatch Logs ejecutar consultas CloudWatch de Logs Insights en su nombre. Esta función necesita permisos para acceder a sus grupos de registros y ejecutar consultas, pero no necesita acceder a los recursos de destino. Permisos necesarios:

- `logs:StartQuery`
- `logs:StopQuery`
- `logs:GetQueryResults`
- `logs:DescribeLogGroups`
- `logs:Unmask`si es necesario desenmascarar los datos

Para grupos de registros cifrados con KMS: `kms:Decrypt` y `kms:DescribeKey` permisos para la clave de KMS utilizada para cifrar los grupos de registros. También es necesario añadir estos permisos.

Requisito de relación de confianza: la función de ejecución de consultas debe incluir una política de confianza que permita que el servicio de CloudWatch registros (`logs.amazonaws.com`) asuma la función. Sin esta relación de confianza, las consultas programadas fallarán y generarán errores de permiso.

Ejemplo de política de confianza para la función de ejecución de consultas:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.amazonaws.com"
            },
```

```
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Ejemplo de política de permisos para la función de ejecución de consultas:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:GetQueryResults",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        }
    ]
}
```

Función de entrega en destino

Permite que CloudWatch Logs entregue los resultados de las consultas al destino elegido. Esta función solo necesita permisos para el servicio de destino específico, siguiendo el principio de privilegios mínimos. Los permisos necesarios varían según el tipo de destino.

Requisito de relación de confianza: la función de entrega de destino también debe incluir una política de confianza que permita al servicio de CloudWatch registros (`logs.amazonaws.com`) asumir la función.

Ejemplo de política de permisos para la función de entrega en destino de S3:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
```

```
            ],
            "Resource": "arn:aws:s3:::your-scheduled-query-results-bucket/*"
        }
    ]
}
```

Esta separación proporciona beneficios prácticos para sus operaciones. Desde el punto de vista de la seguridad, si necesita cambiar el lugar donde se entregan los resultados, solo debe modificar la función de entrega de destino sin cambiar los permisos de ejecución de las consultas. Para garantizar el cumplimiento y la auditoría, puede realizar un seguimiento claro de qué función accede a los datos de registro confidenciales y qué función escribe en sistemas externos. Esto facilita la demostración de que su infraestructura de análisis de registros sigue las mejores prácticas de seguridad.

Uso entre regiones y cuentas

Una consulta programada se crea en una región específica y se ejecuta en esa región. Sin embargo, puede consultar grupos de registros y ofrecer resultados en todas las regiones y cuentas. Debe configurar una o más AWS cuentas como cuentas de supervisión y vincularlas con varias cuentas de origen. Una cuenta de monitoreo es una AWS cuenta central que puede ver e interactuar con los datos de observabilidad generados a partir de las cuentas de origen. Una cuenta de origen es una AWS cuenta individual que genera datos de observabilidad para los recursos que residen en ella. Las cuentas de origen comparten sus datos de observabilidad con la cuenta de supervisión. Por lo tanto, puede configurar consultas programadas desde la cuenta de monitoreo utilizando los grupos de registros de todas las cuentas vinculadas.

Consultando grupos de registros entre regiones

La consulta programada puede acceder a los grupos de registros de cualquier región. Especifique los grupos de registros con su formato ARN completo:. `arn:aws:logs:region:account-id:log-group:log-group-name` La función de ejecución de consultas necesita `logs:StartQuery` y `logs:GetQueryResults` permisos para los grupos de registros en todas las regiones de destino.

> **Important**
>
> Al consultar grupos de registros o entregar resultados en todas las regiones, los datos de registro cruzan los límites regionales. Considere lo siguiente:

- Requisitos de residencia de los datos: asegúrese de que la transferencia de datos entre regiones cumpla con las políticas de gobierno de datos y los requisitos reglamentarios de su organización
- Costes de transferencia de datos: la transferencia de datos entre regiones conlleva cargos adicionales
- Latencia de red: las consultas que acceden a grupos de registros en regiones distantes pueden experimentar una latencia más alta

Para obtener un rendimiento y una rentabilidad óptimos, cree consultas programadas en la misma región que sus grupos de registros principales.

Un enfoque alternativo: utilice la centralización de CloudWatch registros para replicar los datos de registro de varias cuentas y regiones en una cuenta de supervisión central. Esto le permite crear consultas programadas en una sola región que accedan a todos sus registros centralizados, lo que evita las consultas entre regiones y simplifica la administración de los permisos de IAM.

Programe las expresiones y la gestión de las zonas horarias

La programación que defina determina cuándo se ejecuta la consulta y con qué frecuencia. La elección de la expresión de programación correcta afecta al momento en que se reciben los resultados y a la cantidad de datos que se consultan. Comprender los tipos de expresión le ayuda a elegir entre simplicidad y precisión.

Las expresiones cron proporcionan un control preciso sobre el tiempo, lo que le permite especificar horas, días de la semana o días del mes exactos. Utilice expresiones cron cuando necesite que las consultas se ejecuten en un horario laboral específico o que se ajusten a los cronogramas operativos. En la consola, también puede programar consultas mediante sencillas opciones de calendario.

Expresiones cron

Ejecute consultas en momentos específicos. Formato: `cron(minute hour day-of-month month day-of-week year)`. Ejemplos:

- `cron(0 9 * * ? *)`- Todos los días a las 9:00 a.m. UTC
- `cron(0 18 ? * MON-FRI *)`- De lunes a viernes a las 18:00 UTC

- `cron(0 0 1 * ? *)`- El primer día de cada mes a medianoche UTC
- `cron(0 12 ? * SUN *)`- Todos los domingos a las 12:00, hora peninsular española
- `cron(30 8 1 1 ? *)`- El 1 de enero a las 8:30 UTC

Todas las consultas programadas se ejecutan en UTC, independientemente de la zona horaria local o de la ubicación de AWS los recursos. Esto es especialmente importante cuando se programan consultas para el horario laboral o para análisis urgentes. Por ejemplo, si su empresa opera en la hora del este de EE. UU. y desea obtener un informe diario a las 9 a. m. ET, debe tener en cuenta la diferencia UTC (14:00 UTC durante el horario de verano, 13:00 UTC en caso contrario). Planifica tus horarios teniendo en cuenta la hora UTC para asegurarte de que las consultas se ejecuten a las horas previstas.

Elegir un idioma de consulta

Las consultas programadas admiten tres lenguajes de consulta diferentes, y tu elección afecta tanto a la forma en que escribes las consultas como a la facilidad con la que tu equipo puede mantenerlas. El idioma correcto depende de tus requisitos de análisis y de las habilidades actuales de tu equipo.

Si está filtrando y agregando principalmente datos de registro, CloudWatch Logs Insights Query Language ofrece la sintaxis más sencilla. Para las transformaciones de datos complejas en las que es necesario remodelar o enriquecer los datos mediante varios pasos, el enfoque de flujo continuo de PPL facilita el seguimiento de la lógica. Cuando necesite realizar uniones o agregaciones complejas similares a las operaciones de bases de datos, SQL proporciona una sintaxis familiar que los equipos con experiencia en bases de datos pueden adoptar rápidamente.

CloudWatch El lenguaje de consultas de Logs Insights (CWLI)

Diseñado específicamente para el análisis de registros con una sintaxis intuitiva. Ideal para:

- Análisis y filtrado de registros basados en texto
- Agregaciones y estadísticas de series temporales
- Equipos nuevos en el análisis de registros

OpenSearch Lenguaje de procesamiento canalizado (PPL) de Service

Lenguaje de consultas basado en Pipeline con potentes capacidades de transformación de datos. Ideal para:

- Transformaciones y enriquecimiento de datos complejos

- Flujos de trabajo de procesamiento de datos en varios pasos
- Equipos familiarizados con el procesamiento basado en canalizaciones

OpenSearch Lenguaje de consulta estructurado (SQL) de servicios

Sintaxis SQL estándar para consultas conocidas al estilo de una base de datos. Ideal para:

- Uniones y agregaciones complejas
- Inteligencia empresarial e informes
- Equipos con una sólida experiencia en SQL

Selección de destinos y casos de uso

El lugar donde se envían los resultados de las consultas determina lo que se puede hacer con ellos. Esta elección da forma a todo tu flujo de trabajo posterior, ya sea que estés creando análisis a largo plazo, activando respuestas automatizadas o ambas cosas. Comprender los puntos fuertes de cada tipo de destino le ayuda a diseñar la arquitectura adecuada para su caso de uso.

Los destinos de Amazon S3 están optimizados para el almacenamiento y el procesamiento por lotes. Cuando necesite conservar los resultados de las consultas durante meses o años, analizar tendencias a lo largo del tiempo o introducir datos en plataformas de análisis, Amazon S3 ofrece almacenamiento rentable con retención ilimitada. EventBridge los destinos están optimizados para la automatización en tiempo real. Cuando los resultados de las consultas deberían activar acciones inmediatas (como enviar alertas, iniciar flujos de trabajo o actualizar sistemas), EventBridge ofrece resultados en forma de eventos a los que sus aplicaciones pueden responder al instante. De forma predeterminada, todos los eventos de finalización de consultas se envían automáticamente como eventos al bus de eventos predeterminado, lo que permite la integración con sistemas de procesamiento descendente, funciones Lambda u otras arquitecturas basadas en eventos. Los resultados solo se publican en los destinos cuando la consulta se ejecuta correctamente.

Destinos de Amazon S3

Almacene los resultados de las consultas como archivos JSON para conservarlos a largo plazo y procesarlos por lotes. Ideal para:

- Análisis histórico y archivado de datos
- Integración con lagos de datos y plataformas de análisis
- Requisitos de conformidad y auditoría
- Almacenamiento rentable de conjuntos de resultados de gran tamaño

EventBridge destinos

Envíe los resultados de las consultas como eventos para su procesamiento y automatización en tiempo real. Puedes recuperar los resultados de la consulta únicamente con el queryID enviado en el caso de que dure hasta 30 días, ya que almacenamos los resultados durante 30 días. Ideal para:

- Activar respuestas automatizadas a los resultados de las consultas
- Integración con flujos de trabajo sin servidor y funciones Lambda
- Sistemas de alertas y notificaciones en tiempo real
- Arquitecturas y microservicios basados en eventos

Formato y estructura de los resultados de la consulta

Para los destinos de Amazon S3, los resultados de las consultas se entregan en formato JSON con la misma estructura que la respuesta de la GetQueryResults API. Para Amazon, EventBridge comprender el formato de los resultados de las consultas programadas le ayuda a diseñar flujos de trabajo de procesamiento e integración posteriores.

Los resultados de las consultas se entregan en formato JSON con la siguiente estructura:

```
{
    "version": "0",
    "id": "be72061b-eca2-e068-a7e1-83e01d6fe807",
    "detail-type": "Scheduled Query Completed",
    "source": "aws.logs",
    "account": "123456789012",
    "time": "2025-11-18T11:31:48Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:logs:us-east-1:123456789012:scheduled-query:477b4380-b098-474e-9c5e-
e10a8cc2e6e7"
    ],
    "detail": {
        "queryId": "2038fd57-ab4f-4018-bb2f-61d363f4a004",
        "queryString": "fields @timestamp, @message, @logStream, @log\n| sort
 @timestamp desc\n| limit 10000",
        "logGroupIdentifiers": [
            "/aws/lambda/my-function"
        ],
        "status": "Complete",
```

```
        "startTime": 1763465460,
        "statistics": {
            "recordsMatched": 0,
            "recordsScanned": 0,
            "estimatedRecordsSkipped": 0,
            "bytesScanned": 0,
            "estimatedBytesSkipped": 0,
            "logGroupsScanned": 1
        }
    }
}
```

Los elementos clave incluyen:

- `statistics`- Métricas de rendimiento de consultas, incluidos los registros coincidentes, escaneados, los bytes procesados y los datos omitidos estimados
- `startTime`- Cuándo se inició la ejecución de la consulta (marca de tiempo de Unix)
- `queryString`- La consulta real que se ejecutó
- `queryId`- Identificador de consulta de la consulta mediante el cual se pueden recuperar los resultados
- `logGroupIdentifiers`- Lista de grupos de registros consultados
- `status`- Estado de ejecución de la consulta (completa, fallida, etc.)

Referencia de expresión de horario

Utilice estas tablas de referencia para crear expresiones de programación para sus consultas programadas. Todas las horas se indican en UTC.

Sintaxis de expresiones cron

Formato: `cron(minute hour day-of-month month day-of-week year)`

Caso de uso	Expresión Cron	Description (Descripción)	Úselo cuando
Horarios diarios	`cron(0 9 * * ? *)`	Todos los días a las 9:00 a.m. UTC	Informes diarios

Caso de uso	Expresión Cron	Description (Descripción)	Úselo cuando
	`cron(0 */6 * * ? *)`	Cada 6 horas (00:00, 06:00, 12:00, 18:00 UTC)	Monitorización frecuente
	`cron(30 2 * * ? *)`	Todos los días a las 2:30 a.m. UTC	Análisis fuera de las horas pico
Horario de atención	`cron(0 9-17 ? * MON-FRI *)`	Cada hora de 9 a.m. a 5 p.m., de lunes a viernes UTC	Supervisión empresarial
	`cron(0 18 ? * MON-FRI *)`	De lunes a viernes a las 18:00 UTC	Fin del día laborable
	`cron(0 8,12,17 ? * MON-FRI *)`	De lunes a viernes, a las 8 a. m. al mediodía y a las 5 p. m., UTC	Horarios comerciales clave
Horarios semanales	`cron(0 12 ? * SUN *)`	Todos los domingos a las 12:00, hora peninsular española	Resúmenes semanales
	`cron(0 9 ? * MON *)`	Todos los lunes a las 9:00 a.m. UTC	Informes de inicio de semana
	`cron(0 23 ? * FRI *)`	Todos los viernes a las 23:00 UTC	Limpieza de fin de semana
Horarios mensuales	`cron(0 0 1 * ? *)`	El primer día de cada mes a medianoche UTC	Informes mensuales

Caso de uso	Expresión Cron	Description (Descripción)	Úselo cuando
	`cron(0 9 L * ? *)`	Último día de cada mes a las 9:00 a.m. UTC	Procesamiento de fin de mes
	`cron(0 10 1 1,4,7,10 ? *)`	El primer día de cada trimestre a las 10:00 a.m. UTC	Análisis trimestral
Alta frecuencia	`cron(*/15 * * * ? *)`	Cada 15 minutos	Supervisión en tiempo real
	`cron(0,30 * * * ? *)`	Cada 30 minutos (a las :00 y a las :30)	Controles frecuentes
	`cron(0 */2 * * ? *)`	Cada 2 horas	Intervalos regulares
Casos especiales	`cron(30 8 1 1 ? *)`	1 de enero a las 8:30 a.m. UTC	Informes anuales
	`cron(0 6 * * SAT,SUN *)`	Los fines de semana a las 6:00 a.m. UTC	Procesamiento durante el fin
	`cron(0 0 ? * MON#1 *)`	El primer lunes de cada mes a medianoche (UTC)	Planificación mensual

Referencia de campo de expresión cron

Campo	Valores	Caracteres comodín	Ejemplos
Minuto (primero)	0-59	* , - /	0(al final de la hora), */15 (cada 15 minutos), 0,30 (dos veces por hora)
Hora (segunda)	0-23	* , - /	9(9 a.m.), */2 (cada 2 horas), 9-17 (horario comercial)
Day-of-month (3º)	1-31, LARGO, ANCHO	* , - / ?	1(primer día), L (último día), ? (cuando se usa day-of-week)
Mes (cuarto)	1-12 o JAN-DEC	* , - /	1(enero)JAN, 1,4,7,10 (trimestral)
Day-of-week (5)	1-7 o SUN-SAT	* , - / ? # L	MON-FRI(de lunes a viernes)SUN, MON#1 (primer lunes)
Año (6º)	1970-2199	* , - /	*(cada año), 2024 (año específico), 2024-2026 (rango)

Caracteres comodín y expresiones especiales

***** (asterisco)

Coincide con todos los valores del campo. Ejemplo: * en el campo hora significa cada hora.

?(signo de interrogación)

Sin valor específico. Úselo en day-of-month o day-of-week cuando se especifique el otro. Ejemplo: utilice ? in day-of-month al especificar MON-FRI in day-of-week.

- (guion)

Rango de valores. Ejemplo: MON-FRI (de lunes a viernes), 9-17 (de 9 a. m. a 5 p. m.).

,(coma)

Múltiples valores específicos. Ejemplo: `MON,WED,FRI` (lunes, miércoles y viernes), `8,12,17` (8 a. m., mediodía, 5 p. m.).

/(barra)

Valores o incrementos de los pasos. Ejemplo: `0/15` en minutos significa cada 15 minutos a partir del minuto 0 (0, 15, 30, 45). `*/2`en horas significa cada 2 horas.

L(último)

Último día del mes o último día de la semana. Ejemplo: `L` in day-of-month significa el último día del mes. `FRIL`significa el último viernes del mes.

W(día laborable)

Día laborable más cercano. Ejemplo: `15W` significa el día de la semana más cercano al día 15 del mes.

#(enésima aparición)

Enésima aparición de un día laborable del mes. Ejemplo: `MON#1` significa el primer lunes del mes, `FRI#2` significa el segundo viernes del mes.

Patrones comunes y mejores prácticas

- Para aplicaciones empresariales: utilice el horario `MON-FRI` laboral (p. ej.`9-17`) para evitar realizar consultas durante los fines de semana o fuera del horario laboral.
- Para el monitoreo de alta frecuencia: utilice incrementos como `*/15` (cada 15 minutos), pero tenga en cuenta los límites de simultaneidad de consultas.
- Para ahorrar recursos: programe consultas con un uso intensivo de recursos durante las horas de menor actividad y utilice horas tempranas de la mañana, como UTC. `2-6`
- Para los informes mensuales: **L** utilízalos para el último día del mes o para fechas específicas, como el primer día, `1` para garantizar una programación uniforme.

Prácticas recomendadas

Siga estas prácticas recomendadas para garantizar que las operaciones de consulta programadas sean fiables y eficientes:

Optimización de las consultas

- Pruebe las consultas manualmente antes de programarlas para verificar el rendimiento y los resultados
- Utilice índices de filtro al principio de la consulta para reducir el procesamiento de datos
- Limite los intervalos de tiempo para evitar tiempos de espera con grupos de registros de gran volumen
- Tenga en cuenta la complejidad de las consultas y los límites de tiempo de ejecución

Planifique la planificación

- Evite la superposición de ejecuciones asegurándose de que las consultas se completen antes de la próxima ejecución programada
- Tenga en cuenta los retrasos en la ingesta de registros al configurar los intervalos de tiempo
- Usa expresiones cron para momentos específicos
- Distribuya los cronogramas para asegurarse de no alcanzar el límite de simultaneidad de consultas

Supervisión y mantenimiento

- Supervise el historial de ejecución con regularidad para identificar fallos o problemas de rendimiento
- Revise y actualice las funciones de IAM periódicamente para mantener la seguridad
- Pruebe la accesibilidad del destino antes de implementarlo en producción

Autorización

- Todas APIs las consultas programadas se autorizan en el recurso de consulta programado y no en los recursos que ocupan en la entrada, como los grupos de registros. Configure las políticas de IAM en consecuencia
- Gestione la autorización de los grupos de registros mediante la función de ejecución asignada en el APIs

Cómo empezar con las consultas programadas

Al crear una consulta programada, configurará varios componentes clave que definen cómo se ejecuta la consulta y dónde se entregan los resultados. Comprender estos componentes le ayudará a configurar un análisis de registro automatizado eficaz.

Cada consulta programada consta de los siguientes componentes clave:

Configuración de consultas

La cadena de consulta de CloudWatch Logs Insights, los grupos de registros de destino y el lenguaje de consulta que se utilizarán en el análisis.

Expresión de programación

Una expresión cron o un calendario de frecuencias que define cuándo se ejecuta la consulta. Puede especificar la configuración de la zona horaria para garantizar que las consultas se ejecuten a la hora local correcta. La consola muestra una descripción legible de su programación, como «Ejecute la consulta todos los martes a las 15:10 durante un intervalo de tiempo de 5 minutos, con efecto inmediato, en UTC, hasta tiempo indefinido».

Intervalo de tiempo

El período retrospectivo de cada ejecución de una consulta, definido por una diferencia entre la hora de inicio y la hora de ejecución. Esto determina la cantidad de datos históricos que analizará cada ejecución de consulta.

Vista previa del programa de ejecución

La consola muestra las tres siguientes ejecuciones de consultas programadas con fechas y horas exactas (por ejemplo, 28-10-2025, 15:10, UTC; 2025/11/04 15:10, UTC; 2025/11/11/2011 15:10, UTC), lo que le ayuda a comprobar que la programación está configurada correctamente.

Destinos

Dónde se muestran los resultados de la consulta tras una ejecución correcta. Los destinos compatibles incluyen los buckets de Amazon S3 y, de forma predeterminada, los metadatos de los resultados se envían al bus de eventos predeterminado.

Rol de ejecución

Una función de IAM que CloudWatch Logs asume para ejecutar la consulta y entregar los resultados a los destinos especificados.

Antes de crear consultas programadas, asegúrese de tener configurados los permisos y los recursos necesarios.

Crear una consulta programada

Cree una consulta programada que ejecute automáticamente las consultas de CloudWatch Logs Insights y entregue los resultados a los destinos que elija.

Requisitos previos

Antes de crear una consulta programada, asegúrese de tener lo siguiente:

- Grupos de registros: uno o más grupos de registros que contienen los datos que desea analizar
- Función de IAM de ejecución: función de IAM con los siguientes permisos:
  - `logs:StartQuery`- Permiso para iniciar consultas de CloudWatch Logs Insights
  - `logs:GetQueryResults`- Permiso para recuperar los resultados de la consulta
  - `logs:DescribeLogGroups`- Permiso para acceder a la información del grupo de registros. Esto solo es necesario para los grupos de registros basados en prefijos para la detección de grupos de registros
- Permisos de destino: permisos de IAM adicionales para el destino elegido:
  - Para los destinos de Amazon S3: `s3:PutObject`
- Para el AWS CLI uso de la API: AWS credenciales configuradas con permisos para llamar a CloudWatch Logs APIs

Para ver ejemplos detallados de políticas de IAM, consulteAdministración de identidades y accesos para Amazon CloudWatch Logs. También debe tenerse en cuenta que solo puede tener 1000 consultas programadas por cuenta.

Console

Para crear una consulta programada (consola)

1. ¿Abrir la consola CloudWatch de Logs en https://us-east-1.console.aws.amazon.com/cloudwatch/casa? region=us-east-1 #LogsV2:Logs-Insights.
2. En el panel de navegación, selecciona Logs Insights.
3. Seleccione Crear consulta programada.
4. En la sección de definición de consultas:
   a. En Idioma de consulta, elija el idioma de consulta que desee utilizar en la lista.
   b. Para la cadena de consulta, introduzca su consulta de CloudWatch Logs Insights en el cuadro.
   c. Para los grupos de registros, seleccione los grupos de registros que desee consultar en la lista.

5. En la sección de configuración del cronograma:

   a. En la expresión Schedule, configure cuándo se ejecutará la consulta. Elija una de las opciones predefinidas o introduzca una expresión cron personalizada.

   b. En Efectivo al crearse, especifique cuándo se activa la programación. Elija comenzar inmediatamente o en una fecha y hora específicas utilizando el YYYY/MM/DD formato.

   c. En Intervalo de tiempo, especifique el período retrospectivo para cada ejecución de consulta. Introduzca la duración en minutos que define el tiempo transcurrido desde el momento de ejecución de la consulta.

   d. En Continuar indefinidamente, especifique cuándo finaliza la programación. Elija ejecutar indefinidamente o hasta una fecha y hora específicas utilizando YYYY/MM/DD el formato.

6. La consola muestra las tres siguientes ejecuciones de consultas programadas en función de su configuración y muestra las fechas y horas exactas en UTC en las que se ejecutará la consulta.

7. En la sección Publicar los resultados de la consulta en S3 (opcional) (si utilizas el destino S3):

   a. Para el bucket de S3, selecciona Esta cuenta si el bucket de destino está en la misma AWS cuenta, o selecciona Otra cuenta si el bucket está en una AWS cuenta diferente e introduce el ID de cuenta de la cuenta propietaria del bucket.

   b. Para el URI de Amazon S3, introduzca el bucket y el prefijo de Amazon S3 donde se almacenarán los resultados (por ejemplo,`s3://my-bucket/query-results/`). Si seleccionó Esta cuenta, puede elegir Browse Amazon S3 para navegar y seleccionar una ubicación de Amazon S3 existente.

   c. (Opcional) Para el ARN de la clave KMS, introduzca el ARN de una AWS KMS clave administrada por el cliente para cifrar los resultados de la consulta mediante SSE-KMS. La clave debe estar en la misma AWS región que el bucket de Amazon S3 de destino.

8. En la sección Función de IAM para publicar los resultados de las consultas en Amazon S3, elija una de las siguientes opciones:

   a. Elija Crear automáticamente un nuevo rol con los permisos predeterminados para configurar automáticamente un rol de IAM con los permisos necesarios para que CloudWatch Logs entregue los resultados de las consultas a Amazon S3.

   b. Elija Utilizar un rol existente para seleccionar un rol de IAM existente con las políticas necesarias para que CloudWatch Logs entregue los resultados de las consultas a

Amazon S3. Utilice el campo de búsqueda para buscar y seleccionar el rol de IAM adecuado de la lista.

9. En la sección Función de IAM para la ejecución programada de consultas, elija una de las siguientes opciones:

   a. Seleccione Crear automáticamente un nuevo rol con los permisos predeterminados para configurar automáticamente un rol de IAM con los permisos necesarios para que los CloudWatch registros ejecuten las consultas programadas.

   b. Elija Usar un rol existente para seleccionar un rol de IAM existente con las políticas necesarias para que los CloudWatch registros ejecuten las consultas programadas. Utilice el campo de búsqueda para buscar y seleccionar el rol de IAM adecuado de la lista.

10. Seleccione Crear programación para crear la consulta programada.

AWS CLI

Para crear una consulta programada (AWS CLI)

- Utilice el `create-scheduled-query` comando para crear una nueva consulta programada:

```
aws logs create-scheduled-query \
    --name "ErrorAnalysisQuery" \
    --query-language "CWLI" \
    --query-string "fields @timestamp, @message | filter @message like /ERROR/ |
 stats count() by bin(5m)" \
    --schedule-expression "cron(8 * * * ? *)" \
    --execution-role-arn "arn:aws:iam::123456789012:role/
CloudWatchLogsScheduledQueryRole" \
    --log-group-identifiers "/aws/lambda/my-function" "/aws/apigateway/my-api" \
    --state "ENABLED"
```

API

Para crear una consulta programada (API)

- Utilice la `CreateScheduledQuery` acción para crear una nueva consulta programada. El siguiente ejemplo crea una consulta programada que se ejecuta cada hora:

```
{
    "name": "ErrorAnalysisQuery",
    "queryLanguage": "CWLI",
    "queryString": "fields @timestamp, @message | filter @message like /ERROR/ |
 stats count() by bin(5m)",
    "scheduleExpression": "cron(8 * * * ? *)",
    "executionRoleArn": "arn:aws:iam::123456789012:role/
CloudWatchLogsScheduledQueryRole",
    "logGroupIdentifiers": ["/aws/lambda/my-function", "/aws/apigateway/my-
api"],
    "state": "ENABLED"
}
```

Tras crear la consulta programada, puede verla y gestionarla desde la página de consultas programadas y mediante la ListScheduledQueries API, que muestra todas las consultas programadas con sus nombres, fechas de creación, estado de la última ejecución, hora de la última activación y frecuencia de repetición.

Visualización y administración de las consultas programadas

La siguiente información está disponible para cada consulta:

Name

El nombre exclusivo que asignó a la consulta programada. Seleccione el nombre para ver el historial detallado de configuración y ejecución.

Fecha de creación

La fecha en que se creó la consulta programada, que se muestra en YYYY-MM-DD formato.

Estado de la última ejecución

El estado de ejecución de la consulta más reciente. Los valores posibles son:

- Completa: la consulta se ejecutó correctamente y los resultados se enviaron a todos los destinos configurados.
- Error: no se pudo ejecutar la consulta o entregar el resultado. Compruebe el historial de ejecuciones para ver los detalles del error.
- Consulta no válida: la consulta no es válida y tiene problemas de sintaxis

- Tiempo de espera: se ha agotado el tiempo de espera de la consulta. El tiempo de espera de una consulta se agota automáticamente después de 60 minutos

Hora de la última activación

La fecha y la hora en que se ejecutó la consulta por última vez, mostradas en formato YYYY-MM-DD HH:MM:SS. Muestra Nunca si la consulta aún no se ha ejecutado.

Repitiendo cada

La frecuencia de programación de la consulta. Muestra la opción Personalizada para consultas que utilizan expresiones cron o descripciones de frecuencias específicas para programar de forma más sencilla.

La página de consultas programadas proporciona una descripción general de todas las consultas programadas y muestra su estado actual y su historial de ejecución para que pueda ver, supervisar y gestionar todas las consultas programadas desde una ubicación centralizada. Utilice esta información para supervisar el rendimiento de las consultas, identificar problemas y gestionar sus flujos de trabajo de análisis de registros automatizados.

Console

Para ver las consultas programadas (consola)

1. ¿Abrir la consola CloudWatch de Logs en https://us-east-1.console.aws.amazon.com/cloudwatch/casa? region=us-east-1 #LogsV2:Logs-Insights.
2. En la consola de CloudWatch registros, seleccione Consulta programada y Ver consultas programadas.

AWS CLI

Para ver una lista de las consultas programadas (AWS CLI)

- Utilice el `list-scheduled-queries` comando para enumerar todas las consultas programadas:

```
aws logs list-scheduled-queries --max-results 10
```

API

Para enumerar las consultas programadas (API)

- Usa la `ListScheduledQueries` acción para recuperar todas las consultas programadas:

```
{
    "maxResults": 10
}
```

El encabezado de la página de consultas programadas muestra el número total de consultas programadas de su cuenta, lo que le ayuda a realizar un seguimiento del uso y a gestionar sus flujos de trabajo de análisis de registros automatizados de forma eficaz.

Ver el historial de ejecución de consultas programadas

Utilice el historial de ejecución para supervisar el rendimiento de las consultas programadas y solucionar cualquier problema relacionado con la ejecución de las consultas o la entrega de los resultados.

El historial de ejecución muestra el estado de cada consulta ejecutada, incluidas las ejecuciones correctas, los errores y los resultados del procesamiento de destino. Puede utilizar esta información para identificar patrones, diagnosticar problemas y comprobar que las consultas se ejecutan según lo previsto.

Console

Para ver el historial de ejecuciones (consola)

1. En la consola de CloudWatch registros, seleccione Consulta programada y Ver consultas programadas.
2. Seleccione la consulta programada que desee examinar.
3. Elija la pestaña Execution history (Historial de ejecución).

AWS CLI

Para ver el historial de ejecuciones (AWS CLI)

1. Utilice el `get-scheduled-query-history` comando para recuperar el historial de ejecución de una consulta programada:

```
aws logs get-scheduled-query-history \
    --identifier "DailyErrorMonitoring" \
    --start-time 1743379200 \
    --end-time 1743465600 \
    --max-results 10
```

2. Para filtrar por estado de ejecución, añada el `--execution-statuses` parámetro:

```
aws logs get-scheduled-query-history \
    --identifier "DailyErrorMonitoring" \
    --start-time 1743379200 \
    --end-time 1743465600 \
    --max-results 1 \
    --execution-statuses "SUCCEEDED"
```

API

Para ver el historial de ejecuciones (API)

- Utilice la `GetScheduledQueryHistory` acción para recuperar el historial de ejecuciones:

```
{
    "identifier": "DailyErrorMonitoring",
    "startTime": 1743379200,
    "endTime": 1743465600,
    "maxResults": 10,
    "executionStatuses": ["SUCCEEDED", "FAILED"]
}
```

El historial de ejecuciones muestra:

- Estado de ejecución: en ejecución, completa, fallida, tiempo de espera o InvalidQuery
- Hora de activación: cuando se ejecutó la consulta

- Destinos: estado de procesamiento para cada destino configurado, incluidos S3 y EventBridge
- Mensajes de error: detalles sobre cualquier error en la ejecución de la consulta o en el procesamiento del destino

Actualización de una consulta programada

Modifique la configuración de la consulta programada para cambiar la cadena de consulta, el cronograma, los destinos o la función de ejecución a medida que cambien sus requisitos.

Puede actualizar cualquier aspecto de una consulta programada, incluida la cadena de consulta, la expresión del programa, los destinos y la función de ejecución. Los cambios entran en vigor inmediatamente para futuras ejecuciones.

Console

Para actualizar una consulta programada (consola)

1. En la consola de CloudWatch registros, seleccione Consulta programada y Ver consultas programadas.
2. Seleccione la consulta programada que desee actualizar.
3. Elija Edit (Edición de).
4. Modifique la configuración según sea necesario.
5. Seleccione Save changes (Guardar cambios).

AWS CLI

Para actualizar una consulta programada (AWS CLI)

- Utilice el `update-scheduled-query` comando para modificar una consulta programada existente:

```
aws logs update-scheduled-query \
    --identifier "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f" \
    --description "Monitor for ERROR level logs daily" \
    --query-language "LogsQL" \
    --query-string "fields @timestamp, @message | filter @message like /ERROR/"
 \
```

```
    --log-group-identifiers "/aws/lambda/my-function-1" "/aws/lambda/my-
function-2"
```

API

Para actualizar una consulta programada (API)

1. Utilice la `UpdateScheduledQuery` acción para modificar la configuración de la consulta programada:

```
{
    "identifier": "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f",
    "queryString": "fields @timestamp, @message | filter @message like /WARNING|
ERROR/ | stats count() by bin(5m)",
    "scheduleExpression": "cron(0 */2 * * ? *)",
    "state": "ENABLED"
}
```

2. Para actualizar varios parámetros de configuración a la vez:

```
{
    "identifier": "arn:aws:logs:us-east-1:111122223333:scheduled-
query:5e0c0228-1c29-4d26-904f-59f1f1ba3c8f",
    "queryString": "fields @timestamp, @message, @level | filter @level =
 'ERROR'",
    "scheduleExpression": "cron(0 8,12,16 * * ? *)",
    "executionRoleArn": "arn:aws:iam::111122223333:role/
UpdatedScheduledQueryRole",
    "logGroupIdentifiers": ["/aws/lambda/my-function", "/aws/lambda/another-
function"],
    "destinationConfiguration": {
        "s3Configuration": {
            "destinationIdentifier": "s3://111122223333-sqn-results-bucket/
processed-results",
            "roleArn": "arn:aws:iam::111122223333:role/Admin"
        }
    }
}
```

Configuración de los destinos de S3 para las consultas programadas

Configure Amazon S3 como destino para almacenar los resultados de las consultas programadas como archivos JSON para su retención y análisis a largo plazo.

Cuando se utiliza Amazon S3 como destino, los resultados de las consultas se almacenan como archivos JSON en el bucket y el prefijo especificados. Esta opción es ideal para archivar los resultados, realizar análisis por lotes o integrarlos con otros AWS servicios que procesan datos de S3.

Puede enviar los resultados de la consulta a un depósito de Amazon S3 de la misma AWS cuenta que la consulta programada o a un depósito de una AWS cuenta diferente. Si lo desea, también puede cifrar los resultados de las consultas mediante una AWS KMS clave administrada por el cliente (SSE-KMS).

Entregar los resultados a un bucket de Amazon S3 de la misma cuenta

Cuando el bucket de Amazon S3 de destino se encuentra en la misma AWS cuenta que la consulta programada, puede buscar y seleccionar el bucket directamente desde la consola.

Para configurar un destino de Amazon S3 con la misma cuenta (consola)

1. En la sección Publicar los resultados de la consulta en S3, para el bucket de S3, selecciona Esta cuenta.
2. En el caso del URI de Amazon S3, introduzca el bucket y el prefijo de Amazon S3 en los que se almacenarán los resultados (por ejemplos`3://my-bucket/query-results/`) o seleccione Browse Amazon S3 para navegar y seleccionar una ubicación de Amazon S3 existente.
3. (Opcional) Para cifrar los resultados con una AWS KMS clave gestionada por el cliente, introduzca el ARN de la clave en AWS KMS el campo ARN de la clave KMS. La clave debe estar en la misma AWS región que el bucket de Amazon S3 de destino. Si no especificas una AWS KMS clave, se aplicará la configuración de cifrado predeterminada del depósito.
4. En la sección Función de IAM para publicar los resultados de las consultas en Amazon S3, seleccione Crear automáticamente una nueva función con los permisos predeterminados para configurar automáticamente los permisos necesarios, o elija Usar una función existente para seleccionar una función de IAM existente con las políticas necesarias.

La función de IAM de entrega de destino requiere los siguientes permisos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

Entregar los resultados a un bucket de Amazon S3 de otra cuenta

Puede enviar los resultados de las consultas programadas a un bucket de Amazon S3 de otra AWS cuenta. Cuando utilice un bucket multicuenta, debe proporcionar el URI de Amazon S3 y el ID de cuenta de la cuenta propietaria del bucket.

Para configurar un destino de Amazon S3 multicuenta (consola)

1. En la sección Publicar los resultados de la consulta en S3, para el bucket de S3, seleccione Otra cuenta e introduzca el ID de cuenta de la cuenta propietaria del bucket.
2. Para el URI de Amazon S3, introduzca el URI completo de Amazon S3 del bucket de destino y el prefijo en la otra cuenta (por ejemplo,`s3://cross-account-bucket/query-results/`).
3. (Opcional) Para cifrar los resultados con una AWS KMS clave gestionada por el cliente, introduzca el ARN de la clave en AWS KMS el campo ARN de la clave KMS. La clave debe estar en la misma AWS región que el bucket de Amazon S3 de destino.
4. En la sección Función de IAM para publicar los resultados de las consultas en Amazon S3, seleccione Crear automáticamente una nueva función con los permisos predeterminados para configurar automáticamente los permisos necesarios, o elija Usar una función existente para seleccionar una función de IAM existente con las políticas necesarias.

La entrega entre cuentas requiere permisos de ambas partes. La función de IAM de entrega de destino en la cuenta de origen requiere los siguientes permisos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::cross-account-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

La política de bucket de Amazon S3 de la cuenta de destino debe conceder permiso al rol de IAM de la cuenta de origen para escribir objetos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowScheduledQueryRolePutObject",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/my-s3-delivery-role"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::cross-account-bucket/prefix/*"
        }
    ]
}
```

Cifrar los resultados con una clave administrada por el cliente AWS KMS

Si lo desea, puede especificar una AWS KMS clave administrada por el cliente para cifrar los resultados de las consultas enviados a Amazon S3 mediante SSE-KMS. La AWS KMS clave puede estar en la misma cuenta que la consulta programada o en una cuenta diferente.

Al especificar una AWS KMS clave, la consulta programada usa esa clave para cifrar los resultados mediante SSE-KMS. Si no especificas una AWS KMS clave, se aplica la configuración de cifrado predeterminada del depósito. Si el depósito está configurado con el cifrado SSE-KMS predeterminado mediante una clave administrada por el cliente, la función de IAM de entrega de destino aún debe tener `kms:GenerateDataKey` permiso sobre esa clave.

La función de IAM de entrega de destino requiere el `kms:GenerateDataKey` permiso de la clave. AWS KMS El siguiente ejemplo muestra los permisos necesarios para un destino de Amazon S3 con una AWS KMS clave gestionada por el cliente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "kms:GenerateDataKey",
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "s3.us-east-1.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::my-bucket*"
                }
            }
        }
    ]
}
```

Cuando la AWS KMS clave está en una cuenta diferente a la de la función de IAM de entrega de destino, la política AWS KMS clave de la cuenta propietaria de la clave debe conceder explícitamente el acceso a la función:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowScheduledQueryRoleToEncrypt",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/my-s3-delivery-role"
            },
            "Action": "kms:GenerateDataKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "s3.us-east-1.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::my-bucket*"
                }
            }
        }
    ]
}
```

> **Note**
>
> Cuando la AWS KMS clave y la función de IAM de entrega de destino están en la misma cuenta, la política de identidad de IAM por sí sola es suficiente si la política AWS KMS clave incluye la declaración raíz predeterminada «Habilitar las políticas de IAM». Solo se requiere una concesión explícita de la política AWS KMS clave si la política clave no se delega en IAM.

La función de IAM para publicar los resultados de las consultas en Amazon S3 debe configurarse de forma independiente de la función de IAM para la ejecución programada de las consultas. Esta separación permite un control de acceso detallado, en el que la función de ejecución puede ejecutar consultas, mientras que la función de Amazon S3 se encarga específicamente de la entrega de resultados. Ambas funciones deben incluir una política de confianza que permita que el servicio de CloudWatch registros (`logs.amazonaws.com`) asuma la función.

Solución de problemas de consultas programadas

Utilice estos temas de solución de problemas para resolver problemas comunes relacionados con las consultas programadas.

La ejecución de la consulta falla debido a errores de permisos

Resuelva los errores de permisos que impiden que las consultas programadas se ejecuten o entreguen resultados a los destinos.

Los errores de permisos se producen cuando la función de ejecución carece de los permisos necesarios para leer los grupos de registros o escribir en los recursos de destino.

Para resolver errores de permisos

1. Compruebe que la función de ejecución tenga `logs:StartQuery` los grupos de registros de destino y los `logs:DescribeLogGroups` permisos necesarios para ellos. `logs:GetQueryResults`
2. Asegúrese de que la función de ejecución tenga permisos de escritura para los recursos de destino (por ejemplo, `s3:PutObject` para los buckets de S3).
3. Compruebe que la política de confianza permita a CloudWatch Logs asumir la función de ejecución. El rol debe confiar en el principal (`logs.amazonaws.com`) del servicio de registros en su política de confianza.

Las causas más comunes incluyen la falta de permisos de IAM, un recurso incorrecto ARNs en la política o problemas de configuración de la política de confianza.

Para evitar errores de permisos, utilice el principio de privilegios mínimos al crear funciones de ejecución y pruebe los permisos antes de implementar las consultas programadas en producción.

El tiempo de espera de la consulta

Resuelva los errores de tiempo de espera que se producen cuando las consultas programadas superan el límite máximo de tiempo de ejecución.

Los tiempos de espera de las consultas se producen cuando la consulta tarda más de 60 minutos en procesar el rango de datos especificado, a menudo debido a conjuntos de datos grandes o a una lógica de consulta compleja.

Para resolver errores de tiempo de espera

1. Reduzca el intervalo de tiempo reduciendo el intervalo de tiempo de inicio para procesar menos datos por ejecución.
2. Optimice la consulta añadiendo filtros al principio de la consulta para reducir la cantidad de datos procesados. Utilice índices de filtro para reducir el tamaño del escaneo de datos.
3. Considere dividir las consultas complejas en consultas más simples y específicas.

Entre las causas más comunes se incluyen las consultas en intervalos de tiempo extensos, el procesamiento de grupos de registros de gran volumen o el uso de agregaciones complejas sin el filtrado adecuado.

Para evitar tiempos de espera, pruebe las consultas manualmente en CloudWatch Logs Insights con el volumen de datos esperado y optimice el rendimiento antes de programarlas.

El procesamiento de destino falla

Resuelva los errores que se producen cuando los resultados de las consultas programadas no se pueden entregar a los destinos configurados.

Los errores en el procesamiento de destino se producen cuando el bucket de Amazon S3 o el bus de EventBridge eventos de destino son inaccesibles o están mal configurados.

Para resolver errores en los que los resultados de las consultas no se publican en el destino

1. Compruebe que el bucket de Amazon S3 especificado existe y es accesible.
2. Compruebe que la configuración de destino es correcta URIs.
3. Asegúrese de que la función de ejecución tenga los permisos necesarios para escribir en el destino.

Entre las causas más comunes se incluyen los recursos de destino eliminados o cuyo nombre se ha cambiado, el destino URIs incorrecto o los problemas de conectividad de la red.

Para evitar errores en el destino, valide periódicamente las configuraciones de destino y supervise la disponibilidad de los recursos de destino.

Errores de consulta no válidos

Resuelva los errores de sintaxis y lógica en las cadenas de consultas programadas que impiden una ejecución correcta.

Los errores de consulta no válidos se producen cuando la cadena de consulta contiene errores de sintaxis, hace referencia a campos inexistentes o utiliza funciones de lenguaje de consulta no compatibles.

Para resolver errores de consulta no válidos

1. Pruebe la consulta manualmente en CloudWatch Logs Insights para comprobar la sintaxis y la lógica.
2. Compruebe que todos los campos de registro a los que se hace referencia existan en los grupos de registros de destino.
3. Compruebe que las funciones del lenguaje de consulta que utiliza son compatibles con las consultas programadas.

Las causas más comunes incluyen errores tipográficos en los nombres de los campos, una sintaxis de consulta incorrecta o el uso de funciones de consulta que no son compatibles con el entorno de ejecución programada.

Para evitar errores de consulta no válidos, pruebe siempre las consultas de forma interactiva antes de programarlas y utilice las funciones de detección de campos para comprobar los nombres de los campos.

Errores de simultaneidad de consultas

Hay algunos puntos importantes que se mencionan a continuación a la hora de detectar errores de simultaneidad, ya que las consultas programadas utilizan la misma cuota que las consultas de información de Cloudwatch Logs. Se recomienda distribuir los horarios para evitar alcanzar el límite de simultaneidad.

- Cuota: puedes ejecutar hasta 100 consultas simultáneas de CloudWatch Logs Insights por cuenta. AWS
- Cuadros de mando: las consultas que se añaden a los CloudWatch cuadros de mando también se tienen en cuenta para este límite de simultaneidad, ya que se ejecutan al cargar o actualizar el cuadro de mando.

- OpenSearch Servicio PPL/SQL: puede ejecutar hasta 15 consultas PPL o SQL OpenSearch simultáneas por cuenta. OpenSearch AWS
- Consultas entre cuentas: la cuota de simultaneidad se aplica tanto a las consultas entre cuentas únicas como a las consultas entre cuentas. Cuando se utiliza la observabilidad CloudWatch multicuenta, las consultas iniciadas en una cuenta de monitorización contra una cuenta de origen vinculada también se tienen en cuenta para el límite de simultaneidad de la cuenta de monitorización.
- Grupos de registros de acceso poco frecuente: en el caso de los grupos de registros de acceso poco frecuente, el número máximo de consultas simultáneas de Logs Insights se limita a cinco.

Detección de anomalías en registros

Puede detectar anomalías en los datos de registro de dos maneras: creando un detector de anomalías de registro para una supervisión continua o utilizando el `anomaly detection`comando en las consultas de CloudWatch Logs Insights para realizar análisis bajo demanda.

Un detector de anomalías de registro analiza los eventos de registro incorporados a un grupo de registros y encuentra anomalías de manera automática en los datos del registro. La detección de anomalías utiliza métodos de machine learning y el reconocimiento de patrones para establecer líneas base del contenido típico de los registros. Para el análisis bajo demanda, puede usar el `anomaly detection` comando en las consultas de CloudWatch Logs Insights para identificar patrones inusuales en los datos de series temporales. Para obtener más información sobre la detección de anomalías basadas en consultas, consulte Uso de la detección de anomalías en Logs Insights CloudWatch.

Después de crear un detector de anomalías para un grupo de registro, se entrena mediante los eventos de registro de las últimas dos semanas en el grupo de registro para la formación. El período de formación puede tardar 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 ResourceID: 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for ResourceID: 324892398123-1234R
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for ResourceID: 3ff231242342-12345
```

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

```
<Date-1> <Time-2> [INFO] Calling DynamoDB to store for resource id <ResourceID-3>
```

Los campos dentro de un patrón se denominan tokens. Los campos que varían dentro de un patrón, como un ID de solicitud o una marca de tiempo, se denominan tokens dinámicos. Cada valor diferente que se encuentre en un token dinámico se denomina valor de token.

Si CloudWatch Logs puede deducir el tipo de datos que representa un token dinámico, muestra el token como<*string*-*number*>. *string*Es una descripción del tipo de datos que representa el token. *number*Muestra en qué parte del patrón aparece este token, en comparación con los otros tokens dinámicos.

CloudWatch Los registros asignan a la cadena una parte del nombre en función del análisis del contenido de los eventos del registro que lo contienen.

Si CloudWatch Logs no puede deducir el tipo de datos que representa un token dinámico, muestra el token como <Token- *number* > e *number* indica en qué parte del patrón aparece este token, en comparación con los demás tokens dinámicos.

Algunos ejemplos comunes de tokens dinámicos son los códigos de error, las direcciones IP, las marcas de tiempo y las solicitudes. IDs

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.

En este capítulo se describe cómo habilitar la detección de anomalías, ver las anomalías, crear alarmas para los detectores de anomalías de registro y las métricas que publican estos últimos. También describe cómo cifrar el detector de anomalías y sus resultados con. AWS Key Management Service

La creación de detectores de anomalías de registro no conlleva cargos.

Gravedad y prioridad de las anomalías y patrones

A cada anomalía que descubre un detector de anomalías de registro 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 del token aumenta repentinamente un 500 %, esa anomalía puede designarse como de prioridad `HIGH` aunque su gravedad sea `NONE`.
- La gravedad se basa únicamente en las palabras clave que se encuentran en patrones como `FATAL`, `ERROR` y `WARN`. Si no se encuentra ninguna de estas palabras clave, la gravedad del patrón se marca como `NONE`.

Tiempo de visibilidad de anomalías

Al crear un detector de anomalías, debe especificar el período máximo de visibilidad de las anomalías. Es el número de días durante los que la anomalía se muestra en la consola y la devuelve la operación de la ListAnomaliesAPI. Una vez transcurrido este período de tiempo en una anomalía, si continúa ocurriendo, se acepta automáticamente como un comportamiento normal y el modelo detector de anomalías deja de marcarla como tal.

Si no ajusta el tiempo de visibilidad al crear un detector de anomalías, se utilizan 21 días de forma predeterminada.

Supresión de anomalías

Una vez detectada una anomalía, puede 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 esta.

Puede seguir viendo las anomalías suprimidas en la consola. También puede dejar de suprimirlas.

Preguntas frecuentes

¿ AWS Utilizo mis datos para entrenar los 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 registro 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 en los registros.

La detección de anomalías en los registros no es adecuada para: Registrar eventos con estructuras JSON extremadamente largas, como CloudTrail los registros. 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 esta opción 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 supera los 300, la detección de anomalías debería funcionar bien. Para obtener más información sobre el análisis de patrones, consulte Análisis del patrón.

¿Qué se marca como una anomalía?

Los siguientes resultados 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 registro.
- Una variación significativa de un patrón conocido.
- Un valor nuevo de 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 marcarse como anomalías, no todos ellos 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, considere la posibilidad de suprimir las anomalías que no indiquen problemas.

¿Qué ocurre con los datos confidenciales que se enmascaran?

Las partes de los eventos de registro que estén enmascaradas como datos confidenciales no se escanean para detectar anomalías. Para obtener más información, consulte Protección de datos de registro confidenciales con el enmascaramiento.

Uso de la detección de anomalías en Logs Insights CloudWatch

Además de crear detectores de anomalías en los registros para una supervisión continua, también puede utilizar el `anomaly` comando en las consultas de CloudWatch Logs Insights para identificar patrones inusuales en los datos de registro a pedido. Este comando amplía la funcionalidad

`pattern` existente y utiliza machine learning para detectar cinco tipos de anomalías, incluidos los cambios en la frecuencia de los patrones, los nuevos patrones y las variaciones de los tokens.

El comando `anomaly` resulta especialmente útil para:

- Análisis ad hoc de los datos de registro históricos para identificar patrones inusuales
- Investigación de períodos de tiempo específicos para detectar comportamientos anómalos
- Supervisión de aplicaciones como las funciones de Lambda para detectar problemas de ejecución

Para obtener más información sobre cómo usar los comandos `anomaly` en las consultas, consulte anomalía.

Esta detección de anomalías basada en consultas complementa los detectores de anomalías continuos que se describen en las siguientes secciones y proporciona funciones tanto de supervisión en tiempo real como de análisis bajo demanda.

Habilitación de la detección de anomalías en un grupo de registro

Siga estos pasos para usar la CloudWatch consola y crear un detector de anomalías de registro que escanee un grupo de registros en busca de anomalías.

También puede crear detectores de anomalías mediante programación. Para obtener más información, consulte CreateLogAnomalyDetector.

Creación de un detector de anomalías de registro

1. Abra la consola en CloudWatch . https://console.aws.amazon.com/cloudwatch/
2. Elija Registros, Anomalías de registro.
3. Elija Crear detector de anomalías.
4. Seleccione el grupo de registro para el que desee crear el 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 según la frecuencia con la que el grupo de registro reciba nuevos registros. Por ejemplo, si el grupo de registro recibe nuevos eventos de registro en lotes cada 10 minutos, es recomendable establecer la frecuencia de evaluación en 15 minutos.
7. (Opcional) Para configurar el detector de anomalías para que busque anomalías solo en los eventos de registro que contengan determinadas palabras o cadenas, elija Patrones de filtro.

A continuación, introduzca un patrón en Patrón de filtro de la detección de anomalías. Para obtener más información acerca de 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 Mensajes de eventos de registro y, a continuación, elija 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 valor predeterminado, introduzca un nuevo valor en Período máximo de visibilidad de 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 que recopile 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. Elija Habilitar la detección de anomalías.

   Se crea el detector de anomalías y comienza a entrenar su modelo en función de los eventos de registro que incorpora al grupo de registro. Transcurridos unos 15 minutos, la detección de anomalías se activa y comienza a detectar anomalías y a descubrirlas.

Consulta de 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.

Visualización de las anomalías que hayan encontrado todos sus detectores de anomalías de registro

1. Abra la consola en CloudWatch . https://console.aws.amazon.com/cloudwatch/
2. Elija Registros, Anomalías de registro.

   Aparece la tabla Anomalías de registro. El número que aparece en la parte superior, junto a Anomalías de registro, muestra cuántas anomalías de registro se muestran en la tabla. Cada fila de la tabla muestra la siguiente información:

   - La columna Anomalía muestra un breve resumen de la anomalía. Estos resúmenes los genera CloudWatch Logs.
   - Prioridad de la anomalía. La prioridad se calcula automáticamente en función de la cantidad de cambios que se hayan hecho en los eventos de registro y palabras clave como `Exception` que se crean en un evento de registro, entre otras cosas.
   - Patrón de registro en el que se basa la anomalía. Para obtener más información acerca de los patrones, consulte Detección de anomalías en registros.
   - 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.
   - Detector de anomalías muestra el nombre del grupo de registro que contiene los eventos de registro relacionados con esta anomalía. Puede elegir este nombre para ver la página de detalles del grupo de registro.
3. Para seguir inspeccionando una anomalía, pulse el botón de opciones en la fila correspondiente.

   Aparecerá el panel Inspección de patrones, que muestra lo siguiente:

   - 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 Muestras de registros le permite ver 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 es que seleccionó uno.

> **Note**
>
> Se captura un máximo de 10 valores de token por 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 opciones de la fila correspondiente y, a continuación, haga lo siguiente:

   a. Seleccione Acciones y Suprimir anomalía.

   b. A continuación, especifique durante cuánto tiempo quiere suprimir la anomalía.

   c. Para suprimir todas las anomalías relacionadas con este patrón, seleccione Suprimir patrón.

   d. Seleccione Suprimir anomalía.

Visualización de las anomalías encontradas en un solo grupo de registro

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 registro y, a continuación, elija la pestaña Detección de anomalías.

   Aparecerá la tabla Detección de anomalías. El número que aparece en la parte superior, junto a Anomalías de registro, muestra cuántas anomalías de registro se muestran en la tabla. Cada fila de la tabla muestra la siguiente información:

   - La columna Anomalía muestra un breve resumen de la anomalía. Estos resúmenes los genera CloudWatch Logs.
   - Prioridad de la anomalía. La prioridad se calcula automáticamente en función de la cantidad de cambios que se hayan hecho en los eventos de registro y palabras clave como `Exception` que se crean en un evento de registro, entre otras cosas.
   - Patrón de registro en el que se basa la anomalía. Para obtener más información acerca de los patrones, consulte Detección de anomalías en registros.
   - 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 opciones en la fila correspondiente.

   Aparecerá el panel Inspección de patrones, que muestra lo siguiente:

   - 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 Muestras de registros le permite ver 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 es que seleccionó uno.

   > **Note**
   >
   > Se captura un máximo de 10 valores de token por 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 opciones de la fila correspondiente y, a continuación, haga lo siguiente:

   a. Seleccione Acciones y Suprimir anomalía.

   b. A continuación, especifique durante cuánto tiempo quiere suprimir la anomalía.

   c. Para suprimir todas las anomalías relacionadas con este patrón, seleccione Suprimir patrón.

   d. Seleccione Suprimir anomalía.

Creación de alarmas en los detectores de anomalías de registro

Puede crear una alarma para un detector de anomalías de registros que esté un grupo de registro. Puede especificar que la alarma active el estado `ALARM` cuando se encuentre un número específico de anomalías en el grupo de registro durante un período de tiempo determinado. También puede usar filtros para que la alarma solo tenga en cuenta las anomalías con prioridades específicas.

Creación de 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, seleccione Registros, Anomalías de registros.

   Aparecerá la tabla de detectores de anomalías de registro.
3. Seleccione el botón de opciones correspondiente al detector de anomalías en el que desea 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 por prioridad de anomalía, lleve a cabo 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 acerca de los niveles de prioridad, consulte Gravedad y prioridad de las anomalías y patrones.
5. Utilice un umbral de detección de anomalías estático o métrico para la alarma. Esta selección determina cómo se establece el umbral de la 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 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 de la detección de CloudWatch 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 activa el estado `ALARM` si el detector de anomalías encuentra más de este número de alarmas durante un período especificado en el campo 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, creará 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 las CloudWatch alarmas tratan los datos faltantes.
9. Elija Siguiente.
10. En Notificación, elija Agregar notificación y, a continuación, especifique el tema de Amazon SNS al que desee enviar la notificación cuando la alarma tenga una transición al estado `ALARM`, `OK` 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, o bien AWS Systems Manager, 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 runbooks 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 nombres de `AWS/Logs`.

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 este AWS Key Management Service cifrado. Si lo hace, el cifrado se realiza mediante una AWS KMS clave. El cifrado que usa AWS KMS se habilita en el nivel del detector de anomalías, al asociar 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 registro. 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 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 sección `Condition` de este ejemplo limita la utilización de la clave de AWS KMS en la cuenta especificada, pero se puede utilizar para cualquier detector de anomalías.

JSON

```
{
"Version":"2012-10-17",
"Id": "key-default-1",
"Statement": [
    {
        "Sid": "EnableIAMUserPermissions",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:root"
        },
        "Action": "kms:*",
        "Resource": "*"
    },
```

```
        {
            "Sid": "AllowCloudWatchLogsEncryption",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsDescribeKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsReEncryption",
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws-crypto-ec:aws:logs:arn":
 "arn:aws:logs:us-east-1:123456789012:anomaly-detector:*"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-
east-1:123456789012:anomaly-detector:*"
                }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsDescribeKeyForReEncryption",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

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 detector de anomalías

Puede asociar una clave KMS a un detector de anomalías al crearla en la consola o mediante la tecla AWS CLI o APIs.

Paso 4: desvincular una clave de un detector de anomalías

Una vez que ha asociado una clave a un detector de anomalías, la clave no se puede actualizar. La única forma de eliminar la clave es eliminar el detector de anomalías y, a continuación, volver a crearlo.

Solución de problemas con CloudWatch Logs Live Tail

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.

Live Tail solo se admite en los grupos de registro de la clase de registro Estándar. Para obtener más información acerca de las clases de registros, consulte Clases de registro.

En las secciones siguientes se explica cómo utilizar Live Tail en la consola y en AWS CLI. También puede iniciar una sesión de Live Tail de 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.

También se puede usar Live Tail en AWS Toolkit for Visual Studio Code. Para iniciar una sesión de Live Tail desde la paleta de comandos de VS Code, consulte la sección Amazon CloudWatch Logs Live Tail de la Guía del AWS Toolkit for Visual Studio Code usuario.

La función Live Tail está disponible en todas AWS las regiones comerciales. No está disponible en las regiones de China ni en las regiones AWS GovCloud (EE. UU.).

> **Note**
>
> La `StartLiveTail` API enruta las solicitudes mediante la inyección de prefijos de host del SDK. Las versiones del SDK publicadas antes del 1 de abril de 2026 se dirigen a `streaming-logs.`*`Region`*`.amazonaws.com` puntos de enlace de VPC, que no son compatibles. Las versiones del SDK publicadas a partir del 1 de abril de 2026 se dirigen `astream-logs.`*`Region`*`.amazonaws.com`, que son compatibles con los puntos de conexión de VPC. Para configurar un punto de enlace de VPC para esta API, consulta Cómo crear un punto de enlace de VPC para registros. CloudWatch

Inicie una sesión de Live Tail con AWS CLI

El `start-live-tail` AWS CLI comando inicia una sesión de streaming de Live Tail para uno o más grupos de registros en una terminal. Una sesión de Live Tail puede durar hasta tres horas. Si más de 500 eventos de registro por segundo coinciden con el filtro, los eventos de registro que se indican son una muestra del total de eventos de registro; estos son útiles para ofrecer una experiencia de seguimiento en tiempo real. Para obtener más información acerca del comando `start-live-tail`, consulte start-live-tail.

Puede utilizar `start-live-tail` de dos modos:

- solo impresión: este es el modo predeterminado
- interactivo

solo impresión

En el modo `print-only`, los eventos de registro se transmiten en el terminal. Cada segundo se añaden nuevos eventos en la parte inferior, lo que crea una experiencia de seguimiento casi en tiempo real similar a `tail -f` en Linux.

Para iniciar una sesión de Live Tail en modo de solo impresión, escriba el siguiente comando.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs
```

Si utiliza el modo de solo impresión, también puede combinarlo con otros comandos de Linux para aumentar sus capacidades analíticas. El siguiente ejemplo filtra los eventos de registro con la palabra clave `error` e imprime la segunda y la cuarta columna de estos eventos para que pueda extraer información determinada.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs --mode print-only | grep "error" | awk '{print
 $2, $4}'
```

interactivo

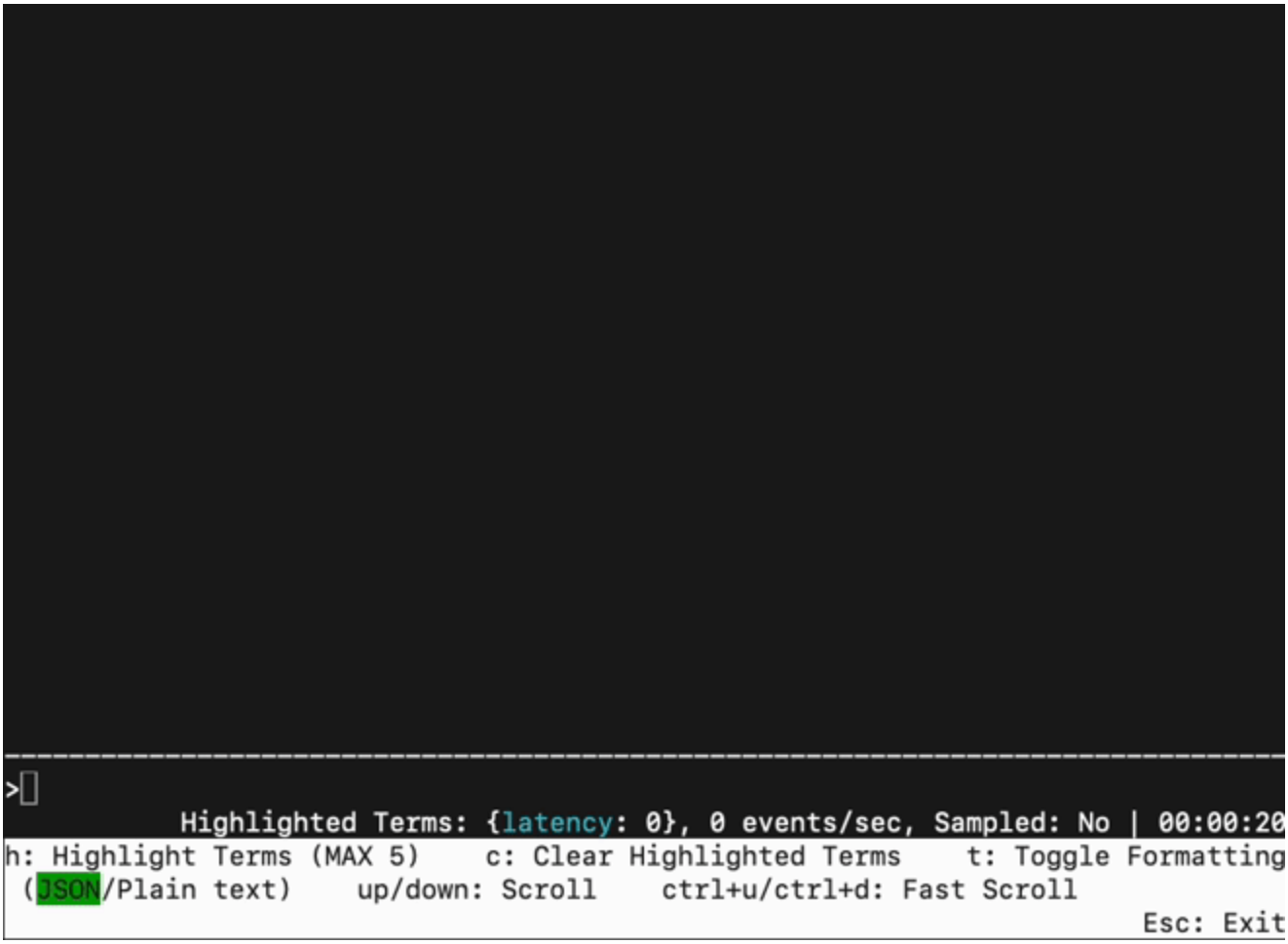
En el modo `interactive`, puede resaltar los términos y cambiar el formato de los eventos de registro de salida entre JSON y texto sin formato. El modo interactivo también muestra información

sobre la sesión de Live Tail, como la duración de la sesión, si se está muestreando la sesión, los términos resaltados actualmente y el recuento de las veces que se han encontrado.

Para iniciar una sesión de Live Tail en modo interactivo, escriba el siguiente comando.

```
aws logs start-live-tail --log-group-identifiers arn:aws:logs:us-
east-1:111111222222:log-group:my-logs --mode interactive
```

Comienza la sesión de Live Tail. En el siguiente vídeo se muestra parte de una sesión de ejemplo.



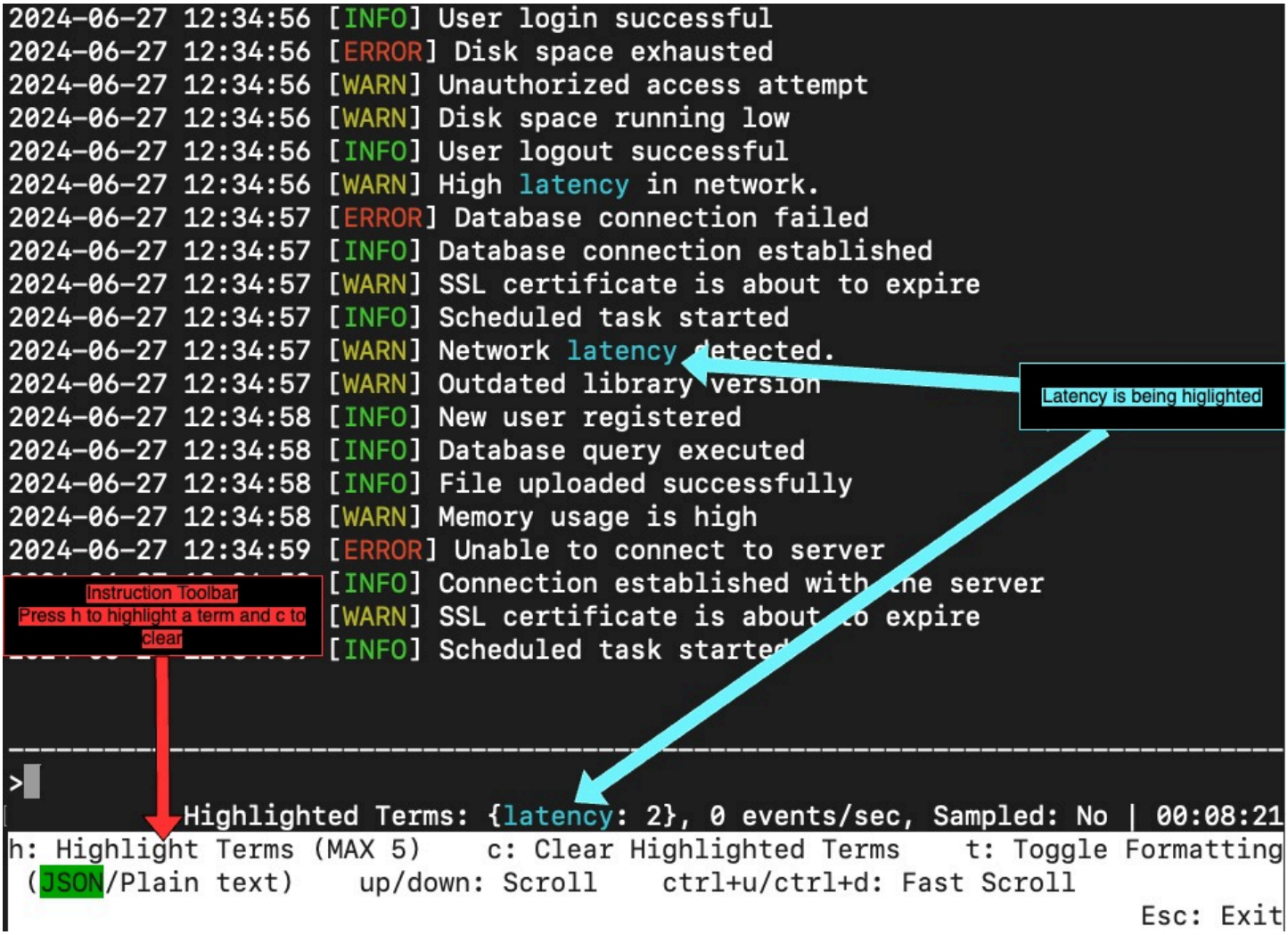
Para resaltar un término en los registros de transmisión, pulse h y, a continuación, introduzca el término. A continuación, se muestra la pantalla una vez que se ha resaltado el término `latency`.

Para borrar un término resaltado, presione c y, a continuación, escriba el número que representa el término que desea dejar de resaltar.

Puede pulsar t para cambiar el formato de visualización de los eventos entrantes entre JSON y el texto sin formato. Esta función de alternancia es bastante útil y solo se produce si el formato del evento de registro es compatible.

Puede usar las teclas de flecha arriba y abajo para desplazarse y usar `CTRL+u` y `CTRL+d` para desplazarse aún más rápido.

La siguiente imagen muestra lo más destacado del término `latency` durante una sesión de Live Tail.



Inicio de una sesión de Live Tail en la consola

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.

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.

> **Important**
>
> Si su equipo de seguridad de red no permite el uso de sockets web, actualmente no puede acceder a la parte de Live Tail de la CloudWatch consola. Puedes usar Live Tail con las API AWS CLI o. Para obtener más información, consulte Inicie una sesión de Live Tail con AWS CLI y StartLiveTail.

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.
   - events/secmuestra el número de eventos de registro ingeridos por segundo que coinciden con los filtros que ha establecido.
   - 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 actualmente, selecciona cualquier lugar 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).

Cross-account Centralización de registros entre regiones

La centralización de datos de Amazon CloudWatch Logs AWS Organizations permite recopilar datos de registro de varias cuentas de miembros en un repositorio de datos mediante reglas de centralización entre cuentas y regiones. Usted define las reglas que replican automáticamente los datos de registro de varias cuentas y Regiones de AWS en una cuenta centralizada dentro de su organización. Esta capacidad optimiza la consolidación de registros para mejorar la supervisión, el análisis y el cumplimiento centralizados en toda la infraestructura. AWS

CloudWatch La centralización de los datos de los registros ofrece flexibilidad de configuración para cumplir con los requisitos operativos y de seguridad, como la posibilidad de configurar una región de respaldo durante la configuración de las reglas en la cuenta de destino para garantizar una mayor resiliencia. Además, tiene pleno control sobre el comportamiento de cifrado de los grupos de registros copiados de las cuentas de origen para gestionar los datos originalmente cifrados con claves KMS administradas por el cliente.

> **Note**
>
> La función de centralización de CloudWatch registros solo procesa los nuevos datos de registro que llegan a las cuentas de origen después de crear la regla de centralización. Los datos de registro históricos (registros que existían antes de la creación de la regla) no están centralizados.

Conceptos de centralización de datos

Antes de empezar a utilizar la centralización de datos de CloudWatch Logs, familiarícese con los siguientes conceptos:

Regla de centralización

Una configuración que define cómo se replican los datos de registro de las cuentas y regiones de origen en una cuenta y región de destino. Las reglas especifican los criterios de origen y la configuración de destino.

Cuenta de origen

La AWS cuenta en la que se originan los datos de registro. Los eventos de registro de las cuentas de origen se replican en la cuenta de destino en función de las reglas de centralización que se definan.

Cuenta de destino

La AWS cuenta de destino donde se almacenan los datos de registro replicados. Esta cuenta sirve como ubicación centralizada para el análisis y la supervisión de los registros.

Región de respaldo

Una región secundaria opcional dentro de la cuenta de destino donde se pueden replicar los datos de registro para aumentar la resiliencia y la recuperación ante desastres.

Cifrado en los registros CloudWatch

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 con el Galois/Counter modo estándar de cifrado avanzado de 256 bits (AES-GCM) para cifrar los datos de registro en reposo. Como alternativa, puede usar el Servicio de administración de AWS claves para este cifrado. Para obtener más información, consulte la documentación sobre el cifrado de CloudWatch registros.

- Cómo funciona el cifrado durante la centralización: la centralización de CloudWatch registros copia activamente los datos de registro en el momento de la ingesta desde las cuentas de origen a las cuentas de destino. Durante este proceso, los datos permanecen cifrados en tránsito mediante una clave de servicio propia AWS . Los datos en reposo de los grupos de registros de origen y destino se cifran mediante el método de cifrado que elija (claves KMS AWS gestionadas por el cliente o propias). Si utiliza una clave KMS administrada por el cliente en sus grupos de registros de destino, añada la etiqueta `LogsManaged = true` a la clave kms para que el servicio de centralización pueda acceder a ella.
- Cuando se requieren permisos de KMS:
  - Si utilizas claves de KMS administradas por el cliente en tus cuentas de origen, CloudWatch Logs requiere permisos de KMS en los siguientes escenarios de ejemplo:
    - Administración del rendimiento: cuando se alcanzan los límites de rendimiento de la centralización, los datos de registro se almacenan temporalmente cifrados con la clave KMS administrada por el cliente hasta que haya ancho de banda disponible.

- Protección y redacción de datos: cuando los grupos de registros de origen tienen habilitadas las políticas de protección de datos, CloudWatch Logs necesita permisos de descifrado para acceder a los datos de registro sin procesar y centralizarlos.

> **Important**
>
> Las reglas de centralización las administra la cuenta de administración de AWS Organizations o el administrador delegado. Para excluir de la centralización los grupos de KMS-encrypted registros gestionados por el cliente, configure las reglas de la siguiente manera: «No centralizar los grupos de registros cifrados con AWS la clave KMS».

Configuración de la centralización de registros

Para configurar la centralización de CloudWatch registros, debe configurar reglas de centralización que definan cómo fluyen los datos de registro desde los grupos de registros de las cuentas de origen a los grupos de registros de su cuenta de destino.

Una vez que la regla de centralización esté habilitada y los eventos de registro se estén replicando en la cuenta de destino, se pueden crear filtros de métricas, suscripciones y cuentas en los grupos de registros centralizados con capacidades de filtrado mejoradas. Estos filtros pueden centrarse en eventos de registro de cuentas y regiones de origen específicas, y pueden emitir información de cuentas y regiones de origen como dimensiones métricas. Para obtener más información, consulte Creación de métricas a partir de eventos de registro mediante filtros.

Requisitos previos

- AWS Organizations debe estar configurada y las cuentas de origen y destino deben pertenecer a la organización.
- El acceso confiable debe estar habilitado para la cuenta de administración y la cuenta de destino CloudWatch, por lo que debe proporcionarse acceso a los datos de registro.

> **Note**
>
> Se recomienda habilitar el acceso confiable a través de la consola, que crea automáticamente el rol vinculado al servicio (SLR) requerido. Si se habilita el acceso de

confianza mediante otros métodos, será necesario crear el rol vinculado al servicio por separado.

Personalización de los nombres de los grupos de registros de destino

Al crear una regla de centralización, puede personalizar la forma en que se estructuran los nombres de los grupos de registros de destino mediante atributos. Estos atributos se sustituyen automáticamente por valores reales cuando se crean los grupos de registros, lo que le permite organizar los registros jerárquicamente en su cuenta de destino. De forma predeterminada, solo se usa el `${source.logGroup}` atributo, que fusiona todos los grupos de registros con el mismo nombre en la cuenta de destino. Si una variable no se puede resolver, hereda el valor de su variable principal en la jerarquía.

Atributos disponibles

Puede usar los siguientes atributos en el patrón de nombre del grupo de registros de destino:

Atributos del nombre del grupo de registros de destino

Atributo	Description (Descripción)
`${source.accountId}`	El ID AWS de cuenta en el que se originó el registro.
`${source.region}`	El Región de AWS lugar donde se originó el registro.
`${source.logGroup}`	El nombre original del grupo de registros de la cuenta de origen.
`${source.org.id}`	Su AWS Organizations ID de la cuenta de origen.
`${source.org.ouId}`	El ID de la unidad organizativa de la cuenta de origen
`${source.org.rootId}`	El ID raíz de la organización
`${source.org.path}`	La ruta organizativa completa desde la cuenta hasta la raíz

Ejemplos

Conserve la estructura original de los grupos de registros

Patrón: `/centralized/${source.accountId}${source.logGroup}`

Resultado: `/centralized/123456789012/aws/lambda/my-function`

Organice por cuenta y región

Patrón: `/centralized/${source.accountId}/${source.region}`

Resultado: `/centralized/123456789012/us-east-1`

Organice por estructura organizativa

Patrón: `/logs/${source.org.id}/${source.org.ouId}/${source.accountId}`

Resultado: `/logs/o-abc123/ou-xyz-12345678/123456789012`

Estructura plana simple

Patrón: `/centralized-logs`

Resultado: `/centralized-logs`

Prácticas recomendadas

- Incluye el ID de la cuenta de origen para identificar fácilmente de qué provienen los registros de la cuenta.
- Incluya la región de origen si va a centralizar desde varias regiones.
- Estructure los nombres de los grupos de registros de destino para que tengan menos de 512 caracteres. CloudWatch Los registros imponen una longitud máxima de 512 caracteres para los nombres de los grupos de registros.

Creación de una regla de centralización

Utilice el siguiente procedimiento para crear una regla de centralización que replique los datos de registro de las cuentas de origen a la cuenta de destino.

Creación de una regla de centralización

1. Navegue hasta la CloudWatch consola de la cuenta de administración o administrador delegado de la organización.
2. Seleccione Configuración.
3. Navegue a la pestaña Organización.
4. Elija Configurar regla.
5. Especifique los detalles de la fuente mediante la configuración de los siguientes campos y, a continuación, seleccione Siguiente:

   a. Nombre de regla de centralización: introduzca un nombre exclusivo para la regla de centralización.

   b. Cuentas de origen: defina los criterios de selección de fuentes para seleccionar las cuentas a partir de las cuales se centralizarán los datos de telemetría. Los criterios de selección pueden incluir:

      - Una de la lista de las cuentas de miembros de la organización.
      - Una lista de las unidades organizativas de la organización.
      - Toda la organización.

      Se pueden proporcionar los criterios de selección de dos modos:

      - Builder: una experiencia basada en clics para generar los criterios de selección de la fuente
      - Editor: un cuadro de texto de formato libre para proporcionar los criterios de selección de la fuente

      Sintaxis compatible para los criterios de selección de fuentes:

      - Claves compatibles: OrganizationId | | OrganizationUnitId AccountId | *
      - Operadores compatibles: = | IN | OR

   c. Regiones de origen: seleccione una lista de regiones para buscar los datos de telemetría que desee centralizar.

6. Especifique los detalles de destino mediante la configuración de los siguientes campos y, a continuación, seleccione Siguiente:

a. Cuenta de destino: seleccione una cuenta de la organización que sirva de destino central para los datos de telemetría.

b. Región de destino: seleccione una región principal que almacene una copia de los datos de telemetría centralizados.

c. Región de respaldo: de manera opcional, seleccione una región que almacene una copia de los datos de telemetría centralizados.

7. Especifique los detalles de telemetría mediante la configuración de los siguientes campos y, a continuación, seleccione Siguiente:

a. Grupos de registro: elija una de las siguientes opciones:

- Todos los grupos de registros: centralice los registros de todos los grupos de registros de las cuentas de origen.
- Filtrar grupo de registros: centralice los registros de un subconjunto de grupos de registros en las cuentas de origen según los criterios de selección. Se pueden proporcionar los criterios de selección de dos modos:
  - Builder: una experiencia basada en las elecciones para generar los criterios de selección
  - Editor: un cuadro de texto de formato libre para proporcionar los criterios de selección

  Hay dos criterios de selección que puede utilizar para filtrar los registros:

  - Criterios de selección de grupos de registros: los criterios de selección que especifican qué grupos de registros de origen se van a centralizar.
    - Claves compatibles: LogGroupName | *
    - Operadores compatibles: = | != | IN | NOT IN | AND | OR | LIKE | NOT LIKE
  - Criterios de selección de fuentes de datos: los criterios de selección que especifican qué fuentes de datos se van a centralizar.
    - Claves compatibles: | DataSourceName DataSourceType
    - Operadores compatibles: = | != | IN | NOT IN | AND | OR | LIKE | NOT LIKE

  Cuando se especifican los criterios de selección del grupo de registros y los criterios de selección de la fuente de datos, un evento de registro debe cumplir con ambos criterios para poder centralizarse.

b. Grupo de registros cifrados de KMS

> **Important**
>
> CloudWatch Las reglas de centralización no podrán entregar los registros de la cuenta de origen a los grupos de registros de destino si la clave de KMS proporcionada en la regla de centralización no permite que CloudWatch Logs la utilice. Si utiliza la clave KMS administrada por el cliente en sus grupos de registros de destino, añada la etiqueta LogsManaged = true a la clave kms. Para obtener más información, consulte Paso 2: establecer permisos en la clave de KMS.

Elija una de las siguientes opciones:

- Centralice los grupos de registros de origen cifrados con claves de KMS administradas por el cliente mediante una clave de KMS administrada por el cliente específica del destino: centralice los eventos de registro de los grupos de registros de origen cifrados con claves de KMS administradas por el cliente en grupos de registro de destino cifrados con una clave de KMS administrada por el cliente en la cuenta de destino.

  Cuando se selecciona esta opción, también se debe establecer lo siguiente:

  - Clave de cifrado de destino ARN: ARN de la clave KMS administrada por el cliente en la cuenta de destino y la región de destino principal, que se asociará a los grupos de registros de destino recién creados.
  - ARN de la clave de cifrado de destino de la copia de seguridad (si se selecciona la región de copia de seguridad): ARN de la clave KMS administrada por el cliente en la cuenta de destino y la región de destino de la copia de seguridad, que se asociará a los grupos de registros de destino recién creados.
  - Omita la centralización en grupos de registros de destino no cifrados (opcional): si ya existe un grupo de registros sin una clave KMS administrada por el cliente, CloudWatch no podrá actualizar su cifrado. Elija esta opción para omitir la centralización de los eventos de registro de los grupos de registros de origen cifrados con claves de KMS administradas por el cliente a grupos de registro de destino que no estén asociados a una clave de KMS administrada por el cliente.
- Centralice los grupos de registros cifrados con claves KMS administradas por el cliente en la cuenta de destino con AWS una clave KMS propia: centralice los eventos de registro de los grupos de registro de origen cifrados con claves KMS administradas por el cliente en grupos de registros de destino recién creados y cifrados con una clave KMS propia AWS .

- No centralice los grupos de registros cifrados con claves KMS administradas por el cliente: evite la centralización de los eventos de registro de los grupos de registro de origen cifrados con claves KMS administradas por el cliente.

8. Revise la regla de centralización, y de manera opcional, realice modificaciones de última hora y elija Crear política de centralización.

Modificación de una regla de centralización

Utilice el siguiente procedimiento para modificar una regla de centralización existente.

Modificación de una regla de centralización

1. Navegue hasta la CloudWatch consola de la cuenta de administración o de administrador delegado de la organización.
2. Seleccione Configuración.
3. Navegue a la pestaña Organización.
4. Elija Administrar reglas.
5. Seleccione la regla que se desea actualizar, y elija Editar.
6. Actualice la configuración de la regla según sea necesario y seleccione Siguiente para continuar con cada paso.
7. En el paso 4, Revisar y configurar, elija Actualizar la política de centralización.

Visualización de una regla de centralización

Utilice el siguiente procedimiento para ver una regla de centralización existente.

Visualización de una regla de centralización

1. Navegue hasta la CloudWatch consola de la cuenta de administración o administrador delegado de la organización.
2. Seleccione Configuración.
3. Navegue a la pestaña Organización.
4. Elija Administrar reglas.

5. Vea una lista de todas las reglas de centralización existentes y elija un nombre de regla específico para ver sus detalles.

Eliminación de una regla de centralización

Utilice el siguiente procedimiento para eliminar una regla de centralización existente.

Eliminación de una regla de centralización

1. Navegue hasta la CloudWatch consola de la cuenta de administración o administrador delegado de la organización.
2. Seleccione Configuración.
3. Navegue a la pestaña Organización.
4. Elija Administrar reglas.
5. Seleccione la regla que quiera eliminar y elija Eliminar.
6. Confirme la eliminación y elija Eliminar.

Supervisión y solución de problemas de las reglas de centralización

Puede supervisar el estado y el rendimiento de las reglas de centralización mediante CloudWatch métricas, la consola de CloudWatch registros y AWS CloudTrail los registros. Esto ayuda a garantizar que los datos de registro se replican correctamente y a identificar cualquier problema con la configuración de centralización.

CloudWatch Logs proporciona:

1. Estado de la regla según la regla de centralización
   a. Seleccione Configuración.
   b. Navegue a la pestaña Organización.
   c. Elija Administrar reglas.
2. Registra las llamadas a la API con AWS CloudTrail
3. CloudWatch también publica métricas para la centralización, como los eventos de registro replicados, los errores y la limitación. Para obtener más información sobre estas métricas y sus dimensiones, consulte. Métricas y dimensiones de centralización

Estado de salud de la regla de centralización

Cada regla de centralización tiene un estado de salud que indica si funciona correctamente. Puede verificar el estado de las reglas desde la consola o mediante programación mediante la API.

Los estados de salud reglamentarios incluyen:

- `HEALTHY`: la regla funciona con normalidad y replica los datos de registro según lo configurado
- `UNHEALTHY`: la regla ha detectado problemas y es posible que no esté replicando los datos correctamente
- `PROVISIONING`: la centralización de la organización está en proceso de creación.

Cuando una regla se marca como UNHEALTHY, el campo `FailureReason` proporciona detalles sobre el problema específico que debe abordarse.

Supervise las llamadas a la API de centralización con AWS CloudTrail

AWS CloudTrail registra las llamadas a la API realizadas al servicio de centralización, lo que le permite realizar un seguimiento de los cambios de configuración y solucionar los problemas de las cuentas que son miembros de su empresa. AWS Organizations

CloudTrail Los eventos clave de la centralización incluyen:

- `CreateCentralizationRuleForOrganization`: cuando se crea una nueva regla de centralización
- `UpdateCentralizationRuleForOrganization`: cuando se modifica una regla existente
- `DeleteCentralizationRuleForOrganization`: cuando se elimina una regla
- `GetCentralizationRuleForOrganization`: cuando se recuperan los detalles de la regla
- `ListCentralizationRulesForOrganization`: cuando las reglas aparecen en la lista

Puede utilizar CloudTrail los registros para auditar los cambios en la configuración de la centralización y correlacionarlos con problemas de rendimiento o errores de replicación.

Recomendaciones de supervisión

Para garantizar que la centralización funcione correctamente, recomendamos configurar CloudWatch alarmas en las métricas de centralización clave que vendemos a Metrics. CloudWatch Esta

supervisión proactiva le ayuda a detectar los problemas de forma temprana y a mantener una centralización de registros fiable en toda la organización.

Las métricas clave que se deben monitorear incluyen:

- `IncomingCopiedBytes`: Supervise el volumen de datos de registro que se replican correctamente en su cuenta de destino. Una caída repentina o la ausencia de esta métrica pueden indicar problemas de centralización.
- `CentralizationError`: Configure alarmas para detectar cualquier error en el proceso de centralización a fin de identificar y resolver los problemas rápidamente.
- `CentralizationThrottled`: Supervise los eventos de limitación que puedan afectar al rendimiento de la replicación de registros.

Para obtener una lista completa de las métricas de centralización disponibles y sus dimensiones, consulte. Métricas y dimensiones de centralización

Si los registros no se están centralizando como se esperaba, revise los siguientes escenarios comunes que pueden impedir la centralización de los registros.

Datos de registro históricos

La función de centralización de CloudWatch registros solo procesa los nuevos datos de registro que llegan a las cuentas de origen después de crear la regla de centralización. Los datos de registro históricos (registros que existían antes de la creación de la regla) no están centralizados.

Permisos de claves de KMS

Las reglas de centralización no podrán entregar los registros de la cuenta de origen a los grupos de registros de destino si la clave KMS proporcionada en la regla de centralización no permite que CloudWatch Logs la utilice. Asegúrese de que la política de claves de KMS conceda los permisos necesarios a CloudWatch los registros. Para obtener más información, consulte Paso 2: establecer permisos en la clave de KMS.

Configuración de claves de KMS gestionada por el cliente

Si seleccionó No centralizar los grupos de registros cifrados con la clave KMS administrada por el cliente durante la creación de la regla, los eventos de registro de los grupos de registros de origen cifrados con la clave KMS administrada por el cliente se omitirán y no se centralizarán.

El cifrado de destino no coincide

Si el grupo de registros de destino ya existe con una configuración de cifrado de KMS diferente a la que especifica la regla de centralización y la resolución de conflictos se establece en SKIP, los registros se eliminarán y se emitirá un `DestinationEncryptionMismatch` error. Por ejemplo, esto ocurre cuando el destino tiene el cifrado predeterminado, pero la regla especifica una clave KMS administrada por el cliente.

El acceso de confianza no está habilitado

El acceso de confianza debe estar habilitado AWS Organizations para CloudWatch que la cuenta de administración y la cuenta de destino puedan acceder a los datos de registro.

Criterios de selección de fuentes

Compruebe que los criterios de selección de fuentes de la regla de centralización estén configurados correctamente:

- Cuentas y regiones: asegúrese de que las cuentas y regiones de origen donde se originan los registros estén incluidas en la regla. Los grupos de registros de cuentas o regiones no especificadas en la regla no se centralizarán.
- Filtros de grupos de registros: si configuró filtros de grupos de registros, solo se centralizarán los grupos de registros que coincidan con los criterios especificados. Compruebe que los criterios de selección de grupos de registros incluyen los grupos de registros que espera centralizar.
- Pertenencia a la organización: tanto las cuentas de origen como las de destino deben pertenecer a la misma AWS Organizations organización. Las cuentas ajenas a la organización no pueden participar en la centralización.

Se alcanzó el límite de cuota de grupos de registros

Si la cuenta de destino ha alcanzado su límite de cuota de grupos de registros, no se pueden crear nuevos grupos de registros para centralizarlos. Compruebe que la cuenta de destino tenga una cuota suficiente para alojar grupos de registros centralizados de todas las cuentas de origen. Si es necesario, puede solicitar un aumento

Se ha superado el límite de longitud del nombre del flujo de registro

Los nombres de los flujos de registro tienen restricciones de longitud máxima. Cuando la centralización replica los flujos de registro en la cuenta de destino, se agrega un sufijo al nombre del flujo de registro. Si el nombre del flujo de registro resultante supera la longitud máxima

permitida, los registros se eliminarán y se emitirá un `InvalidLogStream` error en la cuenta del cliente.

Regla el estado de salud

Compruebe el estado de la regla de centralización en la consola o mediante la `GetCentralizationRuleForOrganization` API. Si la regla está marcada como insalubre, revisa el `FailureReason` campo para obtener detalles específicos sobre el problema.

Para diagnosticar problemas de centralización, revise el estado de la regla de centralización en la consola, compruebe si hay errores o limitaciones en CloudWatch las métricas y examine los AWS CloudTrail registros para detectar errores en las llamadas a la API. Para obtener más información sobre las métricas de centralización, consulte. Métricas y dimensiones de centralización

Uso de grupos de registro 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 registro es un grupo de flujos de registro que comparten la misma configuración de retención, monitoreo y control de acceso. Puede definir grupos de registro 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.

Para las organizaciones que necesitan consolidar los datos de registro de varias cuentas y regiones, puede utilizar la centralización de CloudWatch registros para replicar automáticamente los grupos de registros en una cuenta central. Para obtener más información, consulte Cross-account Centralización de registros entre regiones.

Puede usar los procedimientos de esta sección para trabajar con grupos y flujos de registros.

Cree un grupo de registros en Logs CloudWatch

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 registro

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Administración de registros.
3. Elija Actions (Acciones) y, a continuación, elija Create log group (Crear grupo de registro).
4. Escriba el nombre del grupo de registro y, a continuación, seleccione Crear grupo de registro.

> **Tip**
>
> Puede marcar como favoritos a grupos de registro, 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 registro que desea marcar como favoritos y elija el símbolo de estrella junto a este.

Enviar registros a un grupo de registro

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 Administración de registros.
3. En Log Groups (Grupos de registro), elija el grupo de registro para ver los flujos.
4. En la lista de grupos de registro, elija el nombre del grupo de registro 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.

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, consulte Observabilidad entre cuentas de CloudWatch .

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 Administración de registros.
3. En Log Groups (Grupos de registro), elija el nombre del grupo de registro 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 Administración de registros.
3. En Log Groups (Grupos de registro), elija el nombre del grupo de registro que contiene el flujo de registros que desea buscar.
4. Elija Search Log Group (Buscar grupos de registro).
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 el símbolo del sistema, ejecute el siguiente comando de la filter-log-events. 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 está utilizando 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 registro. Cualquier dato anterior a la configuración de retención actual se eliminará. Puede cambiar la retención de registro de cada grupo de registro 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, elija Logs (Registros), Log groups (Grupos de registro).
3. Busque el grupo de registro que desea actualizar.
4. En la columna Retención de ese grupo de registro, elija la configuración de retención actual; por ejemplo, No caducar nunca.
5. En Configuración de retención, en Marcar eventos como vencidos después de, elija un valor de retención de registros y, a continuación, seleccione Guardar.

Proteger los grupos de registros de la eliminación

Si lo desea, puede activar la protección contra la eliminación para evitar la eliminación accidental de grupos de registros importantes. Para obtener información detallada sobre la protección contra la eliminación, consulteProteger los grupos de registros de la eliminación.

Proteger los grupos de registros de la eliminación

Habilitar la protección contra la eliminación

Puede activar la protección contra la eliminación al crear un grupo de registros nuevo o en grupos de registros existentes. Durante la creación del grupo de registros, seleccione «Protección contra eliminación habilitada» o transfiera el parámetro`--deletion-protection-enabled`. De forma predeterminada, la protección contra la eliminación no está habilitada.

Para habilitar o deshabilitar la protección contra la eliminación en un grupo de registros existente (consola)

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Administración de registros.
3. Seleccione el grupo de registros que desee proteger.
4. Seleccione Acciones y edite la protección contra eliminaciones.
5. En el cuadro de diálogo, revise y, a continuación, envíe los cambios.

Si utiliza el AWS CLI, para habilitar la protección contra la eliminación en un grupo de registros existente:

```
aws logs put-log-group-deletion-protection \
--log-group-identifier "/my-application/logs" \
--deletion-protection-enabled
```

Para eliminar la protección contra la eliminación de un grupo de registros existente:

```
aws logs put-log-group-deletion-protection \
--log-group-identifier "/my-application/logs" \
--no-deletion-protection-enabled
```

Gestión de errores

Si intenta eliminar un grupo de registros con la protección contra eliminación habilitada, recibirá un mensaje `ValidationException` con el siguiente mensaje: «No se puede eliminar el grupo de registros con la protección contra eliminación habilitada». Desactive primero la protección contra la eliminación».

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 registro. 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 registro al crearlo.
- Agregar etiquetas a un grupo de registro existente.
- Enumerar las etiquetas para un grupo de registro.
- Eliminar las etiquetas de un grupo de registro.

Puede utilizar las etiquetas para categorizar los grupos de registro. 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 registro 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 registro 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 puedes empezar una clave de etiqueta con él `aws:` porque este prefijo está reservado para que lo usen. 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 el 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 registro. Si lo desea, puede agregar etiquetas al crear el grupo de registro.

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 registro. Si lo desea, puede agregar etiquetas al crear el grupo de registro.

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 con el Galois/Counter modo

estándar de cifrado avanzado (AES-GCM) de 256 bits para cifrar 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 clave. AWS KMS 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.

> **Important**
>
> CloudWatch Ahora es compatible con Logs`kms:ViaService`, lo que permite a los registros realizar AWS KMS llamadas en tu nombre. Debe añadir esto a sus funciones, que se llaman CloudWatch registros, ya sea en su política clave o en IAM. Para obtener más información, consulte kms: ViaService.

Después de asociar una clave de KMS con un grupo de registro, todos los datos ingeridos recientemente para el grupo de registro se cifran mediante la clave. Estos datos se almacenan en formato cifrado durante todo su período de retención. CloudWatch Logs descifra estos datos siempre 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 registro. 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 registro, 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 existente 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, se otorga permiso al director del servicio de CloudWatch registros y al rol de persona que llama para usar la clave. Esta entidad principal de servicio debe estar en la misma AWS región en la que se almacena la clave 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 registro.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
```

```
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:111122223333: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.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
```

```
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:123456789012:*"
                }
            }
        }
    ]
}
```

Paso 3: Defina los permisos del principal de IAM que realiza la llamada

Agregue permisos al principal de IAM (usuario o rol) que CloudWatch llamará a Logs. APIs Los permisos necesarios dependen de las operaciones que deba realizar el director. Puede añadir estos permisos en la política AWS KMS clave o mediante IAM en el propio rol. CloudWatch Registra `kms:ViaService` los usos para realizar llamadas AWS KMS en nombre del cliente. Para obtener más información, consulte kms: ViaService.

Permisos para asociar una clave KMS a un grupo de registros

El principal de IAM que llame `CreateLogGroup` con uno o varios `kmsKeyId` parámetros debe tener `kms:DescribeKey` permiso sobre la clave de KMS especificada. `AssociateKmsKey` Si la persona que llama no tiene este permiso, la llamada a la API fallará con un. `AccessDeniedException`

El siguiente ejemplo de declaración de política clave otorga los permisos mínimos necesarios para asociar una clave de KMS a un grupo de registros:

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::account_id:role/role_name"
  },
  "Action": [
    "kms:Describe*"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "logs.region.amazonaws.com"
      ]
    }
  }
}
```

Permisos para leer y escribir datos de registro cifrados

Si el principal de IAM también necesita leer o escribir datos de registro cifrados (por ejemplo, al llamar a `PutLogEvents` `GetLogEvents``FilterLogEvents`, o `StartQuery` en un grupo de registros cifrado con una clave KMS administrada por el cliente), necesitará AWS KMS permisos adicionales. El siguiente ejemplo de declaración de política clave otorga el conjunto completo de permisos necesarios tanto para asociar una clave como para leer o escribir datos de registro cifrados:

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::account_id:role/role_name"
  },
```

```
  "Action": [
    "kms:Encrypt",
    "kms:ReEncrypt*",
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:Describe*"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "logs.region.amazonaws.com"
      ]
    }
  }
}
```

Como práctica recomendada, aplique la política únicamente a las funciones que van a interactuar con los grupos de registros AWS KMS cifrados.

Como alternativa, si quieres conceder el conjunto completo de permisos tanto para asociar una clave como para leer o escribir datos de registro cifrados mediante IAM, puedes añadir la siguiente política a la función de persona que llama. Esto se puede añadir a una política de roles existente o adjuntarse a un rol como una política independiente adicional. Si utiliza este método, como práctica recomendada, aplique la política únicamente a AWS KMS las claves que se utilizarán para el cifrado de registros. Para obtener más información, consulte Editar políticas de IAM.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:ReEncrypt*",
                "kms:Decrypt",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Condition": {
```

```
                "StringEquals": {
                    "kms:ViaService": [
                        "logs.us-east-1.amazonaws.com"
                    ]
                }
            },
            "Resource": "arn:aws:kms:us-east-1:444455556666:key/key_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 4: Asocie una clave KMS a un grupo de registros

Puede asociar una clave de KMS a un grupo de registro al crearlo o posteriormente.

Para saber si un grupo de registros ya tiene una clave de KMS asociada, 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 registro 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 registro 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 5: desasociar la clave de un grupo de registros

Para desasociar la clave KMS asociada a un grupo de registros, utilice el siguiente comando: disassociate-kms-key

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

AWS KMS claves y contexto de cifrado

Para mejorar la seguridad de sus AWS Key Management Service claves y sus grupos de CloudWatch registros cifrados, Logs ahora incluye el grupo de registros ARNs como parte del contexto de cifrado utilizado para cifrar sus 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 registro que cifró anteriormente y ahora desea cambiar el grupo de registro para utilizar una nueva clave de KMS que funcione solo para ese grupo de registro, siga estos pasos.

Para convertir un grupo de registro cifrado a fin de utilizar una clave de KMS con una política que la limite a ese grupo de registro

1. Ingrese el siguiente comando para encontrar el ARN de la clave actual del grupo de registro:

   ```
   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:

   JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
```

```
                "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:111122223333: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 registro:

```
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`.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333: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 las políticas de protección de datos de los 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 registro.

> **Note**
>
> Se admite el enmascaramiento de datos confidenciales en los grupos de registros de las clases de registro estándar y de acceso poco frecuente. Para obtener más información acerca de las clases de registros, consulte Clases 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 gestionados, 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. Para algunos tipos de identificadores de datos administrados, la detección también depende de encontrar ciertas palabras clave que rodeen los datos confidenciales. También puede utilizar identificadores de datos personalizados para crear sus propios identificadores de datos adaptados a su 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`	*

Operación	Se necesita permiso de IAM	Recurso
Crea una política de protección de datos con CloudWatch Logs como destino de la auditoría	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`logs:PutResourcePolicy`	*
	`logs:DescribeResourcePolicies`	*
	`logs:DescribeLogGroups`	*
Creación de 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`	*

Operación	Se necesita permiso de IAM	Recurso
	`logs:CreateLogDelivery`	`*`
	`s3:GetBucketPolicy`	`arn:aws:s3::: YOUR_BUCKET`
	`s3:PutBucketPolicy`	`arn:aws:s3::: YOUR_BUCKET`
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 CloudWatch Logs como destino de auditoría	`logs:PutDataProtectionPolicy`
	`logs:CreateLogDelivery`
	`logs:PutResourcePolicy`
	`logs:DescribeResourcePolicies`
	`logs:DescribeLogGroups`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		`*`
		`*`
		`*`
		`*`

Creación de 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`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		`*`
		`arn:aws:s3:::` *YOUR_BUCKET*

Operación	Se necesita permiso de IAM	Recurso
	`s3:PutBucketPolicy`	`arn:aws:s3::: YOUR_BUCKET`
Unmask masked log events	`logs:Unmask`	`arn:aws:logs:::log-group: YOUR_LOG_GROUP :*`
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.

JSON

```
{
"Version":"2012-10-17",
"Statement": [
    {
        "Sid": "AllowLogDeliveryConfiguration",
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogDelivery",
            "logs:PutResourcePolicy",
```

```
                "logs:DescribeLogGroups",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowDataProtectionAndBucketConfiguration",
            "Effect": "Allow",
            "Action": [
                "logs:GetDataProtectionPolicy",
                "logs:DeleteDataProtectionPolicy",
                "logs:PutDataProtectionPolicy",
                "s3:PutBucketPolicy",
                "firehose:TagDeliveryStream",
                "s3:GetBucketPolicy"
            ],
            "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/delivery-
stream-name",
            "arn:aws:s3:::amzn-s3-demo-destination-bucket",
            "arn:aws:logs:us-east-1:111122223333:log-group:log-group-name:*"
            ]
        }
    ]
}
```

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 Identificadores de datos administrados, seleccione los tipos de datos que desea auditar y enmascarar para todos sus grupos de registro. 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 quiere auditar y enmascarar otros tipos de datos mediante identificadores de datos personalizados, seleccione 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 los eventos de registro. 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 define 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. Introduzca 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 registro, flujos de Amazon Data Firehose y buckets de S3, ya deben existir. 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.

En el siguiente ejemplo verá una política de protección de datos que utiliza solo identificadores de datos administrados. Esta política enmascara las direcciones de correo electrónico y los permisos de conducir de los Estados Unidos.

Para obtener información sobre las políticas que especifican identificadores de datos personalizados, consulte Uso 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, elija Logs (Registros), Log groups (Grupos de registro).
3. Elija el nombre del grupo de registro.
4. Elija Actions (Acciones), Create data protection policy (Crear política de protección de datos).
5. En Identificadores de datos administrados, seleccione los tipos de datos que desea auditar y enmascarar en este grupo de registro. 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 mediante los identificadores de datos administrados, consulte Tipos de datos que puede proteger.

6. (Opcional) Si quiere auditar y enmascarar otros tipos de datos mediante identificadores de datos personalizados, seleccione 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 los eventos de registro. 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 define 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. Introduzca 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 registro, flujos de Amazon Data Firehose y buckets de S3, ya deben existir. 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 Registra la sintaxis de consulta del lenguaje 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 servidor con las claves administradas de Amazon S3 (SSE-S3) o con el cifrado del servidor con claves de KMS (SSE-KMS). Para obtener más información, consulte Protección de los datos con el cifrado 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 utilizar identificadores de datos personalizados para crear sus propios 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 administrados y qué países y regiones son relevantes para cada tipo de datos.

En el caso de algunos tipos de datos confidenciales, la protección de datos de CloudWatch Logs 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, US
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.

Identificador	Categoría	Países e idiomas
NationalDrugCode	Estado	EE. UU.
NationalIdentificationNumber	Personal	DE, ES, IT
NationalInsuranceNumber	Personal	GB
NationalProviderId	Estado	EE. UU.
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`, `secret key`, `set-awscredential`	Todos
Clave privada de OpenSSH	`OpenSSHPrivateKey`	Ninguno	Todos
Clave privada de PGP	`PgpPrivateKey`	Ninguno	Todos
Clave privada de Pkcs	`PkcsPrivateKey`	Ninguno	Todos
Clave privada PuTTY	`PuttyPrivateKey`	Ninguno	Todos

Identificador de datos ARNs para los tipos de datos de credenciales

A continuación, se enumeran los nombres de recursos de Amazon (ARNs) para los identificadores de datos que puede añadir a sus políticas de protección de datos.

Identificador de datos de credenciales ARNs
`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 los registros 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`	Ninguno	Todos

Identificador de datos ARNs para los tipos de datos de los dispositivos

A continuación, se enumeran los nombres de recursos de Amazon (ARNs) 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	Francia, Alemania	Incluye números

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
		países. Para obtener más información, consulte la tabla de Palabras clave para números de cuentas bancarias que aparece más adelante en esta sección.	Italia, España, Reino Unido, Estados Unidos	de cuentas bancarias internacionales (IBANs) que constan de hasta 34 caracteres alfanuméricos e incluyen elementos como los códigos 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 tarjeta 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 siga la fórmula del cheque de Luhn y utilice un prefijo de número de tarjeta estándar para cualquiera de los siguiente

Tipo de datos	ID del identificador de datos	Palabra clave necesaria	Países y regiones	Notas
				s tipos de tarjetas de crédito: American Express, Dankort, Diner's Club, Discover, Electron, Japanese Card Bureau (JCB), Mastercard y Visa. UnionPay

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 los números de cuentas bancarias. Esto incluye los números de cuentas bancarias internacionales (IBANs) que constan de hasta 34 caracteres alfanuméricos, incluidos elementos como los 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.
```

Identificador de datos ARNs para los tipos de datos financieros

A continuación, se enumeran los nombres de recursos de Amazon (ARNs) para los identificadores de datos que puede añadir a sus políticas de protección de datos.

Identificador de datos financieros ARNs
`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 `, `sjukförsäkring nummer`, `sjukförsäkringskort `, `suomi ehic-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

Identificador de datos ARNs para los tipos de datos de información de salud protegida (PHI)

A continuación se muestra el identificador de datos Amazon Resource Names (ARNs) que se puede utilizar en las políticas de protección de datos de información médica protegida (PHI).

Identificador de datos PHI ARNs
`arn:aws:dataprotection::aws:data-identifier/DrugEnforcementAgencyNumber-US`
`arn:aws:dataprotection::aws:data-identifier/HealthcareProcedureCode-US`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceCardNumber-EU`

Identificador de datos PHI ARNs
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 los registros 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`	Ninguno	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`	Ninguno	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`	Ninguno	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`, `latitudelongitude` , `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`	Ninguno	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 VINs si 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)

Identificador de datos ARNs para la información de identificación personal (PII)

En la siguiente tabla se enumeran los nombres de recursos de Amazon (ARNs) para los identificadores de datos de información de identificación personal (PII) que puede añadir a sus políticas de protección de datos.

Identificador de datos de PII ARNs
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

Identificador de datos de PII ARNs
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

Identificador de datos de PII ARNs
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

Identificador de datos de PII ARNs
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 (CDIs) le permiten definir sus propias expresiones regulares personalizadas que se pueden utilizar en su 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, puede usar un identificador de datos personalizado para buscar un empleado específico de la empresa. IDs 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 registro. Igual que sucede con los identificadores de datos administrados, los identificadores de datos personalizados definidos en una política a nivel de cuenta funcionan junto a los identificadores de datos personalizados definidos en una política a nivel de grupo de registro.

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 enmascarará los eventos de registro con nueve números después de `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 la API 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 relacionados con un empleado específico de la empresa. IDs

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 que contengan `EmployeeID-` y nueve dígitos después de `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.

Transformación de los registros durante la ingestión

Con la transformación y el enriquecimiento de los registros, puede normalizar todos sus registros en un formato coherente y rico en contexto al momento de incorporarlos a los registros. CloudWatch Puede añadir estructura a sus registros mediante plantillas preconfiguradas para AWS servicios comunes, como AWS WAF Amazon Route 53, o crear transformadores personalizados con analizadores nativos, como Grok. También se puede cambiar el nombre de los atributos existentes y añadir metadatos adicionales a sus registros, como el ID de cuenta y la región.

La transformación de los registros ayuda a simplificar y acortar las consultas de registro en todas las aplicaciones y a simplificar la creación de alertas en los registros. Esta función proporciona la transformación de los tipos de registro más comunes con plantillas de out-of-the-box transformación para las principales fuentes de AWS registro, como los registros de flujo de VPC, Route 53 y. Amazon RDS for PostgreSQL Se pueden usar plantillas de transformación preconfiguradas o crear transformadores personalizados que se adapten a sus necesidades.

La transformación de registros ayuda a administrar los registros emitidos desde diversas fuentes, que varían considerablemente en cuanto al formato y los nombres de los atributos.

Tras crear un transformador, los eventos de registro ingeridos se convierten y almacenan en un formato estándar. Puede aprovechar estos registros transformados para acelerar su experiencia de análisis con las siguientes características:

- Índices de campo
- CloudWatch Registra los campos detectados por Insights
- Flexibilidad con las alarmas mediante filtros métricos
- Reenvío mediante filtros de suscripción
- Creación de datos métricos a partir de eventos de registro con Información de colaboradores, donde se puede elegir que la regla de Información de colaboradores evalúe los eventos de registro antes o después de su transformación.

Las transformaciones solo se producen durante la ingesta de registros. No se pueden transformar los eventos de registro que ya se hayan ingerido. Las transformaciones no son reversibles. Tanto los registros originales como los transformados se almacenan en CloudWatch registros con la misma política de retención. La capacidad de transformación y enriquecimiento de registros está incluida en el precio actual de adquisición de la clase de registro Estándar. Los costos de almacenamiento de

registros se basarán en el tamaño del registro después de la transformación, que puede superar el volumen de registro original.

> **Important**
>
> Una vez transformados los eventos de registro, debe utilizar las consultas de CloudWatch Logs Insights para ver las versiones transformadas de los registros. FilterLogEventsLas acciones GetLogEventsy devuelven solo las versiones originales de los eventos del registro, antes de que se transformaran.

> **Important**
>
> A pesar de que un solo evento de registro PutLogEvents permite hasta 1 MB, la transformación de registros solo puede gestionar eventos de registro de un tamaño inferior a 512 KB. Cualquier evento de registro que supere los 512 kb fallará en la transformación y emitirá un error. El tamaño total de aún PutLogEvents puede superar los 512 kb.

Además de transformarlos en diferentes formatos, también se pueden enriquecer sus registros con un contexto adicional, como el ID de cuenta, la región y la palabra clave. Se extraen del nombre del grupo de registros y de las palabras clave estáticas.

La transformación de registros ayuda con los registros emitidos desde diversas fuentes, que varían considerablemente en cuanto al formato y los nombres de los atributos.

La transformación y el enriquecimiento de registro solo son compatibles con los grupos de registro de la clase de registro Estándar.

Se pueden crear transformadores para grupos de registro individuales y, también se pueden crear transformadores de cuenta que se apliquen a todos o a varios grupos de registro de su cuenta. Si un grupo de registros tiene un transformador a nivel de grupo de registros, ese transformador anula cualquier transformador a nivel de cuenta que, de otro modo, se aplicaría a ese grupo de registros.

Temas

- Creación y administración de transformadores de registro
- Procesadores configurables tipo analizador
- Procesadores integrados para registros vendidos AWS

- Procesadores de mutación de cadena
- Procesadores de mutación JSON
- Procesadores convertidores de tipos de datos
- Métricas y errores de transformación

Creación y administración de transformadores de registro

Un transformador de registro incluye uno o más procesadores que están juntos en una canalización lógica. Cada procesador se aplica a un evento de registro, uno tras otro, en el orden en que aparecen en la configuración del transformador.

Algunos procesadores son del tipo analizador. Cada transformador debe tener al menos un analizador y el primer procesador de un transformador debe ser un analizador.

Algunos de los analizadores están integrados y configurados para un determinado tipo de registro ofrecido de AWS .

Otros tipos de procesadores son los mutadores de cadena, los mutadores JSON y los procesadores de datos.

Se pueden crear transformadores para grupos de registro individuales y, también, se pueden crear transformadores a nivel de cuenta que se apliquen a todos o a varios grupos de registro de su cuenta. Si un grupo de registros tiene un transformador a nivel de grupo de registros, ese transformador anula cualquier transformador a nivel de cuenta que, de otro modo, se aplicaría a ese grupo de registros. Puede tener hasta 20 transformadores a nivel de cuenta en una región de su cuenta.

Al crear un transformador, se deben seguir estas directrices:

- Si incluye un analizador preconfigurado para un tipo de AWS registros vendidos, debe ser el primer procesador de la lista del transformador. Se puede incluir solo un procesador de este tipo en un transformador.
- Se puede incluir solo un procesador `grok` en un transformador.
- Se debe tener al menos un procesador de tipo analizador en un transformador. Se pueden incluir hasta cinco procesadores de tipo analizador. Este límite de cinco incluye tanto los analizadores integrados como los configurables.
- Se pueden tener hasta 20 procesadores en un transformador.

- Solo se puede incluir solo un procesador addKeys en un transformador.
- Se puede incluir solo un procesador copyValue en un transformador.
- Cada transformador puede extraer hasta 200 campos de un evento de registro.
- Cada evento de registro DEBE ser inferior a 512 KB. El tamaño total del registro de eventos aún puede superar los 512 KB.

Temas

Creación de una política de transformadores a nivel de cuenta

Se utilizan los pasos de esta sección para crear una política de transformador que se aplique a todos los grupos de registros de la cuenta o a varios grupos de registros que tengan nombres de grupos de registros que comiencen por la misma cadena. Se pueden tener hasta 20 transformadores a nivel de cuenta en una región.

No se pueden crear dos políticas de transformadores en la misma región que usen el mismo prefijo o que contengan un prefijo dentro de otro. Por ejemplo, si se crea una política de transformador para el prefijo de cadena`/aws/lambda`, no se podrá crear otra con ese prefijo `/aws`. Pero se podría tener un transformador para `/aws/lambda` y otro para `/aws/waf`

Creación de una política de transformadores a nivel de cuenta

1. Abra la CloudWatch consola en. https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación izquierdo, elija Configuración y, a continuación, la pestaña Registros.
3. En la sección Política de transformadores para cuentas, seleccione Creación de política de transformadores.
4. En Nombre de la política del transformador, escriba un nombre para la política nueva.
5. En Selección de grupos de registros, realice una de las operaciones siguientes:

- Seleccione Todos los grupos de registros estándar para que la política de transformador se aplique a todos los grupos de registros de clase Standard de la cuenta.
- Elija Grupos de registros por coincidencia de prefijos para aplicar la política a un subconjunto de grupos de registros cuyos nombres comiencen por la misma cadena. A continuación, introduzca el prefijo de estos grupos de registros en los criterios de selección.

6. En el área Seleccionar analizadores, use Analilzadores para seleccionar un analizador e incluirlo en su transformador.

   Si se trata de un analizador preconfigurado para un tipo de AWS registro vendido, no es necesario que especifique ninguna configuración para él.

   Si es un analizador diferente, se debe especificar su configuración. Para obtener más información, consulte la información de ese procesador en Procesadores configurables tipo analizador.

7. Para añadir otro procesador, elija Seleccionar procesador. A continuación, seleccione el procesador que desee en el cuadro Procesador y rellene los parámetros de configuración.

   Recuerde que los procesadores funcionan con los eventos de registro en el orden en que se agreguen al transformador.

8. (Opcional) Para añadir procesadores adicionales, elija + Procesador y repita el paso anterior.
9. (Opcional) En cualquier momento, se puede probar el transformador que ha construido hasta ahora a partir de un ejemplo de evento de registro. Para ello, lleve a cabo una de las siguientes acciones en la sección Vista previa de transformador:
   - Seleccione hasta cinco grupos de registros en Seleccionar grupos de registros y, a continuación, elija Cargar los últimos eventos de registro. A continuación, elija Probar transformador.
   - Copie los eventos del registro directamente en Registro de eventos de muestra y, a continuación, elija Probar transformador.

   A continuación, aparece la versión transformada del registro.

10. Cuando se hayan terminado de añadir procesadores y esté satisfecho con las pruebas realizadas en los registros de muestra, seleccione Guardar.
11. Cuando haya terminado, seleccione Create (Crear).

Edición o eliminación de una política de transformador a nivel de cuenta

Siga los pasos de esta sección para editar o eliminar una política de transformador a nivel de cuenta.

Edición o eliminación de una política de transformador a nivel de cuenta

1. Abra la consola en CloudWatch . https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación izquierdo, elija Configuración y, a continuación, la pestaña Registros.
3. En la sección Política de cuenta de transformador, elija Administrar.
4. Seleccione el botón situado junto a la política de transformadores que desee administrar y, a continuación, elija Editar o Eliminar.

   Si se va a editar la política, consulte los pasos 5 a 11 en Procesadores configurables tipo analizador para ver las opciones.

Cree un transformador de log-group-level registros desde cero

Sigue estos pasos para crear un log-group-level transformador desde cero.

Uso de la consola para crear un transformador de registros para un grupo de registro

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros), Log groups (Grupos de registro).
3. Seleccione el grupo de registro para el que desea crear un transformador.
4. Elija la pestaña Transformador. Es posible que tenga que desplazarse por la lista de pestañas hacia la derecha para verla.
5. Elija Creación de transformador.
6. En el cuadro Elección de un analizador, seleccione un analizador para incluirlo en su transformador.

   Si se trata de un analizador preconfigurado para un tipo de AWS registro vendido, no es necesario que especifique ninguna configuración para él.

   Si es un analizador diferente, se debe especificar su configuración. Para obtener más información, consulte la información de ese procesador en Procesadores configurables tipo analizador.

7. Para añadir otro procesador, seleccione + Añadir procesador. A continuación, seleccione el procesador que desee en el cuadro Elección de procesadores y rellene los parámetros de configuración.

   Recuerde que los procesadores funcionan con los eventos de registro en el orden en que se agreguen al transformador.

8. (Opcional) En cualquier momento, se puede probar el transformador que ha construido hasta ahora a partir de un ejemplo de evento de registro. Para ello, haga lo siguiente:

   - En la sección Vista previa de la transformación, elija Carga de registro de muestra para cargar un evento de registro de muestra del grupo de registros al que pertenece este transformador o pegue un evento de registro en el cuadro de texto.

     Elija Probar transformador. Aparece la versión transformada del registro.

9. Cuando se hayan terminado de añadir procesadores y esté satisfecho con las pruebas realizadas en los registros de muestra, seleccione Guardar.

Para usar el AWS CLI para crear un transformador de registro desde cero

- Utilice el comando `aws logs put-transformer`. Si se utiliza `parseJSON` como primer procesador, debe analizar todo el evento de registro con `@message` como el campo de origen. Tras el análisis inicial de JSON, se pueden manipular campos específicos en los procesadores posteriores. El siguiente es un ejemplo que crea un transformador que incluye los procesadores `parseJSON` y `addKeys`:

```
 aws logs put-transformer \
  --transformer-config '[{"parseJSON":{"source":"@message"}},{"addKeys":
{"entries":[{"key":"metadata.transformed_in","value":"CloudWatchLogs"},
{"key":"feature","value":"Transformation"}]}},{"trimString":{"withKeys":
["status"]}}]' \
  --log-group-identifier my-log-group-name
```

Cree un log-group-level transformador copiando uno existente

Se puede utilizar la consola para copiar la configuración JSON de un transformador existente. Luego, puede usar ese código para crear un transformador idéntico utilizando el AWS CLI, o puede modificar primero la configuración.

Creación de un transformador de registro copiando uno existente

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros), Log groups (Grupos de registro).
3. Elija el grupo de registro que contiene el transformador que se desea copiar.
4. Seleccione la pestaña Transformaciones. Es posible que tenga que desplazarse por la lista de pestañas hacia la derecha para verla.
5. Elija Administración de transformador.
6. Elija Copia de transformador. De esta forma se copiará el JSON del transformador en el portapeles.
7. Cree un archivo y péguelo en la configuración del transformador. En este ejemplo, llamaremos el archivo `CopiedTransformer.json`.
8. Utilice el AWS CLI para crear un nuevo transformador con esa configuración.

```
aws logs put-transformer --log-group-identifier my-log-group-name \
--transformer-config file://CopiedTransformer.json
```

Edite un log-group-level transformador

Siga los siguientes pasos para editar un transformador de registro existente.

Edición de un transformador de registro

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros), Log groups (Grupos de registro).
3. Elija el grupo de registro que contiene el transformador que desea editar.
4. Seleccione la pestaña Transformaciones. Es posible que tenga que desplazarse por la lista de pestañas hacia la derecha para verla.
5. Elija Administración de transformador.
6. En las secciones Analizadores y Procesadores, realice los cambios que desee.
7. Para añadir otro procesador, seleccione + Añadir procesador. A continuación, seleccione el procesador que desee en el cuadro Procesador y rellene los parámetros de configuración.

   Recuerde que los procesadores funcionan con los eventos de registro en el orden en que se agreguen al transformador.

8. (Opcional) En cualquier momento, se puede probar el transformador que ha construido hasta ahora a partir de un ejemplo de evento de registro. Para ello, haga lo siguiente:

   - En la sección Vista previa de la transformación, elija Carga de registro de muestra para cargar un evento de registro de muestra del grupo de registros al que pertenece este transformador o pegue un evento de registro en el cuadro de texto.

     Elija Probar transformación. Aparece la versión transformada del registro.
9. Cuando se hayan terminado de añadir procesadores y esté satisfecho con las pruebas realizadas en los registros de muestra, seleccione Guardar.

Eliminar un log-group-level transformador

Siga estos pasos para eliminar un transformador de registro.

Eliminación de un transformador de registros

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación, elija Logs (Registros), Log groups (Grupos de registro).
3. Elija el grupo de registro que contiene el transformador que desea editar.
4. Seleccione la pestaña Transformaciones. Es posible que tenga que desplazarse por la lista de pestañas hacia la derecha para verla.
5. Elija Eliminar.
6. En el cuadro de confirmación, elija Política de eliminación.

Procesadores configurables tipo analizador

Esta sección contiene información sobre los procesadores analizadores de datos configurables que puede usar en un transformador de eventos de registro.

Contenido

parseJSON

El procesador parseJSON analiza los eventos del registro JSON e inserta los pares clave-valor JSON extraídos en el destino. Si no se especifica un destino, el procesador coloca el par clave-valor debajo del nodo raíz. Si se utiliza `parseJSON` como primer procesador, debe analizar todo el evento de registro con `@message` como el campo de origen. Tras el análisis inicial de JSON, se pueden manipular campos específicos en los procesadores posteriores.

El contenido `@message` original no se modifica, las nuevas claves se añaden al mensaje.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
origen	Ruta al campo del evento de registro que se analizará. Utilice la notación de puntos para acceder a los campos secundarios. Por ejemplo, `store.book`	No	`@message`	Longitud máxima: 128 Profundidad máxima de clave anidada: 3
destination	El campo de destino de JSON analizado	No	`Parent JSON node`	Longitud máxima: 128 Profundidad máxima de clave anidada: 3

Ejemplo

Supongamos que un evento de registro ingerido tiene un aspecto similar al siguiente:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

Entonces, si tenemos este procesador parseJSON:

```
[
   {
        "parseJSON": {
            "destination": "new_key"
        }
   }
]
```

El evento de registro transformado sería el siguiente.

```
{
    "new_key": {
        "outer_key": {
            "inner_key": "inner_value"
        }
    }
}
```

grok

Utilice el procesador grok para analizar y estructurar los datos no estructurados mediante la coincidencia de patrones. Este procesador también puede extraer campos de los mensajes de registro.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites	Notas
origen	Ruta del campo al que se va a aplicar la coincidencia de grok	No	`@messag`	Longitud máxima: 128	

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites	Notas
				Profundidad máxima de clave anidada: 3	

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites	Notas
emparejar	El patrón grok que coincide con el evento de registro	Sí		Longitud máxima: 512 Patrones grok máximos: 20 Algunos tipos de patrones grok tienen límites de uso individuales. Se puede usar cualquier combinación de los siguientes patrones hasta cinco veces: {URI, URIPARAM, URIPATHPARAM, SPACE, DATA, GREEDYDATA, GREEDYDATA_MULTILINE} Los patrones grok no admiten conversiones de tipos. Para los patrones de formato de registro comunes (APACHE_ACCESS_LOG, NGINX_ACCESS_LOG SYSLOG542	Consulte todos los patrones grok compatibles

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites	Notas
				4), solo se admite incluir los patrones DATA, GREEDYDATA o GREEDYDATA_MULTILINE después del patrón de registro común.	

Estructura de un patrón grok

Esta es la estructura del patrón grok compatible:

```
%{PATTERN_NAME:FIELD_NAME}
```

- PATTERN_NAME: hace referencia a una expresión regular predefinida para hacer coincidir un tipo específico de datos. Solo se admiten los patrones grok predefinidos. No se permite crear patrones personalizados.
- FIELD_NAME: asigna un nombre al valor extraído. `FIELD_NAME` es opcional, pero si no especifica este valor, los datos extraídos se eliminarán del evento de registro transformado. Si `FIELD_NAME` utiliza una notación punteada (p. ej., “parent.child”), se considera una ruta JSON.
- Conversión de tipos: no se admiten las conversiones de tipos explícitos. Utilice el TypeConverter procesador para convertir el tipo de datos de cualquier valor extraído por grok.

Para crear expresiones coincidentes más complejas, puede combinar varios patrones grok. Se pueden combinar hasta 20 patrones grok para que coincidan con un evento de registro. Por ejemplo, esta combinación de patrones `%{NUMBER:timestamp} [%{NUMBER:db} %{IP:client_ip}: %{NUMBER:client_port}] %{GREEDYDATA:data}` se puede usar para extraer campos de una entrada de registro lenta de Redis como esta:

```
1629860738.123456 [0 127.0.0.1:6379] "SET" "key1" "value1"
```

Ejemplos de grok

Ejemplo 1: Utilice grok para extraer un campo de registros no estructurados

Registros de ejemplo:

```
293750 server-01.internal-network.local OK "[Thread-000] token generated"
```

Transformador utilizado:

```
[
     {
         "grok": {
             "match": "%{NUMBER:version} %{HOSTNAME:hostname} %{NOTSPACE:status}
 %{QUOTEDSTRING:logMsg}"
         }
    }
]
```

Salida:

```
{
  "version": "293750",
  "hostname": "server-01.internal-network.local",
  "status": "OK",
  "logMsg": "[Thread-000] token generated"
}
```

Registros de ejemplo:

```
23/Nov/2024:10:25:15 -0900 172.16.0.1 200
```

Transformador utilizado:

```
[
    {
        "grok": {
            "match": "%{HTTPDATE:timestamp} %{IPORHOST:clientip}
 %{NUMBER:response_status}"
        }
    }
```

```
]
```

Salida:

```
{
  "timestamp": "23/Nov/2024:10:25:15 -0900",
  "clientip": "172.16.0.1",
  "response_status": "200"
}
```

Ejemplo 2: Utilice grok en combinación con parseJSON para extraer campos de un evento de registro JSON

Registros de ejemplo:

```
{
    "timestamp": "2024-11-23T16:03:12Z",
    "level": "ERROR",
    "logMsg": "GET /page.html HTTP/1.1"
}
```

Transformador utilizado:

```
[
     {
        "parseJSON": {}
    },
    {
         "grok": {
            "source": "logMsg",
             "match": "%{WORD:http_method} %{NOTSPACE:request} HTTP/
%{NUMBER:http_version}"
         }
    }
]
```

Salida:

```
{
  "timestamp": "2024-11-23T16:03:12Z",
  "level": "ERROR",
```

```
  "logMsg": "GET /page.html HTTP/1.1",
  "http_method": "GET",
  "request": "/page.html",
  "http_version": "1.1"
}
```

Ejemplo 3: patrón grok con anotación punteada en FIELD_NAME

Registros de ejemplo:

```
192.168.1.1 GET /index.html?param=value 200 1234
```

Transformador utilizado:

```
[
    {
        "grok": {
            "match": "%{IP:client.ip} %{WORD:method} %{URIPATHPARAM:request.uri}
 %{NUMBER:response.status} %{NUMBER:response.bytes}"
        }
    }
]
```

Salida:

```
{
  "client": {
    "ip": "192.168.1.1"
  },
  "method": "GET",
  "request": {
    "uri": "/index.html?param=value"
  },
  "response": {
    "status": "200",
    "bytes": "1234"
  }
}
```

Patrones grok admitidos

En las siguientes tablas se enumeran los patrones que admite el procesador `grok`.

Patrones grok generales

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
NOMBRE DE USUARIO o USUARIO	Coincide con uno o más caracteres que pueden incluir letras minúsculas (a-z), letras mayúsculas (A-Z), dígitos (0-9), puntos (.), guiones bajos (_) o guiones medios (-)	20	Entrada: `user123.name-TEST` Patrón: `%{USERNAME:name}` Salida: `{"name": "user123.name-TEST"}`
INT	Coincide con un signo más o menos opcional seguido de uno o más dígitos.	20	Entrada: `-456` Patrón: `%{INT:num}` Salida: `{"num": "-456"}`
BASE10NUM	Coincide con un número entero o un número de punto flotante con signo y coma decimal opcionales	20	Entrada: `-0.67` Patrón: `%{BASE10NUM:num}` Salida: `{"num": "-0.67"}`
BASE16NUM	Hace coincidir los números decimales y hexadecimales con un signo opcional (+ o -) y un prefijo 0x opcional	20	Entrada: `+0xA1B2` Patrón: `%{BASE16NUM:num}` Salida: `{"num": "+0xA1B2"}`
POSINT	Coincide con números enteros positivos sin ceros a la izquierda, compuestos por uno o más dígitos (del 1 al 9 seguido del 0 al 9)	20	Entrada: `123` Patrón: `%{POSINT:num}` Salida: `{"num": "123"}`
NONNEGINT	Coincide con cualquier número entero (formado por uno o más dígitos del 0 al 9),	20	Entrada: `007` Patrón: `%{NONNEGINT:num}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
	incluido el cero y los números con ceros a la izquierda.		Salida: `{"num": "007"}`
WORD	Coincide con palabras completas compuestas por uno o más caracteres verbales (\w), incluidas letras, dígitos y guiones bajos	20	Entrada: `user_123` Patrón: `%{WORD:user}` Salida: `{"user": "user_123"}`
NOTSPACE	Coincide con uno o más caracteres que no sean espacios en blanco.	5	Entrada: `hello_world123` Patrón: `%{NOTSPACE:msg}` Salida: `{"msg": "hello_world123"}`
SPACE	Coincide con 0 o más caracteres que son espacios en blanco.	5	Entrada: `" "` Patrón: `%{SPACE:extra}` Salida: `{"extra": " "}`
DATA	Coincide con cualquier carácter (excepto la línea nueva) cero o más veces, no es codicioso.	5	Entrada: `abc def ghi` Patrón: `%{DATA:x} %{DATA:y}` Salida: `{"x": "abc", "y": "def ghi"}`
GREEDYDATA	Coincide con cualquier carácter (excepto la línea nueva) cero o más veces, es codicioso.	5	Entrada: `abc def ghi` Patrón: `%{GREEDYDATA:x} %{GREEDYDATA:y}` Salida: `{"x": "abc def", "y": "ghi"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo

GREEDYDATA_MULTILINE	Coincide con cualquier carácter (incluida la línea nueva) cero o más veces, es codicioso.	1	Input:
			`abc`
			`def`
			`ghi`
			Patrón: `%{GREEDYDATA_MULTILINE:data}`
			Salida: `{"data": "abc\ndef\nghi"}`

QUOTEDSTRING	Busca cadenas entre comillas (comillas simples o dobles) con caracteres de escape.	20	Entrada: `"Hello, world!"` Patrón: `%{QUOTEDSTRING:msg}` Salida: `{"msg": "Hello, world!"}`
UUID	Coincide con un formato UUID estándar: 8 caracteres hexadecimales, seguidos de tres grupos de 4 caracteres hexadecimales y termina con 12 caracteres hexadecimales, todos separados por guiones medios.	20	Entrada: `550e8400-e29b-41d4-a716-446655440000` Patrón: `%{UUID:id}` Salida: `{"id": "550e8400-e29b-41d4-a716-446655440000"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
URN	Coincide con la sintaxis URN (nombre uniforme de recurso).	20	Entrada: `urn:isbn:0451450523` Patrón: `%{URN:urn}` Salida: `{"urn": "urn:isbn:0451450523"}`

AWS patrones de grok

Patrón	Description (Descripción)	Límite de patrones	Ejemplo
ARN	Hace coincidir los nombres de los recursos de AWS Amazon (ARNs) y captura la partición (`awsaws-cn`, `oaws-us-gov`), el servicio, la región, el ID de cuenta y hasta 5 identificadores jerárquicos de recursos separados por barras diagonales. No coincidirá con la información ARNs que falte entre dos puntos.	5	Entrada: `arn:aws:iam:us-east-1:123456789012:user/johndoe` Patrón: `%{ARN:arn}` Salida: `{"arn": "arn:aws:iam:us-east-1:123456789012:user/johndoe"}`

Patrones de grok de redes

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
CISCOMAC	Coincide con una dirección MAC en formato hexadecimal 4-4-4.	20	Entrada: `0123.4567.89AB`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
			Patrón: `%{CISCOMAC:MacAddress}` Salida: `{"MacAddress": "0123.4567.89AB"}`
WINDOWS/MAC	Coincide con una dirección MAC en formato hexadecimal con guiones medios	20	Entrada: `01-23-45-67-89-AB` Patrón: `%{WINDOWSMAC:MacAddress}` Salida: `{"MacAddress": "01-23-45-67-89-AB"}`
COMMONMAC	Coincide con una dirección MAC en formato hexadecimal con dos puntos.	20	Entrada: `01:23:45:67:89:AB` Patrón: `%{COMMONMAC:MacAddress}` Salida: `{"MacAddress": "01:23:45:67:89:AB"}`
MAC	Coincide con uno de los patrones grok de CISCOMAC, WINDOWSMAC o COMMONMAC	20	Entrada: `01:23:45:67:89:AB` Patrón: `%{MAC:m1}` Salida: `{"m1":"01:23:45:67:89:AB"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
IPV6	Coincide con IPv6 las direcciones, incluidos los formularios comprimidos y las direcciones IPv4 mapeadas IPv6 .	5	Entrada: `2001:db8:3333:4444:5555:6666:7777:8888` Patrón: `%{IPV6:ip}` Salida: `{"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}`
IPV4	Coincide con una IPv4 dirección.	20	Entrada: `192.168.0.1` Patrón: `%{IPV4:ip}` Salida: `{"ip": "192.168.0.1"}`
IP	Coincide con IPv6 las direcciones admitidas por% {IPv6} o con IPv4 las direcciones admitidas por% {IPv4}	5	Entrada: `192.168.0.1` Patrón: `%{IP:ip}` Salida: `{"ip": "192.168.0.1"}`
HOSTNAME o HOST	Coincide con los nombres de dominio, incluidos los subdominios	5	Entrada: `server-01.internal-network.local` Patrón: `%{HOST:host}` Salida: `{"host": "server-01.internal-network.local"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
IPORHOST	Coincide con un nombre de host o una dirección IP	5	Entrada: `2001:db8:3333:4444:5555:6666:7777:8888` Patrón: `%{IPORHOST:ip}` Salida: `{"ip": "2001:db8:3333:4444:5555:6666:7777:8888"}`
HOSTPORT	Coincide con una dirección IP o un nombre de host, tal como lo admite el patrón %{IPORHOST} seguido de dos puntos y un número de puerto, y captura el puerto como “PORT” en el resultado.	5	Entrada: `192.168.1.1:8080` Patrón: `%{HOSTPORT:ip}` Salida: `{"ip":"192.168.1.1:8080","PORT":"8080"}`
URIHOST	Coincide con una dirección IP o un nombre de host, tal como lo admite el patrón %{IPORHOST}, con opción de seguido de dos puntos y un número de puerto, y captura el puerto como “PORT” en el resultado.	5	Entrada: `example.com:443 10.0.0.1` Patrón: `%{URIHOST:host} %{URIHOST:ip}` Salida: `{"host":"example.com:443","port":"443","ip":"10.0.0.1"}`

Patrones grok de rutas

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
UNIXPATH	Coincide con las rutas URL y puede incluir parámetros de consulta.	20	Entrada: `/search?q=regex` Patrón: `%{UNIXPATH:path}` Salida: `{"path":"/search?q=regex"}`
WINPATH	Coincide con las rutas de archivos de Windows.	5	Entrada: `C:\Users\John\Documents\file.txt` Patrón: `%{WINPATH:path}` Salida: `{"path": "C:\\Users\\John\\Documents\\file.txt"}`
PATH	Coincide con las rutas URL o de los archivos de Windows	5	Entrada: `/search?q=regex` Patrón: `%{PATH:path}` Salida: `{"path":"/search?q=regex"}`
TTY	Coincide con las rutas de los dispositivos Unix para terminales y pseudoterminales.	20	Entrada: `/dev/tty1` Patrón: `%{TTY:path}` Salida: `{"path":"/dev/tty1"}`
URIPROTO	Coincide con las letras, seguidas opcionalmente por un carácter más (+) y letras adicionales o caracteres más (+)	20	Entrada: `web+transformer` Patrón: `%{URIPROTO:protocol}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
			Salida: `{"protocol":"web+transformer"}`
URIPATH	Coincide con el componente de ruta de un URI	20	Entrada: `/category/sub-category/product_name` Patrón: `%{URIPATH:path}` Salida: `{"path":"/category/sub-category/product_name"}`
URIPARAM	Coincide con parámetros de consulta de URL	5	Entrada: `?param1=value1&param2=value2` Patrón: `%{URIPARAM:url}` Salida: `{"url":"?param1=value1&param2=value2"}`
URIPATHPARAM	Coincide con una ruta URI seguida, opcionalmente, de parámetros de consulta	5	Entrada: `/category/sub-category/product?id=12345&color=red` Patrón: `%{URIPATHPARAM:path}` Salida: `{"path":"/category/sub-category/product?id=12345&color=red"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
URI	Coincide con un URI completo	5	Entrada: `https://user:password@example.com/path/to/resource?param1=value1&param2=value2` Patrón: `%{URI:uri}` Salida: `{"path":"https://user:password@example.com/path/to/resource?param1=value1&param2=value2"}`

Patrones grok de fecha y hora

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
MONTH	Hace coincidir los nombres de los meses completos o abreviados en inglés como palabras completas	20	Entrada: `Jan` Patrón: `%{MONTH:month}` Salida: `{"month": "Jan"}` Entrada: `January` Patrón: `%{MONTH:month}` Salida: `{"month": "January"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
MONTHNUM	Coincide con los números de los meses del 1 al 12, con el cero inicial opcional para los meses de un solo dígito.	20	Entrada: `5` Patrón: `%{MONTHNUM:month}` Salida: `{"month": "5"}` Entrada: `05` Patrón: `%{MONTHNUM:month}` Salida: `{"month": "05"}`
MONTHNUM	Coincide con números mensuales de dos dígitos del 01 al 12.	20	Entrada: `05` Patrón: `%{MONTHNUM2:month}` Salida: `{"month": "05"}`
MES/DÍA	Coincide con el día del mes del 1 al 31, con el cero inicial opcional.	20	Entrada: `31` Patrón: `%{MONTHDAY:monthDay}` Salida: `{"monthDay":"31"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
YEAR	Coincide con el año en dos o cuatro dígitos	20	Entrada: `2024` Patrón: `%{YEAR:year}` Salida: `{"year":"2024"}` Entrada: `24` Patrón: `%{YEAR:year}` Salida: `{"year":"24"}`
DAY	Coincide con los nombres de los días completos o abreviados.	20	Entrada: `Tuesday` Patrón: `%{DAY:day}` Salida: `{"day":"Tuesday"}`
HOUR	Coincide con la hora en formato de 24 horas con un cero (0) 0-23 a la izquierda opcional.	20	Entrada: `22` Patrón: `%{HOUR:hour}` Salida: `{"hour":"22"}`
MINUTE	Coincide con los minutos (00-59).	20	Entrada: `59` Patrón: `%{MINUTE:min}` Salida: `{"min":"59"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
SECOND	Coincide con un número que representa los segundos (0)0-60, seguido opcionalmente por una coma decimal o dos puntos y uno o más dígitos para las fracciones de minutos	20	Entrada: `3` Patrón: `%{SECOND:second}` Salida: `{"second":"3"}` Entrada: `30.5` Patrón: `%{SECOND:minSec}` Salida: `{"minSec":"30.5"}` Entrada: `30:5` Patrón: `%{SECOND:minSec}` Salida: `{"minSec":"30:5"}`
TIME	Coincide con un formato de hora con horas, minutos y segundos en el formato (H)H:mm:(s)s. Los segundos incluyen los segundos intercalares (0)0-60.	20	Entrada: `09:45:32` Patrón: `%{TIME:time}` Salida: `{"time":"09:45:32"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
DATE_US	Coincide con una fecha con el formato de (M)M/(d)d/(yy)yy or (M)M-(d)d-(yy)yy.	20	Entrada: `11/23/2024` Patrón: `%{DATE_US:date}` Salida: `{"date":"11/23/2024"}` Entrada: `1-01-24` Patrón: `%{DATE_US:date}` Salida: `{"date":"1-01-24"}`
DATE_EU	Coincide con la fecha en el formato de (d)d/(M)M/(yy)yy, (d)d-(M)M-(yy)yy, or (d)d.(M)M.(yy)yy.	20	Entrada: `23/11/2024` Patrón: `%{DATE_EU:date}` Salida: `{"date":"23/11/2024"}` Entrada: `1.01.24` Patrón: `%{DATE_EU:date}` Salida: `{"date":"1.01.24"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
ISO8601_ZONA HORARIA	Coincide con el desplazamiento UTC “Z” o el desplazamiento de zona horaria con dos puntos opcionales en el formato [+-] (H)H(:)mm.	20	Entrada: `+05:30` Patrón: `%{ISO8601_TIMEZONE:tz}` Salida: `{"tz":"+05:30"}` Entrada: `-530` Patrón: `%{ISO8601_TIMEZONE:tz}` Salida: `{"tz":"-530"}` Entrada: `Z` Patrón: `%{ISO8601_TIMEZONE:tz}` Salida: `{"tz":"Z"}`
ISO8601_SEGUNDO	Coincide con un número que representa los segundos (0)0-60, seguido opcionalmente por un punto decimal o dos puntos y uno o más dígitos para fracciones de segundo	20	Entrada: `60` Patrón: `%{ISO8601_SECOND:second}` Salida: `{"second":"60"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
TIMESTAMP_ ISO8601	Coincide con el formato de ISO8601 fecha y hora (yy) yy- (M) M- (d) dT (H) H:mm :mm :((s) s) (Z| [+-] (H) H:mm) con segundos y zona horaria opcionales.	20	Entrada: `2023-05-15T14:30:00+05:30` Patrón: `%{TIMESTAMP_ISO8601:timestamp}` Salida: `{"timestamp":"2023-05-15T14:30:00+05:30"}` Entrada: `23-5-1T1:25+5:30` Patrón: `%{TIMESTAMP_ISO8601:timestamp}` Salida: `{"timestamp":"23-5-1T1:25+5:30"}` Entrada: `23-5-1T1:25Z` Patrón: `%{TIMESTAMP_ISO8601:timestamp}` Salida: `{"timestamp":"23-5-1T1:25Z"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
DATE	Coincide con una fecha en formato estadounidense usando %{DATE_US} o en formato UE usando %{DATE_EU}	20	Entrada: `11/29/2024` Patrón: `%{DATE:date}` Salida: `{"date":"11/29/2024"}` Entrada: `29.11.2024` Patrón: `%{DATE:date}` Salida: `{"date":"29.11.2024"}`
DATESTAMP	Coincide con el patrón %{DATE} seguido del patrón %{TIME}, separado por un espacio o un guion medio.	20	Entrada: `29-11-2024 14:30:00` Patrón: `%{DATESTAMP:dateTime}` Salida: `{"dateTime":"29-11-2024 14:30:00"}`
TZ	Coincide con las abreviaturas de zonas horarias habituales (PST, PDT, MST, MDT, CST CDT, EST, EDT, UTC).	20	Entrada: `PDT` Patrón: `%{TZ:tz}` Salida: `{"tz":"PDT"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
DATESTAMP_RFC822	Coincide con la fecha y la hora en el formato: Día MonthName (D) D (YY) YY (H) H:mm :(s) s Zona horaria	20	Entrada: `Monday Jan 5 23 1:30:00 CDT` Patrón: `%{DATESTAMP_RFC822 :dateTime}` Salida: `{"dateTime":"Monday Jan 5 23 1:30:00 CDT"}` Entrada: `Mon January 15 2023 14:30:00 PST` Patrón: `%{DATESTAMP_RFC822 :dateTime}` Salida: `{"dateTime":"Mon January 15 2023 14:30:00 PST"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
DATESTAMP_RFC2822	RFC2822 Coincide con el formato de fecha y hora: Día, (d) d MonthName (yy) yy (H) H:mm :(s) s Z|[+-] (H) H:mm	20	Entrada: `Mon, 15 May 2023 14:30:00 +0530` Patrón: `%{DATESTAMP_RFC2822:dateTime}` Salida: `{"dateTime":"Mon, 15 May 2023 14:30:00 +0530"}` Entrada: `Monday, 15 Jan 23 14:30:00 Z` Patrón: `%{DATESTAMP_RFC2822:dateTime}` Salida: `{"dateTime":"Monday, 15 Jan 23 14:30:00 Z"}`
DATESTAMP_OTHER	Coincide con la fecha y la hora en el formato: Día MonthName (d) d (H) H:mm :(s) s Zona horaria (yy) yy	20	Entrada: `Mon May 15 14:30:00 PST 2023` Patrón: `%{DATESTAMP_OTHER:dateTime}` Salida: `{"dateTime":"Mon May 15 14:30:00 PST 2023"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
DATESTAMP_EVENTLOG	Coincide con el formato compacto de fecha y hora sin separadores: (yy)yyMM(d)d(H)Hmm(s)s	20	Entrada: `20230515143000` Patrón: `%{DATESTAMP_EVENTLOG:dateTime}` Salida: `{"dateTime":"20230515143000"}`

Patrones grok de registro

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
LOGLEVEL	Coincide con los niveles de registros estándar en distintas mayúsculas y abreviaturas, incluidas las siguientes: `Alert/ALERT` , `Trace/TRACE` , `Debug/DEBUG` , `Notice/NOTICE` , `Info/INFO` , `Warn/Warning/WARN/WARNING` , `Err/Error/ERR/ERROR` , `Crit/Critical/CRIT/CRITICAL` , `Fatal/FATAL` , `Severe/SEVERE` , `Emerg/Emergency/EMERG/EMERGENCY`	20	Entrada: `INFO` Patrón: `%{LOGLEVEL:logLevel}` Salida: `{"logLevel":"INFO"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
HTTPDATE	Coincide con el formato de fecha y hora que se utiliza con frecuencia en los archivos de registro. Formato: (d) MonthName d/ (yy) yy :(H) H:mm : (s) s Zona horaria: coincide con los nombres de los meses en inglés completos o abreviado s (ejemplo MonthName: «enero» o «enero») Zona horaria: coincide con el patrón% {INT} grok	20	Entrada: `23/Nov/20 24:14:30:00 +0640` Patrón: `%{HTTPDAT E:date}` Salida: `{"date":"23/ Nov/2024:14:30:00 +0640"}`
SYSLOGTIM ESTAMP	Coincide con el formato de fecha con MonthName (d) d (H) H:mm :(s) s MonthName: Coincide con los nombres de los meses completos o abreviados en inglés (por ejemplo: «enero» o «enero»)	20	Entrada: `Nov 29 14:30:00` Patrón: `%{SYSLOGT IMESTAMP: dateTime}` Salida: `{"dateTim e":"Nov 29 14:30:00"}`
PROG	Coincide con el nombre de un programa compuesto por una cadena de letras, dígitos, puntos, guiones bajos, barras diagonales, signos porcentuales y guiones.	20	Entrada: `user.prof ile/settings-page` Patrón: `%{PROG:pr ogram}` Salida: `{"program ":"user.profile/ settings-page"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
SYSLOGPRO G	Coincide con el patrón grok PROG seguido opcionalmente de un identificador de proceso entre corchetes.	20	Entrada: `user.profile/settings-page[1234]` Patrón: `%{SYSLOGPROG:programWithId}` Salida: `{"programWithId":"user.profile/settings-page[1234]","program":"user.profile/settings-page","pid":"1234"}`
SYSLOGHOS T	Coincide con un patrón %{HOST} o %{IP}	5	Entrada: `2001:db8:3333:4444:5555:6666:7777:8888` Patrón: `%{SYSLOGHOST:ip}` Salida: `{"ip":"2001:db8:3333:4444:5555:6666:7777:8888"}`

Patrón grok	Description (Descripción)	Límite de patrones	Ejemplo
SYSLOGFAC ILITY	Coincide con la prioridad de syslog en formato decimal. El valor debe estar entre corchetes angulares (**<>**).	20	Entrada: `<13.6>` Patrón: `%{SYSLOGFACILITY:syslog}` Salida: `{"syslog":"<13.6>","facility":"13","priority":"6"}`

Patrones grok de registros comunes

Se pueden utilizar patrones grok personalizados y predefinidos para que coincidan con los formatos de registro de Apache, NGINX y Syslog Protocol (RFC 5424). Al usar estos patrones específicos, deben ser los primeros patrones de la configuración coincidente y ningún otro patrón puede precederlos. Además, solo puede seguirlos con exactamente un DATO. Patrón GREEDYDATA o GREEDYDATA_MULTILINE.

Patrón de Grok	Description (Descripción)	Límite de patrones
APACHE_ACCESS_LOG	Coincide con los registros de acceso de Apache	1
NGINX_ACCESS_LOG	Coincide con los registros de acceso de NGINX	1
SYSLOG5424	Coincide con los registros del protocolo	1

Patrón de Grok	Description (Descripción)	Límite de patrones
	Syslog (RFC 5424)	

A continuación se muestran ejemplos válidos y no válidos del uso de estos patrones de formato de registro comunes.

```
"%{NGINX_ACCESS_LOG} %{DATA}" // Valid
"%{SYSLOG5424}%{DATA:logMsg}" // Valid
"%{APACHE_ACCESS_LOG} %{GREEDYDATA:logMsg}" // Valid
"%{APACHE_ACCESS_LOG} %{SYSLOG5424}" // Invalid (multiple common log patterns used)
"%{NGINX_ACCESS_LOG} %{NUMBER:num}" // Invalid (Only GREEDYDATA and DATA patterns are
 supported with common log patterns)
"%{GREEDYDATA:logMsg} %{SYSLOG5424}" // Invalid (GREEDYDATA and DATA patterns are
 supported only after common log patterns)
```

Ejemplos de formatos de registro

Ejemplo de registro de Apache

Registros de ejemplo:

```
127.0.0.1 - - [03/Aug/2023:12:34:56 +0000] "GET /page.html HTTP/1.1" 200 1234
```

Transformador:

```
[
     {
        "grok": {
            "match": "%{APACHE_ACCESS_LOG}"
        }
    }
]
```

Salida:

```
{
    "request": "/page.html",
```

```
    "http_method": "GET",
    "status_code": 200,
    "http_version": "1.1",
    "response_size": 1234,
    "remote_host": "127.0.0.1",
    "timestamp": "2023-08-03T12:34:56Z"
}
```

Ejemplo de registro de NGINX

Registros de ejemplo:

```
192.168.1.100 - Foo [03/Aug/2023:12:34:56 +0000] "GET /account/login.html HTTP/1.1"
 200 42 "https://www.amazon.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36"
```

Transformador:

```
[
     {
        "grok": {
            "match": "%{NGINX_ACCESS_LOG}"
        }
    }
]
```

Salida:

```
{
    "request": "/account/login.html",
    "referrer": "https://www.amazon.com/",
    "agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/92.0.4515.131 Safari/537.36",
    "http_method": "GET",
    "status_code": 200,
    "auth_user": "Foo",
    "http_version": "1.1",
    "response_size": 42,
    "remote_host": "192.168.1.100",
    "timestamp": "2023-08-03T12:34:56Z"
}
```

Ejemplo de registro del protocolo Syslog (RFC 5424)

Registros de ejemplo:

```
<165>1 2003-10-11T22:14:15.003Z mymachine.example.com evntslog - ID47
 [exampleSDID@32473 iut="3" eventSource= "Application" eventID="1011"]
[examplePriority@32473 class="high"]
```

Transformador:

```
[
     {
        "grok": {
            "match": "%{SYSLOG5424}"
        }
    }
]
```

Salida:

```
{
  "pri": 165,
  "version": 1,
  "timestamp": "2003-10-11T22:14:15.003Z",
  "hostname": "mymachine.example.com",
  "app": "evntslog",
  "msg_id": "ID47",
  "structured_data": "exampleSDID@32473 iut=\"3\" eventSource= \"Application\" eventID=
\"1011\"",
  "message": "[examplePriority@32473 class=\"high\"]"
}
```

csv

El procesador de csv analiza los valores separados por comas (CSV) de los eventos del registro en columnas.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
origen	Ruta al campo del evento de registro que se analizará	No	`@message`	Longitud máxima: 128 Profundidad máxima de clave anidada: 3
delimiter	El carácter utilizado para separar cada columna del evento de registro de valores original separado por comas	No	`,`	Longitud máxima: 1, a menos que el valor sea `\t` o `\s`
quoteCharacter	Carácter utilizado como calificador de texto para una sola columna de datos	No	`"`	Longitud máxima: 1
columns	Lista de nombres que se utilizarán en las columnas del evento de registro transformado.	No	`[column_1, column_2`	Número máximo de columnas de CSV: 100 Longitud máxima: 128 Profundidad máxima de clave anidada: 3
destination	El campo principal en el que colocar los pares de valores clave transformados	No	`Root node`	Longitud máxima: 128 Profundidad máxima de clave anidada: 3

Si `delimiter` se establece en `\t`, se separará cada columna en un carácter de tabulación y `\t` separará cada columna en un solo carácter de espacio.

Ejemplo

Supongamos que parte de un evento de registro ingerido tiene este aspecto:

```
'Akua Mansa':28:'New York: USA'
```

Supongamos que utilizamos solo el procesador csv:

```
[
     {
        "csv": {
            "delimiter": ":",
            "quoteCharacter": "'"
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "column_1": "Akua Mansa",
  "column_2": "28",
  "column_3": "New York: USA"
}
```

Ejemplo 2

Supongamos que un evento de registro ingerido tiene un aspecto similar al siguiente:

```
{
    "timestamp": "2024-11-23T16:03:12Z",
    "type": "user_data",
    "logMsg": "'Akua Mansa':28:'New York: USA'"
}
```

Supongamos que analizamos el evento como JSON, ellos analizan un campo JSON con el procesador csv y especifican los nombres de las columnas y el destino:

```
[
    {
        "parseJSON": {}
    },
    {
        "csv": {
            "source": "logMsg",
            "delimiter": ":",
            "quoteCharacter": "'",
```

```
            "columns":["name","age","location"],
            "destination": "msg"
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
    "timestamp": "2024-11-23T16:03:12Z",
    "logMsg": "'Akua Mansa':28:'New York: USA'",
    "type": "user_data",
    "msg": {
        "name": "Akua Mansa",
        "age": "28",
        "location": "New York: USA"
    }
}
```

parseKeyValue

Usa el parseKeyValueprocesador para analizar un campo específico en pares clave-valor. Puede personalizar el procesador para analizar la información del campo con las siguientes opciones.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
origen	Ruta al campo del evento de registro que se analizará	No	`@message`	Longitud máxima: 128 Profundidad máxima de clave anidada: 3
destination	El campo de destino en el que se van a colocar los pares clave-valor extraídos	No		Longitud máxima: 128
fieldDelimiter	La cadena delimitadora de campo que se utiliza entre los pares clave-valor en los eventos de registro originales	No	&	Longitud máxima: 128

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
keyValueDelimiter	La cadena delimitadora que se utilizará entre la clave y el valor de cada par del evento de registro transformado	No	=	Longitud máxima: 128
nonMatchValue	Un valor para insertar en el campo de valores del resultado cuando un par clave-valor no se divide correctamente.	No		Longitud máxima: 128
keyPrefix	Si se desea añadir un prefijo a todas las claves transformadas, especifíquelo aquí.	No		Longitud máxima: 128
overwriteIfExists	Si se debe sobrescribir el valor si la clave de destino ya existe	No	`false`	

Ejemplo

Ejemplo de evento de registro de muestra:

```
key1:value1!key2:value2!key3:value3!key4
```

Supongamos que se utiliza la siguiente configuración de procesador:

```
[
    {
        "parseKeyValue": {
            "destination": "new_key",
            "fieldDelimiter": "!",
            "keyValueDelimiter": ":",
            "nonMatchValue": "defaultValue",
            "keyPrefix": "parsed_"
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "new_key": {
    "parsed_key1": "value1",
    "parsed_key2": "value2",
    "parsed_key3": "value3",
    "parsed_key4": "defaultValue"
  }
}
```

Procesadores integrados para registros vendidos AWS

Esta sección contiene información sobre los procesadores integrados que puede usar con AWS los servicios que venden registros.

Contenido

parseWAF

Utilice este procesador para analizar los registros AWS WAF vendidos. Toma el contenido de cada nombre de encabezado `httpRequest.headers` y crea claves JSON a partir de él, con el valor correspondiente. También hace lo mismo para `labels`. Estas transformaciones pueden facilitar mucho la consulta de AWS WAF los registros. Para obtener más información sobre el formato de AWS WAF registro, consulte Ejemplos de registro para el tráfico de ACL web.

Este procesador solo acepta `@message` como entrada.

> **Important**
>
> Si utiliza este procesador, debe ser el primer procesador del transformador.

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
  "timestamp": 1576280412771,
  "formatVersion": 1,
  "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
  "terminatingRuleId": "STMTest_SQLi_XSS",
  "terminatingRuleType": "REGULAR",
  "action": "BLOCK",
  "terminatingRuleMatchDetails": [
    {
      "conditionType": "SQL_INJECTION",
      "sensitivityLevel": "HIGH",
      "location": "HEADER",
      "matchedData": ["10", "AND", "1"]
    }
  ],
  "httpSourceName": "-",
  "httpSourceId": "-",
  "ruleGroupList": [],
  "rateBasedRuleList": [],
  "nonTerminatingMatchingRules": [],
  "httpRequest": {
    "clientIp": "1.1.1.1",
    "country": "AU",
    "headers": [
      { "name": "Host", "value": "localhost:1989" },
      { "name": "User-Agent", "value": "curl/7.61.1" },
      { "name": "Accept", "value": "*/*" },
      { "name": "x-stm-test", "value": "10 AND 1=1" }
    ],
    "uri": "/myUri",
    "args": "",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "rid"
  },
  "labels": [{ "name": "value" }]
}
```

La configuración del procesador es la siguiente:

```
[
    {
        "parseWAF": {}
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "httpRequest": {
    "headers": {
      "Host": "localhost:1989",
      "User-Agent": "curl/7.61.1",
      "Accept": "*/*",
      "x-stm-test": "10 AND 1=1"
    },
    "clientIp": "1.1.1.1",
    "country": "AU",
    "uri": "/myUri",
    "args": "",
    "httpVersion": "HTTP/1.1",
    "httpMethod": "GET",
    "requestId": "rid"
  },
  "labels": { "name": "value" },
  "timestamp": 1576280412771,
  "formatVersion": 1,
  "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
  "terminatingRuleId": "STMTest_SQLi_XSS",
  "terminatingRuleType": "REGULAR",
  "action": "BLOCK",
  "terminatingRuleMatchDetails": [
    {
      "conditionType": "SQL_INJECTION",
      "sensitivityLevel": "HIGH",
      "location": "HEADER",
      "matchedData": ["10", "AND", "1"]
    }
  ],
  "httpSourceName": "-",
```

```
  "httpSourceId": "-",
  "ruleGroupList": [],
  "rateBasedRuleList": [],
  "nonTerminatingMatchingRules": []
}
```

parsePostgres

Utilice este procesador para analizar Amazon RDS for PostgreSQL los registros vendidos, extraer campos y convertirlos al formato JSON. Para obtener más información sobre el formato de registro de RDS para PostgreSQL, consulte Archivos de registro de bases de datos de RDS para PostgreSQL.

Este procesador solo acepta `@message` como entrada.

> **Important**
>
> Si utiliza este procesador, debe ser el primer procesador del transformador.

Ejemplo

Ejemplo de evento de registro de muestra:

```
2019-03-10 03:54:59 UTC:10.0.0.123(52834):postgres@logtestdb:[20175]:ERROR: column
 "wrong_column_name" does not exist at character 8
```

La configuración del procesador es la siguiente:

```
[
    {
        "parsePostgres": {}
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "logTime": "2019-03-10 03:54:59 UTC",
  "srcIp": "10.0.0.123(52834)",
```

```
  "userName": "postgres",
  "dbName": "logtestdb",
  "processId": "20175",
  "logLevel": "ERROR"
}
```

parseCloudfront

Utilice este procesador para analizar los Amazon CloudFront registros vendidos, extraer campos y convertirlos al formato JSON. Los valores de los campos codificados se decodifican. Los valores enteros y dobles se tratan como tales. Para obtener más información sobre el formato de Amazon CloudFront registro, consulte Configurar y usar registros estándar (registros de acceso).

Este procesador solo acepta `@message` como entrada.

> **Important**
>
> Si utiliza este procesador, debe ser el primer procesador del transformador.

Ejemplo

Ejemplo de evento de registro de muestra:

```
2019-12-04  21:02:31   LAX1   392    192.0.2.24    GET
 d111111abcdef8.cloudfront.net  /index.html    200    -  Mozilla/5.0%20(Windows
%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36  -  -  Hit
    SOX4xwn4XV6Q4rgb7XiVGOHms_BGlTAC4KyHmureZmBNrjGdRLiNIQ==
 d111111abcdef8.cloudfront.net  https  23 0.001  -  TLSv1.2    ECDHE-RSA-AES128-GCM-
SHA256    Hit    HTTP/2.0   -  -  11040  0.001  Hit    text/html  78 -  -
```

La configuración del procesador es la siguiente:

```
[
    {
        "parseCloudfront": {}
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "date": "2019-12-04",
  "time": "21:02:31",
  "x-edge-location": "LAX1",
  "sc-bytes": 392,
  "c-ip": "192.0.2.24",
  "cs-method": "GET",
  "cs(Host)": "d111111abcdef8.cloudfront.net",
  "cs-uri-stem": "/index.html",
  "sc-status": 200,
  "cs(Referer)": "-",
  "cs(User-Agent)": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36",
  "cs-uri-query": "-",
  "cs(Cookie)": "-",
  "x-edge-result-type": "Hit",
  "x-edge-request-id": "SOX4xwn4XV6Q4rgb7XiVGOHms_BGlTAC4KyHmureZmBNrjGdRLiNIQ==",
  "x-host-header": "d111111abcdef8.cloudfront.net",
  "cs-protocol": "https",
  "cs-bytes": 23,
  "time-taken": 0.001,
  "x-forwarded-for": "-",
  "ssl-protocol": "TLSv1.2",
  "ssl-cipher": "ECDHE-RSA-AES128-GCM-SHA256",
  "x-edge-response-result-type": "Hit",
  "cs-protocol-version": "HTTP/2.0",
  "fle-status": "-",
  "fle-encrypted-fields": "-",
  "c-port": 11040,
  "time-to-first-byte": 0.001,
  "x-edge-detailed-result-type": "Hit",
  "sc-content-type": "text/html",
  "sc-content-len": 78,
  "sc-range-start": "-",
  "sc-range-end": "-"
}
```


parseRoute53

Utilice este procesador para analizar Amazon Route 53 Public Data Plane los registros vendidos, extraer campos y convertirlos al formato JSON. Los valores de los campos codificados se decodifican. Este procesador no admite Amazon Route 53 Resolver registros.

Este procesador solo acepta `@message` como entrada.

> **Important**
>
> Si utiliza este procesador, debe ser el primer procesador del transformador.

Ejemplo

Ejemplo de evento de registro de muestra:

```
1.0 2017-12-13T08:15:50.235Z Z123412341234 example.com AAAA NOERROR TCP IAD12 192.0.2.0
 198.51.100.0/24
```

La configuración del procesador es la siguiente:

```
[
    {
        "parseRoute53": {}
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "version": 1.0,
  "queryTimestamp": "2017-12-13T08:15:50.235Z",
  "hostZoneId": "Z123412341234",
  "queryName": "example.com",
  "queryType": "AAAA",
  "responseCode": "NOERROR",
  "protocol": "TCP",
  "edgeLocation": "IAD12",
  "resolverIp": "192.0.2.0",
  "ednsClientSubnet": "198.51.100.0/24"
}
```

parseVPC

Utilice este procesador para analizar los registros ofrecidos de Amazon VPC, extraer campos y convertirlos a formato JSON. Los valores de los campos codificados se decodifican.

Este procesador solo acepta `@message` como entrada.

> **Important**
>
> Si utiliza este procesador, debe ser el primer procesador del transformador.

Ejemplo

Ejemplo de evento de registro de muestra:

```
2 123456789010 eni-abc123de 192.0.2.0 192.0.2.24 20641 22 6 20 4249 1418530010
 1418530070 ACCEPT OK
```

La configuración del procesador es la siguiente:

```
[
    {
        "parseVPC": {}
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "version": 2,
  "accountId": "123456789010",
  "interfaceId": "eni-abc123de",
  "srcAddr": "192.0.2.0",
  "dstAddr": "192.0.2.24",
  "srcPort": 20641,
  "dstPort": 22,
  "protocol": 6,
  "packets": 20,
  "bytes": 4249,
  "start": 1418530010,
  "end": 1418530070,
  "action": "ACCEPT",
  "logStatus": "OK"
}
```

parseToOCSF

El procesador `parseToOCSF` convierte los registros en eventos Open Cybersecurity Schema Framework (OCSF). El OCSF es un estándar abierto que proporciona un esquema común para los datos de seguridad, lo que permite una mejor interoperabilidad y análisis entre diferentes herramientas y plataformas de seguridad.

Este procesador es especialmente útil para los flujos de trabajo de análisis de seguridad en los que es necesario estandarizar los formatos de registro de varios AWS servicios en un esquema coherente para el análisis posterior.

Parámetros

`eventSource` (obligatorio)

Especifica el AWS servicio o proceso que produce los eventos de registro que se van a convertir. Los valores válidos son:

- `CloudTrail`- CloudTrail registros
- `Route53Resolver`: registros de Route 53 Resolver
- `VPCFlow`: registros de flujo de Amazon VPC
- `EKSAudit`: registros de auditoría de Amazon EKS
- `AWSWAF`- AWS WAF registros

`ocsfVersion` (obligatorio)

Especifica qué versión del esquema OCSF se debe utilizar para los eventos de registro transformados. Versiones compatibles actualmente: `V1.1, V1.5`

`mappingVersion` (opcional)

Especifica la versión del mapeo de transformación de OCSF. Controla la lógica de transformación que se aplica al convertir los registros al formato OCSF. Si no se especifica, utiliza la última versión disponible en el momento de la creación de la política. Las políticas existentes no se actualizan automáticamente cuando se publican nuevas versiones de mapeo. Última versión actual:`v1.5.0`.

Nota: No es compatible cuando `ocsfVersion` es`V1.1`.

`source` (opcional)

La ruta al campo del evento de registro que desea analizar. Si se omite, se analiza todo el mensaje de registro.

Ejemplo

En el siguiente ejemplo se muestra cómo utilizar `parseToOCSF` para convertir los registros de flujo de VPC al formato OCSF:

```
{
  "parseToOCSF": {
    "eventSource": "VPCFlow",
    "ocsfVersion": "V1.1"
  }
}
```

El siguiente ejemplo muestra cómo especificar una versión de mapeo concreta para lograr un comportamiento de transformación coherente:

```
{
  "parseToOCSF": {
    "eventSource": "CloudTrail",
    "ocsfVersion": "V1.5",
    "mappingVersion": "v1.5.0"
  }
}
```

Procesadores de mutación de cadena

Esta sección contiene información sobre los procesadores de mutación de cadenas que puede utilizar con un transformador de eventos de registro.

Contenido

- trimString

lowerCaseString

El procesador `lowerCaseString` convierte una cadena a su versión en minúsculas.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
withKeys	Una lista de claves para convertir a minúsculas	Sí		Número máximo de entradas: 10

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "INNER_VALUE"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `lowerCaseString` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "lowerCaseString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
    "inner_key": "inner_value"
```

```
  }
}
```

upperCaseString

El procesador `upperCaseString` convierte una cadena a su versión en mayúsculas.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
withKeys	Una lista de claves para convertir a mayúsculas	Sí		Número máximo de entradas: 10

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `upperCaseString` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "upperCaseString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
```

```
    "inner_key": "INNER_VALUE"
  }
}
```

splitString

El procesador `splitString` es un tipo de procesador de mutaciones de cadenas que divide un campo en una matriz con un carácter delimitador.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
entries	Matriz de entradas. Cada elemento de la matriz debe contener los campos `source` y `delimiter` .	Sí		Número máximo de entradas: 10
origen	La clave del valor del campo que se va a dividir	Sí		Longitud máxima: 128
delimiter	La cadena delimitadora en la que se divide el valor del campo	Sí		Longitud máxima: 128

Ejemplo 1

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `splitString` con `parseJSON`:

```
[
     {
        "parseJSON": {}
    },
```

```
    {
         "splitString": {
            "entries": [
                {
                    "source": "outer_key.inner_key",
                    "delimiter": "_"
                }
            ]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
    "inner_key": [
      "inner",
      "value"
    ]
  }
}
```

Ejemplo 2

El delimitador en el que se divide la cadena puede tener varios caracteres.

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "item1, item2, item3"
    }
}
```

La configuración del transformador es la siguiente:

```
[
     {
        "parseJSON": {}
    },
    {
         "splitString": {
```

```
            "entries": [
                {
                    "source": "outer_key.inner_key",
                    "delimiter": ", "
                }
            ]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
    "inner_key": [
      "item1",
      "item2",
      "item3"
    ]
  }
}
```

substituteString

El procesador `substituteString` es un tipo de procesador de mutaciones de cadenas que compara el valor de una clave con el de una expresión regular y reemplaza todas las coincidencias por una cadena de reemplazo.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
entries	Matriz de entradas. Cada elemento de la matriz debe contener los campos `source`, `from` y `to`.	Sí		Número máximo de entradas: 10
origen	La clave del campo que se va a modificar	Sí		Longitud máxima: 128 Profundidad máxima de clave anidada: 3

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
desde	La cadena de expresión regular que se sustituirá. Los caracteres de expresiones regulares especiales, como [y], deben estar separados por \\ cuando se usan comillas dobles y con \ cuando se usan comillas simples o cuando se configuran desde la Consola de administración de AWS. Para obtener más información, consulte Class Pattern en el sitio web de Oracle. Se puede concluir un patrón en `(...)` para crear un grupo de captura numerado y crear grupos de captura con el nombre `(?P<group_name>...)` a los que se pueda hacer referencia en el campo `to`.	Sí		Longitud máxima: 128
a	Se puede utilizar la cadena que se sustituirá por cada coincidencia de las retroreferencias `from` con los grupos de captura. Use la forma $n para grupos numerados, como, por ejemplo, `$1`, y use `${group_name}` para grupos con nombre, como `${my_group} .>`	Sí		Longitud máxima: 128 Número de retrorefe rencias: 10 Número máximo de retroreferencias duplicadas: 2

Ejemplo 1

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key1": "[]",
        "inner_key2": "123-345-567",
        "inner_key3": "A cat takes a catnap."
    }
}
```

La configuración del transformador es la siguiente, y utiliza `substituteString` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "substituteString": {
            "entries": [
                {
                    "source": "outer_key.inner_key1",
                    "from": "\\[\\]",
                    "to": "value1"
                },
                {
                    "source": "outer_key.inner_key2",
                    "from": "[0-9]{3}-[0-9]{3}-[0-9]{3}",
                    "to": "xxx-xxx-xxx"
                },
                {
                    "source": "outer_key.inner_key3",
                    "from": "cat",
                    "to": "dog"
                }
            ]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
    "inner_key1": "value1",
```

```
    "inner_key2": "xxx-xxx-xxx",
    "inner_key3": "A dog takes a dognap."
  }
}
```

Ejemplo 2

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key1": "Tom, Dick, and Harry",
        "inner_key2": "arn:aws:sts::123456789012:assumed-role/MyImportantRole/
MySession"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `substituteString` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "substituteString": {
            "entries": [
                {
                    "source": "outer_key.inner_key1",
                    "from": "(\w+), (\w+), and (\w+)",
                    "to": "$1 and $3"
                },
                {
                    "source": "outer_key.inner_key2",
                    "from": "^arn:aws:sts::(?P<account_id>\\d{12}):assumed-role/(?
P<role_name>[\\w+=,.@-]+)/(?P<role_session_name>[\\w+=,.@-]+)$",
                    "to": "${account_id}:${role_name}:${role_session_name}"
                }
            ]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
    "inner_key1": "Tom and Harry",
    "inner_key2": "123456789012:MyImportantRole:MySession"
  }
}
```

trimString

El procesador `trimString` elimina los espacios en blanco desde el principio y el final de una clave.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
withKeys	Una lista de claves que se recortarán	Sí		Número máximo de entradas: 10

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "   inner_value  "
    }
}
```

La configuración del transformador es la siguiente, y utiliza `trimString` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "trimString": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
    "inner_key": "inner_value"
  }
}
```

Procesadores de mutación JSON

Esta sección contiene información sobre los procesadores de mutación JSON que puede usar con un transformador de eventos de registro.

Contenido

addKeys

Utilice el procesador `addKeys` para agregar nuevos pares clave-valor al evento de registro.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
entries	Matriz de entradas. Cada elemento de la matriz puede contener los campos `key`, `value` y `overwriteIfExists` .	Sí		Número máximo de entradas: 5
clave	La clave de la nueva entrada que se va a añadir	Sí		Longitud máxima: 128

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
				Profundidad máxima de clave anidada: 3
valor	El valor de la nueva entrada que se va a añadir	Sí		Longitud máxima: 256.
overwriteIfExists	Si lo establece en `true`, el valor existente se sobrescribe si `key` ya existe en el evento. El valor predeterminado es `false`.	No	false	Sin límite

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `addKeys` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "addKeys": {
            "entries": [
                {
                    "key": "outer_key.new_key",
                    "value": "new_value"
                }
            ]
        }
    }
```

```
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
    "inner_key": "inner_value",
    "new_key": "new_value"
  }
}
```

deleteKeys

Utilice el procesador `deleteKeys` para eliminar los campos de un evento de registro. Estos campos pueden incluir pares clave-valor.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
withKeys	La lista de claves que se eliminarán.	Sí	Sin límite	Número máximo de entradas: 5

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `deleteKeys` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
```

```
        "deleteKeys": {
            "withKeys":["outer_key.inner_key"]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

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

moveKeys

Utilice el procesador `moveKeys` para mover una clave de un campo a otro.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
entries	Matriz de entradas. Cada elemento de la matriz puede contener los campos `source`, `target` y `overwriteIfExists` .	Sí		Número máximo de entradas: 5
origen	La clave para mover	Sí		Longitud máxima: 128 Profundidad máxima de clave anidada: 3
destino	La clave de destino	Sí		Longitud máxima: 128 Profundidad máxima de clave anidada: 3
overwriteIfExists	Si lo establece en `true`, el valor existente se sobrescribe si `key` ya existe en el evento. El valor predeterminado es `false`.	No	false	Sin límite

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key1": {
        "inner_key1": "inner_value1"
    },
    "outer_key2": {
        "inner_key2": "inner_value2"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `moveKeys` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "moveKeys": {
            "entries": [
                {
                    "source": "outer_key1.inner_key1",
                    "target": "outer_key2"
                }
            ]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key1": {},
  "outer_key2": {
    "inner_key2": "inner_value2",
    "inner_key1": "inner_value1"
  }
}
```

renameKeys

Utilice el procesador `renameKeys` para cambiar el nombre de las claves de un evento de registro.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
entries	Matriz de entradas. Cada elemento de la matriz puede contener los campos `key`, `target` y `overwriteIfExists` .	Sí	Sin límite	Número máximo de entradas: 5
clave	La clave cuyo nombre debe cambiarse	Sí	Sin límite	Longitud máxima: 128
destino	El nombre de la nueva clave	Sí	Sin límite	Longitud máxima: 128 Profundidad máxima de clave anidada: 3
overwriteIfExists	Si lo establece en `true`, el valor existente se sobrescribe si `key` ya existe en el evento. El valor predeterminado es `false`.	No	false	Sin límite

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `renameKeys` con `parseJSON`:

```
[
```

```
    {
        "parseJSON": {}
    },
    {
        "renameKeys": {
            "entries": [
                {
                    "key": "outer_key",
                    "target": "new_key"
                }
            ]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "new_key": {
    "inner_key": "inner_value"
  }
}
```

copyValue

Utilice el procesador `copyValue` para copiar los valores de un evento de registro. También se puede utilizar este procesador para añadir metadatos a los eventos de registro, mediante la copia de los valores de las siguientes claves de metadatos en los eventos de registro: `@logGroupName`, `@logGroupStream`, `@accountId`, `@regionName`. Esto se ilustra con el siguiente ejemplo.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
entries	Matriz de entradas. Cada elemento de la matriz puede contener los campos `source`, `target` y `overwriteIfExists` .	Sí		Número máximo de entradas: 5
origen	La clave para copiar	Sí		Longitud máxima: 128

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
				Profundidad máxima de clave anidada: 3
destino	La clave a la que se debe copiar el valor	Sí	Sin límite	Longitud máxima: 128 Profundidad máxima de clave anidada: 3
overwriteIfExists	Si lo establece en `true`, el valor existente se sobrescribe si `key` ya existe en el evento. El valor predeterminado es `false`.	No	false	Sin límite

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": {
        "inner_key": "inner_value"
    }
}
```

La configuración del transformador es la siguiente, y utiliza `copyValue` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "copyValue": {
            "entries": [
                {
                    "key": "outer_key.new_key",
                    "target": "new_key"
                },
                {
                    "source": "@logGroupName",
```

```
                    "target": "log_group_name"
                },
                {
                    "source": "@logGroupStream",
                    "target": "log_group_stream"
                },
                {
                    "source": "@accountId",
                    "target": "account_id"
                },
                {
                    "source": "@regionName",
                    "target": "region_name"
                }
            ]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
  "outer_key": {
    "inner_key": "inner_value"
  },
  "new_key": "inner_value",
  "log_group_name": "myLogGroupName",
  "log_group_stream": "myLogStreamName",
  "account_id": "012345678912",
  "region_name": "us-east-1"
}
```

listToMap

El procesador `listToMap` toma una lista de objetos que contienen campos clave y los convierte en un mapa de claves de destino.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
origen	La clave es ProcessingEvent la lista de objetos que se convertirán en un mapa	Sí		Longitud máxima: 128 Profundidad máxima de clave anidada: 3
clave	La clave de los campos que se van a extraer como claves en el mapa generado	Sí		Longitud máxima: 128
valueKey	Si se especifica, los valores que se especifiquen en este parámetro se extraerán de los objetos `source` y se colocarán en los valores del mapa generado. De lo contrario, los objetos originales de la lista de origen se incluirán en los valores del mapa generado.	No		Longitud máxima: 128
destino	La clave del campo que contendrá el mapa generado	No	Nodo Raíz	Longitud máxima: 128 Profundidad máxima de clave anidada: 3
flatten	Un valor booleano para indicar si la lista se aplanará en elementos individuales o si los valores del mapa generado serán listas. De forma predeterminada, los valores de las claves coincidentes se representarán en una matriz. Configure `flatten` en `true` para convertir la matriz en un valor único en función del valor de `flattenedElement` .	No	false	

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
flattenedElement	Si se establece `true` en `flatten`, utilice `flattenedElement` para especificar qué elemento, `first` o bien `last`, se desea conservar.	Necesario cuando `flatten` se establece en `true`.		El valor solo puede ser `first` o `last`.

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ]
}
```

Transformador para el caso de uso 1: `flatten` es `false`

```
[
    {
        "parseJSON": {}
```

```
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": false
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": [
        "val-a"
    ],
    "b": [
        "val-b1",
        "val-b2"
    ],
    "c": [
        "val-c"
    ]
}
```

Transformador para el caso de uso 2: `flatten` es `true` y `flattenedElement` es `first`

```
[
    {
        "parseJSON": {}
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": true,
            "flattenedElement": "first"
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": "val-a",
    "b": "val-b1",
    "c": "val-c"
}
```

Transformador para el caso de uso 3: `flatten` es `true` y `flattenedElement` es `last`

```
[
    {
        "parseJSON": {}
    },
    {
        "listToMap": {
            "source": "outer_key"
            "key": "inner_key",
            "valueKey": "inner_value",
            "flatten": true,
            "flattenedElement": "last"
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
    "outer_key": [
        {
            "inner_key": "a",
            "inner_value": "val-a"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b1"
        },
        {
            "inner_key": "b",
            "inner_value": "val-b2"
        },
        {
            "inner_key": "c",
            "inner_value": "val-c"
        }
    ],
    "a": "val-a",
    "b": "val-b2",
    "c": "val-c"
}
```

Procesadores convertidores de tipos de datos

Esta sección contiene información sobre los procesadores convertidores de tipos de datos que puede utilizar con un transformador de eventos de registro.

Contenido

typeConverter

Utilice el procesador `typeConverter` para convertir un tipo de valor asociado a la clave especificada en el tipo especificado. Es un procesador de conversión que cambia los tipos de los campos especificados. Los valores se pueden convertir en uno de los siguientes tipos de datos: `integer`, `double`, `string` y `boolean`.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
entries	Matriz de entradas. Cada elemento de la matriz debe contener los campos `key` y `type`.	Sí		Número máximo de entradas: 10
clave	La clave con el valor que se va a convertir a un tipo diferente	Sí		Longitud máxima: 128 Profundidad máxima de clave anidada: 3
type	El tipo al que se va a convertir. Los valores válidos son `integer`, `double`, `string` y `boolean`.	Sí		

Ejemplo

Ejemplo de evento de registro de muestra:

```
{
```

```
    "name": "value",
    "status": "200"
}
```

La configuración del transformador es la siguiente, y utiliza `typeConverter` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "typeConverter": {
            "entries": [
                {
                    "key": "status",
                    "type": "integer"
                }
            ]
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
    "name": "value",
    "status": 200
}
```

datetimeConverter

Utilice el procesador `datetimeConverter` para convertir una cadena de fecha y hora en el formato que especifique.

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
origen	La clave donde aplicar la conversión de fechas.	Sí		Número máximo de entradas: 10

Campo	Description (Descripción)	¿Obligatorio?	Predeterminado	Límites
MatchPattern	Una lista de patrones que se pueden comparar con los del campo `source`	Sí		Número máximo de entradas: 5
destino	El campo JSON en el que se almacenará el resultado.	Sí		Longitud máxima: 128 Profundidad máxima de clave anidada: 3
targetFormat	El formato de fecha y hora que se utilizará para los datos convertidos en el campo de destino.	No	`yyyy-MM-dd'T'HH:mm:ss.SSS'Z`	Longitud máxima: 64
Zona horaria de origen	La zona horaria del campo de origen. Para obtener una lista de los valores posibles, consulte Java Supported Zone Ids and Offsets.	No	UTC	Longitud mínima: 1
targetTimezone	La zona horaria del campo de destino. Para obtener una lista de los valores posibles, consulte Java Supported Zone Ids and Offsets.	No	UTC	Longitud mínima: 1
locale	La región del campo de origen. Para obtener una lista de valores posibles, consulte el método Locale getAvailableLocales () en Java con ejemplos.	Sí		Longitud mínima: 1

Ejemplo

Ejemplo de evento de registro de muestra:

```
{"german_datetime": "Samstag 05. Dezember 1998 11:00:00"}
```

La configuración del transformador es la siguiente, y utiliza `dateTimeConverter` con `parseJSON`:

```
[
    {
        "parseJSON": {}
    },
    {
        "dateTimeConverter": {
            "source": "german_datetime",
            "target": "target_1",
            "locale": "de",
            "matchPatterns": ["EEEE dd. MMMM yyyy HH:mm:ss"],
            "sourceTimezone": "Europe/Berlin",
            "targetTimezone": "America/New_York",
            "targetFormat": "yyyy-MM-dd'T'HH:mm:ss z"
        }
    }
]
```

El evento de registro transformado sería el siguiente.

```
{
    "german_datetime": "Samstag 05. Dezember 1998 11:00:00",
    "target_1": "1998-12-05T17:00:00 MEZ"
}
```

Métricas y errores de transformación

CloudWatch Logs publica las métricas de transformación en CloudWatch. Las métricas incluyen `TransformedLogEvents`, `TransformedBytes` y `TransformationErrors`. Para obtener más información, consulte Métricas y dimensiones del transformador de registro.

Cuando CloudWatch Logs intenta transformar un evento de registro y no lo logra, agrega un campo `@transformationError` del sistema a ese evento de registro. Cuando ejecute una consulta de

CloudWatch Logs Insights, verá este campo en todos los eventos de registro en los que se haya producido un error de transformación. Se puede consultar este campo con una consulta como, por ejemplo, `filter ispresent(@transformationError)` para buscar todos los eventos de transformación fallidos.

Analice con Amazon OpenSearch Service

CloudWatch Logs se integra Amazon OpenSearch Service para permitirle crear paneles de control automáticos seleccionados que muestran las métricas clave que OpenSearch Service obtiene a partir de los registros vendidos desde los servicios. AWS Están disponibles los siguientes paneles:

- Un panel de registros de flujos de Amazon VPC captura los datos de flujo de red para Amazon VPC. Ayuda a analizar el tráfico de la red, detectar patrones inusuales y supervisar el uso de los recursos. Entre las métricas clave que se muestran se incluyen las siguientes:
  - Flujos totales y aceptación y rechazo de estos flujos
  - Patrones de tráfico a lo largo del tiempo
  - Un diagrama de Sankey que ilustra el flujo de datos entre el origen y el destino IPs (los más consultados)
  - Arriba IPs por bytes y paquetes transferidos

  > **Note**
  >
  > Actualmente, solo se admite el formato de campos de la versión 2 de VPC.

- Un panel de AWS WAF registros proporciona información sobre el tráfico web que monitorea AWS WAF. Este panel le ayuda a identificar los patrones de tráfico, las solicitudes bloqueadas y las posibles amenazas procedentes de regiones o regiones específicas IPs. Entre las métricas clave que se muestran se incluyen las siguientes:
  - Total de solicitudes, incluidas las denominadas “ALLOW” y “BLOCK”.
  - Historial de solicitudes a lo largo del tiempo, que muestra las solicitudes permitidas y bloqueadas.
  - Desglose de las solicitudes por nombre de ACL web, solicitudes bloqueadas por regla de finalización y fuente. IPs
  - Una distribución geográfica de los orígenes de las solicitudes.
  - Principales reglas de cliente IPs y de terminación por número de solicitudes.
- Un panel CloudTrail de registros proporciona una descripción general de la actividad de las API en su AWS entorno mediante el uso de CloudTrail registros. Resulta útil para supervisar la actividad de las API, auditar las acciones e identificar posibles problemas de seguridad o conformidad. Entre las métricas clave que se muestran se incluyen las siguientes:

  - Recuento total de eventos e historial de eventos a lo largo del tiempo
  - Un desglose de los eventos por cuenta IDs, categorías y regiones.
  - Principales APIs, servicios y fuentes que IPs intervienen en la generación de eventos.
  - Una tabla de los principales usuarios que generan eventos, en la que se detalla la información de las cuentas de usuario y el recuento de eventos.
- Un panel de control de AWS Network Firewall proporciona una mayor visibilidad del tráfico de la red y ofrece información valiosa para la supervisión y el análisis de la seguridad. Este panel ofrece una visión completa de varias métricas y patrones de la red, para identificar rápido los posibles problemas de seguridad y optimizar las configuraciones de red. Entre las métricas clave que se muestran se incluyen las siguientes:
  - Principales hablantes y protocolos
  - Información sobre los puntos PrivateLink finales
  - Tráfico de indicación de nombre de servidor TLS permitido y bloqueado

Las métricas que se muestran en estos paneles seleccionados se derivan de los análisis de Amazon OpenSearch Service .

Antes de poder ver estos paneles, debe crear un rol de IAM y realizar una integración única de CloudWatch Logs con él. Amazon OpenSearch Service Esta integración única configura los Amazon OpenSearch Service recursos necesarios para crear y renderizar el panel. Se le cobrarán cargos por los OpenSearch servicios utilizados. Para obtener más información, consulta los CloudWatch precios de Amazon.

Puede crear estos paneles seleccionados solo para los grupos de registro en la clase Estándar.

> **Important**
>
> No utilice transformadores de registro para ningún grupo de registros para el que desee crear paneles de registros ofrecidos. La transformación de los eventos de registro provocará que los paneles tengan datos vacíos.

Temas

- Paso 1: Crear la integración con OpenSearch Service
- Paso 2: Creación de paneles de control de registros de venta

- Visualice, edite o elimine los paneles de registros ofrecidos
- Políticas de IAM para usuarios
- Permisos que necesita la integración

Paso 1: Crear la integración con OpenSearch Service

El primer paso es crear la integración con el OpenSearch Servicio, lo que solo debe hacer una vez. La creación de la integración dará lugar a los siguientes recursos en su cuenta.

- Una colección de series OpenSearch Service temporales sin alta disponibilidad.

  Una colección es un conjunto de índices de OpenSearch servicios que funcionan juntos para soportar una carga de trabajo.
- Dos políticas de seguridad para la colección. Uno define el tipo de cifrado, que puede ser con una AWS KMS clave administrada por el cliente o con una clave propiedad del servicio. La otra política define el acceso a la red, lo que permite a la aplicación del OpenSearch servicio acceder a la colección. Para obtener más información, consulta Cifrado de datos en reposo para Amazon OpenSearch Service.
- Una política de acceso a los datos del OpenSearch servicio que define quién puede acceder a los datos de la recopilación.
- Una fuente de datos de consulta directa del OpenSearch servicio con CloudWatch los registros definidos como fuente.
- Una aplicación OpenSearch de servicio con el nombre`aws-analytics`. La aplicación se configurará para permitir la creación de un espacio de trabajo. Si ya existe una aplicación denominada `aws-analytics`, se actualizará para añadir esta colección como origen de datos.
- Un espacio OpenSearch de trabajo de servicio que alojará los paneles y permitirá a todas las personas a las que se haya concedido el acceso leer desde el espacio de trabajo.

Temas

- Permisos necesarios
- Crear la integración

Permisos necesarios

Para crear la integración, debe iniciar sesión en una cuenta que tenga la política de IAM CloudWatchOpenSearchDashboardsFullAccessgestionada o permisos equivalentes, como se muestra aquí. También debe tener estos permisos para eliminar la integración, crear, editar y eliminar paneles, y para actualizar el panel de forma manual.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Sid": "CloudWatchOpenSearchDashboardsIntegration",
            "Effect": "Allow",
            "Action": [
                "logs:ListIntegrations",
                "logs:GetIntegration",
                "logs:DeleteIntegration",
                "logs:PutIntegration",
                "logs:DescribeLogGroups",
                "opensearch:ApplicationAccessAll",
                "iam:ListRoles",
                "iam:ListUsers"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLogsOpensearchReadAPIs",
            "Effect": "Allow",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:BatchGetLifecyclePolicy",
                "es:ListApplications"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
```

```
            "Sid": "CloudWatchLogsOpensearchCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
opensearchservice.amazonaws.com/AWSServiceRoleForAmazonOpenSearchService",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "opensearchservice.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsObservabilityCreateServiceLinkedAccess",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
observability.aoss.amazonaws.com/AWSServiceRoleForAmazonOpenSearchServerless",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "observability.aoss.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionRequestAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateCollection"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
```

```
                "aws:TagKeys": "CloudWatchOpenSearchIntegration"
            }
        }
    },
    {
        "Sid": "CloudWatchLogsApplicationRequestAccess",
        "Effect": "Allow",
        "Action": [
            "es:CreateApplication"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:CalledViaFirst": "logs.amazonaws.com",
                "aws:RequestTag/OpenSearchIntegration": [
                    "Dashboards"
                ]
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": "OpenSearchIntegration"
            }
        }
    },
    {
        "Sid": "CloudWatchLogsCollectionResourceAccess",
        "Effect": "Allow",
        "Action": [
            "aoss:DeleteCollection"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:CalledViaFirst": "logs.amazonaws.com",
                "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                    "Dashboards"
                ]
            }
        }
    },
    {
        "Sid": "CloudWatchLogsApplicationResourceAccess",
        "Effect": "Allow",
        "Action": [
            "es:UpdateApplication",
```

```
                "es:GetApplication"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/OpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsCollectionPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateSecurityPolicy",
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:DeleteSecurityPolicy",
                "aoss:GetAccessPolicy",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAPIAccessAll",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "cloudwatch-logs-*"
                }
            }
        },
```

```
        {
            "Sid": "CloudWatchLogsIndexPolicyAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy",
                "aoss:DeleteAccessPolicy",
                "aoss:GetAccessPolicy",
                "aoss:CreateLifecyclePolicy",
                "aoss:DeleteLifecyclePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:index": "cloudwatch-logs-*",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsDQSRequestQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:AddDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:RequestTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "CloudWatchOpenSearchIntegration"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsStartDirectQueryAccess",
            "Effect": "Allow",
            "Action": [
                "opensearch:StartDirectQuery",
                "opensearch:GetDirectQuery"
            ],
```

```
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*"
        },
        {
            "Sid": "CloudWatchLogsDQSResourceQueryAccess",
            "Effect": "Allow",
            "Action": [
                "es:GetDirectQueryDataSource",
                "es:DeleteDirectQueryDataSource"
            ],
            "Resource": "arn:aws:opensearch:*:*:datasource/cloudwatch_logs_*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
                    "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                        "Dashboards"
                    ]
                }
            }
        },
        {
            "Sid": "CloudWatchLogsPassRoleAccess",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService":
"directquery.opensearchservice.amazonaws.com",
                    "aws:CalledViaFirst": "logs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudWatchLogsAossTagsAccess",
            "Effect": "Allow",
            "Action": [
                "aoss:TagResource"
            ],
            "Resource": "arn:aws:aoss:*:*:collection/*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "logs.amazonaws.com",
```

```
                "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                    "Dashboards"
                ]
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": "CloudWatchOpenSearchIntegration"
            }
        }
    },
    {
        "Sid": "CloudWatchLogsEsApplicationTagsAccess",
        "Effect": "Allow",
        "Action": [
            "es:AddTags"
        ],
        "Resource": "arn:aws:opensearch:*:*:application/*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/OpenSearchIntegration": [
                    "Dashboards"
                ],
                "aws:CalledViaFirst": "logs.amazonaws.com"
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": "OpenSearchIntegration"
            }
        }
    },
    {
        "Sid": "CloudWatchLogsEsDataSourceTagsAccess",
        "Effect": "Allow",
        "Action": [
            "es:AddTags"
        ],
        "Resource": "arn:aws:opensearch:*:*:datasource/*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/CloudWatchOpenSearchIntegration": [
                    "Dashboards"
                ],
                "aws:CalledViaFirst": "logs.amazonaws.com"
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": "CloudWatchOpenSearchIntegration"
```

```
                }
            }
        }
    ]
}
```

Crear la integración

Siga estos pasos para crear la integración.

Para integrar CloudWatch Logs con Amazon OpenSearch Service

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, elija Logs Insights y, a continuación, elija la OpenSearch pestaña Analizar con.
3. Seleccione Crear integración.
4. En Nombre de la integración, introduzca un nombre para la integración.
5. (Opcional) Para cifrar los datos escritos en OpenSearch Service Serverless, introduzca el ARN de la AWS KMS clave que quiere usar en el ARN de la clave de KMS. Para obtener más información, consulta Encryption at rest en la Guía para desarrolladores de Amazon OpenSearch Service.
6. Para la retención de datos, introduzca el tiempo que desea que se conserven los índices de datos del OpenSearch Servicio. Esto también define el periodo máximo durante el cual se pueden ver los datos en los paneles. Si se elige un período de retención de datos más prolongado, se incurrirá en costos adicionales de búsqueda e indexación. Para obtener más información, consulte los precios OpenSearch de Service Serverless.

   El período máximo de retención es de 30 días.

   La duración de la retención de los datos también se utilizará para crear la política del ciclo de vida de la recopilación de OpenSearch servicios.
7. En el caso de la función de IAM para escribir en la OpenSearch colección, cree una nueva función de IAM o seleccione una función de IAM existente para utilizarla para escribir en la OpenSearch colección de servicios.

   Crear un nuevo rol es el método más sencillo y el rol se creará con los permisos necesarios.

> **Note**
>
> Si se crea un rol, se tendrán permisos para leer todos los grupos de registros de la cuenta.

Si se desea seleccionar un rol existente, se deben tener los permisos que aparecen en Permisos que necesita la integración. Como alternativa, se puede elegir Usar un rol existente y, a continuación, en la sección Verificar los permisos de acceso del rol seleccionado, se puede elegir Crear rol. De esta forma, se pueden utilizar los permisos que figuran en Permisos que necesita la integración como una plantilla y modificarlos. Por ejemplo, si se desea especificar un control más preciso de los grupos de registros.

8. En el caso de los roles y los usuarios de IAM que pueden ver los paneles, seleccione cómo quiere conceder el acceso a los roles de IAM y a los usuarios de IAM el acceso al panel de registros ofrecidos:
   - Para limitar el acceso al panel de control solo a algunos usuarios, seleccione Seleccionar los roles de IAM y los usuarios que puedan ver los paneles y, a continuación, en el cuadro de texto, busque y seleccione los roles de IAM y los usuarios de IAM a los que quiere conceder acceso.
   - Para conceder acceso al panel a todos los usuarios, seleccione Permitir que todos los roles y usuarios de esta cuenta vean los paneles.

   > **Important**
   >
   > Al seleccionar roles o usuarios, o elegir a todos los usuarios, solo se añaden a la política de acceso a los datos necesaria para acceder a la colección de OpenSearch servicios que almacena los datos del panel de control. Para que puedan ver los paneles de control de los registros ofrecidos, también se debe conceder a esos roles y usuarios la política de IAM CloudWatchOpenSearchDashboardAccess administrada.

9. Seleccione Creación de integración.

   La creación de la integración tardará unos minutos.

Paso 2: Creación de paneles de control de registros de venta

Una vez creada la integración, se pueden crear paneles de control. Los paneles están disponibles para los logs CloudTrail , logs y logs de flujo de Amazon VPC. AWS WAF

Para crear un panel de registro vendido con las métricas derivadas por el servicio OpenSearch

1. Abra la CloudWatch consola en. https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación izquierdo, elija Logs Insights y, a continuación, elija la OpenSearch pestaña Analizar con.
3. Elija Crear panel.
4. Elija el tipo de registros para los que desea crear el panel de control AWS WAF, los registros de flujo de Amazon VPC o. CloudTrail AWS Network Firewall
5. Introduzca un nombre para el panel y, opcionalmente, una descripción.
6. En cuanto a la frecuencia de sincronización de datos, introduzca la frecuencia con la que desea que OpenSearch Service consulte CloudWatch para que las métricas y los índices creados en el OpenSearch Servicio se puedan sincronizar y actualizar con los nuevos datos. OpenSearch El servicio crea métricas e índices en sus registros para representar el panel.

   Elegir un tiempo más corto mantiene los datos más actualizados e implica costos más altos.
7. Seleccione los grupos de registros de los que desea recopilar datos para este panel. Asegúrese de seleccionar grupos de registros que coincidan con el tipo de panel que va a crear.

   Puede utilizar el botón Examinar grupos de registros y la opción Ver muestras de registro de los grupos de registros seleccionados al seleccionar estas opciones, para asegurarse de que obtiene los grupos de registros que desea.
8. Elija Crear panel.

   Al principio, el panel aparece sin datos. Transcurridos unos minutos, los datos aparecerán en el panel. Cuando los datos aparezcan por primera vez, corresponderán a los últimos 15 minutos de entradas de registro.

Visualice, edite o elimine los paneles de registros ofrecidos

Vea los paneles de registros vendidos en CloudWatch Logs o Service OpenSearch

Para poder ver los paneles, debe iniciar sesión con un director de IAM que tenga la política de IAM. CloudWatchOpenSearchDashboardAccess

Visualización de los paneles de registro ofrecidos

1. Abra la consola en CloudWatch . https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación izquierdo, elija Logs Insights y, a continuación, elija la OpenSearch pestaña Analizar con.
3. Seleccione el panel de control en el cuadro de OpenSearch cuadros de mando.
4. (Opcional) En la esquina superior derecha, selecciona Ver en OpenSearch.

   Se abre la consola de OpenSearch servicio y allí aparece el mismo panel de control. En la consola de OpenSearch servicio, puede realizar cambios en el panel y sus widgets, y estos cambios también estarán visibles cuando visualice el panel en CloudWatch los registros.

Concesión de acceso a otros roles de IAM o usuarios de IAM para ver el panel

Para conceder acceso a otras entidades principales de IAM una vez creada la integración, siga estos pasos.

Concesión de acceso a roles de IAM o usuarios de IAM adicionales al panel de registros ofrecidos

1. Edite la política de acceso a los datos de la colección para añadir estos roles o usuarios. Para obtener más información, consulte Control de acceso a datos para Amazon OpenSearch Service Serverless en la Guía para desarrolladores OpenSearch de servicios.
2. CloudWatchOpenSearchDashboardAccessConcédelo a estos usuarios.
   Para obtener más información acerca del contenido de esta política, consulte CloudWatchOpenSearchDashboardAccess.

Edición de la configuración del panel

Se pueden editar el nombre, la descripción y la frecuencia de sincronización de los paneles de registro cifrados existentes.

Edición de un panel de control de registros ofrecido

1. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En el panel de navegación izquierdo, elija Logs Insights y, a continuación, elija la OpenSearch pestaña Analizar con.
3. Seleccione el panel de control en el cuadro de OpenSearch cuadros de mando.
4. Seleccione Acciones y Cambiar los detalles del panel.
5. Realice los cambios y, a continuación, elija Confirmar cambios.

Eliminación de un panel de registro ofrecido

Puede eliminar un panel de registro ofrecido. Si lo hace, se eliminarán todos el panel, las métricas y los índices creados en la colección de OpenSearch servicios.

> **Note**
>
> Después de eliminar un panel de registro ofrecido, espere al menos seis horas antes de intentar volver a crear ese mismo panel. Si no se espera, el panel que se ha vuelto a crear no funcionará correctamente.

Eliminación de un panel de registro ofrecido

1. Abre la CloudWatch consola en. https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación izquierdo, elija Logs Insights y, a continuación, elija la OpenSearch pestaña Analizar con.
3. Seleccione el panel de control en el cuadro de OpenSearch cuadros de mando.
4. Elija Acciones, Eliminar.
5. Confirme su decisión mediante el ingreso de **`delete`**, y, a continuación, seleccione Eliminar.

Elimine toda la integración del panel de registro vendido con el servicio OpenSearch

Puede eliminar toda la OpenSearch integración. Si lo hace, se eliminarán todos los paneles de registros ofrecidos y los datos que se mostraban en ellos.

Important

Para evitar costes continuos, recomendamos encarecidamente la eliminación manual de los siguientes recursos antes de eliminar la integración. Al eliminar la integración, estos recursos no se eliminan automáticamente y, una vez eliminada la integración, no se los podrá acceder para eliminarlos. Para buscar los nombres de los recursos que se van a eliminar, consulte el siguiente procedimiento.

- La fuente de datos
- La colección
- La política de acceso a los datos
- La política de cifrado
- La política de red
- La política del ciclo de vida

Para eliminar toda la integración del panel de control de Vended Log con Service OpenSearch

1. Abre la CloudWatch consola en. https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación izquierdo, elija Configuración.
3. Elija la pestaña Logs (Registros).
4. En la sección de OpenSearch integración, selecciona Eliminar integración.

   La siguiente pantalla muestra los nombres de los recursos del OpenSearch servicio que debe eliminar antes de eliminar la integración.
5. Confirme su decisión mediante el ingreso de **`delete`**, y, a continuación, seleccione Eliminación de la integración.

Políticas de IAM para usuarios

CloudWatch Logs ha creado dos políticas de IAM CloudWatchOpenSearchDashboardsFullAccessy CloudWatchOpenSearchDashboardAccess. En la siguiente tabla se enumeran las acciones que permite cada una de estas políticas.

Action	Política de IAM	Se necesitan permisos adicionales
Creación de integración	CloudWatchOpenSearchDashboardsFullAccess	
Eliminación de integración	CloudWatchOpenSearchDashboardsFullAccess	
Creación de panel	CloudWatchOpenSearchDashboardsFullAccess	
Edición de panel	CloudWatchOpenSearchDashboardsFullAccess	
Eliminación de panel	CloudWatchOpenSearchDashboardsFullAccess	
Actualización del panel de control con Sincronizar ahora	CloudWatchOpenSearchDashboardsFullAccess	
Visualización de la integración en Configuración	CloudWatchOpenSearchDashboardAccess o CloudWatchOpenSearchDashboardsFullAccess	
Visualización del panel	CloudWatchOpenSearchDashboardAccess o CloudWatchOpenSearchDashboardsFullAccess	Especifique el rol o el usuario al crear la integración, o edite la política de acceso a los datos de la colección para agregar estos roles o usuarios. Para obtener más información, consulte Control

Action	Política de IAM	Se necesitan permisos adicionales
		de acceso a datos para Amazon OpenSearch Service Serverless en la Guía para desarrolladores OpenSearch de servicios.
Vea el panel de control en la consola OpenSearch de servicio	CloudWatchOpenSearchDashboardAccess o CloudWatchOpenSearchDashboardsFullAccess	Especifique el rol o el usuario al crear la integración, o edite la política de acceso a los datos de la colección para agregar estos roles o usuarios. Para obtener más información, consulte Control de acceso a datos para Amazon OpenSearch Service Serverless en la Guía para desarrolladores OpenSearch de servicios.

Permisos que necesita la integración

Si crea un rol de IAM para que lo utilice la integración, en lugar de permitir que CloudWatch Logs cree el rol, debe incluir la siguiente política de permisos y confianza. Para obtener más información sobre cómo crear un rol de IAM, consulte Crear un rol para delegar permisos a un AWS servicio.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsAccess",
      "Effect": "Allow",
      "Action": [
```

```
        "logs:StartQuery",
        "logs:GetLogGroupFields",
        "logs:GetQueryResults"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "CloudWatchLogsDescribeLogGroupsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    },
    {
        "Sid": "AmazonOpenSearchCollectionAccess",
        "Effect": "Allow",
        "Action": [
            "aoss:APIAccessAll"
        ],
        "Resource": "*",
        "Condition": {
            "StringLike": {
                "aoss:collection": "cloudwatch-logs-*"
            }
        }
    }
  ]
}
```

Note

El rol anterior otorga acceso a la lectura de todos los grupos de registros de la cuenta, lo que permite crear paneles para cualquier cuenta de registro, incluidos los grupos de registros entre cuentas. Si desea restringir el acceso a grupos de registros específicos y crear paneles solo para esos grupos de registros, se puede actualizar la primera declaración de esa política de la siguiente manera:

```
{
```

```
      "Sid": "CloudWatchLogsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:StartQuery",
        "logs:GetLogGroupFields",
        "logs:GetQueryResults"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:123456789012:log-group:myLogGroup:*",
        "arn:aws:logs:us-east-1:123456789012:log-group:myLogGroup"
      ]
}
```

Acceda a los registros con la integración de tablas S3

La integración de S3 Tables con CloudWatch le permite acceder a los datos de registro ingeridos CloudWatch mediante motores de análisis como Amazon Athena, Amazon Redshift y herramientas de terceros que admiten la conexión a las tiendas de Apache. Iceberg-compatible Esta integración le permite realizar un análisis exhaustivo de los registros con las herramientas que prefiera y correlacionar los datos de los CloudWatch registros con los que no son datos. CloudWatch

Comprensión de la integración de tablas de S3

Amazon S3 Tables Integration es una solución totalmente gestionada que permite que los registros de los CloudWatch registros estén disponibles como tablas de Amazon S3 gestionadas. Con esta integración, obtiene una mayor flexibilidad a la hora de analizar sus registros, además de las funciones de CloudWatch Logs.

La integración funciona mediante la creación de un bucket de tablas de Amazon S3 gestionado (`aws-cloudwatch`) y la asociación de fuentes de registro específicas con tablas de Amazon S3 en función del nombre y el tipo de fuente de datos (que se pueden gestionar desde la pestaña Administración de CloudWatch registros > Fuentes de datos de Logs Console). Una vez asociados, se puede acceder a los datos de CloudWatch Logs a través de las tablas de Amazon S3 que utilizan el formato Apache Iceberg. Este formato proporciona una forma estandarizada para que varios motores de análisis consulten los datos de manera eficiente. Esta función está disponible sin coste adicional.

Componentes básicos

Tablas de Apache Iceberg

: El formato de tabla subyacente utilizado por S3 Tables, que proporciona almacenamiento de datos estructurado y permite la compatibilidad con varios motores de análisis.

Asociación de fuentes de datos

: El proceso de vincular fuentes de CloudWatch registros específicas a la integración de S3 Tables en función de la fuente de datos y los criterios de tipo.

Flujo de datos a tablas de S3

Comprender cómo fluyen los datos entre CloudWatch los registros y las tablas de S3 le ayuda a planificar la integración y administrar los datos de registro de manera eficaz.

Al crear una asociación, CloudWatch Logs envía automáticamente nuevos eventos de registro que coinciden con el nombre y el tipo de la fuente de datos asociada a un depósito de tabla de CloudWatch-managed S3. Estos eventos aparecen en el espacio de nombres de los registros, en la tabla correspondiente a esa fuente de datos. Los procesos de integración solo registran los eventos recibidos una vez creada la asociación. Los datos de registro existentes no se rellenan. Al crear una integración de tablas de S3, si deja seleccionada la casilla de verificación Habilitar todos los tipos y fuentes de registro disponibles en la tabla de S3, todas las fuentes de datos de su cuenta se asocian automáticamente y se envían a las tablas de S3 de forma predeterminada, incluidas las fuentes de datos que se agreguen en el futuro. Para enviar solo fuentes de datos específicas a las tablas de S3, desactive esta casilla de verificación durante la creación de la integración y, a continuación, asocie individualmente las fuentes de datos que desee incluir.

La retención de datos en el depósito de tablas de S3 coincide con la política de retención establecida para el grupo de registros. Por ejemplo, si estableces un grupo de registros con una retención de 1 día, CloudWatch Logs elimina los datos tanto de los CloudWatch registros como de la tabla de S3 al cabo de un día. Al eliminar un grupo de registros o un flujo de CloudWatch registros, Logs también elimina los datos del depósito de tablas de S3.

Cuándo usar la integración de tablas de S3

Considere la posibilidad de utilizar la integración de S3 Tables para correlacionar los datos de registro con otros datos externos o que no sean CloudWatch datos o cuando prefiera utilizar otras herramientas de análisis, como Amazon Athena, para realizar análisis CloudWatch de los datos de Logs. Utilice esta integración cuando necesite funciones que vayan más allá de las disponibles en CloudWatch Logs. Esta integración es especialmente valiosa cuando:

- Necesita ejecutar SQL-like consultas complejas en grandes volúmenes de datos de registro
- Desea integrar el análisis de registros con los flujos de trabajo y las herramientas de análisis existentes
- Necesita capacidades integrales de análisis de registros que abarquen múltiples fuentes de datos

Las tablas de S3 creadas mediante esta integración no conllevan cargos adicionales de almacenamiento o mantenimiento de tablas, aparte de los precios actuales de CloudWatch ingesta y almacenamiento.

Requisitos previos

Antes de implementar la integración, asegúrese de disponer de lo siguiente:

- Datos CloudWatch de registros existentes
- Los permisos de IAM adecuados para el acceso entre servicios entre CloudWatch registros y tablas de S3, tal y como se describe en la siguiente sección

Permisos de IAM

Para integrar CloudWatch los registros con las tablas de S3, debe configurar los permisos de IAM para dos entidades distintas: el usuario o la función que configura la integración y la función de servicio que CloudWatch Logs asume al escribir los datos en las tablas de S3.

Para el rol o el usuario que crea la integración

El usuario o rol que configura la integración requiere los siguientes permisos:

- `observabilityadmin:CreateS3TableIntegration`para crear la integración y `logs:AssociateSourceToS3TableIntegration` añadir fuentes
- `s3tables:CreateTableBuckets3tables:PutTableBucketEncryption`, y `s3tables:PutTableBucketPolicy` para configurar el bucket de tablas de S3

Para el rol de servicio

Adjunte la siguiente política de IAM a la función de servicio de IAM que CloudWatch Logs utiliza para escribir datos en el depósito de la tabla. Esta política otorga permiso para escribir en las tablas. Sustituya *`aws-region`* y *`log-group-name`* por su AWS región, ID de cuenta y nombre de grupo de registro. *`123456789012`*

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

```
            "Effect": "Allow",
            "Action": [
                "logs:integrateWithS3Table"
            ],
            "Resource": ["arn:aws:logs:aws-region:123456789012:log-group:log-group-
name"],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

Adjunte la siguiente política de confianza a la función de servicio de IAM que asumirá CloudWatch Logs al escribir los datos de registro en las tablas de S3. Este rol se crea o selecciona durante la configuración de la integración. Las condiciones restringen la función, por lo que CloudWatch Logs solo puede asumirla para la cuenta y el grupo de registros especificados. Sustituya *aws-region* y *log-group-name* por su AWS región, ID de cuenta y nombre del grupo de registros. *123456789012*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:aws-region:123456789012:log-
group:log-group-name"]
                }
            }
        }
    ]
```

```
}
```

Política de claves de KMS (para datos cifrados)

Si utiliza una clave gestionada por el cliente para cifrar los datos de registro, debe conceder acceso a la clave al director del CloudWatch servicio y al principal del servicio de mantenimiento de S3 Tables. Añada las siguientes declaraciones a su política de claves de KMS. Sustituya los valores de los marcadores de posición por su Cuenta de AWS ID, región, ID de clave de KMS y ARN de tabla S3 o bucket de tabla.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EnableSystemTablesKeyUsage",
            "Effect": "Allow",
            "Principal": {
                "Service": "systemtables.cloudwatch.amazonaws.com"
            },
            "Action": [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:aws-region:123456789012:key/key-id",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        },
        {
            "Sid": "EnableKeyUsage",
            "Effect": "Allow",
            "Principal": {
                "Service": "maintenance.s3tables.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:aws-region:123456789012:key/key-id",
```

```
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "<table-or-table-bucket-arn>/*"
                }
            }
        }
    ]
}
```

Introducción

Para empezar a integrar las tablas de S3, debe configurar la integración entre sus CloudWatch registros y las tablas de S3. Este proceso implica configurar las asociaciones de fuentes de datos y configurar los permisos de IAM adecuados.

Para crear una integración de tablas de S3

1. Abra la consola CloudWatch de Logs en https://console.aws.amazon.com/cloudwatch/».
2. Elija Configuración, Global, Crear integración de tablas S3.
3. Personalice la forma en que se cifrarán los registros en las tablas de S3 y la función que CloudWatch los registros utilizarán para escribir los registros en las tablas de S3.
4. Si desea que todas las fuentes de datos se asocien automáticamente a la integración, deje seleccionada la casilla Habilitar que todos los tipos y fuentes de registro estén disponibles en la tabla de S3 (está seleccionada de forma predeterminada). Si desea asociar solo fuentes de datos específicas, desactive esta casilla de verificación.
5. Seleccione Crear integración de tablas S3.

> **Note**
>
> Si seleccionó Habilitar todas las fuentes y tipos de registro para que estén disponibles en la tabla S3 durante la creación, todas las fuentes de datos se asociarán automáticamente, incluidas las que se agreguen en el futuro. Puedes detenerte aquí. Los siguientes pasos solo son necesarios si ha desactivado la casilla de verificación y desea asociar fuentes de datos específicas.

Para asociar fuentes a una integración de tablas de S3

1. Abra la consola CloudWatch de registros en https://console.aws.amazon.com/cloudwatch/».
2. Seleccione Configuración, Global y Gestione la integración de tablas de S3.
3. Elija Asociar fuente de datos.
4. Seleccione el nombre y el tipo de fuente de datos para los que desee habilitar la integración.
5. Elija Asociar fuente de datos.

Para asociar fuentes a una integración de tablas de S3 desde la página de administración de registros

1. Abra la consola CloudWatch de registros en https://console.aws.amazon.com/cloudwatch/».
2. Seleccione Administración de registros en el panel de navegación.
3. Seleccione la pestaña Fuentes de datos.
4. Elija el nombre y el tipo de fuente de datos que desee integrar.
5. Elija las acciones de la fuente de datos.
6. Seleccione Asociar con la integración de tablas de S3.
7. Revise las fuentes de datos y, a continuación, elija Asociar fuente de datos.

Antes de poder utilizar los datos, debe realizar los siguientes 3 pasos:

1. Integración de las tablas de Amazon S3 con los servicios de AWS análisis: uso de la consola de Amazon S3
2. Configurar los permisos de Lake Formation
3. Conéctese con las herramientas de análisis

Integración de las tablas de Amazon S3 con AWS servicios de análisis: uso de la consola Amazon S3 (Link)

Para habilitar la integración de las tablas S3 mediante la consola S3

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. En el panel de navegación izquierdo, elija Buckets de tablas.
3. Selecciona Activar la integración en la parte superior.

4. La primera vez que integra grupos de tablas en una región, Amazon S3 crea un nuevo rol de servicio de IAM en su nombre. Esta función permite a Lake Formation acceder a todos los grupos de tablas de tu cuenta y federar el acceso a tus tablas en AWS Glue Data Catalog.

Configurar los permisos de Lake Formation

Si bien CloudWatch Logs tiene permiso para escribir en la tabla (configurado en los pasos anteriores), los usuarios y los roles de análisis no tienen permiso automáticamente para leer los datos. Debe conceder el acceso de forma explícita mediante AWS Lake Formation. Debe hacer esto para cada director de IAM al que desee dar acceso a la tabla.

Para conceder acceso a las consultas a los usuarios o roles

Debe conceder los permisos SELECT y DESCRIBE a los principales de IAM (usuarios o roles) que ejecutarán las consultas en Athena o Redshift.

1. Abre la consola de AWS Lake Formation.
2. En el panel de navegación, en Permisos, seleccione Permisos de lago de datos.
3. Elija Conceder.
4. Responsables: seleccione los usuarios o las funciones de IAM a las que es necesario acceder (p. ej., sus analistas de datos o la función de administrador que utiliza actualmente).
5. LF-Tags o recursos del catálogo: seleccione los recursos del catálogo de datos con nombre asignado.
6. Bases de datos y tablas:
   - Seleccione el depósito de tablas de S3 creado por la CloudWatch integración (`aws-cloudwatch`).
   - Seleccione la tabla específica asociada a su fuente de datos (opcional).
7. Permisos de tabla: seleccione Seleccionar y describir.
8. Elija Conceder.

> **Note**
>
> Si encuentra errores de «Acceso denegado» al consultar los registros en Athena, asegúrese de que el usuario que ejecuta la consulta tenga los permisos de IAM adecuados para Athena y los permisos de Lake Formation definidos anteriormente.

Obtenga más información sobre los permisos de Lake Formation en https://docs.aws.amazon.com/lake-formation/latest/dg/granting-catalog-permissions.html.

Conéctese con las herramientas de análisis

Una vez concedidos los permisos, puede configurar su servicio de análisis preferido para consultar las tablas de S3. Las tablas S3 utilizan el formato Apache Iceberg, que Amazon Athena, Amazon Redshift y Amazon EMR admiten de forma nativa.

Para consultar datos de registro en Amazon Athena

Amazon Athena interactúa con S3 Tables a través del catálogo de Amazon S3 Tables.

Para configurar Athena para que consulte sus datos de registro

1. Abra la consola de Amazon Athena en. https://console.aws.amazon.com/athena/
2. En el editor de consultas, seleccione el catálogo de Amazon S3 Tables en el menú desplegable de fuentes de datos.
3. Si no ve el catálogo, asegúrese de haber completado los pasos de permiso de Lake Formation anteriores para su función de usuario específica.
4. Una vez seleccionado el catálogo, las tablas de registro aparecerán en la lista de bases de datos. Ahora puede ejecutar consultas SQL estándar con sus datos de registro.

Ejemplo de consulta: `SELECT * FROM "amazon_vpc__flow" LIMIT 100;`

Obtenga más información sobre cómo conectar los servicios de análisis con las tablas de S3 en https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-tables-integrating-aws.html.

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.

Si ha configurado cuentas de miembros AWS Organizations y trabaja con ellas, puede utilizar la centralización de registros para recopilar datos de registro de las cuentas de origen y convertirlos en una cuenta de supervisión central.

Al trabajar con grupos de registros centralizados, puede utilizar estas dimensiones de los campos del sistema al crear filtros de métricas.

- `@aws.account`- Esta dimensión representa el ID de AWS cuenta desde el que se originó el evento de registro.
- `@aws.region`- Esta dimensión representa la AWS región en la que se generó el evento de registro.

Estas dimensiones ayudan a identificar el origen de los datos de registro, lo que permite filtrar y analizar de forma más detallada las métricas derivadas de los registros centralizados. Para obtener más información, consulte Cross-account Centralización de registros entre regiones.

Si un grupo de registro con una suscripción utiliza la transformación de registros, el patrón de filtro se aplica a las versiones transformadas de los eventos de registro. Para obtener más información, consulte Transformación de los registros durante la ingestión.

> **Note**
>
> Los filtros de métricas solo se admiten en los grupos de registro de la clase de registro Estándar. Para obtener más información acerca de las clases de registros, consulte Clases de registro.

Puede utilizar 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. Al probar un patrón de filtro, la vista previa Resultados del filtro muestra hasta las 50 primeras líneas de registro coincidentes con fines de validación. Si la marca de tiempo de los resultados filtrados es anterior a la hora de creación de la métrica, no se muestra ningún registro.

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. Como las dimensiones forman parte del identificador único de una métrica, cada vez que se extrae un name/value par único de los 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 de métricas de las consultas de CloudWatch Logs Insights
> 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 Registra la sintaxis de consulta del lenguaje 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 registro 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

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 name/value valores y solo están disponibles para JSON y patrones de filtro 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 cobres altos cargos accidentales, Amazon podría deshabilitar tu filtro de métricas si genera 1000 name/value pares diferentes para dimensiones específicas durante un período de tiempo determinado.
> Puede crear alarmas de facturación que le notifiquen los cargos estimados. Para obtener más información, consulte Creación de una alarma de facturación para monitorear los cargos estimados de AWS.

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 registro).
3. Seleccione o cree un grupo de registro.

   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 registro
- 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 registro

Para crear un filtro de métrica para un grupo de registro, 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 registro).
3. Elija el nombre del grupo de registro.
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) Si utiliza grupos de registros centralizados, en Criterios de selección de filtros, puede especificar los filtros en función de la cuenta de origen (`@aws.account`), la región de origen (`@aws.region`) o ambas condiciones.
7. (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.
8. Elija Next (Siguiente) y luego ingrese un nombre para el filtro de métricas.
9. 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.
10. Para Metric name (Nombre de métrica), ingrese un nombre para la nueva métrica.
11. 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`.

12. (Opcional) En Unit (Unidad), seleccione una unidad para asignar a la métrica. Si no especifica una unidad, se configura como `None`.
13. (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.

14. 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 registro). 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 registro.
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 registro. 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 registro.
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 (Detalles de métrica), en Metric Namespace (Espacio de nombres de métrica), 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. Seleccione Siguiente y, a continuación, en Nombre del filtro, escriba HTTP404Errores.
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 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 registro 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. Para el nombre de la métrica, introduzca 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 registro 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 métrico, utilice 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 registro.

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 registro, 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 registro.

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 registro 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 registro.
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.

También puede utilizar la centralización de CloudWatch registros para replicar los datos de registro de varias cuentas y regiones en una ubicación central. Para obtener más información, consulte Cross-account Centralización de registros entre regiones.

Para empezar a suscribirse al registro de eventos, cree el recurso receptor, como una transmisión de Amazon Kinesis Data Streams, donde se entregarán los eventos. Un filtro de suscripción define el patrón de filtrado que se utilizará para filtrar los eventos de registro que se enviarán a su AWS recurso, así como la información sobre a dónde enviar los eventos de registro coincidentes. Los eventos de registro se envían al recurso receptor poco después de ser ingeridos, normalmente en menos de tres minutos.

> **Note**
>
> Si un grupo de registros con una suscripción utiliza la transformación de registros, el patrón de filtro se compara con las versiones transformadas de los eventos de registro. Para obtener más información, consulte Transformación de los registros durante la ingestión.

Puede crear suscripciones en el nivel de cuenta y en el de grupo de registro. Cada cuenta puede tener un filtro de suscripciones a nivel de cuenta por región. Cada grupo de registro 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 Logs también genera 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 en los grupos de registro de la clase de registro Estándar. Para obtener más información acerca de las clases de registros, consulte Clases de registro.

> **Note**
>
> Los filtros de suscripción pueden recopilar eventos de registro 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 es posible.

Para el procesamiento por lotes y el análisis de los datos de registro de forma programada, considere la posibilidad de utilizarAutomatizar el análisis de registros con consultas programadas. Las consultas programadas ejecutan consultas de CloudWatch Logs Insights automáticamente y entregan los resultados a destinos como los buckets de Amazon S3 o los autobuses de EventBridge eventos de Amazon.

> **Note**
>
> Los filtros de suscripción garantizan la entrega de los eventos al menos una vez, aunque en ocasiones pueden producirse eventos duplicados.

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 Amazon 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 registro, 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 registro 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

Son los criterios que se usan para seleccionar los grupos de registro a los que se ha aplicado el filtro de suscripción a nivel de cuenta. Si no los especifica, el filtro de suscripción de nivel de cuenta se aplica a todos los grupos de registro de la cuenta. Este campo se utiliza para evitar bucles de registro infinitos. Para obtener más información sobre el problema de bucles de registro infinitos, consulte Prevención de recursión de registros.

Los criterios de selección tienen un límite de 25 KB.

En el caso de los grupos de registros centralizados, también se incluyen los siguientes elementos clave. Estos elementos se pueden utilizar como criterios de selección de campo para ayudar en la identificación de la fuente de los datos de registro, lo que permite filtrar y analizar de forma más detallada las métricas derivadas de los registros centralizados.

@aws.account

Este campo identifica el ID de AWS cuenta desde el que se originó el evento de registro.

@aws.region

Este campo identifica la AWS región en la que se generó el evento de registro.

Filtros de suscripción a nivel de grupo de registro

Puede utilizar un filtro de suscripción con Amazon Kinesis Data Streams AWS Lambda, Amazon Data Firehose o Amazon Service. OpenSearch Todos los registros enviados a un servicio a través de un filtro de suscripción están codificados en base64 y comprimidos con el formato gzip. Si

utiliza registros centralizados con su cuenta AWS Organizations, puede optar por emitir el campo `@aws.account` y del `@aws.region` sistema para identificar qué datos provienen de qué cuentas y regiones de su organización. En esta sección se proporcionan ejemplos que puede seguir para crear un filtro de suscripción de CloudWatch registros que envíe datos de registro a Firehose, Lambda, Amazon Kinesis Data Streams y Service. OpenSearch

> **Note**
>
> Si quiere buscar sus datos de registro, consulte Sintaxis de patrones y filtros.

Ejemplos

- Ejemplo 1: filtros de suscripción con Amazon Kinesis Data Streams
- Ejemplo 2: filtros de suscripción con AWS Lambda
- Ejemplo 3: filtros de suscripción con Amazon Data Firehose
- Ejemplo 4: filtros de suscripción con Amazon OpenSearch Service

Ejemplo 1: filtros de suscripción con Amazon Kinesis Data Streams

El siguiente ejemplo asocia un filtro de suscripción con un grupo de registro que contiene eventos de AWS CloudTrail . El filtro de suscripción envía todas las actividades registradas realizadas con AWS las credenciales «Root» a una transmisión de Amazon 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, lo que 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 su transmisión en Amazon 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 Amazon 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 Amazon Kinesis Data Streams

1. Crear un flujo de destino mediante 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 Amazon 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 con el rol mediante el siguiente comando put-role-policy:

```
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 de CloudWatch Logs. El filtro de suscripción inicia de inmediato el flujo de datos de registro en tiempo real desde el grupo de registro 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 comprobar que esto está ocurriendo cogiendo un iterador de fragmentos de Amazon Kinesis Data Streams y utilizando el comando get-records de Amazon Kinesis Data Streams para recuperar algunos registros de Amazon 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 puede que tenga que realizar esta llamada varias veces antes de que Amazon Kinesis Data Streams comience a devolver datos.

Cabe esperar ver una respuesta en una gama de registros. El atributo Data de un registro de Amazon Kinesis Data Streams está codificado en base64 y comprimido en 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 registro 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 Amazon 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 registro del marcador por el grupo de registro 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 registro del marcador por el grupo de registro 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 ver el resultado de la función Lambda, navegue hasta la función Lambda, donde verá el resultado 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 registro 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, puede usar la característica de descompresión de Firehose para descomprimir automáticamente los registros. Para obtener más información, consulte Enviar CloudWatch registros a Firehose.

> **Note**
>
> Antes de crear el flujo de Firehose, calcule el volumen de los datos de registro que se generarán. Asegúrese de crear un flujo de Firehose que pueda gestionar este volumen. Si el flujo no puede gestionar el volumen, se limitará el flujo de registros. Para obtener más información acerca de los límites de volumen del flujo de Firehose, consulte Límites de datos de Amazon Data Firehose.

Creación de un filtro de suscripción para Firehose

1. Cree un bucket de Amazon Simple Storage Service (Amazon S3). Le recomendamos que utilice 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 amzn-s3-demo-bucket2 --create-bucket-configuration
 LocationConstraint=region
```

A continuación, se muestra un ejemplo de la salida:

```
{
    "Location": "/amzn-s3-demo-bucket2"
}
```

2. Cree el rol de IAM que concede permiso a Amazon Data Firehose para incluir 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. Cree una política de permisos para definir las acciones que Firehose puede realizar en su 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:::amzn-s3-demo-bucket2",
          "arn:aws:s3:::amzn-s3-demo-bucket2/*" ]
    }
  ]
}
```

5. Asocie la política de permisos con el rol mediante el siguiente comando put-role-policy:

```
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 el rol y el bucket que creó: ARNs

```
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:::amzn-s3-demo-bucket2"}'
```

Tenga en cuenta que Firehose utiliza automáticamente un prefijo en formato de hora YYYY/MM/DD/HH UTC 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:::amzn-s3-demo-bucket2",
                    "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 con el rol utilizando el comando put-role-policy:

```
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 de inmediato el flujo de datos de registro en tiempo real desde el grupo de registro elegido a su flujo 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. Los datos comenzarán a aparecer en su instancia de Amazon S3 en función del intervalo de búfer de tiempo definido en el flujo 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 'amzn-s3-demo-bucket2' --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 'amzn-s3-demo-bucket2' --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
```

Ejemplo 4: filtros de suscripción con Amazon OpenSearch Service

En este ejemplo, crearás una suscripción a CloudWatch Logs que envía los eventos de registro entrantes que coincidan con tus filtros definidos a tu dominio de OpenSearch servicio.

Para crear un filtro de suscripción para el OpenSearch Servicio

1. Cree un dominio OpenSearch de servicio. Para obtener más información, consulte Creación de dominios OpenSearch de servicio
2. Abra la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
3. En el panel de navegación, seleccione Grupos de registro.
4. Seleccione el nombre del grupo de registro.
5. Selecciona Acciones, Filtros de suscripción o Crear filtro de suscripción a Amazon OpenSearch Service.
6. Elija si desea transmitir a un clúster de esta cuenta u otra cuenta.
   - Si se eligió esta cuenta, seleccione el dominio que creó en el paso 1.

- Si eligió Otra cuenta, introduzca el ARN y el punto de conexión de ese dominio.

7. Si eligió otra cuenta, proporcione el ARN del dominio y el punto de enlace.
8. Para el clúster OpenSearch de Amazon Service, elija el nombre del clúster al que se entregarán los datos del grupo de registros.
9. Elija un formato de registro.
10. En Patrón de filtro de suscripción, escriba los términos o el patrón que desea buscar en los eventos de registro. Esto garantiza que envíe solo los datos que le interesan a su clúster OpenSearch de servicio. Para obtener más información, consulte Sintaxis del patrón de filtro para filtros métricos.
11. (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.
12. Elija Start streaming (Comenzar streaming).

Filtros de suscripción a nivel de cuenta

> **Important**
>
> Existe el riesgo de provocar un ciclo recursivo infinito con los filtros de suscripción que, si no se aborda, puede provocar un gran aumento de la facturación por incorporación. 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 registro que incorporan 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 registro excluir, consulte Prevención de recursión de registros.

Puede establecer una política de suscripción a nivel de cuenta que incluya un subconjunto de grupos de registro en la cuenta. La política de suscripción de la cuenta puede funcionar con Amazon Kinesis Data Streams AWS Lambda o Amazon Data Firehose. Todos los registros enviados a un servicio a través de una política de suscripción a nivel de cuenta están codificados en base64 y comprimidos con el formato gzip. En esta sección se proporcionan ejemplos que puede seguir para crear una suscripción a nivel de cuenta para Amazon Kinesis Data Streams, Lambda y Firehose.

> **Note**
>
> Para ver una lista de todas las políticas de filtrado de suscripciones de su cuenta, utilice el comando `describe-account-policies` 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 Amazon 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 Amazon Kinesis Data Streams

Antes de crear una transmisión de datos de Amazon Kinesis Data Streams para utilizarla 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 un flujo no tiene suficientes fragmentos, se limitará. Para obtener más información sobre los límites de volumen de transmisión, consulte Cuotas y límites en la documentación de Amazon Kinesis Data Streams.

> **Warning**
>
> Como los eventos de registro de varios grupos de registro 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 Amazon Kinesis Data Streams CloudWatch con métricas. Esto le permitirá identificar cualquier limitación y ajustar la configuración en consecuencia. Por ejemplo, la `DeliveryThrottling` métrica registra el número de eventos de registro por los que se limitó 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 su transmisión en Amazon 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 Amazon Kinesis Data Streams. Si envía demasiados datos al flujo, es posible que deba reducir el tamaño del filtro o ajustar sus criterios.

En el siguiente ejemplo, se utiliza una política de suscripción a nivel de cuenta para reenviar todos los eventos de registro a una transmisión de Amazon Kinesis Data Streams. El patrón de filtro hace coincidir los eventos del registro con el texto `Test` y los reenvía a la transmisión de Amazon Kinesis Data Streams.

Para crear una política de suscripción a nivel de cuenta para Amazon Kinesis Data Streams

1. Crear un flujo de destino mediante 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": "EXAMPLE84634633746074317682114555",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
```

```
                    "EXAMPLE6888184566795038319814587845913527021815726l0"
                }
            }
        ]
    }
}
```

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 utilice la consola de IAM para crearla.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/TestStream"
    }
  ]
}
```

6. Asocie la política de permisos con el rol mediante el siguiente comando put-role-policy:

```
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 la política de filtrado de suscripciones de CloudWatch Logs. La política inicia de inmediato el flujo de datos del registro en tiempo real hacia su flujo. En este ejemplo, se transmiten todos los eventos de registro que contienen la cadena `ERROR`, excepto los de los grupos de registro 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 campo `selection-criteria` es opcional, pero es importante usarlo si quiere excluir de un filtro de suscripción los grupos de registro que pueden provocar una recursión infinita de registros. Para obtener más información sobre este problema y determinar qué grupos de registro excluir, consulte Prevención de recursión de registros. Actualmente, NOT IN es el único operador compatible para `selection-criteria`.

   Puede comprobar el flujo de eventos de registro utilizando un iterador de fragmentos de Amazon Kinesis Data Streams y utilizando el comando Amazon Kinesis Data Streams para obtener algunos registros de Amazon 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 Amazon Kinesis Data Streams comience a devolver datos.

Cabe esperar ver una respuesta en una gama de registros. El atributo Data de un registro de Amazon Kinesis Data Streams está codificado en base64 y comprimido en 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 en la estructura de datos anterior son los siguientes:

messageType

Los mensajes de datos utilizarán el tipo “DATA_MESSAGE”. A veces, CloudWatch los registros pueden emitir registros de Amazon 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 registro 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.

policyLevel

Es el nivel en el que se aplicó la política. “ACCOUNT_LEVEL_POLICY” es el `policyLevel` de la política de filtrado de suscripciones a nivel 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, se limitará el flujo de registros. Como los eventos de registro de todos los grupos de registro o de un subconjunto de los grupos de registro 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.

Creación de una política de filtrado de suscripciones a nivel de cuenta para Lambda

1. Crea 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. Use el siguiente comando para reemplazar la cuenta de marcador de posición por la suya propia.

```
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. En este ejemplo, se transmiten todos los eventos de registro que contienen la cadena ERROR, excepto los de los grupos de registro denominados `LogGroupToExclude1` y `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 tu transmisión todos los eventos de registro entrantes que coincidan con el patrón del filtro y los criterios de selección.

El campo `selection-criteria` es opcional, pero es importante usarlo si quiere excluir de un filtro de suscripción los grupos de registro que pueden provocar una recursión infinita de registros. Para obtener más información sobre este problema y determinar qué grupos de registro excluir, consulte Prevención de recursión de registros. Actualmente, NOT IN es el único operador compatible para `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 ver el resultado de la función Lambda, navegue hasta la función Lambda, donde verá el resultado 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 registro de la función de Lambda de destino. Esto se hace para evitar una recursión infinita de registros que pueda provocar un aumento en la facturación por incorporación. Para obtener más información sobre este problema, consulte Prevención de recursión de registros.

Los elementos clave en la estructura de datos anterior son los siguientes:

messageType

Los mensajes de datos utilizarán el tipo “DATA_MESSAGE”. A veces, CloudWatch los registros pueden emitir registros de Amazon 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 registro 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.

policyLevel

Es el nivel en el que se aplicó la política. “ACCOUNT_LEVEL_POLICY” es el `policyLevel` de la política de filtrado de suscripciones a nivel 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, puede usar la característica 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 el flujo de Firehose, calcule el volumen de los datos de registro que se generarán. Asegúrese de crear un flujo de Firehose que pueda gestionar este volumen. Si el flujo no puede gestionar el volumen, se limitará el flujo de registros. Para obtener más información acerca de los límites de volumen del flujo de Firehose, consulte Límites de datos de Amazon Data Firehose.

Creación de un filtro de suscripción para Firehose

1. Cree un bucket de Amazon Simple Storage Service (Amazon S3). Le recomendamos que utilice 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 amzn-s3-demo-bucket2 --create-bucket-configuration
 LocationConstraint=region
```

A continuación, se muestra un ejemplo de la salida:

```
{
    "Location": "/amzn-s3-demo-bucket2"
}
```

2. Cree el rol de IAM que concede permiso a Amazon Data Firehose para incluir 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. Cree una política de permisos para definir las acciones que Firehose puede realizar en su 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:::amzn-s3-demo-bucket2",
          "arn:aws:s3:::amzn-s3-demo-bucket2/*" ]
    }
  ]
}
```

5. Asocie la política de permisos con el rol mediante el siguiente comando put-role-policy:

```
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 el rol y el bucket que creó: ARNs

```
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:::amzn-s3-demo-bucket2"}'
```

NFirehose utiliza automáticamente un prefijo en formato de hora YYYY/MM/DD/HH UTC 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 unos minutos hasta que se active el flujo. 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:::amzn-s3-demo-bucket2",
                    "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 con el rol utilizando el comando put-role-policy:

```
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 de inmediato el flujo de datos de registro en tiempo real desde el grupo de registro 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 campo `selection-criteria` es opcional, pero es importante usarlo si quiere excluir de un filtro de suscripción los grupos de registro que pueden provocar una recursión infinita de registros. Para obtener más información sobre este problema y determinar qué grupos de registro excluir, consulte Prevención de recursión de registros. Actualmente, NOT IN es el único operador compatible para `selection-criteria`.

    Los datos comenzarán a aparecer en su instancia de Amazon S3 en función del intervalo de búfer de tiempo definido en el flujo 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 'amzn-s3-demo-bucket2' --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 'amzn-s3-demo-bucket2' --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 entre cuentas 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 Amazon 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. Se puede recopilar una transmisión en tiempo real de los datos de eventos en esas cuentas y entregarla a los grupos de seguridad de la información, que pueden usar Amazon 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 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 Este de EE. UU. (Norte de Virginia).

Si ha configurado cuentas de miembros AWS Organizations y trabaja con ellas, puede utilizar la centralización de registros para recopilar datos de registro de las cuentas de origen en una cuenta de supervisión central.

Al trabajar con grupos de registros centralizados, se pueden utilizar estas dimensiones de los campos del sistema al crear filtros de suscripción.

- `@aws.account`- Esta dimensión representa el ID de AWS cuenta desde el que se originó el evento de registro.
- `@aws.region`- Esta dimensión representa la AWS región en la que se generó el evento de registro.

Estas dimensiones ayudan a identificar el origen de los datos de registro, lo que permite filtrar y analizar de forma más detallada las métricas derivadas de los registros centralizados.

Temas

- Intercambio de datos de registro entre cuentas y regiones mediante Amazon Kinesis Data Streams
- Uso compartido de datos de registro entre cuentas y regiones mediante Firehose

- Suscripciones a nivel de cuenta multirregional mediante Amazon Kinesis Data Streams
- Suscripciones a nivel de cuenta entre cuentas y regiones mediante Firehose

Intercambio de datos de registro entre cuentas y regiones mediante Amazon 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 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.

  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 Amazon Kinesis Data Streams y permite a CloudWatch Logs saber que el destinatario desea 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 AWS recurso 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

- Modificación de la suscripción al destino en tiempo de ejecución

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 Amazon 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 del destinatario, cree una transmisión de destino en Amazon 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 la. 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. Anote el valor de DestinationArn que se devuelve en la carga:

```
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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999: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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111111111111"
            },
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999: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 arn:aws:logs PutSubscriptionFilter::55559999:Destination:TestDestination. *region* 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 la función de IAM necesaria para las suscripciones de registros multicuenta, utilice 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. Cree una política de permisos para definir las acciones que CloudWatch Logs puede realizar en su 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: Permisos Add/validate de IAM para el destino de varias 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 de IAM o usuario de IAM se utiliza para ejecutar los comandos de registro. 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 *RoleName* o. *UserName*

2. Inicie sesión Consola de administración de AWS 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.

   La siguiente política proporciona permisos para crear un filtro de suscripción en cualquier recurso de destino solo en una sola AWS cuenta, la cuenta: 999999999999

   JSON

   ```
   {
        "Version":"2012-10-17",
        "Statement": [
           {
               "Sid": "AllowSubscriptionFiltersOnAccountResources",
               "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 nombrado `sampleDestination` en una sola AWS cuenta, la cuenta`123456789012`:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSubscriptionFiltersonAccountResource",
            "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 registro 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 registro 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 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 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 registro 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: Permisos Add/validate de IAM para el destino de varias 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 Amazon 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 Amazon Kinesis Data Streams y utilizando el comando aws kinesis get-records para obtener algunos registros de Amazon 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 comando get-records varias veces antes de que Amazon Kinesis Data Streams comience a devolver datos.

Debería ver una respuesta con una serie de registros de Amazon Kinesis Data Streams. El atributo de datos del registro de Amazon Kinesis Data Streams se comprime en formato gzip y, a continuación, se codifica 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 ID de AWS cuenta de los datos de registro originarios.

logGroup

El nombre del grupo de registro 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 Amazon 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

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "222222222222"
            },
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999: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 cuentas 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 entre cuentas de los registros creados por los servicios enumerados en Habilitar el registro desde AWS servicios. Si no está trabajando con registros creados por uno de estos grupos de registro, 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 y empezar a usar la organización IDs para los 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.

   JSON

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

```
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": "arn:aws:logs:us-
east-1:999999999999: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 usar una política de acceso con un identificador de organización en lugar de una política de acceso que muestre una AWS cuenta específica IDs, incluye el `force` parámetro.

> **Warning**
>
> Si está trabajando con registros enviados por uno de los AWS servicios incluidos en la listaHabilitar el registro desde AWS servicios, antes de realizar este paso, debe haber actualizado los filtros de suscripción de todas las cuentas de remitentes, tal y como se explica en la secciónPaso 1: actualizar los filtros de suscripción.

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Uso compartido 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 Amazon Kinesis Data Streams y permite a CloudWatch Logs saber que el destinatario desea 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.

En el ejemplo de esta sección, se utiliza un flujo de entrega de Firehose con almacenamiento de Amazon S3. También puede configurar flujos de entrega de Firehose con diferentes parámetros. 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 AWS recurso al que apunta el destino puede estar ubicado en una región diferente.

> **Note**
>
> Se admite el filtro de suscripción de Firehose para el flujo de entrega de una misma cuenta y entre regiones.

Temas

- Paso 4: crear un filtro de suscripción
- Validación del flujo de eventos de registro
- Modificación de la suscripción al destino en tiempo de ejecución

Paso 1: creación de un flujo de entrega de Firehose

> **Important**
>
> Antes de realizar los siguientes pasos, debe utilizar una política de acceso para que Firehose pueda acceder a su bucket de Amazon S3. Para obtener más información, consulte Control del acceso 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.

Creación de un flujo de entrega de Firehose que se utilice como destino

1. Cree un bucket de Amazon S3:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-1
```

2. Cree el rol de IAM que concede permiso a Firehose para incluir datos en el bucket.

   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. Cree una política de permisos para definir las acciones que Firehose puede realizar en su 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:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
    }]
}
```

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. Introduzca el comando siguiente para crear el flujo de entrega de Firehose. Sustituya *my-role-arn* y *amzn-s3-demo-bucket2-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:::amzn-s3-demo-bucket"}'
```

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. Espere a que el flujo de Firehose que creó en Paso 1: creación de un flujo de entrega de Firehose se active. Puede utilizar el siguiente comando para comprobar la StreamDescription. 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:::amzn-s3-demo-bucket",
                       "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:::amzn-s3-demo-bucket",
                    "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

   JSON

```
{
  "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 entre varias cuentas utilizarán la acción PutSubscriptionFilter para enviar eventos de registro al destino:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Paso 3: Permisos Add/validate de IAM para el destino multicuenta

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 de IAM o usuario de IAM se utiliza para ejecutar los comandos de registro. 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 *RoleName* o. *UserName*

2. Inicie sesión Consola de administración de AWS 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.

   La siguiente política proporciona permisos para crear un filtro de suscripción en cualquier recurso de destino solo en una sola AWS cuenta, la cuenta: 999999999999

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid":
 "AllowSubscriptionFiltersOnAnyResourceInOneSpecificAccount",
            "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 nombrado `sampleDestination` en una sola AWS cuenta, la cuenta`123456789012`:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
        "Sid": "AllowSubscriptionFiltersOnSpecificResource",
            "Effect": "Allow",
```

```
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

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: Permisos Add/validate de IAM para el destino multicuenta.

```
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 su bucket de Amazon S3 en función del intervalo de tiempo de búfer que se establece en el flujo 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 'amzn-s3-demo-bucket'
```

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 'amzn-s3-demo-bucket' --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

   JSON

```
{
  "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 Amazon 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 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.

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 Amazon Kinesis Data Streams y permite a CloudWatch Logs saber que el destinatario desea 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 AWS recurso 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 Amazon 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 del destinatario, cree una transmisión de destino en Amazon 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 la. 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": "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. Anote el valor de DestinationArn que se devuelve en la carga:

```
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.

   JSON

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

```
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999: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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111111111111"
            },
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999: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 arn:aws:logs PutSubscriptionFilter::55559999:Destination:TestDestination. *region* 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: crear 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: crear 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 la función de IAM necesaria para las suscripciones de registros multicuenta, utilice 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. Cree una política de permisos para definir las acciones que CloudWatch Logs puede realizar en su 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: crear una política de filtrado de suscripciones a nivel de cuenta.

Paso 3: crear 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 registro 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 registro 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 de envío. El filtro se asocia a la cuenta 111111111111 del remitente para que cada evento de registro que coincida con el filtro y los criterios de selección se envíe al destino creado anteriormente. Ese destino encapsula una transmisión llamada "». RecipientStream

El campo `selection-criteria` es opcional, pero es importante usarlo si quiere excluir de un filtro de suscripción los grupos de registro que pueden provocar una recursión infinita de registros. Para obtener más información sobre este problema y determinar qué grupos de registro excluir, consulte Prevención de recursión de registros. Actualmente, NOT IN es el único operador compatible para `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 y el destino de la cuenta del remitente deben estar en la misma región de AWS . Sin embargo, el destino puede apuntar a un AWS recurso, como una transmisión de Amazon 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 a la transmisión 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 Amazon Kinesis Data Streams y utilizando el comando aws kinesis get-records para obtener algunos registros de Amazon 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 Amazon Kinesis Data Streams comience a devolver datos.

Debería ver una respuesta con una serie de registros de Amazon Kinesis Data Streams. El atributo de datos del registro de Amazon Kinesis Data Streams se comprime en formato gzip y, a continuación, se codifica 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 en la estructura de datos anterior son los siguientes:

messageType

Los mensajes de datos utilizarán el tipo “DATA_MESSAGE”. A veces, CloudWatch los registros pueden emitir registros de Amazon 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 registro 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.

policyLevel

Es el nivel en el que se aplicó la política. “ACCOUNT_LEVEL_POLICY” es el `policyLevel` de la política de filtrado de suscripciones a nivel 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

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": "222222222222"
            },
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
```

```
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999: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 cuentas 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 entre cuentas de los registros creados por los servicios enumerados en Habilitar el registro desde AWS servicios. Si no está trabajando con registros creados por uno de estos grupos de registro, 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 y empezar a usar la organización IDs para los 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. Introduzca el siguiente comando para actualizar la política de filtrado 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.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "logs:PutSubscriptionFilter",
                "logs:PutAccountPolicy"
            ],
            "Resource": "arn:aws:logs:us-
east-1:999999999999: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 usar una política de acceso con un identificador de organización en lugar de una política de acceso que muestre una AWS cuenta específica IDs, incluye el `force` parámetro.

> **Warning**
>
> Si está trabajando con registros enviados por uno de los AWS servicios incluidos en la listaHabilitar el registro desde AWS servicios, antes de realizar este paso, debe haber actualizado los filtros de suscripción de todas las cuentas de remitentes, tal y como se explica en la sección Paso 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 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 Amazon Kinesis Data Streams y permite a CloudWatch Logs saber que el destinatario desea

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.

En el ejemplo de esta sección, se utiliza un flujo de entrega de Firehose con almacenamiento de Amazon S3. También puede configurar flujos de entrega de Firehose con diferentes parámetros. 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 AWS recurso al que apunta el destino puede estar ubicado en una región diferente.

> **Note**
>
> Se admite el filtro de suscripción de Firehose para el flujo de entrega de una misma cuenta y entre regiones.

Temas

Paso 1: creación de un flujo de entrega de Firehose

> **Important**
>
> Antes de realizar los siguientes pasos, debe utilizar una política de acceso para que Firehose pueda acceder a su bucket de Amazon S3. Para obtener más información, consulte Control del acceso 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.

Creación de un flujo de entrega de Firehose que se utilice como destino

1. Cree un bucket de Amazon S3:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-1
```

2. Cree el rol de IAM que concede permiso a Firehose para incluir datos en el bucket.

   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. Cree una política de permisos para definir las acciones que Firehose puede realizar en su 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:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
    }]
}
```

   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. Introduzca el comando siguiente para crear el flujo de entrega de Firehose. Sustituya *my-role-arn* y *amzn-s3-demo-bucket2-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:::amzn-s3-demo-bucket"}'
```

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. Espere a que el flujo de Firehose que creó en Paso 1: creación de un flujo de entrega de Firehose se active. Puede utilizar el siguiente comando para comprobar la StreamDescription. 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:::amzn-s3-demo-bucket",
                    "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:::amzn-s3-demo-bucket",
                    "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). Puede utilizar un editor de texto para incluir esta política en el archivo ~/AccessPolicy.json:

JSON

```
{
  "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 acciones `logs:PutSubscriptionFilter` y `logs:PutAccountPolicy` para acceder al destino. Los usuarios entre cuentas utilizarán las acciones `PutSubscriptionFilter` y `PutAccountPolicy` para enviar eventos de registro al destino.

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Paso 3: crear una política de filtrado de suscripciones a nivel de cuenta

Cambie a la cuenta de envío, que es 111111111111 en este ejemplo. Ahora creará la política de filtrado de suscripciones a nivel de cuenta en la cuenta de envío. En el siguiente ejemplo, el filtro hace que cada evento de registro que contiene la cadena `ERROR` en todos los grupos de registro excepto en dos se envíe al destino creado 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 y el destino de la cuenta desde la que se realiza el envío deben estar en la misma región de AWS . 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 su bucket de Amazon S3 en función del intervalo de tiempo de búfer que se establece en el flujo 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 'amzn-s3-demo-bucket'
```

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 'amzn-s3-demo-bucket' --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 pueden tener el siguiente 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. Coloca esta política en el archivo ~/ .json: 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 puedes 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.

Se recomienda utilizar las claves de contexto de condición global `aws:SourceArn`, `aws:SourceAccount`, `aws:SourceOrgID` y `aws:SourceOrgPaths` en las políticas de recursos para limitar los permisos que concede a otro servicio para el 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 Amazon Kinesis Data Streams y Firehose Paso 2: creación de un destino y muestran cómo puede utilizar la clave contextual de la condición global para ayudar a evitar `aws:SourceArn` el confuso problema de los diputados.

Prevención de recursión de registros

Con los filtros de suscripción, existe el riesgo de provocar una recursión infinita de los registros, lo que, si no se impide, 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 suscripción está asociado a un grupo de registro que recibe eventos de registro como resultado del flujo de trabajo de entrega de suscripciones. Los registros incorporados en el grupo de registro se entregarán al destino, lo que provocará que el grupo de registro incorpore 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 y que envía eventos de registro a Amazon S3. Además, también hay una función de 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 registro de la función de Lambda, los eventos de registro que cree 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; en consecuencia, 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 ingesta 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 registro que forman parte de su flujo de trabajo de entrega de suscripciones. En el caso de los filtros de suscripción a nivel de

cuenta, use el parámetro `selectionCriteria` de la API `PutAccountPolicy` para excluir estos grupos de registro de la política.

Al excluir los grupos de registros, tenga en cuenta los siguientes AWS servicios que producen registros y pueden formar parte de los 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 que creó el grupo de registro 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 es necesario excluir el grupo de registro de la función de Lambda de destino mediante `selectionCriteria`, 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 Registra la sintaxis de consulta del lenguaje 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, los 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 una ocurrencia del término correspondiente. 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 “gris”
  - `%^starting|^initializing|^shutting down%` puede coincidir con “a partir de... ”, o “inicializando... ”, o “apagando”, pero no coincidirá con “omitir la inicialización... ”
  - `%abcc|ab[^c]$` puede coincidir con “abcc... ” y “aba... ” pero no coincidirá con “aac... ”
- `\`: 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 correspondiente. 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 correspondiente. 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 ignorarán todos los signos de interrogación.
> Por ejemplo, el siguiente patrón de filtrado coincide con todos los eventos que contienen la palabra `REQUEST`, pero se ignoran los términos del filtro del signo de interrogación (“?”) y no tienen 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 variable `NOT EXISTS` para devolver registros JSON que no contienen 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 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 requisito principal para los registros es el almacenamiento o el procesamiento en uno de estos servicios, puede conseguir fácilmente que el servicio que crea los registros los envíe directamente a Amazon S3 o a Firehose sin configuración adicional.

Incluso si publica registros directamente en Amazon S3 o Firehose, se aplican cargos de CloudWatch envío. Si envía registros a Amazon S3, *AWS_REGION*`-S3-Egress-Bytes` los cargos aparecen en Cost Explorer o en su factura. Si envías registros a Firehose, aparecen *AWS_REGION*`-FH-Egress-Bytes` los cargos. Para obtener más información sobre los precios de los registros vendidos, consulta la pestaña Logs en 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 como compatibles [permisos V1] y admitidos [permisos V2] en lathe section called “Destinos de registro compatibles”. Para obtener información sobre estos permisos necesarios, consulte las secciones que aparecen después de la tabla.
<integration-catalog>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>

<integration></integration>
<integration></integration>
<integration></integration>

<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
<integration></integration>
</integration-catalog>

Destinos de registro compatibles

En la siguiente tabla se muestran los destinos que admite cada AWS servicio para el envío de registros y la versión de permisos necesaria.

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Registros de acceso a Amazon API Gateway	Registros proporcionados	Compatibles [permisos V1]			
AWS AppSync logs	Registros personalizados	compatible			
Registros de Amazon Aurora MySQL	Registros personalizados	compatible			
Amazon Bedrock Registro de bases de conocimiento	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Amazon Bedrock Registro de agentes	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Amazon Bedrock AgentCore Tiempo de ejecución	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]
Amazon Bedrock AgentCore Puerta de enlace	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]
Amazon Bedrock AgentCore Identidad	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]
Amazon Bedrock AgentCore Memoria	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]
Amazon Bedrock AgentCore Pagos	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Amazon Bedrock AgentCore Herramientas	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]
Registros de métricas de calidad multimedia y registros de mensajes SIP de Amazon Chime	Registros proporcionados	Compatibles [permisos V1]			
CloudFront: registros de acceso	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
AWS CloudHSM registros de auditoría	Registros personalizados	compatible			
CloudWatch Evidentemente, registros de eventos de evaluación	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]		
CloudWatch Registros de Internet Monitor	Registros proporcionados		Compatibles [permisos V1]		

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
CloudTrail logs	Registros personalizados	compatible			
AWS CodeBuild logs	Registros personalizados	compatible			
Amazon CodeWhisperer registros de eventos	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Amazon Cognito logs	Registros proporcionados	Compatibles [permisos V1]			
Registros de clientes de Amazon Connect	Registros personalizados	compatible			
AWS DataSync logs	Registros personalizados	compatible			

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Agente de AWS DevOps logs	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Registros de Amazon ElastiCache (Redis OSS)	Registros proporcionados	Compatibles [permisos V1]		Compatibles [permisos V1]	
AWS Elastic Beanstalk logs	Registros personalizados	compatible			
Registros de Amazon Elastic Container Service	Registros personalizados	compatible			
Registros del modo automático de Amazon Elastic Kubernetes Service	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Registros del plano de control de Amazon Elastic Kubernetes Service	Registros proporcionados	compatible			

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
AWS Elemental MediaPackage registros de acceso	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
AWS Elemental MediaTailor logs	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
AWS Entity Resolution logs	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Amazon EventBridge Registro de tuberías	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
Amazon EventBridge autobuses de eventos	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
AWS Fargate logs	Registros personalizados	compatible			

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
AWS Fault Injection Service registros de experimentos	Registros proporcionados		Compatibles [permisos V1]		
Amazon FinSpace	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
AWS Global Accelerator registros de flujo	Registros proporcionados		Compatibles [permisos V1]		
AWS Glue registros de trabajos	Registros personalizados	compatible			
registros de errores de IAM Identity Center	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Registros de chat de Amazon Interactive Video Service	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
AWS IoT logs	Registros personalizados	compatible			
AWS IoT FleetWise logs	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
AWS Lambda logs	Registros proporcionados	Soportado	Soportado	compatible	
Registros de Amazon Macie	Registros personalizados	compatible			
Registros de Amazon SES	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
AWS Mainframe Modernization	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Amazon Managed Service para registros de Prometheus	Registros proporcionados	Compatibles [permisos V1]			
Registros de corredores de Amazon MSK	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
Registros de Amazon MSK Connect	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
Registros de Amazon MQ	Registros personalizados	compatible			
AWS Registros de Network Firewall	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
AWS Registros de proxy de Network Firewall	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Registros de acceso a Network Load Balancer	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
OpenSearch logs	Registros personalizados	compatible			
Registros OpenSearch de ingestión de Amazon Service	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
AWS PCS logs	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Registros de conectores de Amazon Q Business	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Registros de conversaciones de Amazon Q Business	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Registros de comentarios y chat rápidos de Amazon	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Registros de PostgreSQL de Amazon Relational Database Service	Registros personalizados	compatible			
AWS Registros de RTB Fabric	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
AWS Security Hub (CSPM)	Registros proporcionados	Compatible [permisos V2]			
AWS Security Hub	Registros proporcionados	Compatible [permisos V2]			
Registros de consultas de DNS públicos de Amazon Route 53	Registros proporcionados	compatible			

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Registros de consultas de resolución de Amazon Route 53	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]		
Eventos de Amazon SageMaker AI	Registros proporcionados	Compatibles [permisos V1]			
Eventos para trabajadores de Amazon SageMaker AI	Registros proporcionados	Compatibles [permisos V1]			
AWS Site-to-Site Registros de VPN	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
Registros de Amazon Simple Email Service	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
Registros de Amazon Simple Notification Service	Registros personalizados	compatible			

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Registros de la política de protección de datos de Amazon Simple Notification Service	Registros personalizados	compatible			
Archivos de fuentes de datos de EC2 Spot Instance	Registros proporcionados		Compatibles [permisos V1]		
AWS Step Functions Registros del flujo de trabajo rápido y del flujo de trabajo estándar	Registros proporcionados	Compatibles [permisos V1]			
Registros de auditoría y registros de estado de Storage Gateway	Registros proporcionados	Compatibles [permisos V1]			
AWS Transfer Family logs	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
Acceso verificado de AWS logs	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	

Fuente de registros	Tipo de registro	the section called “Los registros se envían a CloudWatch Logs”	the section called “Registros enviados a Amazon S3”	the section called “Registros enviados a Firehose”	the section called “Rastros enviados a X-Ray”
Registros de flujo de Amazon Virtual Private Cloud	Registros proporcionados	compatible	Compatibles [permisos V1]	Compatibles [permisos V1]	
Registros de acceso a Amazon VPC Lattice	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	Compatibles [permisos V1]	
Amazon VPC Route Server	Registros proporcionados	Compatible [permisos V2]	Compatible [permisos V2]	Compatible [permisos V2]	
AWS WAF logs	Registros proporcionados	Compatibles [permisos V1]	Compatibles [permisos V1]	compatible	
Amazon WorkMail registros de auditoría	Registros proporcionados	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 permitir que los AWS servicios enumerados en la tabla anterior envíen sus registros a estos destinos, debe iniciar sesión como un usuario que tenga ciertos permisos.

Además, se deben conceder permisos AWS para 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 realizar entregas entre cuentas, debe crear manualmente las políticas de permisos.

Si decide configurar AWS automáticamente los permisos y las políticas de recursos necesarios cuando usted o alguien de su organización configura por primera vez el envío de registros, el usuario que configura 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 los siguientes temas, se proporcionan más detalles para cada uno de estos destinos.

Temas

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 permiso `logs:DescribeLogGroups`, `logs:DescribeResourcePolicies` o `logs:PutResourcePolicy`, asegúrese de configurar el ARN de su línea `Resource` para que utilice un comodín `*`, en lugar de especificar solo un nombre de grupo de registro. 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 de Logs, solo necesitará el `logs:CreateLogDelivery` permiso para configurar el envío de otro de estos tipos de registros a ese mismo grupo de registros.

Política de recursos del grupo de registro

El grupo de registro 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` En el caso de las suscripciones recién creadas, las políticas de recursos se configuran a nivel de grupo de registros y tienen un tamaño máximo de 51 200 bytes. Si una política de recursos a nivel de cuenta existente ya concede permisos mediante caracteres comodín, no se crearía una política independiente a nivel de grupo de registros. Para comprobar la política de Group-level recursos de registro de un grupo de registros específico, utilice el `describe-resource-policies` comando con el `--resource-arn` parámetro establecido en el ARN del grupo de registros y el `--policy-scope` parámetro establecido en. `RESOURCE`

JSON

```
{
    "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-1:111122223333:log-group:my-log-group:log-
stream:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

El límite de la política de recursos del grupo de registros es de 51.200 bytes. Una vez alcanzado este límite, AWS no podrá añadir nuevos permisos. Esto requiere que los clientes modifiquen manualmente la política para conceder al `delivery.logs.amazonaws.com` servicio los permisos principales sobre las `logs:PutLogEvents` acciones `logs:CreateLogStream` y. Los clientes

deben usar un prefijo de nombre de grupo de registros con caracteres comodín como, por ejemplo, `/aws/vendedlogs/*` y usar este nombre de grupo de registros para futuras creaciones de Delivery.

JSON

```
{
    "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-1:111122223333:log-group:my-log-group/aws/
vendedlogs/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

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, al habilitar la ACL en el depósito CloudFront también se concederán permisos a todas las demás fuentes de registro que utilicen este depósito.

Important

Si envía registros a un bucket de Amazon S3 y la política del bucket contiene un elemento `NotAction` o `NotPrincipal`, no podrá añadir automáticamente los permisos de entrega de registros al bucket ni crear una suscripción de registros. Para crear una suscripción de registros correctamente, debe añadir manualmente los permisos de entrega de registros a la política del bucket y, a continuación, crear la suscripción de registros. Para obtener más información, consulte las instrucciones en esta sección.
Si el depósito tiene un cifrado del lado del servidor mediante una AWS KMS clave administrada por el cliente, también debe agregar la política de claves para su clave administrada por el cliente. Para obtener más información, consulte Amazon S3.
Si el depósito de destino tiene SSE-KMS activada una clave de depósito, la política de claves KMS gestionada por el cliente adjunta ya no funciona como se esperaba para todas las solicitudes. Para obtener más información, consulte Reducir el costo de usar SSE-KMS las claves de bucket de Amazon S3.

Si utiliza registros vendidos y cifrado S3 con una AWS KMS clave administrada por el cliente, debe usar un ARN de AWS KMS clave totalmente cualificado en lugar de un ID de clave al configurar el depósito. Para obtener más información, consulte put-bucket-encryption.

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 actualmente no tiene una política de recursos, y el usuario que configura el registro tiene los permisos `S3:GetBucketPolicy` y `S3:PutBucketPolicy` para el bucket, AWS crea automáticamente la siguiente política cuando empiece a enviar los registros a Amazon S3.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
```

```
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-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-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

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 AWS CloudTrail si no se ha concedido el `s3:ListBucket` permiso. `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 del servidor del bucket de Amazon S3

Puede proteger los datos de su bucket de Amazon S3 habilitando el cifrado del lado del servidor con S3-managed claves de Amazon (SSE-S3) o el cifrado del lado del servidor con una AWS KMS clave almacenada en (). AWS Key Management Service SSE-KMS Para obtener más información, consulte Protección de los datos con el cifrado del servidor.

Si lo desea SSE-S3, no se requiere ninguna configuración adicional. Amazon S3 se encarga de la clave de cifrado.

> **Warning**
>
> Si lo desea 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. 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 lo desea 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 se envían los tipos de registros enumerados en la tabla de la sección anterior a Firehose:

Permisos de usuario

Para poder configurar el envío de cualquiera de estos tipos de registros a Firehose por primera vez, debe iniciar sesión en una cuenta con los siguientes permisos.

- `logs:CreateLogDelivery`
- `firehose:TagDeliveryStream`
- `iam:CreateServiceLinkedRole`

Si alguno de estos tipos de registros ya se envió a Firehose, entonces para configurar el envío de otro de estos tipos de registros a Firehose solo necesita tener los permisos `logs:CreateLogDelivery` y `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.

JSON

```
{
    "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:

JSON

```
{
  "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 y X-Ray para la entrega de rastreo.

La entrega de un registro funcional consta de tres elementos:

- Una `DeliverySource`, que es un objeto lógico que representa el recurso que realmente envía los registros.

- Un `DeliveryDestination`, que es un objeto lógico que representa el destino de entrega real.
- Un `Delivery`, que conecta un origen de entrega con el destino de la 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 el origen de entrega de la cuenta A hasta el destino de la entrega en la cuenta B. En el caso de las entregas entre cuentas, debe crear manualmente las políticas de permisos.
- 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 que genera los registros. Para ello, siga estos pasos.
>
> 1. Elimine el `Delivery` mediante la DeleteDeliveryoperación.
> 2. Elimine el `DeliverySource` mediante la DeleteDeliverySourceoperación.
> 3. Si lo `DeliveryDestination` asociado al `DeliverySource` que acaba de eliminar se usa solo para esto específico`DeliverySource`, puede eliminarlo mediante la DeleteDeliveryDestinationsoperación.

Contenido

- Registros enviados a Firehose
- Rastros enviados a X-Ray

Registros enviados a CloudWatch Logs

Permisos de usuario

Para habilitar el envío de CloudWatch registros a Logs, debe iniciar sesión con los siguientes permisos.

JSON

```
{
    "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",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
            "arn:aws:logs:us-east-1:444455556666:delivery-source:*",
            "arn:aws:logs:us-east-1:777788889999:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
```

```
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyCWL",
            "Effect": "Allow",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:123456789012:*"
            ]
        }
    ]
}
```

Política de recursos del grupo de registro

El grupo de registro 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` En el caso de las suscripciones recién creadas, las políticas de recursos se configuran a nivel de grupo de registros y tienen un tamaño máximo de 51 200 bytes. Si una política de recursos a nivel de cuenta existente ya concede permisos mediante caracteres comodín, no se crearía una política independiente a nivel de grupo de registros. Para comprobar la política de Group-level recursos de registro de un grupo de registros específico, utilice el `describe-resource-policies` comando con el `--resource-arn` parámetro establecido en el ARN del grupo de registros y el `--policy-scope` parámetro establecido en. `RESOURCE`

JSON

```
{
    "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-1:111122223333:log-group:my-log-group:log-
stream:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

El límite de la política de recursos del grupo de registros es de 51.200 bytes. Una vez alcanzado este límite, AWS no podrá añadir nuevos permisos. Esto requiere que los clientes modifiquen manualmente la política para conceder al `delivery.logs.amazonaws.com` servicio los permisos principales sobre las `logs:PutLogEvents` acciones `logs:CreateLogStream` y. Los clientes

deben usar un prefijo de nombre de grupo de registros con caracteres comodín como, por ejemplo, `/aws/vendedlogs/*` y usar este nombre de grupo de registros para futuras creaciones de Delivery.

JSON

```
{
    "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-1:111122223333:log-group:my-log-group/aws/
vendedlogs/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "0123456789"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:111122223333:*"
                    ]
                }
            }
        }
    ]
}
```

Registros enviados a Amazon S3

Permisos de usuario

Para habilitar el envío de registros a Amazon S3, debe iniciar sesión con los siguientes permisos.

JSON

```
{
    "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",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-source:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
```

```
            ],
            "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 actualmente no tiene una política de recursos, y el usuario que configura el registro tiene los permisos `S3:GetBucketPolicy` y `S3:PutBucketPolicy` para el bucket, AWS crea automáticamente la siguiente política cuando empiece a enviar los registros a Amazon S3.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-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-1:111122223333: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 AWS CloudTrail 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 del servidor del bucket de Amazon S3

Puede proteger los datos de su bucket de Amazon S3 habilitando el cifrado del lado del servidor con S3-managed claves de Amazon (SSE-S3) o el cifrado del lado del servidor con una AWS KMS clave almacenada en (). AWS Key Management Service SSE-KMS Para obtener más información, consulte Protección de los datos con el cifrado del servidor.

Si lo desea SSE-S3, no se requiere ninguna configuración adicional. Amazon S3 se encarga de la clave de cifrado.

> **Warning**
>
> Si lo desea 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. 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 lo desea 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, debe iniciar sesión con los siguientes permisos.

JSON

```
{
    "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",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
```

```
    "arn:aws:logs:us-east-1:111122223333:delivery-source:*",
    "arn:aws:logs:us-east-1:111122223333:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyFH",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/*"
            ]
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::111122223333: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.

JSON

```
{
    "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:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
```

```
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Rastros enviados a X-Ray

Permisos de usuario

Para habilitar el envío de rastreos a AWS X-Ray, debe iniciar sesión con los siguientes permisos.

JSON

```
{
    "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",
                "logs:UpdateDeliveryConfiguration"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:111122223333:delivery:*",
            "arn:aws:logs:us-east-1:111122223333:delivery-source:*",
            "arn:aws:logs:us-east-1:111122223333:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
```

```
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries",
                "logs:DescribeConfigurationTemplates"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyXRay",
            "Effect": "Allow",
            "Action": [
                "xray:PutResourcePolicy",
                "xray:ListResourcePolicies",
                "xray:GetTraceSegmentDestination"
            ],
            "Resource": "*"
        }
    ]
}
```

X-Ray política de recursos

La cuenta de destino a la que se envían los registros debe tener una política de recursos que incluya determinados permisos. Cuando el usuario que configura el rastreo tiene `xray:PutResourcePolicy` `xray:ListResourcePolicies` permisos en la cuenta, crea AWS automáticamente la política de recursos al empezar a enviar seguimientos a X-Ray. La política que se crea depende del servicio de origen:

Amazon Bedrock AgentCore resources

AWS crea una política de recursos por tipo de recurso. La política utiliza patrones comodín que se limitan a los límites de la cuenta y abarcan todos los recursos del mismo tipo de Amazon Bedrock AgentCore recurso de la cuenta. Por ejemplo, si un recurso de Amazon Bedrock AgentCorememoria está habilitado para la entrega de trazas, la política cubre todos los recursos de memoria de esa cuenta, incluidos los recursos de memoria que se creen en el futuro.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "xray:PutTraceSegments",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ForAllValues:ArnLike": {
          "logs:LogGeneratingResourceArns": "arn:aws:bedrock-agentcore:us-
east-1:123456789012:memory/*"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:delivery-
source:*"
        }
      }
    }
  ]
}
```

Otros AWS servicios

Para otros servicios que admiten el envío de rastreos, AWS crea una política de recursos que se limita al recurso de origen específico.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "xray:PutTraceSegments",
```

```
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ForAllValues:ArnLike": {
          "logs:LogGeneratingResourceArns": "arn:aws:bedrock:us-
east-1:123456789012:knowledge-base/KnowledgeBaseId"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:delivery-
source:xray-test"
        }
      }
    }
  ]
}
```

Habilita la búsqueda de transacciones

Para habilitar el envío de seguimientos a X-Ray, debes habilitar la búsqueda de transacciones.

Service-specific permisos

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 acción `AllowVendedLogDeliveryForResource` en los recursos que ofrecen registros dentro de ese servicio. Para estos servicios, utilice la siguiente política y sustituya *`service`* y por *`resource-type`* los valores correspondientes. Para ver los valores específicos de cada servicio en estos campos, consulte la página de documentación de esos servicios para ver los registros que se ofrecen. En el siguiente ejemplo, la política se ha actualizado para permitir los registros ofrecidos de Amazon SES.

JSON

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

```
            "Effect": "Allow",
            "Action": [
                "ses:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:ses:us-east-1:123456789012:resource-type/*"
        }
    ]
}
```

Console-specific permisos

Además de los permisos que se enumeran en las secciones anteriores, si va a configurar la entrega de registros mediante la consola en lugar de las API, también necesitará los siguientes permisos adicionales:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogDeliveryActionsConsoleCWL",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:CreateLogGroup"
            ],
            "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": [
                "*"
            ]
        }
    ]
}
```

Cross-account ejemplo de entrega

En este ejemplo, se usan dos cuentas. La cuenta con el recurso generador de registros es la cuenta A, ID:*123456789012*, y la cuenta con el recurso que consume registros es la cuenta B, ID:. *111122223333*

La cuenta A quiere entregar registros de la base de Amazon Bedrock conocimientos de su cuenta con el ARN arn:aws:bedrock: ::knowledge-base/. *us-east-1 123456789012 kb-12345678*

En este ejemplo, la cuenta A necesita estos permisos:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowVendedLogDeliveryForKnowledgeBase",
            "Effect": "Allow",
            "Action": [
                "bedrock:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:bedrock:us-east-1:123456789012:knowledge-
base/XXXXXXXXXX"
        },
```

```
        {
            "Sid": "CreateLogDeliveryPermissions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliverySource",
                "logs:CreateDelivery"
            ],
            "Resource": [
            "arn:aws:logs:us-east-1:123456789012:delivery-source:*",
    "arn:aws:logs:us-east-1:123456789012:delivery:*",
    "arn:aws:logs:us-east-1:444455556666:delivery-destination:*"
            ]
        }
    ]
}
```

Creación de origen de entrega

Para empezar, la cuenta A crea un origen de entrega con su base de conocimientos básica:

```
aws logs put-delivery-source --name my-delivery-source --log-type APPLICATION_LOGS --
resource-arn arn:aws:bedrock:region:AAAAAAAAAAAA:knowledge-base/XXXXXXXXXX
```

A continuación, la cuenta B debe crear el destino de entrega mediante uno de los siguientes flujos:

- Configuración de la entrega a un bucket de Amazon S3
- Configuración de la entrega a un flujo de Firehose

Configuración de la entrega a un bucket de Amazon S3

La cuenta B desea recibir los registros en el bucket de S3 con el ARN arn:aws:s3:::amzn-s3-demo-bucket. En este ejemplo, la cuenta B necesitará los siguientes permisos:

JSON

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

```
            "Sid": "PutLogDestinationPermissions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliveryDestination",
                "logs:PutDeliveryDestinationPolicy"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:*"
        }
    ]
}
```

El bucket necesitará los siguientes permisos en su política de bucket:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogsDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/AWSLogs/123456789012/
*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                    "123456789012"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:delivery-source:my-
delivery-source"
                    ]
```

```
                }
            }
        }
    ]
}
```

Si el depósito está cifrado con, asegúrese de que la política de claves tenga los permisos adecuados. SSE-KMS AWS KMS Por ejemplo, si la clave de KMS es `arn:aws:kms:us-east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`, utilice lo siguiente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogsGenerateDataKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                    "123456789012"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:delivery-source:my-
delivery-source"
                    ]
                }
            }
        }
    ]
```

```
}
```

A continuación, la cuenta B puede crear un destino de entrega con el bucket de S3 como recurso de destino:

```
aws logs put-delivery-destination --name my-s3-delivery-destination --delivery-
destination-configuration "destinationResourceArn=arn:aws:s3:::amzn-s3-demo-bucket"
```

Luego, la cuenta B crea una política de destino de entrega en el destino de entrega recién creado, lo que permitirá a la cuenta A crear un registro de entrega. La política que se añadirá al destino de entrega recién creado es la siguiente:

JSON

```
    {
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {
                "AWS": "123456789012"
            },
            "Action": [
                "logs:CreateDelivery"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:amzn-s3-demo-bucket"
        }
    ]
}
```

Esta política se guardará en el ordenador de la cuenta B como `destination-policy-s3.json`. Para adjuntar este recurso, la cuenta B ejecutará el siguiente comando:

```
aws logs put-delivery-destination-policy --delivery-destination-name my-s3-delivery-
destination --delivery-destination-policy file://destination-policy-s3.json
```

Por último, la cuenta A crea la entrega, que vincula el origen de entrega de la cuenta A con el destino de la entrega de la cuenta B.

```
aws logs create-delivery --delivery-source-name my-delivery-source --delivery-
destination-arn arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:my-s3-delivery-
destination
```

Configuración de la entrega a un flujo de Firehose

En este ejemplo, la cuenta B quiere recibir registros en su flujo de Firehose. La transmisión Firehose tiene el siguiente ARN y está configurada para usar el tipo de transmisión de DirectPut entrega:

`arn:aws:firehose:us-east-1:111122223333:deliverystream/log-delivery-stream`

En este ejemplo, la cuenta B necesita estos permisos:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFirehoseCreateSLR",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        },
        {
            "Sid": "AllowFirehoseTagging",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": "arn:aws:firehose:us-
east-1:111122223333:deliverystream/X"
        },
        {
            "Sid": "AllowFirehoseDeliveryDestination",
            "Effect": "Allow",
```

```
            "Action": [
                "logs:PutDeliveryDestination",
                "logs:PutDeliveryDestinationPolicy"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:*"
        }
    ]
}
```

El flujo de Firehose debe tener la etiqueta `LogDeliveryEnabled` configurada en `true`.

A continuación, la cuenta B crea un destino de entrega con el flujo de Firehose como recurso de destino:

```
aws logs put-delivery-destination --name my-fh-delivery-destination --delivery-
destination-configuration
 "destinationResourceArn=arn:aws:firehose:region:BBBBBBBBBBBB:deliverystream/X"
```

Luego, la cuenta B crea una política de destino de entrega en el destino de entrega recién creado, lo que permitirá a la cuenta A crear un registro de entrega. La política que se añadirá al destino de entrega recién creado es la siguiente:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDelivery",
            "Effect": "Allow",
            "Principal": {
            "AWS": "123456789012"
            },
            "Action": [
                "logs:CreateDelivery"
            ],
            "Resource": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:amzn-s3-demo-bucket"
        }
    ]
```

```
}
```

Esta política se guardará en el ordenador de la cuenta B como `destination-policy-fh.json`. Para adjuntar este recurso, la cuenta B ejecuta el siguiente comando:

```
aws logs put-delivery-destination-policy --delivery-destination-name my-fh-delivery-
destination --delivery-destination-policy file://destination-policy-fh.json
```

Por último, la cuenta A crea la entrega, que vincula el origen de entrega de la cuenta A con el destino de la entrega de la cuenta B.

```
aws logs create-delivery --delivery-source-name my-delivery-source --delivery-
destination-arn arn:aws:logs:region:BBBBBBBBBBBB:delivery-destination:my-fh-delivery-
destination
```

Cross-service prevención confusa de diputados

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. Cross-service la suplantación de identidad puede producirse cuando un servicio (el servicio de llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger sus datos en 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 contexto de condición global en las políticas de recursos para limitar los permisos que CloudWatch Logs otorga 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. Se usa `aws:SourceOrgPaths` para asociar cualquier recurso de las cuentas dentro de una AWS Organizations ruta con el 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 se puede utilizar la clave contextual de la condición global para ayudar a evitar `aws:SourceArn` el confuso problema de los diputados.

Automatice la activación de los registros con las reglas de activación de la telemetría

Puede usar las reglas de activación de la telemetría para configurar automáticamente la recopilación de registros para sus recursos. AWS Las reglas le ayudan a estandarizar la recopilación de registros en toda su organización o sus cuentas y a garantizar una cobertura de supervisión uniforme.

Las reglas de habilitación le permiten:

- Habilite automáticamente el registro de los recursos nuevos y existentes que coincidan con el ámbito de su regla
- Aplica las reglas a nivel de organización, unidad organizativa (OU) o cuenta individual
- Filtre los recursos a los que afecta una regla mediante etiquetas

Actualmente, las reglas de habilitación admiten las siguientes AWS fuentes de telemetría: Amazon VPC Flow Logs, Route 53 Resolver Query AWS WAF Logs, NLB Access Logs, Amazon EKS Control

Plane Logs, Data CloudTrail and Management Events, Amazon Bedrock Logs, Amazon EC2 Detailed Metrics, Security AWS Hub, AgentCore Amazon Bedrock Gateway, Amazon Bedrock AgentCore Memory y Distribution. AgentCore CloudFront

Para obtener detalles completos sobre las reglas de habilitación, incluidos el comportamiento de las reglas, la gestión de las reglas, la solución de problemas y las consideraciones específicas del servicio, consulte las reglas de habilitación de telemetría en la Guía del usuario de Amazon. CloudWatch

Jerarquía de evaluación de reglas

Las reglas de habilitación se evalúan jerárquicamente: primero las reglas organizativas, luego las reglas y, por último, las reglas a nivel de cuenta. OU-level Las reglas de los niveles superiores proporcionan la telemetría de referencia. Lower-level las reglas pueden añadir telemetría adicional, pero no reducirla. Si existen reglas contradictorias en el mismo ámbito, no se aplicará ninguna hasta que se resuelva el conflicto.

Crear una regla de habilitación

Creación de una regla de activación de la telemetría

1. Abra la CloudWatch consola en. https://console.aws.amazon.com/cloudwatch/
2. En el panel de navegación, elija Configuración de la telemetría.
3. Elija la pestaña Reglas de activación.
4. Seleccione Agregar regla.
5. Especifique el nombre de la regla, el ámbito (organización, unidad organizativa o cuenta), la fuente de datos y el tipo de telemetría.
6. Si lo desea, añada etiquetas para filtrar los recursos a los que afecta la regla.
7. Seleccione Crear regla.

Habilitar el registro desde fuentes de terceros

CloudWatch Logs admite la ingesta de registros de fuentes de terceros mediante integraciones directas de API e integraciones de buckets de Amazon S3. También puede recibir información de seguridad adicional de terceros a través de AWS Security Hub (CSPM).

Integraciones directas de terceros

CloudWatch Logs proporciona integraciones directas con las siguientes fuentes de terceros. Estas integraciones utilizan conexiones API directas o integraciones de bucket de Amazon S3 para incorporar los registros en Logs: CloudWatch

- CrowdStrike Falcon
- Microsoft Office 365
- Okta Auth0
- ID de Microsoft Entra
- NGFW de Palo Alto Networks
- Eventos de Microsoft Windows
- Wiz
- Zscaler Internet Access
- Okta SSO
- SentinelOne Seguridad de terminales
- GitHub Registros de auditoría
- ServiceNow CMDB
- Cisco Umbrella

Para obtener instrucciones de configuración, consulta la guía de configuración de la integración de terceros en la Guía del CloudWatch usuario de Amazon.

Fuentes adicionales a través de AWS Security Hub (CSPM)

Además de las integraciones directas, los socios de seguridad externos envían los resultados al AWS Security Hub CSPM, que luego están disponibles como fuentes de datos en los registros.

CloudWatch En la siguiente tabla se enumeran las integraciones de los socios CSPM de Security Hub y su tipo de integración.

Para habilitar los hallazgos de CSPM de Security Hub como fuente de datos en los CloudWatch registros, cree una regla de activación de telemetría para Security AWS Hub en la consola. CloudWatch La regla de activación se configura CloudWatch para incorporar automáticamente las conclusiones del CSPM de Security Hub a un grupo de registros gestionado. Para step-by-step obtener instrucciones, consulte las reglas de activación de la telemetría en la Guía del usuario de Amazon. CloudWatch

Partner	Integración
3 — NTA CORESec	Envía los resultados a través de Security Hub (CSPM)
Lógica de alertas: SIEMless gestión de amenazas	Envía los resultados a través de Security Hub (CSPM)
Aqua Security: plataforma de seguridad nativa de la nube	Envía los resultados a través de Security Hub (CSPM)
Aqua Security — Kube-bench	Envía los resultados a través de Security Hub (CSPM)
Armadura: armadura en cualquier lugar	Envía los resultados a través de Security Hub (CSPM)
AttackIQ	Envía los resultados a través de Security Hub (CSPM)
Barracuda Networks: guardián de la seguridad en la nube	Envía los resultados a través de Security Hub (CSPM)
BigID — BigID Enterprise	Envía los resultados a través de Security Hub (CSPM)
Hexágono azul	Envía los resultados a través de Security Hub (CSPM)

Partner	Integración
Punto de control: CloudGuard IaaS	Envía los resultados a través de Security Hub (CSPM)
Check Point: CloudGuard gestión de la postura	Envía los resultados a través de Security Hub (CSPM)
Claridad — Domo	Envía los resultados a través de Security Hub (CSPM)
Seguridad del almacenamiento en la nube: antivirus para Amazon S3	Envía los resultados a través de Security Hub (CSPM)
Contrast Security: Contrast Assess	Envía los resultados a través de Security Hub (CSPM)
CrowdStrike — Halcón CrowdStrike	Envía los resultados a través de Security Hub (CSPM)
CyberArk — Análisis de amenazas privilegiado	Envía los resultados a través de Security Hub (CSPM)
Teorema de los datos	Envía los resultados a través de Security Hub (CSPM)
Datos	Envía los resultados a través de Security Hub (CSPM)
Forcepoint — CASB	Envía los resultados a través de Security Hub (CSPM)
Forcepoint: puerta de enlace de seguridad en la nube	Envía los resultados a través de Security Hub (CSPM)
Forcepoint — DLP	Envía los resultados a través de Security Hub (CSPM)
Forcepoint — NGFW	Envía los resultados a través de Security Hub (CSPM)

Partner	Integración
Fuga	Envía los resultados a través de Security Hub (CSPM)
Guardicore — Centra	Envía los resultados a través de Security Hub (CSPM)
HackerOne — Inteligencia sobre vulnerabilidades	Envía los resultados a través de Security Hub (CSPM)
JFrog — Radiografía	Envía los resultados a través de Security Hub (CSPM)
Juniper Networks: firewall vSRX de próxima generación	Envía los resultados a través de Security Hub (CSPM)
k9 Security: analizador de acceso	Envía los resultados a través de Security Hub (CSPM)
Encajes	Envía los resultados a través de Security Hub (CSPM)
McAfee — MVISION CHNAPP	Envía los resultados a través de Security Hub (CSPM)
NETSCOUT: investigador cibernético	Envía los resultados a través de Security Hub (CSPM)
Orca: plataforma de seguridad en la nube	Envía los resultados a través de Security Hub (CSPM)
Palo Alto Networks: Prisma Cloud Compute	Envía los resultados a través de Security Hub (CSPM)
Palo Alto Networks: Prisma Cloud Enterprise	Envía los resultados a través de Security Hub (CSPM)
Plerion: plataforma de seguridad en la nube	Envía los resultados a través de Security Hub (CSPM)

Partner	Integración
Merodeador	Envía los resultados a través de Security Hub (CSPM)
Qualys: gestión de vulnerabilidades	Envía los resultados a través de Security Hub (CSPM)
Rapid7 — InsightVM	Envía los resultados a través de Security Hub (CSPM)
SentinelOne	Envía los resultados a través de Security Hub (CSPM)
Snyk	Envía los resultados a través de Security Hub (CSPM)
Sonrai Security — Sonrai Dig	Envía los resultados a través de Security Hub (CSPM)
Sophos: protección de servidores	Envía los resultados a través de Security Hub (CSPM)
StackRox — Seguridad de Kubernetes	Envía los resultados a través de Security Hub (CSPM)
Sumo Logic: análisis de datos de máquinas	Envía los resultados a través de Security Hub (CSPM)
Symantec: protección de la carga de trabajo en la nube	Envía los resultados a través de Security Hub (CSPM)
Tenable.io	Envía los resultados a través de Security Hub (CSPM)
Trend Micro – Cloud One	Envía los resultados a través de Security Hub (CSPM)
Vectra — Cognito Detect	Envía los resultados a través de Security Hub (CSPM)

Partner	Integración
Wiz	Envía los resultados a través de Security Hub (CSPM)
Caveonix — Caveonix Cloud	Envía y recibe los resultados a través de Security Hub (CSPM)
Cloud Custodian	Envía y recibe los resultados a través de Security Hub (CSPM)
DisruptOps	Envía y recibe los resultados a través de Security Hub (CSPM)
Kion	Envía y recibe los resultados a través de Security Hub (CSPM)
Turbot	Envía y recibe los resultados a través de Security Hub (CSPM)

Note

Esta lista refleja las integraciones de los socios de Security Hub que envían las conclusiones en el momento de redactar este artículo. Dado que AWS Security Hub añade nuevas integraciones de socios con regularidad, consulte Integraciones de productos de terceros con Security Hub CSPM en la Guía del usuario de AWS Security Hub para obtener la up-to-date lista más amplia de socios disponibles.

Exportación de datos de registro a Simple Storage Service (Amazon S3)

Este capítulo proporciona información para que pueda exportar los datos de registro desde sus grupos de registro a un bucket de Amazon S3 y utilizarlos en el procesamiento y análisis personalizados o para cargarlos en otros sistemas. Puede exportar a un bucket de S3 de la misma cuenta o de una cuenta diferente.

Se 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

Le recomendamos que no exporte regularmente a Amazon S3 para archivar sus registros de forma continua. Para ese caso de uso, se recomienda el uso de suscripciones. Para obtener más información sobre las suscripciones, consulte Procesamiento en tiempo real de datos de registros con suscripciones.

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 organizar los datos exportados, especifique un prefijo para cada tarea de exportación que se utilizará como prefijo clave de Amazon S3 para todos los objetos exportados. Por ejemplo, `prod/app-logs/2026-01-03/` o `log-group-name/backup/`

> **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 en el grupo de registro que se incorporaron durante o después de este momento.

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`amzn-s3-demo-bucket`.

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 desde la misma cuenta (consola)

Si el bucket de Amazon S3 está en la misma cuenta que los registros que se exportan, siga las instrucciones de esta sección.

Temas

Cómo crear un bucket de Amazon S3 (consola)

Te recomendamos que utilices un depósito creado específicamente para CloudWatch los registros. Sin embargo, si desea utilizar un bucket existente, puede pasar al paso 2.

> **Note**
>
> El bucket de Amazon S3 debe residir en la misma región que los datos de registro para exportar. CloudWatch Logs no admite la exportación de datos a buckets de Amazon S3 de una región diferente.

Creación de un bucket de Amazon 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, elija la región en la que residen sus 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.

Configure los permisos de acceso (consola)

Para crear la tarea de exportación, tendrás que iniciar sesión con el rol de `AmazonS3ReadOnlyAccess` IAM y tener 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 gestionados 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 agregue 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.

Establecer permisos en un bucket de Amazon S3 (consola)

De forma predeterminada, todos los buckets y objetos de Amazon S3 son privados. Solo el propietario del recurso, el Cuenta de AWS que creó el depósito, puede acceder al depósito y a cualquier objeto que contenga. 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 Amazon 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 cuentas IDs de la `aws:SourceAccount` clave serían las cuentas desde las que un usuario puede exportar los datos de registro a su bucket de Amazon 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 ha creado.
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 `amzn-s3-demo-bucket` 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.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
          "Sid": "AllowCloudWatchLogsGetBucketAcl",
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
```

```
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                      "123456789012",
                      "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:us-east-1:123456789012:log-group:*",
                          "arn:aws:logs:us-east-1:111122223333:log-group:*"
                      ]
              }
            }
        },
        {
            "Sid": "AllowCloudWatchLogsPutObject",
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "123456789012",
                      "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:us-east-1:123456789012:log-group:*",
                          "arn:aws:logs:us-east-1:111122223333: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 los datos de registro a su bucket de Amazon 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) Exportación a un bucket de Amazon S3 de destino cifrado con SSE-KMS (consola)

Este paso solo es necesario si va a exportar a un bucket de Amazon S3 que utilice 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 /kms. https://console.aws.amazon.com
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 Siguiente.
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, *Region* sustitúyalo por la región de sus registros y *account-ARN* sustitúyalo por el ARN de la cuenta propietaria de la clave KMS.

    JSON

```
{
    "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. Seleccione Save changes (Guardar cambios).
15. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
16. Encuentre el bucket que creó en Crear un bucket de S3 (CLI) 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 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. Seleccione Save changes (Guardar cambios).

Crea una tarea de exportación (consola)

En este procedimiento, se crea la tarea de exportación para exportar los registros de 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 Configure los permisos de acceso (consola).
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 registro) elija el nombre del grupo de registro.
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 registro 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, 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 multicuenta (consola)

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

Crear un bucket de Amazon S3 para la exportación entre cuentas (consola)

Le recomendamos que utilice un bucket creado específicamente para CloudWatch Logs. Sin embargo, si desea utilizar un depósito existente, puede omitir este procedimiento.

> **Note**
>
> El bucket de Amazon S3 debe residir en la misma región que los datos de registro para exportar. CloudWatch Logs no admite la exportación de datos a buckets de Amazon S3 de una región diferente.

Creación de un bucket de Amazon 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, elija la región en la que residen sus 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.

Configura los permisos de acceso para la exportación entre cuentas (consola)

En primer lugar, debe crear una nueva política de IAM para permitir que CloudWatch Logs incluya la `s3:PutObject` acción del bucket Amazon S3 de destino en la cuenta de destino.

Además de la `s3:PutObject` acción, las acciones adicionales incluidas en la política dependen de si el bucket de destino utiliza el AWS KMS cifrado o si lo ha ACLs habilitado mediante la configuración de propiedad de objetos de S3.

- Si utiliza el cifrado de KMS, añada las acciones `kms:GenerateDataKey` y `kms:Decrypt` para el recurso clave
- Si ACLs están habilitadas en el bucket, añada la `s3:PutObjectAcl` acción para el recurso del bucket

Cambie `amzn-s3-demo-bucket` el nombre del depósito de S3 de destino en las siguientes políticas.

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 utiliza AWS KMS cifrado, pegue la siguiente política en el editor.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

   Si el depósito de destino utiliza AWS KMS cifrado, pegue la siguiente política en el editor.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
```

```
  ]
}
```

Si ACLs están habilitadas en el bucket de destino, añada el bloque s3: PutObjectAcl al s3: PutObject Action en las políticas anteriores.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
               "s3:PutObject",
               "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

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 una tarea de exportación, debe iniciar sesión con un rol de IAM que tenga la política `AmazonS3ReadOnlyAccess` administrada asociada, la política de IAM creada anteriormente 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 gestionados 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 agregue 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.

Configure los permisos en un bucket de S3 para la exportación entre cuentas (consola)

De forma predeterminada, los buckets y los objetos de S3 son privados. Solo el propietario del recurso, el Cuenta de AWS que creó el depósito, puede acceder al depósito y a cualquier objeto que contenga. 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 cuentas IDs 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 ha creado.
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 `amzn-s3-demo-bucket` por el nombre de su bucket de S3. Asegúrese de especificar el punto de conexión de la región correcta, como `us-east-1`, en Entidad principal.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "123456789012",
                    "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:123456789012:log-group:*",
                        "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
            }
          }
      },
```

```
        {
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "123456789012",
                      "111122223333"
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:us-east-1:123456789012:log-group:*",
                          "arn:aws:logs:us-east-1:111122223333:log-group:*"
                      ]
              }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
              "AWS": "arn:aws:iam::111122223333:role/role_name"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "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) Exportación a un bucket de Amazon S3 de destino cifrado con SSE-KMS para la exportación entre cuentas (consola)

Este procedimiento 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 /kms. https://console.aws.amazon.com
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 Siguiente.
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, *us-east-1* sustitúyalo por la región de sus registros, *account-ARN* por el ARN de la cuenta propietaria de la clave de KMS, *123456789012* por el número de cuenta que posee la clave de KMS, *key_id* por el ID de la clave de kms-key y por el rol utilizado para *role_name* crear la tarea de exportación.

    JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
            "Service": "logs.us-east-1.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::111122223333:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:123456789012:key/key-id"
        }
    ]
}
```

14. Seleccione Save changes (Guardar cambios).
15. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
16. Encuentre el bucket que creó en Crear un bucket de S3 (CLI) 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 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. Selecciona Elige una de tus AWS KMS claves y busca la clave que creaste.
21. En Bucket key (Clave de bucket), seleccione Enable (Habilitar).
22. Seleccione Save changes (Guardar cambios).

Crea una tarea de exportación para exportarla entre cuentas (consola)

En este procedimiento, creará la tarea de exportación para exportar los registros de 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 Configure los permisos de acceso (consola).
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 registro) elija el nombre del grupo de registro.
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 registro 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`amzn-s3-demo-bucket`. 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 desde la misma cuenta (CLI)
- Exportación multicuenta (CLI)

Exportación desde la misma cuenta (CLI)

Si el bucket de Amazon S3 está en la misma cuenta que los registros que se exportan, siga las instrucciones de esta sección.

Temas

Crear un bucket de S3 (CLI)

Le recomendamos que utilice un depósito que se haya creado específicamente para CloudWatch los registros. Sin embargo, si desea utilizar un depósito existente, puede omitir este procedimiento.

> **Note**
>
> El bucket 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 amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-2
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "Location": "/amzn-s3-demo-bucket"
```

```
}
```

Configurar permisos de acceso (CLI)

Para crear la tarea de exportación más adelante, tendrás que iniciar sesión con el rol de `AmazonS3ReadOnlyAccess` IAM 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 gestionados 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 agregue 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.

Establecer permisos en un bucket de S3 (CLI)

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 cuentas IDs 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 `amzn-s3-demo-bucket` 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-east-1`. Utilice un editor de texto para crear este archivo de política. No utilice la consola de IAM.

   JSON

```
{
"Version":"2012-10-17",
"Statement": [
  {
      "Sid": "AllowGetBucketAcl",
      "Action": "s3:GetBucketAcl",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
      "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
      "Condition": {
        "StringEquals": {
            "aws:SourceAccount": [
                "123456789012",
                "111122223333"
            ]
        },
        "ArnLike": {
```

```
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:123456789012:log-group:*",
                        "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowPutObject",
            "Action": "s3:PutObject",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                        "123456789012",
                        "111122223333"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:123456789012:log-group:*",
                        "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
                }
            }
        }
    ]
}
```

2. Configura la política que acabas de añadir como política de acceso a tu 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 amzn-s3-demo-bucket --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) Exportación a un bucket de Amazon S3 de destino cifrado con SSE-KMS (CLI)

Este procedimiento 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:

   - *`Region`*Sustitúyalo por la región de sus registros.
   - *`account-ARN`*Sustitúyalo por el ARN de la cuenta propietaria de la clave KMS.

   JSON

```
{
    "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. Introduzca 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-ARN: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. Introduzca el siguiente comando y *amzn-s3-demo-bucket* sustitúyalo por el nombre del depósito al que va a exportar los registros.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-
encryption-configuration file://bucketencryption.json
```

   Si el comando no devuelve ningún error, el proceso se ha realizado correctamente.

Crear una tarea de exportación (CLI)

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 Configurar permisos de acceso (CLI).
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 "amzn-s3-demo-bucket" --destination-prefix "export-task-output"
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Exportación multicuenta (CLI)

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

- Cree un bucket de S3 para la exportación entre cuentas (CLI)
- Configurar los permisos de acceso para la exportación multicuenta (CLI)
- Establezca permisos en un bucket de S3 para la exportación entre cuentas (CLI)
- (Opcional) Exportación a un bucket de Amazon S3 de destino cifrado con SSE-KMS para la exportación entre cuentas (CLI)
- Crear una tarea de exportación para la exportación multicuenta (CLI)

Cree un bucket de S3 para la exportación entre cuentas (CLI)

Le recomendamos que utilice un depósito que se haya creado específicamente para CloudWatch los registros. Sin embargo, si desea utilizar un bucket existente, puede pasar al paso 2.

> **Note**
>
> El bucket de S3 debe residir en la misma región que los datos de registro que se van a exportar. CloudWatch Logs no admite 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 amzn-s3-demo-bucket --create-bucket-configuration
 LocationConstraint=us-east-2
```

A continuación, se muestra un ejemplo del resultado.

```
{
    "Location": "/amzn-s3-demo-bucket"
}
```

Configurar los permisos de acceso para la exportación multicuenta (CLI)

En primer lugar, debe crear una nueva política de IAM para permitir que CloudWatch Logs incluya la `s3:PutObject` acción del bucket Amazon S3 de destino en la cuenta de destino.

Además de la `s3:PutObject` acción, las acciones adicionales incluidas en la política dependen de si el bucket de destino utiliza el AWS KMS cifrado o si lo ha ACLs habilitado mediante la configuración de propiedad de objetos de S3.

- Si utiliza el cifrado de KMS, añada las acciones `kms:GenerateDataKey` y `kms:Decrypt` para el recurso clave
- Si ACLs están habilitadas en el bucket, añada la `s3:PutObjectAcl` acción para el recurso del bucket

Cambie `amzn-s3-demo-bucket` el nombre del depósito de S3 de destino en las siguientes políticas.

La política que cree depende de si el bucket de destino utiliza el cifrado de AWS KMS . Si no utiliza el AWS KMS cifrado, cree una política con el siguiente contenido.

JSON

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

```
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Si el depósito de destino usa AWS KMS cifrado, cree una política con el siguiente contenido.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Si ACLs están habilitados en el bucket de destino, añada el bloque s3: PutObjectAcl al s3: PutObject Action en las políticas anteriores.

JSON

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

```
            "Effect": "Allow",
            "Action": [
               "s3:PutObject",
               "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Para crear una tarea de exportación, debe iniciar sesión con un rol de IAM que tenga la política `AmazonS3ReadOnlyAccess` administrada asociada, la política de IAM creada anteriormente 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 gestionados 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 agregue 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.

Establezca permisos en un bucket de S3 para la exportación entre cuentas (CLI)

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 cuentas IDs 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 con un nombre `policy.json` y añada la siguiente política de acceso, `amzn-s3-demo-bucket` cambiándola por el nombre del bucket de S3 de destino, `Principal` al punto final de la región a la que va a exportar los datos de registro, por ejemplo`us-west-1`. Utilice un editor de texto para crear este archivo de política. No utilice la consola de IAM.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
          "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
          "Condition": {
```

```
            "StringEquals": {
                "aws:SourceAccount": [
                "123456789012",
                "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:log-group:*",
                    "arn:aws:logs:us-east-1:111122223333:log-group:*"
                     ]
            }
        }
    },
    {
        "Action": "s3:PutObject" ,
        "Effect": "Allow",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "Principal": { "Service": "logs.us-east-1.amazonaws.com" },
        "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                "123456789012",
                "111122223333"
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                    "arn:aws:logs:us-east-1:123456789012:log-group:*",
                    "arn:aws:logs:us-east-1:111122223333:log-group:*"
                    ]
            }
        }
    },
    {
        "Effect": "Allow",
        "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/role_name"
        },
        "Action": "s3:PutObject",
        "Resource": "arn:aws:s3:::>amzn-s3-demo-bucket/*",
        "Condition": {
            "StringEquals": {
```

```
                "s3:x-amz-acl": "bucket-owner-full-control"
            }
          }
        }
    ]
}
```

2. Configure 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 amzn-s3-demo-bucket --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) Exportación a un bucket de Amazon S3 de destino cifrado con SSE-KMS para la exportación entre cuentas (CLI)

Este procedimiento 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:

   - *us-east-1*Sustitúyalo por la región de sus registros.
   - *account-ARN*Sustitúyalo por el ARN de la cuenta propietaria de la clave KMS.
   - *123456789012*Sustitúyalo por el número de cuenta propietario de la clave KMS.
   - *key_id*con el identificador de la clave kms.

- *role_name*con el rol utilizado para crear la tarea de exportación.

JSON

```
{
"Version":"2012-10-17",
"Statement": [
    {
        "Sid": "AllowCWLServicePrincipalUsage",
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.us-east-1.amazonaws.com"
        },
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "*"
    },
    {
        "Sid": "EnableIAMUserPermissions",
        "Effect": "Allow",
        "Principal": {
            "AWS": "account-ARN"
        },
        "Action": [
            "kms:GetKeyPolicy*",
            "kms:PutKeyPolicy*",
            "kms:DescribeKey*",
            "kms:CreateAlias*",
            "kms:ScheduleKeyDeletion*",
            "kms:Decrypt"
        ],
        "Resource": "*"
    },
    {
        "Sid": "EnableIAMRolePermissions",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/role_name"
        },
```

```
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

2. Introduzca 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-1:123456789012: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. Introduzca el siguiente comando y *amzn-s3-demo-bucket* sustitúyalo por el nombre del depósito al que va a exportar los registros.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-
encryption-configuration file://bucketencryption.json
```

   Si el comando no devuelve ningún error, el proceso se ha realizado correctamente.

Crear una tarea de exportación para la exportación multicuenta (CLI)

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 Configurar permisos de acceso (CLI).
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 "amzn-s3-demo-bucket" --destination-prefix "export-
task-output"
```

   A continuación, se muestra un ejemplo del resultado.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Describa las tareas de exportación (CLI)

Después de crear una tarea de exportación, puede obtener el estado actual de la tarea.

Para describir las tareas de exportación mediante la 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": "amzn-s3-demo-bucket",
      "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
   }]
}
```

Cancelar una tarea de exportación (CLI)

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 en Amazon Logs para poder transmitir datos a tu clúster de Amazon OpenSearch Service prácticamente en tiempo real. 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 acerca de las clases de registros, consulte Clases de registro.

En función de la cantidad de datos de registro que se transmitan, considere la posibilidad de establecer un límite de simultaneidad a nivel de función. Para obtener más información, consulte Escalado de funciones de Lambda

> **Note**
>
> Dado que la transmisión de grandes cantidades de datos de CloudWatch Logs al OpenSearch Servicio puede generar altos cargos por uso, le recomendamos que cree un presupuesto en la Administración de facturación y costos de AWS consola. Para obtener más información, consulta Cómo administrar tus costes con AWS Budgets.

En esta sección se describen los requisitos previos que debe cumplir antes de suscribir un grupo de registros a OpenSearch Service. También describe cómo suscribir un grupo de registros al OpenSearch Servicio.

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 Cómo 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:

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOpenSearchStreamingAccess",
            "Action": [
                "es:*"
            ],
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:es:us-
east-1:123456789012:domain/cloudwatch-logs/*"
        }
    ]
}
```

- 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 SDKs

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 distintas funciones de servicio, es posible ver las acciones en contexto en los escenarios relacionados.

Los escenarios son ejemplos de código que muestran cómo llevar a cabo una tarea específica a través de llamadas a varias funciones dentro del servicio o combinado con otros 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

Ejemplos básicos de CloudWatch registros que utilizan AWS SDKs

Los siguientes ejemplos de código muestran cómo utilizar los conceptos básicos de Amazon CloudWatch Logs con AWS SDKs.

Ejemplos

Acciones para los CloudWatch registros que utilizan AWS SDKs

Los siguientes ejemplos de código muestran cómo realizar acciones de CloudWatch Logs individuales con AWS SDKs. Cada ejemplo incluye un enlace a GitHub, donde puede encontrar instrucciones para configurar y ejecutar el código.

Estos extractos se denominan API de CloudWatch registros y son fragmentos de código de programas más grandes que deben ejecutarse en su contexto. Puede ver las acciones en contexto en Escenarios para CloudWatch los registros que utilizan AWS SDKs.

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

- AssociateKmsKeyÚselo con un AWS SDK
- CancelExportTaskÚselo con un AWS SDK
- CreateExportTaskÚselo con un AWS SDK
- Úselo CreateLogGroup con un AWS SDK o CLI
- Úselo CreateLogStream con un AWS SDK o CLI
- Úselo DeleteLogGroup con un AWS SDK o CLI
- DeleteSubscriptionFilterÚselo con un AWS SDK
- DescribeExportTasksÚselo con un AWS SDK
- Úselo DescribeLogGroups con un AWS SDK o CLI
- Úselo DescribeLogStreams con un AWS SDK o CLI
- DescribeSubscriptionFiltersÚselo con un AWS SDK
- Úselo GetLogEvents con un AWS SDK o CLI
- GetQueryResultsÚselo con un AWS SDK
- PutSubscriptionFilterÚselo con un AWS SDK
- StartLiveTailÚselo con un AWS SDK
- StartQueryÚselo con un AWS SDK

`AssociateKmsKey`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `AssociateKmsKey`.

.NET

SDK para .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 para .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.

`CancelExportTask`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `CancelExportTask`.

.NET

SDK para .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 para .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.

`CreateExportTask`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `CreateExportTask`.

.NET

SDK para .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 = "amzn-s3-demo-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 para .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

Los siguientes ejemplos de código muestran cómo utilizar `CreateLogGroup`.

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:

- Configuración de Amazon ECS Service Connect

.NET

SDK para .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 para .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 para 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

Los siguientes ejemplos de código muestran cómo utilizar `CreateLogStream`.

.NET

SDK para .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 para .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

Los siguientes ejemplos de código muestran cómo utilizar `DeleteLogGroup`.

Los ejemplos de acciones son extractos de código de programas más grandes y deben ejecutarse en contexto. Puede ver esta acción en contexto en los siguientes ejemplos de código:

- Configuración de Amazon ECS Service Connect
- Creación de su primera función de Lambda

.NET

SDK para .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 para .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 para 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.

`DeleteSubscriptionFilter`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran 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 para 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 para 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 más información, consulte la Guía para desarrolladores de AWS SDK para JavaScript.
- Para obtener más información sobre la API, consulta DeleteSubscriptionFilterla Referencia AWS SDK para 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.fromEnvironment { 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.

`DescribeExportTasks`Úselo con un AWS SDK

En el siguiente ejemplo de código, se muestra cómo utilizar `DescribeExportTasks`.

.NET

SDK para .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 para .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

Los siguientes ejemplos de código muestran cómo utilizar `DescribeLogGroups`.

.NET

SDK para .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 para .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?.every((lg) => !!lg)) {
      logGroups.push(...page.logGroups);
    }
  }

  console.log(logGroups);
  return logGroups;
};
```

- Para obtener más información sobre la API, consulta DescribeLogGroupsla Referencia AWS SDK para 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 **`DescribeLogStreams`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `DescribeLogStreams`.

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:

- Creación de su primera función de Lambda

CLI

AWS CLI

El siguiente comando muestra todos los flujos de registro que comienzan con el prefijo `2015` del grupo de registro `my-logs`:

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-
prefix 2015
```

Salida:

```
{
```

```
    "logStreams": [
        {
            "creationTime": 1433189871774,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150531",
            "logStreamName": "20150531",
            "storedBytes": 0
        },
        {
            "creationTime": 1433189873898,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:log-
stream:20150601",
            "logStreamName": "20150601",
            "storedBytes": 0
        }
    ]
}
```

- Para obtener más información sobre la API, consulte DescribeLogStreamsla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

Busca flujos de registro dentro de un grupo de registros específico que coincidan con un prefijo determinado.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CloudWatchLogsSearch {

    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <logGroupName> <logStreamName>

                Where:
                  logGroupName - The name of the log group (for example,
 WeathertopJavaContainerLogs).
                  logStreamName - The name of the log stream (for example,
 weathertop-java-stream).
                  pattern - the pattern to use (for example, INFO)

                """;

        if (args.length != 3) {
            System.out.print(usage);
            System.exit(1);
        }

        String logGroupName = args[0] ;
        String logStreamName = args[1] ;
        String pattern = args[2] ;

        CloudWatchLogsClient cwlClient = CloudWatchLogsClient.builder()
                .region(Region.US_EAST_1)
                .build();

        searchLogStreamsAndFilterEvents(cwlClient, logGroupName, logStreamName,
 pattern);
    }

    /**
     * Searches for log streams with a specific prefix within a log group and
 filters log events based on a specified pattern.
     *
     * @param cwlClient       the CloudWatchLogsClient used to interact with AWS
 CloudWatch Logs
     * @param logGroupName    the name of the log group to search within
```

```
     * @param logStreamPrefix the prefix of the log streams to search for
     * @param pattern         the pattern to filter log events by
     */
    public static void searchLogStreamsAndFilterEvents(CloudWatchLogsClient
 cwlClient, String logGroupName, String logStreamPrefix, String pattern) {
        DescribeLogStreamsRequest describeLogStreamsRequest =
 DescribeLogStreamsRequest.builder()
                .logGroupName(logGroupName)
                .logStreamNamePrefix(logStreamPrefix)
                .build();

        DescribeLogStreamsResponse describeLogStreamsResponse =
 cwlClient.describeLogStreams(describeLogStreamsRequest);
        List<LogStream> logStreams = describeLogStreamsResponse.logStreams();

        for (LogStream logStream : logStreams) {
            String logStreamName = logStream.logStreamName();
            System.out.println("Searching in log stream: " + logStreamName);

            FilterLogEventsRequest filterLogEventsRequest =
 FilterLogEventsRequest.builder()
                    .logGroupName(logGroupName)
                    .logStreamNames(logStreamName)
                    .filterPattern(pattern)
                    .build();

            FilterLogEventsResponse filterLogEventsResponse =
 cwlClient.filterLogEvents(filterLogEventsRequest);

            for (FilteredLogEvent event : filterLogEventsResponse.events()) {
                System.out.println(event.message());
            }


 System.out.println("--------------------------------------------------"); //
 Separator for better readability
        }
    }
}
```

Imprime los metadatos sobre el flujo de registro más reciente de un grupo de registros especificado.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CloudWatchLogQuery {
    public static void main(final String[] args) {
        final String usage = """
                Usage:
                  <logGroupName>

                Where:
                  logGroupName - The name of the log group (for example, /aws/
lambda/ChatAIHandler).
                """;

        if (args.length != 1) {
            System.out.print(usage);
            System.exit(1);
        }

        String logGroupName = "/aws/lambda/ChatAIHandler" ; //args[0];
        CloudWatchLogsClient logsClient = CloudWatchLogsClient.builder()
                .region(Region.US_EAST_1)
                .build();

        describeMostRecentLogStream(logsClient, logGroupName);
    }

    /**
     * Describes and prints metadata about the most recent log stream in the
 specified log group.
     *
     * @param logsClient   the CloudWatchLogsClient used to interact with AWS
 CloudWatch Logs
     * @param logGroupName the name of the log group
     */
```

```
    public static void describeMostRecentLogStream(CloudWatchLogsClient
 logsClient, String logGroupName) {
        DescribeLogStreamsRequest streamsRequest =
 DescribeLogStreamsRequest.builder()
                .logGroupName(logGroupName)
                .orderBy(OrderBy.LAST_EVENT_TIME)
                .descending(true)
                .limit(1)
                .build();

        try {
            DescribeLogStreamsResponse streamsResponse =
 logsClient.describeLogStreams(streamsRequest);
            List<LogStream> logStreams = streamsResponse.logStreams();

            if (logStreams.isEmpty()) {
                System.out.println("No log streams found for log group: " +
 logGroupName);
                return;
            }

            LogStream stream = logStreams.get(0);
            System.out.println("Most Recent Log Stream:");
            System.out.println("  Name: " + stream.logStreamName());
            System.out.println("  ARN: " + stream.arn());
            System.out.println("  Creation Time: " + stream.creationTime());
            System.out.println("  First Event Time: " +
 stream.firstEventTimestamp());
            System.out.println("  Last Event Time: " +
 stream.lastEventTimestamp());
            System.out.println("  Stored Bytes: " + stream.storedBytes());
            System.out.println("  Upload Sequence Token: " +
 stream.uploadSequenceToken());

        } catch (CloudWatchLogsException e) {
            System.err.println("Failed to describe log stream: " +
 e.awsErrorDetails().errorMessage());
        }
    }
}
```

- Para obtener más información sobre la API, consulta DescribeLogStreamsla Referencia AWS SDK for Java 2.x 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.

`DescribeSubscriptionFilters`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran 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 DescribeSubscriptionFilterslla Referencia AWS SDK para 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 DescribeSubscriptionFilterslaReferencia 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 DescribeSubscriptionFilterslla Referencia AWS SDK para 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 más información, consulte la Guía para desarrolladores de AWS SDK para JavaScript.
- Para obtener más información sobre la API, consulta DescribeSubscriptionFiltersla Referencia AWS SDK para 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.fromEnvironment { 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 **`GetLogEvents`** con un AWS SDK o CLI

Los siguientes ejemplos de código muestran cómo utilizar `GetLogEvents`.

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:

- Creación de su primera función de Lambda

CLI

AWS CLI

El siguiente comando recupera eventos de registro de un flujo de registro denominado `20150601` en el grupo de registro `my-logs`:

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

Salida:

```
{
    "nextForwardToken":
 "f/31961209122447488583055879464742346735121166569214640130",
    "events": [
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
            "ingestionTime": 1433190516679,
            "timestamp": 1433190184356,
            "message": "Example Event 1"
        },
        {
            "ingestionTime": 1433190494190,
            "timestamp": 1433190184358,
            "message": "Example Event 2"
```

```
        }
    ],
    "nextBackwardToken":
 "b/319612091223582856022617569449886743245533732682167091 20"
}
```

- Para obtener más información sobre la API, consulte GetLogEventsla Referencia de AWS CLI comandos.

Java

SDK para Java 2.x

> **Note**
>
> Hay más información al respecto GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsResponse;
import software.amazon.awssdk.services.cloudwatchlogs.model.GetLogEventsRequest;
import software.amazon.awssdk.services.cloudwatchlogs.model.GetLogEventsResponse;

import java.time.Instant;
import java.time.temporal.ChronoUnit;

/**
 * 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 GetLogEvents {

    public static void main(String[] args) {

        final String usage = """

                Usage:
                  <logGroupName> <logStreamName>

                Where:
                  logGroupName - The name of the log group (for example,
 myloggroup).
                  logStreamName - The name of the log stream (for example,
 mystream).

                """;

       // if (args.length != 2) {
       //     System.out.print(usage);
       //     System.exit(1);
//        }

        String logGroupName = "WeathertopJavaContainerLogs" ; //args[0];
        String logStreamName = "weathertop-java-stream" ; //args[1];

        Region region = Region.US_EAST_1 ;
        CloudWatchLogsClient cloudWatchLogsClient =
 CloudWatchLogsClient.builder()
                .region(region)
                .build();

        getCWLogEvents(cloudWatchLogsClient, logGroupName, logStreamName);
        cloudWatchLogsClient.close();
    }

    public static void getCWLogEvents(CloudWatchLogsClient cloudWatchLogsClient,
                                      String logGroupName,
                                      String logStreamPrefix) {
        try {
            // First, find the exact log stream name
            DescribeLogStreamsRequest describeRequest =
 DescribeLogStreamsRequest.builder()
                    .logGroupName(logGroupName)
                    .logStreamNamePrefix(logStreamPrefix)
```

```
                    .limit(1) // get the first matching stream
                    .build();

            DescribeLogStreamsResponse describeResponse =
cloudWatchLogsClient.describeLogStreams(describeRequest);

            if (describeResponse.logStreams().isEmpty()) {
                System.out.println("No matching log streams found for prefix: " +
logStreamPrefix);
                return;
            }

            String exactLogStreamName =
describeResponse.logStreams().get(0).logStreamName();
            System.out.println("Using exact log stream: " + exactLogStreamName);

            long startTime = Instant.now().minus(7,
ChronoUnit.DAYS).toEpochMilli();
            long endTime = Instant.now().toEpochMilli();

            GetLogEventsRequest getLogEventsRequest =
GetLogEventsRequest.builder()
                    .logGroupName(logGroupName)
                    .logStreamName(exactLogStreamName) // <-- exact name, not
prefix
                    .startTime(startTime)
                    .endTime(endTime)
                    .startFromHead(true)
                    .build();

            GetLogEventsResponse response =
cloudWatchLogsClient.getLogEvents(getLogEventsRequest);

            if (response.events().isEmpty()) {
                System.out.println("No log events found in the past 7 days.");
            } else {
                response.events().forEach(e -> System.out.println(e.message()));
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

```
}
```

- Para obtener más información sobre la API, consulta GetLogEventsla Referencia AWS SDK for Java 2.x 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.

`GetQueryResults`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran 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

.NET

SDK para .NET (v4)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
   /// <summary>
   /// Gets the results of a CloudWatch Logs Insights query.
   /// </summary>
   /// <param name="queryId">The ID of the query.</param>
   /// <returns>The query results response.</returns>
   public async Task<GetQueryResultsResponse?> GetQueryResultsAsync(string
queryId)
   {
       try
       {
           var request = new GetQueryResultsRequest
```

```
            {
                QueryId = queryId
            };

            var response = await
_amazonCloudWatchLogs.GetQueryResultsAsync(request);
            return response;
        }
        catch (ResourceNotFoundException ex)
        {
            _logger.LogError($"Query not found: {ex.Message}");
            return null;
        }
        catch (Exception ex)
        {
            _logger.LogError($"An error occurred while getting query results:
{ex.Message}");
            return null;
        }
    }
```

- Para obtener más información sobre la API, consulta GetQueryResultsla Referencia AWS SDK para .NET 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.

```
/**
 * 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 para 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).

SAP ABAP

SDK para SAP ABAP

> **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.

```
    TRY.
        oo_result = lo_cwl->getqueryresults(
          iv_queryid = iv_query_id ).

        " Display query status and result count
        DATA(lv_status) = oo_result->get_status( ).
        DATA(lt_results) = oo_result->get_results( ).
        DATA(lv_result_count) = lines( lt_results ).

        MESSAGE |Query status: { lv_status }. Retrieved { lv_result_count } log
event(s).| TYPE 'I'.
      CATCH /aws1/cx_cwlinvalidparameterex.
        MESSAGE 'Invalid parameter.' TYPE 'E'.
      CATCH /aws1/cx_cwlresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'E'.
      CATCH /aws1/cx_cwlserviceunavailex.
        MESSAGE 'Service unavailable.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte GetQueryResultsla referencia sobre la API ABAP del AWS SDK para SAP.

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.

`PutSubscriptionFilter`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran 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 para 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 para 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 más información, consulte la Guía para desarrolladores de AWS SDK para JavaScript.
- Para obtener más información sobre la API, consulta PutSubscriptionFilterla Referencia AWS SDK para 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.

`StartLiveTail`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran cómo utilizar `StartLiveTail`.

.NET

SDK para .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 para .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 para 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 para 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.

`StartQuery`Úselo con un AWS SDK

Los siguientes ejemplos de código muestran 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

.NET

SDK para .NET (v4)

> **Note**
>
> Hay más información GitHub. Busque el ejemplo completo y aprenda a configurar y ejecutar en el Repositorio de ejemplos de código de AWS.

```
    /// <summary>
    /// Starts a CloudWatch Logs Insights query.
    /// </summary>
    /// <param name="logGroupName">The name of the log group to query.</param>
    /// <param name="queryString">The CloudWatch Logs Insights query string.</
param>
    /// <param name="startTime">The start time for the query (seconds since
 epoch).</param>
    /// <param name="endTime">The end time for the query (seconds since epoch).</
param>
    /// <param name="limit">The maximum number of results to return.</param>
    /// <returns>The query ID if successful, null otherwise.</returns>
    public async Task<string?> StartQueryAsync(
        string logGroupName,
        string queryString,
        long startTime,
        long endTime,
        int limit = 10000)
    {
        try
        {
            var request = new StartQueryRequest
            {
                LogGroupName = logGroupName,
                QueryString = queryString,
                StartTime = startTime,
                EndTime = endTime,
                Limit = limit
            };

            var response = await _amazonCloudWatchLogs.StartQueryAsync(request);
            return response.QueryId;
        }
        catch (InvalidParameterException ex)
        {
            _logger.LogError($"Invalid parameter for query: {ex.Message}");
            return null;
        }
        catch (ResourceNotFoundException ex)
        {
            _logger.LogError($"Log group not found: {ex.Message}");
            return null;
        }
```

```
        catch (Exception ex)
        {
            _logger.LogError($"An error occurred while starting query:
{ex.Message}");
            return null;
        }
    }
```

- Para obtener más información sobre la API, consulta StartQueryla Referencia AWS SDK para .NET 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.

```
/**
 * 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 para 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_group,
                    startTime=start_time,
                    endTime=end_time,
                    queryString=self.query_string,
                    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_group,
                startTime=start_time,
                endTime=end_time,
                queryString=self.query_string,
                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).

SAP ABAP

SDK para SAP ABAP

> **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.

```
    TRY.
        " iv_log_group_name = '/aws/lambda/my-function'
        " iv_query_string = 'fields @timestamp, @message | sort @timestamp desc |
limit 20'
        " iv_start_time and iv_end_time must be in Unix epoch milliseconds (ms
since Jan 1, 1970 00:00:00 UTC)
        oo_result = lo_cwl->startquery(
```

```
          iv_loggroupname = iv_log_group_name
          iv_starttime    = iv_start_time
          iv_endtime      = iv_end_time
          iv_querystring  = iv_query_string
          iv_limit        = iv_limit ).

        " Display the query ID for tracking
        DATA(lv_query_id) = oo_result->get_queryid( ).
        MESSAGE |Query started successfully with ID: { lv_query_id }| TYPE 'I'.
      CATCH /aws1/cx_cwlinvalidparameterex.
        MESSAGE 'Invalid parameter.' TYPE 'E'.
      CATCH /aws1/cx_cwllimitexceededex.
        MESSAGE 'Limit exceeded.' TYPE 'E'.
      CATCH /aws1/cx_cwlmalformedqueryex.
        MESSAGE 'Malformed query.' TYPE 'E'.
      CATCH /aws1/cx_cwlresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'E'.
      CATCH /aws1/cx_cwlserviceunavailex.
        MESSAGE 'Service unavailable.' TYPE 'E'.
    ENDTRY.
```

- Para obtener más información sobre la API, consulte StartQueryla referencia sobre la API ABAP del AWS SDK para SAP.

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.

Escenarios para CloudWatch los registros que utilizan AWS SDKs

Los siguientes ejemplos de código muestran cómo implementar escenarios comunes en CloudWatch Logs with AWS SDKs. Estos escenarios le muestran cómo realizar tareas específicas mediante la llamada a varias funciones de CloudWatch Logs o combinándolas con otras Servicios de AWS. En cada escenario se incluye un enlace al código fuente completo, con instrucciones de configuración y ejecución del código.

Los escenarios requieren un nivel intermedio de experiencia para entender las acciones de servicio en su contexto.

Ejemplos

- Configuración de Amazon ECS Service Connect
- Creación de su primera función de Lambda
- Usa CloudWatch los registros para ejecutar una consulta grande
- Uso de eventos programados para invocar una función de Lambda

Configuración de Amazon ECS Service Connect

En el siguiente ejemplo de código, se muestra cómo:

- Crear la infraestructura de la VPC
- Configurar el registro
- Crear el clúster de ECS
- Configurar roles de IAM
- Crear el servicio con Service Connect
- Verifique la implementación
- Eliminar recursos

Bash

AWS CLI con el script Bash

> **Note**
>
> Hay más información. GitHub Encuentre el ejemplo completo y obtenga información sobre cómo configurarlo y ejecutarlo en el repositorio de Tutoriales para desarrolladores de ejemplo.

```
#!/bin/bash

# ECS Service Connect Tutorial Script v4 - Modified to use Default VPC
# This script creates an ECS cluster with Service Connect and deploys an nginx
 service
# Uses the default VPC to avoid VPC limits

set -e  # Exit on any error
```

```
# Configuration
SCRIPT_NAME="ECS Service Connect Tutorial"
LOG_FILE="ecs-service-connect-tutorial-v4-default-vpc.log"
REGION=${AWS_DEFAULT_REGION:-${AWS_REGION:-$(aws configure get region 2>/dev/
null)}}
if [ -z "$REGION" ]; then
    echo "ERROR: No AWS region configured."
    echo "Set one with: aws configure set region us-east-1"
    exit 1
fi
ENV_PREFIX="tutorial"
CLUSTER_NAME="${ENV_PREFIX}-cluster"
NAMESPACE_NAME="service-connect"

# Generate random suffix for unique resource names
RANDOM_SUFFIX=$(openssl rand -hex 6)

# Arrays to track created resources for cleanup
declare -a CREATED_RESOURCES=()

# Logging function
log() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}

# Error handling function
handle_error() {
    log "ERROR: Script failed at line $1"
    log "Attempting to clean up resources..."
    cleanup_resources
    exit 1
}

# Set up error handling
trap 'handle_error $LINENO' ERR

# Function to add resource to tracking array
track_resource() {
    CREATED_RESOURCES+=("$1")
    log "Tracking resource: $1"
}

# Function to check if command output contains actual errors
```

```
check_for_errors() {
    local output="$1"
    local command_name="$2"

    # Check for specific AWS CLI error patterns, not just any occurrence of
 "error"
    if echo "$output" | grep -qi "An error occurred\|InvalidParameterException\|
AccessDenied\|ValidationException\|ResourceNotFoundException"; then
        log "ERROR in $command_name: $output"
        return 1
    fi
    return 0
}

# Function to get AWS account ID
get_account_id() {
    ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
    log "Using AWS Account ID: $ACCOUNT_ID"
}

# Function to wait for resources to be ready
wait_for_resource() {
    local resource_type="$1"
    local resource_id="$2"

    case "$resource_type" in
        "cluster")
            log "Waiting for cluster $resource_id to be active..."
            local attempt=1
            local max_attempts=30
            while [ $attempt -le $max_attempts ]; do
                local status=$(aws ecs describe-clusters --clusters
 "$resource_id" --query 'clusters[0].status' --output text)
                if [ "$status" = "ACTIVE" ]; then
                    log "Cluster is now active"
                    return 0
                fi
                log "Cluster status: $status (attempt $attempt/$max_attempts)"
                sleep 10
                ((attempt++))
            done
            log "ERROR: Cluster did not become active within expected time"
            return 1
            ;;
```

```
        "service")
            log "Waiting for service $resource_id to be stable..."
            aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$resource_id"
            ;;
        "nat-gateway")
            log "Waiting for NAT Gateway $resource_id to be available..."
            aws ec2 wait nat-gateway-available --nat-gateway-ids "$resource_id"
            ;;
    esac
}

# Function to use default VPC infrastructure
setup_default_vpc_infrastructure() {
    log "Using default VPC infrastructure..."

    # Get default VPC
    VPC_ID=$(aws ec2 describe-vpcs --filters "Name=isDefault,Values=true" --query
 'Vpcs[0].VpcId' --output text)
    if [[ "$VPC_ID" == "None" || -z "$VPC_ID" ]]; then
        log "ERROR: No default VPC found. Please create a default VPC first."
        exit 1
    fi
    log "Using default VPC: $VPC_ID"

    # Get default subnets
    SUBNETS=$(aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC_ID"
 "Name=default-for-az,Values=true" --query 'Subnets[].SubnetId' --output text)
    SUBNET_ARRAY=($SUBNETS)

    if [ ${#SUBNET_ARRAY[@]} -lt 2 ]; then
        log "ERROR: Need at least 2 subnets for ECS Service Connect. Found:
 ${#SUBNET_ARRAY[@]}"
        exit 1
    fi

    PUBLIC_SUBNET1=${SUBNET_ARRAY[0]}
    PUBLIC_SUBNET2=${SUBNET_ARRAY[1]}

    log "Using subnets: $PUBLIC_SUBNET1, $PUBLIC_SUBNET2"

    # Create security group for ECS tasks
    SG_OUTPUT=$(aws ec2 create-security-group \
        --group-name "${ENV_PREFIX}-ecs-sg-${RANDOM_SUFFIX}" \
```

```
        --description "Security group for ECS Service Connect tutorial" \
        --vpc-id "$VPC_ID" \
        --tag-specifications 'ResourceType=security-
group,Tags=[{Key=project,Value=doc-smith},{Key=tutorial,Value=amazon-ecs-service-
connect}]' 2>&1)
    check_for_errors "$SG_OUTPUT" "create-security-group"
    SECURITY_GROUP_ID=$(echo "$SG_OUTPUT" | grep -o '"GroupId": "[^"]*"' | cut -
d'"' -f4)
    track_resource "SG:$SECURITY_GROUP_ID"
    log "Created security group: $SECURITY_GROUP_ID"

    # Add inbound rules to security group
    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 80 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    aws ec2 authorize-security-group-ingress \
        --group-id "$SECURITY_GROUP_ID" \
        --protocol tcp \
        --port 443 \
        --cidr 0.0.0.0/0 >/dev/null 2>&1 || true

    log "Default VPC infrastructure setup completed"
}

# Function to create CloudWatch log groups
create_log_groups() {
    log "Creating CloudWatch log groups..."

    # Create log group for nginx container
    aws logs create-log-group --log-group-name "/ecs/service-connect-nginx"
 --tags project=doc-smith,tutorial=amazon-ecs-service-connect 2>&1 | grep -v
 "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-nginx created"
            track_resource "LOG_GROUP:/ecs/service-connect-nginx"
        else
            log "Log group /ecs/service-connect-nginx already exists"
        fi
    }

    # Create log group for service connect proxy
```

```
    aws logs create-log-group --log-group-name "/ecs/service-connect-proxy"
 --tags project=doc-smith,tutorial=amazon-ecs-service-connect 2>&1 | grep -v
 "ResourceAlreadyExistsException" || {
        if [ ${PIPESTATUS[0]} -eq 0 ]; then
            log "Log group /ecs/service-connect-proxy created"
            track_resource "LOG_GROUP:/ecs/service-connect-proxy"
        else
            log "Log group /ecs/service-connect-proxy already exists"
        fi
    }
}

# Function to create ECS cluster with Service Connect
create_ecs_cluster() {
    log "Creating ECS cluster with Service Connect..."

    CLUSTER_OUTPUT=$(aws ecs create-cluster \
        --cluster-name "$CLUSTER_NAME" \
        --service-connect-defaults namespace="$NAMESPACE_NAME" \
        --tags key=Environment,value=tutorial key=project,value=doc-smith
 key=tutorial,value=amazon-ecs-service-connect 2>&1)
    check_for_errors "$CLUSTER_OUTPUT" "create-cluster"

    track_resource "CLUSTER:$CLUSTER_NAME"
    log "Created ECS cluster: $CLUSTER_NAME"

    wait_for_resource "cluster" "$CLUSTER_NAME"

    # Track the Service Connect namespace that gets created
    # Wait a moment for the namespace to be created
    sleep 5
    NAMESPACE_ID=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

    if [[ -n "$NAMESPACE_ID" && "$NAMESPACE_ID" != "None" ]]; then
        track_resource "NAMESPACE:$NAMESPACE_ID"
        log "Service Connect namespace created: $NAMESPACE_ID"
    fi
}

# Function to create IAM roles
create_iam_roles() {
```

```
    log "Creating IAM roles..."

    # Check if ecsTaskExecutionRole exists
    if aws iam get-role --role-name ecsTaskExecutionRole >/dev/null 2>&1; then
        log "IAM role ecsTaskExecutionRole exists"
    else
        log "Creating ecsTaskExecutionRole..."
        aws iam create-role \
            --role-name ecsTaskExecutionRole \
            --assume-role-policy-document '{
                "Version":"2012-10-17",
                "Statement": [{
                    "Effect": "Allow",
                    "Principal": {"Service": "ecs-tasks.amazonaws.com"},
                    "Action": "sts:AssumeRole"
                }]
            }' >/dev/null 2>&1
        aws iam attach-role-policy \
            --role-name ecsTaskExecutionRole \
            --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy >/dev/null 2>&1
        track_resource "ROLE:ecsTaskExecutionRole"
        aws iam tag-role --role-name ecsTaskExecutionRole --tags
 Key=project,Value=doc-smith Key=tutorial,Value=amazon-ecs-service-connect
        log "Created ecsTaskExecutionRole"
        sleep 10
    fi

    # Check if ecsTaskRole exists, create if not
    if aws iam get-role --role-name ecsTaskRole >/dev/null 2>&1; then
        log "IAM role ecsTaskRole exists"
    else
        log "IAM role ecsTaskRole does not exist, will create it"

        # Create trust policy for ECS tasks
        cat > /tmp/ecs-task-trust-policy.json << EOF
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            },
```

```
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF

        aws iam create-role \
            --role-name ecsTaskRole \
            --assume-role-policy-document file:///tmp/ecs-task-trust-policy.json
 >/dev/null

        track_resource "IAM_ROLE:ecsTaskRole"
        aws iam tag-role --role-name ecsTaskRole --tags Key=project,Value=doc-
smith Key=tutorial,Value=amazon-ecs-service-connect
        log "Created ecsTaskRole"

        # Wait for role to be available
        sleep 10
    fi
}

# Function to create task definition
create_task_definition() {
    log "Creating task definition..."

    # Create task definition JSON
    cat > /tmp/task-definition.json << EOF
{
    "family": "service-connect-nginx",
    "networkMode": "awsvpc",
    "requiresCompatibilities": ["FARGATE"],
    "cpu": "256",
    "memory": "512",
    "executionRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskExecutionRole",
    "taskRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskRole",
    "containerDefinitions": [
        {
            "name": "nginx",
            "image": "public.ecr.aws/docker/library/nginx:latest",
            "portMappings": [
                {
                    "containerPort": 80,
                    "protocol": "tcp",
                    "name": "nginx-port"
```

```
                }
            ],
            "essential": true,
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/service-connect-nginx",
                    "awslogs-region": "${REGION}",
                    "awslogs-stream-prefix": "ecs"
                }
            }
        }
    ]
}
EOF

    TASK_DEF_OUTPUT=$(aws ecs register-task-definition --cli-input-json file:///
tmp/task-definition.json 2>&1)
    check_for_errors "$TASK_DEF_OUTPUT" "register-task-definition"

    TASK_DEF_ARN=$(echo "$TASK_DEF_OUTPUT" | grep -o '"taskDefinitionArn":
 "[^"]*"' | cut -d'"' -f4)
    track_resource "TASK_DEF:service-connect-nginx"
    log "Created task definition: $TASK_DEF_ARN"

    # Clean up temporary file
    rm -f /tmp/task-definition.json
}

# Function to create ECS service with Service Connect
create_ecs_service() {
    log "Creating ECS service with Service Connect..."

    # Create service definition JSON
    cat > /tmp/service-definition.json << EOF
{
    "serviceName": "service-connect-nginx-service",
    "cluster": "${CLUSTER_NAME}",
    "taskDefinition": "service-connect-nginx",
    "desiredCount": 1,
    "launchType": "FARGATE",
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": ["${PUBLIC_SUBNET1}", "${PUBLIC_SUBNET2}"],
```

```
            "securityGroups": ["${SECURITY_GROUP_ID}"],
            "assignPublicIp": "ENABLED"
        }
    },
    "serviceConnectConfiguration": {
        "enabled": true,
        "namespace": "${NAMESPACE_NAME}",
        "services": [
            {
                "portName": "nginx-port",
                "discoveryName": "nginx",
                "clientAliases": [
                    {
                        "port": 80,
                        "dnsName": "nginx"
                    }
                ]
            }
        ],
        "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
                "awslogs-group": "/ecs/service-connect-proxy",
                "awslogs-region": "${REGION}",
                "awslogs-stream-prefix": "ecs-service-connect"
            }
        }
    },
    "tags": [
        {
            "key": "Environment",
            "value": "tutorial"
        },
        {
            "key": "project",
            "value": "doc-smith"
        },
        {
            "key": "tutorial",
            "value": "amazon-ecs-service-connect"
        }
    ]
}
EOF
```

```
    SERVICE_OUTPUT=$(aws ecs create-service --cli-input-json file:///tmp/service-
definition.json 2>&1)
    check_for_errors "$SERVICE_OUTPUT" "create-service"

    track_resource "SERVICE:service-connect-nginx-service"
    log "Created ECS service: service-connect-nginx-service"

    wait_for_resource "service" "service-connect-nginx-service"

    # Clean up temporary file
    rm -f /tmp/service-definition.json
}

# Function to verify deployment
verify_deployment() {
    log "Verifying deployment..."

    # Check service status
    SERVICE_STATUS=$(aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].status' --output text)
    log "Service status: $SERVICE_STATUS"

    # Check running tasks
    RUNNING_COUNT=$(aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].runningCount' --output text)
    log "Running tasks: $RUNNING_COUNT"

    # Get task ARN
    TASK_ARN=$(aws ecs list-tasks \
        --cluster "$CLUSTER_NAME" \
        --service-name "service-connect-nginx-service" \
        --query 'taskArns[0]' --output text)

    if [[ "$TASK_ARN" != "None" && -n "$TASK_ARN" ]]; then
        log "Task ARN: $TASK_ARN"

        # Try to get task IP address
        TASK_IP=$(aws ecs describe-tasks \
            --cluster "$CLUSTER_NAME" \
```

```
            --tasks "$TASK_ARN" \
            --query 'tasks[0].attachments[0].details[?
name==`privateIPv4Address`].value' \
            --output text 2>/dev/null || echo "")

        if [[ -n "$TASK_IP" && "$TASK_IP" != "None" ]]; then
            log "Task IP address: $TASK_IP"
        else
            log "Could not retrieve task IP address"
        fi
    fi

    # Check Service Connect namespace
    NAMESPACE_STATUS=$(aws servicediscovery list-namespaces \
        --filters Name=TYPE,Values=HTTP \
        --query "Namespaces[?Name=='$NAMESPACE_NAME'].Id" --output text 2>/dev/
null || echo "")

    if [[ -n "$NAMESPACE_STATUS" && "$NAMESPACE_STATUS" != "None" ]]; then
        log "Service Connect namespace '$NAMESPACE_NAME' is active"
    else
        log "Service Connect namespace '$NAMESPACE_NAME' not found or not active"
    fi

    # Display Service Connect configuration
    log "Service Connect configuration:"
    aws ecs describe-services \
        --cluster "$CLUSTER_NAME" \
        --services "service-connect-nginx-service" \
        --query 'services[0].serviceConnectConfiguration' 2>/dev/null || true
}

# Function to display created resources
display_resources() {
    echo ""
    echo "============================================="
    echo "CREATED RESOURCES"
    echo "============================================="
    for resource in "${CREATED_RESOURCES[@]}"; do
        echo "- $resource"
    done
    echo "============================================="
    echo ""
}
```

```
# Function to cleanup resources
cleanup_resources() {
    log "Starting cleanup process..."

    # Delete resources in reverse order of creation
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        resource="${CREATED_RESOURCES[i]}"
        resource_type=$(echo "$resource" | cut -d':' -f1)
        resource_id=$(echo "$resource" | cut -d':' -f2)

        log "Cleaning up $resource_type: $resource_id"

        case "$resource_type" in
            "SERVICE")
                aws ecs update-service --cluster "$CLUSTER_NAME" --service
 "$resource_id" --desired-count 0 2>&1 | grep -qi "error" && log "Warning: Failed
 to scale down service $resource_id"
                aws ecs wait services-stable --cluster "$CLUSTER_NAME" --services
 "$resource_id" 2>/dev/null || true
                aws ecs delete-service --cluster "$CLUSTER_NAME" --service
 "$resource_id" --force 2>&1 | grep -qi "error" && log "Warning: Failed to delete
 service $resource_id"
                ;;
            "TASK_DEF")
                TASK_DEF_ARNS=$(aws ecs list-task-definitions --family-prefix
 "$resource_id" --query 'taskDefinitionArns' --output text 2>/dev/null)
                for arn in $TASK_DEF_ARNS; do
                    aws ecs deregister-task-definition --task-definition "$arn"
 >/dev/null 2>&1 || true
                done
                ;;
            "ROLE")
                aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
                aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete role $resource_id"
                ;;
            "IAM_ROLE")
                aws iam detach-role-policy --role-name "$resource_id" --policy-
arn "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy" 2>/
dev/null || true
```

```
                aws iam delete-role --role-name "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete role $resource_id"
                ;;
            "CLUSTER")
                aws ecs delete-cluster --cluster "$resource_id" 2>&1 | grep -qi
 "error" && log "Warning: Failed to delete cluster $resource_id"
                ;;
            "SG")
                for attempt in 1 2 3 4 5; do
                    if aws ec2 delete-security-group --group-id "$resource_id"
 2>/dev/null; then
                        break
                    fi
                    log "Security group $resource_id still has dependencies,
 retrying in 30s ($attempt/5)..."
                    sleep 30
                done
                ;;
            "LOG_GROUP")
                aws logs delete-log-group --log-group-name "$resource_id" 2>&1 |
 grep -qi "error" && log "Warning: Failed to delete log group $resource_id"
                ;;
            "NAMESPACE")
                # First, delete any services in the namespace
                NAMESPACE_SERVICES=$(aws servicediscovery list-services \
                    --filters Name=NAMESPACE_ID,Values="$resource_id" \
                    --query 'Services[].Id' --output text 2>/dev/null || echo "")

                if [[ -n "$NAMESPACE_SERVICES" && "$NAMESPACE_SERVICES" !=
 "None" ]]; then
                    for service_id in $NAMESPACE_SERVICES; do
                        aws servicediscovery delete-service --id "$service_id" >/
dev/null 2>&1 || true
                        sleep 2
                    done
                fi

                # Then delete the namespace
                aws servicediscovery delete-namespace --id "$resource_id" >/dev/
null 2>&1 || true
                ;;
        esac

        sleep 2  # Brief pause between deletions
```

```
    done

    # Clean up temporary files
    rm -f /tmp/ecs-task-trust-policy.json
    rm -f /tmp/task-definition.json
    rm -f /tmp/service-definition.json

    log "Cleanup completed"
}

# Main execution
main() {
    log "Starting $SCRIPT_NAME v4 (Default VPC)"
    log "Region: $REGION"
    log "Log file: $LOG_FILE"

    # Get AWS account ID
    get_account_id

    # Setup infrastructure using default VPC
    setup_default_vpc_infrastructure

    # Create CloudWatch log groups
    create_log_groups

    # Create ECS cluster
    create_ecs_cluster

    # Create IAM roles
    create_iam_roles

    # Create task definition
    create_task_definition

    # Create ECS service
    create_ecs_service

    # Verify deployment
    verify_deployment

    log "Tutorial completed successfully!"

    # Display created resources
    display_resources
```

```
    # Ask user if they want to clean up
    echo ""
    echo "==========================================="
    echo "CLEANUP CONFIRMATION"
    echo "==========================================="
    echo "Do you want to clean up all created resources? (y/n): "
    CLEANUP_CHOICE="y"

    if [[ "$CLEANUP_CHOICE" =~ ^[Yy]$ ]]; then
        cleanup_resources
        log "All resources have been cleaned up"
    else
        log "Resources left intact. You can clean them up later by running the
 cleanup function."
        echo ""
        echo "To clean up resources later, you can use the AWS CLI commands or
 the AWS Management Console."
        echo "Remember to delete resources in the correct order to avoid
 dependency issues."
    fi
}

# Make script executable and run
chmod +x "$0"
main "$@"
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de comandos de AWS CLI .
  - AttachRolePolicy
  - AuthorizeSecurityGroupIngress
  - CreateCluster
  - CreateLogGroup
  - CreateRole
  - CreateSecurityGroup
  - CreateService
  - DeleteCluster
  - DeleteLogGroup

- DeleteNamespace
- DeleteRole
- DeleteSecurityGroup
- DeleteService
- DeregisterTaskDefinition
- DescribeClusters
- DescribeServices
- DescribeSubnets
- DescribeTasks
- DescribeVpcs
- DetachRolePolicy
- GetCallerIdentity
- GetRole
- ListNamespaces
- ListServices
- ListTaskDefinitions
- ListTasks
- RegisterTaskDefinition
- UpdateService
- Wait

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.

Creación de su primera función de Lambda

En el siguiente ejemplo de código, se muestra cómo:

- Crear un rol de IAM para Lambda
- Crear el código de función
- Crear una función de Lambda

- Poner a prueba la función de Lambda
- Eliminar recursos

Bash

AWS CLI con el script Bash

> **Note**
>
> Hay más información. GitHub Encuentre el ejemplo completo y obtenga información sobre cómo configurarlo y ejecutarlo en el repositorio de Tutoriales para desarrolladores de ejemplo.

```
#!/bin/bash
# AWS Lambda - Create Your First Function
# This script creates a Lambda function, invokes it with a test event,
# views CloudWatch logs, and cleans up all resources.
#
# Source: https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html
#
# Resources created:
#   - IAM role (Lambda execution role with basic logging permissions)
#   - Lambda function (Python 3.13 or Node.js 22.x runtime)
#   - CloudWatch log group (created automatically by Lambda on invocation)

set -eE -o pipefail

################################################################################
# Setup
################################################################################

UNIQUE_ID=$(head -c 8 /dev/urandom | od -An -tx1 | tr -d ' ')
FUNCTION_NAME="my-lambda-function-${UNIQUE_ID}"
ROLE_NAME="lambda-execution-role-${UNIQUE_ID}"
LOG_GROUP_NAME="/aws/lambda/${FUNCTION_NAME}"

TEMP_DIR=$(mktemp -d)
readonly TEMP_DIR
LOG_FILE="${TEMP_DIR}/lambda-gettingstarted.log"
```

```
exec > >(tee -a "$LOG_FILE") 2>&1

declare -a CREATED_RESOURCES

################################################################################
# Helper functions
################################################################################

cleanup_resources() {
    # Disable error trap to prevent recursion during cleanup
    trap - ERR
    set +eE

    echo ""
    echo "Cleaning up resources..."
    echo ""

    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        local RESOURCE="${CREATED_RESOURCES[$i]}"
        local TYPE="${RESOURCE%%:*}"
        local NAME="${RESOURCE#*:}"

        case "$TYPE" in
            log-group)
                echo "Deleting CloudWatch log group: ${NAME}"
                aws logs delete-log-group \
                    --log-group-name "$NAME" 2>&1 || echo "  WARNING: Could not
 delete log group ${NAME}."
                ;;
            lambda-function)
                echo "Deleting Lambda function: ${NAME}"
                aws lambda delete-function \
                    --function-name "$NAME" 2>&1 || echo "  WARNING: Could not
 delete Lambda function ${NAME}."
                echo "  Waiting for function deletion to complete..."
                local DELETE_WAIT=0
                while aws lambda get-function --function-name "$NAME" > /dev/null
 2>&1; do
                    sleep 2
                    DELETE_WAIT=$((DELETE_WAIT + 2))
                    if [ "$DELETE_WAIT" -ge 60 ]; then
                        echo "  WARNING: Timed out waiting for function
 deletion."
                        break
```

```
                    fi
                done
                ;;
            iam-role-policy)
                local ROLE_PART="${NAME%%|*}"
                local POLICY_PART="${NAME#*|}"
                echo "Detaching policy from role: ${ROLE_PART}"
                aws iam detach-role-policy \
                    --role-name "$ROLE_PART" \
                    --policy-arn "$POLICY_PART" 2>&1 || echo "  WARNING: Could
 not detach policy from role ${ROLE_PART}."
                ;;
            iam-role)
                echo "Deleting IAM role: ${NAME}"
                aws iam delete-role \
                    --role-name "$NAME" 2>&1 || echo "  WARNING: Could not delete
 IAM role ${NAME}."
                ;;
        esac
    done

    if [ -d "$TEMP_DIR" ]; then
        rm -rf "$TEMP_DIR"
    fi

    echo ""
    echo "Cleanup complete."
}

handle_error() {
    echo ""
    echo "========================================="
    echo "ERROR: Script failed at $1"
    echo "========================================="
    echo ""
    if [ ${#CREATED_RESOURCES[@]} -gt 0 ]; then
        echo "Attempting to clean up ${#CREATED_RESOURCES[@]} resource(s)..."
        cleanup_resources
    fi
    exit 1
}

trap 'handle_error "line $LINENO"' ERR
```

```
wait_for_resource() {
    local DESCRIPTION="$1"
    local COMMAND="$2"
    local TARGET_VALUE="$3"
    local TIMEOUT=300
    local ELAPSED=0
    local INTERVAL=5

    echo "Waiting for ${DESCRIPTION}..."
    while true; do
        local RESULT
        RESULT=$(eval "$COMMAND" 2>&1) || true
        if echo "$RESULT" | grep -q "$TARGET_VALUE"; then
            echo "  ${DESCRIPTION} is ready."
            return 0
        fi
        if [ "$ELAPSED" -ge "$TIMEOUT" ]; then
            echo "ERROR: Timed out waiting for ${DESCRIPTION} after ${TIMEOUT}
 seconds."
            return 1
        fi
        sleep "$INTERVAL"
        ELAPSED=$((ELAPSED + INTERVAL))
    done
}

validate_input() {
    local input="$1"
    local pattern="$2"
    if ! [[ "$input" =~ $pattern ]]; then
        echo "ERROR: Invalid input: $input"
        return 1
    fi
    return 0
}

################################################################################
# Region pre-check
################################################################################

CONFIGURED_REGION=$(aws configure get region 2>/dev/null || true)
if [ -z "$CONFIGURED_REGION" ] && [ -z "$AWS_DEFAULT_REGION" ] && [ -z
 "$AWS_REGION" ]; then
    echo "ERROR: No AWS region configured."
```

```
    echo "Run 'aws configure set region <region>' or export AWS_DEFAULT_REGION."
    exit 1
fi

################################################################################
# Runtime selection
################################################################################

echo ""
echo "=========================================="
echo "AWS Lambda - Create Your First Function"
echo "=========================================="
echo ""
echo "Select a runtime for your Lambda function:"
echo "  1) Python 3.13"
echo "  2) Node.js 22.x"
echo ""
echo "Using default: Python 3.13"
RUNTIME_CHOICE="1"

case "$RUNTIME_CHOICE" in
    1)
        RUNTIME="python3.13"
        HANDLER="lambda_function.lambda_handler"
        CODE_FILE="lambda_function.py"
        cat > "${TEMP_DIR}/${CODE_FILE}" << 'PYTHON_EOF'
import json
import logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
def lambda_handler(event, context):
    if not isinstance(event, dict) or 'length' not in event or 'width' not in
 event:
        raise ValueError('Event must contain length and width')
    try:
        length = float(event['length'])
        width = float(event['width'])
        if length < 0 or width < 0:
            raise ValueError('Length and width must be non-negative')
        area = calculate_area(length, width)
        print(f'The area is {area}')
        logger.info(f'CloudWatch logs group: {context.log_group_name}')
        return json.dumps({'area': area})
    except (TypeError, ValueError) as e:
```

```
        logger.error(f'Error processing input: {str(e)}')
        raise
def calculate_area(length, width):
    return length * width
PYTHON_EOF
        echo "Selected runtime: Python 3.13"
        ;;
    2)
        RUNTIME="nodejs22.x"
        HANDLER="index.handler"
        CODE_FILE="index.mjs"
        cat > "${TEMP_DIR}/${CODE_FILE}" << 'NODEJS_EOF'
export const handler = async (event, context) => {
  if (!event || typeof event.length !== 'number' || typeof event.width !==
 'number') {
    throw new Error('Event must contain numeric length and width');
  }
  if (event.length < 0 || event.width < 0) {
    throw new Error('Length and width must be non-negative');
  }
  const area = event.length * event.width;
  console.log(`The area is ${area}`);
  console.log('CloudWatch log group: ', context.logGroupName);
  return JSON.stringify({area});
};
NODEJS_EOF
        echo "Selected runtime: Node.js 22.x"
        ;;
    *)
        echo "ERROR: Invalid choice. Please enter 1 or 2."
        exit 1
        ;;
esac

################################################################################
# Step 1: Create IAM execution role
################################################################################

echo ""
echo "=========================================="
echo "Step 1: Create IAM execution role"
echo "=========================================="
echo ""
```

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

echo "Creating IAM role: ${ROLE_NAME}"
ROLE_OUTPUT=$(aws iam create-role \
    --role-name "$ROLE_NAME" \
    --assume-role-policy-document "$TRUST_POLICY" \
    --query 'Role.Arn' \
    --output text 2>&1)

if ! validate_input "$ROLE_OUTPUT" "^arn:aws:iam::[0-9]+:role/"; then
    echo "ERROR: Failed to create IAM role"
    exit 1
fi

echo "$ROLE_OUTPUT"
ROLE_ARN="$ROLE_OUTPUT"
CREATED_RESOURCES+=("iam-role:${ROLE_NAME}")
echo "Role ARN: ${ROLE_ARN}"

aws iam tag-role \
    --role-name "$ROLE_NAME" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=lambda-gettingstarted

echo ""
echo "Attaching AWSLambdaBasicExecutionRole policy..."
aws iam attach-role-policy \
    --role-name "$ROLE_NAME" \
    --policy-arn "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole" 2>&1
CREATED_RESOURCES+=("iam-role-policy:${ROLE_NAME}|arn:aws:iam::aws:policy/
service-role/AWSLambdaBasicExecutionRole")
echo "Policy attached."
```

```
# IAM roles can take a few seconds to propagate
echo "Waiting for IAM role to propagate..."
sleep 10

################################################################################
# Step 2: Create Lambda function
################################################################################

echo ""
echo "=========================================="
echo "Step 2: Create Lambda function"
echo "=========================================="
echo ""

echo "Creating deployment package..."
ORIGINAL_DIR=$(pwd)
cd "$TEMP_DIR" || exit 1
zip -j function.zip "$CODE_FILE" > /dev/null 2>&1 || {
    echo "ERROR: Failed to create deployment package"
    exit 1
}
cd "$ORIGINAL_DIR" || exit 1

if [ ! -f "${TEMP_DIR}/function.zip" ]; then
    echo "ERROR: Deployment package creation failed"
    exit 1
fi

echo "Creating Lambda function: ${FUNCTION_NAME}"
echo "  Runtime: ${RUNTIME}"
echo "  Handler: ${HANDLER}"
echo ""

CREATE_OUTPUT=$(aws lambda create-function \
    --function-name "$FUNCTION_NAME" \
    --runtime "$RUNTIME" \
    --role "$ROLE_ARN" \
    --handler "$HANDLER" \
    --architectures x86_64 \
    --zip-file "fileb://${TEMP_DIR}/function.zip" \
    --tags project=doc-smith,tutorial=lambda-gettingstarted \
    --query '[FunctionName, FunctionArn, Runtime, State]' \
    --output text 2>&1)
```

```
if [ -z "$CREATE_OUTPUT" ]; then
    echo "ERROR: Failed to create Lambda function"
    exit 1
fi

echo "$CREATE_OUTPUT"
CREATED_RESOURCES+=("lambda-function:${FUNCTION_NAME}")

wait_for_resource "Lambda function to become Active" \
    "aws lambda get-function-configuration --function-name ${FUNCTION_NAME} --
query State --output text" \
    "Active"

################################################################################
# Step 3: Invoke the function
################################################################################

echo ""
echo "=========================================="
echo "Step 3: Invoke the function"
echo "=========================================="
echo ""

TEST_EVENT='{"length": 6, "width": 7}'
echo "Invoking function with test event: ${TEST_EVENT}"
echo ""

echo "$TEST_EVENT" > "${TEMP_DIR}/test-event.json"

if ! validate_input "$TEST_EVENT" '"length": [0-9]+, "width": [0-9]+'; then
    echo "ERROR: Invalid test event format"
    exit 1
fi

INVOKE_OUTPUT=$(aws lambda invoke \
    --function-name "$FUNCTION_NAME" \
    --payload "fileb://${TEMP_DIR}/test-event.json" \
    --cli-read-timeout 30 \
    "${TEMP_DIR}/response.json" 2>&1)
echo "$INVOKE_OUTPUT"

if [ ! -f "${TEMP_DIR}/response.json" ]; then
    echo "ERROR: No response file generated"
    exit 1
```

```
fi

RESPONSE=$(cat "${TEMP_DIR}/response.json")
echo ""
echo "Function response: ${RESPONSE}"
echo ""

if echo "$INVOKE_OUTPUT" | grep -qi "functionerror"; then
    echo "WARNING: Function returned an error."
fi

################################################################################
# Step 4: View CloudWatch logs
################################################################################

echo ""
echo "============================================"
echo "Step 4: View CloudWatch Logs"
echo "============================================"
echo ""

echo "Log group: ${LOG_GROUP_NAME}"
echo ""

echo "Waiting for CloudWatch logs to be available..."

LOG_STREAMS=""
for i in $(seq 1 6); do
    LOG_STREAMS=$(aws logs describe-log-streams \
        --log-group-name "$LOG_GROUP_NAME" \
        --order-by LastEventTime \
        --descending \
        --query 'logStreams[0].logStreamName' \
        --output text 2>/dev/null) || true
    if [ -n "$LOG_STREAMS" ] && [ "$LOG_STREAMS" != "None" ]; then
        break
    fi
    LOG_STREAMS=""
    sleep 5
done

if [ -n "$LOG_STREAMS" ] && [ "$LOG_STREAMS" != "None" ]; then
    echo "Latest log stream: ${LOG_STREAMS}"
    echo ""
```

```
    echo "--- Log events ---"
    LOG_EVENTS=$(aws logs get-log-events \
        --log-group-name "$LOG_GROUP_NAME" \
        --log-stream-name "$LOG_STREAMS" \
        --query 'events[].message' \
        --output text 2>&1) || true
    echo "$LOG_EVENTS"
    echo "--- End of log events ---"
else
    echo "No log streams found yet. Logs may take a moment to appear."
    echo "You can view them in the CloudWatch console:"
    echo "  Log group: ${LOG_GROUP_NAME}"
fi

CREATED_RESOURCES+=("log-group:${LOG_GROUP_NAME}")

aws logs tag-log-group \
    --log-group-name "$LOG_GROUP_NAME" \
    --tags project=doc-smith,tutorial=lambda-gettingstarted

################################################################################
# Summary and cleanup
################################################################################

echo ""
echo "==========================================="
echo "SUMMARY"
echo "==========================================="
echo ""
echo "Resources created:"
echo "  IAM role:          ${ROLE_NAME}"
echo "  Lambda function:   ${FUNCTION_NAME}"
echo "  CloudWatch logs:   ${LOG_GROUP_NAME}"
echo ""
echo "==========================================="
echo "CLEANUP"
echo "==========================================="
echo ""
echo "Cleaning up all created resources..."
cleanup_resources

echo ""
echo "Done."
```

- Para obtener información sobre la API, consulte los siguientes temas en la Referencia de comandos de AWS CLI .
  - AttachRolePolicy
  - CreateFunction
  - CreateRole
  - DeleteFunction
  - DeleteLogGroup
  - DeleteRole
  - DescribeLogStreams
  - DetachRolePolicy
  - GetFunction
  - GetFunctionConfiguration
  - GetLogEvents
  - Invoke

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.

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.

.NET

SDK para .NET (v4)

> **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 flujo de trabajo principal que muestra el escenario de consultas de gran tamaño.

```
using System.Diagnostics;
using System.Text.RegularExpressions;
using Amazon.CloudFormation;
using Amazon.CloudFormation.Model;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
using CloudWatchLogsActions;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;

namespace CloudWatchLogsScenario;

public class LargeQueryWorkflow
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.
    This .NET code example performs the following tasks for the CloudWatch Logs
 Large Query workflow:

    1. Prepare the Application:
       - Prompt the user to deploy CloudFormation stack and generate sample logs.
       - Deploy the CloudFormation template for resource creation.
       - Generate 50,000 sample log entries using CloudWatch Logs API.
       - Wait 5 minutes for logs to be fully ingested.

    2. Execute Large Query:
       - Perform recursive queries to retrieve all logs using binary search.
       - Display progress for each query executed.
       - Show total execution time and logs found.

    3. Clean up:
       - Prompt the user to delete the CloudFormation stack and all resources.
       - Destroy the CloudFormation stack and wait until removed.
    */

    public static ILogger<LargeQueryWorkflow> _logger = null!;
    public static CloudWatchLogsWrapper _wrapper = null!;
    public static IAmazonCloudFormation _amazonCloudFormation = null!;

    private static string _logGroupName = "/workflows/cloudwatch-logs/large-
query";
```

```
    private static string _logStreamName = "stream1";
    private static long _queryStartDate;
    private static long _queryEndDate;

    public static bool _interactive = true;
    public static string _stackName = "CloudWatchLargeQueryStack";
    private static string _stackResourcePath = "../../../../../../../scenarios/
features/cloudwatch_logs_large_query/resources/stack.yaml";

    public static async Task Main(string[] args)
    {
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter("Microsoft", LogLevel.Information))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonCloudWatchLogs>()
                    .AddAWSService<IAmazonCloudFormation>()
                    .AddTransient<CloudWatchLogsWrapper>()
            )
            .Build();

        if (_interactive)
        {
            _logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
                .CreateLogger<LargeQueryWorkflow>();

            _wrapper = host.Services.GetRequiredService<CloudWatchLogsWrapper>();
            _amazonCloudFormation =
 host.Services.GetRequiredService<IAmazonCloudFormation>();
       }

       Console.WriteLine(new string('-', 80));
       Console.WriteLine("Welcome to the CloudWatch Logs Large Query
 Scenario.");
       Console.WriteLine(new string('-', 80));
       Console.WriteLine("This scenario demonstrates how to perform large-scale
 queries on");
       Console.WriteLine("CloudWatch Logs using recursive binary search to
 retrieve more than");
       Console.WriteLine("the 10,000 result limit.");
       Console.WriteLine();

       try
```

```
        {
            Console.WriteLine(new string('-', 80));
            var prepareSuccess = await PrepareApplication();
            Console.WriteLine(new string('-', 80));

            if (prepareSuccess)
            {
                Console.WriteLine(new string('-', 80));
                await ExecuteLargeQuery();
                Console.WriteLine(new string('-', 80));
            }

            Console.WriteLine(new string('-', 80));
            await Cleanup();
            Console.WriteLine(new string('-', 80));
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, "There was a problem with the scenario,
initiating cleanup...");
            _interactive = false;
            await Cleanup();
        }

        Console.WriteLine("CloudWatch Logs Large Query scenario completed.");
    }

    /// <summary>
    /// Runs the scenario workflow. Used for testing.
    /// </summary>
    public static async Task RunScenario()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the CloudWatch Logs Large Query
Scenario.");
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("This scenario demonstrates how to perform large-scale
queries on");
        Console.WriteLine("CloudWatch Logs using recursive binary search to
retrieve more than");
        Console.WriteLine("the 10,000 result limit.");
        Console.WriteLine();

        try
```

```
        {
            Console.WriteLine(new string('-', 80));
            var prepareSuccess = await PrepareApplication();
            Console.WriteLine(new string('-', 80));

            if (prepareSuccess)
            {
                Console.WriteLine(new string('-', 80));
                await ExecuteLargeQuery();
                Console.WriteLine(new string('-', 80));
            }

            Console.WriteLine(new string('-', 80));
            await Cleanup();
            Console.WriteLine(new string('-', 80));
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, "There was a problem with the scenario,
initiating cleanup...");
            _interactive = false;
            await Cleanup();
        }

        Console.WriteLine("CloudWatch Logs Large Query scenario completed.");
    }

    /// <summary>
    /// Prepares the application by creating the necessary resources.
    /// </summary>
    /// <returns>True if the application was prepared successfully.</returns>
    public static async Task<bool> PrepareApplication()
    {
        Console.WriteLine("Preparing the application...");
        Console.WriteLine();

        try
        {
            var deployStack = !_interactive || GetYesNoResponse(
                "Would you like to deploy the CloudFormation stack and generate
sample logs? (y/n) ");

            if (deployStack)
            {
```

```
                if (_interactive)
                {
                    Console.Write(
                        $"Enter a path for the CloudFormation stack
resource .yaml file (or press Enter for default '{_stackResourcePath}'): ");
                    string? inputPath = Console.ReadLine();
                    if (!string.IsNullOrWhiteSpace(inputPath))
                    {
                        _stackResourcePath = inputPath;
                    }
                }

                _stackName = PromptUserForStackName();

                var deploySuccess = await DeployCloudFormationStack(_stackName);

                if (deploySuccess)
                {
                    Console.WriteLine();
                    Console.WriteLine("Generating 50,000 sample log entries...");
                    var generateSuccess = await GenerateSampleLogs();

                    if (generateSuccess)
                    {
                        Console.WriteLine();
                        Console.WriteLine("Sample logs created. Waiting 5 minutes
for logs to be fully ingested...");
                        await WaitWithCountdown(300);

                        Console.WriteLine("Application preparation complete.");
                        return true;
                    }
                }
            }
            else
            {
                _logGroupName = PromptUserForInput("Enter the log group name ",
_logGroupName);
                _logStreamName = PromptUserForInput("Enter the log stream name ",
_logStreamName);

                var startDateMs = PromptUserForLong("Enter the query start date
(milliseconds since epoch): ");
```

```
                var endDateMs = PromptUserForLong("Enter the query end date
(milliseconds since epoch): ");

                _queryStartDate = startDateMs / 1000;
                _queryEndDate = endDateMs / 1000;

                Console.WriteLine("Application preparation complete.");
                return true;
            }
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, "An error occurred while preparing the
application.");
        }

        Console.WriteLine("Application preparation failed.");
        return false;
    }

    /// <summary>
    /// Deploys the CloudFormation stack with the necessary resources.
    /// </summary>
    /// <param name="stackName">The name of the CloudFormation stack.</param>
    /// <returns>True if the stack was deployed successfully.</returns>
    private static async Task<bool> DeployCloudFormationStack(string stackName)
    {
        Console.WriteLine($"\nDeploying CloudFormation stack: {stackName}");

        try
        {
            var request = new CreateStackRequest
            {
                StackName = stackName,
                TemplateBody = await File.ReadAllTextAsync(_stackResourcePath)
            };

            var response = await _amazonCloudFormation.CreateStackAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"CloudFormation stack creation started:
{stackName}");
```

```
                bool stackCreated = await
WaitForStackCompletion(response.StackId);

                if (stackCreated)
                {
                    Console.WriteLine("CloudFormation stack created
successfully.");
                    return true;
                }
                else
                {
                    _logger.LogError($"CloudFormation stack creation failed:
{stackName}");
                    return false;
                }
            }
            else
            {
                _logger.LogError($"Failed to create CloudFormation stack:
{stackName}");
                return false;
            }
        }
        catch (AlreadyExistsException)
        {
            _logger.LogWarning($"CloudFormation stack '{stackName}' already
exists. Please provide a unique name.");
            var newStackName = PromptUserForStackName();
            return await DeployCloudFormationStack(newStackName);
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"An error occurred while deploying the
CloudFormation stack: {stackName}");
            return false;
        }
    }

    /// <summary>
    /// Waits for the CloudFormation stack to be in the CREATE_COMPLETE state.
    /// </summary>
    /// <param name="stackId">The ID of the CloudFormation stack.</param>
    /// <returns>True if the stack was created successfully.</returns>
    private static async Task<bool> WaitForStackCompletion(string stackId)
```

```
    {
        int retryCount = 0;
        const int maxRetries = 30;
        const int retryDelay = 10000;

        while (retryCount < maxRetries)
        {
            var describeStacksRequest = new DescribeStacksRequest
            {
                StackName = stackId
            };

            var describeStacksResponse = await
_amazonCloudFormation.DescribeStacksAsync(describeStacksRequest);

            if (describeStacksResponse.Stacks.Count > 0)
            {
                if (describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.CREATE_COMPLETE)
                {
                    return true;
                }
                if (describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.CREATE_FAILED ||
                    describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.ROLLBACK_COMPLETE)
                {
                    return false;
                }
            }

            Console.WriteLine("Waiting for CloudFormation stack creation to
complete...");
            await Task.Delay(retryDelay);
            retryCount++;
        }

        _logger.LogError("Timed out waiting for CloudFormation stack creation to
complete.");
        return false;
    }

    /// <summary>
    /// Generates sample logs directly using CloudWatch Logs API.
```

```
    /// Creates 50,000 log entries spanning 5 minutes.
    /// </summary>
    /// <returns>True if logs were generated successfully.</returns>
    private static async Task<bool> GenerateSampleLogs()
    {
        const int totalEntries = 50000;
        const int entriesPerBatch = 10000;
        const int fiveMinutesMs = 5 * 60 * 1000;

        try
        {
            // Calculate timestamps
            var startTimeMs = DateTimeOffset.UtcNow.ToUnixTimeMilliseconds();
            var timestampIncrement = fiveMinutesMs / totalEntries;

            Console.WriteLine($"Generating {totalEntries} log entries...");

            var entryCount = 0;
            var currentTimestamp = startTimeMs;
            var numBatches = totalEntries / entriesPerBatch;

            // Generate and upload logs in batches
            for (int batchNum = 0; batchNum < numBatches; batchNum++)
            {
                var logEvents = new List<InputLogEvent>();

                for (int i = 0; i < entriesPerBatch; i++)
                {
                    logEvents.Add(new InputLogEvent
                    {
                        Timestamp =
DateTimeOffset.FromUnixTimeMilliseconds(currentTimestamp).UtcDateTime,
                        Message = $"Entry {entryCount}"
                    });

                    entryCount++;
                    currentTimestamp += timestampIncrement;
                }

                // Upload batch
                var success = await _wrapper.PutLogEventsAsync(_logGroupName,
_logStreamName, logEvents);
                if (!success)
                {
```

```
                    _logger.LogError($"Failed to upload batch {batchNum + 1}/
{numBatches}");
                    return false;
                }

                Console.WriteLine($"Uploaded batch {batchNum + 1}/{numBatches}");
            }

            // Set query date range (convert milliseconds to seconds for query
 API)
            _queryStartDate = startTimeMs / 1000;
            _queryEndDate = (currentTimestamp - timestampIncrement) / 1000;

            Console.WriteLine($"Query start date:
 {DateTimeOffset.FromUnixTimeSeconds(_queryStartDate):yyyy-MM-
ddTHH:mm:ss.fffZ}");
            Console.WriteLine($"Query end date:
 {DateTimeOffset.FromUnixTimeSeconds(_queryEndDate):yyyy-MM-ddTHH:mm:ss.fffZ}");
            Console.WriteLine($"Successfully uploaded {totalEntries} log
 entries");

            return true;
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, "An error occurred while generating sample
 logs.");
            return false;
        }
    }

    /// <summary>
    /// Executes the large query workflow.
    /// </summary>
    public static async Task ExecuteLargeQuery()
    {
        Console.WriteLine("Starting recursive query to retrieve all logs...");
        Console.WriteLine();

        var queryLimit = PromptUserForInteger("Enter the query limit (max 10000)
", 10000);
        if (queryLimit > 10000) queryLimit = 10000;

        var queryString = "fields @timestamp, @message | sort @timestamp asc";
```

```
        var stopwatch = Stopwatch.StartNew();
        var allResults = await PerformLargeQuery(_logGroupName, queryString,
_queryStartDate, _queryEndDate, queryLimit);
        stopwatch.Stop();

        Console.WriteLine();
        Console.WriteLine($"Queries finished in
{stopwatch.Elapsed.TotalSeconds:F3} seconds.");
        Console.WriteLine($"Total logs found: {allResults.Count}");

        // Check for duplicates
        Console.WriteLine();
        Console.WriteLine("Checking for duplicate logs...");
        var duplicates = FindDuplicateLogs(allResults);
        if (duplicates.Count > 0)
        {
            Console.WriteLine($"WARNING: Found {duplicates.Count} duplicate log
entries!");
            Console.WriteLine("Duplicate entries (showing first 10):");
            foreach (var dup in duplicates.Take(10))
            {
                Console.WriteLine($"  [{dup.Timestamp}] {dup.Message} (appears
{dup.Count} times)");
            }

            var uniqueCount = allResults.Count - duplicates.Sum(d => d.Count -
1);
            Console.WriteLine($"Unique logs: {uniqueCount}");
        }
        else
        {
            Console.WriteLine("No duplicates found. All logs are unique.");
        }
        Console.WriteLine();

        var viewSample = !_interactive || GetYesNoResponse("Would you like to see
a sample of the logs? (y/n) ");
        if (viewSample)
        {
            Console.WriteLine();
            Console.WriteLine($"Sample logs (first 10 of {allResults.Count}):");
            for (int i = 0; i < Math.Min(10, allResults.Count); i++)
            {
```

```
                var timestamp = allResults[i].Find(f => f.Field ==
 "@timestamp")?.Value ?? "N/A";
                var message = allResults[i].Find(f => f.Field ==
 "@message")?.Value ?? "N/A";
                Console.WriteLine($"[{timestamp}] {message}");
            }
        }
    }

    /// <summary>
    /// Performs a large query using recursive binary search.
    /// </summary>
    private static async Task<List<List<ResultField>>> PerformLargeQuery(
        string logGroupName,
        string queryString,
        long startTime,
        long endTime,
        int limit)
    {
        var queryId = await _wrapper.StartQueryAsync(logGroupName, queryString,
 startTime, endTime, limit);
        if (queryId == null)
        {
            return new List<List<ResultField>>();
        }

        var results = await PollQueryResults(queryId);
        if (results == null || results.Count == 0)
        {
            return new List<List<ResultField>>();
        }

        var startDate =
 DateTimeOffset.FromUnixTimeSeconds(startTime).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
        var endDate = DateTimeOffset.FromUnixTimeSeconds(endTime).ToString("yyyy-
MM-ddTHH:mm:ss.fffZ");
        Console.WriteLine($"Query date range: {startDate} ({startTime}s) to
 {endDate} ({endTime}s). Found {results.Count} logs.");

        if (results.Count < limit)
        {
            Console.WriteLine($"  -> Returning {results.Count} logs (less than
 limit of {limit})");
```

```
            return results;
        }

        Console.WriteLine($"  -> Hit limit of {limit}. Need to split and
 recurse.");

        // Get the timestamp of the last log (sorted to find the actual last one)
        var lastLogTimestamp = GetLastLogTimestamp(results);
        if (lastLogTimestamp == null)
        {
            Console.WriteLine($"  -> No timestamp found in results. Returning
 {results.Count} logs.");
            return results;
        }

        Console.WriteLine($"  -> Last log timestamp: {lastLogTimestamp}");

        // Parse the timestamp and add 1 millisecond to avoid querying the same
 log again
        var lastLogDate = DateTimeOffset.Parse(lastLogTimestamp + " +0000");
        Console.WriteLine($"  -> Last log as DateTimeOffset: {lastLogDate:yyyy-
MM-ddTHH:mm:ss.fffZ} ({lastLogDate.ToUnixTimeSeconds()}s)");

        var offsetLastLogDate = lastLogDate.AddMilliseconds(1);
        Console.WriteLine($"  -> Offset timestamp (last
 + 1ms): {offsetLastLogDate:yyyy-MM-ddTHH:mm:ss.fffZ}
 ({offsetLastLogDate.ToUnixTimeSeconds()}s)");

        // Convert to seconds, but round UP to the next second to avoid
 overlapping with logs in the same second
        // This ensures we don't re-query logs that share the same second as the
 last log
        var offsetLastLogTime = offsetLastLogDate.ToUnixTimeSeconds();
        if (offsetLastLogDate.Millisecond > 0)
        {
            offsetLastLogTime++; // Move to the next full second
            Console.WriteLine($"  -> Adjusted to next full second:
 {offsetLastLogTime}s
 ({DateTimeOffset.FromUnixTimeSeconds(offsetLastLogTime):yyyy-MM-
ddTHH:mm:ss.fffZ})");
        }

        Console.WriteLine($"  -> Comparing:
 offsetLastLogTime={offsetLastLogTime}s vs endTime={endTime}s");
```

```
        Console.WriteLine($"  -> End time as date:
 {DateTimeOffset.FromUnixTimeSeconds(endTime):yyyy-MM-ddTHH:mm:ss.fffZ}");

        // Check if there's any time range left to query
        if (offsetLastLogTime >= endTime)
        {
            Console.WriteLine($"  -> No time range left to query. Offset time
 ({offsetLastLogTime}s) >= end time ({endTime}s)");
            return results;
        }

        // Split the remaining date range in half
        var (range1Start, range1End, range2Start, range2End) =
 SplitDateRange(offsetLastLogTime, endTime);

        var range1StartDate =
 DateTimeOffset.FromUnixTimeSeconds(range1Start).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
        var range1EndDate =
 DateTimeOffset.FromUnixTimeSeconds(range1End).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
        var range2StartDate =
 DateTimeOffset.FromUnixTimeSeconds(range2Start).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");
        var range2EndDate =
 DateTimeOffset.FromUnixTimeSeconds(range2End).ToString("yyyy-MM-
ddTHH:mm:ss.fffZ");

        Console.WriteLine($"  -> Splitting remaining range:");
        Console.WriteLine($"     Range 1: {range1StartDate} ({range1Start}s) to
 {range1EndDate} ({range1End}s)");
        Console.WriteLine($"     Range 2: {range2StartDate} ({range2Start}s) to
 {range2EndDate} ({range2End}s)");

        // Query both halves recursively
        Console.WriteLine($"  -> Querying range 1...");
        var results1 = await PerformLargeQuery(logGroupName, queryString,
 range1Start, range1End, limit);
        Console.WriteLine($"  -> Range 1 returned {results1.Count} logs");

        Console.WriteLine($"  -> Querying range 2...");
        var results2 = await PerformLargeQuery(logGroupName, queryString,
 range2Start, range2End, limit);
        Console.WriteLine($"  -> Range 2 returned {results2.Count} logs");
```

```
        // Combine all results
        var allResults = new List<List<ResultField>>(results);
        allResults.AddRange(results1);
        allResults.AddRange(results2);

        Console.WriteLine($"  -> Combined total: {allResults.Count} logs
({results.Count} + {results1.Count} + {results2.Count})");

        return allResults;
    }

    /// <summary>
    /// Gets the timestamp string of the most recent log from a list of logs.
    /// Sorts timestamps to find the actual last one.
    /// </summary>
    private static string? GetLastLogTimestamp(List<List<ResultField>> logs)
    {
        var timestamps = logs
            .Select(log => log.Find(f => f.Field == "@timestamp")?.Value)
            .Where(t => !string.IsNullOrEmpty(t))
            .OrderBy(t => t)
            .ToList();

        if (timestamps.Count == 0)
        {
            return null;
        }

        return timestamps[timestamps.Count - 1];
    }

    /// <summary>
    /// Splits a date range in half.
    /// Range 2 starts at midpoint + 1 second to avoid overlap.
    /// </summary>
    private static (long range1Start, long range1End, long range2Start, long
range2End) SplitDateRange(long startTime, long endTime)
    {
        var midpoint = startTime + (endTime - startTime) / 2;
        // Range 2 starts at midpoint + 1 to avoid querying the same second twice
        return (startTime, midpoint, midpoint + 1, endTime);
    }
```

```
    /// <summary>
    /// Polls for query results until complete.
    /// </summary>
    private static async Task<List<List<ResultField>>?> PollQueryResults(string
queryId)
    {
        int retryCount = 0;
        const int maxRetries = 60;
        const int retryDelay = 1000;

        while (retryCount < maxRetries)
        {
            var response = await _wrapper.GetQueryResultsAsync(queryId);
            if (response == null)
            {
                return null;
            }

            if (response.Status == QueryStatus.Complete)
            {
                return response.Results;
            }

            if (response.Status == QueryStatus.Failed ||
                response.Status == QueryStatus.Cancelled ||
                response.Status == QueryStatus.Timeout ||
                response.Status == QueryStatus.Unknown)
            {
                _logger.LogError($"Query failed with status: {response.Status}");
                return null;
            }

            await Task.Delay(retryDelay);
            retryCount++;
        }

        _logger.LogError("Timed out waiting for query results.");
        return null;
    }

    /// <summary>
    /// Cleans up the resources created during the scenario.
    /// </summary>
    public static async Task<bool> Cleanup()
```

```
    {
        var cleanup = !_interactive || GetYesNoResponse(
            "Do you want to delete the CloudFormation stack and all resources?
(y/n) ");

        if (cleanup)
        {
            try
            {
                var stackDeleteSuccess = await
DeleteCloudFormationStack(_stackName, false);
                return stackDeleteSuccess;
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, "An error occurred while cleaning up the
resources.");
                return false;
            }
        }

        Console.WriteLine($"Resources will remain. Stack name: {_stackName}, Log
group: {_logGroupName}");
        _logger.LogInformation("CloudWatch Logs Large Query scenario is
complete.");
        return true;
    }

    /// <summary>
    /// Deletes the CloudFormation stack and waits for confirmation.
    /// </summary>
    private static async Task<bool> DeleteCloudFormationStack(string stackName,
bool forceDelete)
    {
        var request = new DeleteStackRequest
        {
            StackName = stackName,
        };

        if (forceDelete)
        {
            request.DeletionMode = DeletionMode.FORCE_DELETE_STACK;
        }
```

```
        await _amazonCloudFormation.DeleteStackAsync(request);
        Console.WriteLine($"CloudFormation stack '{stackName}' is being deleted.
This may take a few minutes.");

        bool stackDeleted = await WaitForStackDeletion(stackName, forceDelete);

        if (stackDeleted)
        {
            Console.WriteLine($"CloudFormation stack '{stackName}' has been
deleted.");
            return true;
        }
        else
        {
            _logger.LogError($"Failed to delete CloudFormation stack
'{stackName}'.");
            return false;
        }
    }

    /// <summary>
    /// Waits for the stack to be deleted.
    /// </summary>
    private static async Task<bool> WaitForStackDeletion(string stackName, bool
forceDelete)
    {
        int retryCount = 0;
        const int maxRetries = 30;
        const int retryDelay = 10000;

        while (retryCount < maxRetries)
        {
            var describeStacksRequest = new DescribeStacksRequest
            {
                StackName = stackName
            };

            try
            {
                var describeStacksResponse = await
_amazonCloudFormation.DescribeStacksAsync(describeStacksRequest);

                if (describeStacksResponse.Stacks.Count == 0 ||
```

```
                    describeStacksResponse.Stacks[0].StackStatus ==
StackStatus.DELETE_COMPLETE)
                {
                    return true;
                }

                if (!forceDelete && describeStacksResponse.Stacks[0].StackStatus
== StackStatus.DELETE_FAILED)
                {
                    return await DeleteCloudFormationStack(stackName, true);
                }
            }
            catch (AmazonCloudFormationException ex) when (ex.ErrorCode ==
"ValidationError")
            {
                return true;
            }

            Console.WriteLine($"Waiting for CloudFormation stack '{stackName}' to
be deleted...");
            await Task.Delay(retryDelay);
            retryCount++;
        }

        _logger.LogError($"Timed out waiting for CloudFormation stack
'{stackName}' to be deleted.");
        return false;
    }

    /// <summary>
    /// Waits with a countdown display.
    /// </summary>
    private static async Task WaitWithCountdown(int seconds)
    {
        for (int i = seconds; i > 0; i--)
        {
            Console.Write($"\rWaiting: {i} seconds remaining...  ");
            await Task.Delay(1000);
        }
        Console.WriteLine("\rWait complete.                    ");
    }

    /// <summary>
    /// Helper method to get a yes or no response from the user.
```

```
    /// </summary>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null && ynResponse.Equals("y",
StringComparison.InvariantCultureIgnoreCase);
        return response;
    }

    /// <summary>
    /// Prompts the user for a stack name.
    /// </summary>
    private static string PromptUserForStackName()
    {
        if (_interactive)
        {
            Console.Write($"Enter a name for the CloudFormation stack (press
Enter for default '{_stackName}'): ");
            string? input = Console.ReadLine();
            if (!string.IsNullOrWhiteSpace(input))
            {
                var regex = "[a-zA-Z][-a-zA-Z0-9]*";
                if (!Regex.IsMatch(input, regex))
                {
                    Console.WriteLine($"Invalid stack name. Using default:
{_stackName}");
                    return _stackName;
                }
                return input;
            }
        }
        return _stackName;
    }

    /// <summary>
    /// Prompts the user for input with a default value.
    /// </summary>
    private static string PromptUserForInput(string prompt, string defaultValue)
    {
        if (_interactive)
        {
            Console.Write($"{prompt}(press Enter for default '{defaultValue}'):
");
```

```
            string? input = Console.ReadLine();
            return string.IsNullOrWhiteSpace(input) ? defaultValue : input;
        }
        return defaultValue;
    }

    /// <summary>
    /// Prompts the user for an integer value.
    /// </summary>
    private static int PromptUserForInteger(string prompt, int defaultValue)
    {
        if (_interactive)
        {
            Console.Write($"{prompt}(press Enter for default '{defaultValue}'):
");
            string? input = Console.ReadLine();
            if (string.IsNullOrWhiteSpace(input) || !int.TryParse(input, out var
result))
            {
                return defaultValue;
            }
            return result;
        }
        return defaultValue;
    }

    /// <summary>
    /// Prompts the user for a long value.
    /// </summary>
    private static long PromptUserForLong(string prompt)
    {
        if (_interactive)
        {
            Console.Write(prompt);
            string? input = Console.ReadLine();
            if (long.TryParse(input, out var result))
            {
                return result;
            }
        }
        return 0;
    }

    /// <summary>
```

```
    /// Finds duplicate log entries based on timestamp and message.
    /// </summary>
    private static List<(string Timestamp, string Message, int Count)>
 FindDuplicateLogs(List<List<ResultField>> logs)
    {
        var logSignatures = new Dictionary<string, int>();

        foreach (var log in logs)
        {
            var timestamp = log.Find(f => f.Field == "@timestamp")?.Value ?? "";
            var message = log.Find(f => f.Field == "@message")?.Value ?? "";
            var signature = $"{timestamp}|{message}";

            if (logSignatures.ContainsKey(signature))
            {
                logSignatures[signature]++;
            }
            else
            {
                logSignatures[signature] = 1;
            }
        }

        return logSignatures
            .Where(kvp => kvp.Value > 1)
            .Select(kvp =>
            {
                var parts = kvp.Key.Split('|');
                return (Timestamp: parts[0], Message: parts[1], Count:
 kvp.Value);
            })
            .OrderByDescending(x => x.Count)
            .ToList();
    }
}
```

- Para obtener detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para .NET .
  - GetQueryResults
  - StartQuery

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(Number.parseInt(process.env.QUERY_START_DATE)),
    new Date(Number.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]);
}

/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}

/**
 * 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 [];
      }
      throw err;
    }
  }

  /**
   * 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;
    }
  }

  /**
   * 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 detalles sobre la API, consulte los siguientes temas en la Referencia de la API de AWS SDK para 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.

```
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",
)

DEFAULT_QUERY_LOG_GROUP = "/workflows/cloudwatch-logs/large-query"


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.
       Fetches the environment variable for log group, returning the default
value if it
       does not exist.

       :return: Tuple of query start date and end date as integers and the log
group.
       :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)

       try:
           log_group = os.environ["QUERY_LOG_GROUP"]
```

```
        except KeyError:
            logging.warning("No QUERY_LOG_GROUP environment variable, using
 default value")
            log_group = DEFAULT_QUERY_LOG_GROUP

        return query_start_date, query_end_date, log_group

    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",
        query="fields @timestamp, @message | sort @timestamp asc"
    ):
        """
        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
        :param query: Query string to pass to the CloudWatchQuery instance
        :type query: str
        """
        cloudwatch_query = CloudWatchQuery(
            log_group=log_group,
            query_string=query
        )
        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, log_group =
 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,
 log_group=log_group)


if __name__ == "__main__":
    main()
```

Este módulo procesa CloudWatch las consultas que superan los 10 000 resultados.

```
import logging
import time
```

```
from datetime import datetime
import threading
import boto3

from date_utilities import DateUtilities

DEFAULT_QUERY = "fields @timestamp, @message | sort @timestamp asc"
DEFAULT_LOG_GROUP = "/workflows/cloudwatch-logs/large-query"

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.

    :vartype date_range: tuple
    :ivar limit: Maximum number of log entries to return.
    :vartype limit: int
    :log_group str: Name of the log group to query
    :query_string str: query
    """

    def __init__(self, log_group: str = DEFAULT_LOG_GROUP, query_string:
 str=DEFAULT_QUERY) -> None:
        self.lock = threading.Lock()
        self.log_group = log_group
        self.query_string = query_string
        self.query_results = []
        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}"
            f"\n       LOG GROUP: {self.log_group}"
        )
        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
```

```
    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_group,
                    startTime=start_time,
                    endTime=end_time,
                    queryString=self.query_string,
                    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_group,
               startTime=start_time,
               endTime=end_time,
               queryString=self.query_string,
               limit=max_logs,
           )
           return response["queryId"]
       except client.exceptions.ResourceNotFoundException as e:
           raise DateOutOfBoundsError(f"Resource not found: {e}")


   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 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.

Uso de eventos programados para invocar una función de Lambda

Los siguientes ejemplos de código muestran cómo crear una AWS Lambda función invocada por un evento EventBridge programado de Amazon.

Java

SDK para Java 2.x

Muestra cómo crear un evento EventBridge programado de Amazon que invoque una AWS Lambda función. Configure EventBridge para usar una expresión cron para programar cuándo se invoca la función Lambda. En este ejemplo, creará una función de Lambda utilizando la API de tiempo de ejecución de Java de Lambda. En este ejemplo, se invocan diferentes AWS servicios para realizar un caso de uso específico. Este ejemplo indica cómo crear una aplicación que envíe un mensaje de texto a sus empleados para felicitarles por su primer aniversario.

Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- CloudWatch Registros
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

JavaScript

SDK para JavaScript (v3)

Muestra cómo crear un evento EventBridge programado de Amazon que invoque una AWS Lambda función. Configure EventBridge para usar una expresión cron para programar cuándo se invoca la función Lambda. En este ejemplo, se crea una función Lambda mediante la API de tiempo de ejecución de JavaScript Lambda. En este ejemplo, se invocan diferentes AWS servicios para realizar un caso de uso específico. Este ejemplo indica cómo crear una aplicación que envíe un mensaje de texto a sus empleados para felicitarles por su primer aniversario.

Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en GitHub.

Este ejemplo también está disponible en la Guía para desarrolladores de AWS SDK para JavaScript v3.

Servicios utilizados en este ejemplo

- CloudWatch Registros
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

Python

SDK para Python (Boto3)

En este ejemplo se muestra cómo registrar una AWS Lambda función como destino de un EventBridge evento programado de Amazon. El controlador Lambda escribe un mensaje descriptivo y los datos completos del evento en Amazon CloudWatch Logs para su posterior recuperación.

- Implementa una función de Lambda.
- Crea un evento EventBridge programado y convierte la función Lambda en el objetivo.
- Otorga permiso para EventBridge invocar la función Lambda.
- Imprime los datos más recientes de CloudWatch los registros para mostrar el resultado de las invocaciones programadas.
- Limpia todos los recursos creados durante la demostración.

Es mejor ver este ejemplo en GitHub. Para obtener el código fuente completo y las instrucciones sobre cómo configurarlo y ejecutarlo, consulte el ejemplo completo en GitHub.

Servicios utilizados en este ejemplo

- CloudWatch Registros
- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

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.

Cifre las tablas de búsqueda en CloudWatch los registros mediante AWS Key Management Service

Los datos de las tablas de consulta siempre se cifran en CloudWatch los registros. De forma predeterminada, CloudWatch Logs utiliza el cifrado del lado del servidor con el Galois/Counter modo estándar de cifrado avanzado (AES-GCM) de 256 bits para cifrar los datos de las tablas de consulta en reposo. Como alternativa, puede utilizar AWS Key Management Service para este cifrado. Si lo hace, el cifrado se realiza mediante una clave. AWS KMS El uso del cifrado AWS KMS se habilita en el nivel de la tabla de consulta, asociando una clave de KMS a una tabla de consulta, ya sea al crear la tabla de consulta o al actualizarla.

> **Important**
>
> CloudWatch Los registros solo admiten claves KMS simétricas. No utilice una clave asimétrica para cifrar los datos de las tablas de búsqueda. Para obtener más información, consulte Utilización de claves simétricas y asimétricas.

Tras asociar una clave KMS a una tabla de consulta, todos los datos almacenados en la tabla de consulta se cifran con esta clave. CloudWatch Logs descifra estos datos siempre 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 una tabla de consulta, CloudWatch Logs cifra los datos mediante el método de cifrado predeterminado de CloudWatch Logs. Sin embargo, si la clave se deshabilita o se elimina antes de desasociarla, CloudWatch Logs no podrá leer los datos que se cifraron con esa clave.

Para obtener información general sobre cómo utiliza CloudWatch Logs AWS KMS para cifrar los datos de registro, consulte. Cifre los datos de registro en los CloudWatch registros mediante AWS Key Management Service

Cómo utiliza CloudWatch Logs AWS KMS para las tablas de búsqueda

CloudWatch Logs utiliza el cifrado de AWS KMS sobres para proteger los datos de las tablas de consulta. Al asociar una clave KMS a una tabla de búsqueda, CloudWatch Logs envía una `GenerateDataKey` solicitud a AWS KMS. AWS KMS genera una clave de cifrado de datos (DEK) única y devuelve una copia de texto simple y una copia cifrada de la DEK. CloudWatch Logs utiliza la DEK de texto plano para cifrar los datos de la tabla de consulta y, a continuación, almacena la DEK cifrada junto con los datos cifrados. La DEK de texto simple no se almacena y se descarta de la memoria después de usarla.

Cuando CloudWatch Logs necesita leer los datos de la tabla de consulta, envía una `Decrypt` solicitud a AWS KMS la DEK cifrada. AWS KMS descifra el DEK y devuelve el DEK en texto plano a CloudWatch Logs, que luego lo utiliza para descifrar los datos de la tabla de consulta.

CloudWatch Logs utiliza el siguiente contexto de cifrado al realizar solicitudes a: AWS KMS

```
{
    "aws:logs:arn": "arn:aws:logs:region:account-id:lookup-table:lookup-table-name"
}
```

Puede utilizar este contexto de cifrado en las políticas de IAM y las políticas AWS KMS clave para controlar el acceso a la clave de KMS. Para obtener más información, consulte AWS KMS claves y contexto de cifrado.

Permisos necesarios

Para utilizar el AWS KMS cifrado con tablas de consulta, el principal de IAM debe tener los siguientes AWS KMS permisos en la clave de KMS:

- `kms:Decrypt`
- `kms:GenerateDataKey`

El `kms:Decrypt` permiso es necesario para acceder a `GetLookupTable` una tabla de consulta cifrada con una clave KMS, de modo que CloudWatch Logs pueda descifrar los datos en su nombre. También se requiere el `kms:Decrypt` permiso de la clave (la clave de KMS utilizada para cifrar la

tabla de consulta) cuando se llama `StartQuery` con una consulta que utiliza el `lookup` comando de una tabla de consulta cifrada. El `kms:GenerateDataKey` permiso es necesario cuando se llama `CreateLookupTable` o `UpdateLookupTable` con una clave KMS, de modo que CloudWatch Logs pueda generar una clave de cifrado de datos para cifrar los datos de la tabla de consulta.

Además, el servicio de CloudWatch registros debe tener permiso para usar la clave KMS. Para conceder estos permisos, añada una declaración de política a la política de claves de KMS, tal y como se describe en la siguiente sección.

Paso 1: Crear una AWS KMS clave

Para crear una clave KMS simétrica, 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"
        ]
    }
}
```

Registre el ARN clave. La necesitará en los siguientes pasos.

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. Debe conceder al servicio de CloudWatch registros el permiso principal para usar la clave y también el permiso al rol que realiza la llamada para usar la clave.

En primer lugar, guarde la política predeterminada para su clave de 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 `policy.json` archivo en un editor de texto y añada la siguiente declaración para conceder al servicio de CloudWatch registros el permiso principal para usar la clave. En este ejemplo, se utiliza una `Condition` sección que coincide con el contexto de cifrado para restringir la clave a una tabla de consulta específica.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logs.region.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:lookup-table:lookup-table-name",
            "aws:SourceAccount": "account-id",
            "aws:SourceArn": "arn:aws:logs:region:account-id:lookup-table:lookup-table-
name"
        }
    }
}
```

A continuación, añada permisos al rol que llamará a los CloudWatch Logs `CreateLookupTable` o `UpdateLookupTable` APIs. CloudWatch Los registros `kms:ViaService` se utilizan para realizar llamadas AWS KMS en nombre del cliente. Para obtener más información, consulte kms: ViaService.

```
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::account-id:role/role-name"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": [
                "logs.region.amazonaws.com"
            ]
        }
    }
}
```

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 una tabla de consulta

Puede asociar una clave de KMS a una tabla de consulta al crearla mediante la `CreateLookupTable` API o actualizar una tabla de consulta existente mediante la `UpdateLookupTable` API. Ambas APIs forman parte de AWSLogsConfigService.

Para asociar la clave KMS a una tabla de consulta al crearla

Utilice la `CreateLookupTable` API y especifique el `kmsKeyArn` parámetro con el ARN de la clave KMS:

```
aws logs create-lookup-table \
```

```
    --lookup-table-name my-lookup-table \
    --kms-key-arn "arn:aws:kms:region:account-id:key/key-id"
```

Para asociar la clave KMS a una tabla de consulta existente

Utilice la `UpdateLookupTable` API y especifique el `kmsKeyArn` parámetro con el ARN de la clave KMS:

```
aws logs update-lookup-table \
    --lookup-table-name my-lookup-table \
    --kms-key-arn "arn:aws:kms:region:account-id:key/key-id"
```

Consideraciones

- Tras asociar o desasociar una clave KMS de una tabla de consulta, 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 de la tabla de búsqueda cifrados de los CloudWatch registros ya no se podrán recuperar.
- Para realizar los pasos de este tema, debe tener los siguientes permisos:`kms:CreateKey`, `kms:GetKeyPolicykms:PutKeyPolicy`, y los permisos de CloudWatch Logs adecuados para llamar a `CreateLookupTable` o`UpdateLookupTable`.

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 la 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. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre privacidad de datos y los . Para obtener más información sobre la protección de datos en Europa, consulte el Centro del Reglamento General de Protección de Datos (RGPD).

Para fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.

- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con CloudWatch registros u otros usos de la Servicios de AWS consola, la API o los SDK. AWS CLI AWS Cualquier dato que introduzca en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya 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 el cifrado del lado del servidor y utiliza el cifrado del lado del servidor con el Galois/Counter modo estándar de cifrado avanzado de 256 bits (AES-GCM) para cifrar los datos de registro en reposo.

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 cifrado de extremo a extremo 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 gestionados 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 agregue 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 gestionados 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 agregue 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
- Titularidad de los recursos
- Administración del acceso a los recursos
- Especificar elementos de la política: acciones, efectos y entidades principales
- Especificación de las condiciones de una política

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 registro	arn:aws:logs: ::log-group: *region account-id log_group_name*
Flujo de registro	arn:aws:logs: ::log-group *region* ::log-stream: *account-id log_group_name log-stream-name*
Destino	arn:aws:logs: *region* ::destination: *account-id destination_name*

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 entre cuentas y regiones.

Temas

- Permisos de grupo de registro y Información de colaboradores
- Resource-based políticas

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, consulte Uso de Contributor Insights para analizar High-Cardinality datos.

Al conceder a un usuario los `cloudwatch:GetInsightRuleReport` permisos `cloudwatch:PutInsightRule` y, dicho usuario puede crear una regla que evalúe cualquier grupo de CloudWatch registros en Logs 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.

Resource-based políticas

CloudWatch Logs admite políticas basadas en recursos para los destinos, que puede utilizar 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 a otra cuenta de AWS con el ID de cuenta 111122223333 suscribir los grupos de registro al destino `arn:aws:logs:us-east-1:123456789012:destination:testDestination`.

JSON

```
{
  "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 de CloudWatch 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:

JSON

```
{
  "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 registro 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 (*) del `Resource` valor otorga permiso para realizar las acciones enumeradas en cualquier recurso de CloudWatch Logs.

- Para limitar los permisos a acciones de grupos de registros específicos (por ejemplo,`DescribeLogStreams`)`CreateLogGroup`, sustituya el comodín por el ARN del grupo de registros: `arn:aws:logs:`*`us-west-2`*`:`*`123456789012`*`:log-group:`*`MyLogGroup`*
- Para limitar los permisos a acciones de flujo de registro específicas (por ejemplo`CreateLogStream`,`PutLogEvents`), sustituya el comodín por el ARN del flujo de registro (coincide con todos los flujos) `arn:aws:logs:`*`us-west-2`*`:`*`123456789012`*`:log-group:`*`MyLogGroup`*`:log-stream:`*`MyStream`* o (flujo específico) `arn:aws:logs:`*`us-west-2`*`:`*`123456789012`*`:log-group:`*`MyLogGroup`*`:*`

Consulta la referencia de autorización del servicio para ver qué tipo de recurso requiere cada API.

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 los `cloudwatch:GenerateQueryResultsSummary` permisos

`cloudwatch:GenerateQuery` y, por lo tanto, los usuarios con esta política pueden generar una cadena de consulta de CloudWatch Logs Insights a partir de un mensaje en lenguaje natural. Para ver el contenido completo de la política, consulte la Guía CloudWatchLogsFullAccessde referencia de políticas AWS gestionadas.

CloudWatchLogsReadOnlyAccess

La CloudWatchLogsReadOnlyAccesspolítica otorga acceso de solo lectura a los CloudWatch registros. Incluye los `cloudwatch:GenerateQueryResultsSummary` permisos `cloudwatch:GenerateQuery` y, de este modo, los usuarios con esta política pueden generar una cadena de consulta de CloudWatch Logs Insights a partir de un mensaje en lenguaje natural. Para ver el contenido completo de la política, consulte la Guía CloudWatchLogsReadOnlyAccessde referencia de políticas AWS gestionadas.

CloudWatchOpenSearchDashboardsFullAccess

La CloudWatchOpenSearchDashboardsFullAccesspolítica otorga acceso para crear, administrar y eliminar integraciones con el OpenSearch Servicio, y para crear paneles de control de registros vendidos en esas integraciones. Para obtener más información, consulte Analice con Amazon OpenSearch Service.

Para ver el contenido completo de la política, consulte la Guía de referencia de políticas CloudWatchOpenSearchDashboardsFullAccess AWSgestionadas.

CloudWatchOpenSearchDashboardAccess

La CloudWatchOpenSearchDashboardAccesspolítica otorga acceso para ver los paneles de control de registros vendidos que se crean con Amazon OpenSearch Service análisis. Para obtener más información, consulte Analice con Amazon OpenSearch Service.

> **Important**
>
> Además de conceder esta política, para permitir que un rol o un usuario puedan ver los paneles de registros vendidos, también debe especificarlos al crear la integración con el Servicio. OpenSearch Para obtener más información, consulte Paso 1: Crear la integración con OpenSearch Service.

Para ver el contenido completo de la política, consulte la Guía CloudWatchOpenSearchDashboardAccessde referencia de políticas AWS gestionadas.

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, consulte Observabilidad entre cuentas de CloudWatch .

Para ver el contenido completo de la política, consulta la Guía CloudWatchLogsCrossAccountSharingConfigurationde referencia de políticas AWS gestionadas.

CloudWatchLogsAPIKeyAccess

La CloudWatchLogsAPIKeyAccesspolítica permite la autenticación de claves de la API de CloudWatch registros y la ingesta de registros cifrados. Esta política otorga permisos para autenticarse mediante tokens portadores y escribir eventos de registro en los CloudWatch registros, además de AWS KMS permisos adicionales para descifrar y generar claves de datos cuando los registros están cifrados.

Esta política le concede los siguientes permisos:

- `logs`— Permite a los directores autenticarse mediante identificadores portadores de claves de la API y escribir los eventos de registro en las secuencias de Logs. CloudWatch
- `kms`— Permite a los directores leer los metadatos AWS KMS clave, generar claves de datos para el cifrado y descifrar los datos. Estos permisos admiten CloudWatch registros cifrados, lo que permite al servicio cifrar los datos de registro mediante claves administradas por el cliente. AWS KMS El acceso está restringido a las operaciones realizadas a través del CloudWatch servicio de registros.

Para ver más detalles sobre la política, incluyendo la última versión del documento de política JSON, consulte CloudWatchLogsAPIKeyAccess en la Guía de referencia de políticas administradas de AWS .

CloudWatch Registra las actualizaciones de AWS políticas administradas

Consulta los detalles sobre las actualizaciones de las políticas AWS gestionadas para CloudWatch los registros desde que este servicio comenzó a realizar el seguimiento de 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
CloudWatchLogsAPIKeyAccess: Política nueva.	CloudWatch Los registros agregaron una nueva política gestionada CloudWatchLogsAPIKeyAccess. Esta política permite la autenticación de claves de la API de CloudWatch registros y la ingesta de registros cifrados, lo que otorga permisos para autenticarse mediante tokens portadores y escribir eventos de registro en los registros. CloudWatch	17 de febrero de 2026
CloudWatchLogsFullAccess: actualización de una política existente.	CloudWatch Registra los permisos añadidos a CloudWatchLogsFullAccess. Se agregaron permisos para las acciones de administración de observabilidad a fin de permitir el acceso de solo lectura a las canalizaciones de telemetría y a las integraciones de tablas de S3.	2 de diciembre de 2025
CloudWatchLogsReadOnlyAccess: actualización de una política existente.	CloudWatch Registra los permisos añadidos a CloudWatchLogsReadOnlyAccess. Se agregaron permisos para las acciones de administr	2 de diciembre de 2025

Cambio	Descripción	Fecha
	ación de observabilidad a fin de permitir el acceso de solo lectura a las canalizaciones de telemetría y a las integraciones de tablas de S3.	
CloudWatchLogsFullAccess: actualización de una política existente.	CloudWatch Registra los permisos añadidos a CloudWatchLogsFullAccess. Se agregaron permisos para `cloudwatch:GenerateQueryResultsSummary` de manera que se permite la generación de un resumen en lenguaje natural de los resultados de la consulta.	20 de mayo de 2025
CloudWatchLogsReadOnlyAccess: actualización de una política existente.	CloudWatch Registra los permisos agregados a CloudWatchLogsReadOnlyAccess. Se agregaron permisos para `cloudwatch:GenerateQueryResultsSummary` de manera que se permite la generación de un resumen en lenguaje natural de los resultados de la consulta.	20 de mayo de 2025

Cambio	Descripción	Fecha
CloudWatchLogsFullAccess: actualización de una política existente.	CloudWatch Registra los permisos agregados a CloudWatchLogsFullAccess. Se agregaron permisos para el IAM Amazon OpenSearch Service y para permitir la integración de CloudWatch Logs con el OpenSearch servicio para algunas funciones.	1 de diciembre de 2024
CloudWatchOpenSearchDashboardsFullAccess—Nueva política de IAM.	CloudWatch Logs agregó una nueva política de IAM, CloudWatchOpenSearchDashboardsFullAccess.- Esta política otorga acceso para crear, administrar y eliminar integraciones con el OpenSearch Servicio, y para crear, administrar y eliminar paneles de registros vendidos en esas integraciones. Para obtener más información, consulte Analice con Amazon OpenSearch Service.	1 de diciembre de 2024

Cambio	Descripción	Fecha
CloudWatchOpenSearchDashboardAccess— Nueva política de IAM.	CloudWatch Logs agregó una nueva política de IAM, CloudWatchOpenSearchDashboardAccess.- Esta política otorga acceso a los paneles de control de registros vendidos con la tecnología de. Amazon OpenSearch Service Para obtener más información, consulte Analice con Amazon OpenSearch Service.	1 de diciembre de 2024
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

Cambio	Descripción	Fecha
CloudWatchLogsReadOnlyAccess: actualización de una política existente.	CloudWatch agregó un permiso para CloudWatchLogsReadOnlyAccess. 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 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

Cambio	Descripción	Fecha
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
CloudWatchLogsReadOnlyAccess: 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 registros o a un flujo de registros

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.

JSON

```
{
  "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.

JSON

```
{
    "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 registros o a un flujo de registros

CloudWatch Los registros tienen dos tipos de recursos con diferentes formatos de ARN:

- Grupo de registros: `arn:aws:logs:`*`region`*`:`*`account`*`:log-group:`*`LogGroupName`*
- Flujo de registro: `arn:aws:logs:`*`region`*`:`*`account`*`:log-group:`*`LogGroupName`*`:log-stream:`*`StreamName`*

Al escribir políticas de IAM, el formato ARN que utilices debe coincidir con el tipo de recurso al que se autoriza la API. Consulta la referencia de autorización del servicio para ver qué tipo de recurso requiere cada API.

Ejemplo 3a: permitir el acceso a acciones a nivel de grupo de registros en un grupo de registros específico

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:DeleteLogGroup",
        "logs:PutRetentionPolicy",
```

```
        "logs:PutSubscriptionFilter",
        "logs:DescribeLogStreams"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName"
      }
   ]
}
```

Estas acciones autorizan el tipo de recurso. `log-group` Se admite el formato ARN estándar (sin él`:*`).

Ejemplo 3b: permitir el acceso a acciones a nivel de flujo de registro en un grupo de registros específico

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName:*"
      }
   ]
}
```

Estas acciones autorizan el tipo de recurso. `log-stream` El `:*` sufijo es necesario para hacer coincidir todos los flujos de registro del grupo de registros o para especificar un flujo específico con `:log-stream:`*`StreamName`* él.

Ejemplo 3c: Política combinada para las acciones de grupos de registros y de flujo de registros

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

```
          "logs:PutRetentionPolicy",
          "logs:DescribeLogStreams",
          "logs:FilterLogEvents"
        ],
        "Effect": "Allow",
        "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName"
      },
      {
        "Action": [
          "logs:CreateLogStream",
          "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 registro al mismo tiempo que les impide tener acceso a otros grupos de registro. 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 registro, 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 registro que tienen el valor `Team` para la clave de etiqueta `Green`.

JSON

```
{
    "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 los AWS 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.

JSON

```
{
    "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 registro a un bucket de Amazon S3.
CreateLogGroup	`logs:CreateLogGroup`

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
	Necesario para crear un nuevo grupo de registro.
CreateLogStream	`logs:CreateLogStream` Necesario para crear un nuevo flujo de registros en un grupo de registro.
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 registro 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 registro.
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 registro.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Necesario para eliminar el filtro de suscripción asociado a un grupo de registro.
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 registro.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
DescribeMetricFilters	`logs:DescribeMetricFilters` Necesario para ver todas las métricas asociadas a un grupo de registro.
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 las políticas de recursos de CloudWatch Logs.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Necesario para ver todos los filtros de suscripci ón asociados con un grupo de registro.
FilterLogEvents	`logs:FilterLogEvents` Necesario para ordenar los eventos de registros por patrón de filtro de grupo de registro.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
GetLogEvents	`logs:GetLogEvents` Necesario para recuperar eventos de registro de un flujo de registros.
GetLogGroupFields	`logs:GetLogGroupFields` Necesario para recuperar la lista de campos que se incluyen en los eventos de registro de un grupo de registro.
GetLogRecord	`logs:GetLogRecord` Necesario para recuperar los detalles de un único evento de registro.
GetLogObject	`logs:GetLogRecord` Necesario para obtener el contenido de una gran parte de los eventos de registro que se han ingerido a través de la PutOpenTelemetryLogs API.
GetQueryResults	`logs:GetQueryResults` Necesario para recuperar los resultados de las consultas de CloudWatch Logs Insights.
ListEntitiesForLogGroup (permiso CloudWatch solo para consola)	`logs:ListEntitiesForLogGroup` Obligatorio para buscar las entidades asociadas a un grupo de registro. Necesario para explorar los registros relacionados en la CloudWatch consola.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
ListLogGroupsForEntity (permiso CloudWatch solo para consola)	`logs:ListLogGroupsForEntity` Obligatorio para buscar los grupos de registros asociados a una entidad. Necesario para explorar los registros relacionados en la CloudWatch consola.
ListTagsLogGroup	`logs:ListTagsLogGroup` Necesario para ver las etiquetas asociadas a un grupo de registro.
ListLogGroups	`logs:DescribeLogGroups` Necesario para ver todos los grupos de registro asociados a la cuenta.
ProcessWithPipeline (solo con permiso)	`logs:ProcessWithPipeline` Necesario para procesar y transformar los eventos del registro mediante transform adores de canalización antes de almacenarlos. Este permiso debe concederse al rol de IAM transferido como rol de origen de CloudWatch registros en una configuración de canalización. El recurso es el ARN del grupo de registros.
PutDestination	`logs:PutDestination` Necesario para crear o actualizar un flujo de registros de destino (como, por ejemplo, un flujo de Kinesis).

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
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.
PutMetricFilter	`logs:PutMetricFilter` Necesario para crear o actualizar un filtro de métricas y asociarlo a un grupo de registro.
PutQueryDefinition	`logs:PutQueryDefinition` Necesario para guardar una consulta en CloudWatch Logs Insights, incluidas las consultas guardadas con parámetros.
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 registro.

CloudWatch Registra las operaciones de la API	Permisos necesarios (acciones de API)
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Necesario para crear o actualizar un filtro de suscripción y asociarlo a un grupo de registro.
StartQuery	`logs:StartQuery` Necesario para iniciar las consultas CloudWatch de Logs Insights. Para ejecutar una consulta guardada con parámetros, también necesita `logs:DescribeQueryDefinitions` .
StopQuery	`logs:StopQuery` Necesario para detener una consulta de CloudWatch Logs Insights que está en curso.
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 Service-linked Los roles 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 que tengan la palabra Sí en la columna Service-LinkedRol. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Service-linked permisos de rol para CloudWatch los registros

CloudWatch Logs usa el rol vinculado al servicio denominado. AWSServiceRoleForLogDelivery CloudWatch Logs usa esta función vinculada al servicio para escribir registros directamente en Firehose. Para obtener más información, consulte Habilitar el registro desde AWS servicios.

El rol vinculado al servicio AWSServiceRoleForLogDelivery confía en los siguientes servicios para asumir el rol:

- `logs.amazonaws.com`

La política de permisos del rol permite a CloudWatch Logs realizar las siguientes acciones en los recursos especificados. AWSServiceRoleForLogDeliveryPolicy Para ver el documento de política de JSON completo, consulte AWSServiceRoleForLogDeliveryPolicyla Guía de referencia de políticas AWS gestionadas.

- Acción: `firehose:PutRecordfirehose:PutRecordBatch`, y `firehose:ListTagsForDeliveryStream` en todas las transmisiones de entrega de Firehose que tengan una etiqueta con una `LogDeliveryEnabled` clave con un valor de. `true` Esta etiqueta se adjunta automáticamente a una transmisión de entrega de Firehose al crear una suscripción para entregar los troncos a Firehose.
- Acción: `kms:GenerateDataKey` y `kms:Decrypt` en todas AWS KMS las claves, pero solo cuando la solicitud se realiza a través de Firehose (se aplica mediante la clave de

`kms:ViaService` condición establecida en). `firehose.*.amazonaws.com` Estos permisos permiten a Logs entregar CloudWatch registros a las transmisiones de entrega de Firehose que utilizan cifrado del lado del servidor con claves administradas por el cliente (). SSE-CMK La política AWS KMS clave del cliente también debe conceder de forma independiente el acceso a la función vinculada al servicio.

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 los permisos de Service-Linked rol en la Guía del usuario de IAM.

Crear un rol vinculado a un servicio para los registros 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 Consola de administración de AWS, 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 Edición de un Service-Linked rol en la Guía del usuario de IAM.

Eliminar un rol vinculado a un servicio para los registros 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 supervise 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 AWSServiceRoleForLogDelivery rol vinculado al servicio

- Deje de enviar registros directamente a los flujos de Firehose.

Para eliminar manualmente el 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 Service-Linked

Regiones compatibles con las funciones CloudWatch vinculadas al servicio de registros

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.

CloudWatch Registra las actualizaciones de AWS roles vinculados a servicios

Consulte los detalles sobre las actualizaciones de la función vinculada al AWS servicio de CloudWatch Logs 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	CloudWatch Registros agregados `kms:GenerateDataKey` y `kms:Decrypt` permisos a la política de	15 de mayo de 2026

Cambio	Descripción	Fecha
servicio: se actualiza a una política existente	IAM asociados al rol vinculado al AWSServiceRoleForL ogDeliveryservicio. Estos permisos están sujetos a la clave de `kms:ViaService` condición que solo permite su uso a través de Firehose. Este cambio permite a Logs entregar CloudWatch registros a las transmisiones de entrega de Firehose que utilizan cifrado del lado del servidor con claves administradas por el cliente (). SSE-CMK	
AWSServiceRoleForL ogDelivery política de funciones vinculadas al servicio: actualización de una política existente	CloudWatch Los registros cambiaron los permisos de la política de IAM asociados al rol vinculado al AWSServic eRoleForLogDeliveryservicio. Se realizó el siguiente cambio: • La clave de condición `firehose:ResourceT ag/LogDeliveryEnab led": "true"` se cambió a `aws:ResourceTag/Lo gDeliveryEnabled": "true"`.	15 de julio de 2021

Cambio	Descripción	Fecha
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

Validación de conformidad para Amazon CloudWatch Logs

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

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

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. Para obtener más información sobre su responsabilidad de conformidad al utilizarlos Servicios de AWS, consulte AWS la documentación de seguridad.

Resiliencia en Amazon CloudWatch Logs

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. 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 demora. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

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

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 Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a 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.

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. CloudWatch Logs admite puntos de enlace de IPv4 VPC en todas las regiones y admite puntos de IPv6 enlace en todas las regiones.

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. ***Region***.logs. Para conectarse con un punto final FIPS, el servicio que debe elegir es. `com.amazonaws.Region.logs-fips` No necesita cambiar ninguna configuración de los CloudWatch registros. Para obtener más información, consulte Creación de un punto de conexión de tipo interfaz en la Guía del usuario de Amazon VPC.

Algunos CloudWatch registros APIs, como StartLiveTail y GetLogObject, se alojan en un punto final diferente y en un punto final de VPC diferentes:. `stream-logs.Region.amazonaws.com` Para crear un punto final de VPC de interfaz para estos APIs, el servicio que debe elegir es com.amazonaws. ***Region***.stream-logs. Para conectarse con un punto final FIPS, el servicio que debe elegir es. `com.amazonaws.Region.stream-logs-fips`

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 final 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, selecciona com.amazonaws. *Region*.logs y selecciona Crear punto final.
4. Selecciona com.amazonaws. ***Region***El punto de enlace .logs y selecciona la pestaña 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 puntos finales de VPC específicos VPCs o 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 CloudTrailun servicio que proporciona un registro de las acciones realizadas por un usuario, rol o un Servicio de AWS. CloudTrail captura las llamadas a la API para CloudWatch los registros como eventos. Las llamadas capturadas incluyen llamadas desde la consola de CloudWatch Logs y llamadas de código a las operaciones de la API de CloudWatch Logs. 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, cuándo se realizó y detalles adicionales.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales del usuario raíz o del usuario.
- Si la solicitud se realizó en nombre de un usuario de IAM Identity Center.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro Servicio de AWS.

CloudTrail está activa en tu cuenta Cuenta de AWS cuando creas la cuenta y tienes acceso automáticamente al historial de CloudTrail eventos. El historial de CloudTrail eventos proporciona un registro visible, consultable, descargable e inmutable de los últimos 90 días de eventos de gestión registrados en un. Región de AWSPara obtener más información, consulte Cómo trabajar con el historial de CloudTrail eventos en la Guía del usuario.AWS CloudTrail La visualización del historial de eventos no conlleva ningún CloudTrail cargo.

Para tener un registro continuo de los eventos de Cuenta de AWS los últimos 90 días, crea un almacén de datos de eventos de senderos o CloudTraillagos.

CloudTrail senderos

> Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. Todos los senderos creados con él Consola de administración de AWS son multirregionales. Puede crear un registro de seguimiento de una sola región o multirregionales mediante la AWS CLI. Se recomienda crear un sendero multirregional, ya que puedes capturar toda la actividad de tu

Regiones de AWS cuenta. Si crea un registro de seguimiento de una sola región, solo podrá ver los eventos registrados en la Región de AWS del registro de seguimiento. Para obtener más información acerca de los registros de seguimiento, consulte Creación de un registro de seguimiento para su Cuenta de AWS y Creación de un registro de seguimiento para una organización en la Guía del usuario de AWS CloudTrail .

Puede enviar una copia de sus eventos de administración en curso a su bucket de Amazon S3 sin coste alguno CloudTrail mediante la creación de una ruta; sin embargo, hay cargos por almacenamiento en Amazon S3. Para obtener más información sobre CloudTrail los precios, consulte AWS CloudTrail Precios. Para obtener información acerca de los precios de Amazon S3, consulte Precios de Amazon S3.

CloudTrail Almacenes de datos de eventos en Lake

CloudTrail Lake le permite ejecutar consultas basadas en SQL en sus eventos. CloudTrail Lake convierte los eventos existentes en formato JSON basado en filas al formato Apache ORC. ORC es un formato de almacenamiento en columnas optimizado para una recuperación rápida de datos. Los eventos se agregan en almacenes de datos de eventos, que son recopilaciones inmutables de eventos en función de criterios que se seleccionan aplicando selectores de eventos avanzados. Los selectores que se aplican a un almacén de datos de eventos controlan los eventos que perduran y están disponibles para la consulta. Para obtener más información sobre CloudTrail Lake, consulte Cómo trabajar con AWS CloudTrail Lake en la Guía del AWS CloudTrail usuario.

CloudTrail Los almacenes de datos y las consultas sobre eventos de Lake conllevan costes. Cuando crea un almacén de datos de eventos, debe elegir la opción de precios que desee utilizar para él. La opción de precios determina el costo de la incorporación y el almacenamiento de los eventos, así como el período de retención predeterminado y máximo del almacén de datos de eventos. Para obtener más información sobre CloudTrail los precios, consulte AWS CloudTrail Precios.

CloudWatch Logs permite registrar las siguientes acciones como eventos en los archivos de CloudTrail registro:

- AssociateKmsKey
- CancelExportTask
- CreateDelivery
- CreateExportTask

- CreateLogAnomalyDetector
- CreateLogGroup
- CreateLogStream
- DeleteAccountPolicy
- DeleteDataProtectionPolicy
- DeleteDelivery
- DeleteDeliveryDestination
- DeleteDeliveryDestinationPolicy
- DeleteDeliverySource
- DeleteDestination
- DeleteIndexPolicy
- DeleteIntegration
- DeleteLogAnomalyDetector
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteQueryDefinition
- DeleteResourcePolicy
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- DeleteTransformer
- DescribeAccountPolicies
- DescribeConfigurationTemplates
- DescribeDeliveries
- DescribeDeliveryDestinations
- DescribeDeliverySources
- DescribeDestinations
- DescribeExportTasks
- DescribeFieldIndexes
- DescribeIndexPolicies

- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeQueryDefinitions
- DescribeResourcePolicies
- DescribeSubscriptionFilters
- DisassociateKmsKey
- FilterLogEvents
- GetDataProtectionPolicy
- GetDelivery
- GetDeliveryDestination
- GetDeliveryDestinationPolicy
- GetDeliverySource
- GetIntegration
- GetLogAnomalyDetector
- GetLogEvents
- GetLogGroupFields
- GetLogRecord
- GetQueryResults
- GetTransformer
- ListAnomalies
- ListIntegrations
- ListLogAnomalyDetectors
- ListLogGroups
- ListLogGroupsForQuery
- ListTagsForResource
- ListTagsLogGroup
- PutAccountPolicy
- PutDataProtectionPolicy

- PutDeliveryDestination
- PutDeliveryDestinationPolicy
- PutDeliverySource
- PutDestination
- PutDestinationPolicy
- PutIndexPolicy
- PutIntegration
- PutMetricFilter
- PutQueryDefinition
- PutResourcePolicy
- PutRetentionPolicy
- PutSubscriptionFilter
- PutTransformer
- StartLiveTail
- StartQuery
- StopQuery
- TagResource
- TestMetricFilter
- TestTransformer
- UntagResource
- UpdateAnomaly
- UpdateDeliveryConfiguration
- UpdateLogAnomalyDetector

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 userIdentity de CloudTrail .

Consulta la información de generación 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"
}
```

Monitorización con CloudWatch métricas

Puedes usar las tablas de esta sección para revisar las estadísticas que Amazon CloudWatch Logs envía a Amazon CloudWatch cada minuto.

CloudWatch Registra las métricas

El espacio de nombres de `AWS/Logs` incluye las siguientes métricas.

Métrica	Description (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	Description (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
`EMFDisabledErrors`	Número de eventos de registro con formato EMF que se ignoraron debido a una política de cuenta activa del tipo METRIC_EXTRACTION_POLICY que coincide con el grupo de registros. Para obtener más información sobre las políticas de extracción de métricas, consulte. PutAccountPolicy Dimensiones válidas: `LogGroupName` Estadísticas válidas: suma Unidades: ninguna

Métrica	Description (Descripción)
`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
`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 integrado, consulte Especificación: formato métrico CloudWatch 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

Métrica	Description (Descripción)
`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
`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
`Forwarded LogEvents`	El número de eventos de registro reenviados al destino de la suscripción. Dimensiones válidas: LogGroupName, DestinationType, FilterName, PolicyLevel Estadísticas válidas: suma Unidades: ninguna

Métrica	Description (Descripción)
`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 registro. 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 registro. Dimensiones válidas: LogGroupName Estadísticas válidas: suma Unidades: ninguna
`LogEventsWithFindings`	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

Métrica	Description (Descripción)
`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 la mayoría de las métricas de CloudWatch Logs se muestran en la siguiente tabla.

Dimensión	Description (Descripción)
`LogGroupName`	El nombre del grupo de CloudWatch registros para el que se muestran las métricas.
`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 registro 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 (?).

Dimensiones de las métricas del filtro de suscripciones

Las dimensiones de las métricas relacionadas con los filtros de suscripción a nivel de cuenta se enumeran en la siguiente tabla.

Dimensión	Description (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 registro 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 (?).

Métricas y dimensiones del transformador de registro

CloudWatch Logs publica las siguientes métricas de transformación de registros CloudWatch en el `AWS/Logs` espacio de nombres.

Métrica	Description (Descripción)
`TransformationErrors`	La cantidad de errores detectados al transformar los eventos de registro con el transformador especificado. Unidad: ninguna Estadística válida: suma
`TransformedBytes`	El volumen de la salida de los eventos de registro transform ados, en bytes sin comprimir. Unidad: bytes Estadística válida: suma

Métrica	Description (Descripción)
`TransformedLogEvents`	La cantidad de eventos de registro transformados. Unidad: ninguna Estadística válida: suma

Las métricas de transformador utilizan las siguientes dimensiones.

Dimensión	Description (Descripción)
`LogGroupname`	Esta dimensión solo se usa para los transformadores. log-group -level
`PolicyLevel`	Esta dimensión solo se usa para transformadores a nivel de grupo de cuentas. Actualmente, el único valor válido para esta dimensión es `AccountPolicy`

Métricas y dimensiones de centralización

Para una supervisión centralizada en varias cuentas y regiones, puede utilizar la centralización de CloudWatch registros para consolidar los datos y las métricas de los registros en una ubicación central. Para obtener más información, consulte Cross-account Centralización de registros entre regiones.

CloudWatch Logs publica las siguientes métricas de centralización CloudWatch en el `AWS/Logs` espacio de nombres. Estas métricas le ayudan a supervisar la replicación de los datos de registro de las cuentas de origen a las cuentas de destino al utilizar las reglas de centralización de CloudWatch Logs.

> **Note**
>
> CloudWatch comienza a generar informes sobre las métricas de centralización poco después de crear una regla de centralización. Las métricas se publican con el máximo esfuerzo

y solo registran los eventos de registro nuevos que llegan después de crear la regla de centralización.

Métrica	Description (Descripción)	Publicado en
`IncomingCopiedBytes`	El volumen de datos de registro en bytes sin comprimir que se replicó en la cuenta de destino. Esta métrica se aplica solo a los nuevos eventos de registro que llegan después de crear la regla de centralización. Dimensiones válidas: SourceRegion, SourceAccount Estadísticas válidas: suma Unidades: bytes	Cuenta de destino
`IncomingCopiedLogEvents`	El número de eventos de registro que se replicaron en la cuenta de destino. Esta métrica se aplica solo a los nuevos eventos de registro que llegan después de crear la regla de centralización. Dimensiones válidas: SourceRegion, SourceAccount Estadísticas válidas: suma Unidades: ninguna	Cuenta de destino
`OutgoingCopiedBytes`	El volumen de datos de registro en bytes sin comprimir que se envió desde las cuentas de origen a la cuenta de destino. Esta métrica se aplica solo a los eventos de registro nuevos que se copian en el destino después de crear la regla de centralización.	Cuenta de origen

Métrica	Description (Descripción)	Publicado en
	Dimensiones válidas: DestinationRegion Estadísticas válidas: suma Unidades: bytes	
`OutgoingCopiedLogEvents`	El número de eventos de registro que se enviaron desde las cuentas de origen a la cuenta de destino. Esta métrica se aplica solo a los eventos de registro nuevos que se copian en el destino después de crear la regla de centralización. Dimensiones válidas: DestinationRegion Estadísticas válidas: suma Unidades: ninguna	Cuenta de origen
`CentralizationError`	El número de errores encontrados al replicar los datos de registro. Los errores se pueden producir por varios motivos, como problemas con los permisos clave del KMS, los límites de las cuotas de los grupos de registros o las discordancias entre los niveles de registro. Para identificar grupos de registros o flujos de registros específicos que no pudieron replicars e y sus motivos, supervise la `ErrorType` dimensión. Dimensiones válidas: ErrorType Estadísticas válidas: suma Unidades: ninguna	Cuenta de destino

Métrica	Description (Descripción)	Publicado en
`Centraliz ationThrottled`	El número de veces que se ha limitado el proceso de centralización. La limitación hace que el procesamiento de la centralización sea más lento, pero no impide que los datos de registro se repliquen. Dimensiones válidas: DestinationRegion Estadísticas válidas: suma Unidades: ninguna	Cuenta de origen

Las métricas de centralización utilizan las siguientes dimensiones.

Dimensión	Description (Descripción)
`SourceRegion`	La AWS región en la que se originaron los datos de registro de origen.
`SourceAccount`	El ID de AWS cuenta en el que se originaron los datos de registro de origen.
`DestinationRegion`	La AWS región en la que se replican los datos de registro.
`ErrorType`	El tipo de error que se ha producido durante la centralización. Los valores posibles son: • `LogGroupQuotaExceeded` : La cuenta de destino ha alcanzado su cuota de grupos de registros. • `InvalidKMS` : Falta la clave KMS, está eliminada, no está habilitada o es asimétrica en lugar de simétrica. • `AccessDenied` : Se denegó el acceso al intentar replicar los datos de registro. Esto puede ocurrir debido a permisos insuficientes, como permisos de clave de KMS incorrectos, permisos de IAM faltantes o políticas de recursos restrictivas.

Dimensión	Description (Descripción)
	• `LogTierMismatch` : El nivel de registro de los grupos de registro de origen y destino no coincide. • `InvalidLogStream` : Se encontró un parámetro no válido al crear o escribir en un flujo de registro, por ejemplo, cuando el nombre del flujo de registro supera la longitud máxima. • `InvalidLogGroup` : Se encontró un parámetro no válido al crear o configurar un grupo de registros en la cuenta de destino. • `DestinationEncryptionMismatch` : El grupo de registros de destino ya existe con una configuración de cifrado de KMS diferente a la que especifica la regla de centraliz ación.

CloudWatch Registra las métricas de uso del servicio

CloudWatch Logs envía métricas CloudWatch que rastrean el uso de las operaciones de la API de 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, consulteCloudWatch 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 enumeran 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	Description (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	Description (Descripción)
`Account metrics`	Utilice esta dimensión para obtener una suma de la métrica de todos los CloudWatch registros APIs. 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`.

Dimensión	Description (Descripción)
`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

Puedes usar la tabla de esta sección para revisar las cuotas de servicio predeterminadas, también denominadas límites, de una AWS cuenta en Amazon CloudWatch Logs. La mayoría de las cuotas de servicio, pero no todas, aparecen en el espacio de nombres Amazon CloudWatch Logs de la consola Service Quotas.

> **Note**
>
> Si quiere solicitar un aumento de dichas cuotas, consulte el procedimiento más adelante en esta sección.

Name	Predeterminado	Ajusta	Description (Descripción)
Tarea de exportación activa	Cada región admitida: 1	No	Número de tareas de exportación activas (en ejecución o pendientes) por cuenta
AssociateKmsKey límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas de associate-kms-key por segundo por región account/per
AssociateSourceToS3TableIntegration límite máximo de transacciones por segundo	Cada región admitida: 5	No	El número máximo de AssociateSourceToS 3TableIntegration llamadas por segundo por región account/per
AssociateSourceToS3TableIntegration límite máximo de transacciones por segundo en ráfaga	Cada región admitida: 10	No	El número máximo de AssociateSourceToS 3TableIntegration llamadas por segundo por account/per región en una ráfaga

Name	Predeterminado	Ajusta	Description (Descripción)
Tamaño de lote	Cada región admitida: 1 megabyte	No	El tamaño máximo de lote en MB de una solicitud de colocar los eventos de registro
CancelExportTask límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas de cancelación y exportación de tareas por segundo por región account/per
CancelImportTask límite máximo de transacciones por segundo	Cada región admitida: 1	No	El número máximo de CancelImportTask llamadas por segundo por región account/per
CancelImportTask límite máximo de transacciones por segundo en ráfaga	Cada región admitida: 1	No	El número máximo de CancelImportTask llamadas por segundo por account/per región en una ráfaga
CreateExportTask límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas de creación y exportación de tareas por segundo por región account/per
CreateImportTask límite máximo de transacciones por segundo	Cada región admitida: 1	No	El número máximo de CreateImportTask llamadas por segundo por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
CreateImportTask límite máximo de transacciones por segundo en ráfaga	Cada región admitida: 1	No	El número máximo de CreateImportTask llamadas por segundo por account/per región en una ráfaga
CreateLogGroup límite máximo de transacciones por segundo	Cada región admitida: 10 por segundo	Sí	El número máximo de llamadas de creación de grupos de registro por segundo por región account/per
CreateLogStream límite máximo de transacciones por segundo	Cada región admitida: 50 por segundo	Sí	El número máximo de llamadas de creación de secuencias de registro por segundo por región account/per
Archivado de datos	Cada región admitida: 5 gigabytes	No	El tamaño máximo en GB del archivado de datos gratuito
DeleteDataProtectionPolicy límite máximo de transacciones por segundo	Cada región admitida: 5	No	El número máximo de llamadas por segundo por región relacionadas con la política de eliminación de datos account/per
DeleteDestination límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas eliminadas de destinos por segundo por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
DeleteLogGroup límite máximo de transacciones por segundo	Cada región admitida: 10 por segundo	Sí	El número máximo de llamadas a grupos para eliminar registros por segundo y por región account/per
DeleteLogStream límite máximo de transacciones por segundo	Cada región admitida: 15 por segundo	Sí	El número máximo de llamadas de borrado de registros por segundo por región account/per
DeleteMetricFilter límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas con filtro de eliminación de métricas por segundo por región account/per
DeleteRetentionPolicy límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas relaciona das con la política de eliminación y retención por segundo por región account/per
DeleteSubscriptionFilter límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas al filtro de eliminación de suscripci ones por segundo por región account/per
DeleteTransformer límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas a eliminar transformadores por segundo y por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
DescribeDestinations límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas descritas como destinos por segundo por región account/per
DescribeExportTasks límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas de descripción y exportación de tareas por segundo por región account/per
DescribeImportTaskBatches límite máximo de transacciones por segundo	Cada región admitida: 10	No	El número máximo de DescribeImportTask Batches llamadas por segundo por región account/per
DescribeImportTaskBatches límite máximo de transacciones por segundo en ráfaga	Cada región admitida: 10	No	El número máximo de DescribeImportTask Batches llamadas por segundo por account/per región en una ráfaga
DescribeImportTasks límite máximo de transacciones por segundo	Cada región admitida: 10	No	El número máximo de DescribeImportTasks llamadas por segundo por región account/per
DescribeImportTasks límite máximo de transacciones por segundo en ráfaga	Cada región admitida: 10	No	El número máximo de DescribeImportTasks llamadas por segundo por account/per región en una ráfaga

Name	Predeterminado	Ajusta	Description (Descripción)
DescribeLogGroups límite máximo de transacciones por segundo	Cada región admitida: 10 por segundo	Sí	El número máximo de llamadas de Describe-Log-Groups por segundo por región account/per
DescribeLogStreams límite máximo de transacciones por segundo	Cada región admitida: 25 por segundo	Sí	El número máximo de llamadas descritas y registradas por segundo por región account/per
DescribeMetricFilters límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas de Describe-Metric-Filters por segundo por región account/per
DescribeSubscriptionFilters límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas descritas, filtradas por suscripción y por segundo por región account/per
DisassociateSourceFromS3TableIntegration límite máximo de transacciones por segundo	Cada región admitida: 5	No	El número máximo de DisassociateSourceFromS3TableIntegration llamadas por segundo por región account/per
DisassociateSourceFromS3TableIntegration límite máximo de transacciones por segundo en ráfaga	Cada región admitida: 10	No	El número máximo de DisassociateSourceFromS3TableIntegration llamadas por segundo por account/per región en una ráfaga

Name	Predeterminado	Ajusta	Description (Descripción)
Tamaño de eventos	Cada región admitida: 1024 kilobytes	No	Tamaño máximo del evento de registro (en KB).
FilterLogEvents límite máximo de transacciones por segundo	us-east-1: 25 por segundo ap-northeast-3: 5 por segundo ap-southeast-3: 5 por segundo ca-west-1:5 por segundo eu-central-1: 5 por segundo il-central-1:5 por segundo Cada una de las demás regiones compatibles: 10 por segundo	No	El número máximo de llamadas a filter-log-events por segundo por región account/per
GetDataProtectionPolicy límite máximo de transacciones por segundo	Cada región admitida: 5	No	El número máximo de llamadas a get-data-protection-policy por segundo por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
GetLogEvents límite máximo de transacciones por segundo	us-west-2: 10 por segundo ap-northeast-3: 10 por segundo ap-southeast-3: 10 por segundo ca-west-1:10 por segundo eu-central-1: 10 por segundo eu-west-1: 10 por segundo eu-west-3: 30 por segundo il-central-1:10 por segundo Cada una de las demás regiones compatibles: 25 por segundo	No	El número máximo de llamadas a get-log-events por segundo por región account/per
GetQueryResults límite máximo de transacciones por segundo	me-central-1:6 Cada una de las demás regiones compatibles: 10	No	El número máximo de llamadas a get-query-results por segundo por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
GetTransformer límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas a get-trans former por segundo por región account/per
ListSourcesForS3TableIntegration límite máximo de transacciones por segundo	Cada región admitida: 5	No	El número máximo de ListSourcesForS3Ta bleIntegration llamadas por segundo por región account/per
ListSourcesForS3TableIntegration límite máximo de transacciones por segundo en ráfaga	Cada región admitida: 10	No	El número máximo de ListSourcesForS3Ta bleIntegration llamadas por segundo por account/ per región en una ráfaga
ListTagsForResource límite máximo de transacciones por segundo	ca-central-1:30 por segundo me-central-1:5 por segundo Cada una de las demás regiones compatibles: 10 por segundo	No	El número máximo de llamadas a la lista de etiquetas para los recursos por segundo y por región account/per
ListTagsLogGroup límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas a listas, etiquetas, registros y grupos por segundo por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
Límite de sesiones simultáneas de Live Tail	Cada región admitida: 15	Sí	El número máximo de sesiones de Live Tail simultáneas activas por cuenta
Grupos de registro	Cada región admitida: 1 000 000	Sí	El número máximo de grupos de registro que puede tener una cuenta.
Grupos de registros escaneados por sesión de Live Tail	Cada región admitida: 10	No	El número máximo de grupos de registro que se pueden escanear por sesión de Live Tail
Filtros de métricas por grupo de registro	Cada región admitida: 100	No	El número de filtros de métricas por grupo de registro
PutBearerTokenAuthentication límite máximo de transacciones por segundo	Cada región admitida: 5	No	El número máximo de llamadas de autentica ción put-bearer-token por segundo por región account/per
PutBearerTokenAuthentication límite máximo de transacciones por segundo en ráfaga	Cada región admitida: 10	No	El número máximo de llamadas de autenticación tipo put-bearer-token por segundo y región en una ráfaga account/per
PutDataProtectionPolicy límite máximo de transacciones por segundo	Cada región admitida: 5	No	El número máximo de llamadas basadas en la política de protección de datos por segundo por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
PutDestination límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas de destino final por segundo por región account/per
PutDestinationPolicy límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas basadas en la política de destino final por segundo por región account/per
PutLogEvents límite máximo de transacciones por segundo	Cada región compatible: 5000 por segundo	Sí	El número máximo de llamadas de put-log-events por segundo por región account/per
PutMetricFilter límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas con filtro Put-Metric por segundo por región account/per
PutRetentionPolicy límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas basadas en la política de retención de datos por segundo por región account/per
PutSubscriptionFilter límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas con filtro de subscripción por segundo por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
PutTransformer límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas con transform adores de venta por segundo y por región account/per
Políticas de recursos	Cada región admitida: 10	No	El número máximo de políticas de recursos por región account/per
StartLiveTail límite máximo de transacci ones por segundo	Cada región admitida: 5	No	El número máximo de llamadas iniciadas en directo por segundo y por región. account/per Este límite se aplica de forma independiente tanto a la consola como a la API
StartQuery límite máximo de transacci ones por segundo	me-central-1:6 Cada una de las demás regiones compatibles: 10	No	El número máximo de llamadas de inicio de consulta por segundo por región account/per
Filtros de suscripción por grupo de registro	Cada región admitida: 2	No	El número de filtros de suscripción por grupo de registro
TagLogGroup límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas a grupos de registro de etiquetas por segundo por región account/per

Name	Predeterminado	Ajusta	Description (Descripción)
TagResource límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas a recursos de etiquetas por segundo y por región account/per
TestMetricFilter límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas con filtro métrico de prueba por segundo por región account/per
TestTransformer límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas a un transform ador de prueba por segundo por región account/per
UntagLogGroup límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas sin etiquetar y registrar grupos por segundo por región account/per
UntagResource límite máximo de transacciones por segundo	Cada región admitida: 5 por segundo	No	El número máximo de llamadas a recursos sin etiquetar por segundo por región account/per
detectores de anomalías de registros	Cada región admitida: 500	Sí	El número máximo de detectores de anomalías de registro activos

Administrar las cuotas del servicio de registros CloudWatch

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.

Consola de administración de AWS

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 CloudWatch de servicio de 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

En la siguiente tabla, se describen 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
Se actualizó la AWSServiceRoleForLogDelivery política de funciones vinculadas a los servicios	CloudWatch Registros agregados `kms:GenerateDataKey` y `kms:Decrypt` permisos a la política de roles vinculados al AWSServiceRoleForLogDelivery servicio. Estos permisos permiten la entrega de registros a las transmisiones de entrega de Firehose que utilizan cifrado del lado del servidor con claves administradas por el cliente (). SSE-CMK Para obtener más información, consulte CloudWatch Registra las actualizaciones de las funciones vinculadas al AWS servicio.	15 de mayo de 2026
Consultas guardadas con parámetros para CloudWatch Logs Insights	Ahora puede crear plantillas de consultas reutilizables con parámetros con nombre en CloudWatch Logs Insights. Defina una plantilla única y transfiera valores diferentes en tiempo de ejecución en lugar de mantener varias consultas guardadas casi idénticas.	26 de marzo de 2026

	Para obtener más información, consulte Uso de consultas guardadas con parámetros.	
Nueva política CloudWatch de registros gestionados CloudWatchLogsAPIKeyAccess	CloudWatch Logs ha publicado una nueva política gestionada `CloudWatchLogsAPIKeyAccess` que permite la autenticación de claves de la API de Logs y la ingesta de CloudWatch registros cifrados. Para obtener más información, consulte CloudWatch Registra las actualizaciones de las políticas AWS gestionadas.	17 de febrero de 2026
CloudWatchLogsReadOnlyAccess Política actualizada	Se agregaron permisos para las acciones de administración de la observabilidad a fin de permitir el acceso de solo lectura CloudWatchLogsReadOnlyAccessa las canalizaciones de telemetría y a las integraciones de tablas de S3.	2 de diciembre de 2025
CloudWatchLogsFullAccess Política actualizada	Se agregaron permisos para las acciones de administración de la observabilidad a fin de permitir el acceso de solo lectura CloudWatchLogsFullAccessa las canalizaciones de telemetría y a las integraciones de tablas de S3.	2 de diciembre de 2025

Nueva función de administración de CloudWatch registros en Logs	CloudWatch consolida la administración de registros en un solo servicio con capacidades de gobierno integradas sin almacenar ni mantener varias copias de los mismos datos en diferentes herramientas y almacenes de datos. Para obtener más información, consulte Administración de registros.	2 de diciembre de 2025
Nueva función de ingesta y normalización de datos en Logs CloudWatch	CloudWatch recopila automáticamente los AWS registros vendidos en todas las cuentas y AWS regiones mediante la integración con AWS Organizations y conectores prediseñados para fuentes de terceros. Para obtener más información, consulte Transform logs during ingestion.	2 de diciembre de 2025

Nueva función de análisis y consultas basadas en fuentes de datos de CloudWatch Logs Insights	CloudWatch Insights presenta facetas para las consultas. Las facetas son útiles para analizar los registros, ya que permiten filtrar y agrupar los datos de forma interactiva sin ejecutar consultas. Una faceta es un campo de los registros (por ejemplo, ServiceName o StatusCode) que permite filtrar, agregar y analizar los distintos grupos de registros. Para obtener más información, consulte Usar facetas para agrupar y explorar los registros.	2 de diciembre de 2025
Nueva integración de tablas S3	La integración de S3 Tables con CloudWatch le permite acceder a los datos de registro ingeridos CloudWatch mediante motores de análisis como Amazon Athena, Amazon Redshift y herramientas de terceros que admiten la conexión a almacenes de Iceberg-compatible Apache. Para obtener más información, consulte Acceder a los registros con la integración de S3 Tables.	2 de diciembre de 2025

CloudWatch Los registros agregaron soporte para la Cross-account centralización de registros entre regiones	CloudWatch Logs agregó soporte para la centraliz ación de registros Cross-acc ount entre regiones, lo que permite a CloudWatch Logs recopilar datos de registro de las cuentas de los AWS Organizations miembros en una cuenta central para su análisis. Para obtener más información, consulte Centraliz ación de registros Cross-acc ountentre regiones.	17 de septiembre de 2025
CloudWatch Ejemplo de política actualizada de detección de anomalías en los registros	CloudWatch Logs actualizó la política de ejemplo de KMS para reducir su alcance y mejorar la seguridad añadiendo condiciones para `aws:SourceAccount` y`aws:SourceArn` . Para obtener más información, consulte Cifrar un detector de anomalías y sus resultados con KMS AWS	14 de julio de 2025

CloudWatch Logs Insights añade compatibilidad con los registros de Amazon VPC Route Server	CloudWatch Logs Insights añade compatibilidad con los registros de Amazon VPC Route Server. Para obtener más información, consulte Habilitar el registro desde servicios de AWS. La nueva fuente de registro de Amazon VPC Route Server `EVENT_LOGS` está documentada en PutDeliverySource	12 de junio de 2025
CloudWatch Logs Insights añade soporte para AWS PCS registros	CloudWatch Logs Insights añade compatibilidad con los AWS PCS registros. Para obtener más información, consulte Habilitar el registro desde servicios de AWS. Las nuevas fuentes de registro AWS PCS `PCS_SCHEDULER_LOGS` y `PCS_JOBCOMP_LOGS` están documentadas en PutDeliverySource	12 de junio de 2025
CloudWatch Logs Insights añade compatibilidad con los AWS Entity Resolution registros	CloudWatch Logs Insights añade compatibilidad con los AWS Entity Resolution registros. Para obtener más información, consulte Habilitar el registro desde servicios de AWS. La nueva fuente de registro de AWS Entity Resolution, `WORKFLOW_LOGS` está documentada en PutDeliverySource	22 de mayo de 2025

CloudWatch Registra las actualizaciones de políticas gestionadas para admitir los resúmenes en lenguaje natural	Se `cloudwatch:GenerateQueryResultsSummary` agregaron permisos para generar un resumen en lenguaje natural de los resultados de la consulta, que CloudWatchLogsReadOnlyAccesspermiten generar un resumen en lenguaje natural. CloudWatchLogsFullAccess Para ver el contenido de las políticas, consulte CloudWatchLogsFullAccessy consulte CloudWatchLogsReadOnlyAccessla Guía de referencia de políticas AWS gestionadas.	20 de mayo de 2025
CloudWatch Logs Insights añade soporte para generar resúmenes en lenguaje natural a partir de los resultados de las consultas de CloudWatch Logs Insights	Se ha añadido compatibilidad con los resúmenes en lenguaje natural en CloudWatch Logs Insights. Esta característica genera resúmenes de los resultados de las consultas de lectura humana, disponibles actualmente en el Este de EE. UU. (Norte de Virginia). Para obtener más información, consulte Generar resúmenes en lenguaje natural a partir de los resultados de las consultas de CloudWatch Logs Insights.	20 de mayo de 2025

CloudWatchOpenSearchDashboardAccessy CloudWatchOpenSearchDashboardsFullAccessson nuevas políticas de IAM	CloudWatch Los registros agregaron dos nuevas políticas de IAM y CloudWatchOpenSearchDashboardsFullAccess. CloudWatchOpenSearchDashboardAccess CloudWatchOpenSearchDashboardsFullAccessconcede permiso para crear y gestionar integraciones con OpenSearch el Servicio. CloudWatchOpenSearchDashboardAccessotorga acceso para ver los paneles de control de registros vendidos que se crean en estas integraciones. Para obtener más información, consulta los paneles de registro de ventas con tecnología de Amazon OpenSearch Service.	1 de diciembre de 2024
CloudWatchLogsFullAccesspolítica actualizada	CloudWatch Los registros agregaron permisos Amazon OpenSearch Service e IAM a la CloudWatchLogsFullAccesspolítica para permitir la integración de CloudWatch Logs con el OpenSearch servicio para algunas funciones.	1 de diciembre de 2024

CloudWatch Logs Insights añade nuevos tipos de estructura a la sintaxis de consultas	CloudWatch Logs Insights añade el `unnest` comando y dos funciones JSON, que permiten utilizar cadenas JSON como mapas y listas. Para obtener más información, consulte Tipos de estructura.	21 de noviembre de 2024
CloudWatch Logs admite la transformación de registros durante la ingesta de registros	Se pueden crear transform adores de registros que puedan modificar los eventos del registro en el momento de la ingestión, lo que ayudará a normalizar los registros en diferentes formatos y fuentes diferentes para convertir los en formatos coherente s y ricos en contexto. Para obtener más información, consulte Transform logs during ingestion.	20 de noviembre de 2024

CloudWatch Logs Insights añade la indexación de campos	CloudWatch Logs Insights ha añadido soporte para la indexación de registros por campos. Cuando, a continuación, utilice un índice de campos en una consulta de CloudWatch Logs Insights, la consulta intenta omitir el procesamiento de los eventos de registro que se sabe que no incluyen el campo indexado. Para obtener más información, consulte Crear índices de campos para mejorar el rendimiento de las consultas y reducir el volumen de digitalización.	20 de noviembre de 2024
CloudWatch El soporte de Logs Insights para la generación de consultas en lenguaje natural está generalmente disponible	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

CloudWatchLogsRead OnlyAccesspolítica actualizada	CloudWatch Logs agregó el `cloudwatch:GenerateQuery` permiso para CloudWatchLogsRead OnlyAccessque 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
CloudWatchLogsFull Accesspolítica actualizada	CloudWatch Logs agregó el `cloudwatch:GenerateQuery` permiso para CloudWatchLogsFull Accessque 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
CloudWatch Logs añade un análisis de patrones de registro	CloudWatch Ahora, Logs busca patrones en los eventos de registro cada vez que se realiza una consulta de CloudWatch Logs Insights. Para obtener más información, consulte Análisis de patrones.	26 de noviembre de 2023

CloudWatch Logs añade la detección de anomalías en el registro	Puede crear un detector de anomalías de registros para un grupo de registro. Un detector de anomalías de registro analiza los eventos de registro incorporados al grupo de registro y busca anomalías en los datos del registro. Para obtener más informaci ón, consulte Detección de anomalías de registro.	26 de noviembre de 2023
CloudWatch Logs añade una función de comparación	Ahora puede usar CloudWatc h Logs Insights para comparar los cambios en sus eventos de registro a lo largo del tiempo. Para obtener más información, consulte Comparar (diferenc iar) con intervalos de tiempo anteriores.	26 de noviembre de 2023
CloudWatch Logs añade 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

Account-level políticas de protección de datos	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 característica 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

CloudWatchLogsReadOnlyAccesspolítica actualizada	CloudWatch Registra los permisos añadidos 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
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	Description (Descripción)	Fecha de la versión
Puntos de conexión de VPC de la 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 CloudWatch Logs? o Registro de consultas de DNS en la Guía para desarrolladores de Amazon Route 53.	7 de septiembre de 2017
Etiquetar grupos de registro	Puede utilizar las etiquetas para categorizar los grupos de registro. Para obtener más información, 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 registro 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	Se ha añadido AWS CloudTrail soporte para CloudWatch Logs. Para obtener más informaci	10 de marzo de 2016

Cambio	Description (Descripción)	Fecha de la versión
l soporte para Amazon CloudWatch Logs y nuevas métricas CloudWatch de Logs	ón, consulte El registro CloudWatch registra las operaciones de la API y la consola en AWS CloudTrail.	
Se agregó soporte para la exportación de CloudWatch 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
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 la Glosario 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.