aws

Guía del usuario

Amazon Kinesis Agent para Microsoft Windows

Amazon Kinesis Agent para Microsoft Windows: Guía del usuario

Table of Contents

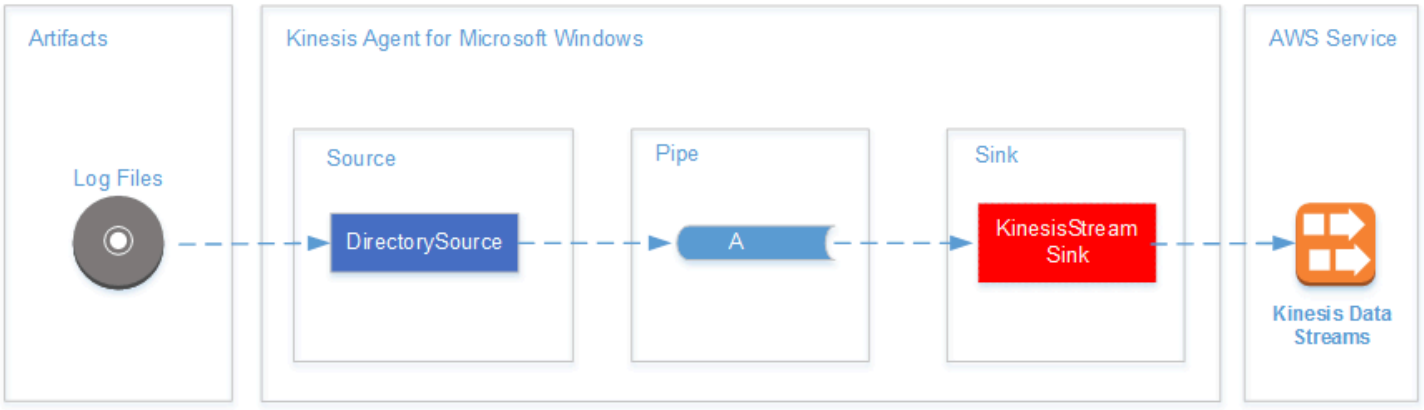
¿Qué es Amazon Kinesis Agent para Microsoft Windows?

Amazon Kinesis Agent para Microsoft Windows (Kinesis Agent para Windows) es un agente que se puede configurar y que admite ampliaciones. Se ejecuta en flotas de servidores y equipos de escritorio Windows, tanto en el entorno local como en la nube de AWS. Kinesis Agent para Windows recopila, analiza, transforma y transmite registros, eventos y métricas de forma eficaz y fiable a diferentes servicios de AWS, comoKinesis Data Streams,Kinesis Data Firehose,Amazon CloudWatch, yRegistros de CloudWatch.

En estos servicios, puede almacenar, analizar y visualizar los datos utilizando otros servicios de AWS diferentes, entre los que se incluyen los siguientes:

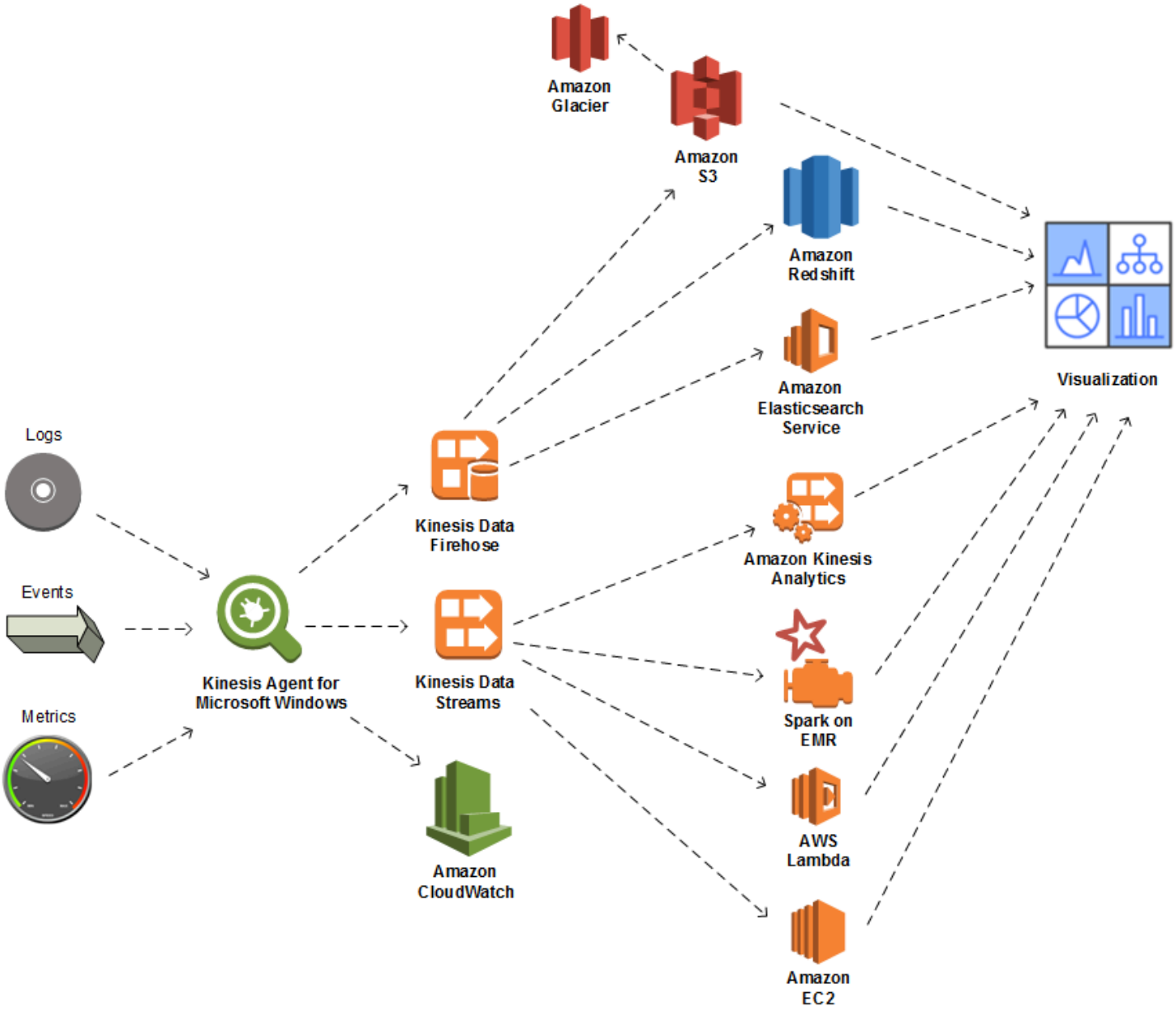
- Amazon Simple Storage Service (Amazon S3)
- Amazon Redshift
- Amazon Elasticsearch Service (Amazon ES)
- Kinesis Data Analytics
- Amazon QuickSight
- Amazon Athena
- Kibana

En el siguiente diagrama, se ilustra una sencilla configuración de Kinesis Agent para Windows que transmite archivos de registro a Kinesis Data Streams.



Para obtener más información sobre los orígenes, las canalizaciones y los receptores, consulte Conceptos de Amazon Kinesis Agent para Microsoft Windows.

En el siguiente diagrama, se ilustran algunas de las formas en las que pueden crearse canalizaciones de datos personalizadas en tiempo real utilizando marcos de procesamiento de secuencias. Estos marcos incluyen Kinesis Data Analytics, Apache Spark en Amazon EMR y AWS Lambda.



Temas

- Acerca de AWS
- ¿Qué se puede hacer con Kinesis Agent para Windows?
- Benefits
- Introducción a Kinesis Agent para Windows

Acerca de AWS

Amazon Web Services (AWS) es una colección de servicios de infraestructura digital que se pueden usar para desarrollar aplicaciones. Los servicios incluyen informática, almacenamiento, base de datos, análisis y sincronización de aplicaciones (mensajería y puesta en cola). AWS usa un modelo de servicio de pago por uso. Solo se le cobrará por los servicios que usted (o su aplicación) usa. Además, para poder usar los servicios más accesibles a la hora de crear prototipos y experimentos, AWS ofrece una capa de uso gratuita. En esta capa, los servicios son gratuitos por debajo de determinado nivel de uso. Para obtener más información acerca de los costos de AWS y laCapa gratuitaConsulte laCentro de recursos introductorios. Para crear una cuenta de AWS, abra la página de inicio de AWS y regístrese.

¿Qué se puede hacer con Kinesis Agent para Windows?

Kinesis Agent para Windows cuenta con las siguientes características y funcionalidades:

Recopilación de registros, eventos y métricas

Kinesis Agent for Windows recopila, analiza, transforma y transmite registros, eventos y métricas procedentes de flotas de servidores y equipos de escritorio a uno o varios servicios de AWS. La carga que reciben los servicios pueden estar en un formato diferente al de la fuente original. Por ejemplo, un registro podría estar almacenado en un determinado formato de texto (por ejemplo, en formato syslog) en un servidor. Kinesis Agent para Windows puede recopilar y analizar ese texto y, de forma opcional, transformarlo a formato JSON, por ejemplo, antes de transmitirlo a AWS. De este modo, el procesamiento resultaría más fácil en algunos servicios de AWS que consumen archivos JSON. Kinesis Data Analytics puede procesar de forma ininterrumpida los datos que se transmiten a Kinesis Data Streams para generar métricas adicionales y agregadas, que, a su vez, pueden alimentar paneles en directo. Puede almacenar los datos utilizando diferentes servicios de AWS (como Amazon S3) en función de cómo se utilicen los datos más adelante en una canalización de datos.

Integración con los servicios de AWS

Puede configurar Kinesis Agent para Windows para que envíe archivos de registro, eventos y métricas a diferentes servicios de AWS:

- Kinesis Data FirehoseAlmacene fácilmente los datos transmitidos por streaming en Amazon S3, Amazon Redshift, Amazon ES oSplunkPara obtener más análisis de.
- Kinesis Data Streams: procese los datos transmitidos mediante aplicaciones personalizadas alojadas en Kinesis Data Analytics o Apache Spark enAmazon EMR. O use código personalizado que se ejecuta enAmazon EC2Las instancias o funciones sin servidor personalizadas que se ejecuten enAWS Lambda.
- CloudWatch: vea las métricas transmitidas en gráficos, que puede combinar en paneles. A continuación, establezca las alarmas de CloudWatch que se activen mediante valores de métricas que infrinjan los umbrales preestablecidos.
- Registros de CloudWatch: almacene registros y eventos transmitidos por secuencias, y vea y busque en AWS Management Console, o procese más adelante en una canalización de datos.

Rápida instalación y configuración

Puede instalar y configurar Kinesis Agent para Windows en unos pocos pasos. Para obtener más información, consulte Instalación de Kinesis Agent para Windows y Configuración de Amazon Kinesis Agent para Microsoft Windows. A través de un sencillo archivo de configuración declarativo, se especifica lo siguiente:

- Los orígenes y los formatos de los registros, eventos y métricas que se van a recopilar.
- Las transformaciones que se van a aplicar a los datos recopilados. Pueden incluirse datos adicionales, mientras que los datos existentes pueden transformarse y filtrarse.
- Los destinos a los que se van a transmitir los datos finales, así como el almacenamiento en búfer, la fragmentación y el formato de las cargas de trabajo que se transmiten.

Kinesis Agent para Windows cuenta con analizadores integrados para archivos de registro generados por servicios empresariales comunes de Microsoft, como:

- Microsoft Exchange
- SharePoint
- Controladores de dominio de Active Directory
- Servidores DHCP

Sin administración permanente

Kinesis Agent para Windows se adapta automáticamente a diferentes situaciones sin perder ningún dato. Esto incluye la rotación de registros, la recuperación después de reiniciar y las interrupciones temporales de las redes o los servicios. Puede configurar Kinesis Agent para Windows para que se actualice automáticamente con las nuevas versiones. No se requiere la intervención del operador en ninguna de estas situaciones.

Amplíe el uso de una arquitectura abierta

Si las funcionalidades declarativas y los complementos integrados resultan insuficientes para supervisar sistemas de servidores o equipos de escritorio, puede ampliar Kinesis Agent para Windows creando complementos. Estos nuevos complementos habilitarán nuevos orígenes y destinos para los registros, eventos y métricas. El código fuente de Kinesis Agent para Windows está disponible enhttps://github.com/awslabs/kinesis-agent-windows.

Benefits

Kinesis Agent para Windows realiza la recopilación inicial de datos, la transformación y la transmisión de registros, eventos y métricas para canalizaciones de datos. La creación de estas canalizaciones de datos tiene numerosos beneficios:

Análisis y visualización

La integración de Kinesis Agent para Windows con Kinesis Data Firehose y sus funcionalidades de transformación facilitan la integración de diferentes servicios de análisis y visualización:

- Amazon QuickSight— Un servicio de BI basado en la nube que puede ingerir desde muchas fuentes diferentes. Kinesis Agent para Windows puede transformar datos y transmitirlos a Amazon S3 y Amazon Redshift a través de Kinesis Data Firehose. Este proceso permite extraer profundos conocimientos de los datos a través de las visualizaciones de Amazon QuickSight.
- Athena— Un servicio de consulta interactivo que permite la consulta de datos basada en SQL. Kinesis Agent para Windows puede transformar y transmitir datos a Amazon S3 a través de Kinesis Data Firehose. Athena puede ejecutar interactivamente consultas SQL contra esos datos para inspeccionar y analizar rápidamente registros y eventos.
- Kibana— Una herramienta de visualización de datos de código abierto. Kinesis Agent para Windows puede transformar y transmitir datos a Amazon ES a través de Kinesis Data Firehose. Kibana puede utilizarse también para explorar dichos datos. Cree y abra diferentes visualizaciones, como histogramas, gráficos circulares, gráficos de líneas, mapas térmicos y gráficos geoespaciales.

Security

Las canalizaciones de análisis de datos de registro y eventos que incluyen Kinesis Agent para Windows pueden detectar y recibir alertas sobre infracciones de seguridad en las organizaciones, lo que puede ayudarle a bloquear o detener ataques.

Rendimiento de la aplicación

Kinesis Agent para Windows puede recopilar datos de registros, eventos y métricas sobre el rendimiento de una aplicación o servicio. Estos datos pueden analizarse después a través de una canalización completa. Este análisis le ayudará a mejorar el rendimiento y la fiabilidad de la aplicación y el servicio detectando y notificando cualquier defecto que, de otro modo, podría pasar desapercibido. Por ejemplo, puede detectar cambios significativos en los tiempos de ejecución de llamadas a la API del servicio. Cuando se aplica a una implementación, esta funcionalidad le ayuda a localizar y resolver problemas de rendimiento nuevos en los servicios que ya posee.

Operaciones de servicio

Las canalizaciones de datos pueden analizar los datos recopilados para predecir posibles problemas operativos y proporcionar información acerca de cómo evitar interrupciones del servicio. Por ejemplo, puede analizar registros, eventos y métricas para determinar el uso actual y previsto de la capacidad a fin de que pueda incorporar capacidad adicional online antes de que los usuarios se vean afectados. Si se produce una interrupción del servicio, puede analizar los datos para determinar el impacto en los clientes durante este periodo.

Auditing

Las canalizaciones de datos pueden procesar los registros, los eventos y las métricas que recopila y transforma Kinesis Agent para Windows. Estos datos procesados pueden auditarse después con diferentes servicios de AWS. Por ejemplo, Kinesis Data Firehose podría recibir una secuencia de datos de Kinesis Agent para Windows, que almacena los datos en Amazon S3. Estos datos podrían auditarse ejecutando consultas SQL interactivas a través de Athena.

Archiving

A menudo, los datos operativos más importantes son datos que se han recopilado recientemente. Sin embargo, el análisis de los datos recopilados sobre las aplicaciones y servicios durante varios años también pueden ser útiles, por ejemplo, para realizar planificaciones a largo plazo. Puede resultar costoso mantener grandes cantidades de datos. Kinesis Agent para Windows puede recopilar, transformar y almacenar datos en Amazon S3 a través de Kinesis Data Firehose. Por consiguiente,Amazon S3 Glacierestá disponible para reducir los costos que supone archivar los datos más antiguos.

Alerting

Kinesis Agent para Windows transmite métricas a CloudWatch. A su vez, puede crear alarmas de CloudWatch para enviar una notificación a través deAmazon Simple Notification Service (Amazon SNS)cuando una métrica infringe sistemáticamente un umbral específico. Esto proporciona a los ingenieros una mejor perspectiva de los problemas operacionales de sus aplicaciones y servicios.

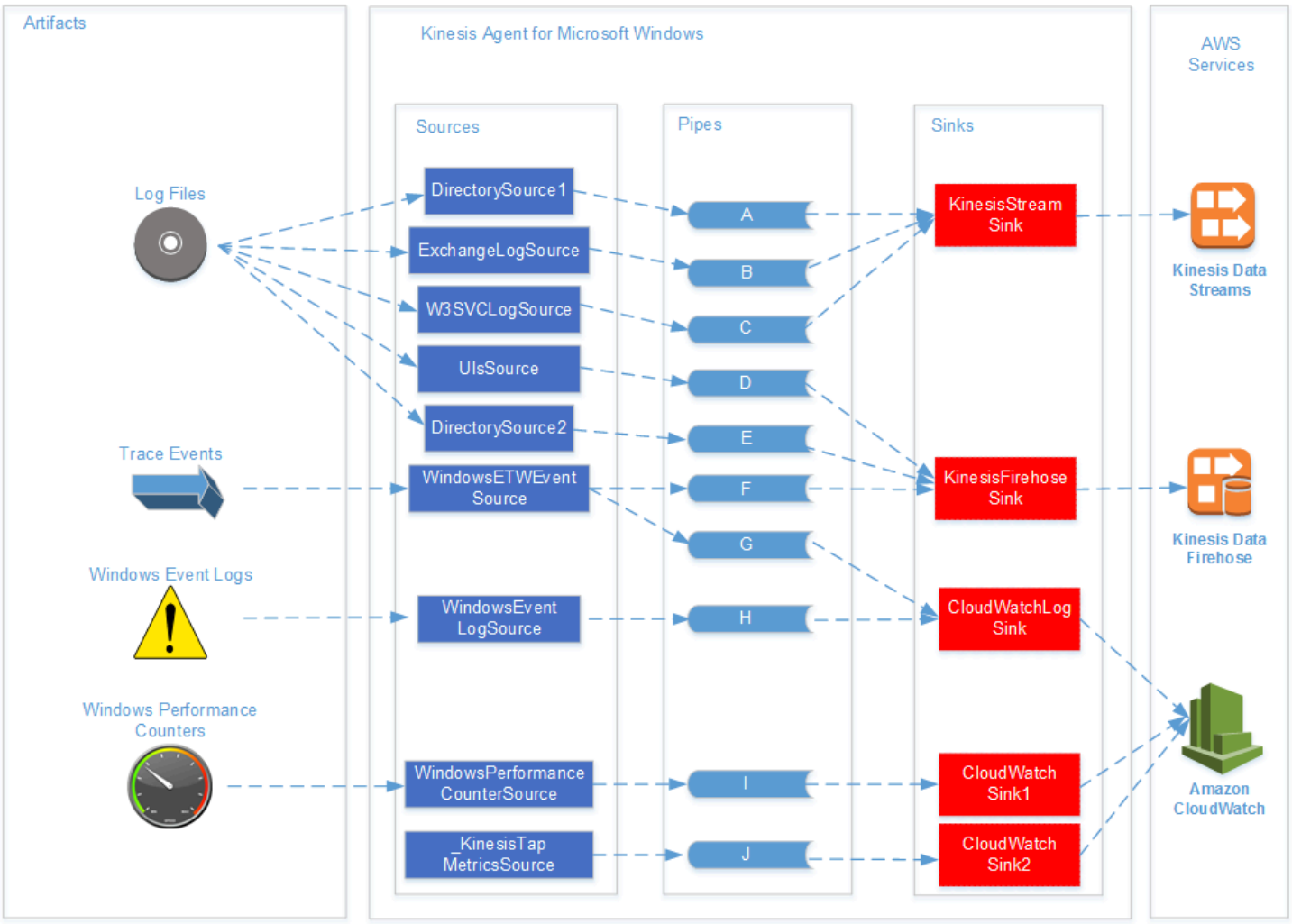
Introducción a Kinesis Agent para Windows

Para obtener información acerca de Kinesis Agent para Windows, le recomendamos que empiece por las siguientes secciones:

- Conceptos de Amazon Kinesis Agent para Microsoft Windows
- Introducción a Amazon Kinesis Agent para Microsoft Windows

Conceptos de Amazon Kinesis Agent para Microsoft Windows

Conocer los conceptos clave de Amazon Kinesis Agent para Microsoft Windows (Kinesis Agent para Windows) puede ayudar a recopilar y transmitir datos que se encuentran en flotas de servidores y equipos de escritorio al resto de la canalización de datos para su procesamiento.



En este diagrama de una canalización de datos, se ilustran los siguientes componentes y procesos:

Los servidores y escritorios tienen artefactos como archivos de registro, eventos y métricas recopilados por uno o más agentes Kinesis para Windowssources. Los datos pueden transformarse después; por ejemplo, de un formato de texto sin formato a un objeto.

Los datos (ya sea en forma de objeto o de texto) pueden fluir a uno o más agentes de Kinesis para WindowsCanalizaciones. Una canalización conecta una fuente a un agente Kinesis para

Windowssink. La canalización también puede filtrar los datos para descartar los que no son necesarios.

Los receptores pueden tomar los datos convertidos en objetos y transformarlos a su vez en formato JSON o XML. El receptor envía los datos a un servicio de AWS específico, como Kinesis Data Streams, Kinesis Data Firehose o Amazon CloudWatch.

Si se usan varias canalizaciones, un solo origen puede enviar los mismos datos a varios receptores (consulte, por ejemplo, las canalizaciones F y G del diagrama). Si se usan varias canalizaciones, también puede ocurrir que diversos orígenes transmitan datos a un mismo receptor (consulte, por ejemplo, las canalizaciones A, B y C del diagrama). También se pueden utilizar varias canalizaciones para transmitir datos desde diversos receptores a diversos orígenes. Los orígenes, los receptores y las canalizaciones tienen diferentes tipos y puede haber varios orígenes, receptores o canalizaciones del mismo tipo.

Para ver ejemplos de los archivos de configuración en los que se declaran orígenes, receptores y canalizaciones, consulte Ejemplos de configuración de Kinesis Agent para Windows.

Temas

Canalizaciones de datos

ACanalización de datosse utiliza para recopilar, procesar, visualizar y posiblemente generar alarmas para aplicaciones y servicios. Kinesis Agent para Windows encaja en las canalizaciones de datos al principio, donde se recopilan registros, eventos y métricas de flotas de equipos de escritorio o servidores. Kinesis Agent para Windows transmite los datos recopilados a los diversos servicios de AWS que forman el resto de la canalización de datos. Una canalización de datos tiene una finalidad específica; por ejemplo, ver el estado de un determinado servicio en tiempo real para ayudar a los ingenieros a administrar el servicio de forma más eficaz. Una canalización de datos sobre el estado de un servicio puede hacer lo siguiente:

- Alertar a los ingenieros de problemas antes de que estos afecten a la experiencia de los clientes con los servicios.

- Ayudar a los ingenieros a administrar con eficacia el costo del servicio mostrando tendencias de uso de recursos. Estas tendencias les permitan ajustar los niveles de los recursos de forma adecuada o incluso implementar escenarios de escalado automático.
- Proporcionar información útil sobre la causa raíz de los problemas que notifican los clientes del servicio. Esto permite acelerar la resolución de los problemas y reducir los costos de soporte.

Para ver un ejemplo paso a paso acerca de la creación de una canalización de datos con el agente de Kinesis para Windows, consulteTutorial: Transmitir archivos de registro JSON a Amazon S3 mediante Kinesis Agent para Windows.

Sources

Un agente Kinesis para Windowsorigenrecopila registros, eventos o métricas. Un origen recopila un tipo concreto de datos de un determinando productor en función del tipo de origen. Por ejemplo, el tipo `DirectorySource` recopila archivos de registro de determinados directorios del sistema de archivos. Si los datos aún no está estructurados (como ocurre con algunos tipos de archivos de registro), los orígenes pueden resultar útiles para convertir la representación textual de los datos en un formato estructurado. Cada fuente se corresponde con unDeclaración de origenen el agente Kinesis para Windows`appsettings.json`Archivo de configuración de. La declaración del origen proporciona detalles esenciales sobre la configuración del origen para adaptarlo a los requisitos de recopilación específicos. El tipo de detalles que se pueden configurar varía en función del tipo de origen. Por ejemplo, el tipo de origen `DirectorySource` requiere que se especifique el directorio en el que se encuentran los archivos de registro.

Para obtener más información sobre los tipos y las declaraciones de orígenes, consulte Declaraciones de origen.

Sinks

Un agente Kinesis para Windowssinktoma los datos recopilados por un origen de Kinesis Agent para Windows y transmite esos datos a uno de los diversos servicios de AWS posibles que forman el resto de la canalización de datos. Cada receptor se corresponde con unDeclaración de receptoresen el agente Kinesis para Windows`appsettings.json`Archivo de configuración de. La declaración del receptor proporciona detalles esenciales sobre la configuración del receptor para adaptarlo a los requisitos de transmisión específicos. El tipo de detalles que se pueden configurar varía en función del tipo de receptor. Por ejemplo, algunos tipos de receptores permiten que una declaración de

receptor especifique una determinada serialización `Format` para que se realice en los datos que les llegan. Cuando esta opción está especificada en la declaración del receptor, la serialización de los datos recopilados se produce antes de estos datos se transmitan al servicio de AWS asociado al receptor.

Para obtener más información sobre los tipos y declaraciones de receptores, consulte Declaraciones de receptores.

Pipes

Un agente Kinesis para WindowsCanalizacionesConecta la salida de un origen de Kinesis Agent para Windows con la entrada de un receptor de Kinesis Agent para Windows. También pueden transformar los datos a medida que se transmiten a la canalización. Cada canalización se corresponde con una determinada declaración de canalización en el agente de Kinesis para Windows`appsettings.json`Archivo de configuración de. La declaración de la canalización proporciona detalles esenciales sobre la configuración, como el origen y el receptor de la canalización.

Para obtener más información sobre los tipos y declaraciones de canalizaciones, consulte Declaraciones de canalizaciones.

Introducción a Amazon Kinesis Agent para Microsoft Windows

Puede utilizar Amazon Kinesis Agent para Microsoft Windows (Kinesis Agent para Windows) para recopilar, analizar, transformar y transmitir registros, eventos y métricas de su flota de Windows a diversos servicios de AWS. La información que se incluye a continuación contiene requisitos previos e instrucciones paso a paso para instalar y configurar Kinesis Agent para Windows.

Temas

Prerequisites

Antes de instalar Kinesis Agent para Windows, asegúrese de que cumple los siguientes requisitos previos:

- Familiaridad con los conceptos de Kinesis Agent para Windows. Para obtener más información, consulte Conceptos de Amazon Kinesis Agent para Microsoft Windows.
- Cuenta de AWS para utilizar los distintos servicios de AWS relacionados con la canalización de datos. Para obtener más información sobre la creación y configuración de cuentas de AWS, consulte Configuración de una cuenta de AWS.
- Microsoft .NET Framework 4.6 o una versión posterior en todos los equipos o servidores que ejecutan Kinesis Agent para Windows. Para obtener más información, consulte Instalación de .NET Framework para desarrolladores en la documentación de Microsoft .NET.

  Para determinar la versión más reciente de.NET Framework que está instalada en un equipo o servidor, utilice el siguiente script de PowerShell:

```
[System.Version](
(Get-ChildItem 'HKLM:\SOFTWARE\Microsoft\NET Framework Setup\NDP' -recurse `
| Get-ItemProperty -Name Version -ErrorAction SilentlyContinue `
```

```
    | Where-Object { ($_.PSChildName -match 'Full') } `
    | Select-Object Version | Sort-Object -Property Version -Descending)[0]).Version
```

- Tiene las secuencias con las que desea enviar datos desde Kinesis Agent para Windows (si utiliza Amazon Kinesis Data Streams). Cree las transmisiones utilizando la herramientaConsola de Kinesis Data Streams, elAWS CLI, o bienHerramientas de AWS para Windows PowerShell. Para obtener más información, consulteCreación y actualización de secuencias de datosen laGuía del desarrollador de Amazon Kinesis Data Streams.
- La entrega de Firehose muestra las secuencias con las que desea enviar datos desde Kinesis Agent para Windows (si utiliza Amazon Kinesis Data Firehose). Cree flujos de entrega con la herramientaConsola Kinesis Data Firehose, elAWS CLI, o bienHerramientas de AWS para Windows PowerShell. Para obtener más información, consulte la sección sobre creación de una secuencia de entrega de Amazon Kinesis Data Firehose en la guía para desarrolladores de Amazon Kinesis Data Firehose.

Configuración de una cuenta de AWS

Si no dispone de una cuenta de AWS, siga estos pasos para crear una.

Para inscribirse en una cuenta de AWS

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

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

Para crearse usted mismo un usuario administrador y agregarlo a un grupo de administradores (consola)

1. Inicie sesión en la consola de IAM como propietario de la cuenta seleccionando Root user (Usuario raíz) y escribiendo la dirección de correo electrónico de su cuenta de AWS. En la siguiente página, escriba su contraseña.

> **Note**
>
> Le recomendamos que siga la práctica recomendada de utilizar el comando de**Administrator**Usuario de IAM que sigue y bloquea de forma segura las credenciales de usuario raíz. Inicie sesión como usuario raíz únicamente para realizar algunas tareas de administración de servicios y de cuentas.

2. En el panel de navegación, elija Users (Usuarios) y, a continuación, elija Add user (Añadir usuario).
3. En User name (Nombre de usuario), escriba **`Administrator`**.
4. Active la casilla de verificación situada junto al AWS Management Console access (Acceso a la consola de administración de AWS). A continuación, seleccione Custom password (Contraseña personalizada) y luego escriba la nueva contraseña en el cuadro de texto.
5. (Opcional) De forma predeterminada, AWS requiere que el nuevo usuario cree una nueva contraseña la primera vez que inicia sesión. Puede quitar la marca de selección de la casilla de verificación situada junto a User must create a new password at next sign-in (El usuario debe crear una nueva contraseña en el siguiente inicio de sesión) para permitir al nuevo usuario restablecer su contraseña después de iniciar sesión.
6. SeleccionarSiguiente: Permisos.
7. En Set permissions (Establecer persmisos), elija Add user to group (Añadir usuario a grupo).
8. Elija Create group (Crear grupo).
9. En el cuadro de diálogo Create group (Crear grupo), en Group name (Nombre del grupo) escriba **`Administrators`**.
10. SeleccionarPolíticas de filtradoY, a continuación, seleccioneAWS administrado - función de trabajopara filtrar el contenido de la tabla.
11. En la lista de políticas, active la casilla de verificación AdministratorAccess. A continuación, elija Create group (Crear grupo).

> **Note**
>
> Debe activar el acceso de usuario y rol de IAM a Facturación antes de poder utilizar los permisos `AdministratorAccess` para acceder a la consola de AWS Billing and Cost Management. Para ello, siga las instrucciones que se indican en el paso 1 del tutorial sobre cómo delegar el acceso a la consola de facturación.

12. Retroceda a la lista de grupos y active la casilla de verificación del nuevo grupo. Elija Refresh si es necesario para ver el grupo en la lista.
13. SeleccionarSiguiente: Tags (Etiquetas):.
14. (Opcional) Añadir metadatos al rol asociando las etiquetas como pares de clave-valor. Para obtener más información sobre el uso de etiquetas en IAM, consulte Etiquetado de entidades de IAM en la guía del usuario de IAM.
15. SeleccionarSiguiente: Review (Revisar)Para ver la lista de suscripciones a grupos que se van a añadir al nuevo usuario. Cuando esté listo para continuar, elija Create user (Crear usuario).

Puede usar este mismo proceso para crear más grupos y usuarios, y para conceder a los usuarios acceso los recursos de su cuenta de AWS. Para obtener información sobre cómo usar las políticas que restringen los permisos de los usuarios a recursos de AWS específicos, consulte Administración de acceso y Políticas de ejemplo.

Para registrarse en AWS y crear una cuenta de administrador

1. Si no tiene una cuenta de AWS, consultehttps://aws.amazon.com/. Elija Cree una cuenta de AWS y siga las instrucciones online.

   Parte del procedimiento de inscripción consiste en recibir una llamada telefónica e introducir un número PIN con el teclado del teléfono.
2. Inicie sesión en la consola de administración de AWS y abra la consola de IAM en https://console.aws.amazon.com/iam/.
3. En el panel de navegación, elija Groups (Grupos) y, a continuación, elija Create New Group (Crear nuevo grupo).
4. En Group Name (Nombre de grupo), escriba el nombre del grupo, como **`Administrators`**, y seleccione Next Step (Paso siguiente).
5. En la lista de políticas, seleccione la casilla de verificación situada junto a la política AdministratorAccess. Puede utilizar el menú de Filter (filtros) y el cuadro de Search (búsqueda) para filtrar la lista de políticas.
6. Elija Next Step (Paso siguiente). Seleccione Create Group (Crear grupo). El nuevo grupo aparecerá debajo de Group Name (Nombre de grupo).
7. En el panel de navegación, seleccione Users y después Create New Users.

8. En el cuadro 1, escriba un nombre de usuario, desactive la casilla situada junto a Generate an access key for each user (Generar una clave de acceso para cada usuario) y seleccione Create (Crear).
9. En la lista de usuarios, elija el nombre (no la casilla) del usuario que acaba de crear. Puede utilizar el cuadro Search (Buscar) para buscar el nombre de usuario.
10. Seleccione la pestaña Groups (Grupos) y haga clic en Add User to Groups (Añadir usuario a grupos).
11. Active la casilla situada junto al grupo de administradores y seleccione Add to Groups (Añadir a grupos).
12. Elija la pestaña Security credentials (Credenciales de seguridad). En Sign-In Credentials (Credenciales de inicio de sesión), elija Manage Password (Administrar contraseña).
13. Seleccione Assign a custom password (Asignar una contraseña personalizada), especifique una contraseña en los cuadros Password (Contraseña) y Confirm Password (Confirmar contraseña) y haga clic en Apply (Aplicar).

Instalación de Kinesis Agent para Windows

Hay tres formas de instalar Kinesis Agent para Windows:

- Instale con MSI (un paquete de instalación de Windows).
- Instalar desdeAWS Systems Manager, un conjunto de servicios para la administración de servidores y equipos de escritorio.
- Ejecutar un script de PowerShell.

> **Note**
>
> En las siguientes instrucciones, se utilizan de forma ocasional los términos KinesisTap y AWSKinesisTap. Estas palabras significan lo mismo que Kinesis Agent para Windows, pero debe especificarlas tal y como están cuando ejecute estas instrucciones.

Instalar Kinesis Agent para Windows con MSI

Puede descargar el paquete MSI de Kinesis Agent para Windows desde larepositorio kinesis-agent-windows en GitHub. Después de descargar el MSI, use Windows para iniciarlo y siga las

instrucciones del instalador. Después de la instalación, puede desinstalar como lo haría con cualquier aplicación de Windows.

También puede utilizar la herramienta de comandomsiexecEn el símbolo del sistema de Windows para instalar de forma silenciosa, activar el registro y desinstalar tal y como se muestra en los siguientes ejemplos. Reemplazar`AWSKinesisTap.1.1.216.4.msi` `with the appropriate version of Kinesis Agent for Windows for your application.`

Para instalar Kinesis Agent para Windows de forma silenciosa:

```
msiexec /i AWSKinesisTap.1.1.216.4.msi /q
```

Para registrar mensajes de instalación para solucionar problemas en un archivo llamado***logfile.log***:

```
msiexec /i AWSKinesisTap.1.1.216.4.msi /q /L*V logfile.log
```

Para desinstalar Kinesis Agent para Windows mediante el símbolo del sistema:

```
msiexec.exe /x {ADAB3982-68AA-4B45-AE09-7B9C03F3EBD3} /q
```

Instalar Kinesis Agent para Windows mediante AWS Systems Manager

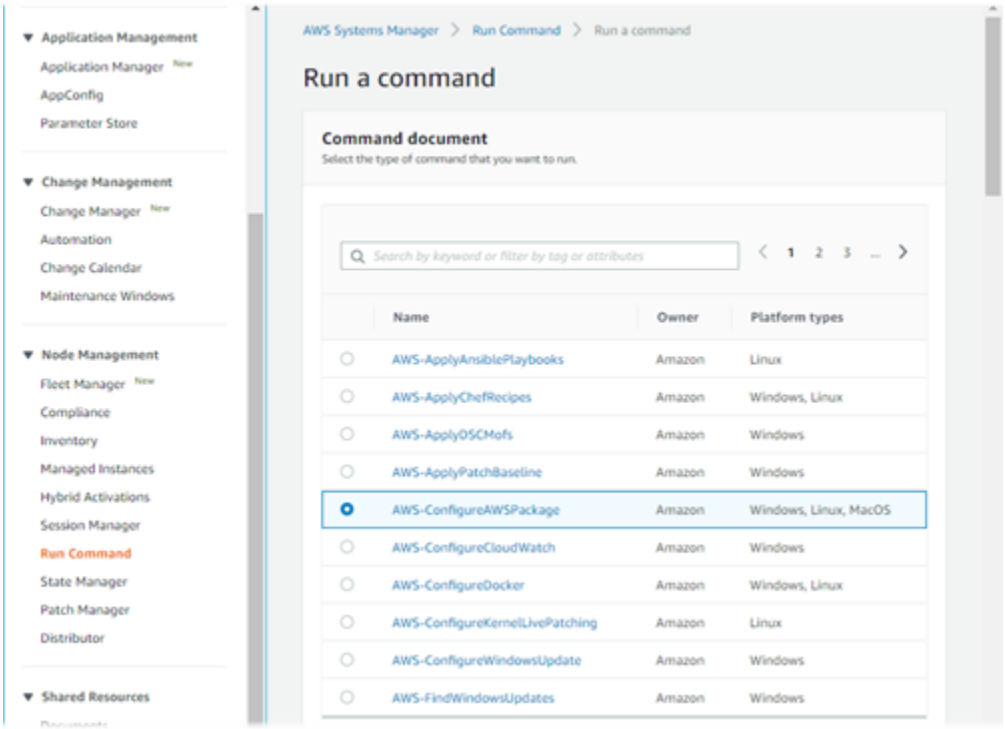
Siga estos pasos para instalar Kinesis Agent para Windows mediante el comando de ejecución de Systems Manager Run Command. Para obtener más información acerca de Run Command, consulteAWS Systems Manager Run Commanden laGuía del usuario de AWS Systems Manager. Además de utilizar Systems Manager Run Command, también puede utilizar Systems ManagerPeríodos de mantenimientoyState Managerpara automatizar la implementación de Kinesis Agent para Windows a lo largo del tiempo.

> **Note**
>
> La instalación de Systems Manager para Kinesis Agent para Windows está disponible en las regiones de AWS que se indican en la lista de AWSAWS Systems Managerexcepto las siguientes:
>
> - cn-north-1
> - cn-northwest-1

- Todas las regiones de AWS GovCloud.

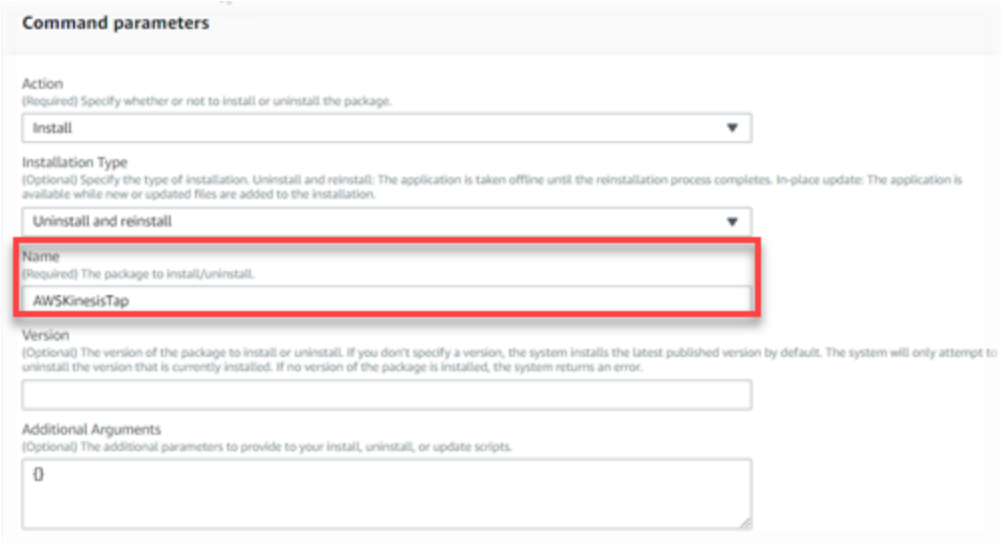
Para instalar Kinesis Agent para Windows mediante el Systems Manager

1. Asegúrese de que la versión 2.2.58.0 o posterior del agente de SSM de está instalada en las instancias en las que desea instalar Kinesis Agent para Windows. Para obtener más información, consulteInstalación y configuración del agente de SSM en instancias de Windowsen laGuía del usuario de AWS Systems Manager.
2. Abra la consola de AWS Systems Manager enhttps://console.aws.amazon.com/systems-manager/.
3. En el panel de navegación, en la sección deAdministración de nodos, elijaRun CommandY luego seleccioneRun Command.
4. Desde laDocumento de comandos, seleccione la casillaAWS-ConfigureAWSPackageno válido.



5. UNDERParámetros de comando, paraNombre, escribaAWSKINISTAP. Deje otras opciones con sus valores predeterminados.

> **Note**
>
> DejeVersionPara especificar la última versión del paquete AWSKinesistap. Si lo desea, también puede introducir una versión específica para instalar.



6. UNDERImplementación, especifique las instancias en las que desea ejecutar el comando. Puede elegir especificar instancias basadas en etiquetas asociadas a instancias, puede elegir instancias manualmente o puede especificar un grupo de recursos que incluya instancias.
7. Deje el resto de opciones con sus valores predeterminados y seleccioneEjecución de.

Instalar el agente Kinesis para Windows mediante PowerShell

Utilice un editor de texto para copiar los siguientes comandos en un archivo y guardarlo como un script de PowerShell. Utilizamos`InstallKinesisAgent.ps1`En el siguiente ejemplo.

```
Param(
    [ValidateSet("prod", "beta", "test")]
    [string] $environment = 'prod',
    [string] $version,
    [string] $baseurl
)

# Self-elevate the script if required.
```

```
if (-Not ([Security.Principal.WindowsPrincipal]
 [Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltI
 'Administrator')) {
    if ([int](Get-CimInstance -Class Win32_OperatingSystem | Select-Object -
ExpandProperty BuildNumber) -ge 6000) {
        $CommandLine = '-File "' + $MyInvocation.MyCommand.Path + '" ' +
 $MyInvocation.UnboundArguments
        Start-Process -FilePath PowerShell.exe -Verb Runas -ArgumentList $CommandLine
        Exit
    }
}

# Allows input to change base url. Useful for testing.
if ($baseurl) {
    if (!$baseUrl.EndsWith("/")) {
        throw "Invalid baseurl param value. Must end with a trailing forward slash
 ('/')"
    }

    $kinesistapBaseUrl = $baseurl
} else {
    $kinesistapBaseUrl = "https://s3-us-west-2.amazonaws.com/kinesis-agent-windows/
downloads/"
}

Write-Host "Using $kinesistapBaseUrl as base url"

$webClient = New-Object System.Net.WebClient

try {
    $packageJson = $webClient.DownloadString($kinesistapBaseUrl + 'packages.json' + '?
_t=' + [System.DateTime]::Now.Ticks) | ConvertFrom-Json
} catch {
    throw "Downloading package list failed."
}



if ($version) {
    $kinesistapPackage = $packageJson.packages | Where-Object { $_.packageName -eq
 "AWSKinesisTap.$version.nupkg" }

    if ($null -eq $kinesistapPackage) {
        throw "No package found matching input version $version"
    }
```

```
} else {
    $packageJson = $packageJson.packages | Where-Object { $_.packageName -match
 ".nupkg" }
    $kinesistapPackage = $packageJson[0]
}

$packageName = $kinesistapPackage.packageName
$checksum = $kinesistapPackage.checksum

#Create %TEMP%/kinesistap if not exists
$kinesistapTempDir = Join-Path $env:TEMP 'kinesistap'
if (![System.IO.Directory]::Exists($kinesistapTempDir)) {[void]
[System.IO.Directory]::CreateDirectory($kinesistapTempDir)}

#Download KinesisTap.x.x.x.x.nupkg package
$kinesistapNupkgPath = Join-Path $kinesistapTempDir $packageName
$webClient.DownloadFile($kinesistapBaseUrl + $packageName, $kinesistapNupkgPath)
$kinesistapUnzipPath = $kinesistapNupkgPath.Replace('.nupkg', '')

# Calculates hash of downloaded file. Downlevel compatible using .Net hashing on PS < 4
if ($PSVersionTable.PSVersion.Major -ge 4) {
    $calculatedHash = Get-FileHash $kinesistapNupkgPath -Algorithm SHA256
    $hashAsString = $calculatedHash.Hash.ToLower()
} else {
    $sha256 = New-Object System.Security.Cryptography.SHA256CryptoServiceProvider
    $calculatedHash =
 [System.BitConverter]::ToString($sha256.ComputeHash([System.IO.File]::ReadAllBytes($kinesistap
    $hashAsString = $calculatedHash.Replace("-", "").ToLower()
}

if ($checksum -eq $hashAsString) {
 Write-Host 'Local file hash matches checksum.' -ForegroundColor Green
} else {
 throw ("Get-FileHash does not match! Package may be corrupted.")
}

#Delete Unzip path if not empty
if ([System.IO.Directory]::Exists($kinesistapUnzipPath)) {Remove-Item –Path
 $kinesistapUnzipPath -Recurse -Force}

#Unzip KinesisTap.x.x.x.x.nupkg package
$null =
 [System.Reflection.Assembly]::LoadWithPartialName('System.IO.Compression.FileSystem')
```

```
[System.IO.Compression.ZipFile]::ExtractToDirectory($kinesistapNupkgPath,
 $kinesistapUnzipPath)

#Execute chocolaeyInstall.ps1 in the package and wait for completion.
$installScript = Join-Path $kinesistapUnzipPath '\tools\chocolateyInstall.ps1'
& $installScript

# Verify service installed.
$serviceName = 'AWSKinesisTap'
$service = Get-Service -Name $serviceName -ErrorAction Ignore
if ($null -eq $service) {
    throw ("Service not installed correctly.")
} else {
    Write-Host "Kinesis Tap Installed." -ForegroundColor Green
    Write-Host "After configuring run the following to start the service: Start-Service
 -Name $serviceName." -ForegroundColor Green
}
```

Abra una ventana del símbolo del sistema con permisos elevados. En el directorio en el que descargó el archivo, utilice el comando siguiente para ejecutar el script:

```
PowerShell.exe -File ".\InstallKinesisAgent.ps1"
```

Para instalar una versión específica de Kinesis Agent para Windows, agregue el comando`-version`Opción:

```
PowerShell.exe -File ".\InstallKinesisAgent.ps1" -version "version"
```

Reemplazar*version*con un número de versión válido de Kinesis Agent para Windows. Para obtener información sobre la versión, consulte larepositorio kinesis-agent-windows en GitHub.

Existen muchas herramientas de implementación que pueden ejecutar scripts de PowerShell de forma remota. Pueden utilizarse para automatizar la instalación de Kinesis Agent para Windows en flotas de equipos de escritorio o servidores.

Configuración e inicio del agente Kinesis para Windows

Después de instalar Kinesis Agent para Windows, debe configurar e iniciar el agente. Una vez hecho esto, no será necesario realizar ninguna otra operación.

Para configurar e iniciar Kinesis Agent para Windows

1. Cree e implemente un archivo de configuración de Kinesis Agent para Windows. Este archivo configura los orígenes, los receptores y las canalizaciones, junto con otros elementos de configuración globales.

   Para obtener más información acerca de la configuración de Kinesis Agent para Windows, consulteConfiguración de Amazon Kinesis Agent para Microsoft Windows.

   Para completar ejemplos de archivos de configuración que puede personalizar e instalar, consulte Ejemplos de configuración de Kinesis Agent para Windows.
2. Abra una ventana de símbolo del sistema de PowerShell con privilegios elevados e inicie Kinesis Agent para Windows con el siguiente comando de PowerShell:

   ```
   Start-Service -Name AWSKinesisTap
   ```

Configuración de Amazon Kinesis Agent para Microsoft Windows

Antes de iniciar Amazon Kinesis Agent para Microsoft Windows, debe crear un archivo de configuración e implementarlo. El archivo de configuración proporciona la información necesaria para recopilar, transformar y transmitir datos de los servidores y equipos de escritorio Windows a distintos servicios de AWS. Los archivos de configuración definen conjuntos de orígenes, receptores y canalizaciones que conectan orígenes con receptores, así como otras transformaciones opcionales.

El archivo de configuración de Kinesis Agent para Windows se denomina`appsettings.json`. Implemente este archivo en `%PROGRAMFILES%\Amazon\AWSKinesisTap`.

Temas

Estructura de configuración básica

La estructura básica del archivo de configuración de Amazon Kinesis Agent para Microsoft Windows es un documento JSON con la siguiente plantilla:

```
{
     "Sources": [ ],
     "Sinks": [ ],
     "Pipes": [ ]
}
```

- El valor de `Sources` es uno o varios Declaraciones de origen.
- El valor de `Sinks` es uno o varios Declaraciones de receptores.
- El valor de `Pipes` es uno o varios Declaraciones de canalizaciones.

Para obtener más información sobre los conceptos de origen, canalización y receptor de Kinesis Agent para Windows, consulteConceptos de Amazon Kinesis Agent para Microsoft Windows.

El siguiente ejemplo es un`appsettings.json`Configure para que transmita eventos de registro de la aplicación de Windows a Kinesis Data Firehose.

```
{
  "Sources": [
    {
      "LogName": "Application",
      "Id": "ApplicationLog",
      "SourceType": "WindowsEventLogSource"
    }
  ],
  "Sinks": [
    {
      "StreamName": "ApplicationLogFirehoseStream",
      "Region": "us-west-2",
      "Id": "MyKinesisFirehoseSink",
      "SinkType": "KinesisFirehose"
    }
  ],
  "Pipes": [
    {
      "Id": "ApplicationLogTotestKinesisFirehoseSink",
      "SourceRef": "ApplicationLog",
      "SinkRef": "MyKinesisFirehoseSink"
    }
  ]
}
```

Para obtener información sobre cada tipo de declaración, consulte las secciones siguientes:

- Declaraciones de origen
- Declaraciones de receptores
- Declaraciones de canalizaciones

Diferenciación entre mayúsculas y minúsculas en la configuración

Por lo general, los archivos con formato JSON distinguen entre mayúsculas y minúsculas, pero no debe presuponer que todas las claves y los valores de los archivos de configuración de Kinesis

Agent para Windows de también lo hacen. Algunas claves y valores del archivo de configuración `appsettings.json` no distinguen entre mayúsculas y minúsculas; por ejemplo:

- El valor del par clave-valor `Format` de los receptores. Para obtener más información, consulte Declaraciones de receptores.
- El valor del par clave-valor `SourceType` de los orígenes, el par clave-valor `SinkType` de los receptores y el par clave-valor `Type` de las canalizaciones y los complementos.
- El valor del par clave-valor `RecordParser` del origen `DirectorySource`. Para obtener más información, consulte Configuración de DirectorySource.
- El valor del par clave-valor `InitialPosition` de los orígenes. Para obtener más información, consulte Configuración de Bookmark.
- Los prefijos de las sustituciones de variables. Para obtener más información, consulte Configuración de sustituciones de variables de receptor.

Declaraciones de origen

En Amazon Kinesis Agent para Microsoft Windows,Declaraciones de origenDescriba dónde y qué datos de registro, eventos y métrica deben recopilarse. También brindan la posibilidad de especificar información que permita analizar esos datos de modo que puedan transformarse. En las secciones siguientes, se describen las configuraciones de los tipos de origen integrados que están disponibles en Kinesis Agent para Windows. Como Kinesis Agent para Windows es ampliable, se pueden agregar tipos de orígenes personalizados. Normalmente, todos los tipos de orígenes necesitan pares clave-valor específicos de los objetos de configuración que son importantes para cada tipo de origen.

Todas las declaraciones de origen deben tener al menos los siguientes pares clave-valor:

`Id`

Cadena única que identifica un determinado objeto de origen en el archivo de configuración.

`SourceType`

Nombre del tipo de origen de este objeto de origen. El tipo de origen especifica el origen de los datos del registro, el evento o la métrica que este objeto de origen recopila. También controla qué otros aspectos del origen pueden declararse.

Para ver ejemplos de archivos de configuración completos en los que se utilizan diferentes tipos de declaraciones de origen, consulte Transmisión a desde varios orígenes de Kinesis Data Streams.

Temas

Configuración de DirectorySource

Overview

El tipo de origen `DirectorySource` recopila registros de archivos que están almacenados en el directorio especificado. Como los archivos de registro están disponibles en muchos formatos diferentes, la declaración `DirectorySource` permite especificar el formato de los datos del archivo de registro. Posteriormente, el contenido del registro se puede convertir a un formato estándar, como JSON o XML, antes de transmitirlo a los diferentes servicios de AWS.

A continuación, se muestra una declaración `DirectorySource` de ejemplo:

```
{
    "Id": "myLog",
    "SourceType": "DirectorySource",
    "Directory": "C:\\Program Data\\MyCompany\\MyService\\logs",
    "FileNameFilter": "*.log",
    "IncludeSubdirectories": true,
    "IncludeDirectoryFilter": "cpu\\cpu-1;cpu\\cpu-2;load;memory",
    "RecordParser": "Timestamp",
    "TimestampFormat": "yyyy-MM-dd HH:mm:ss.ffff",
    "Pattern": "\\d{4}-\\d{2}-\\d(2}",
    "ExtractionPattern": "",
```

```
    "TimeZoneKind": "UTC",
    "SkipLines": 0,
    "Encoding": "utf-16",
    "ExtractionRegexOptions": "Multiline"
}
```

Todas las declaraciones `DirectorySource` pueden proporcionar los siguientes pares clave-valor:

`SourceType`

Tiene que ser la cadena literal `"DirectorySource"` (obligatorio).

`Directory`

Ruta del directorio que contiene los archivos de registro (obligatorio).

`FileNameFilter`

Es opcional y limita el conjunto de archivos del directorio donde se van a recopilar los datos de registro en función de un patrón de nombre de archivo con caracteres comodín. Si tiene varios patrones de nombre de archivo de registro, esta función le permite utilizar un únicoDirectorySourceComo se muestra en el ejemplo siguiente.

```
FileNameFilter: "*.log|*.txt"
```

Los administradores del sistema a veces comprimen los archivos de registro antes de archivarlos. Si especifica`"*.*"`in`FileNameFilter`, ahora se excluyen los archivos comprimidos conocidos. Esta función impide que`.zip`,`.gz`, y`.bz2`se transmitan accidentalmente. Si este par clave-valor no se especifica, los datos de todos los archivos del directorio se recopilan de forma predeterminada.

`IncludeSubdirectories`

Especifica que se supervisan los subdirectorios a una profundidad arbitraria limitada por el sistema operativo. Esta función es útil para monitorear servidores web con múltiples sitios web. También puede utilizar la`IncludeDirectoryFilter`atributo para supervisar sólo ciertos subdirectorios especificados en el filtro.

`RecordParser`

Especifica cómo el tipo de origen `DirectorySource` debe procesar los archivos de registro que se encuentran en el directorio especificado. Este par clave-valor es obligatorio y los valores válidos son los siguientes:

- `SingleLine`— Cada línea del archivo de registro es un registro de registro.
- `SingleLineJson`— Cada línea del archivo de registro es un registro de registro con formato JSON. Este analizador resulta útil si desea añadir más pares clave-valor al archivo JSON utilizando la decoración de objetos. Para obtener más información, consulte Configuración de decoraciones de receptores. Para ver un ejemplo en el que se utiliza el analizador de recursos `SingleLineJson`, consulte Tutorial: Transmitir archivos de registro JSON a Amazon S3 mediante Kinesis Agent para Windows.
- `Timestamp`— Una o más líneas pueden incluir un registro de registro. Esta entrada de registro comienza con una marca temporal. Esta opción requiere que se especifique el par clave-valor `TimestampFormat`.
- `Regex`— Cada registro comienza con texto que coincide con una expresión regular determinada. Esta opción requiere que se especifique el par clave-valor `Pattern`.
- `SysLog`— Indica que el archivo de registro está escrito en el archivosyslogformato estándar. Las entradas del archivo de registro se analizan en función de dicha especificación.
- `Delimited`— Una versión más simple del analizador de registros Regex donde los elementos de datos de los registros están separados por un delimitador consistente. Esta opción resulta más fácil de utilizar y se ejecuta más rápido que el analizador Regex, por lo que es la preferida cuando está disponible. Si utiliza esta opción, debe especificar el par clave-valor `Delimiter`.

`TimestampField`

Especifica qué campo JSON contiene la marca temporal del registro. Solo se utiliza con `SingleLineJson RecordParser`. Este par clave-valor es opcional. Si no se especifica, Kinesis Agent para Windows utiliza como marca temporal el momento en que se leyó el registro. Una ventaja de especificar este par clave-valor es que las estadísticas de latencia generadas por Kinesis Agent para Windows son más precisas.

`TimestampFormat`

Especifica cómo se van a analizar la fecha y la hora asociadas con el registro. El valor es la cadena `epoch` o una cadena con formato de fecha y hora de .NET. Si el valor es `epoch`, el valor temporal se analizará utilizando el sistema temporal UNIX Epoch. Para obtener más información sobre el sistema temporal UNIX Epoch, consulte Unix time (Tiempo Unix). Para obtener más información acerca las cadenas con formato de fecha y hora de .NET, consulte Cadenas con formato de fecha y hora personalizado en la documentación de Microsoft .NET. Este par clave-valor solo es obligatorio si se especifica el analizador de registros `Timestamp` o si se especifica el analizador de `SingleLineJson` registros junto con el par clave-valor `TimestampField`.

`Pattern`

Especifica una expresión regular que debe coincidir con la primera línea de una entrada que podría estar repartida en varias líneas. Este par clave-valor solo es necesario en el analizador de registros `Regex`.

`ExtractionPattern`

Especifica una expresión regular que debe utilizar grupos con nombre. El registro se analiza utilizando esta expresión regular y los grupos con nombre conforman los campos del registro analizado. Estos campos se utilizan después como base para crear objetos JSON, objetos XML o documentos que los receptores transmiten a diversos servicios de AWS. Este par clave-valor es opcional y está disponible con la`Regex`y el analizador de marcas de tiempo.

El nombre del grupo `Timestamp` se procesa de forma diferente, lo que le indica al analizador `Regex` cuál es el campo que contiene la fecha y la hora de cada entrada del archivo de registro.

`Delimiter`

Especifica el carácter o cadena que separa los elementos de cada entrada del registro. Este par clave-valor debe utilizarse (y solamente puede utilizarse) con el analizador de registros `Delimited`. Utilice la secuencia de dos caracteres `\t` para representar el carácter de tabulación.

`HeaderPattern`

Especifica una expresión regular que busca la línea del archivo de registro que contiene el conjunto de encabezados de la entrada del registro. Si el archivo de registro no contiene información sobre el encabezado, utilice el par clave-valor `Headers` para especificar los encabezados implícitos. El par clave-valor `HeaderPattern` es opcional y solo es válido con el analizador de registros `Delimited`.

> **Note**
>
> Si hay una entrada en una columna con un encabezado vacío (la longitud es 0), los datos de esa columna se filtrarán desde la salida final del resultado analizado de `DirectorySource`.

`Headers`

Especifica los nombres de las columnas de datos que se han analizado con el delimitador especificado. Este par clave-valor es opcional y solo es válido con el analizador de registros `Delimited`.

> **Note**
>
> Si hay una entrada en una columna con un encabezado vacío (la longitud es 0), los datos de esa columna se filtrarán desde la salida final del resultado analizado de `DirectorySource`.

`RecordPattern`

Especifica una expresión regular que identifica las líneas del archivo de registro que contienen datos de la entrada del registro. Salvo la línea de encabezado opcional identificada por `HeaderPattern`, las líneas que no coinciden con el valor de `RecordPattern` especificado no se tienen en cuenta durante el procesamiento del registro. Este par clave-valor es opcional y solo es válido con el analizador de registros `Delimited`. Si no se proporciona, de forma predeterminada, se considera que las líneas que no coinciden con el valor opcional de `HeaderPattern` o de `CommentPattern` son líneas que contienen datos de registro que se pueden analizar.

`CommentPattern`

Especifica una expresión regular que identifica las líneas del archivo de registro que deberían excluirse antes de analizar los datos del archivo de registro. Este par clave-valor es opcional y solo es válido con el analizador de registros `Delimited`. Si no se proporciona, de forma predeterminada, se considera que las líneas que no coinciden con el valor opcional de `HeaderPattern` son líneas que contienen datos de registro que se pueden analizar, a menos que se especifique `RecordPattern`.

`TimeZoneKind`

Especifica si la marca temporal del archivo de registro corresponde a la zona horaria local o a la zona horaria UTC. Este valor es opcional y su valor predeterminado es UTC. Los únicos valores válidos para este par clave-valor son `Local` o `UTC`. La marca temporal nunca se modifica si `TimeZoneKind` no se especifica o si el valor es UTC. La marca de tiempo se convierte a UTC cuando el parámetro`TimeZoneKind`Valor es`Local`El receptor que recibe la marca temporal

es CloudWatch Logs o el registro analizado se envía a otros receptores. Las fechas y las horas insertadas en mensajes no se convierten.

`SkipLines`

Cuando se especifica, controla el número de líneas que se omiten al comienzo de cada archivo de registro antes de que se realice el análisis. Es opcional y el valor predeterminado es 0.

Codificación

De forma predeterminada, Kinesis Agent para Windows puede detectar automáticamente la codificación desde bytemark. Sin embargo, es posible que la codificación automática no funcione correctamente en algunos formatos Unicode antiguos. En el ejemplo siguiente se especifica la codificación necesaria para transmitir un registro de Microsoft SQL Server.

```
"Encoding": "utf-16"
```

Para obtener una lista de los nombres de codificación, consulteLista de codificacionesen la documentación de Microsoft.NET.

ExtraccionRegexOptions

Puede usar`ExtractionRegexOptions`Para simplificar las expresiones regulares. Este par clave-valor es opcional. El valor predeterminado es `"None"`.

En el siguiente ejemplo se especifica que la`"."`coincide con cualquier carácter, incluyendo`\r\n`.

```
"ExtractionRegexOptions" = "Multiline"
```

Para obtener una lista de los posibles campos de ExtractionRegexOptions, consulte laRegexOptions enumen la documentación de Microsoft.NET.

Analizador de registros **Regex**

Puede analizar registros de texto no estructurados utilizando el analizador de registros `Regex` con los pares clave-valor `TimestampFormat`, `Pattern` y `ExtractionPattern`. Por ejemplo, supongamos que tiene un archivo de registro similar al siguiente:

```
[FATAL][2017/05/03 21:31:00.534][0x00003ca8][0000059c][][ActivationSubSystem]
[GetActivationForSystemID][0] 'ActivationException.File: EQCASLicensingSubSystem.cpp'
[FATAL][2017/05/03 21:31:00.535][0x00003ca8][0000059c][][ActivationSubSystem]
[GetActivationForSystemID][0] 'ActivationException.Line: 3999'
```

Puede especificar las siguientes expresiones regulares para que el par clave-valor `Pattern` le ayude a dividir el archivo de registro en varios registros distintos:

```
^\[\w+\]\[(?<TimeStamp>\d{4}/\d{2}/\d{2} \d{2}:\d{2}:\d{2}\.\d{3})\]
```

Esta expresión regular coincidirá con la siguiente secuencia:

1. Comienzo de la cadena que se está evaluando.
2. Uno o varios caracteres entre corchetes.
3. Una marca temporal entre corchetes. La marca temporal encontrará coincidencias con la siguiente secuencia:
   a. Un año de cuatro dígitos
   b. Una barra inclinada
   c. Un mes de dos dígitos.
   d. Una barra inclinada
   e. Un día de dos dígitos.
   f. Un carácter de espacio
   g. Una hora de dos dígitos
   h. Un símbolo de dos puntos
   i. Un minuto de dos dígitos
   j. Un símbolo de dos puntos
   k. Un segundo de dos dígitos
   l. Un punto
   m. Un milisegundo de tres dígitos

Puede especificar el siguiente formato para que el par clave-valor `TimestampFormat` convierta el texto de la marca temporal en un formato de fecha y hora:

```
yyyy/MM/dd HH:mm:ss.fff
```

Puede utilizar la siguiente expresión regular para extraer los campos de la entrada de registro mediante el par clave-valor `ExtractionPattern`.

```
^\[(?<Severity>\w+)\]\[(?<TimeStamp>\d{4}/\d{2}/\d{2} \d{2}:\d{2}:\d{2}\.
\d{3})\]\[[^]]*\]\[[^]]*\]\[[^]]*\]\[(?<SubSystem>\w+)\]\[(?<Module>\w+)\]\[[^]]*\] '(?
<Message>.*)'$
```

Esta expresión regular coincidirá con los siguientes grupos en este orden:

1. `Severity`: uno o varios caracteres entre corchetes.
2. `TimeStamp`— Consulte la descripción anterior de la marca de tiempo.
3. Se omiten tres secuencias de cero o más caracteres entre corchetes sin nombre.
4. `SubSystem`: uno o varios caracteres entre corchetes.
5. `Module`: uno o varios caracteres entre corchetes.
6. Se omite una secuencia de cero o más caracteres entre corchetes sin nombre.
7. Se omite un espacio sin nombre.
8. `Message`— Cero o más caracteres rodeados de comillas simples.

La siguiente declaración de origen combina estas expresiones regulares y el formato de fecha y hora a fin de proporcionar instrucciones completas para que analice este tipo de archivo de registro.

```
{
    "Id": "PrintLog",
    "SourceType": "DirectorySource",
    "Directory": "C:\\temp\\PrintLogTest",
    "FileNameFilter": "*.log",
    "RecordParser": "Regex",
    "TimestampFormat": "yyyy/MM/dd HH:mm:ss.fff",
    "Pattern": "^\\[\\w+\\]\\[(?<TimeStamp>\\d{4}/\\d{2}/\\d{2} \\d{2}:\\d{2}:\\d{2}\\.
\\d{3})\\]",
    "ExtractionPattern": "^\\[(?<Severity>\\w+)\\]\\[(?<TimeStamp>\\d{4}/\\d{2}/\\d{2}
 \\d{2}:\\d{2}:\\d{2}\\.\\d{3})\\]\\[[^]]*\\]\\[[^]]*\\]\\[[^]]*\\]\\[(?<SubSystem>\\w
+)\\]\\[(?<Module>\\w+)\\]\\[[^]]*\\] '(?<Message>.*)'$",
    "TimeZoneKind": "UTC"
}
```

> **Note**
>
> Las barras diagonales invertidas de los archivos con formato JSON deben incluirse en secuencias de escape utilizando barras diagonales invertidas adicionales.

Para obtener más información sobre las expresiones regulares, consulte Lenguaje de expresiones regulares - Referencia rápida en la documentación de Microsoft .NET.

Analizador de registros **`Delimited`**

Puede utilizar el analizador de registros `Delimited` para analizar registros semiestructurados y archivos de datos en los que hay una secuencia de caracteres que se utiliza sistemáticamente para separar las columnas de datos de cada fila de datos. Por ejemplo, los archivos CSV utilizan una coma para separar las columnas de datos, mientras que los archivos TSV utilizan una tabulación.

Supongamos que desea analizar un archivo de registro con el formato de base de datos NPS generado por un servidor de políticas de red. Este archivo tendría un aspecto similar al siguiente:

```
"NPS-
MASTER","IAS",03/22/2018,23:07:55,1,"user1","Domain1\user1",,,,,,,,0,"192.168.86.137","Nate
 - Test 1",,,,,,,1,,0,"311 1 192.168.0.213 03/15/2018 08:14:29
 1",,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,"Use Windows authentication for all users",1,,,,
"NPS-
MASTER","IAS",03/22/2018,23:07:55,3,,"Domain1\user1",,,,,,,,0,"192.168.86.137","Nate
 - Test 1",,,,,,,1,,16,"311 1 192.168.0.213 03/15/2018 08:14:29
 1",,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,"Use Windows authentication for all users",1,,,,
```

El archivo de configuración `appsettings.json` del siguiente ejemplo contiene una declaración `DirectorySource` que utiliza el analizador de registros `Delimited` para analizar este texto en una representación de objetos. A continuación, transmite los datos con formato JSON a Kinesis Data Firehose:

```
{
    "Sources": [
        {
            "Id": "NPS",
            "SourceType": "DirectorySource",
            "Directory": "C:\\temp\\NPS",
            "FileNameFilter": "*.log",
```

```
            "RecordParser": "Delimited",
            "Delimiter": ",",
            "Headers": "ComputerName,ServiceName,Record-Date,Record-Time,Packet-
Type,User-Name,Fully-Qualified-Distinguished-Name,Called-Station-ID,Calling-Station-
ID,Callback-Number,Framed-IP-Address,NAS-Identifier,NAS-IP-Address,NAS-Port,Client-
Vendor,Client-IP-Address,Client-Friendly-Name,Event-Timestamp,Port-Limit,NAS-Port-
Type,Connect-Info,Framed-Protocol,Service-Type,Authentication-Type,Policy-Name,Reason-
Code,Class,Session-Timeout,Idle-Timeout,Termination-Action,EAP-Friendly-Name,Acct-
Status-Type,Acct-Delay-Time,Acct-Input-Octets,Acct-Output-Octets,Acct-Session-Id,Acct-
Authentic,Acct-Session-Time,Acct-Input-Packets,Acct-Output-Packets,Acct-Terminate-
Cause,Acct-Multi-Ssn-ID,Acct-Link-Count,Acct-Interim-Interval,Tunnel-Type,Tunnel-
Medium-Type,Tunnel-Client-Endpt,Tunnel-Server-Endpt,Acct-Tunnel-Conn,Tunnel-Pvt-
Group-ID,Tunnel-Assignment-ID,Tunnel-Preference,MS-Acct-Auth-Type,MS-Acct-EAP-Type,MS-
RAS-Version,MS-RAS-Vendor,MS-CHAP-Error,MS-CHAP-Domain,MS-MPPE-Encryption-Types,MS-
MPPE-Encryption-Policy,Proxy-Policy-Name,Provider-Type,Provider-Name,Remote-Server-
Address,MS-RAS-Client-Name,MS-RAS-Client-Version",
            "TimestampField": "{Record-Date} {Record-Time}",
            "TimestampFormat": "MM/dd/yyyy HH:mm:ss"
        }
    ],
    "Sinks": [
        {
            "Id": "npslogtest",
            "SinkType": "KinesisFirehose",
            "Region": "us-west-2",
            "StreamName": "npslogtest",
            "Format": "json"
        }
    ],
    "Pipes": [
        {
            "Id": "W3SVCLog1ToKinesisStream",
            "SourceRef": "NPS",
            "SinkRef": "npslogtest"
        }
    ]
}
```

Los datos con formato JSON que se transmiten a Kinesis Data Firehose tienen un aspecto similar al siguiente:

```
{
    "ComputerName": "NPS-MASTER",
```

```
    "ServiceName": "IAS",
    "Record-Date": "03/22/2018",
    "Record-Time": "23:07:55",
    "Packet-Type": "1",
    "User-Name": "user1",
    "Fully-Qualified-Distinguished-Name": "Domain1\\user1",
    "Called-Station-ID": "",
    "Calling-Station-ID": "",
    "Callback-Number": "",
    "Framed-IP-Address": "",
    "NAS-Identifier": "",
    "NAS-IP-Address": "",
    "NAS-Port": "",
    "Client-Vendor": "0",
    "Client-IP-Address": "192.168.86.137",
    "Client-Friendly-Name": "Nate - Test 1",
    "Event-Timestamp": "",
    "Port-Limit": "",
    "NAS-Port-Type": "",
    "Connect-Info": "",
    "Framed-Protocol": "",
    "Service-Type": "",
    "Authentication-Type": "1",
    "Policy-Name": "",
    "Reason-Code": "0",
    "Class": "311 1 192.168.0.213 03/15/2018 08:14:29 1",
    "Session-Timeout": "",
    "Idle-Timeout": "",
    "Termination-Action": "",
    "EAP-Friendly-Name": "",
    "Acct-Status-Type": "",
    "Acct-Delay-Time": "",
    "Acct-Input-Octets": "",
    "Acct-Output-Octets": "",
    "Acct-Session-Id": "",
    "Acct-Authentic": "",
    "Acct-Session-Time": "",
    "Acct-Input-Packets": "",
    "Acct-Output-Packets": "",
    "Acct-Terminate-Cause": "",
    "Acct-Multi-Ssn-ID": "",
    "Acct-Link-Count": "",
    "Acct-Interim-Interval": "",
    "Tunnel-Type": "",
```

```
    "Tunnel-Medium-Type": "",
    "Tunnel-Client-Endpt": "",
    "Tunnel-Server-Endpt": "",
    "Acct-Tunnel-Conn": "",
    "Tunnel-Pvt-Group-ID": "",
    "Tunnel-Assignment-ID": "",
    "Tunnel-Preference": "",
    "MS-Acct-Auth-Type": "",
    "MS-Acct-EAP-Type": "",
    "MS-RAS-Version": "",
    "MS-RAS-Vendor": "",
    "MS-CHAP-Error": "",
    "MS-CHAP-Domain": "",
    "MS-MPPE-Encryption-Types": "",
    "MS-MPPE-Encryption-Policy": "",
    "Proxy-Policy-Name": "Use Windows authentication for all users",
    "Provider-Type": "1",
    "Provider-Name": "",
    "Remote-Server-Address": "",
    "MS-RAS-Client-Name": "",
    "MS-RAS-Client-Version": ""
}
```

Analizador de registros **`SysLog`**

En el caso del analizador de registros `SysLog`, la salida analizada del origen contiene la siguiente información:

Atributo	Tipo	Descripción
`SysLogTimeStamp`	Cadena	Fecha y hora originales del archivo de registro con formato syslog.
`Hostname`	Cadena	Nombre del equipo en el que se encuentra el archivo de registro con formato syslog.
`Program`	Cadena	Nombre de la aplicación o servicio que generó el archivo de registro.

Atributo	Tipo	Descripción
`Message`	Cadena	Mensaje de registro generado por la aplicación o servicio.
`TimeStamp`	Cadena	Fecha y hora analizadas en formato ISO 8601.

A continuación, se muestra un ejemplo de datos SysLog convertidos a formato JSON:

```
{
    "SysLogTimeStamp": "Jun 18 01:34:56",
    "Hostname": "myhost1.example.mydomain.com",
    "Program": "mymailservice:",
    "Message": "Info: ICID 123456789 close",
    "TimeStamp": "2017-06-18T01:34.56.000"
}
```

Summary

A continuación, se muestra un resumen de los pares clave-valor disponibles para el origen `DirectorySource` y los analizadores `RecordParser` relacionados con estos pares clave-valor.

Nombre de clave	RecordParser	Notas
`SourceType`	Obligatorio en todos los casos	Debe tener el valor `Directory Source`
`Directory`	Obligatorio en todos los casos	
`FileNameFilter`	Opcional en todos los casos	
`RecordParser`	Obligatorio en todos los casos	

Nombre de clave	RecordParser	Notas
`TimestampField`	Opcional para `SingleLineJson`	
`TimestampFormat`	Obligatorio para `Timestamp` ; obligatorio para `SingleLineJson` si se especifica `TimestampField` .	
`Pattern`	Obligatorio para `Regex`	
`ExtractionPattern`	Opcional para `Regex`	Obligatorio para `Regex` si el receptor especifica un formato `json` o `xml`.
`Delimiter`	Obligatorio para `Delimited`	
`HeaderPattern`	Opcional para `Delimited`	
`Headers`	Opcional para `Delimited`	
`RecordPattern`	Opcional para `Delimited`	
`CommentPattern`	Opcional para `Delimited`	
`TimeZoneKind`	Opcional para `Regex`, `Timestamp` , `SysLog` y `SingleLineJson` cuando se identifica un campo de marca temporal.	

Nombre de clave	RecordParser	Notas
`SkipLines`	Opcional en todos los casos	

Configuración de ExchangeLogSource

El tipo `ExchangeLogSource` se utiliza para recopilar registros de Microsoft Exchange. Exchange genera registros con diferentes tipos de formatos de registro. Este tipo de origen analiza todos ellos. Aunque es posible analizarlos utilizando el tipo `DirectorySource` con el analizador de registros `Regex`, resulta mucho más sencillo utilizar `ExchangeLogSource`, ya que no es necesario diseñar y proporcionar expresiones regulares para los formatos de los archivos de registro. A continuación, se muestra una declaración `ExchangeLogSource` de ejemplo:

```
{
   "Id": "MyExchangeLog",
   "SourceType": "ExchangeLogSource",
   "Directory": "C:\\temp\\ExchangeLogTest",
   "FileNameFilter": "*.log"
}
```

Todas las declaraciones de Exchange pueden proporcionar los siguientes pares clave-valor:

`SourceType`

Tiene que ser la cadena literal `"ExchangeLogSource"` (obligatorio).

`Directory`

Ruta del directorio que contiene los archivos de registro (obligatorio).

`FileNameFilter`

Es opcional y limita el conjunto de archivos del directorio donde se van a recopilar los datos de registro en función de un patrón de nombre de archivo con caracteres comodín. Si este par clave-valor no se especifica, de forma predeterminada, se recopilan los datos de registro de todos los archivos del directorio.

`TimestampField`

Nombre de la columna que contiene la fecha y la hora del registro. Este par clave-valor es opcional y no tiene que especificarse si el nombre de campo es `date-time` o `DateTime`. De lo contrario, es obligatorio.

Configuración de W3SVCLogSource

El tipo `W3SVCLogSource` se utiliza para recopilar registros de Internet Information Services (IIS) para Windows.

A continuación, se muestra una declaración `W3SVCLogSource` de ejemplo:

```
{
   "Id": "MyW3SVCLog",
   "SourceType": "W3SVCLogSource",
   "Directory": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
   "FileNameFilter": "*.log"
}
```

Todas las declaraciones `W3SVCLogSource` pueden proporcionar los siguientes pares clave-valor:

`SourceType`

Tiene que ser la cadena literal "`W3SVCLogSource`" (obligatorio).

`Directory`

Ruta del directorio que contiene los archivos de registro (obligatorio).

`FileNameFilter`

Es opcional y limita el conjunto de archivos del directorio donde se van a recopilar los datos de registro en función de un patrón de nombre de archivo con caracteres comodín. Si este par clave-valor no se especifica, de forma predeterminada, se recopilan los datos de registro de todos los archivos del directorio.

Configuración de UlsSource

El tipo `UlsSource` se utiliza para recopilar registros de Microsoft SharePoint. A continuación, se muestra una declaración `UlsSource` de ejemplo:

```
{
    "Id": "UlsSource",
    "SourceType": "UlsSource",
    "Directory": "C:\\temp\\uls",
    "FileNameFilter": "*.log"
}
```

Todas las declaraciones `UlsSource` pueden proporcionar los siguientes pares clave-valor:

`SourceType`

Tiene que ser la cadena literal `"UlsSource"` (obligatorio).

`Directory`

Ruta del directorio que contiene los archivos de registro (obligatorio).

`FileNameFilter`

Es opcional y limita el conjunto de archivos del directorio donde se van a recopilar los datos de registro en función de un patrón de nombre de archivo con caracteres comodín. Si este par clave-valor no se especifica, de forma predeterminada, se recopilan los datos de registro de todos los archivos del directorio.

Configuración de WindowsEventLogSource

El tipo `WindowsEventLogSource` se utiliza para recopilar eventos del servicio Registro de eventos de Windows. A continuación, se muestra una declaración `WindowsEventLogSource` de ejemplo:

```
{
    "Id": "mySecurityLog",
    "SourceType": "WindowsEventLogSource",
    "LogName": "Security"
}
```

Todas las declaraciones `WindowsEventLogSource` pueden proporcionar los siguientes pares clave-valor:

`SourceType`

Tiene que ser la cadena literal `"WindowsEventLogSource"` (obligatorio).

`LogName`

Eventos recopilados en el registro especificado. Los valores habituales son `Application`, `Security` y `System`, pero se puede especificar cualquier nombre de registro de eventos de Windows que sea válido. Este par clave-valor es obligatorio.

`Query`

Es opcional y puede limitar los eventos que se van a incluir en la salida de `WindowsEventLogSource`. Si este par clave-valor no se especifica, de forma predeterminada, se incluyen todos los eventos en la salida. Para obtener más información sobre la sintaxis de este valor, consulte Event Queries and Event XML en la documentación de Windows. Para obtener más información sobre las definiciones de nivel de registro, consulte Event Types en la documentación de Windows.

`IncludeEventData`

Es opcional y permite recopilar y transmitir datos de eventos que son específicos de un proveedor y están asociados con los eventos de un registro de Windows que se ha especificado cuando el valor de este par clave-valor es `"true"`. Solo se incluyen los datos de eventos que pueden serializarse correctamente. Este par clave-valor es opcional y, si no se especifica, no se recopilan datos de eventos específicos de un proveedor.

> **Note**
>
> Si se incluyen datos de eventos, puede aumentar significativamente la cantidad de datos transmitidos desde este origen. El tamaño máximo de un evento puede ser de 262 143 bytes, incluidos sus datos.

La salida analizada de `WindowsEventLogSource` contiene la siguiente información:

Atributo	Tipo	Descripción
`EventId`	Int	Identificador del tipo de evento.
`Description`	Cadena	Texto que describe los detalles del evento.
`LevelDisplayName`	Cadena	Categoría del evento. Puede ser una de las siguientes: Error, Warning

Atributo	Tipo	Descripción
		(Advertencia), Information (Información), Success Audit (Resultados positivos de auditoría) o Failure Audit (Resultados negativos de auditoría).
`LogName`	Cadena	Lugar en el que se registró el evento (los valores más comunes suelen ser `Application` , `Security` y `System`, pero hay muchas otras posibilidades).
`MachineName`	Cadena	Equipo que registró el evento.
`ProviderName`	Cadena	Aplicación o servicio que registró el evento.
`TimeCreated`	Cadena	Momento en que tuvo lugar el evento en formato ISO 8601.
`Index`	Int	Lugar del registro donde se encuentra la entrada.
`UserName`	Cadena	Persona que realizó la entrada, si se sabe quién es.

Atributo	Tipo	Descripción
`Keywords`	Cadena	El tipo de evento. Los valores estándar son `AuditFailure` (eventos de auditoría de seguridad con errores), `AuditSuccess` (eventos de auditoría de seguridad correctos), `Classic` (eventos generados con la función `RaiseEvent`), `Correlation Hint` (eventos de transferencia), `SQM` (eventos del mecanismo de calidad del servicio) , `WDI Context` (eventos contextuales de Infraestructura de diagnóstico de Windows) y `WDI Diag` (eventos de diagnóstico de Infraestructura de diagnóstico de Windows).
`EventData`	Lista de objetos	(Opcional) Datos adicionales específicos de un proveedor sobre el evento de registro. Solo se incluye si el valor del par clave-valor `IncludeEventData` es `"true"`.

A continuación, se muestra un ejemplo de un evento convertido a formato JSON:

```
{[
    "EventId": 7036,
    "Description": "The Amazon SSM Agent service entered the stopped state.",
    "LevelDisplayName": "Informational",
    "LogName": "System",
    "MachineName": "mymachine.mycompany.com",
    "ProviderName": "Service Control Manager",
    "TimeCreated": "2017-10-04T16:42:53.8921205Z",
    "Index": 462335,
    "UserName": null,
    "Keywords": "Classic",
    "EventData": [
```

```
    "Amazon SSM Agent",
    "stopped",
    "rPctBAMZFhYubF8zVLcrBd3bTTcNzHvY5Jc2Br0aMrxxx=="
]}
```

Configuración de WindowSeventLogPollingSource

`WindowsEventLogPollingSource`utiliza un mecanismo basado en sondeo para recopilar todos los eventos nuevos del registro de eventos que coincidan con los parámetros configurados. El intervalo de sondeo se actualiza dinámicamente entre 100 ms y 5000 ms dependiendo de cuántos eventos se reunieron durante la última encuesta. A continuación, se muestra una declaración `WindowsEventLogPollingSource` de ejemplo:

```
{
    "Id": "MySecurityLog",
    "SourceType": "WindowsEventLogPollingSource",
    "LogName": "Security",
    "IncludeEventData": "true",
    "Query": "",
    "CustomFilters": "ExcludeOwnSecurityEvents"
}
```

Todas las declaraciones `WindowsEventLogPollingSource` pueden proporcionar los siguientes pares clave-valor:

`SourceType`

Tiene que ser la cadena literal "`WindowsEventLogPollingSource`" (obligatorio).

`LogName`

Especifica el registro. Las opciones válidas son`Application`,`Security`,`System`, u otros registros válidos.

`IncludeEventData`

Opcional. Cuando`true`, especifica que EventData adicional cuando se transmite como JSON y XML se incluye. El valor predeterminado es `false`.

`Query`

Opcional. Los registros de eventos de Windows admiten la consulta de eventos mediante expresiones XPath, que puede especificar mediante`Query`. Para obtener más información, consulte[Consultas de eventos y XML de eventos]en la documentación de Microsoft.

`CustomFilters`

Opcional. Una lista de filtros separados por un punto y coma (`;`). Se pueden especificar los siguientes filtros.

`ExcludeOwnSecurityEvents`

Excluye los eventos de seguridad generados por Kinesis Agent para Windows.

Configuración de WindowsETWEventSource

El tipo `WindowsETWEventSource` se utiliza para recopilar rastros de eventos de servicios y aplicaciones con una característica llamada Seguimiento de eventos para Windows (ETW). Para obtener más información, consulte Event Tracing en la documentación de Windows.

A continuación, se muestra una declaración `WindowsETWEventSource` de ejemplo:

```
{
    "Id": "ClrETWEventSource",
    "SourceType": "WindowsETWEventSource",
    "ProviderName": "Microsoft-Windows-DotNETRuntime",
    "TraceLevel": "Verbose",
    "MatchAnyKeyword": 32768
}
```

Todas las declaraciones `WindowsETWEventSource` pueden proporcionar los siguientes pares clave-valor:

`SourceType`

Tiene que ser la cadena literal "`WindowsETWEventSource`" (obligatorio).

`ProviderName`

Especifica qué proveedor de eventos se va a utilizar para recopilar los eventos de rastreo. Debe ser un nombre de ETW válido asignado a un proveedor instalado. Para determinar qué

proveedores están instalados, ejecute lo siguiente en una ventana del símbolo del sistema de Windows:

```
logman query providers
```

`TraceLevel`

Especifica qué categorías de los eventos de rastreo deben recopilarse. Los valores permitidos son `Critical`, `Error`, `Warning`, `Informational` y `Verbose`. El significado exacto depende del proveedor de ETW seleccionado.

`MatchAnyKeyword`

Este valor es un número de 64 bits, donde cada bit representa una palabra clave individual. Cada palabra clave describe una categoría de eventos que se va a recopilar. Para obtener información sobre las palabras clave admitidas, sus valores y cómo se relacionan con `TraceLevel`, consulte la documentación de ese proveedor. Por ejemplo, para obtener información sobre el proveedor ETW de CLR Palabras clave y niveles ETW de CLR en la documentación de Microsoft .NET Framework.

En el ejemplo anterior, 32768 (0x00008000) representa la `ExceptionKeyword` del proveedor ETW de CLR que indica al proveedor que recopile información sobre las excepciones generadas. Aunque JSON no admite de forma nativa constantes hexadecimales, puede especificarlas para `MatchAnyKeyword` incluyéndolas en una cadena. También puede especificar varias constantes separadas por comas. Por ejemplo, utilice lo siguiente para especificar `ExceptionKeyword` y `SecurityKeyword` (0x00000400):

```
{
   "Id": "MyClrETWEventSource",
   "SourceType": "WindowsETWEventSource",
   "ProviderName": "Microsoft-Windows-DotNETRuntime",
   "TraceLevel": "Verbose",
   "MatchAnyKeyword": "0x00008000, 0x00000400"
}
```

Para asegurarse de que todas las palabras clave especificadas pueden utilizarse con un proveedor, se combinan varios valores de las palabras clave con OR y se pasan a ese proveedor.

La salida de `WindowsETWEventSource` contiene la siguiente información sobre cada evento:

Atributo	Tipo	Descripción
`EventName`	Cadena	Tipo de evento que se produjo.
`ProviderName`	Cadena	Proveedor que detectó el evento.
`FormattedMessage`	Cadena	Resumen del evento en formato de texto.
`ProcessID`	Int	Proceso que registró el evento.
`ExecutingThreadID`	Int	Subproceso que registró el evento.
`MachineName`	Cadena	Nombre del equipo de escritorio o del servidor que registra el evento.
`Payload`	Tabla hash	Tabla con una clave de cadena y cualquier tipo de objeto como valor. La clave es el nombre del elemento de carga, mientras que el valor es el valor del elemento de carga. La carga depende del proveedor.

A continuación, se muestra un ejemplo de un evento convertido a formato JSON:

```
{
     "EventName": "Exception/Start",
     "ProviderName": "Microsoft-Windows-DotNETRuntime",
     "FormattedMessage": "ExceptionType=System.Exception;\r
\nExceptionMessage=Intentionally unhandled exception.;\r\nExceptionEIP=0x2ab0499;\r
\nExceptionHRESULT=-2,146,233,088;\r\nExceptionFlags=CLSCompliant;\r\nClrInstanceID=9
 ",
     "ProcessID": 3328,
     "ExecutingThreadID": 6172,
     "MachineName": "MyHost.MyCompany.com",
     "Payload":
```

```
      {
        "ExceptionType": "System.Exception",
        "ExceptionMessage": "Intentionally unhandled exception.",
        "ExceptionEIP": 44762265,
        "ExceptionHRESULT": -2146233088,
        "ExceptionFlags": 16,
        "ClrInstanceID": 9
      }
}
```

Configuración de WindowsPerformanceCounterSource

El tipo `WindowsPerformanceCounterSource` recopila métricas de contadores de rendimiento de Windows. A continuación, se muestra una declaración `WindowsPerformanceCounterSource` de ejemplo:

```
{
 "Id": "MyPerformanceCounter",
 "SourceType": "WindowsPerformanceCounterSource",
 "Categories": [{
   "Category": "Server",
   "Counters": ["Files Open", "Logon Total", "Logon/sec", "Pool Nonpaged Bytes"]
  },
  {
   "Category": "System",
   "Counters": ["Processes", "Processor Queue Length", "System Up Time"]
  },
  {
   "Category": "LogicalDisk",
   "Instances": "*",
   "Counters": [
    "% Free Space", "Avg. Disk Queue Length",
    {
     "Counter": "Disk Reads/sec",
     "Unit": "Count/Second"
    },
    "Disk Writes/sec"
   ]
  },
  {
   "Category": "Network Adapter",
   "Instances": "^Local Area Connection\* \d$",
   "Counters": ["Bytes Received/sec", "Bytes Sent/sec"]
```

```
  }
 ]
}
```

Todas las declaraciones `WindowsPerformanceCounterSource` pueden proporcionar los siguientes pares clave-valor:

`SourceType`

Tiene que ser la cadena literal `"WindowsPerformanceCounterSource"` (obligatorio).

`Categories`

Especifica un conjunto de grupos de métricas de contadores de rendimiento que se van a recopilar en Windows. Cada grupo de métricas contiene los siguientes pares clave-valor:

`Category`

Especifica el conjunto de métricas de contadores que se van a recopilar (obligatorio).

`Instances`

Especifica el conjunto de objetos de interés cuando hay un único conjunto de contadores de rendimiento para cada objeto. Por ejemplo, cuando la categoría es `LogicalDisk`, hay un conjunto de contadores de rendimiento para cada unidad de disco. Este par clave-valor es opcional. Puede utilizar los caracteres comodín `*` y `?` para encontrar coincidencias con varias instancias. Para sumar valores de todas las instancias, especifique `_Total`.

También puede utilizar`InstanceRegex`, que acepta expresiones regulares que contienen el`*`carácter comodín como parte del nombre de la instancia.

`Counters`

Especifica qué métricas de la categoría especificada se van a recopilar. Este par clave-valor es obligatorio. Puede utilizar los caracteres comodín `*` y `?` para encontrar varios contadores. Puede especificar `Counters` utilizando el nombre y la unidad o solamente el nombre. Si no se especifican las unidades del contador, el agente de Kinesis para Windows intentará deducirlas a partir del nombre. Si esta deducción no es correcta, es posible especificar la unidad de forma explícita. Si lo desea, puede cambiar los nombres de `Counter`. Existe una representación más compleja de un contador, que sería un objeto con los siguientes pares clave-valor:

`Counter`

Nombre del contador. Este par clave-valor es obligatorio.

`Rename`

Nombre del contador que se va a presentar al receptor. Este par clave-valor es opcional.

`Unit`

Significado del valor asociado al contador. Para obtener una lista completa de los nombres de unidades válidos, consulte la documentación sobre unidades enMetricDatumen laReferencia del API de Amazon CloudWatch.

A continuación, se muestra un ejemplo de una especificación de contadores compleja.

```
{
   "Counter": "Disk Reads/sec,
   "Rename": "Disk Reads per second",
   "Unit": "Count/Second"
}
```

`WindowsPerformanceCounterSource`Solo se puede utilizar con una canalización que especifique un receptor de Amazon CloudWatch. Utilice un receptor diferente si las métricas integradas en Kinesis Agent para Windows también se transmiten a CloudWatch. Examine el registro de Kinesis Agent para Windows después de iniciar un servicio para determinar qué unidades se dedujo que usaban los contadores si no se especificó ninguna unidad en el cuadro de`WindowsPerformanceCounterSource`Declaraciones. Utilice PowerShell para determinar los nombres válidos de las categorías, las instancias y los contadores.

Para obtener información sobre todas las categorías, incluidos los contadores asociados a conjuntos de contadores, ejecute este comando en una ventana de PowerShell:

```
Get-Counter -ListSet * | Sort-Object
```

Para determinar qué instancias están disponibles en cada uno de los contadores del conjunto de contadores, ejecute un comando similar al del siguiente ejemplo en una ventana de PowerShell:

```
Get-Counter -Counter "\Process(*)\% Processor Time"
```

El valor del parámetro `Counter` debe ser una de las rutas de un miembro `PathsWithInstances` mostrado en la anterior invocación del comando `Get-Counter -ListSet`.

Origen de métricas integrado en Windows en

Además de las fuentes de métricas ordinarias como el`WindowsPerformanceCounterSource`tipo (consulteConfiguración de WindowsPerformanceCounterSource), el tipo de receptor de CloudWatch puede recibir métricas de un origen especial que recopile métricas sobre el propio agente de Kinesis para Windows. Las métricas de Kinesis Agent para Windows también están disponibles en la`KinesisTap`de los contadores de rendimiento de Windows.

La`MetricsFilter`El par clave-valor de las declaraciones de receptores de CloudWatch especifica qué métricas se transmiten a CloudWatch desde el origen de métricas de integrado en Kinesis Agent para Windows. El valor es una cadena que contiene una o varias expresiones de filtro separadas por punto y coma; por ejemplo:

`"MetricsFilter": "ExpresiónDeFiltro1;ExpresiónDeFiltro2"`

Una métrica que coincide con una o varias expresiones de filtro se transmite a CloudWatch.

Las métricas de instancias únicas son de carácter global y no están vinculadas a un determinado origen o receptor. Las métricas de instancias múltiples son dimensionales y dependen de la declaración `Id` del origen o receptor. Cada tipo de origen o receptor puede tener un conjunto diferente de métricas.

Para obtener una lista de los nombres de métrica del agente de Kinesis para Windows, consulteLista de métricas del agente Kinesis para Windows.

En las métricas de instancias únicas, la expresión de filtro es el nombre de la métricas; por ejemplo:

```
"MetricsFilter": "SourcesFailedToStart;SinksFailedToStart"
```

En el caso de las métricas de instancias múltiples, la expresión de filtro es el nombre de la métrica, un punto (`.`) y el `Id` de la declaración del origen o receptor que generó esa métrica. Por ejemplo, supongamos que hay una declaración de un receptor cuyo `Id` es `MyFirehose`:

```
"MetricsFilter": "KinesisFirehoseRecordsFailedNonrecoverable.MyFirehose"
```

Puede utilizar patrones de caracteres comodín especiales diseñados para distinguir entre las métricas de instancias únicas y las métricas de instancias múltiples.

- El asterisco (`*`) busca coincidencias con cero o más caracteres, salvo el punto (`.`).
- El signo de interrogación de cierre (`?`) busca coincidencias con un único carácter, salvo el punto.
- Cualquier otro carácter solamente coincide consigo mismo.
- `_Total` es un token especial que genera la suma de todos los valores de las instancias múltiples coincidentes que hay en la dimensión.

En el siguiente ejemplo, se buscan coincidencias con todas las métricas de instancias únicas:

```
"MetricsFilter": "*"
```

Como los asteriscos no buscan coincidencias con el carácter de punto, solo se incluyen las métricas de instancias únicas.

En el siguiente ejemplo, se buscan coincidencias con todas las métricas de instancias múltiples:

```
"MetricsFilter": "*.*"
```

En el siguiente ejemplo, se buscan coincidencias con todas las métricas (únicas y múltiples):

```
"MetricsFilter": "*;*.*"
```

En el siguiente ejemplo, se suman todas las métricas de instancias múltiples de todos los orígenes y receptores:

```
"MetricsFilter": "*._Total"
```

En el siguiente ejemplo, se suman todas las métricas de Kinesis Data Firehose de Kinesis para todos los receptores de Firehose de datos de Kinesis:

```
"MetricsFilter": "*Firehose*._Total"
```

En el siguiente ejemplo, se buscan coincidencias con todas las métricas con error de las instancias únicas y múltiples:

```
"MetricsFilter": "*Failed*;*Error*.*;*Failed*.*"
```

En el siguiente ejemplo, se buscan coincidencias con todas las métricas con error no recuperables que se han sumado en todos los orígenes y receptores:

```
"MetricsFilter": "*Nonrecoverable*._Total"
```

Para obtener información acerca de cómo especificar una canalización que utilice el origen de métricas integrado en Kinesis Agent para Windows, consulteConfiguración del agente Kinesis para tuberías métricas de Windows.

Lista de métricas del agente Kinesis para Windows

A continuación, se muestra una lista con las métricas de instancias únicas y múltiples que están disponibles para Kinesis Agent para Windows.

Métricas de instancias únicas

Las siguientes métricas de instancias únicas están disponibles:

`KinesisTapBuildNumber`

El número de versión de Kinesis Agent para Windows.

`PipesConnected`

Número de canalizaciones que han conectado correctamente el origen con el receptor.

`PipesFailedToConnect`

Número de canalizaciones que no han conectado correctamente el origen con el receptor.

`SinkFactoriesFailedToLoad`

Número de tipos de receptores que no se cargaron correctamente en el agente de Kinesis para Windows.

`SinkFactoriesLoaded`

Número de tipos de receptores que se cargaron correctamente en el agente de Kinesis para Windows.

`SinksFailedToStart`

Número de receptores que no se iniciaron correctamente; por lo general, debido a una declaración incorrecta del receptor.

`SinksStarted`

Número de receptores que se iniciaron correctamente.

`SourcesFailedToStart`

Número de orígenes que no se iniciaron correctamente; por lo general, debido a una declaración incorrecta del origen.

`SourcesStarted`

Número de orígenes que se iniciaron correctamente.

`SourceFactoriesFailedToLoad`

Número de tipos de orígenes que no se cargaron correctamente en el agente de Kinesis para Windows.

`SourceFactoriesLoaded`

Número de tipos de orígenes que se cargaron correctamente en el agente de Kinesis para Windows.

Métricas de instancias múltiples

Las siguientes métricas de instancias múltiples están disponibles:

Métricas de DirectorySource

`DirectorySourceBytesRead`

Número de bytes leídos durante el intervalo de este `DirectorySource`.

`DirectorySourceBytesToRead`

Número conocido de bytes disponibles para leer que aún no ha leído Kinesis Agent para Windows.

`DirectorySourceFilesToProcess`

Número de archivos conocidos que debe examinarse y que aún no ha examinado el agente de Kinesis para Windows.

`DirectorySourceRecordsRead`

Número de registros que se han leído durante el intervalo de este `DirectorySource`.

Métricas de WindowsEventLogSource

`EventLogSourceEventsError`

Número de eventos de un registro de eventos de Windows que no se han leído correctamente.

`EventLogSourceEventsRead`

Número de eventos de un registro de eventos de Windows que se han leído correctamente.

Métricas de receptores de KinesisFirehose

`KinesisFirehoseBytesAccepted`

Número de bytes aceptados durante el intervalo.

`KinesisFirehoseClientLatency`

Tiempo transcurrido entre la generación del registro y su transmisión al servicio de Kinesis Data Firehose.

`KinesisFirehoseLatency`

Tiempo transcurrido entre el inicio y el final de la transmisión del registro en el servicio de Kinesis Data Firehose.

`KinesisFirehoseNonrecoverableServiceErrors`

Número de veces que los registros no pudieron enviarse sin error al servicio de Kinesis Data Firehose a pesar de que volviera a intentarse.

`KinesisFirehoseRecordsAttempted`

Número de registros que han intentado transmitirse al servicio de Kinesis Data Firehose.

`KinesisFirehoseRecordsFailedNonrecoverable`

Número de registros que no pudieron transmitirse correctamente al servicio de Kinesis Data Firehose a pesar de que volviera a intentarse.

`KinesisFirehoseRecordsFailedRecoverable`

Número de registros que se transmitieron correctamente al servicio de Kinesis Data Firehose, pero solo cuando se necesitaron varios intentos.

`KinesisFirehoseRecordsSuccess`

Número de registros que se transmitieron correctamente al servicio de Kinesis Data Firehose sin que fueran necesarios varios intentos.

`KinesisFirehoseRecoverableServiceErrors`

Número de veces que los registros pudieron enviarse correctamente al servicio de Kinesis Data Firehose, pero solo cuando se necesitaron varios intentos.

Métricas de KinesisStream

`KinesisStreamBytesAccepted`

Número de bytes aceptados durante el intervalo.

`KinesisStreamClientLatency`

Tiempo transcurrido entre la generación del registro y su transmisión al servicio de Kinesis Data Streams.

`KinesisStreamLatency`

Tiempo transcurrido entre el inicio y el final de la transmisión del registro en el servicio de Kinesis Data Streams.

`KinesisStreamNonrecoverableServiceErrors`

Número de veces que los registros no pudieron enviarse sin error al servicio de Kinesis Data Streams a pesar de que volviera a intentarse.

`KinesisStreamRecordsAttempted`

Número de registros que han intentado transmitirse al servicio de Kinesis Data Streams.

`KinesisStreamRecordsFailedNonrecoverable`

Número de registros que no pudieron transmitirse correctamente al servicio de secuencias de Kinesis Data Streams a pesar de que volviera a intentarse.

`KinesisStreamRecordsFailedRecoverable`

Número de registros que se transmitieron correctamente al servicio de Kinesis Data Streams, pero solo cuando se necesitaron varios intentos.

`KinesisStreamRecordsSuccess`

Número de registros que se transmitieron correctamente al servicio de secuencias de Kinesis Data Streams sin que fueran necesarios varios intentos.

`KinesisStreamRecoverableServiceErrors`

Número de veces que los registros pudieron enviarse correctamente al servicio de Kinesis Data Streams, pero solo cuando se necesitaron varios intentos.

Métricas de CloudWatchLog

`CloudWatchLogBytesAccepted`

Número de bytes aceptados durante el intervalo.

`CloudWatchLogClientLatency`

Tiempo transcurrido entre la generación del registro y su transmisión al servicio de registros de CloudWatch Logs.

`CloudWatchLogLatency`

Tiempo transcurrido entre el inicio y el final de la transmisión del registro en el servicio de CloudWatch Logs.

`CloudWatchLogNonrecoverableServiceErrors`

Número de veces que los registros no pudieron enviarse sin error al servicio de registros de CloudWatch a pesar de que volviera a intentarse.

`CloudWatchLogRecordsAttempted`

Número de registros que han intentado transmitirse al servicio de CloudWatch Logs.

`CloudWatchLogRecordsFailedNonrecoverable`

Número de registros que no pudieron transmitirse correctamente al servicio de registros de CloudWatch a pesar de que volviera a intentarse.

`CloudWatchLogRecordsFailedRecoverable`

Número de registros que se transmitieron correctamente al servicio de CloudWatch Logs, pero solo cuando se necesitaron varios intentos.

`CloudWatchLogRecordsSuccess`

Número de registros que se transmitieron correctamente al servicio de registros de CloudWatch sin que fueran necesarios varios intentos.

`CloudWatchLogRecoverableServiceErrors`

Número de veces que los registros pudieron enviarse correctamente al servicio de CloudWatch Logs, pero solo cuando se necesitaron varios intentos.

Métricas de CloudWatch

`CloudWatchLatency`

Tiempo transcurrido de media entre el inicio y el final de la transmisión de métricas en el servicio de CloudWatch.

`CloudWatchNonrecoverableServiceErrors`

Número de veces que las métricas no pudieron enviarse sin error al servicio de CloudWatch a pesar de que volviera a intentarse.

`CloudWatchRecoverableServiceErrors`

Número de veces que las métricas pudieron enviarse sin error al servicio de CloudWatch, pero solo cuando se necesitaron varios intentos.

`CloudWatchServiceSuccess`

Número de veces que las métricas pudieron enviarse sin error al servicio de CloudWatch sin que fueran necesarios varios intentos.

Configuración de Bookmark

De forma predeterminada, el agente de Kinesis para Windows envía registros a los receptores que se crearon después de iniciar el agente. A veces, resulta útil enviar registros anteriores; por ejemplo, registros que se crearon durante el periodo de tiempo que estuvo detenido por Kinesis Agent para Windows durante una actualización automática. La característica de marcador (bookmark) realiza un seguimiento de lo que los registros han enviado a los receptores. Cuando Kinesis Agent para Windows está en modo marcador y se inicia, envía todas las entradas de registro que se crearon tras la detención del agente de Kinesis para Windows, junto con las entradas que se crearon posteriormente. Para controlar este comportamiento, las declaraciones de origen basadas en archivos pueden incluir opcionalmente los siguientes pares clave-valor:

`InitialPosition`

Especifica la situación inicial del marcador (bookmark). Los valores posibles son los siguientes:

`EOS`

Especifica el final de la secuencia (EOS). Solo se envían a los receptores las entradas de registro que se crearon mientras el agente estaba en ejecución.

`0`

Todos los eventos y entradas de registro se envían desde el primer momento. A continuación, se crea un marcador para garantizar que finalmente se envían todas las nuevas entradas de registro y los eventos que se crearon tras la generación del marcador (bookmark), tanto si estaba en ejecución como si no.

`Bookmark`

El marcador (bookmark) se ha inicializado justo después del último registro o evento. A continuación, se crea un marcador para garantizar que finalmente se envían todas las nuevas entradas de registro y los eventos que se crearon tras la generación del marcador (bookmark), tanto si estaba en ejecución como si no.

De forma predeterminada, los marcadores están habilitadas. Los archivos se almacenan en la`%ProgramData%\Amazon\KinesisTap`.

`Timestamp`

Se envían los registros y eventos creados después del valor `InitialPositionTimestamp` (cuya descripción se incluye a continuación). A continuación, se crea un marcador para

garantizar que finalmente se envían todas las nuevas entradas de registro y los eventos que se crearon tras la generación del marcador (bookmark) Kinesis tanto si estaba en ejecución como si no.

`InitialPositionTimestamp`

Especifica la marca temporal de la entrada de registro o el evento más antiguo que desea tener. Especifique este par clave-valor solamente cuando el valor de `InitialPosition` sea `Timestamp`.

`BookmarkOnBufferFlush`

Esta configuración se puede añadir a cualquier fuente que se pueda marcar. Cuando se establece en`true`, garantiza que las actualizaciones de marcadores solo se produzcan cuando un receptor envía correctamente un evento a AWS. Solo puede suscribir un único receptor a una fuente. Si envía registros a varios destinos, duplique sus fuentes para evitar posibles problemas con la pérdida de datos.

Cuando Kinesis Agent para Windows ha estado detenido durante un largo periodo de tiempo, puede ser necesario eliminar estos marcadores, ya que podría ocurrir que las entradas de registro y los eventos señalados con estos marcadores no existieran ya. Los archivos de marcadores (bookmark) de un determinado id de origen se encuentran en `%PROGRAMDATA%\Amazon\AWSKinesisTap \`*`source id`*`.bm`.

Los marcadores (bookmarks) no funcionan en archivos truncados o cuyo nombre ha cambiado. Debido a su naturaleza, no se pueden utilizar marcadores (bookmarks) con los contadores de rendimiento y los contadores de los eventos de ETW.

Declaraciones de receptores

Las declaraciones de receptores especifican dónde y en qué formato deben enviarse los registros, los eventos y las métricas a los diferentes servicios de AWS. En las secciones siguientes, se describen las configuraciones de los tipos de receptores integrados que están disponibles en Amazon Kinesis Agent para Microsoft Windows. Como Kinesis Agent para Windows es ampliable, puede agregar tipos de receptores personalizados. Normalmente, todos los tipos de receptores necesitan pares clave-valor únicos en las declaraciones de configuración que sean pertinentes para dicho tipo de receptor.

Todas las declaraciones de receptores pueden contener los siguientes pares clave-valor:

`Id`

Cadena única que identifica un receptor específico en el archivo de configuración (obligatorio).

`SinkType`

Nombre del tipo de este receptor (obligatorio). El tipo de receptor especifica el destino de los datos de registro, eventos o métricas que este receptor va a transmitir.

`AccessKey`

Especifica la clave de acceso de AWS que va a utilizar para autorizar el acceso al servicio de AWS que está asociado al tipo de receptor. Este par clave-valor es opcional. Para obtener más información, consulte Configuración de seguridad de los receptores.

`SecretKey`

Especifica la clave secreta de AWS que va a utilizar para autorizar el acceso al servicio de AWS que está asociado al tipo de receptor. Este par clave-valor es opcional. Para obtener más información, consulte Configuración de seguridad de los receptores.

`Region`

Especifica la región de AWS que contiene los recursos de destino de la transmisión. Este par clave-valor es opcional.

`ProfileName`

Especifica el perfil de AWS que se va a utilizar para la autenticación. Este par clave-valor es opcional, pero, si se especifica, anula cualquier clave de acceso y clave secreta que se hayan especificado. Para obtener más información, consulte Configuración de seguridad de los receptores.

`RoleARN`

Especifica el rol de IAM que se va a utilizar para obtener acceso al servicio de AWS que está asociado al tipo de receptor. Esta opción es útil cuando el agente de Kinesis para Windows se ejecuta en una instancia EC2 pero sería más conveniente utilizar un rol diferente al que se hace referencia en el perfil de instancia. Por ejemplo, se puede utilizar un rol de acceso entre cuentas para destinar recursos que no estén en la misma cuenta de AWS que la instancia EC2. Este par clave-valor es opcional.

`Format`

Especifica el tipo de serialización que se aplica a los registros y los datos de eventos antes de transmitirlos. Los valores válidos son `json` y `xml`. Esta opción es útil cuando los análisis

posteriores de la canalización de datos requieren o prefieren que los datos estén en un determinado formato. Este par clave-valor es opcional y, si no se especifica, el receptor transmitirá los datos del origen en formato de texto normal al servicio de AWS que esté asociado con este tipo de receptor.

`TextDecoration`

Si no se especifica `Format`, `TextDecoration` indica qué otro texto debe incluirse al transmitir entradas de registro o eventos. Para obtener más información, consulte Configuración de decoraciones de receptores. Este par clave-valor es opcional.

`ObjectDecoration`

Cuando se especifica `Format`, `ObjectDecoration` indica qué otros datos se van a incluir en la entrada del registro o del evento antes de su serialización y transmisión. Para obtener más información, consulte Configuración de decoraciones de receptores. Este par clave-valor es opcional.

`BufferInterval`

Para minimizar el número de llamadas de API al servicio de AWS asociado con el tipo de receptor, Kinesis Agent for Windows almacena en el búfer varias entradas de registro, eventos o métricas antes de realizar la transmisión. Esto permite ahorrar dinero en los servicios que cobran por cada llamada a la API. `BufferInterval` especifica el tiempo máximo (en segundos) que los registros deben almacenarse en el búfer antes de transmitirse al servicio de AWS. Este par clave-valor es opcional y, si se especifica, debe utilizarse una cadena para representar el valor.

`BufferSize`

Para minimizar el número de llamadas de API al servicio de AWS asociado con el tipo de receptor, Kinesis Agent for Windows almacena en el búfer varias entradas de registro, eventos o métricas antes de realizar la transmisión. Esto permite ahorrar dinero en los servicios que cobran por cada llamada a la API. `BufferSize` especifica el número máximo de registros que se van a almacenar en el búfer antes de transmitirse al servicio de AWS. Este par clave-valor es opcional y, si se especifica, debe utilizarse una cadena para representar el valor.

`MaxAttempts`

Especifica el número máximo de veces que Kinesis Agent for Windows intenta transmitir un conjunto de entradas de registro, eventos y métricas a un servicio de AWS si se producen errores continuados durante la transmisión. Este par clave-valor es opcional. Si se especifica, debe utilizarse una cadena para representar el valor. El valor predeterminado es "3".

Para ver algunos ejemplos de archivos de configuración completos en los que se utilizan varios tipos de receptores, consulte Transmisión a los receptores desde el registro de eventos de la aplicación de Windows.

Temas

Configuración de receptores **`KinesisStream`**

La`KinesisStream`El tipo de receptor transmite entradas de registro y eventos al servicio Kinesis Data Streams. Normalmente, los datos que se transmiten a Kinesis Data Streams se procesan en una o varias aplicaciones personalizadas que se ejecutan utilizando diversos servicios de AWS. Los datos se transmiten en una secuencia con nombre que se configura con Kinesis Data Streams. Para obtener más información, consulte laGuía del desarrollador de Amazon Kinesis Data Streams.

A continuación, se muestra un ejemplo de una declaración de receptores de Kinesis Data Streams:

```
{
    "Id": "TestKinesisStreamSink",
    "SinkType": "KinesisStream",
```

```
    "StreamName": "MyTestStream",
    "Region": "us-west-2"
}
```

Todas las declaraciones de receptores de `KinesisStream` pueden proporcionar estos otros pares clave-valor:

`SinkType`

Tiene que especificarse y el valor debe ser la cadena literal `KinesisStream`.

`StreamName`

Especifica el nombre de la secuencia de datos de Kinesis que recibe los datos transmitidos desde el`KinesisStream`tipo de fregadero (requerido). Antes de transmitir los datos, configure la secuencia en la consola de administración de AWS, la CLI de AWS o a través de una aplicación con la API de Kinesis Data Streams.

`RecordsPerSecond`

Especifica el número máximo de registros transmitidos a Kinesis Data Streams por segundo. Este par clave-valor es opcional. Si se especifica, debe utilizarse un número entero para representar el valor. El valor predeterminado es 1000 registros.

`BytesPerSecond`

Especifica el número máximo de bytes transmitidos a Kinesis Data Streams por segundo. Este par clave-valor es opcional. Si se especifica, debe utilizarse un número entero para representar el valor. El valor predeterminado es 1 MB.

El valor predeterminado de `BufferInterval` en este tipo de receptor es de 1 segundo, mientras que el valor predeterminado de `BufferSize` es de 500 registros.

Configuración de receptores **`KinesisFirehose`**

La`KinesisFirehose`El tipo de receptor transmite entradas de registro y eventos al servicio Kinesis Data Firehose. Kinesis Data Firehose proporciona los datos transmitidos a otros servicios para su almacenamiento. Por lo general, los datos almacenados suelen analizarse después, en fases posteriores de la canalización de datos. Los datos se transmiten a una secuencia de entrega con nombre que se configura con Kinesis Data Firehose. Para obtener más información, consulte laGuía del desarrollador de Amazon Kinesis Data Firehose.

A continuación, se muestra un ejemplo de una declaración de receptores de Kinesis Data Firehose:

```
{
   "Id": "TestKinesisFirehoseSink",
   "SinkType": "KinesisFirehose",
   "StreamName": "MyTestFirehoseDeliveryStream",
   "Region": "us-east-1",
   "CombineRecords": "true"
}
```

Todas las declaraciones de receptores de `KinesisFirehose` pueden proporcionar estos otros pares clave-valor:

`SinkType`

Tiene que especificarse y el valor debe ser la cadena literal `KinesisFirehose`.

`StreamName`

Especifica el nombre de la secuencia de entrega de Kinesis Data Firehose que recibe los datos transmitidos desde el`KinesisStream`tipo de fregadero (requerido). Antes de transmitir los datos, configure la secuencia de entrega utilizando la consola de administración de AWS, la CLI de AWS o a través de una aplicación con la API de Kinesis Data Firehose.

`CombineRecords`

Cuando se establece en`true`, especifica combinar varios registros pequeños en un registro grande con un tamaño máximo de 5 KB. Este par clave-valor es opcional. Los registros combinados con esta función están separados por`\n`. Si utiliza AWS Lambda para transformar un registro de Kinesis Data Firehose, la función Lambda debe tener en cuenta el carácter separador.

`RecordsPerSecond`

Especifica el número máximo de registros que se transmiten a Kinesis Data Streams por segundo. Este par clave-valor es opcional. Si se especifica, debe utilizarse un número entero para representar el valor. El valor predeterminado es 5000 registros.

`BytesPerSecond`

Especifica el número máximo de bytes que se transmiten a Kinesis Data Streams por segundo. Este par clave-valor es opcional. Si se especifica, debe utilizarse un número entero para representar el valor. El valor predeterminado es 5 MB.

El valor predeterminado de `BufferInterval` en este tipo de receptor es de 1 segundo, mientras que el valor predeterminado de `BufferSize` es de 500 registros.

Configuración de receptores de CloudWatch

La`CloudWatch`El tipo de receptor transmite métricas al servicio CloudWatch. Puede consultar las métricas en AWS Management Console. Para obtener más información, consulte la Guía del usuario de Amazon CloudWatch.

A continuación, se muestra una declaración de receptores de `CloudWatch` de ejemplo:

```
{
   "Id": "CloudWatchSink",
   "SinkType": "CloudWatch"
}
```

Todas las declaraciones de receptores de `CloudWatch` pueden proporcionar estos otros pares clave-valor:

`SinkType`

Tiene que especificarse y el valor debe ser la cadena literal `CloudWatch`.

`Interval`

Especifica con qué frecuencia (en segundos) Kinesis Agent para Windows informa sobre las métricas al servicio CloudWatch. Este par clave-valor es opcional. Si se especifica, debe utilizarse un número entero para representar el valor. El valor de predeterminado es de 60 segundos. Especifique 1 segundo si desea métricas de CloudWatch de alta resolución.

`Namespace`

Especifica el espacio de nombres de CloudWatch en el que se notifican los datos de las métricas. Los espacios de nombres de CloudWatch agrupan conjuntos de métricas. Este par clave-valor es opcional. El valor predeterminado es `KinesisTap`.

`Dimensions`

Especifica las dimensiones de CloudWatch que se utilizan para aislar conjuntos de métricas dentro de un espacio de nombres. Esto puede resultar útil para proporcionar diferentes conjuntos de datos de métricas en cada equipo o servidor, por ejemplo.

Este par clave-valor es opcional y, si se especifica, el valor debe ajustarse al siguiente formato: "clave1=valor1;clave2=valor2...". El valor predeterminado es "ComputerName={computername};InstanceId={instance_id}". Este valor permite la sustitución de variables del receptor. Para obtener más información, consulte Configuración de sustituciones de variables de receptor.

`MetricsFilter`

Especifica qué métricas se transmiten a CloudWatch desde el origen de métricas integrado de Kinesis Agent para Windows. Para obtener más información acerca del origen de métricas de Kinesis Agent para Windows, como información sobre la sintaxis del valor de este par, consulteOrigen de métricas integrado en Windows en.

Configuración de receptores **`CloudWatchLogs`**

La`CloudWatchLogs`El tipo de receptor transmite entradas de registro y eventos al servicio de registros de Amazon CloudWatch Logs. Puede consultar los registros en AWS Management Console o procesarlos en otras etapas de las canalizaciones de datos. Los datos se transmiten en una secuencia de registro con nombre que se configura en CloudWatch Logs. Las secuencias de registro se organizan en grupos de registros con nombre. Para obtener más información, consulte laGuía del usuario de Amazon CloudWatch Logs.

A continuación, se muestra un ejemplo de una declaración de receptores de CloudWatch Logs

```
{
   "Id": "MyCloudWatchLogsSink",
   "SinkType": "CloudWatchLogs",
   "BufferInterval": "60",
   "BufferSize": "100",
   "Region": "us-west-2",
   "LogGroup": "MyTestLogGroup",
   "LogStream": "MyTestStream"
}
```

Todas las declaraciones de receptores de `CloudWatchLogs` deben proporcionar estos otros pares clave-valor:

`SinkType`

Debe ser la cadena literal `CloudWatchLogs`.

`LogGroup`

Especifica el nombre del grupo de registros de CloudWatch Logs que contiene la secuencia de registro que recibe las entradas de registro y eventos que transmite la propiedad`CloudWatchLogs`tipo de fregadero. Si el grupo de registros especificado no existe, Kinesis Agent para Windows intenta crearlo.

`LogStream`

Especifica el nombre de la secuencia de registro de CloudWatch Logs que recibe la secuencia de registro y eventos de`CloudWatchLogs`tipo de fregadero. Este valor permite la sustitución de variables del receptor. Para obtener más información, consulte Configuración de sustituciones de variables de receptor. Si la secuencia de registro especificada no existe, Kinesis Agent para Windows intenta crearla.

El valor predeterminado de `BufferInterval` en este tipo de receptor es de 1 segundo, mientras que el valor predeterminado de `BufferSize` es de 500 registros. El tamaño máximo del búfer es de 10 000 registros.

Local**`FileSystem`**Configuración de receptores

El tipo de fregadero`FileSystem`guarda registros de eventos y registros en un archivo del sistema de archivos local en lugar de transmitirlos a los servicios de AWS.`FileSystem`son útiles para pruebas y diagnósticos. Por ejemplo, puede utilizar este tipo de receptor para examinar los registros antes de enviarlos a AWS.

con`FileSystem`, también puede utilizar parámetros de configuración para simular el procesamiento por lotes, la limitación y la retry-on-error para imitar el comportamiento de los sumideros de AWS reales.

Todos los registros de todas las fuentes conectadas a un`FileSystem`se guardan en el único archivo especificado como`FilePath`. Si`FilePath`No se especifica, los registros se guardan en un archivo denominado*`SinkId`*`.txt`en la`%TEMP%`, que por lo general es`C:\Users\`*`UserName`*`\AppData\Local\Temp`, donde*`SinkId`*es el identificador único del receptor y*`UserName`*es el nombre de usuario de Windows del usuario activo.

Este tipo de receptor admite atributos de decoración de texto. Para obtener más información, consulte Configuración de decoraciones de receptores.

Un ejemplo de`FileSystem`En el siguiente ejemplo se muestra la configuración de tipo de receptor.

```
{
    "Id": "LocalFileSink",
    "SinkType": "FileSystem",
    "FilePath": "C:\\ProgramData\\Amazon\\local_sink.txt",
    "Format": "json",
    "TextDecoration": "",
    "ObjectDecoration": ""
}
```

LaFileSystemLa configuración consta de los siguientes pares clave-valor.

`SinkType`

Debe ser la cadena literal `FileSystem`.

`FilePath`

Especifica la ruta de acceso y el archivo donde se guardan los registros. Este par clave-valor es opcional. Si no se especifica, el valor predeterminado es*TempPath*`\\`*SinkId*`.txt`, donde*TempPath*es la carpeta almacenada en la`%TEMP%`Variable y*SinkId*es el identificador único del receptor.

`Format`

Especifica el formato del evento que se va a`json`or`xml`. Este par de claves es opcional y distingue entre mayúsculas y minúsculas. Si se omite, los eventos se escriben en el archivo en texto sin formato.

`TextDecoration`

Sólo se aplica a eventos escritos en texto sin formato. Este par clave-valor es opcional.

`ObjectDecoration`

Solo se aplica a eventos donde`Format`toma el valor`json`. Este par clave-valor es opcional.

Uso avanzado: limitación de registros y simulación de fallas

`FileSystem`puede imitar el comportamiento de los sumideros de AWS simulando la limitación de registros. Puede utilizar los siguientes pares clave-valor para especificar atributos de limitación de registros y simulación de errores.

Al adquirir un bloqueo en el archivo de destino y evitar las escrituras en él, puede usar`FileSystem`para simular y examinar el comportamiento de los sumideros de AWS cuando falla la red.

En el siguiente ejemplo se muestra una`FileSystem`con atributos de simulación.

```
{
    "Id": "LocalFileSink",
    "SinkType": "FileSystem",
    "FilePath": "C:\\ProgramData\\Amazon\\local_sink.txt",
    "TextDecoration": "",
    "RequestsPerSecond": "100",
    "BufferSize": "10",
    "MaxBatchSize": "1024"
}
```

`RequestsPerSecond`

Opcional y especificado como un tipo de cadena. Si se omite, el valor predeterminado es`"5"`. Controla la velocidad de solicitudes que procesa el receptor, es decir, escribe en el archivo, no el número de registros. Kinesis Agent para Windows realiza solicitudes por lotes a los extremos de AWS, por lo que una solicitud puede contener varios registros.

`BufferSize`

Opcional y especificado como tipo de cadena. Especifica el número máximo de registros de eventos que el receptor realiza por lotes antes de guardarlos en el archivo.

`MaxBatchSize`

Opcional y especificado como un tipo de cadena. Especifica la cantidad máxima de datos de registro de eventos en bytes que el receptor de lotes antes de guardarlos en el archivo.

El límite máximo de velocidad de registro es una función de`BufferSize`, que determina el número máximo de registros por solicitud y`RequestsPerSecond`. Puede calcular el límite de velocidad de registro por segundo utilizando la siguiente fórmula.

RecordRate=`BufferSize`*`RequestsPerSecond`

Dados los valores de configuración en el ejemplo anterior, hay una velocidad máxima de registro de 1000 registros por segundo.

Configuración de seguridad de los receptores

Configuración de la autenticación

Para que Kinesis Agent for Windows pueda transmitir registros, eventos y métricas a los servicios de AWS, el acceso debe estar autenticado. Hay varias formas para proporcionar autenticación a Kinesis Agent para Windows. La forma elegida dependerá de la situación en la que se esté ejecutando Kinesis Agent para Windows y los requisitos de seguridad específicos de cada organización.

- Si Kinesis Agent para Windows se ejecuta en un host de Amazon EC2, la forma más sencilla y segura de proporcionar la autenticación es crear un rol de IAM con acceso suficiente a las operaciones necesarias en los servicios de AWS correspondientes y un perfil de instancia EC2 que haga referencia a dicho rol. Para obtener más información sobre la creación de perfiles de instancia, consulte Uso de perfiles de instancia. Para obtener información sobre las políticas que deben asociarse al rol de IAM, consulteConfiguración de la autorización .

  Después de crear el perfil de instancia, puede asociarlo con cualquier instancia EC2 que utilice el agente de Kinesis para Windows. Si las instancias ya tienen un perfil de instancia asociado, puede vincular las políticas apropiadas al rol que esté asociado a ese perfil de instancia.
- Si Kinesis Agent para Windows se ejecuta en un host de EC2 de una determinada cuenta pero los recursos de destino del receptor se encuentran en una cuenta diferente, puede crear un rol de IAM para obtener acceso entre cuentas. Para obtener más información, consulteTutorial: Delegación del acceso entre cuentas de AWS mediante roles de IAM. Después de crear el rol entre cuentas, especifique el nombre de recurso de Amazon (ARN) de dicho rol como el valor del parámetro`RoleARN`Par clave-valor en la declaración del receptor. A continuación, Kinesis Agent para Windows intentará adoptar el rol entre cuentas especificado cuando obtenga acceso a los recursos de AWS que están asociados con ese tipo de receptor.
- Si Kinesis Agent para Windows se ejecuta fuera de Amazon EC2 (por ejemplo, en el entorno local), existen varias opciones:
  - Si es aceptable registrar el servidor o el equipo de escritorio locales como una instancia administrada de Amazon EC2 Systems Manager, utilice el siguiente proceso para configurar la autenticación:
    1. Utilice el proceso que se describe en Configuración de AWS Systems Manager en entornos híbridos para crear un rol de servicio, crear una activación de una instancia administrada e instalar el agente de SSM.
    2. Asocie las políticas adecuadas al rol de servicio para permitir que Kinesis Agent para Windows tenga acceso a los recursos necesarios para transmitir datos desde los receptores

configurados. Para obtener información sobre las políticas que deben asociarse al rol de IAM, consulteConfiguración de la autorización .

3. Utilice el proceso descrito enConfiguraciónProfileRefreshingAWSCredentialProviderActualizar credenciales de AWSpara actualizar las credenciales de AWS.

Este es el método recomendado para las instancias que no son EC2, ya que SSM y AWS administran las credenciales de forma segura.

- Si es aceptable ejecutar el servicio AWSKinesisTap de para Kinesis Agent para Windows con un usuario específico en lugar de con la cuenta del sistema predeterminada, utilice el siguiente proceso:

  1. Cree un usuario de IAM en la cuenta de AWS en la que se utilizarán los servicios de AWS. Registre la clave de acceso y la clave secreta de este usuario durante el proceso de creación. Necesitará esta información más adelante en este mismo procedimiento.
  2. Asocie políticas al usuario de IAM que autoricen el acceso a las operaciones necesarias para los servicios correspondientes. Para obtener información sobre las políticas que deben asociarse al usuario de IAM, consulteConfiguración de la autorización .
  3. Modifique el servicio AWSKinesisTap en cada equipo o servidor para que se ejecute con un usuario específico en lugar de con la cuenta del sistema predeterminada.
  4. Cree un perfil en el almacén de SDK con la clave de acceso y la clave secreta registradas anteriormente. Para obtener más información, consulte Configuración de credenciales de AWS.
  5. Actualice el archivo `AWSKinesisTap.exe.config` del directorio `%PROGRAMFILES%\Amazon\AWSKinesisTap` para especificar el nombre del perfil que se creó en el paso anterior. Para obtener más información, consulte Configuración de credenciales de AWS.

  Este es el procedimiento recomendado para los hosts que no son de EC2 y que no pueden ser instancias administradas, ya que las credenciales están cifradas en ese determinado host y usuario.

- Si es necesario ejecutar el servicio AWSKinesisTap de Kinesis Agent para Windows con la cuenta del sistema predeterminada, debe utilizar un archivo de credenciales compartidas. El motivo es que la cuenta del sistema no tiene un perfil de usuarios de Windows que pueda usarse con el almacén de SDK. Los archivos de credenciales compartidas no están cifrados, por lo que no recomendamos este procedimiento. Para obtener información acerca de cómo utilizar archivos de configuración compartidos, consulteConfiguración de credenciales de AWSen laAWS SDK para .NET. Si utiliza este enfoque, le recomendamos que utilice el cifrado NTFS y el

acceso restringido a archivos con el archivo de configuración compartido. Es necesario que una plataforma de administración cambie regularmente las claves y que el archivo de configuración compartido se actualice cuando se modifiquen.

Aunque las claves de acceso y las claves secretas se pueden proporcionar directamente en las declaraciones de receptores, se desaconseja este enfoque, ya que las declaraciones no están cifradas.

Configuración de la autorización

Asocie las políticas adecuadas que rigen al usuario o rol de IAM que utilizará Kinesis Agent para Windows para transmitir datos a los servicios de AWS:

Kinesis Data Streams

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord",
                "kinesis:PutRecords"
            ],
            "Resource": "arn:aws:kinesis:*:*:stream/*"
        }
    ]
}
```

Para limitar la autorización a una región, cuenta o nombre de secuencia específicos, sustituya los asteriscos del ARN que correspondan por valores específicos. Para obtener más información, consulte la sección "Nombres de recursos de Amazon (ARN) para Kinesis Data Streams" de Control de acceso a los recursos de Amazon Kinesis Data Streams mediante IAM.

Kinesis Data Firehose

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

```
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Resource": "arn:aws:firehose:*:*:deliverystream/*"
        }
    ]
}
```

Para limitar la autorización a una región, cuenta o nombre de secuencia de entrega específicos, sustituya los asteriscos del ARN que correspondan por valores específicos. Para obtener más información, consulteControl del acceso con Amazon Kinesis Data Firehoseen laGuía del desarrollador de Amazon Kinesis Data Firehose.

CloudWatch

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        }
    ]
}
```

Para obtener más información, consulteIntroducción a la administración de permisos de acceso para sus recursos de CloudWatchen laGuía del usuario de Amazon CloudWatch Logs.

CloudWatch Logs con grupos de registros y secuencias de registro existentes

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

```
            "Sid": "VisualEditor3",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:*"
        },
        {
            "Sid": "VisualEditor4",
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:*:*:log-group:*:*:*"
        }
    ]
}
```

Para restringir el acceso a una región, cuenta, secuencia de registro o grupo de registros específicos, sustituya los asteriscos que correspondan en los ARN por los valores apropiados. Para obtener más información, consulteIntroducción a la administración de permisos de acceso para sus recursos de CloudWatch Logsen laGuía del usuario de Amazon CloudWatch Logs.

CloudWatch Logs con permisos adicionales para que Kinesis Agent para Windows pueda crear grupos de registros y secuencias de registro

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor5",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:*"
        },
        {
            "Sid": "VisualEditor6",
```

```
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:*:*:log-group:*:*:*"
        },
        {
            "Sid": "VisualEditor7",
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "*"
        }
    ]
}
```

Para restringir el acceso a una región, cuenta, secuencia de registro o grupo de registros específicos, sustituya los asteriscos que correspondan en los ARN por los valores apropiados. Para obtener más información, consulteIntroducción a la administración de permisos de acceso para sus recursos de CloudWatch Logsen laGuía del usuario de Amazon CloudWatch Logs.

Permisos necesarios para la ampliación de variables de etiquetas de EC2

Para poder utilizar ampliaciones de variables con el prefijo `ec2tag`, se necesita el permiso `ec2:Describe*`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "VisualEditor8",
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    }
   ]
}
```

> **Note**
>
> Puede combinar varias instrucciones en una única política siempre que el `Sid` de cada instrucción sea único dentro de dicha política. Para obtener más información acerca de la creación de políticas, consulteCreación de políticas de IAMen laGuía del usuario de IAM.

Configuración**ProfileRefreshingAWSCredentialProvider**Actualizar credenciales de AWS

Si utiliza AWS Systems Manager para entornos híbridos para administrar credenciales de AWS, Systems Manager rotará las credenciales de sesión en`c:\Windows\System32\config\systemprofile\.aws\credentials`. Para obtener más información acerca de Systems Manager para entornos híbridos, consulteConfiguración de AWS Systems Manager para entornos híbridosen laGuía del usuario de AWS Systems Manager.

Debido a que AWS .net SDK no recopila las nuevas credenciales automáticamente, proporcionamos el`ProfileRefreshingAWSCredentialProvider`para actualizar las credenciales.

Puede utilizar la`CredentialRef`de cualquier configuración de AWS sync para hacer referencia a un atributo`Credentials`en la que se establece la`CredentialType`El atributo se establece en`ProfileRefreshingAWSCredentialProvider`Como se muestra en el siguiente ejemplo.

```
{
    "Sinks": [{
        "Id": "myCloudWatchLogsSink",
        "SinkType": "CloudWatchLogs",
        "CredentialRef": "ssmcred",
        "Region": "us-west-2",
        "LogGroup": "myLogGroup",
        "LogStream": "myLogStream"
    }],
    "Credentials": [{
        "Id": "ssmcred",
        "CredentialType": "ProfileRefreshingAWSCredentialProvider",
        "Profile": "default",
        "FilePath": "%USERPROFILE%//.aws//credentials",
        "RefreshingInterval": 300
    }]
}
```

Una definición de credencial consta de los siguientes atributos como pares clave-valor.

`Id`

Define la cadena que las definiciones de sumidero pueden especificar usando`CredentialRef`para hacer referencia a esta configuración de credenciales.

`CredentialType`

Establezca en la cadena literal`ProfileRefreshingAWSCredentialProvider`.

`Profile`

Opcional. El valor predeterminado es `default`.

`FilePath`

Opcional. Especifica la ruta al archivo de credenciales de AWS. Si se omite, `%USERPROFILE %/.aws/credentials` es la opción predeterminada.

`RefreshingInterval`

Opcional. Frecuencia con la que se actualizan las credenciales, en segundos. Si se omite, `300` es la opción predeterminada.

Configuración de decoraciones de receptores

Las declaraciones de receptores pueden incluir de forma opcional pares clave-valor que especifiquen otros datos para transmitirlos a los diferentes servicios de AWS y mejorar así los registros que se recopilan del origen.

`TextDecoration`

Utilice este par clave-valor cuando `Format` esté especificado en la declaración del receptor. El valor es una cadena con formato especial en la que se produce una sustitución de variables. Por ejemplo, supongamos que el valor de `TextDecoration` que se proporciona a un receptor es `"{ComputerName}:::{timestamp:yyyy-MM-dd HH:mm:ss}:::{_record}"`. Si un origen emite una entrada de registro que contiene el texto `The system has resumed from sleep.` y ese origen está conectado al receptor a través de una canalización, el texto `MyComputer1:::2017-10-26 06:14:22:::The system has resumed from sleep.` se transmite al servicio de AWS asociado con ese tipo de receptor. La variable `{_record}` hace referencia al registro de texto original proporcionado por el origen.

`ObjectDecoration`

Utilice este par clave-valor cuando `Format` esté especificado en la declaración del receptor para añadir información adicional antes de la serialización de los registros. Por ejemplo, supongamos que se proporciona un `ObjectDecoration` con el valor

`"ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd HH:mm:ss}"` a un receptor en el que el valor de `Format` es JSON. El archivo JSON resultante que se transmite al servicio de AWS asociado con el tipo de receptor contiene los siguientes pares clave-valor, además de los datos originales del origen:

```
{
    ComputerName: "MyComputer2",
    DT: "2017-10-17 21:09:04"
}
```

Para ver un ejemplo del uso de `ObjectDecoration`, consulte Tutorial: Transmitir archivos de registro JSON a Amazon S3 mediante Kinesis Agent para Windows.

`ObjectDecorationEx`

Especifica una expresión, que permite una extracción y formato de datos más flexibles en comparación con`ObjectDecoration`. Este campo se puede utilizar cuando el formato del fregadero es`json`. A continuación se muestra la sintaxis de expresión.

```
"ObjectDecorationEx":
 "attribute1={expression1};attribute2={expression2};attribute3={expression3}(;...)"
```

Por ejemplo, la siguiente frase`ObjectDecorationEx`Atributo:

```
"ObjectDecorationEx":
 "host={env:ComputerName};message={upper(_record)};time={format(_timestamp,
 'yyyyMMdd')}"
```

Transforma el registro literal:

```
System log message
```

En un objeto JSON de la siguiente manera, con los valores devueltos por las expresiones:

```
{
    "host": "EC2AMAZ-1234",
    "message": "SYSTEM LOG MESSAGE",
    "time": "20210201"
}
```

Para obtener más información sobre la formulación de expresiones, consulteConsejos para escribir expresiones. La mayoría de losObjectDecorationdebería funcionar utilizando la nueva sintaxis con la excepción de las variables de marca de tiempo. A{timestamp:yyyyMMdd}enObjectDecorationse expresa como{format(_timestamp,'yyyyMMdd')}inObjectDecorationEx.

`TextDecorationEx`

Especifica una expresión, que permite una extracción y formato de datos más flexibles en comparación conTextDecoration, como se muestra en el siguiente ejemplo.

```
"TextDecorationEx": "Message '{lower(_record)}' at {format(_timestamp, 'yyyy-MM-
dd')}"
```

Puede usarTextDecorationExpara componer objetos JSON. Utilice '@ {'para escapar de la llave abierta, tal y como se muestra en el siguiente ejemplo.

```
"TextDecorationEx": "@{ \"var\": \"{upper($myvar1)}\" }"
```

Si el tipo de origen conectado al receptor es `DirectorySource`, el receptor puede utilizar tres variables adicionales:

`_FilePath`

Ruta completa del archivo de registro.

`_FileName`

Nombre y extensión del archivo.

`_Position`

Número entero que representa el lugar del archivo de registro en que se encuentra la entrada del registro.

Estas variables son útiles cuando se utiliza un origen que recopila entradas de registro de varios archivos conectados a un receptor que transmite todos los registros en una única secuencia. Al inyectar los valores de estas variables en la transmisión de registros, se pueden realizar análisis posteriores en la canalización de datos para ordenar los registros por archivos y por su ubicación dentro de cada archivo.

Consejos para escribir expresiones

Una expresión puede ser uno de los siguientes:

- Una expresión de variable.
- Una expresión constante, por ejemplo,`'hello'`,`1`,`1.21`,`null`,`true`,`false`.
- Expresión de invocación que llama a una función, tal y como se muestra en el siguiente ejemplo.

```
regexp_extract('Info: MID 118667291 ICID 197973259 RID 0 To: <jd@acme.com>', 'To: (\\
\\S+)', 1)
```

Caracteres especiales

Se requieren dos barras diagonales inversas para escapar de caracteres especiales.

Nesting

Las invocaciones de función se pueden anidar, tal y como se muestra en el siguiente ejemplo.

```
format(date(2018, 11, 28), 'MMddyyyy')
```

Variables

Hay tres tipos de variables: local, meta y global.

- Variables localesEmpieza por una`$`tales como`$message`. Se utilizan para resolver la propiedad del objeto de evento, una entrada si el evento es un diccionario, o un atributo si el evento es un objeto JSON. Si la variable local contiene espacio o caracteres especiales, utilice una variable local entre comillas como`$'date created'`.
- Variables metaComenzar con un guion bajo (`_`) y se utilizan para resolver los metadatos del evento. Todos los tipos de eventos admiten las siguientes metavariables.

  `_timestamp`

  La marca temporal del evento.

  `_record`

  La representación del texto sin formato del evento.

  Los eventos de registro admiten las siguientes metavariables adicionales.

`_filepath`

`_filename`

`_position`

`_linenumber`

- Variables globalesresolver variables de entorno, metadatos de instancia EC2 o EC2tag. Para lograr un mejor rendimiento, le recomendamos que utilice el prefijo para limitar el ámbito de búsqueda, como`{env:ComputerName}`,`{ec2:InstanceId}`, y`{ec2tag:Name}`.

Funciones integradas

Kinesis Agent para Windows admite las siguientes funciones integradas. Si alguno de los argumentos son`NULL`y la función no está diseñada para manejar`NULL`, un`NULL`se devuelve.

```
//string functions
int length(string input)
string lower(string input)
string lpad(string input, int size, string padstring)
string ltrim(string input)
string rpad(string input, int size, string padstring)
string rtrim(string input)
string substr(string input, int start)
string substr(string input, int start, int length)
string trim(string input)
string upper(string str)

//regular expression functions
string regexp_extract(string input, string pattern)
string regexp_extract(string input, string pattern, int group)

//date functions
DateTime date(int year, int month, int day)
DateTime date(int year, int month, int day, int hour, int minute, int second)
DateTime date(int year, int month, int day, int hour, int minute, int second, int
 millisecond)

//conversion functions
```

```
int? parse_int(string input)
decimal? parse_decimal(string input)
DateTime? parse_date(string input, string format)
string format(object o, string format)

//coalesce functions
object coalesce(object obj1, object obj2)
object coalesce(object obj1, object obj2, object obj3)
object coalesce(object obj1, object obj2, object obj3, object obj4)
object coalesce(object obj1, object obj2, object obj3, object obj4, object obj5)
object coalesce(object obj1, object obj2, object obj3, object obj4, object obj5, object
 obj6)
```

Configuración de sustituciones de variables de receptor

Las declaraciones de receptores `KinesisStream`, `KinesisFirehose` y `CloudWatchLogs` necesitan un par clave-valor `LogStream` o `StreamName`. El valor de estos pares puede contener referencias a variables que resuelve automáticamente el agente de Kinesis para Windows. Para`CloudWatchLogs`, el`LogGroup`El par clave-valor también es necesario y puede contener referencias de variables que resuelve automáticamente el agente de Kinesis para Windows. Las variables se especifican utilizando la plantilla {*`prefix:variablename`*}, donde *`prefix:`* es opcional. Los prefijos admitidos son los siguientes:

- `env`— La referencia de variable se resuelve en el valor de la variable de entorno que tiene el mismo nombre.
- `ec2`— La referencia de variable se resuelve en los metadatos de la instancia de EC2 que tienen el mismo nombre.
- `ec2tag`— La referencia de variable se resuelve en el valor de la etiqueta de instancia de EC2 que tiene el mismo nombre. El permiso `ec2:Describe*` es necesario para obtener acceso a las etiquetas de las instancias. Para obtener más información, consulte Permisos necesarios para la ampliación de variables de etiquetas de EC2.

Si no se especifica el prefijo y hay una variable de entorno con el mismo nombre que `variablename`, la referencia de variable se resuelve en el valor de la variable de entorno. De lo contrario, si `variablename` es `hostname` o `instance_id`, la referencia de variable se resuelve en el valor de los metadatos de EC2 que tienen el mismo nombre. En cualquier otro caso, la referencia de variable no se resuelve.

A continuación, se muestran algunos ejemplos de pares clave-valor válidos con referencias de variables:

```
"LogStream": "LogStream_{instance_id}"
"LogStream": "LogStream_{hostname}"
"LogStream": "LogStream_{ec2:local-hostname}"
"LogStream": "LogStream_{computername}"
"LogStream": "LogStream_{env:computername}"
```

En las declaraciones de receptores `CloudWatchLogs`, se puede utilizar una variable de marca temporal con formato especial que permite que la marca temporal de la entrada original del registro o el evento del origen modifique el nombre de la secuencia de registro. El formato es el siguiente `{timestamp:timeformat}`. Vea el siguiente ejemplo:

```
"LogStream": "LogStream_{timestamp:yyyyMMdd}"
```

Si la entrada del registro o el evento se hubiera generado el 5 de junio de 2017, el valor del par clave-valor `LogStream` del ejemplo anterior se resolvería en `"LogStream_20170605"`.

Si se ha autorizado, el tipo de receptor `CloudWatchLogs` puede crear automáticamente nuevas secuencias de registro cuando sea necesario en función de los nombres generados. Esto no puede hacerse con otros tipos de receptores, ya que deberían realizarse otros ajustes de configuración aparte del nombre de la secuencia.

Existen sustituciones de variables especiales que se producen en el texto y en las decoraciones de objetos. Para obtener más información, consulte Configuración de decoraciones de receptores.

Configuración de colas de receptores

Las declaraciones de receptores `KinesisStream`, `KinesisFirehose` y `CloudWatchLogs` pueden permitir de forma opcional que se pongan en cola los recursos que no han podido realizar correctamente la transmisión al servicio de AWS asociado con esos tipos de receptores por problemas de conectividad temporales. Para permitir la creación de colas y que la transmisión se intente de nuevo automáticamente cuando se restablezca la conectividad, utilice los siguientes pares clave-valor en las declaraciones de receptores:

`QueueType`

Especifica el tipo de mecanismo de creación de colas que se va a utilizar. El único valor admitido es `file`, lo que indica que los registros deben ponerse en cola en un archivo. Este par clave-valor es necesario para poder habilitar la característica de creación de colas de Kinesis Agent para Windows. Si no se especifica, el comportamiento predeterminado es que las colas solo se crean en memoria y no se puede realizar la transmisión cuando se alcanzan los límites de creación de colas en memoria.

`QueuePath`

Especifica la ruta de la carpeta que contiene los archivos de los registros en cola. Este par clave-valor es opcional. El valor predeterminado es `%PROGRAMDATA%\KinesisTap\Queue\SinkId`, donde SinkId es el identificador asignado como el valor del `Id` de la declaración del receptor.

`QueueMaxBatches`

Limita la cantidad total de espacio que Kinesis Agent para Windows puede consumir cuando los registros se ponen en cola para su transmisión. La cantidad de espacio está restringida al valor de este par clave-valor multiplicado por el número máximo de bytes por lote. El número máximo de bytes por lote en los tipos de receptores `KinesisStream`, `KinesisFirehose` y `CloudWatchLogs` son 5 MB, 4 MB y 1 MB, respectivamente. Cuando se alcanza este límite, los errores de transmisión no se ponen en cola y se notifican como errores no recuperables. Este par clave-valor es opcional. El valor predeterminado es de 10 000 lotes.

Configuración de un proxy para receptores

Si desea configurar un proxy para todos los tipos de receptores de Kinesis Agent para Windows que tengan a acceso a los servicios de AWS, edite el archivo de configuración de Kinesis Agent para Windows que se encuentra en`%Program Files%\Amazon\KinesisTap\AWSKinesisTap.exe.config`. Para obtener instrucciones, consulte la`proxy`sección enReferencia de los archivos de configuración para el AWS SDK for .NETen laGuía para desarrolladores de AWS SDK para .NET.

Configurar la resolución de variables en más atributos de sumidero

El siguiente ejemplo muestra una configuración de receptores que usa el`Region`Variable de entorno para el valor de`Region`Par clave-valor de atributo. Para`RoleARN`, especifica la clave de etiqueta `EC2MyRoleARN`, que evalúa el valor asociado a esa clave.

```
"Id": "myCloudWatchLogsSink",
"SinkType": "CloudWatchLogs",
"LogGroup": "EC2Logs",
"LogStream": "logs-{instance_id}"
"Region": "{env:Region}"
"RoleARN": "{ec2tag:MyRoleARN}"
```

Configuración de puntos finales regionales de AWS STS al utilizar la propiedad RoleARN en los sumideros de AWS

Esta característica solo se aplica si usa Kinesistap en Amazon EC2 y usa el`RoleARN`de los sumideros de AWS para asumir un rol de IAM externo para autenticarse con los servicios de AWS de destino.

Estableciendo`UseSTSRegionalEndpoints`De `atrue`, puede especificar que un agente use el extremo regional (por ejemplo,`https://sts.us-east-1.amazonaws.com`) en lugar del punto final global (por ejemplo,`https://sts.amazonaws.com`). El uso de un extremo STS regional reduce la latencia de ida y vuelta para la operación y limita el impacto de las fallas en el servicio global de endpoints.

Configuración de VPC Endpoint para los sumideros de AWS

Puede especificar un extremo de VPC en la configuración del receptor para`CloudWatchLogs`,`CloudWatch`,`KinesisStreams`, y`KinesisFirehose`tipos de fregadero. Un punto de enlace de la VPC le permite conectar de forma privada la VPC a los servicios de AWS compatibles y a servicios de punto de enlace de la VPC habilitados por AWS PrivateLink sin necesidad de una gateway de Internet, un dispositivo NAT, una conexión de VPN o una conexión de AWS Direct Connect. Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con los recursos del servicio. El tráfico entre su VPC y el servicio no sale de la red de Amazon. Para obtener más información, consultePuntos de conexión de la VPCen laGuía del usuario de Amazon VPC.

Puede especificar el extremo de la VPC mediante la herramienta`ServiceURL`Propiedad como se muestra en el siguiente ejemplo de una`CloudWatchLogs`configuración de sumidero. Establezca el valor de`ServiceURL`al valor que se muestra en la pestañaDetalles del punto de enlace de la VPCmediante la consola de Amazon VPC.

```
{
```

```
    "Id": "myCloudWatchLogsSink",
    "SinkType": "CloudWatchLogs",
    "LogGroup": "EC2Logs",
    "LogStream": "logs-{instance_id}",
    "ServiceURL":"https://vpce-ab1c234de56-ab7cdefg.logs.us-east-1.vpce.amazonaws.com"
}
```

Configuración de un medio alternativo de proxy

Esta función le permite configurar un servidor proxy en una configuración de receptor utilizando el soporte de proxy integrado en AWS SDK en lugar de .NET. Anteriormente, la única forma de configurar el agente para que use un proxy era usar una característica nativa de .NET, que enrutaba automáticamente todas las solicitudes HTTP/S a través del proxy definido en el archivo proxy.

Si está utilizando actualmente el agente con un servidor proxy, no necesita cambiar para utilizar este método.

Puede utilizar la`ProxyHost`y`ProxyPort`Para configurar un proxy alternativo tal y como se muestra en el siguiente ejemplo.

```
{
    "Id": "myCloudWatchLogsSink",
    "SinkType": "CloudWatchLogs",
    "LogGroup": "EC2Logs",
    "LogStream": "logs-{instance_id}",
    "Region": "us-west-2",
    "ProxyHost": "myproxy.mydnsdomain.com",
    "ProxyPort": "8080"
}
```

Declaraciones de canalizaciones

UsarDeclaraciones canalizacionespara conectar una fuente (consulteDeclaraciones de origen) a un fregadero (verDeclaraciones de receptores) en Amazon Kinesis Agent para Microsoft Windows. Las declaraciones de canalizaciones se expresan como un objeto JSON. Una vez que se inicia Kinesis Agent para Windows, comienzan a recopilarse los registros, los eventos o las métricas del origen de una determinada canalización. Estos datos se transmiten a diversos servicios de AWS utilizando el receptor asociado a dicha canalización.

A continuación, se muestra un ejemplo de una declaración de canalización :

```
{
   "Id": "MyAppLogToCloudWatchLogs",
   "SourceRef": "MyAppLog",
   "SinkRef": "MyCloudWatchLogsSink"
}
```

Temas

Configuración de canalizaciones

Todas las declaraciones de canalizaciones pueden contener los siguientes pares clave-valor:

`Id`

Especifica el nombre de la canalización (obligatorio). Debe ser único en el archivo de configuración.

`Type`

Especifica el tipo de transformación (si procede) que la canalización aplica cuando los datos de registro se transfieren del origen al receptor. El único valor admitido es `RegexFilterPipe`. Este valor permite que la representación textual subyacente de la entrada del registro pueda filtrarse con expresiones regulares. El filtrado puede reducir los costos de transmisión y almacenamiento al enviar exclusivamente las entradas de registro pertinentes a fases posteriores de la canalización de datos. Este par clave-valor es opcional. El valor predeterminado no proporciona ninguna transformación.

`FilterPattern`

Especifica la expresión regular de las canalizaciones `RegexFilterPipe` que se utiliza para filtrar las entradas de registro recopiladas por el origen antes de transferirlas al receptor. Las canalizaciones de tipo `RegexFilterPipe` transmiten las entradas de registro cuando la expresión regular encuentra coincidencias con la representación textual subyacente del registro. Las entradas de registro estructuradas que se generan (por ejemplo, cuando se utiliza el par clave-valor `ExtractionPattern` en una declaración `DirectorySource`) se pueden filtrar con el mecanismo `RegexFilterPipe`, ya que este mecanismo trabaja con la representación textual

original antes de que se aplique ninguna conversión. Este par clave-valor es opcional, pero debe proporcionarse si la canalización especifica el tipo `RegexFilterPipe`.

A continuación, se muestra un ejemplo de una declaración de canalización `RegexFilterPipe`:

```
{
 "Id": "MyAppLog2ToFirehose",
 "Type": "RegexFilterPipe",
 "SourceRef": "MyAppLog2",
 "SinkRef": "MyFirehose",
 "FilterPattern": "^(10|11),.*",
 "IgnoreCase": false,
 "Negate": false
}
```

`SourceRef`

Especifica el nombre (el valor del par clave-valor `Id`) de la declaración de origen que define el origen que recopila los datos de registro, eventos y métricas de la canalización (obligatorio).

`SinkRef`

Especifica el nombre (el valor del par clave-valor `Id`) de la declaración del receptor que define el receptor que recibe los datos de registro, eventos y métricas de la canalización (obligatorio).

`IgnoreCase`

Opcional. Acepta valores de`true`o`false`. Cuando se establece en`true`, la Regex coincidirá con los registros sin distinción entre mayúsculas y minúsculas.

`Negate`

Opcional. Acepta valores de`true`o`false`. Cuando se establece en`true`, la tubería reenviará los registros queNo lo hagaCoinciden con la expresión regular.

Para ver un ejemplo de un archivo de configuración completo que utiliza el tipo de canalización `RegexFilterPipe`, consulte Uso de canalizaciones.

Configuración del agente Kinesis para tuberías métricas de Windows

Hay un origen de métricas integrado llamado`_KinesisTapMetricsSource`que produce métricas sobre Kinesis Agent para Windows. Si hay un`CloudWatch`declaración de sumidero con

unaIddeMyCloudWatchSinkLa siguiente declaración de canalización de ejemplo transfiere las métricas generadas en en en Kinesis Agent para Windows a ese receptor:

```
{
   "Id": "KinesisAgentMetricsToCloudWatch",
   "SourceRef": "_KinesisTapMetricsSource",
   "SinkRef": "MyCloudWatchSink"
}
```

Para obtener más información sobre el origen de métricas integrado en Kinesis Agent para Windows, consulteOrigen de métricas integrado en Windows en.

Si el archivo de configuración también transmite métricas de los contadores de rendimiento de Windows, le recomendamos que utilice una canalización y un receptor diferentes en lugar de utilizar el mismo receptor para las métricas de Kinesis Agent para Windows y las métricas de los contadores de rendimiento de Windows.

Configuración de actualizaciones automáticas

Usar`appsettings.json`Para permitir que se actualicen automáticamente el archivo de configuración de Amazon Kinesis Agent para Microsoft Windows y el archivo de configuración de Kinesis Agent para Windows. Para controlar el comportamiento de las actualizaciones, especifique el par clave-valor `Plugins` en el mismo nivel del archivo de configuración que `Sources`, `Sinks` y `Pipes`.

El par clave-valor `Plugins` especifica la funcionalidad general complementaria que no se incluye específicamente en las categorías de los orígenes, receptores y canalizaciones. Por ejemplo, hay un complemento para actualizar el agente de Kinesis para Windows y otro para actualizar el archivo`appsettings.json`Archivo de configuración de. Los complementos se representan como objetos JSON y siempre tienen el par clave-valor `Type`. `Type` determina qué otros pares clave-valor pueden especificarse en el complemento. Los tipos de complementos compatibles son los siguientes:

`PackageUpdate`

Especifica que Kinesis Agent para Windows debe comprobar periódicamente el archivo de configuración de una versión del paquete. Si el archivo de versión del paquete indica que debe instalarse una versión diferente de Kinesis Agent para Windows, Kinesis Agent para Windows descarga esa versión y la instala. El par clave-valor del complemento `PackageUpdate` incluye:

`Type`

El valor debe ser la cadena `PackageUpdate` y es obligatorio.

`Interval`

Especifica la frecuencia en minutos con la que debe comprobarse si el archivo de versión del paquete ha experimentado cambios utilizando un formato de cadena. Este par clave-valor es opcional. Si no se especifica, el valor predeterminado es 60 minutos. Si el valor es inferior a 1, no se realiza ninguna comprobación.

`PackageVersion`

Especifica la ubicación del archivo JSON de la versión del paquete. El archivo puede estar en un recurso compartido de archivos (`file://`), un sitio web (`http://`), o Amazon S3 (`s3://`). Por ejemplo, un valor de`s3://mycompany/config/agent-package-version.json`indica que Kinesis Agent para Windows debe comprobar el contenido de la`config/agent-package-version.json`En el archivo`mycompany`Bucket de Amazon S3. Las actualizaciones deberían realizarse en función del contenido de ese archivo.

> **Note**
>
> El valor de la propiedad`PackageVersion`El par clave-valor distingue entre mayúsculas y minúsculas en Amazon S3.

En el siguiente ejemplo, se muestra el contenido de un archivo de versión del paquete:

```
{
    "Name": "AWSKinesisTap",
    "Version": "1.0.0.106",
    "PackageUrl": "https://s3-us-west-2.amazonaws.com/kinesis-agent-windows/
downloads/AWSKinesisTap.{Version}.nupkg"
}
```

La`Version`Especifica qué versión de Kinesis Agent para Windows debe instalarse si aún no se ha instalado. La referencia a la variable `{Version}` de `PackageUrl` resuelve el valor que se especifica para el par clave-valor `Version`. En este ejemplo, la variable se resuelve en la cadena `1.0.0.106`. Esta resolución de la variable se proporciona para que pueda haber un lugar específico en el archivo de versión del paquete en el que se almacene la versión deseada. Puede utilizar varios archivos de versión del paquete para controlar la velocidad de

publicación de nuevas versiones de Kinesis Agent para Windows y validar una nueva versión antes de realizar una implementación de mayor tamaño. Para revertir una implementación de Kinesis Agent para Windows, modifique uno o varios archivos de versión del paquete y especifique una versión anterior de Kinesis Agent para Windows que sepa que funciona en su entorno.

El valor del par clave-valor `PackageVersion` se verá afectado por la sustitución de variables para facilitar la selección automática de diferentes archivos de versión de paquetes. Para obtener más información sobre la sustitución de variables, consulte Configuración de sustituciones de variables de receptor.

`AccessKey`

Especifica qué clave de acceso debe utilizarse para autenticar el acceso al archivo de versión del paquete en Amazon S3. Este par clave-valor es opcional. No se recomienda utilizar este par clave-valor. Para conocer otros enfoques de autenticación recomendados, consulte Configuración de la autenticación .

`SecretKey`

Especifica qué clave secreta debe utilizarse para autenticar el acceso al archivo de versión del paquete en Amazon S3. Este par clave-valor es opcional. No se recomienda utilizar este par clave-valor. Para conocer otros enfoques de autenticación recomendados, consulte Configuración de la autenticación .

`Region`

Especifica qué punto de enlace de la región debe utilizarse para obtener acceso al archivo de versión del paquete desde Amazon S3. Este par clave-valor es opcional.

`ProfileName`

Especifica qué perfil de seguridad debe utilizarse para autenticar el acceso al archivo de versión del paquete en Amazon S3. Para obtener más información, consulte Configuración de la autenticación . Este par clave-valor es opcional.

`RoleARN`

Especifica qué rol debe adoptarse para autenticar el acceso al archivo de versión del paquete en Amazon S3 en un escenario de acceso entre cuentas. Para obtener más información, consulte Configuración de la autenticación . Este par clave-valor es opcional.

Si no se especifica un complemento `PackageUpdate`, no se comprueba ningún archivo de versión del paquete para determinar si es necesario actualizarlo.

`ConfigUpdate`

Especifica que Kinesis Agent para Windows debe comprobar periódicamente si hay un`appsettings.json`Archivo de configuración guardado en un recurso compartido de archivos, un sitio web o Amazon S3. Si hay un archivo de configuración actualizado, lo descarga e instala el agente de Kinesis para Windows.`ConfigUpdate`El par clave-valor incluye lo siguiente:

`Type`

El valor debe ser la cadena `ConfigUpdate` y es obligatorio.

`Interval`

Especifica la frecuencia en minutos con la que debe comprobarse si hay un nuevo archivo de configuración utilizando un formato de cadena. Este par clave-valor es opcional y, si no se especifica, el valor predeterminado es 5 minutos. Si el valor es inferior a 1, no se comprueba si hay un archivo de configuración actualizado.

`Source`

Especifica dónde debe buscarse un archivo de configuración actualizado. El archivo puede estar en un recurso compartido de archivos (`file://`), un sitio web (`http://`), o Amazon S3 (`s3://`). Por ejemplo, un valor des`3://mycompany/config/appsettings.json`indica que Kinesis Agent para Windows debe comprobar si hay actualizaciones en el`config/appsettings.json`En el archivo`mycompany`Bucket de Amazon S3.

> **Note**
>
> El valor de la propiedad`Source`El par clave-valor distingue entre mayúsculas y minúsculas en Amazon S3.

El valor del par clave-valor `Source` se verá afectado por la sustitución de variables para facilitar la selección automática de diferentes archivos de configuración. Para obtener más información sobre la sustitución de variables, consulte Configuración de sustituciones de variables de receptor.

`Destination`

Especifica dónde se va a almacenar el archivo de configuración en el equipo local. Puede ser una ruta relativa, una ruta completa o una ruta que contenga referencias a variables de entorno, como `%PROGRAMDATA%`. Si la ruta es relativa, lo es respecto a la ubicación en la

que está instalado el agente de Kinesis para Windows. Normalmente, el valor debería ser `.\appsettings.json`. Este par clave-valor es obligatorio.

`AccessKey`

Especifica qué clave de acceso debe utilizarse para autenticar el acceso al archivo de configuración de Amazon S3. Este par clave-valor es opcional. No se recomienda utilizar este par clave-valor. Para conocer otros enfoques de autenticación recomendados, consulte Configuración de la autenticación .

`SecretKey`

Especifica qué clave secreta debe utilizarse para autenticar el acceso al archivo de configuración de Amazon S3. Este par clave-valor es opcional. No se recomienda utilizar este par clave-valor. Para conocer otros enfoques de autenticación recomendados, consulte Configuración de la autenticación .

`Region`

Especifica qué punto de enlace de la región debe utilizarse para obtener acceso al archivo de configuración de Amazon S3. Este par clave-valor es opcional.

`ProfileName`

Especifica qué perfil de seguridad debe utilizarse para autenticar el acceso al archivo de configuración de Amazon S3. Para obtener más información, consulte Configuración de la autenticación . Este par clave-valor es opcional.

`RoleARN`

Especifica qué rol debe adoptarse para autenticar el acceso al archivo de configuración de Amazon S3 en un escenario de acceso entre cuentas. Para obtener más información, consulte Configuración de la autenticación . Este par clave-valor es opcional.

Si no se especifica un complemento `ConfigUpdate`, no se comprueba ningún archivo de configuración para determinar si es necesario actualizarlo.

A continuación, se muestra un ejemplo de un archivo de configuración `appsettings.json` que ilustra el uso de los complementos `PackageUpdate` y `ConfigUpdate`. En este ejemplo, hay un archivo de versión del paquete que se encuentra en la carpeta`mycompany`Bucket de Amazon S3 denominado`config/agent-package-version.json`. Aproximadamente cada dos horas, se comprueba si esta archivo ha experimentado cambios. Si se especifica una versión diferente de

Kinesis Agent para Windows en el archivo de versión del paquete, la versión del agente especificada se instala en la ubicación especificada en el archivo de versión del paquete.

Además, hay un`appsettings.json`almacenado en el archivo`mycompany`Bucket de Amazon S3 denominado`config/appsettings.json`. Aproximadamente, cada 30 minutos ese archivo se compara con el archivo de configuración actual. Si son diferentes, el archivo de configuración actualizado se descarga de Amazon S3 y se instala en la ubicación local habitual de`appsettings.json`Archivo de configuración de.

```
{
  "Sources": [
    {
      "Id": "ApplicationLogSource",
      "SourceType": "DirectorySource",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "RecordParser": "SingleLine"
    }
  ],
  "Sinks": [
    {
       "Id": "ApplicationLogKinesisFirehoseSink",
       "SinkType": "KinesisFirehose",
       "StreamName": "ApplicationLogFirehoseDeliveryStream",
       "Region": "us-east-1"
    }
    ],
  "Pipes": [
    {
      "Id": "ApplicationLogSourceToApplicationLogKinesisFirehoseSink",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "ApplicationLogKinesisFirehoseSink"
    }
  ],
  "Plugins": [
    {
      "Type": "PackageUpdate"
      "Interval": "120",
      "PackageVersion": "s3://mycompany/config/agent-package-version.json"
    },
    {
      "Type": "ConfigUpdate",
      "Interval": "30",
```

```
      "Source": "s3://mycompany/config/appsettings.json",
      "Destination": ".\appSettings.json"
    }
  ]
}
```

Ejemplos de configuración de Kinesis Agent para Windows

La`appsettings.json`es un documento JSON que controla el modo en que Amazon Kinesis Agent para Microsoft Windows recopila registros, eventos y métricas. También controla el modo en que Kinesis Agent para Windows transforma los datos y los transmite a los diferentes servicios de AWS. Para obtener más información sobre las declaraciones de orígenes, receptores y canalizaciones del archivo de configuración Declaraciones de origen, consulteDeclaraciones de receptores y Declaraciones de canalizaciones.

Las siguientes secciones contienen ejemplos de archivos de configuración diseñados para diferentes tipos de escenarios.

Temas

- Transmisión a desde varios orígenes de Kinesis Data Streams
- Transmisión a los receptores desde el registro de eventos de la aplicación de Windows
- Uso de canalizaciones
- Uso de varios orígenes y canalizaciones

Transmisión a desde varios orígenes de Kinesis Data Streams

El siguiente ejemplo`appsettings.json`Los archivos de configuración ilustran registros y eventos de transmisión a desde varios orígenes de Kinesis Data Streams y desde contadores de rendimiento de Windows a métricas de Amazon CloudWatch.

Analizador de registros **`SysLog`**, **`DirectorySource`**

El siguiente archivo transmite entradas de registro de formato syslog a partir de todos los archivos con un valor de`.log`extensión de archivo en el archivo`C:\LogSource\`al directorio`SyslogKinesisDataStream`Transmisión a Kinesis Data Streams en la región us-east-1. Para garantizar que todos los datos de los archivos de registro se envían aunque el agente se apague y se reinicie posteriormente, se establece un marcador. Una aplicación personalizada puede leer y procesar los registros procedentes de la secuencia `SyslogKinesisDataStream`.

```
{
  "Sources": [
    {
      "Id": "SyslogDirectorySource",
      "SourceType": "DirectorySource",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "RecordParser": "SysLog",
      "TimeZoneKind": "UTC",
      "InitialPosition": "Bookmark"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "SyslogKinesisDataStream",
      "Region": "us-east-1"
    }
  ],
  "Pipes": [
    {
      "Id": "SyslogDS2KSSink",
      "SourceRef": "SyslogDirectorySource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

Analizador de registros **`SingleLineJson`**, **`DirectorySource`**

El siguiente archivo transmite entradas de registro con formato JSON a partir de todos los archivos con un valor de `.log` extensión de archivo en el archivo `C:\LogSource\` al directorio `JsonKinesisDataStream` Transmisión a Kinesis Data Streams en la región us-east-1. Antes de la transmisión, los pares clave-valor de las claves `ComputerName` y `DT` se añaden a cada objeto JSON, con valores para el nombre del equipo y para la fecha y hora de procesamiento del registro. Una aplicación personalizada puede leer y procesar los registros procedentes de la secuencia `JsonKinesisDataStream`.

```
{
  "Sources": [
    {
```

```
      "Id": "JsonLogSource",
      "SourceType": "DirectorySource",
      "RecordParser": "SingleLineJson",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "InitialPosition": 0
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "JsonKinesisDataStream",
      "Region": "us-east-1",
      "Format": "json",
      "ObjectDecoration": "ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd
 HH:mm:ss}"
    }
  ],
  "Pipes": [
    {
      "Id": "JsonLogSourceToKinesisStreamSink",
      "SourceRef": "JsonLogSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

ExchangeLogSource

El siguiente archivo transmite entradas de registro generadas por Microsoft Exchange y almacenadas en archivos con el archivo `.log`extensión de en la extensión`C:\temp\ExchangeLog\`al directorio`ExchangeKinesisDataStream`Transmisión de datos de Kinesis en la región us-east-1 con formato JSON. Aunque los registros de Exchange no estén en formato JSON, Kinesis Agent para Windows puede analizarlos y convertirlos a formato JSON. Antes de la transmisión, los pares clave-valor de las claves `ComputerName` y `DT` se añaden a cada objeto JSON, con valores para el nombre del equipo y para la fecha y hora de procesamiento del registro. Una aplicación personalizada puede leer y procesar los registros procedentes de la secuencia `ExchangeKinesisDataStream`.

```
{
  "Sources": [
```

```
    {
       "Id": "ExchangeSource",
       "SourceType": "ExchangeLogSource",
       "Directory": "C:\\temp\\ExchangeLog\",
       "FileNameFilter": "*.log"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "ExchangeKinesisDataStream",
      "Region": "us-east-1",
      "Format": "json",
      "ObjectDecoration": "ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd
 HH:mm:ss}"
    }
  ],
  "Pipes": [
    {
      "Id": "ExchangeSourceToKinesisStreamSink",
      "SourceRef": "ExchangeSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

W3SVCLogSource

El siguiente archivo transmite entradas de registro de Internet Information Services (IIS) para Windows almacenadas en la ubicación estándar de esos archivos a la carpeta`IISKinesisDataStream`Transmisión a Kinesis Data Streams en la región us-east-1. Una aplicación personalizada puede leer y procesar los registros procedentes de la secuencia `IISKinesisDataStream`. IIS es un servidor web de Windows.

```
{
  "Sources": [
    {
       "Id": "IISLogSource",
       "SourceType": "W3SVCLogSource",
       "Directory": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
```

```
       "FileNameFilter": "*.log"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "IISKinesisDataStream",
      "Region": "us-east-1"
    }
  ],
  "Pipes": [
    {
      "Id": "IISLogSourceToKinesisStreamSink",
      "SourceRef": "IISLogSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

`WindowsEventLogSource` con Query

El siguiente archivo transmite eventos de registro desde el registro de sucesos del sistema Windows que tienen un nivel de`Critical`or`Error`(menor o igual que 2)`SystemKinesisDataStream`Transmisión de datos de Kinesis en la región us-east-1 con formato JSON. Una aplicación personalizada puede leer y procesar los registros procedentes de la secuencia `SystemKinesisDataStream`.

```
{
  "Sources": [
    {
         "Id": "SystemLogSource",
         "SourceType": "WindowsEventLogSource",
         "LogName": "System",
         "Query": "*[System/Level<=2]"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "SystemKinesisDataStream",
      "Region": "us-east-1",
```

```
      "Format": "json"
    }
  ],
  "Pipes": [
    {
      "Id": "SLSourceToKSSink",
      "SourceRef": "SystemLogSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

`WindowsETWEventSource`

El siguiente archivo transmite excepciones y eventos de seguridad de Microsoft Common Language Runtime (CLR) a la clase`ClrKinesisDataStream`Transmisión de datos de Kinesis en la región us-east-1 con formato JSON. Una aplicación personalizada puede leer y procesar los registros procedentes de la secuencia `ClrKinesisDataStream`.

```
{
  "Sources": [
    {
       "Id": "ClrETWEventSource",
       "SourceType": "WindowsETWEventSource",
       "ProviderName": "Microsoft-Windows-DotNETRuntime",
       "TraceLevel": "Verbose",
       "MatchAnyKeyword": "0x00008000, 0x00000400"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "ClrKinesisDataStream",
      "Region": "us-east-1",
      "Format": "json"
    }
  ],
  "Pipes": [
    {
      "Id": "ETWSourceToKSSink",
      "SourceRef": "ClrETWEventSource",
      "SinkRef": "KinesisStreamSink"
```

```
    }
  ]
}
```

WindowsPerformanceCounterSource

El siguiente archivo transmite contadores de rendimiento sobre el total de archivos abiertos, el total de intentos de inicio de sesión desde el reinicio, el número de lecturas de disco por segundo y el porcentaje de espacio libre en disco a métricas de CloudWatch en la región us-east-1. Puede representar estas métricas en CloudWatch, crear paneles a partir de los gráficos y establecer alarmas que envíen notificaciones cuando se superen los umbrales.

```
{
  "Sources": [
    {
      "Id": "PerformanceCounter",
      "SourceType": "WindowsPerformanceCounterSource",
      "Categories": [
        {
          "Category": "Server",
          "Counters": [
            "Files Open",
            "Logon Total"
          ]
        },
        {
          "Category": "LogicalDisk",
          "Instances": "*",
          "Counters": [
            "% Free Space",
            {
              "Counter": "Disk Reads/sec",
              "Unit": "Count/Second"
            }
          ]
        }
      ],
    }
  ],
  "Sinks": [
    {
      "Namespace": "MyServiceMetrics",
      "Region": "us-east-1",
```

```
      "Id": "CloudWatchSink",
      "SinkType": "CloudWatch"
    }
  ],
  "Pipes": [
    {
      "Id": "PerformanceCounterToCloudWatch",
      "SourceRef": "PerformanceCounter",
      "SinkRef": "CloudWatchSink"
    }
  ]
}
```

Transmisión a los receptores desde el registro de eventos de la aplicación de Windows

El siguiente ejemploappsettings.jsonLos archivos de configuración ilustran la transmisión de registros de eventos de la aplicación de Windows a varios receptores de Amazon Kinesis Agent para Microsoft Windows. Para ver ejemplos sobre el uso de los tipos de receptores `KinesisStream` y `CloudWatch`, consulte Transmisión a desde varios orígenes de Kinesis Data Streams.

`KinesisFirehose`

Las siguientes secuencias de archivos`Critical`o`Error`Eventos de registro de aplicaciones de Windows en el`WindowsLogFirehoseDeliveryStream`Transmisión de entrega de Kinesis Data Firehose en la región us-east-1. Si se interrumpe la conectividad con Kinesis Data Firehose, los eventos se ponen en cola en memoria. A continuación, si es necesario, se ponen en cola en un archivo del disco hasta que se restaura la conectividad. Después, los eventos van saliendo de las colas y se envían junto con los nuevos eventos que haya.

Puede configurar Kinesis Data Firehose para que guarde los datos transmitidos a varios tipos de almacenamiento y servicios de análisis de acuerdo con los requisitos de la canalización de datos.

```
{
  "Sources": [
    {
          "Id": "ApplicationLogSource",
          "SourceType": "WindowsEventLogSource",
          "LogName": "Application",
          "Query": "*[System/Level<=2]"
```

```
    }
  ],
  "Sinks": [
    {
       "Id": "WindowsLogKinesisFirehoseSink",
       "SinkType": "KinesisFirehose",
       "StreamName": "WindowsLogFirehoseDeliveryStream",
       "Region": "us-east-1",
       "QueueType": "file"
    }
    ],
  "Pipes": [
    {
      "Id": "ALSource2ALKFSink",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "WindowsLogKinesisFirehoseSink"
    }
  ]
}
```

CloudWatchLogs

Las siguientes secuencias de archivos`Critical`o`Error`Registro de eventos de registro de la aplicación de Windows a CloudWatch Logs de entradas de registro en`MyServiceApplicationLog-Group`grupo de registros. El nombre de cada secuencia comienza por `Stream-`. Estos nombres terminan por un año de cuatro dígitos, un mes de dos dígitos y un día de dos dígitos que indican la creación de la secuencia (por ejemplo, la secuencia `Stream-20180501` se creó el 1 de mayo de 2018).

```
{
  "Sources": [
    {
         "Id": "ApplicationLogSource",
         "SourceType": "WindowsEventLogSource",
         "LogName": "Application",
         "Query": "*[System/Level<=2]"
    }
  ],
  "Sinks": [
    {
      "Id": "CloudWatchLogsSink",
      "SinkType": "CloudWatchLogs",
```

```
      "LogGroup": "MyServiceApplicationLog-Group",
      "LogStream": "Stream-{timestamp:yyyyMMdd}",
      "Region": "us-east-1",
      "Format": "json"
    }
  ],
  "Pipes": [
    {
      "Id": "ALSource2CWLSink",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "CloudWatchLogsSink"
    }
  ]
}
```

Uso de canalizaciones

El siguiente archivo de configuración `appsettings.json` de ejemplo ilustra el uso de características relacionadas con las canalizaciones.

En este ejemplo, se transmiten entradas del registro de la carpeta`c:\LogSource\`a la`ApplicationLogFirehoseDeliveryStream`Transmisión de entrega de Kinesis Data Firehose. Solamente contiene las líneas que coinciden con la expresión regular especificada por el par clave-valor `FilterPattern`. En concreto, solo hay líneas del archivo de registro que comiencen por`10`or`11`se transmiten a Kinesis Data Firehose.

```
{
  "Sources": [
    {
      "Id": "ApplicationLogSource",
      "SourceType": "DirectorySource",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "RecordParser": "SingleLine"
    }
  ],
  "Sinks": [
    {
       "Id": "ApplicationLogKinesisFirehoseSink",
       "SinkType": "KinesisFirehose",
       "StreamName": "ApplicationLogFirehoseDeliveryStream",
       "Region": "us-east-1"
```

```
    }
    ],
  "Pipes": [
    {
      "Id": "ALSourceToALKFSink",
      "Type": "RegexFilterPipe",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "ApplicationLogKinesisFirehoseSink",
      "FilterPattern": "^(10|11),.*"
    }
  ]
}
```

Uso de varios orígenes y canalizaciones

El siguiente archivo de configuración `appsettings.json` de ejemplo ilustra el uso de varios orígenes y canalizaciones.

En este ejemplo, se transmiten registros de eventos de Windows de aplicaciones, seguridad y sistema a la carpeta`EventLogStream`Transmisión de entrega de Kinesis Data Firehose a través de tres orígenes, tres canalizaciones y un receptor.

```
{
    "Sources": [
  {
    "Id": "ApplicationLog",
    "SourceType": "WindowsEventLogSource",
    "LogName": "Application"
  },
  {
    "Id": "SecurityLog",
    "SourceType": "WindowsEventLogSource",
    "LogName": "Security"
  },
  {
    "Id": "SystemLog",
    "SourceType": "WindowsEventLogSource",
    "LogName": "System"
  }
    ],
    "Sinks": [
  {
```

```
    "Id": "EventLogSink",
    "SinkType": "KinesisFirehose",
    "StreamName": "EventLogStream",
    "Format": "json"
  },
    ],
    "Pipes": [
  {
    "Id": "ApplicationLogToFirehose",
    "SourceRef": "ApplicationLog",
    "SinkRef": "EventLogSink"
  },
  {
    "Id": "SecurityLogToFirehose",
    "SourceRef": "SecurityLog",
    "SinkRef": "EventLogSink"
  },
  {
    "Id": "SystemLogToFirehose",
    "SourceRef": "SystemLog",
    "SinkRef": "EventLogSink"
  }
    ]
}
```

Configuración de telemetría

Para facilitar el soporte técnico, de forma predeterminada, Amazon Kinesis Agent para Microsoft Windows recopila estadísticas sobre el funcionamiento del agente y las envía a AWS. Esta información no contiene información de identificación personal. Tampoco incluye ningún dato que usted haya recopilado o enviado a los servicios de AWS. Recopilamos aproximadamente 1—2 KB de estos datos métricos cada 60 minutos.

Puede cancelar la recopilación y la transmisión de estas estadísticas. Para ello, añada la siguiente par clave-valor al archivo de configuración `appsettings.json` en el mismo nivel en el que se encuentran los orígenes, los receptores y las canalizaciones:

```
"Telemetrics":
    { "off": "true" }
```

Por ejemplo, el siguiente archivo de configuración configura un origen, un receptor y una canalización y deshabilita los datos de telemetría:

```
{
  "Sources": [
    {
      "Id": "ApplicationLogSource",
      "SourceType": "DirectorySource",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "RecordParser": "SingleLine"
    }
  ],
  "Sinks": [
    {
       "Id": "ApplicationLogKinesisFirehoseSink",
       "SinkType": "KinesisFirehose",
       "StreamName": "ApplicationLogFirehoseDeliveryStream",
       "Region": "us-east-1"
    }
    ],
  "Pipes": [
    {
      "Id": "ApplicationLogSourceToApplicationLogKinesisFirehoseSink",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "ApplicationLogKinesisFirehoseSink"
    }
  ],
  "Telemetrics":
    {
      "off": "true"
    }
}
```

Cuando la telemetría está habilitada, recopilamos las siguientes métricas:

`ClientId`

ID único que se asigna automáticamente cuando se instala el software.

`ClientTimestamp`

Fecha y hora en que se recopilaron los datos de telemetría.

`OSDescription`

Descripción del sistema operativo.

`DotnetFramework`

Versión del marco dotnet actual.

`MemoryUsage`

Cantidad de memoria consumida por Kinesis Agent para Windows (MB).

`CPUUsage`

Cantidad porcentual Kinesis de CPU por parte de en formato decimal. Por ejemplo, 0,01 significa 1 %.

`InstanceId`

Identificador de instancia de Amazon EC2 si Kinesis Agent para Windows se está ejecutando en una instancia de Amazon EC2.

`InstanceType (string)`

Tipo de instancia de Amazon EC2 si Kinesis Agent para Windows se está ejecutando en una instancia de Amazon EC2.

Además, recopilamos las métricas que se indican en Lista de métricas del agente Kinesis para Windows.

Tutorial: Transmitir archivos de registro JSON a Amazon S3 mediante Kinesis Agent para Windows

En este tutorial, se explica paso a paso cómo se configura una canalización de datos con Amazon Kinesis Agent para Microsoft Windows (Kinesis Agent para Windows).

El tutorial incluye los siguientes pasos:

- Uso de Kinesis Agent para Windows para transmitir archivos de registro en formato JSON aAmazon Simple Storage Service (Amazon S3)a través deAmazon Kinesis Data Firehose. Para obtener información acerca de Kinesis Agent para Windows, consulte¿Qué es Amazon Kinesis Agent para Microsoft Windows?.
- Mejora de los datos de registro con decoraciones de objetos antes de transmitirlos. Para obtener más información, consulte Configuración de decoraciones de receptores.
- UsoAmazon AthenaPara buscar tipos concretos de entradas de registro.

Prerequisites

Si aún no tiene una cuenta de AWS, siga las instrucciones que se describen enConfiguración de una cuenta de AWSpara obtener uno.

Temas

Paso 1: Configuración de los servicios de AWS

Siga estos pasos para preparar el entorno para la transmisión datos de registro a Amazon Simple Storage Service (Amazon S3) con Amazon Kinesis Agent para Microsoft Windows. Para obtener más información y conocer los requisitos previos, consulte Tutorial: Transmisión de archivos de registro JSON a Amazon S3.

Utilice AWS Management Console para configurar AWS Identity and Access Management (IAM), Amazon S3, Kinesis Data Firehose y Amazon Elastic Compute Cloud (Amazon EC2) para preparar la transmisión de datos de registro desde una instancia de EC2 a Amazon S3.

Temas

Configuración de políticas y roles de IAM

Cree la siguiente política, que autoriza a Kinesis Agent para Windows a transmitir registros a una secuencia de entrega de Kinesis Data Firehose específica:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Resource": "arn:aws:firehose:region:account-id:deliverystream/log-
delivery-stream"
        }
    ]
}
```

Reemplazar*region*Con el nombre de la región de AWS en la que se creará la secuencia de entrega de Kinesis Data Firehose (`us-east-1`, por ejemplo). Sustituya *`account-id`* por el ID de 12 dígitos de la cuenta de AWS en la que se va a crear la secuencia de entrega.

En la barra de navegación, elijaSoportey, a continuación,Centro de soporte de. El número de cuenta (ID) de 12 dígitos que ha iniciado sesión en este momento aparece en laCentro de soporte dePanel de navegación de.

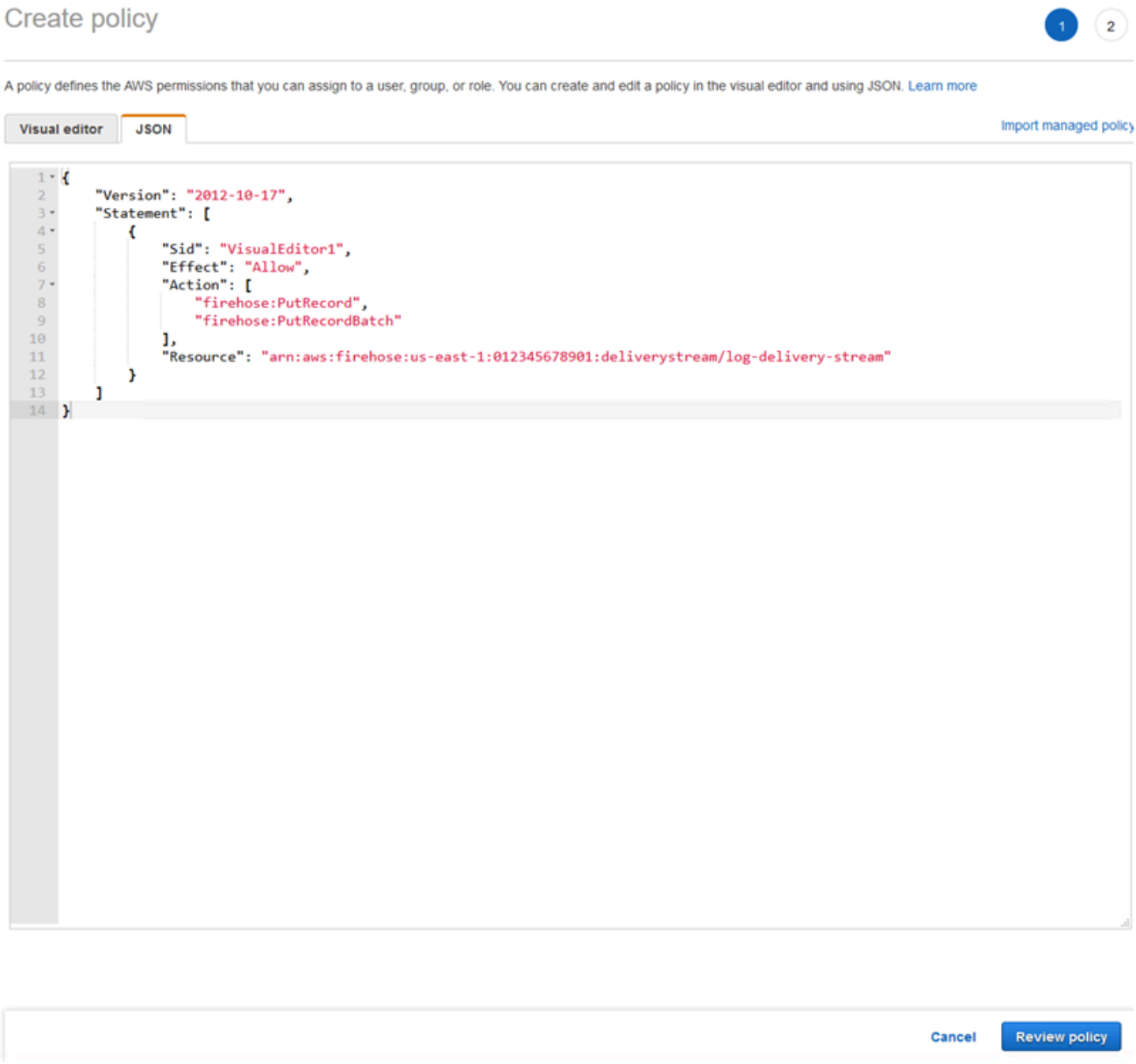
Cree la política siguiendo el procedimiento que se describe a continuación. Llame `log-delivery-stream-access-policy` a la política.

Para crear una política con el editor de políticas JSON

1. Inicie sesión en la consola de administración de AWS y abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la izquierda, seleccione Políticas.

   Si es la primera vez que elige Policies (Políticas), aparecerá la página Welcome to Managed Policies (Bienvenido a políticas administradas). Elija Get Started.
3. En la parte superior de la página, seleccione Crear política.
4. Seleccione la pestaña JSON.
5. Especifique un documento de política JSON. Para obtener información sobre el lenguaje de políticas de IAM, consulteReferencia de políticas JSON de IAMen laGuía del usuario de IAM.
6. Cuando haya terminado, seleccione Review policy. El validador de políticas notifica los errores de sintaxis.

   > **Note**
   >
   > Puede alternar entre las pestañas Visual editor (Editor visual) y JSON en cualquier momento. Sin embargo, si realiza cambios o eligeRevisión de políticas deen laVisual editor (Editor visual)Es posible que IAM reestructure la política para optimizarla para el editor visual. Para obtener más información, consulteReestructuración de políticasen laGuía del usuario de IAM.
7. En la página Review Policy (Revisar política), especifique un nombre en el campo Name (Nombre) y una descripción (opcional) en el campo Description (Descripción) para la política que está creando. Revise la política Summary para ver los permisos concedidos por su política. A continuación, elija Create policy para guardar su trabajo.



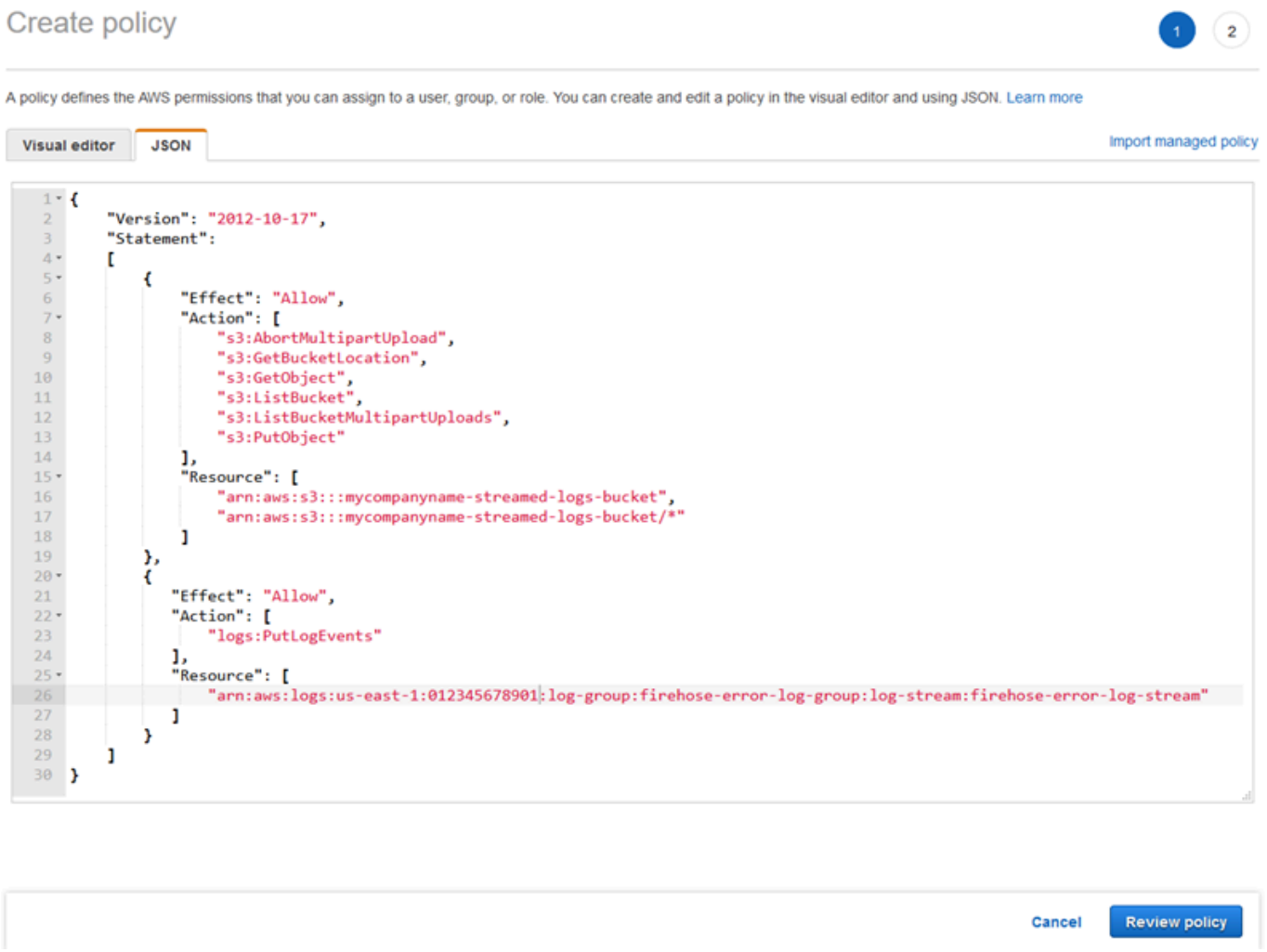
Para crear el rol que proporciona a Kinesis Data Firehose acceso a un bucket de S3

1. Utilizando el procedimiento anterior, cree una política llamada `firehose-s3-access-policy` que se defina mediante el código JSON siguiente:

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

```
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
           "Effect": "Allow",
           "Action": [
               "logs:PutLogEvents"
           ],
           "Resource": [
               "arn:aws:logs:region:account-id:log-group:firehose-error-log-
group:log-stream:firehose-error-log-stream"
           ]
        }
    ]
}
```

Sustituya *bucket-name* por el nombre único del bucket donde se van a almacenar los registros. Reemplazar*region*En la región de AWS en la que se van a crear el grupo de registros y la secuencia de registros de CloudWatch. Estos sirven para registrar los errores que se producen durante la transmisión de datos a Amazon S3 a través de Kinesis Data Firehose. Sustituya *account-id* por el ID de 12 dígitos de la cuenta en la que se van a crear el grupo de registros y la secuencia de registro.



2. En el panel de navegación de la consola de IAM, seleccione Roles y, a continuación, seleccione Create role.
3. Elija el iconoServicio de AWSSeleccione el tipo de rol y, a continuación, elija laKinesisservicio de.
4. SeleccionarKinesis Data FirehosePara este caso de uso y, a continuación, elijaSiguiente: Permisos.
5. En el cuadro de búsqueda, escriba**`firehose-s3-access-policy`**Elija esa política y, a continuación, elijaSiguiente: Review (Revisar).
6. En el cuadro Nombre del rol, escriba **`firehose-s3-access-role`**.
7. Elija Create role.

Para crear el rol que se va a asociar al perfil de la instancia EC2 que va a ejecutar Kinesis Agent para Windows

1. En el panel de navegación de la consola de IAM, seleccione Roles y, a continuación, seleccione Create role.
2. Elija el iconoServicio de AWSSeleccione y, a continuación, elijaEC2.
3. SeleccionarSiguiente: Permisos.
4. En el cuadro de búsqueda, escriba **`log-delivery-stream-access-policy`**.
5. Elija la política y, a continuación, elijaSiguiente: Review (Revisar).
6. En el cuadro Nombre del rol, escriba **`kinesis-agent-instance-role`**.
7. Elija Create role.

Cree el bucket de Amazon S3

Cree el bucket de S3 en el que Kinesis Data Firehose transmite los registros.

Si desea crear el bucket de S3 para almacenar registros

1. Abra la consola de Amazon S3 en https://console.aws.amazon.com/s3.
2. Elija Create bucket (Crear bucket).
3. En el cuadro Nombre del bucket, escriba el nombre único del bucket de S3 que seleccionó en Configuración de políticas y roles de IAM.
4. Elija la región en la que debe crearse el bucket. Por lo general, es la misma región en la que desea crear la secuencia de entrega de Kinesis Data Firehose y la instancia Amazon EC2.
5. Seleccione Create (Crear).

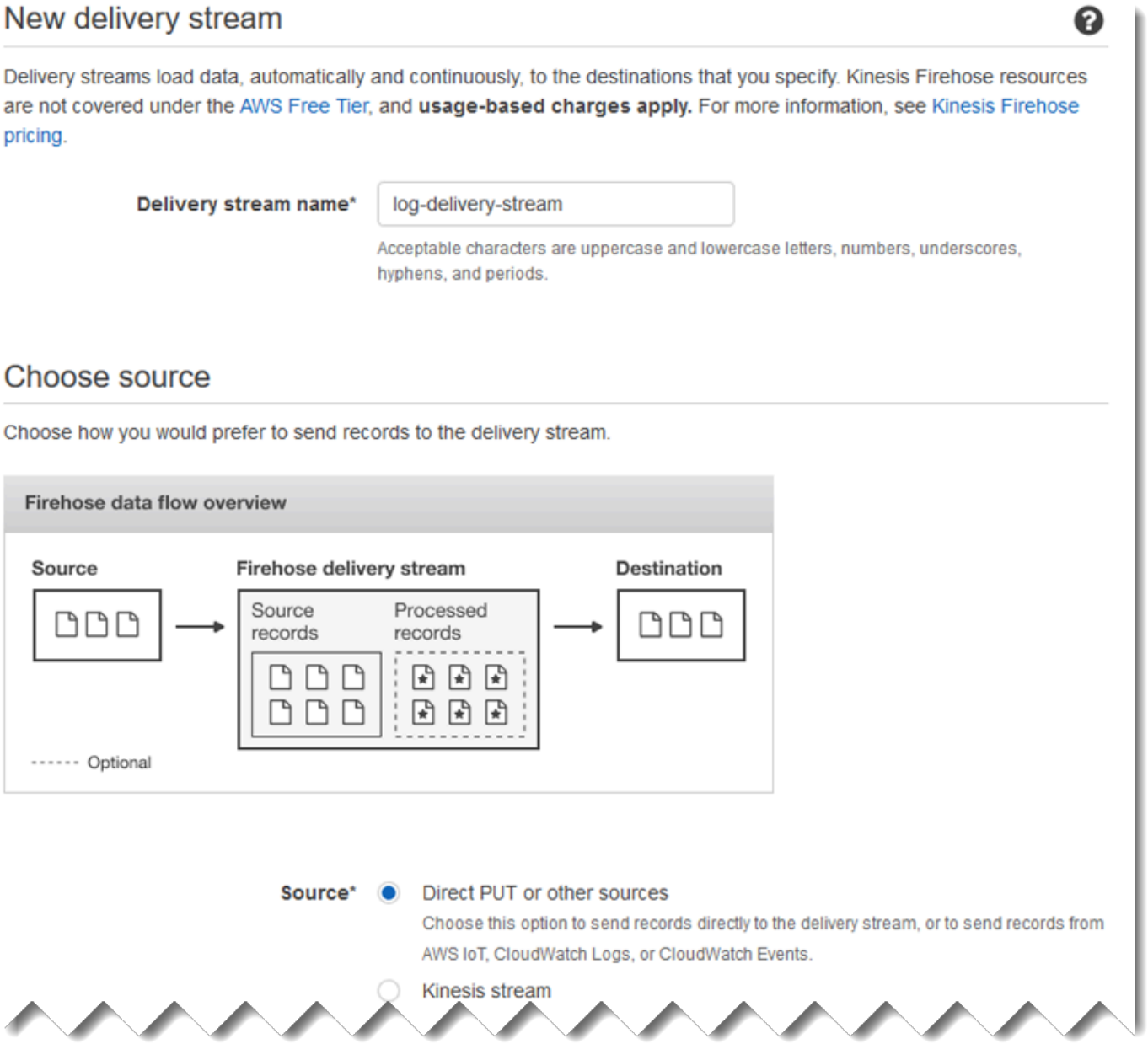
Creación de la secuencia de entrega de Kinesis Data Firehose

Cree la secuencia de entrega de Kinesis Data Firehose que almacenará los registros transmitidos en Amazon S3.

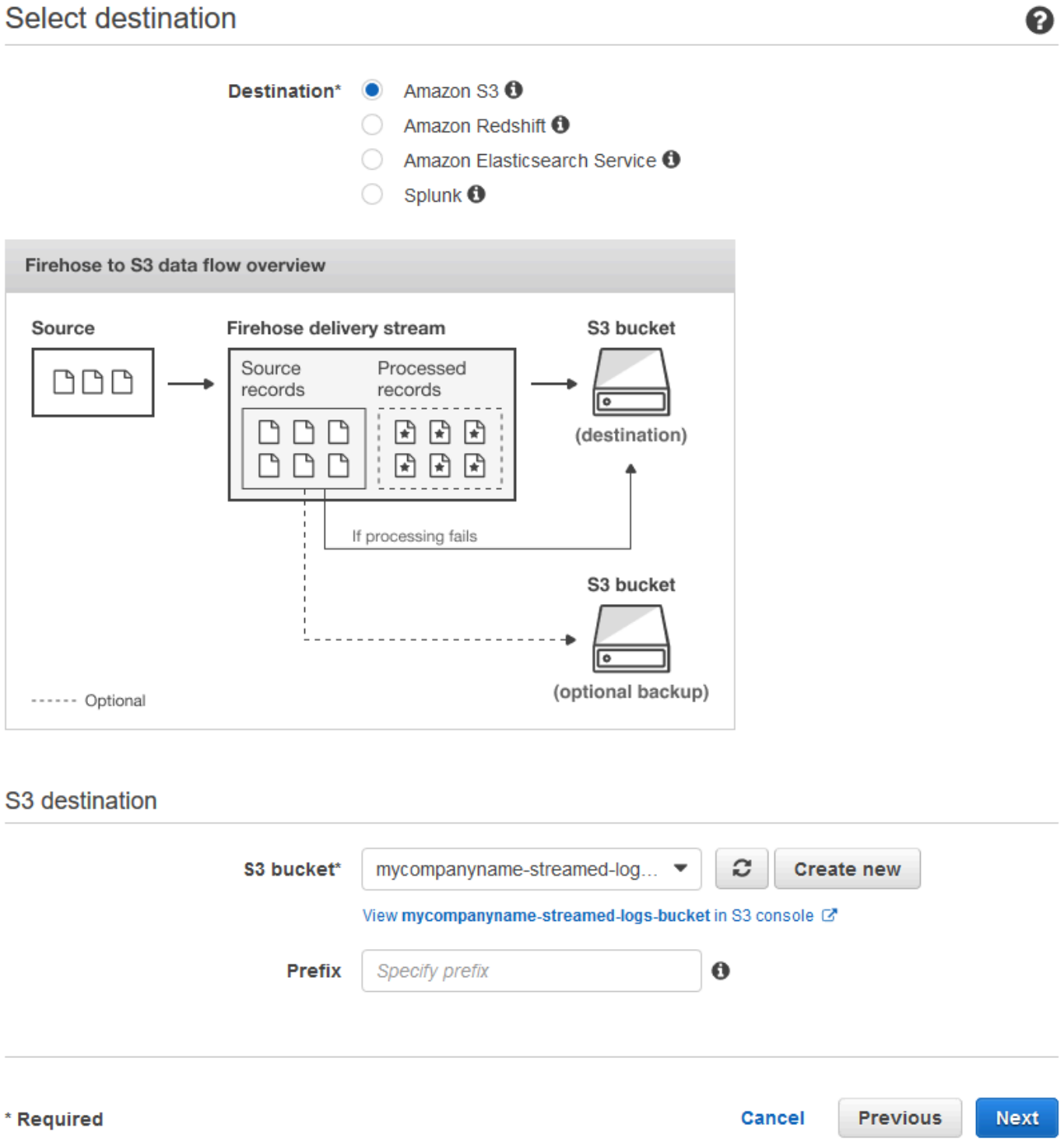
Para crear la secuencia de entrega de Kinesis Data Firehose

1. Abra la consola de Kinesis Data Firehose enhttps://console.aws.amazon.com/firehose/.
2. Elija Create Delivery Stream.

3. En el cuadro Delivery stream name (Nombre de la secuencia de entrega), escriba **`log-delivery-stream`**.
4. En Source (Origen), seleccione Direct PUT or other sources (Direct PUT u otros orígenes).



5. Seleccione Siguiente.
6. Vuelva a seleccionar Siguiente.
7. Para el destino, elijaAmazon S3.
8. En el caso del bucket de S3, seleccione el nombre del bucket que creó en Cree el bucket de Amazon S3.



9. Seleccione Siguiente.
10. En el cuadro Buffer interval (Intervalo de búfer), escriba **`60`**.
11. En IAM role (Rol de IAM), seleccione Create new or choose (Crear nuevo o elegir).
12. En IAM Role (Rol de IAM), seleccione `firehose-s3-access-role`.

13. Elija Allow.

Configure settings

Configure buffer, compression, logging, and IAM role settings for your delivery stream.

S3 buffer conditions

Firehose buffers incoming records before delivering them to your S3 bucket. Record delivery will be triggered once either of these conditions has been satisfied. Learn more

Buffer size* 5 MB

Specify a buffer size between 1-128 MB

Buffer interval* 60 seconds

Specify a buffer interval between 60-900 seconds

S3 compression and encryption

Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using a KMS master key. Learn more

S3 compression* (•) Disabled ( ) GZIP ( ) Snappy ( ) Zip

S3 encryption* (•) Disabled ( ) Enabled

Error logging

Firehose can log record delivery errors to CloudWatch Logs. If enabled, a CloudWatch log group and corresponding log streams are created on your behalf. Learn more

Error logging* ( ) Disabled (•) Enabled

IAM role

Firehose uses an IAM role to access your specified resources, such as the S3 bucket and KMS key. Learn more

IAM role* firehose-s3-access-role

Create new or choose

14. Seleccione Siguiente.
15. Elija Create delivery stream (Crear flujo de entrega).

Crear la instancia de Amazon EC2 para ejecutar Kinesis Agent para Windows

Cree la instancia EC2 que utiliza Kinesis Agent para Windows para transmitir registros de registro a través de Kinesis Data Firehose.

Para crear la instancia EC2

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Siga las instrucciones que se indican en Introducción a las instancias Amazon EC2 de Windows e incorpore estos otros pasos:
   - En el rol de IAM de la instancia, seleccione `kinesis-agent-instance-role`.
   - Si aún no tiene una Virtual Private Cloud (VPC) con una conexión pública a Internet, siga las instrucciones que se indican enConfiguración de con Amazon EC2en laGuía del usuario de instancias de Windows Amazon EC2.
   - Cree o utilice un grupo de seguridad que limite el acceso a la instancia para que dicho acceso solo pueda hacerse desde su equipo o desde los equipos de la organización. Para obtener más información, consulteConfiguración de con Amazon EC2en laGuía del usuario de instancias de Windows Amazon EC2.
   - Si especifica un par de claves existente, asegúrese de tener acceso a la clave privada del par de claves. También puede crear un nuevo par de claves y guardarlo en un lugar seguro.
   - Antes de continuar, espere hasta que la instancia se ejecute y haya completado las dos comprobaciones de estado.
   - La instancia necesita una dirección IP pública. Si no se ha asignado ninguna, siga las instrucciones que se indican enDirecciones IP elásticasen laGuía del usuario de instancias de Windows Amazon EC2.

Pasos siguientes

Paso 2: Instalar, configurar y ejecutar el agente Kinesis para Windows

Paso 2: Instalar, configurar y ejecutar el agente Kinesis para Windows

En este paso, utilizará AWS Management Console para conectarse de forma remota a la instancia que lanzó enCrear la instancia de Amazon EC2 para ejecutar Kinesis Agent para Windows. Luego, instalará Amazon Kinesis Agent para Microsoft Windows en la instancia, creará e implementará el archivo de configuración de Kinesis Agent para Windows e iniciará laAWSKINISTAPservicio de.

1. Conéctese en remoto a la instancia a través del Protocolo de escritorio remoto (RDP) siguiendo las instrucciones que se indican enPaso 2: Conéctese a la instanciaen laGuía del usuario de instancias de Windows Amazon EC2.
2. En la instancia, utilice el Administrador de servidores de Windows para deshabilitar la configuración de seguridad mejorada de Microsoft Internet Explorer para los usuarios y los administradores. Para obtener más información, consulte How To Turn Off Internet Explorer Enhanced Security Configuration en el sitio web de Microsoft TechNet.
3. En la instancia, instale y configure Kinesis Agent para Windows. Para obtener más información, consulte Instalación de Kinesis Agent para Windows.
4. En la instancia, utilice el Bloc de notas para crear un archivo de configuración de Kinesis Agent para Windows. Guarde el archivo en `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`. Añada el siguiente contenido al archivo de configuración:

```
{
  "Sources": [
    {
      "Id": "JsonLogSource",
      "SourceType": "DirectorySource",
      "RecordParser": "SingleLineJson",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "InitialPosition": 0
    }
  ],
  "Sinks": [
    {
      "Id": "FirehoseLogStream",
      "SinkType": "KinesisFirehose",
      "StreamName": "log-delivery-stream",
      "Region": "us-east-1",
      "Format": "json",
```

```
      "ObjectDecoration": "ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd
 HH:mm:ss}"
    }
  ],
  "Pipes": [
    {
      "Id": "JsonLogSourceToFirehoseLogStream",
      "SourceRef": "JsonLogSource",
      "SinkRef": "FirehoseLogStream"
    }
  ]
}
```

Este archivo configura Kinesis Agent para Windows para enviar registros de registro con formato JSON desde los archivos de la`c:\logsource\`(el directorioorigen) a un flujo de entrega de Kinesis Data Firehose llamado`log-delivery-stream`(elsink). Antes de transmitir cada registro a Kinesis Data Firehose, este se mejora con dos pares clave-valor adicionales que contienen el nombre del equipo y una marca temporal.

5. Cree el directorio `c:\LogSource\` y utilice el Bloc de notas para crear un archivo `test.log` en ese directorio con el siguiente contenido:

```
{ "Message": "Copasetic message 1", "Severity": "Information" }
{ "Message": "Copasetic message 2", "Severity": "Information" }
{ "Message": "Problem message 2", "Severity": "Error" }
{ "Message": "Copasetic message 3", "Severity": "Information" }
```

6. En una sesión de PowerShell con permisos elevados, utilice el siguiente comando para iniciar el servicio AWSKinesisTap:

```
Start-Service -ServiceName AWSKinesisTap
```

7. En el Explorador de archivos, navegue al directorio `%PROGRAMDATA%\Amazon \AWSKinesisTap\logs`. Abra el archivo de registro más reciente. La archivo de registro debería tener un aspecto similar al siguiente:

```
2018-09-28 23:51:02.2472 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.AWS.AWSEventSinkFactory.
2018-09-28 23:51:02.2784 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Windows.PerformanceCounterSinkFactory.
2018-09-28 23:51:02.5753 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Core.DirectorySourceFactory.
```

```
2018-09-28 23:51:02.5909 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.ExchangeSource.ExchangeSourceFactory.
2018-09-28 23:51:02.5909 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Uls.UlsSourceFactory.
2018-09-28 23:51:02.5909 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Windows.WindowsSourceFactory.
2018-09-28 23:51:02.9347 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Core.Pipes.PipeFactory.
2018-09-28 23:51:03.5128 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.AutoUpdate.AutoUpdateFactory.
2018-09-28 23:51:03.5440 Amazon.KinesisTap.Hosting.LogManager INFO Performance
 counter sink  started.
2018-09-28 23:51:03.7628 Amazon.KinesisTap.Hosting.LogManager INFO
 KinesisFirehoseSink id FirehoseLogStream for StreamName log-delivery-stream
 started.
2018-09-28 23:51:03.7784 Amazon.KinesisTap.Hosting.LogManager INFO Connected source
 JsonLogSource to sink FirehoseLogStream
2018-09-28 23:51:03.7940 Amazon.KinesisTap.Hosting.LogManager INFO DirectorySource
 id JsonLogSource watching directory C:\LogSource\ with filter *.log started.
```

Este archivo de registro indica que el servicio se ha iniciado y que ahora se están recopilando entradas de registro en el directorio `c:\LogSource\`. Cada línea se analiza como un único objeto JSON. Los pares clave-valor del nombre del equipo y la marca temporal se añaden a cada objeto. Luego se transmite a Kinesis Data Firehose.

8. En un minuto o dos, vaya al bucket de Amazon S3 que creó enCree el bucket de Amazon S3Uso de la consola de administración de AWS. Compruebe que ha elegido la región correcta en la consola.

   En ese bucket, hay una carpeta con el año actual. Ábrala y encontrará una carpeta con el mes actual. Si la abre, encontrará una carpeta con el día de hoy. Ábrala y verá una carpeta con la hora actual (UTC). Si la abre, verá uno o varios elementos que empiezan por el nombre `log-delivery-stream`.

9. Abra el contenido del elemento más reciente para confirmar que los registros de registro se han almacenado correctamente en Amazon S3 con las mejoras deseadas. Si todo está configurado correctamente, el contenido será similar al siguiente:

```
{"Message":"Copasetic message 1","Severity":"Information","ComputerName":"EC2AMAZ-
ABCDEFGH","DT":"2018-09-28 23:51:04"}
{"Message":"Copasetic message 2","Severity":"Information","ComputerName":"EC2AMAZ-
ABCDEFGH","DT":"2018-09-28 23:51:04"}
{"Message":"Problem message 2","Severity":"Error","ComputerName":"EC2AMAZ-
ABCDEFGH","DT":"2018-09-28 23:51:04"}
{"Message":"Copasetic message 3","Severity":"Information","ComputerName":"EC2AMAZ-
ABCDEFGH","DT":"2018-09-28 23:51:04"}
```

10. Para obtener más información acerca de cómo resolver cualquiera de los problemas siguientes, consulte Solución de problemas de Amazon Kinesis Agent para Microsoft Windows:

    - El archivo de registro de Kinesis Agent para Windows contiene errores.
    - Las carpetas o elementos que se esperaban en Amazon S3 no existen.
    - El contenido de un artículo de Amazon S3 no es correcto.

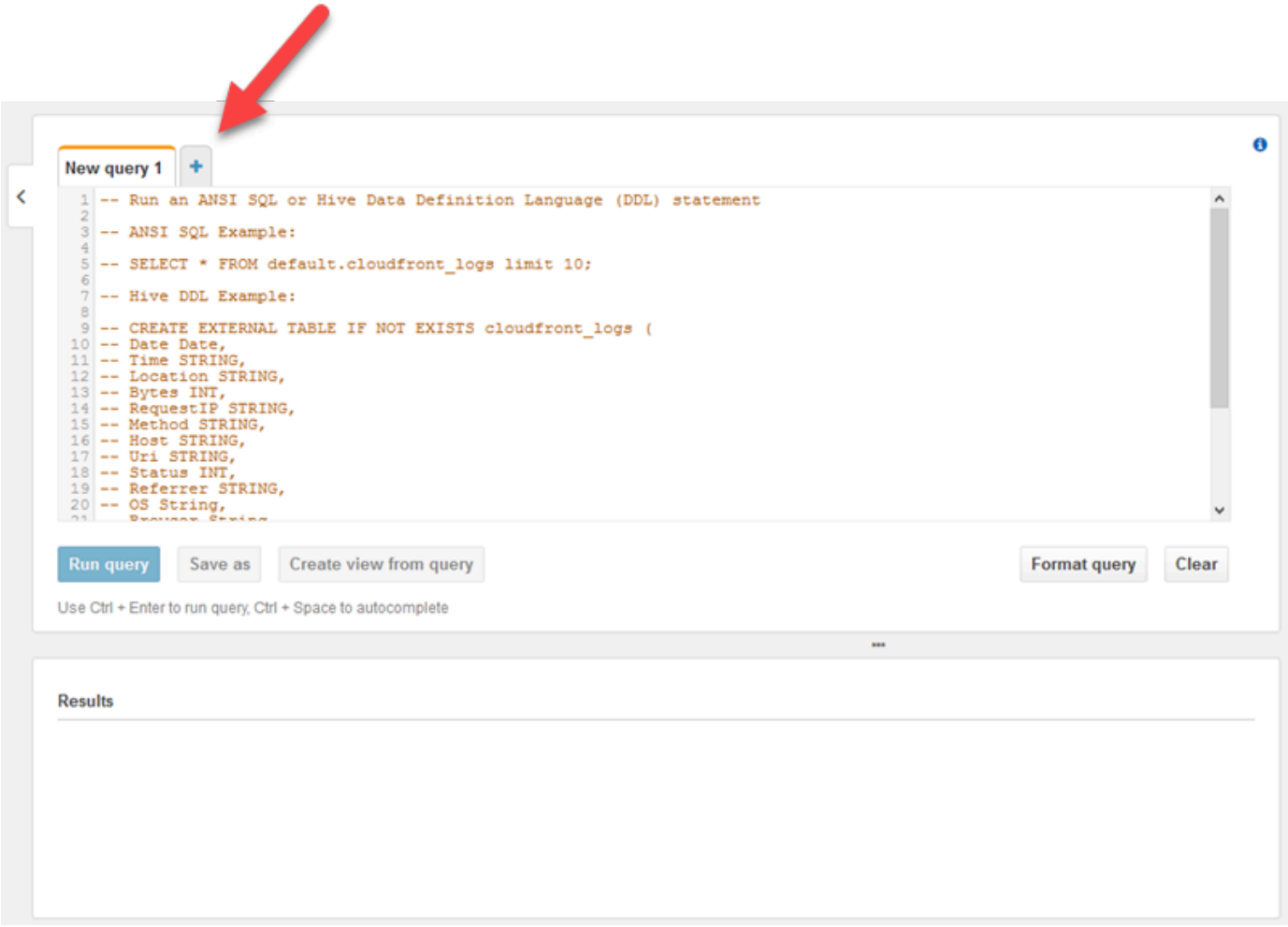
Pasos siguientes

Paso 3: Consulta de los datos de registro en Amazon S3

Paso 3: Consulta de los datos de registro en Amazon S3

En el paso final de este agente de Amazon Kinesis para Microsoft WindowsTutorial dePuede utilizar Amazon Athena para consultar los datos de registro almacenados en Amazon Simple Storage Service (Amazon S3).

1. Abra la consola de Athena en https://console.aws.amazon.com/athena/.
2. Elija el signo más (+) en la ventana de consulta de Athena para crear una nueva ventana de consulta.



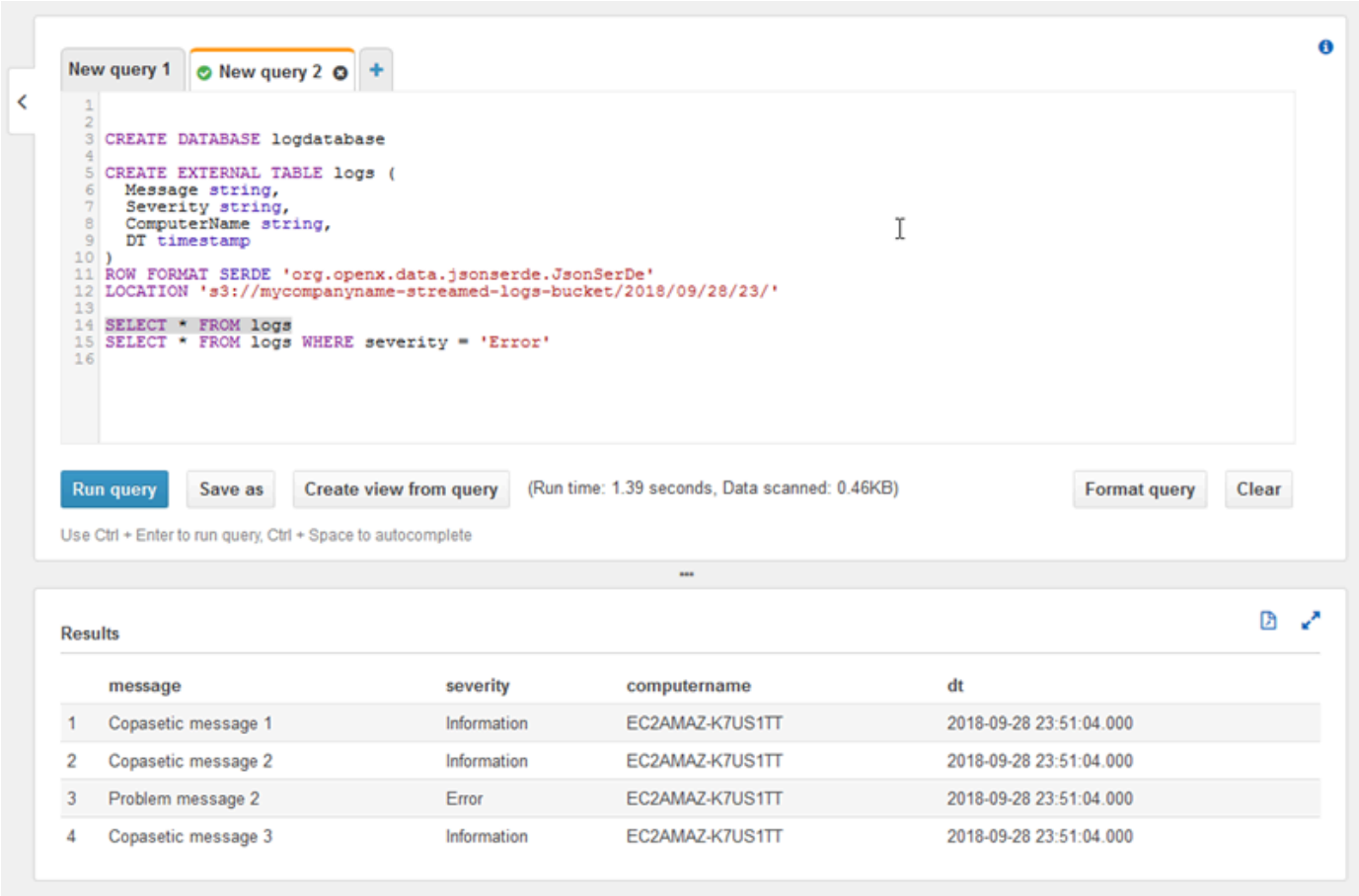
3. Escriba el siguiente texto en la ventana de consulta:

```
CREATE DATABASE logdatabase

CREATE EXTERNAL TABLE logs (
  Message string,
  Severity string,
```

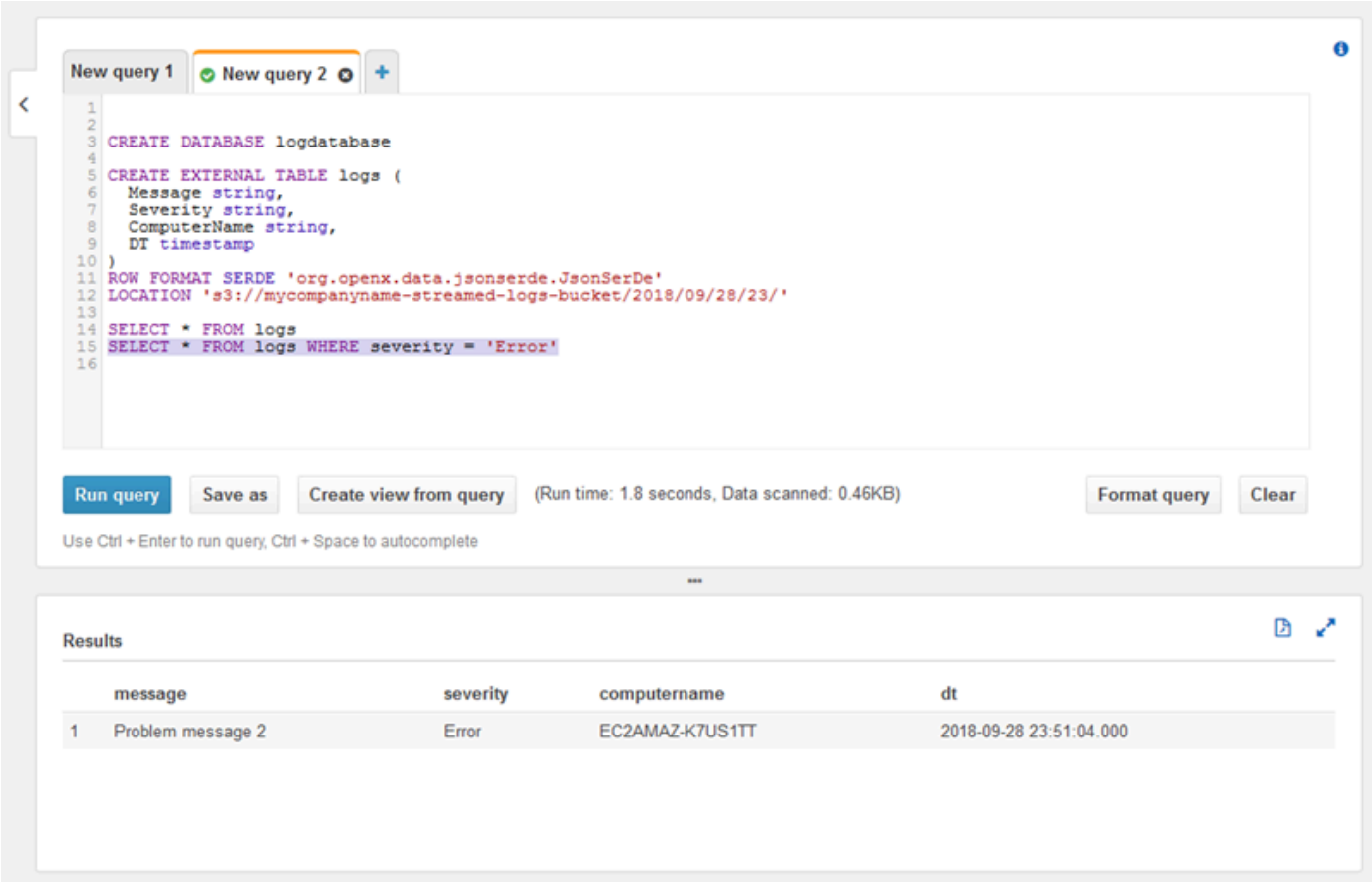
```
  ComputerName string,
  DT timestamp
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://bucket/year/month/day/hour/'

SELECT * FROM logs
SELECT * FROM logs WHERE severity = 'Error'
```

Sustituya *bucket* por el nombre del bucket que creó en Cree el bucket de Amazon S3. Reemplazar*year*,*month*,*day*y*hour*Con el año, el mes, el día y la hora en que se creó el archivo de registro de Amazon S3 en UTC.

4. Seleccione el texto de la instrucción `CREATE DATABASE` y haga clic en Run query (Ejecutar consulta). Esto crea la base de datos de registros en Athena.
5. Seleccione el texto de la instrucción `CREATE EXTERNAL TABLE` y haga clic en Run query (Ejecutar consulta). Esto crea una tabla Athena que hace referencia al bucket de S3 con los datos de registro, donde el esquema de JSON se asigna al esquema de la tabla Athena.
6. Seleccione el texto de la primera instrucción `SELECT` y haga clic en Run query (Ejecutar consulta). Esto mostrará todas las filas de la tabla.



7. Seleccione el texto de la segunda instrucción `SELECT` y haga clic en Run query (Ejecutar consulta). Esto solamente muestra las filas de la tabla que representan entradas de registro con una gravedad de nivel `Error`. Este tipo de consulta busca entradas de registro interesantes a partir de un número de registros que puede llegar a ser muy grande.



Pasos siguientes

Utilice la consola de administración de AWS para eliminar los recursos que se han creado durante este tutorial:

1. Termine la instancia EC2 (consulte el paso 3 de Introducción a las instancias Amazon EC2 de Windows).

   > **Important**
   >
   > Si ha iniciado una instancia que no estaba dentro de lacapa gratuita de AWSSe le cobrará por la instancia hasta que la termine.

2. Elimine la secuencia de entrega de Kinesis Data Firehose.

   a. Abra la consola de Kinesis Data Firehose enhttps://console.aws.amazon.com/firehose/.

   b. Seleccione la secuencia de entrega que creó.

c. Elija Eliminar.

d. Seleccione Delete delivery stream (Eliminar secuencia de entrega).

3. Elimine el bucket de S3. Para obtener instrucciones, consulte¿Cómo elimino un bucket de S3?en laGuía del usuario de Amazon Simple Storage Service Console.

Para obtener más información, consulte los siguientes temas:

- Configuración de Amazon Kinesis Agent para Microsoft Windows
- ¿Qué es Amazon Kinesis Data Firehose?
- ¿Qué es Amazon S3?
- ¿Qué es Amazon Athena?

Solución de problemas de Amazon Kinesis Agent para Microsoft Windows

Utilice las siguientes instrucciones para diagnosticar y corregir problemas relacionados con el agente de Amazon Kinesis Agent para Microsoft Windows.

Temas

No se transmiten datos desde los servidores o equipos de escritorio a los servicios de AWS esperados

Symptoms

Cuando examina los registros, eventos y métricas alojados en diversos servicios de AWS que están configuradas para recibir secuencias de datos de Kinesis Agent para Windows, Kinesis Agent para Windows no transmite nada.

Causes

Existen varias causas posibles para este problema:

- Hay un origen, receptor o canalización que no está configurado de forma correcta.
- La autenticación de Kinesis Agent para Windows no está configurada correctamente.
- La autorización de Kinesis Agent para Windows no está configurada correctamente.
- Existe un error en una expresión regular incluida en una declaración `DirectorySource`.
- Se ha especificado un directorio que no existe en una declaración `DirectorySource`.

- Se han proporcionado valores no válidos a los servicios de AWS, que luego han rechazado las solicitudes de Kinesis Agent para Windows.
- Un receptor hace referencia a un recurso que no existe en la región de AWS especificada o implícita.
- Se ha especificado una consulta no válida en una declaración `WindowsEventLogSource`.
- Se ha especificado un valor que no es válido en el par clave-valor `InitialPosition` de un origen.
- El archivo de configuración `appsettings.json` no se ajusta al esquema JSON de dicho archivo.
- Los datos se transmiten a una región distinta de la que se seleccionó en AWS Management Console.
- Kinesis Agent para Windows no está instalado correctamente o no se está ejecutando.

Resolutions

Para solucionar problemas relacionados con la transmisión de datos, siga estos pasos:

1. Examine los registros de Kinesis Agent para Windows en el`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`Directorio. Busque la cadena `ERROR`.
   a. Si un origen o receptor no se carga, haga lo siguiente:
      i. Examine el mensaje de error y busque el `Id` del origen o receptor.
      ii. Compruebe la declaración del origen o receptor que se corresponde con ese `Id` en el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` para ver los errores relacionados con el mensaje recibido. Para obtener más información, consulte Configuración de Amazon Kinesis Agent para Microsoft Windows.
      iii. Corrija todos los problemas del archivo de configuración relacionados con el error.
      iv. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.
   b. Si el mensaje de error indica que no se encontró un objeto `SourceRef` o `SinkRef` en una canalización, siga estos pasos:
      i. Anote el valor de `Id` de la canalización.
      ii. Examine la declaración de la canalización en el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` que se corresponde con el valor de `Id` anotado. Asegúrese de que, en los valores de los pares clave-valor `SourceRef` y `SinkRef`,

los `Id` de las declaraciones del origen y del receptor a los que desea hacer referencia están bien escritos. Corrija los errores tipográficos u ortográficos. Si falta una declaración del origen o del receptor en el archivo de configuración, añádala. Para obtener más información, consulte Configuración de Amazon Kinesis Agent para Microsoft Windows.

iii. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

c. Si el mensaje de error indica que un determinado usuario o rol de IAM de no está autorizado para realizar ciertas operaciones, siga estos pasos:

i. Asegúrese de que Kinesis Agent para Windows esté utilizando el usuario o rol de IAM correcto. Si no lo es, reviseConfiguración de seguridad de los receptoresy ajuste la forma en que Kinesis Agent para Windows se autentica para garantizar que se utilice el usuario o rol de IAM correcto.

ii. Si se está utilizando el rol o usuario de IAM de correcto, examine a través de la Consola de administración de AWS las políticas que están asociadas a dicho usuario o rol. Asegúrese de que el usuario o el rol tienen todos los permisos mencionados en el mensaje de error en todos los recursos de AWS a los que tiene acceso Kinesis Agent para Windows. Para obtener más información, consulte Configuración de la autorización .

iii. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de seguridad se han resuelto.

d. Si el mensaje de error indica que hay un error en los argumentos al analizar una expresión regular que se encuentra en el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`, siga estos pasos:

i. Examine la expresión regular en el archivo de configuración.

ii. Verifique la sintaxis de la expresión regular. Existen varios sitios web que puede utilizar para verificar expresiones regulares. También puede utilizar las siguientes líneas de comandos para comprobar las expresiones regulares de una declaración de origen `DirectorySource`:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
ktdiag.exe /r sourceId
```

Sustituya *`sourceId`* por el valor del par clave-valor `Id` de la declaración de origen `DirectorySource` que tiene la expresión regular incorrecta.

iii. Realice las correcciones necesarias en la expresión regular del archivo de configuración.

iv. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

e. Si el mensaje de error indica que hay un error en los argumentos al analizar una expresión regular que no se encuentra en el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` y que está relacionada con un determinado receptor, siga estos pasos:

i. Localice la declaración del receptor en el archivo de configuración.

ii. Verifique que los pares clave-valor que están específicamente relacionados con un servicio de AWS utilizan nombres que se ajustan a las reglas de validación de dicho servicio. Por ejemplo, los nombres de grupo de CloudWatch Logs de solo pueden contener un determinado conjunto de caracteres que se especifica utilizando la expresión regular`[\.\-_/#A-Za-z0-9]+`.

iii. Corrija los nombres que no sean válidos en los pares clave-valor de la declaración del receptor y asegúrese de que esos recursos estén configurados correctamente en AWS.

iv. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

f. Si el mensaje de error indica que un origen o receptor no se puede cargar porque hay un parámetro que falta o es nulo, siga estos pasos:

i. Anote el `Id` del origen o receptor.

ii. Localice la declaración del origen o del receptor que coincida con el `Id` que anotó en el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

iii. Revise los pares clave-valor que se proporcionan en la declaración del origen o receptor y compárelos con los requisitos del tipo de receptor u origen que se indican en la documentación de Configuración de Amazon Kinesis Agent para Microsoft Windows del tipo de receptor correspondiente. Añada los pares clave-valor que falten en la declaración del origen o receptor.

iv. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

g. Si el mensaje de error indica que hay un nombre de directorio que no es válido, siga estos pasos:

i. Localice el nombre de directorio incorrecto en el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

ii. Compruebe que este directorio existe y que contiene los archivos de registro que deben transmitirse.

iii. Corrija los errores que haya en el nombre de directorio especificado en el archivo de configuración.

iv. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

h. Si el mensaje de error indica que hay un recurso que no existe:

i. Localice la referencia del recurso que no existe en una declaración del receptor del archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

ii. Utilice la consola de administración de AWS para localizar el recurso en la región de AWS correcta que debe utilizarse en la declaración del receptor. Compárelo con el que se especificó en el archivo de configuración.

iii. Cambie la declaración del receptor en el archivo de configuración para que tenga la región y el nombre de recurso apropiados.

iv. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

i. Si el mensaje de error indica que una consulta no es válida en un determinado `WindowsEventLogSource`, siga estos pasos:

i. En el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap \appsettings.json`, localice la declaración `WindowsEventLogSource` que tiene el mismo `Id` que aparece en el mensaje de error.

ii. Verifique que el valor del par `Query` de la declaración del origen se ajusta a las especificaciones de Event queries and Event XML.

iii. Realice los cambios necesarios en la consulta para que se ajuste a estas especificaciones.

iv. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

j. Si el mensaje de error indica que hay una posición inicial que no es válida, siga estos pasos:

i. En el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap \appsettings.json`, localice la declaración del origen que tiene el mismo `Id` que aparece en el mensaje de error.

ii. Cambie el valor del par clave-valor `InitialPosition` en la declaración del origen para que se ajuste a los valores permitidos, tal y como se describe en Configuración de Bookmark.

iii. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

2. Asegúrese de que el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` se ajusta al esquema JSON.

   a. En una ventana de símbolo del sistema, invoque las líneas siguientes:

   ```
   cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
   %PROGRAMFILES%\Amazon\AWSKinesisTap\ktdiag.exe /c
   ```

   b. Corrija los problemas que detecte con el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

   c. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para verificar que los problemas de configuración se han resuelto.

3. Cambie el nivel de registro para intentar obtener información más detallada.

   a. Sustituya el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\nlog.xml` por el contenido siguiente:

   ```
   <?xml version="1.0" encoding="utf-8" ?>
   <nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.nlog-project.org/schemas/NLog.xsd NLog.xsd"
         autoReload="true"
         throwExceptions="false"
         internalLogLevel="Off" internalLogFile="c:\temp\nlog-internal.log" >

     <!--
     See https://github.com/nlog/nlog/wiki/Configuration-file
     for information on customizing logging rules and outputs.
      -->
     <targets>
       <!--
       add your targets here
       See https://github.com/nlog/NLog/wiki/Targets for possible targets.
       See https://github.com/nlog/NLog/wiki/Layout-Renderers for the possible layout
    renderers.
       -->

       <target name="logfile"
               xsi:type="File"
               layout="${longdate} ${logger} ${uppercase:${level}} ${message}"
   ```

```
            fileName="${specialfolder:folder=CommonApplicationData}/Amazon/
KinesisTap/logs/KinesisTap.log"
     maxArchiveFiles="90"
     archiveFileName="${specialfolder:folder=CommonApplicationData}/Amazon/
KinesisTap/logs/Archive-{################}.log"
     archiveNumbering="Date"
     archiveDateFormat="yyyy-MM-dd"
     archiveEvery="Day"
     />
  </targets>

  <rules>
    <logger name="*" minlevel="Debug" writeTo="logfile" />
  </rules>
</nlog>
```

b. Detenga e inicie el servicio `AWSKinesisTap`. A continuación, compruebe el archivo de registro más reciente para ver si hay otros mensajes en el registro que pudieran ayudar a diagnosticar y resolver el problema.

4. Compruebe que está examinando los recursos de la región correcta de la consola de administración de AWS.
5. Compruebe que el agente Kinesis Agent para Windows está instalado y en ejecución.
   a. En Windows, seleccione Iniciar y vaya al Panel de control, Herramientas administrativas y Servicios.
   b. Busque el servicio AWSKinesisTap.
   c. Si el servicio AWSKinesisTap no está visible, instale Kinesis Agent para Windows siguiendo las instrucciones que se indican en Introducción a Amazon Kinesis Agent para Microsoft Windows.
   d. Si el servicio está visible, determine si está en ejecución. Si no lo está, abra el menú contextual del servicio (haga clic con el botón derecho) y seleccione Iniciar.
   e. Compruebe que el servicio se ha iniciado; para ello, consulte el archivo de registro más reciente del directorio `%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`.

Se aplica a

Esta información es aplicable a Kinesis Agent para versión 1.0.115 y siguientes.

A veces faltan datos esperados

Symptoms

La mayor parte del tiempo, Kinesis Agent para Windows transmite los datos, pero a veces falta alguno.

Causes

Existen varias causas posibles para este problema:

- No se está utilizando la característica de marcadores (bookmarks).
- Se han sobrepasado los límites de velocidad de datos en los servicios de AWS de acuerdo con la configuración actual de dichos servicios.
- Se han sobrepasado los límites de velocidad de las llamadas a la API en los servicios de AWS en,`appsettings.json`y los límites de la cuenta de AWS.

Resolutions

Para solucionar problemas relacionados con la ausencia de datos, siga estos pasos:

1. Considere la posibilidad de utilizar la característica de marcadores que se describe en Configuración de Bookmark. Le ayudará a garantizar que se envían todos los datos, incluso cuando Kinesis Agent para Windows se detiene y vuelve a iniciarse.
2. Utilice las métricas integradas en Kinesis Agent para Windows para detectar problemas:
   a. Habilite la transmisión de métricas de Kinesis Agent para Windows, tal y como se describe enConfiguración del agente Kinesis para tuberías métricas de Windows.
   b. Si hay un número importante de errores no recuperables en uno o varios receptores, determine cuántos bytes o registros se envían por segundo. A continuación, determine si este valor está dentro de los límites configurados para esos servicios de AWS en la región y la cuenta en las que se están transmitiendo los datos.
   c. Cuando se superen los límites, reduzca la velocidad o la cantidad de datos enviados, solicite un aumento de los límites e incremente la fragmentación, si procede.
   d. Después de realizar estos ajustes, continúe monitorizando las métricas integradas en Kinesis Agent para Windows en para asegurarse de que se ha resuelto la situación.

Para obtener más información sobre los límites de Kinesis Data Streams, consulteLímites de Kinesis Data Streamsen laGuía para desarrolladores de Kinesis Data Streams. Para obtener más información sobre los límites de Kinesis Data Firehose, consulteLímites de Amazon Kinesis Data Firehose.

Se aplica a

Esta información es aplicable a Kinesis Agent para versión 1.0.115 y siguientes.

Los datos llegan con un formato incorrecto

Symptoms

Los datos que llegan al servicio de AWS tienen un formato incorrecto.

Causes

Existen varias causas posibles para este problema:

- El valor del par clave-valor `Format` de la declaración del receptor en el archivo de configuración `appsettings.json` no es correcto.
- El valor del par clave-valor `RecordParser` de una declaración `DirectorySource` no es correcto.
- Las expresiones regulares de una declaración `DirectorySource` que utiliza el analizador de registros `Regex` no son correctas.

Resolutions

Para solucionar problemas relacionados con un formato incorrecto, siga estos pasos:

1. Revise las declaraciones de receptores en el archivo de configuración `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.
2. Asegúrese de que el valor correcto del par clave-valor `Format` está especificado en cada declaración del receptor. Para obtener más información, consulte Declaraciones de receptores.
3. Si hay orígenes con declaraciones `DirectorySource` conectados mediante canalizaciones a receptores que especifican valores `xml` o `json` en el par `Format`, asegúrese de que estos orígenes especifican uno de los siguientes valores en el par `RecordParser`:

- `SingleLineJson`
- `Regex`
- `SysLog`
- `Delimited`

Otros analizadores de registros solo utilizan texto y no funcionan correctamente con receptores que requieren un formato XML o JSON.

4. Si el tipo de origen `DirectorySource` no analiza correctamente los registros, invoque las siguientes líneas en una ventana del símbolo del sistema para verificar los pares clave-valor de la expresión regular y la marca temporal especificados en la declaración `DirectorySource`:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
ktdiag.exe /r sourceID
```

Sustituya *`sourceID`* por el valor del par clave-valor `Id` de la declaración de origen `DirectorySource` que parece no funcionar correctamente. Corrija los problemas notificados por `ktdiag.exe`.

Se aplica a

Esta información es aplicable a Kinesis Agent para versión 1.0.115 y siguientes.

Problemas de rendimiento

Symptoms

Después de instalar e iniciar Kinesis Agent para Windows, las aplicaciones y los servicios han aumentado las latencias.

Causes

Existen varias causas posibles para este problema:

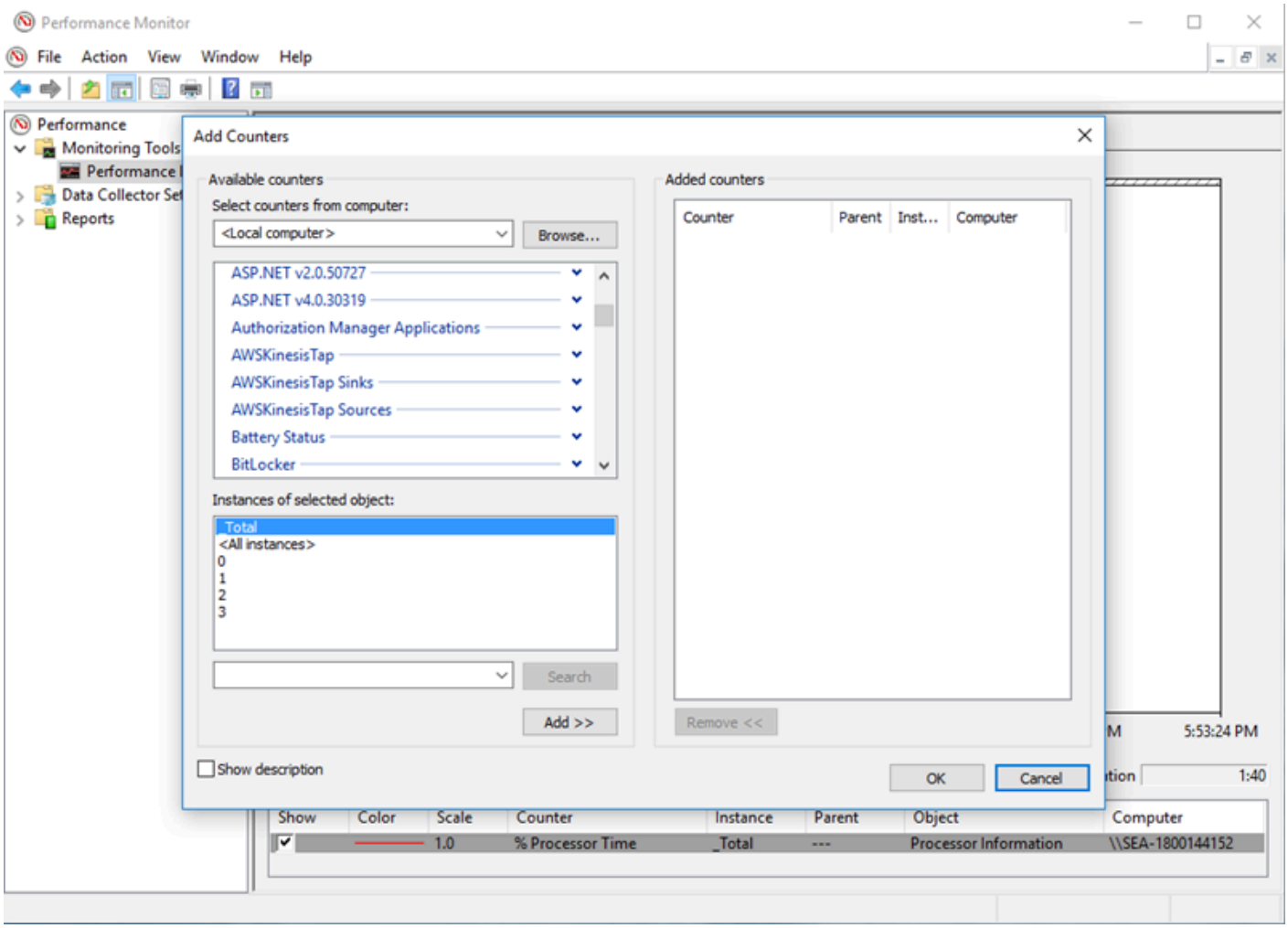
- La máquina en la que se ejecuta Kinesis Agent para Windows no tiene capacidad suficiente para transmitir la cantidad de datos deseada.
- Se están transmitiendo datos innecesarios a uno o varios servicios de AWS.

- Kinesis Agent para Windows transmite datos a servicios de AWS que no están configurados para una velocidad de datos tan elevada.
- Kinesis Agent para Windows está invocando operaciones en servicios de AWS en una cuenta en la que el límite de velocidad de las llamadas a la API es muy bajo.

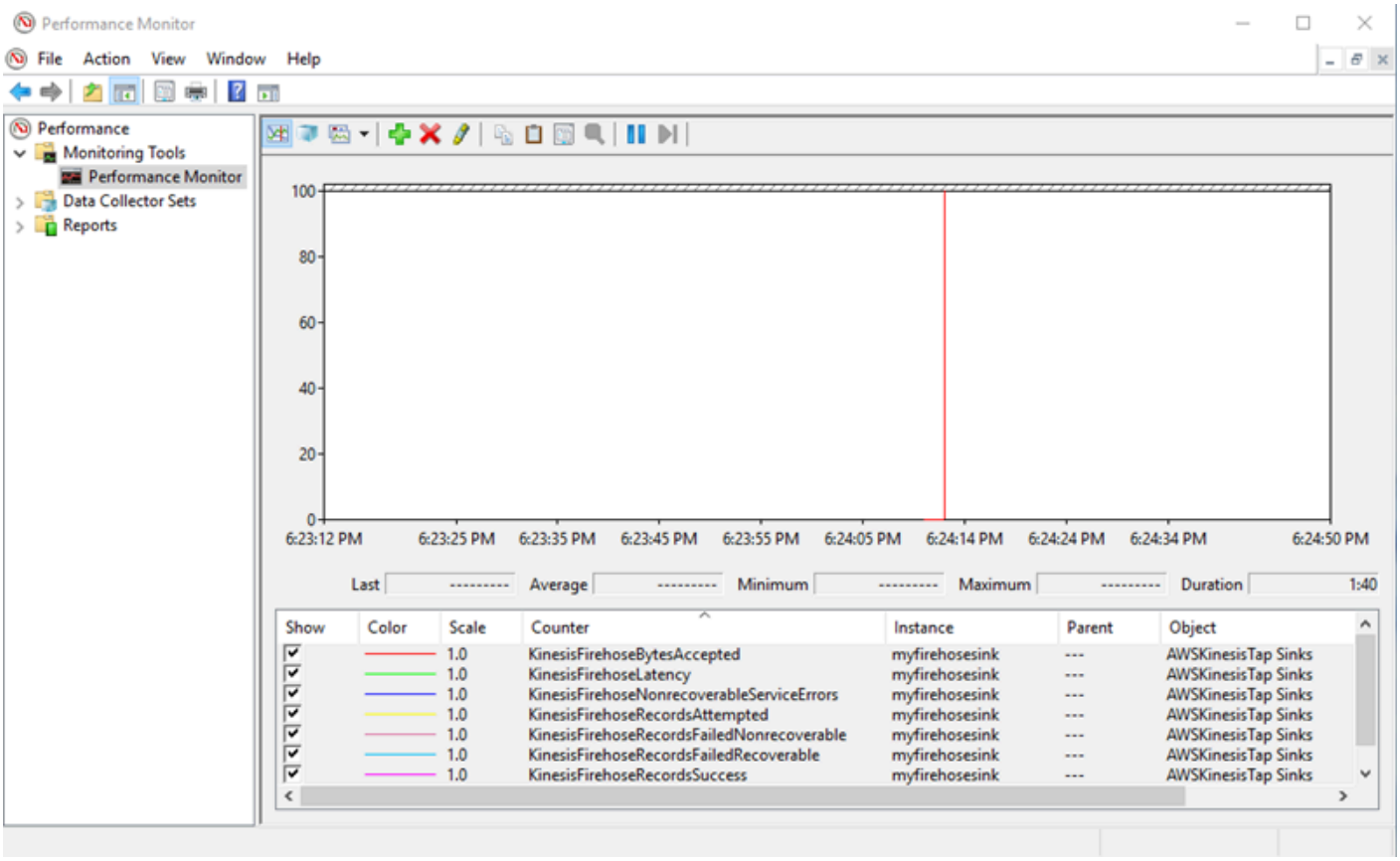
Resolutions

Para solucionar problemas relacionados con el rendimiento, siga estos pasos:

1. Utilice la aplicación Monitor de recursos de Windows para comprobar la memoria, la CPU, el disco y el uso de la red. Si necesita transmitir grandes cantidades de datos con Kinesis Agent para Windows, es posible que tenga que aprovisionar una máquina con mayor capacidad en algunas de estas áreas, en función de la configuración.
2. También podría reducir la cantidad de datos que se registran utilizando filtros:
   - Consulte el par clave-valor `Query` en Configuración de WindowsEventLogSource.
   - Consulte el filtrado de canalizaciones en Configuración de canalizaciones.
   - Consulte el filtrado de métricas de Amazon CloudWatch enConfiguración de receptores de CloudWatch).
3. Utilice la aplicación Monitor de rendimiento de Windows para ver las métricas de Kinesis Agent para Windows de o transmitir estas métricas a CloudWatch (consulteOrigen de métricas integrado en Windows en). En la aplicación Monitor de rendimiento de Windows, puede agregar contadores para los receptores y orígenes de Kinesis Agent para Windows. Estos contadores están organizados en las categorías AWSKinesisTap Sinks y AWSKinesisTap Sources.



Por ejemplo, para diagnosticar problemas de rendimiento de Kinesis Data Firehose, añada la opciónKinesis FirehoseContadores de rendimiento.



Si hay un gran número de errores no recuperables, examine los últimos registros de Kinesis Agent para Windows en la`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`Directorio. Si se produce una limitación controlada en los receptores `KinesisStream` o `KinesisFirehose`, siga estos pasos:

- Si se produce una limitación controlada debido a que los datos se transmiten demasiado rápido, considere la posibilidad de aumentar el número de particiones en la secuencia de datos de Kinesis. Para obtener más información, consulteCambio en los fragmentos, escalado y procesamiento paraleloen laGuía para desarrolladores de Kinesis Data Streams.
- Considere la posibilidad de aumentar el límite de llamadas a la API en Kinesis Data Streams en o el tamaño del búfer del receptor si se produce una limitación limitada en las llamadas a la API. Para obtener más información, consulteLímites de Kinesis Data Streamsen laGuía para desarrolladores de Kinesis Data Streams.
- Si los datos se transmiten demasiado rápido, considere la posibilidad de solicitar un aumento de los límites de velocidad para la secuencia de entrega de Kinesis Data Firehose. O bien, si las llamadas a la API están sufriendo limitaciones controladas, puede solicitar un aumento del límite de llamadas a la API (consulte Límites de Amazon Kinesis Data Firehose Limits) o del tamaño del búfer del receptor.

- Después de aumentar el número de particiones de una secuencia de Kinesis Data Streams de o el límite de velocidad de una secuencia de entrega de Kinesis Data Firehose, revise el agente de Kinesis para Windows`appsettings.json`Archivo de configuración de para aumentar los registros por segundo o los bytes por segundo del receptor. De lo contrario, Kinesis Agent para Windows no podrá sacar provecho del aumento de los límites.

Se aplica a

Esta información es aplicable a Kinesis Agent para versión 1.0.115 y siguientes.

Espacio en disco agotado

Symptoms

Kinesis Agent para Windows se está ejecutando en un equipo que tiene muy poco espacio en una o varias unidades de disco.

Causes

Existen varias causas posibles para este problema:

- El archivo de configuración de registro de Kinesis Agent para Windows no es correcto.
- La cola persistente de Kinesis Agent para Windows no está configurada correctamente.
- Otra aplicación o servicio está consumiendo espacio en disco.

Resolutions

Para solucionar problemas relacionados con el espacio en disco, siga estos pasos:

- Si no hay mucho espacio en el disco que contiene los archivos de registro del agente de Kinesis para Windows, examine el directorio del archivo de registro (normalmente, es)`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`). Asegúrese de que tanto el número como el tamaño de los archivos de registro que se están guardando es razonable. Puede controlar la ubicación, la conservación y el nivel de detalle de los registros de Kinesis Agent para Windows al editar el archivo`%PROGRAMFILES%\Amazon\AWSKinesisTap\Nlog.xml`Archivo de configuración de.
- Si la característica de creación de colas del receptor está habilitada, examine las declaraciones de receptores que utilicen esta característica. Asegúrese de que el par clave-valor `QueuePath` hace

referencia a una unidad de disco con espacio suficiente para tener el número máximo de lotes especificado mediante el par clave-valor `QueueMaxBatches`. Si no es posible, reduzca el valor del par clave-valor `QueueMaxBatches` para que los datos se adapten fácilmente al espacio restante de la unidad de disco especificada.

- Utilice el explorador de archivos de Windows para localizar los archivos que consumen espacio en disco y transferir o eliminar el exceso de archivos. Cambie la configuración de la aplicación o servicio que consume grandes cantidades de espacio en disco.

Se aplica a

Esta información es aplicable a Kinesis Agent para versión 1.0.115 y siguientes.

Herramientas para solucionar problemas

Además de verificar el archivo de configuración de, también puede usar la herramienta`ktdiag.exe`, que proporciona algunas otras funcionalidades para diagnosticar y resolver problemas cuando se configura y utiliza Kinesis Agent para Windows. La herramienta `ktdiag.exe` se encuentra en el directorio `%PROGRAMFILES%\Amazon\AWSKinesisTap`.

- Si cree que hay archivos de registro con un determinado patrón que se están escribiendo en un directorio pero que no los está procesando Kinesis Agent para Windows, utilice la herramienta`/w`para comprobar que se están detectando estos cambios. Por ejemplo, supongamos que espera que los archivos de registro con el patrón `*.log` se escriban en el directorio `c:\foo`. Puede utilizar el conmutador `/w` al ejecutar la herramienta `ktdiag.exe` y especificar el directorio y el patrón del archivo:

  ```
  cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
  ktdiag /w c:\foo *.log
  ```

  Si los archivos de registro se están escribiendo, verá una salida similar a la siguiente:

  ```
  Type any key to exit this program...
  File: c:\foo\log1.log ChangeType: Created
  File: c:\foo\log1.log ChangeType: Deleted
  File: c:\foo\log1.log ChangeType: Created
  File: c:\foo\log1.log ChangeType: Changed
  File: c:\foo\log1.log ChangeType: Changed
  File: c:\foo\log1.log ChangeType: Changed
  ```

```
File: c:\foo\log1.log ChangeType: Changed
```

Si no se produce este tipo de salida, significa que la aplicación o el servicio tienen algún problema para escribir los registros o que hay un problema de seguridad en lugar de un problema con Kinesis Agent para Windows. Si se produce este tipo de salida pero el agente Kinesis Agent para Windows sigue sin procesar los registros de, consulteNo se transmiten datos desde los servidores o equipos de escritorio a los servicios de AWS esperados.

- En ocasiones, los registros solo se escriben de vez en cuando, pero convendría verificar que el agente Kinesis para Windows está funcionando correctamente. Utilice el conmutador `/log4net` para simular que una aplicación está escribiendo registros con la biblioteca `Log4net`; por ejemplo:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
KTDiag.exe /log4net c:\foo\log2.log
```

De este modo, se escribe un archivo de registro de tipo `Log4net` en el archivo de registro `c:\foo\log2.log` y se guardan las nuevas entradas hasta que se presiona una tecla. Puede configurar diversas opciones utilizando otros conmutadores que pueden especificarse detrás del nombre de archivo:

Bloqueo `-lm`, `-li` o `-le`

Puede especificar uno de los siguientes conmutadores de bloqueo para controlar el modo en que se bloquea el archivo de registro:

`-lm`

En el archivo de registro, se utiliza la cantidad mínima de bloqueo, lo que permite el máximo acceso al archivo de registro.

`-li`

Solo los subprocesos que forman parte del mismo proceso pueden obtener acceso al registro al mismo tiempo.

`-le`

Solo puede obtener acceso al registro un subproceso cada vez. Esta es la opción predeterminada.

`-tn:`*`milisegundos`*

Especifica el número de *`milisegundos`* entre las entradas de registro que se escriben. El valor predeterminado es de 1000 milisegundos (1 segundo).

`-sm:`*`bytes`*

Especifica el número de *`bytes`* de cada entrada del registro. El valor predeterminado es 1000 bytes.

`-bk:`*`número`*

Especifica el *`número`* de entradas del registro que se van a escribir a la vez. El valor predeterminado es 1.

- A veces puede resultar útil para simular una aplicación que escribe en el registro de eventos de Windows. Utilice el conmutador `/e` para escribir entradas en un registro de eventos de Windows; por ejemplo:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
KTDiag.exe /e Application
```

De este modo, se escriben entradas en el registro de eventos de la aplicación de Windows. Si lo desea, también puede especificar las siguientes opciones adicionales tras el nombre del registro:

`-tn:`*`milisegundos`*

Especifica el número de *`milisegundos`* entre las entradas de registro que se escriben. El valor predeterminado es de 1000 milisegundos (1 segundo).

`-sm:`*`bytes`*

Especifica el número de *`bytes`* de cada entrada del registro. El valor predeterminado es 1000 bytes.

`-bk:`*`número`*

Especifica el *`número`* de entradas del registro que se van a escribir a la vez. El valor predeterminado es 1.

Creación de Kinesis Agent para complementos de Windows

En la mayoría de las situaciones, no es necesario crear un complemento de Amazon Kinesis Agent para Microsoft Windows. Kinesis Agent para Windows tiene una gran capacidad de configuración y dispone de orígenes y receptores muy eficaces, como`DirectorySource`y`KinesisStream`, que son suficientes para la mayoría de los escenarios. Para obtener más información sobre los orígenes y receptores existentes, consulte Configuración de Amazon Kinesis Agent para Microsoft Windows.

En escenarios poco comunes, puede ser necesario ampliar el agente de Kinesis para Windows con un complemento personalizado. Algunos de estos escenarios podrían ser los siguientes:

- Se está empaquetando una declaración `DirectorySource` compleja con analizadores de recursos `Regex` o `Delimited` para que resulte fácil aplicarla en diferentes tipos de archivos de configuración.
- Se está creando un nuevo origen que no está basado en un archivo o que sobrepasa las funcionalidades de análisis que proporcionan los analizadores de registros existentes.
- Se está creando un receptor para un servicio de AWS que actualmente no es compatible.

Temas

Introducción a los complementos de Kinesis Agent para Windows

No hay nada especial en los complementos personalizados. Todos los orígenes y receptores existentes utilizan los mismos mecanismos que usan los complementos personalizados para cargarse cuando se inicia el agente Kinesis para Windows y crean instancias de los complementos apropiados después de leer el`appsettings.json`Archivo de configuración.

Cuando se inicia el agente Kinesis para Windows, se produce la siguiente secuencia:

1. Kinesis Agent para Windows analiza los ensamblados en el directorio de instalación (`%PROGRAMFILES%\Amazon\AWSKinesisTap`) para las clases que implementan

elIFactory<T>definida en la interfazAmazon.KinesisTap.CoreAssembly EnAmazon.KinesisTap.Core\Infrastructure\IFactory.csen el código fuente de Kinesis Agent para Windows.

2. Kinesis Agent para Windows carga los ensamblados que contienen estas clases e invoca el métodoRegisterFactoryen estas clases.
3. Kinesis Agent para Windows carga elappsettings.jsonArchivo de configuración. En cada origen y receptor del archivo de configuración, se examinan los pares clave-valor SourceType y SinkType. Si hay factorías registradas con el mismo nombre que los valores de los pares clave-valor SourceType y SinkType, se invoca el método CreateInstance en dichas factorías. El método CreateInstance recibe la configuración y otra información como un objeto IPluginContext. El método CreateInstance es responsable de configurar e inicializar el complemento.

Para que un complemento funcione correctamente, debe haber una clase de factoría registrada que cree el complemento y debe definirse la propia clase del complemento.

El código fuente de Kinesis Agent para Windows se encuentra enhttps://github.com/awslabs/kinesis-agent-windows.

Implementación del agente Kinesis para fábricas de complementos de Windows

Siga estos pasos para implementar una factoría de complementos de Kinesis Agent para Windows.

Para crear una fábrica de complementos de Kinesis Agent para Windows

1. Cree un proyecto de biblioteca de C# que tenga como destino .NET Framework 4.6.
2. Añada una referencia al ensamblado Amazon.KinesisTap.Core. Este ensamblado se encuentra en el%PROGRAMFILES%\Amazon\AWSKinesisTapdespués de la instalación de Kinesis Agent para Windows.
3. Utilice NuGet para instalar el paquete Microsoft.Extensions.Configuration.Abstractions.
4. Utilice NuGet para instalar el paquete System.Reactive.
5. Utilice NuGet para instalar el paquete Microsoft.Extensions.Logging.

6. Cree una clase de factoría que implemente `IFactory<IEventSource>` para los orígenes o `IFactory<IEventSink>` para los receptores. Añada los métodos `CreateInstance` y `RegisterFactory`.

   Por ejemplo, el siguiente código crea una factoría de complementos de Kinesis Agent para Windows que crea un origen que genera datos aleatorios:

```
using System;
using Amazon.KinesisTap.Core;
using Microsoft.Extensions.Configuration;

namespace MyCompany.MySources
{
    public class RandomSourceFactory : IFactory<ISource>
    {
        public void RegisterFactory(IFactoryCatalog<ISource> catalog)
        {
            catalog.RegisterFactory("randomsource", this);
        }

        public ISource CreateInstance(string entry, IPlugInContext context)
        {
            IConfiguration config = context.Configuration;

            switch (entry.ToLower())
            {
                case "randomsource":
                    string rateString = config["Rate"];
                    string maxString = config["Max"];
                    TimeSpan rate;
                    int max;

                    if (string.IsNullOrWhiteSpace(rateString))
                    {
                        rate = TimeSpan.FromSeconds(30);
                    }
                    else
                    {
                        if (!TimeSpan.TryParse(rateString, out rate))
                        {
                            throw new Exception($"Rate {rateString} is invalid for
 RandomSource.");
                        }
```

```
                    }

                    if (string.IsNullOrWhiteSpace(maxString))
                    {
                        max = 1000;
                    }
                    else
                    {
                        if (!int.TryParse(maxString, out max))
                        {
                            throw new Exception($"Max {maxString} is invalid for
 RandomSource.");
                        }
                    }

                    return new RandomSource(rate, max, context);
                default:
                    throw new ArgumentException($"Source {entry} is not
 recognized.", entry);
            }
        }
    }
}
```

La instrucción `switch` se utilizará en el método `CreateInstance` en caso de que finalmente desee mejorar la factoría para crear diferentes tipos de instancias.

Para crear un factoría de receptores que cree un receptor que no haga nada, utilice una clase similar a la siguiente:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Amazon.KinesisTap.Core;
using Microsoft.Extensions.Configuration;

namespace MyCompany.MySinks
{
    public class NullSinkFactory : IFactory<IEventSink>
    {
        public void RegisterFactory(IFactoryCatalog<IEventSink> catalog)
```

```
        {
            catalog.RegisterFactory("nullsink", this);
        }

        public IEventSink CreateInstance(string entry, IPlugInContext context)
        {
            IConfiguration config = context.Configuration;

            switch (entry.ToLower())
            {
                case "nullsink":
                    return new NullSink(context);
                default:
                    throw new Exception("Unrecognized sink type {entry}.");
            }
        }
    }
}
```

Implementación del agente Kinesis para fuentes de complementos de Windows

Siga estos pasos para implementar un origen de complementos de Kinesis Agent para Windows.

Para crear un origen de complemento de Kinesis Agent para Windows

1. Añada una clase que implemente la interfaz `IEventSource<out T>` al proyecto que creó con anterioridad para el origen.

   Por ejemplo, utilice el siguiente código para definir un origen que genere datos aleatorios:

```
using System;
using System.Reactive.Subjects;
using System.Timers;
using Amazon.KinesisTap.Core;
using Microsoft.Extensions.Logging;

namespace MyCompany.MySources
{
    public class RandomSource : EventSource<RandomData>, IDisposable
    {
```

```
        private TimeSpan _rate;
        private int _max;
        private Timer _timer = null;
        private Random _random = new Random();
        private ISubject<IEnvelope<RandomData>> _recordSubject = new
Subject<IEnvelope<RandomData>>();



        public RandomSource(TimeSpan rate, int max, IPlugInContext context) :
base(context)
        {
            _rate = rate;
            _max = max;
        }

        public override void Start()
        {
            try
            {
                CleanupTimer();
                _timer = new Timer(_rate.TotalMilliseconds);
                _timer.Elapsed += (Object source, ElapsedEventArgs args) =>
                {
                    var data = new RandomData()
                    {
                        RandomValue = _random.Next(_max)
                    };
                    _recordSubject.OnNext(new Envelope<RandomData>(data));
                };
                _timer.AutoReset = true;
                _timer.Enabled = true;
                _logger?.LogInformation($"Random source id {this.Id} started with
rate {_rate.TotalMilliseconds}.");
            }
            catch (Exception e)
            {
                _logger?.LogError($"Exception during start of RandomSource id
{this.Id}: {e}");
            }
        }

        public override void Stop()
        {
```

```
            try
            {
                CleanupTimer();
                _logger?.LogInformation($"Random source id {this.Id} stopped.");

            }
            catch (Exception e)
            {
                _logger?.LogError($"Exception during stop of RandomSource id
 {this.Id}: {e}");
            }
        }

        private void CleanupTimer()
        {
            if (_timer != null)
            {
                _timer.Enabled = false;
                _timer?.Dispose();
                _timer = null;
            }
        }

        public override IDisposable Subscribe(IObserver<IEnvelope<RandomData>>
 observer)
        {
            return this._recordSubject.Subscribe(observer);
        }

        public void Dispose()
        {
            CleanupTimer();
        }
    }
}
```

En este ejemplo, la clase `RandomSource` hereda de la clase `EventSource<T>`, ya que proporciona la propiedad `Id`. Aunque este ejemplo no admite marcadores, esta clase base también resulta útil para implementar esta funcionalidad. Los sobres constituyen un mecanismo para almacenar metadatos y envolver datos arbitrarios para transmitirlos a los receptores. La clase `RandomData` se define en el siguiente paso y representa el tipo de objeto de salida de este origen.

2. Añada al proyecto que definió anteriormente una clase que contenga los datos transmitidos desde el origen.

   Por ejemplo, un contenedor de datos aleatorios podría definirse del modo siguiente:

```
namespace MyCompany.MySources
{
    public class RandomData
    {
        public int RandomValue { get; set; }
    }
}
```

3. Compile el proyecto definido anteriormente.
4. Copie el ensamblado en el directorio de instalación de Kinesis Agent para Windows.
5. Cree o actualice un`appsettings.json`que utilice el nuevo origen y sitúelo en el directorio de instalación de Kinesis Agent para Windows.
6. Detenga e inicie Kinesis Agent para Windows.
7. Compruebe el archivo de registro de Kinesis Agent para Windows actual (normalmente se encuentra en la carpeta`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`) para comprobar que no hay ningún problema con el complemento de origen personalizado.
8. Asegúrese de que llegan datos al servicio de AWS deseado.

Para ver un ejemplo de cómo ampliar el`DirectorySource`Para implementar el análisis de un determinado formato de registro, consulte`Amazon.KinesisTap.Uls\UlsSourceFactory.cs`y`Amazon.KinesisTap.Uls\UlsLogParser.cs`en el código fuente de Kinesis Agent para Windows.

Para obtener un ejemplo acerca de cómo crear un origen que proporcione la funcionalidad de marcadores, consulte`Amazon.KinesisTap.Windows\WindowsSourceFactory.cs`y`Amazon.KinesisTap.Windows\EventLogSource.cs`en el código fuente de Kinesis Agent para Windows.

Implementación del agente Kinesis para sumideros de complementos de Windows

Siga estos pasos para implementar un receptor de complementos de Kinesis Agent para Windows.

Para crear un receptor del complemento de Kinesis Agent para Windows

1. Añada una clase que implemente la interfaz `IEventSink` al proyecto que definió anteriormente.

   Por ejemplo, el código siguiente implementa un receptor que no hace nada más que registrar la llegada de entradas de registro y luego las descarta.

```
using Amazon.KinesisTap.Core;
using Microsoft.Extensions.Logging;

namespace MyCompany.MySinks
{
    public class NullSink : EventSink
    {
        public NullSink(IPlugInContext context) : base(context)
        {
        }

        public override void OnNext(IEnvelope envelope)
        {
            _logger.LogInformation($"Null sink {Id} received
 {GetRecord(envelope)}.");
        }

        public override void Start()
        {
            _logger.LogInformation($"Null sink {Id} starting.");
        }

        public override void Stop()
        {
            _logger.LogInformation($"Null sink {Id} stopped.");
        }
    }
}
```

   En este ejemplo, la clase del receptor `NullSink` hereda de la clase `EventSink` porque le brinda la posibilidad de transformar registros en diferentes formatos de serialización, como JSON y XML.

2. Compile el proyecto definido anteriormente.
3. Copie el ensamblado en el directorio de instalación de Kinesis Agent para Windows.

4. Cree o actualice un`appsettings.json`que utilice el nuevo receptor y sitúelo en el directorio de instalación de Kinesis Agent para Windows. Por ejemplo, para utilizar los complementos personalizados `RandomSource` y `NullSink`, puede utilizar el siguiente archivo de configuración `appsettings.json`:

```
{
  "Sources": [
  {
 "Id": "MyRandomSource",
 "SourceType": "RandomSource",
 "Rate": "00:00:10",
 "Max": 50
  }

  ],
  "Sinks": [
  {
    "Id": "MyNullSink",
 "SinkType": "NullSink",
 "Format": "json"
  }
  ],
  "Pipes": [
    {
   "Id": "MyRandomToNullPipe",
   "SourceRef": "MyRandomSource",
   "SinkRef": "MyNullSink"
 }
  ]
}
```

   Esta configuración crea un origen que envía una instancia de `RandomData` con un `RandomValue` establecido en un número aleatorio entre 0 y 50 cada 10 segundos. Crea un receptor que transforma las instancias de `RandomData` entrantes en JSON, registra este código JSON y después descarta las instancias. No olvide incluir las dos factorías del ejemplo, la clase del origen `RandomSource` la clase del receptor `NullSink` en el proyecto que definió anteriormente para poder utilizar el archivo de configuración del ejemplo.

5. Detenga e inicie Kinesis Agent para Windows.

6. Compruebe el archivo de registro de Kinesis Agent para Windows actual (normalmente se encuentra en la carpeta`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`) para comprobar que no hay ningún problema con el complemento del receptor personalizado.
7. Asegúrese de que llegan datos al servicio de AWS deseado. Como la clase `NullSink` de ejemplo no transmite datos a los servicios de AWS, puede comprobar que el receptor funciona correctamente buscando mensajes de registro que indiquen que se han recibido los registros.

   Por ejemplo, podría ver un archivo de registro similar al siguiente:

```
2018-10-18 12:36:36.3647 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.AWS.AWSEventSinkFactory.
2018-10-18 12:36:36.4018 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Windows.PerformanceCounterSinkFactory.
2018-10-18 12:36:36.4018 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory MyCompany.MySinks.NullSinkFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Core.DirectorySourceFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.ExchangeSource.ExchangeSourceFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Uls.UlsSourceFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Windows.WindowsSourceFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory MyCompany.MySources.RandomSourceFactory.
2018-10-18 12:36:36.9601 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Core.Pipes.PipeFactory.
2018-10-18 12:36:37.4694 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.AutoUpdate.AutoUpdateFactory.
2018-10-18 12:36:37.4807 Amazon.KinesisTap.Hosting.LogManager INFO Performance
 counter sink  started.
2018-10-18 12:36:37.6250 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink starting.
2018-10-18 12:36:37.6250 Amazon.KinesisTap.Hosting.LogManager INFO Connected source
 MyRandomSource to sink MyNullSink
2018-10-18 12:36:37.6333 Amazon.KinesisTap.Hosting.LogManager INFO Random source id
 MyRandomSource started with rate 10000.
2018-10-18 12:36:47.8084 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":14}.
2018-10-18 12:36:57.6339 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":5}.
```

```
2018-10-18 12:37:07.6490 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":9}.
2018-10-18 12:37:17.6494 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":47}.
2018-10-18 12:37:27.6520 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":25}.
2018-10-18 12:37:37.6676 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":21}.
2018-10-18 12:37:47.6688 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":29}.
2018-10-18 12:37:57.6700 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":22}.
2018-10-18 12:38:07.6838 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":32}.
2018-10-18 12:38:17.6848 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":12}.
2018-10-18 12:38:27.6866 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":46}.
2018-10-18 12:38:37.6880 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":48}.
2018-10-18 12:38:47.6893 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":39}.
2018-10-18 12:38:57.6906 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":18}.
2018-10-18 12:39:07.6995 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":6}.
2018-10-18 12:39:17.7004 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":0}.
2018-10-18 12:39:27.7021 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":3}.
2018-10-18 12:39:37.7023 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":19}.
```

Si crea un receptor con acceso a los servicios de AWS, hay clases base que podrían resultarle útiles. Para un fregadero que utiliza el`AWSBufferedEventSink`clase base, consulte`Amazon.KinesisTap.AWS\CloudWatchLogsSink.cs`en el código fuente de Kinesis Agent para Windows.

Guía del usuario de Amazon Kinesis Agent para Microsoft Windows

Versión de API: 10/10/2018

En la siguiente tabla se describen los cambios en elGuía del usuario de Amazon Kinesis Agent para Microsoft Windows(este documento).

update-history-change	update-history-description	update-history-date
Actualización de la documentación importante	Se han agregado instrucciones para la instalación de MSI. Se actualizó la configuración de DirectorySource y se agregó WindowsEventLogPolling Para la configuración del receptor, se agregó la configuración de sincronización del sistema de archivos local; ProfileRefrreshing AWScredentialProvider; información sobre decoraciones de texto, resolución de variables en atributos de receptor, configuración de extremos regionales STS para sumideros, configuración de extremos VPC y configuración de servidores proxy alternativos. Para tuberías, se han añadido atributos de configuración.	23 de febrero de 2021
Actualización de la documentación	Se ha actualizado para informar de que las especificaciones de las ubicaciones de	7 de noviembre de 2018

	Amazon S3 distinguen entre mayúsculas y	
Lanzamiento inicial de la versión 1.0.0.0.115	Primera versión de la Guía del usuario de Kinesis Agent para Windows.	5 de noviembre de 2018

Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el glosario de AWS en la referencia general de AWS.

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.