aws

Documento técnico de AWS

Soluciones de datos de streaming en AWS con Amazon Kinesis

Soluciones de datos de streaming en AWS con Amazon Kinesis: Documento técnico de AWS

Table of Contents

Soluciones de datos de streaming en AWS

Fecha de publicación: 1 de septiembre de 2021 (Revisiones del documento)

Resumen

Los ingenieros de datos, los analistas de datos y los desarrolladores de macrodatos desean hacer evolucionar sus análisis de lotes en tiempo real para que sus empresas puedan conocer lo que sus clientes, aplicaciones y productos están haciendo en este momento y reaccionar con rapidez. Este documento técnico analiza la evolución de los análisis desde la fase de los análisis por lotes hasta el tiempo real. Describe cómo se pueden utilizar servicios como Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon EMR, Amazon Kinesis Data Analytics, Amazon Managed Streaming for Apache Kafka (Amazon MSK) y otros servicios para implementar aplicaciones en tiempo real y proporciona patrones de diseño comunes que los utilizan estos servicios.

Introducción

Las empresas actuales reciben datos a gran escala y velocidad debido al crecimiento explosivo de los orígenes de datos que generan secuencias de datos de forma continua. Ya sean datos de registro de los servidores de aplicaciones, datos de secuencias de clics de sitios web y aplicaciones móviles o datos de telemetría de dispositivos del Internet de las cosas (IoT), todo contiene información que puede ayudarle a obtener información sobre lo que sus clientes, aplicaciones y productos están haciendo en ese momento.

Tener la capacidad de procesar y analizar estos datos en tiempo real es esencial para realizar acciones como supervisar continuamente sus aplicaciones para garantizar un elevado tiempo de actividad del servicio y personalizar las ofertas promocionales y las recomendaciones de productos. El procesamiento en tiempo real y casi real también puede convertir otros casos de uso comunes, como el análisis de sitios web y el machine learning, en más precisos y procesables al hacer que los datos estén disponibles para estas aplicaciones en segundos o minutos en lugar de tardar horas o días.

Escenarios de aplicaciones en tiempo real y casi real

Puede los servicios de datos de streaming para aplicaciones de tiempo real y casi real, como supervisión de aplicaciones, detección de fraude y tablas de clasificaciones en directo. Los casos de uso en tiempo real requieren latencias de milisegundos de extremo a extremo, desde la ingesta hasta el procesamiento, pasando por el envío de los resultados a los almacenes de datos de destino y otros sistemas. Por ejemplo, Netflix usa Amazon Kinesis Data Streams para controlar las comunicaciones entre todas sus aplicaciones con el objetivo de detectar y corregir errores rápidamente, lo que garantiza un muy buen nivel de tiempo de actividad y disponibilidad a sus clientes. Aunque el caso de uso más aplicable habitualmente es la supervisión del rendimiento de las aplicaciones, hay un número cada vez mayor de aplicaciones en tiempo real en tecnología publicitaria, juegos e IoT que se incluyen en esta categoría.

Los casos de uso habituales casi en tiempo real incluyen el análisis de almacenes de datos para la ciencia de datos y el machine learning (ML). Puede utilizar soluciones de datos de streaming para cargar continuamente datos en tiempo real en sus lagos de datos. A continuación, puede actualizar modelos de machine learning con mayor frecuencia a medida que se pongan a disposición nuevos datos, lo que garantiza la precisión y la fiabilidad de los resultados. Por ejemplo, Zillow utiliza Kinesis Data Streams para recopilar datos de registros públicos y listas de varios servicios de listados (MLS), y luego proporcionar a los compradores y vendedores de propiedades las estimaciones más

actualizadas del valor de la vivienda casi en tiempo real. ZipRecruiter usa Amazon MSK para sus canalizaciones de registros de eventos. Estos son los componentes críticos de la infraestructura, que recopilan, almacenan y procesan continuamente más de 6 000 000 000 de eventos por día desde el centro de empleo de ZipRecruiter.

Diferencias entre procesamiento por lotes y secuencias

Necesita un conjunto de herramientas distinto para recopilar, preparar y procesar datos de streaming en tiempo real en vez de las herramientas que ha utilizado tradicionalmente para el análisis por lotes. Con el análisis tradicional, se recopilan los datos, se cargan periódicamente en una base de datos y se analizan horas, días o semanas después. El análisis de datos en tiempo real requiere un enfoque distinto. Las aplicaciones de procesamiento de secuencias procesan los datos de forma continua en tiempo real, incluso antes de almacenarlos. Los datos de streaming pueden llegar a un ritmo vertiginoso y los volúmenes de datos pueden aumentarse o reducirse en cualquier momento. Las plataformas de procesamiento de datos de secuencias deben poder gestionar la velocidad y la variabilidad de los datos entrantes y procesarlos a medida que llegan, a menudo de millones a cientos de millones de eventos por hora.

Desafíos de procesamiento de secuencias

El procesamiento de datos en tiempo real a medida que llegan puede permitirle tomar decisiones mucho más rápido de lo que es posible con las tecnologías de análisis de datos tradicionales. Sin embargo, crear y utilizar sus propias canalizaciones de datos de streaming personalizadas es complicado y requiere muchos recursos:

- Debe crear un sistema que pueda recopilar, preparar y transmitir de manera rentable los datos procedentes simultáneamente de miles de orígenes de datos.
- Debe ajustar los recursos de computación y de almacenamiento para que los datos se agrupen y se transmitan de manera eficiente para obtener el máximo rendimiento y una baja latencia.
- Debe implementar y administrar una flota de servidores para escalar el sistema y poder gestionar las velocidades variables de los datos que va a enviar.

La actualización de la versión es un proceso complejo y costoso. Una vez que haya creado esta plataforma, debe supervisar el sistema y recuperar cualquier error de servidor o red al poniéndose al día con el procesamiento de datos desde el punto apropiado de la secuencia, sin crear datos duplicados. También necesita un equipo dedicado a la administración de la infraestructura. Todo esto

requiere tiempo y dinero valiosos y, al final, la mayoría de las empresas simplemente nunca llegan allí y deben conformarse con el statu quo y operar sus empresas con información que tiene horas o días de antigüedad.

Soluciones de datos de streaming: ejemplos

Escenario 1: oferta de Internet basada en la ubicación

La empresa InternetProvider presta servicios de Internet con diferentes opciones de ancho de banda a los usuarios de todo el mundo. Cuando un usuario se registra en Internet, la empresa InternetProvider proporciona al usuario distintas opciones de ancho de banda según su ubicación geográfica. En función de estos requisitos, la empresa InternetProvider implementó Amazon Kinesis Data Streams para consumir los detalles y la ubicación del usuario. Los detalles y la ubicación del usuario se enriquecen con distintas opciones de ancho de banda antes de volver a publicarse en la aplicación. AWS Lambdapermite este enriquecimiento en tiempo real.

Procesamiento de secuencias de datos con AWS Lambda

Amazon Kinesis Data Streams

Amazon Kinesis Data Streams le permite crear aplicaciones personalizadas en tiempo real mediante marcos de procesamiento de streaming conocidos y cargar datos de streaming en numerosos almacenes de datos. Una secuencia de Kinesis se puede configurar para recibir continuamente eventos de cientos de miles de productores de datos entregados desde orígenes como transmisiones de clics de sitios web, sensores de IoT, fuentes de redes sociales y registros de aplicaciones. En milisegundos, los datos están disponibles para que su aplicación los lea y procese.

Al implementar una solución con Kinesis Data Streams, crea aplicaciones de procesamiento de datos personalizadas que se denominan aplicaciones de Kinesis Data Streams. Una aplicación típica de Kinesis Data Streams lee los datos de una secuencia de Kinesis como registros de datos.

Los datos incluidos en Kinesis Data Streams garantizan una alta disponibilidad y elasticidad, y están disponibles en cuestión de milisegundos. Puede agregar constantemente distintos tipos de datos (por ejemplo, secuencias de clics, registros de aplicaciones y medios sociales) de cientos de miles de

fuentes a una secuencia de Kinesis. En cuestión de segundos, los datos estarán disponibles en sus aplicaciones de Kinesis para su lectura y procesamiento desde la secuencia.

Amazon Kinesis Data Streams es un servicio de datos de streaming completamente administrado. Se encarga de administrar la infraestructura, el almacenamiento, las redes y la configuración que se necesitan para transmitir sus datos al nivel de su caudal.

Envío de datos a Amazon Kinesis Data Streams

Hay varias formas de enviar datos a Kinesis Data Streams, lo que le proporciona flexibilidad en los diseños de sus soluciones.

- Puede escribir código con uno de los SDK de AWS que son compatibles con varios lenguajes conocidos.
- Puede utilizar el agente de Amazon Kinesis, una herramienta para enviar datos a Kinesis Data Streams.

Amazon Kinesis Producer Library (KPL) simplifica el desarrollo de aplicaciones productoras, lo cual permite a los desarrolladores alcanzar un alto rendimiento de escritura en una o más secuencias de datos de Kinesis.

KPL es una biblioteca fácil de usar y con una gran capacidad de configuración que se instala en sus hosts. Ejerce de intermediaria entre el código de aplicación del productor y las acciones de la API de Kinesis Streams. Para obtener más información sobre KPL y su capacidad de producir eventos de forma sincrónica y asincrónica con ejemplos de código, consulte Escribir en sus secuencias de datos de Kinesis con KPL

Hay dos operaciones distintas en la API de Kinesis Data Streams que agregan datos a una secuencia: `PutRecords` y `PutRecord`. La operación `PutRecords` envía varios registros a su secuencia por solicitud HTTP, mientras que `PutRecord` envía un registro por solicitud HTTP. Para obtener un mayor rendimiento en la mayoría de las aplicaciones, utilice `PutRecords`.

Para obtener más información sobre estas API, consulte Agregar datos a una secuencia. Los detalles de cada operación de la API se pueden encontrar en la referencia de la API de Amazon Kinesis Data Streams.

Procesamiento de datos en Amazon Kinesis Data Streams

Para leer y procesar datos de secuencias de Kinesis, debe crear una aplicación de consumidor. Hay varias formas de crear consumidores para Kinesis Data Streams. Algunos de estos enfoques

incluyen el uso de Amazon Kinesis Data Analytics para analizar datos de streaming mediante KCL, AWS Lambda, trabajos ETL de streaming de AWS Glue y la API de Kinesis Data Streams directamente.

Las aplicaciones de consumidor para Kinesis Data Streams se pueden desarrollar con KCL, que le ayuda a consumir y procesar datos de Kinesis Data Streams. KCL se encarga de muchas de las tareas complejas asociadas a la computación distribuida, como el equilibrio de carga entre varias instancias, la respuesta a errores en las instancias, la creación de puntos de control en registros procesados y la reacción a cambios en las particiones. La KCL le permite concentrarse en la lógica de procesamiento de registros de escritura. Para obtener más información sobre cómo crear su propia aplicación KCL, consulte Uso de la Kinesis Client Library.

Puede suscribir las funciones Lambda para que lean lotes de registros automáticamente de la secuencia de Kinesis y los procese si son detectados en dicha secuencia. AWS Lambda sondea periódicamente la secuencia (una vez por segundo) en busca de nuevos registros y, cuando los detecta, invoca la función Lambda que pasa los nuevos registros como parámetros. La función Lambda solo se ejecuta cuando se detectan nuevos registros. Puede asignar una función Lambda a un consumidor de rendimiento compartido (iterador estándar)

Puede crear un consumidor que utilice una característica llamada distribución ramificada mejorada cuando necesite un rendimiento dedicado que no desee competir con otros consumidores que reciben datos de la secuencia. Esta característica permite a los consumidores recibir registros de una secuencia con un rendimiento de hasta dos MB de datos por segundo por partición.

En la mayoría de los casos, usar Kinesis Data Analytics, KCL, AWS Glue oAWS Lambda debe usarse para procesar los datos de una secuencia. Sin embargo, si lo prefiere, puede crear una aplicación de consumidor desde cero con la API de Kinesis Data Streams. La API de Kinesis Data Streams ofrece los métodos `GetShardIterator` y `GetRecords` para recuperar datos de una secuencia.

En este modelo de extracción, el código extrae los datos directamente de las particiones de la secuencia. Para obtener más información sobre cómo escribir su propia aplicación de consumidor con la API, consulte Desarrollo de consumidores personalizados con rendimiento compartido con AWS SDK para Java. Los detalles sobre la API se pueden encontrar en la referencia de la API de Amazon Kinesis Data Streams.

Procesamiento de secuencias de datos con AWS Lambda

AWS Lambda le permite ejecutar código sin aprovisionar ni administrar servidores. Con Lambda, puede ejecutar código para casi cualquier tipo de aplicación o servicio backend sin administrar nada. Solo tiene que cargar su código y Lambda se ocupará de todo lo necesario para ejecutarlo y escalarlo con alta disponibilidad. Puede configurar el código para que se desencadene automáticamente desde otros servicios de AWS o puede llamarlo directamente desde cualquier aplicación web o móvil.

AWS Lambda se integra de forma nativa con Amazon Kinesis Data Streams. Las complejidades del sondeo, los puntos de comprobación y la gestión de errores se abstraen cuando se utiliza esta integración nativa. Esto permite que el código de función Lambda se centre en el procesamiento de lógica empresarial.

Puede asignar una función Lambda a un consumidor rendimiento compartido (iterador estándar) o de rendimiento dedicado con distribución ramificada mejorada. Con un iterador estándar, Lambda sondea cada partición de la secuencia de Kinesis en busca de registros utilizando el protocolo HTTP. Para minimizar la latencia y maximizar el rendimiento de lectura, puede crear un consumidor de flujo de datos con distribución ramificada mejorada. Los consumidores de secuencias en esta arquitectura obtienen una conexión dedicada a cada partición sin competir con otras aplicaciones que leen la misma secuencia. Amazon Kinesis Data Streams envía los registros a Lambda por HTTP/2.

De forma predeterminada, AWS Lambda llama a su función en cuanto los registros están disponibles en la secuencia. Para almacenar en búfer los registros para escenarios de lotes, puede implementar una ventana de lote de hasta cinco minutos en el origen del evento. Si la función devuelve un error, Lambda volverá a intentar ejecutar el lote hasta que el procesamiento se realice correctamente o los datos caduquen.

Resumen

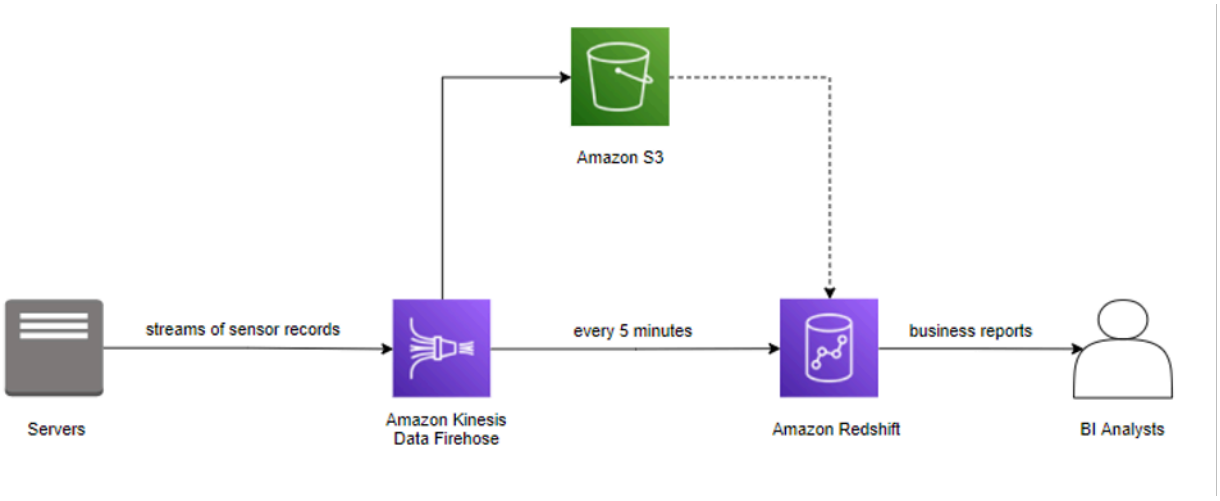
La empresa InternetProvider aprovecho Amazon Kinesis Data Streams para transmitir los detalles y la ubicación de usuario. AWS Lambda consumió la secuencia del registro para enriquecer los datos con opciones de ancho de banda almacenadas en la biblioteca de la función. Después del enriquecimiento, AWS Lambda publica las opciones de ancho de banda en la aplicación. Amazon Kinesis Data Streams y AWS Lambda gestionaron el aprovisionamiento y la administración de servidores, lo que permitió a la empresa InternetProvider centrarse más en el desarrollo de aplicaciones empresariales.

Escenario 2: datos casi en tiempo real para los equipos de seguridad

La empresa ABC2Badge proporciona sensores e insignias para eventos corporativos o a gran escala, como AWS re:Invent. Los usuarios se inscriben en el evento y reciben insignias únicas que los sensores captan en todo el campus. A medida que los usuarios pasan por un sensor, su información anónima se registra en una base de datos relacional.

En un evento próximo, debido al gran volumen de asistentes, el equipo de seguridad del evento solicitó a ABC2Badge que recopilara datos para las áreas de mayor concentración del campus cada 15 minutos. Esto le dará al equipo de seguridad tiempo suficiente para reaccionar y repartir al personal de seguridad proporcionalmente a las áreas concentradas. Debido a este nuevo requisito del equipo de seguridad y la inexperiencia de crear una solución de streaming, para procesar los datos casi en tiempo real, ABC2Badge busca una solución simple pero escalable y fiable.

Su solución de almacenamiento de datos actual es Amazon Redshift. Al revisar las características de los servicios de Amazon Kinesis, reconoció que Amazon Kinesis Data Firehose puede recibir una secuencia de registros de datos, agrupar los registros en lotes según el tamaño del búfer o el intervalo de tiempo, e insertarlos en Amazon Redshift. Creó una secuencia de entrega de Kinesis Data Firehose y la configuró para que copiara los datos en sus tablas de Amazon Redshift cada cinco minutos. Como parte de esta nueva solución, utilizaron el agente de Amazon Kinesis en sus servidores. Cada cinco minutos, Kinesis Data Firehose carga datos en Amazon Redshift, donde el equipo de inteligencia empresarial (BI) puede realizar el análisis y enviar los datos al equipo de seguridad cada 15 minutos.



Nueva solución con Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose es la forma más fácil de cargar datos de streaming en AWS. Puede capturar, transformar y cargar datos de streaming en Amazon Kinesis Data Analytics, Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service (OpenSearch Service) y Splunk. Además, Kinesis Data Firehose puede cargar datos de streaming en cualquier punto de conexión HTTP personalizado o punto de conexión HTTP propiedad de proveedores de servicios de terceros compatibles.

Kinesis Data Firehose permite el análisis casi en tiempo real con herramientas y paneles de inteligencia empresarial existentes que ya está utilizando en la actualidad. Se trata de un servicio sin servidor completamente administrado cuya escala se ajusta de forma automática para adaptarse al rendimiento de los datos y que no precisa administración permanente. Kinesis Data Firehose puede procesar por lotes, comprimir y cifrar los datos antes de cargarlos, a fin de minimizar la cantidad de almacenamiento utilizado en el destino y aumentar la seguridad. También puede transformar los datos de origen con AWS Lambda y entregar los datos transformados a los destinos. Usted configura los productores de datos para que envíen datos a Kinesis Data Firehose, que entrega inmediatamente los datos al destino que usted especifique.

Envío de datos a una secuencia de entrega de Firehose

Para enviar datos a su secuencia de entrega, hay varias opciones. AWS ofrece SDK para muchos lenguajes de programación conocidos, cada uno de los cuales proporciona las API para Amazon Kinesis Data Firehose. AWS dispone de una utilidad para ayudar a enviar datos a su secuencia de entrega. Kinesis Data Firehose se ha integrado con otros servicios de AWS para enviar datos directamente desde esos servicios a su secuencia de entrega.

Uso del agente de Amazon Kinesis

El agente de Amazon Kinesis Agent es una aplicación de software independiente que supervisa continuamente un conjunto de archivos de registro para que se envíen nuevos datos a la secuencia de entrega. El agente gestiona automáticamente la rotación de archivos, los puntos de control, los reintentos en caso de error y emite métricas de Amazon CloudWatch para la supervisión y la solución de problemas de la secuencia de entrega. Se pueden aplicar al agente configuraciones adicionales, como el preprocesamiento de datos, la supervisión de varios directorios de archivos y la escritura en varias secuencias de entrega.

El agente se puede instalar en servidores basados en Linux o Windows, como servidores web, de registro y de bases de datos. Una vez instalado el agente, solo tiene que especificar los archivos de

registro que supervisará y la secuencia de entrega a la que se enviará. El agente enviará nuevos datos de manera duradera y fiable a la secuencia de entrega.

Uso de la API con AWS SDK y los servicios de AWS como origen

La API de Kinesis Data Firehose ofrece dos operaciones para enviar datos a su secuencia de entrega. `PutRecord` envía un registro de datos en una sola llamada. `PutRecordBatch` puede enviar varios registros de datos en una sola llamada y puede obtener un mayor rendimiento por productor. En cada método, debe especificar el nombre de la secuencia de entrega y el registro de datos, o matriz de registros de datos, al utilizar este método. Para obtener más información y código de muestra para las operaciones de la API de Kinesis Data Firehose, consulte Escribir en una secuencia de entrega de Firehose con el SDK de AWS.

Kinesis Data Firehose también se ejecuta con Kinesis Data Firehose, CloudWatch Logs, CloudWatch Events, Amazon Simple Notification Service (Amazon SNS), Amazon API Gateway y AWS IoT. Puede enviar sus secuencias de datos, registros, eventos y datos de IoT de manera escalable y fiable directamente a un destino de Kinesis Data Firehose.

Procesamiento de datos antes de la entrega en destino

En algunos casos, es recomendable que transforme o mejore sus datos de streaming antes de que se entreguen a su destino. Por ejemplo, los productores de datos pueden enviar texto no estructurado en cada registro de datos y debe transformarlo en JSON antes de entregarlo a OpenSearch Service. O puede que desee convertir los datos JSON a un formato de archivo en columnas, como Apache Parquet o Apache ORC antes de almacenar los datos en Amazon S3.

Kinesis Data Firehose tiene incorporada la capacidad de conversión de formato de datos incorporada. Con esto, puede convertir fácilmente sus secuencias de datos JSON en formatos de archivo Apache Parquet o Apache ORC.

Flujo de transformación de datos

Para habilitar las transformaciones de datos en streaming, Kinesis Data Firehose utiliza una función Lambda que ha creado para transformar los datos. Kinesis Data Firehose almacena en búfer los datos entrantes en un tamaño de búfer especificado para la función y, a continuación, invoca la función Lambda especificada de forma asíncrona. Los datos transformados se envían de Lambda a Kinesis Data Firehose y Kinesis Data Firehose entrega los datos al destino.

Conversión de formato de datos

También puede activar la conversión de formato de datos de Kinesis Data Firehose, que convertirá su secuencia de datos JSON a Apache Parquet o Apache ORC. Esta función solo puede convertir JSON a Apache Parquet o Apache ORC. Si tiene datos en CSV, puede transformarlos mediante una función Lambda a JSON y, después, aplicar la conversión de formato de datos.

Entrega de datos

Como secuencia de entrega casi en tiempo real, Kinesis Data Firehose almacena en búfer los datos entrantes. Una vez alcanzados los umbrales de almacenamiento en búfer de la secuencia de entrega, los datos se entregan en el destino que ha configurado. Existen algunas diferencias en la forma en que Kinesis Data Firehose entrega los datos en cada destino, que este documento analiza en las siguientes secciones.

Amazon S3

Amazon S3 es un almacenamiento de objetos con una interfaz de servicios web sencilla para almacenar y recuperar el volumen de datos que desee desde cualquier ubicación de la Web. Se ha diseñado para ofrecer una durabilidad del 99,999999999 % y escalar más allá de billones de objetos a nivel mundial.

Entrega de datos a Amazon S3

Para entregar datos a Amazon S3, Kinesis Data Firehose primero concatena varios registros entrantes en función de la configuración de búfer de su secuencia de entrega y, a continuación, los entrega a Amazon S3 como un objeto de S3. La frecuencia de entrega de datos a S3 viene determinada por el tamaño del búfer de S3 (de 1 a 128 MB) o el intervalo del búfer (de 60 a 900 segundos), lo que suceda antes.

La entrega de datos al bucket de S3 podría generar errores por varias razones. Por ejemplo, es posible que el bucket ya no exista o que el rol de AWS Identity and Access Management (IAM) adoptado por Kinesis Data Firehose suponga que puede no tener acceso al bucket. En esos casos, Kinesis Data Firehose intenta realizar de nuevo la entrega durante un máximo de 24 horas hasta que se completa. El tiempo máximo de almacenamiento de datos de Kinesis Data Firehose es de 24 horas. Si, una vez transcurridas esas 24 horas, no se pueden entregar los datos, estos se pierden.

Amazon Redshift

Amazon Redshift es un almacenamiento de datos rápido y completamente administrado que permite analizar todos los datos mediante el uso de SQL estándar y las herramientas de inteligencia

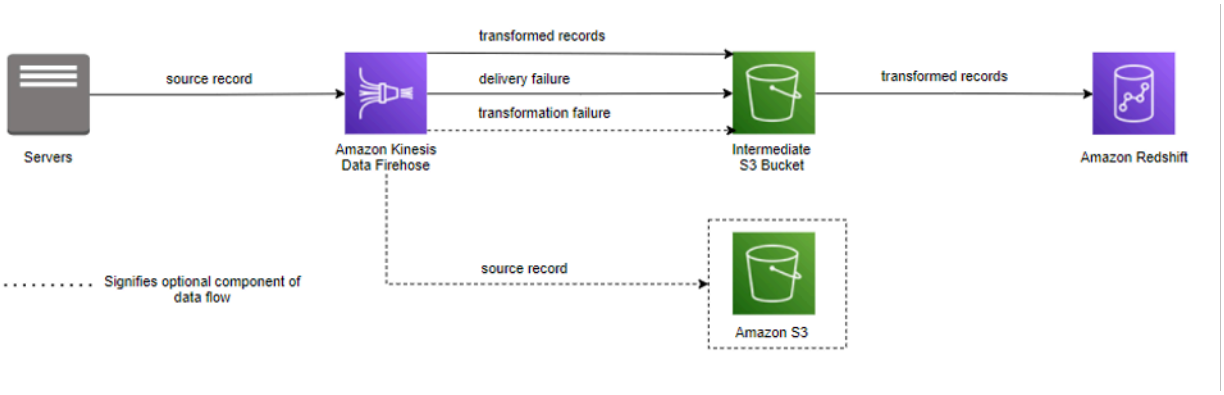
empresarial existentes de forma sencilla y rentable. Permite ejecutar consultas analíticas complejas en petabytes de datos estructurados con una sofisticada optimización de consultas, almacenamiento en columnas en discos locales de alto rendimiento y ejecución masiva de consultas paralelas.

Entrega de datos a Amazon Redshift

Para la entrega de datos a Amazon Redshift, Kinesis Data Firehose primero envía los datos entrantes a su bucket de S3 en el formato descrito anteriormente. A continuación, Kinesis Data Firehose emite un comando COPY de Amazon Redshift para cargar los datos del bucket de S3 en el clúster de Amazon Redshift.

La frecuencia con que se ejecutan las operaciones `COPY` de datos de S3 a Amazon Redshift depende de la velocidad con la que el clúster de Redshift pueda completar el comando `COPY`. Si es un destino de Amazon Redshift, puede especificar durante cuánto tiempo se reintenta la entrega (de 0 a 7200 segundos) al crear una secuencia de entrega para gestionar los errores de entrega de datos. Kinesis Data Firehose lo reintenta durante el tiempo especificado y omite ese lote concreto de objetos S3 si no lo consigue. La información de los objetos omitidos se entrega al bucket de S3 en forma de archivo de manifiesto, que puede utilizar para reposiciones manuales.

A continuación, se presenta un diagrama de arquitectura del flujo de datos de Kinesis Data Firehose a Amazon Redshift. Si bien este flujo de datos es exclusivo de Amazon Redshift, Kinesis Data Firehose sigue patrones similares para los demás objetivos de destino.



Flujo de datos de Kinesis Data Firehose a Amazon Redshift

Amazon OpenSearch Service (OpenSearch Service)

OpenSearch Service es un servicio completamente administrado que ofrece funcionalidades en tiempo real y API de OpenSearch fáciles de usar junto con la disponibilidad, la escalabilidad y la seguridad que requieren las cargas de trabajo de producción. OpenSearch Service facilita las

operaciones de implementar, usar y escalar OpenSearch para los análisis de registro, la búsqueda de texto completo y la supervisión de aplicaciones.

Entrega de datos a OpenSearch Service

Para entregar datos a OpenSearch Service, Kinesis Data Firehose primero almacena en búfer los registros entrantes a partir de la configuración de búfer de la secuencia de entrega y, después, genera una solicitud masiva de OpenSearch para indexar varios registros en el clúster de OpenSearch. La frecuencia de entrega de datos a OpenSearch Service viene determinada por los valores de tamaño del búfer de OpenSearch (de 1 a 100 MB) y de intervalo del búfer (de 60 a 900 segundos), lo que suceda antes.

Si el destino es OpenSearch Service, puede especificar durante cuánto tiempo se reintenta la entrega (de 0 a 7200 segundos) al crear la secuencia de entrega. Kinesis Data Firehose reintenta la entrega durante el tiempo especificado y, si no lo consigue, omite la solicitud de indexación. Los documentos ignorados se entregan al bucket de S3 en forma de archivo de manifiesto, en la carpeta `elasticsearch_failed/`, que puede utilizar para reposiciones manuales.

Amazon Kinesis Data Firehose puede rotar su índice de OpenSearch Service en función de una duración de tiempo. En función de la opción de rotación seleccionada (`NoRotation`, `OneHour`, `OneDay`, `OneWeek` o `OneMonth`), Kinesis Data Firehose añadirá una parte de la marca de tiempo de llegada en tiempo universal coordinado (UTC) al nombre de índice especificado.

Punto de conexión HTTP personalizado o proveedor de servicios externo compatible

Kinesis Data Firehose puede enviar datos a puntos de conexión HTTP personalizados o a proveedores externos compatibles, como Datadog, Dynatrace, LogicMonitor, MongoDB, New Relic, Splunk y Sumo Logic.

Punto de conexión HTTP personalizado o proveedor de servicios externo compatible

Para que Kinesis Data Firehose entregue datos correctamente a puntos de conexión HTTP personalizados, estos puntos de conexión deben aceptar solicitudes y enviar respuestas con ciertos formatos de solicitud y respuesta de Kinesis Data Firehose.

Al entregar datos a un punto de conexión HTTP propiedad de un proveedor de servicios externo compatible, puede usar el servicio AWS Lambda integrado para crear una función que transforme los registros entrantes al formato que coincida con el formato que espera la integración del proveedor de servicios.

Para la frecuencia de entrega de datos, cada proveedor de servicios tiene un tamaño de búfer recomendado. Colabore con su proveedor de servicios para obtener más información sobre el tamaño de búfer recomendado. Para la gestión de errores en la entrega de datos, Kinesis Data Firehose primero establece una conexión con el punto de conexión HTTP esperando una respuesta del destino. Kinesis Data Firehose sigue estableciendo la conexión hasta que caduca la duración de los reintentos. Después de eso, Kinesis Data Firehose lo considera un error de entrega de datos y realiza una copia de seguridad de los datos en el bucket de S3.

Resumen

Kinesis Data Firehose puede entregar sus datos de streaming de forma persistente a un destino admitido. Es una solución completamente administrada, que requiere poco o ningún desarrollo. Para la empresa ABC2Badge, usar Kinesis Data Firehose era la decisión lógica. Ya utilizaba Amazon Redshift como solución de almacenamiento de datos. Como sus orígenes de datos escribía continuamente en los registros de transacciones, pudo aprovechar el agente de Amazon Kinesis para transmitir esos datos sin escribir código adicional. Ahora que la empresa ABC2Badge ha creado un flujo de registros de sensores y los recibe a través de Kinesis Data Firehose, puede usarlo como base para el caso de uso del equipo de seguridad.

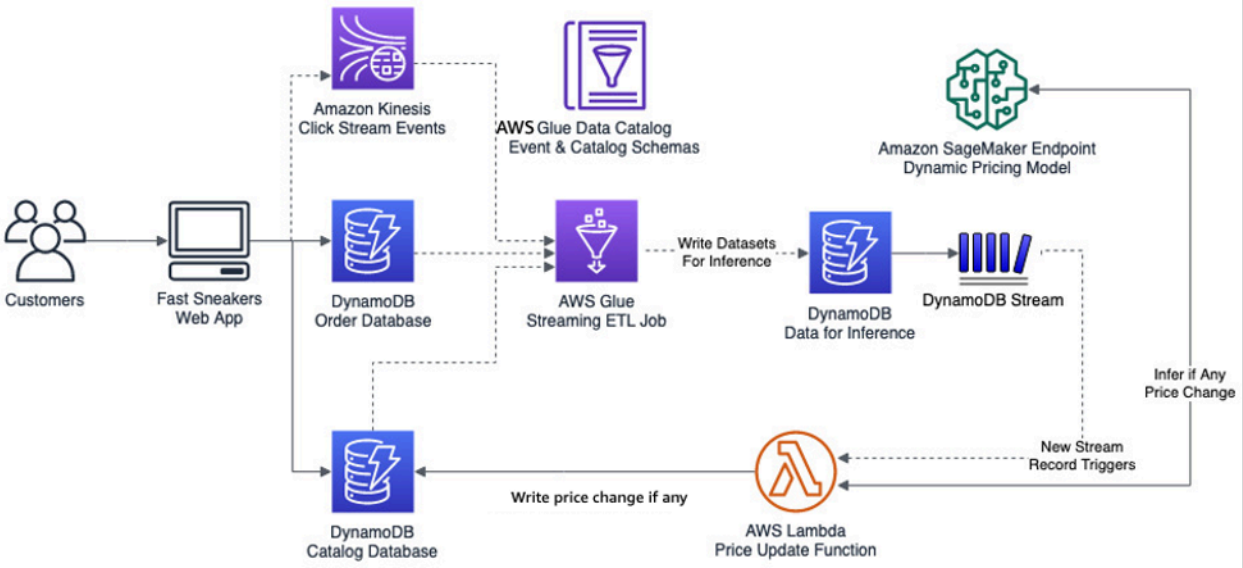
Escenario 3: preparación de los datos de secuencia de clics para procesos de análisis de datos

Fast Sneakers es una boutique de moda dedicada a zapatillas de moda. El precio de cualquier par de zapatos puede subir o bajar según el inventario y las tendencias, por ejemplo, que la noche anterior se viera a un famoso o una estrella del deporte usando zapatillas de marca en la televisión. Es importante que Fast Sneakers realice un seguimiento de esas tendencias y las analice para maximizar sus ingresos.

Fast Sneakers no quiere introducir gastos generales adicionales en el proyecto con una nueva infraestructura que mantener. Desea poder dividir el desarrollo entre las partes apropiadas, donde los ingenieros de datos puedan centrarse en la transformación de datos y los científicos de datos puedan trabajar en su funcionalidad de machine learning de forma independiente.

Para reaccionar rápidamente y ajustar los precios de forma automática de acuerdo con la demanda, Fast Sneakers transmite eventos importantes (como los datos de clics de interés y de compra), lo que transforma y aumenta los datos de evento y los envía a un modelo de machine learning. Su modelo de machine learning puede determinar si es necesario un ajuste de precios. Esto permite

a Fast Sneakers modificar automáticamente los precios para maximizar las ganancias de sus productos.



Ajustes de precios de Fast Sneakers en tiempo real

Este diagrama de arquitectura muestra la solución de streaming en tiempo real que Fast Sneakers creó con Kinesis Data Streams, AWS Glue y DynamoDB Streams. Al aprovechar estos servicios, tiene una solución que es elástica y fiable sin necesidad de dedicar tiempo a configurar ni mantener la infraestructura de apoyo. Puede dedicar el tiempo a lo que aporta valor a su empresa si se centra en un trabajo de extracción, transformación, carga (ETL) de streaming y su modelo de machine learning.

Para comprender mejor la arquitectura y las tecnologías que se usan en su carga de trabajo, a continuación se ofrecen algunos detalles de los servicios usados.

Streaming de AWS Glue y AWS Glue

AWS Glue es un servicio ETL completamente administrado que puede utilizar para catalogar los datos, limpiarlos, completarlos y trasladarlos de manera fiable de un almacén de datos a otro. Con AWS Glue, puede reducir significativamente el coste, la complejidad y el tiempo dedicado a crear trabajos ETL. AWS Glue no tiene servidor, por lo que no hay infraestructura que configurar ni administrar. Solo paga por los recursos utilizados mientras se ejecutan sus trabajos.

Al utilizar AWS Glue, puede crear una aplicación de consumidor con un trabajo ETL de streaming de AWS Glue. Esto le permite utilizar la escritura de Apache Spark y otros módulos basados en Spark para consumir y procesar los datos de los eventos. La siguiente sección de este documento profundiza en este escenario.

AWS Glue Data Catalog

AWS Glue Data Catalog contiene referencias a datos que se usan como orígenes y destinos de sus trabajos ETL en AWS Glue. AWS Glue Data Catalog es un índice para las métricas de ubicación, esquema y tiempo de ejecución de sus datos. Puede usar la información de Data Catalog para crear y supervisar sus trabajos ETL. La información de Data Catalog se almacena como tablas de metadatos en las que cada tabla especifica un único almacén de datos. Al configurar un rastreador, puede evaluar automáticamente numerosos tipos de almacenes de datos, incluidos los almacenes conectados de DynamoDB, S3 y Java Database Connectivity (JDBC), extraer metadatos y esquemas, y, a continuación, crear definiciones de tablas en AWS Glue Data Catalog.

Para trabajar con Amazon Kinesis Data Streams en trabajos ETL de streaming de AWS Glue, una práctica recomendada consiste en definir la secuencia en una tabla de una base de datos de AWS Glue Data Catalog. Defina una tabla de origen de secuencia con la secuencia de Kinesis, uno de los muchos formatos admitidos (CSV, JSON, ORC, Parquet, Avro o un formato de cliente con Grok). Puede introducir un esquema manualmente o puede dejar este paso a su trabajo de AWS Glue para que lo determine durante el tiempo de ejecución del trabajo.

Trabajo de ETL de streaming de AWS Glue

AWS Glue ejecuta los trabajos de ETL en un entorno Apache Spark sin servidor. AWS Glue ejecuta estos trabajos en recursos virtuales que aprovisiona y gestiona en su propia cuenta de servicio. Además de poder ejecutar trabajos basados en Apache Spark, AWS Glue proporciona un nivel adicional de funcionalidad en Spark con DynamicFrames.

Los DynamicFrames son tablas distribuidas que admiten datos anidados, como estructuras y matrices. Cada registro se autodescribe y está diseñado para flexibilidad de esquemas con datos semiestructurados. Un registro en un `DynamicFrame` contiene tanto los datos como el esquema que describe los datos. Apache Spark `DataFrames` y `DynamicFrames` son compatibles con los scripts de ETL, y puede convertirlos de uno a otro. Los `DynamicFrames` proporcionan un conjunto de transformaciones avanzadas para la limpieza de datos y ETL.

Al utilizar Spark Streaming en su trabajo de AWS Glue, puede crear trabajos de ETL de streaming que se ejecuten de forma continua y consuman datos de orígenes de streaming como Amazon Kinesis Data Streams, Apache Kafka y Amazon MSK. Los trabajos pueden limpiar, combinar y transformar los datos y, a continuación, cargar los resultados en los almacenes, incluidos los almacenes de datos de Amazon S3, Amazon DynamoDB o JDBC.

AWS Glue procesa y escribe datos en periodos de 100 segundos de forma predeterminada. Esto permite que los datos se procesen de forma eficiente y que las agregaciones se realicen en los datos que lleguen más tarde de lo previsto. Puede configurar el tamaño de la ventana si lo ajuste para adaptarse a la velocidad de respuesta en comparación con la precisión de su agregación. Los trabajos de streaming de AWS Glue utilizan puntos de control para realizar un seguimiento de los datos que se han leído en Kinesis Data Stream. Para obtener una guía sobre la creación de un trabajo ETL de streaming en AWS Glue, puede consultar Agregar trabajos ETL de streaming en AWS Glue

Amazon DynamoDB

Amazon DynamoDB es una base de datos de clave-valor y documentos que ofrece un rendimiento de milisegundos de un solo dígito a cualquier escala. Se trata de una base de datos completamente administrada, de varias regiones, multiactiva y duradera, con seguridad integrada, copia de seguridad y restauración, así como almacenamiento en caché en memoria para aplicaciones a escala de Internet. DynamoDB puede gestionar más de 10 billones de solicitudes por día y puede admitir picos de más de 20 millones de solicitudes por segundo.

Captura de datos de cambios para DynamoDB Streams

Una secuencia de DynamoDB es un flujo ordenado de información sobre los cambios que se realizan en los elementos de una tabla de DynamoDB. Cuando se habilita una secuencia en una tabla, DynamoDB obtiene información sobre cada modificación de los elementos de datos de esa tabla. DynamoDB se integra con AWS Lambda para que pueda crear desencadenadores, fragmentos de código que responden automáticamente a los eventos en secuencias de DynamoDB. Con los desencadenadores, puede crear aplicaciones que reaccionen ante las modificaciones de datos en las tablas de DynamoDB.

Cuando se habilita una secuencia en una tabla, puede asociar el nombre de recurso de Amazon (ARN) de la secuencia con una función que haya escrito. Inmediatamente después de modificar un elemento de la tabla, aparece un nuevo registro en el flujo de la tabla. AWS Lambda sondea el flujo e invoca a la función Lambda de forma sincrónica cuando detecta nuevos registros de flujo.

Amazon SageMaker y puntos de conexión de servicio de Amazon SageMaker

Amazon SageMaker es una plataforma completamente administrada que permite a los desarrolladores y científicos de datos crear, entrenar e implementar modelos de machine learning de

forma rápida y a cualquier escala. SageMaker incluye módulos que se pueden utilizar en conjunto o de manera independiente para crear, entrenar e implementar modelos de machine learning. Con los puntos de conexión de servicio de Amazon SageMaker, puede crear puntos de conexión alojados administrados para la inferencia en tiempo real con un modelo implementado que haya desarrollado dentro o fuera de Amazon SageMaker.

Al utilizar AWS SDK, puede invocar un punto de conexión de SageMaker que pase información de tipo de contenido junto con el contenido y, a continuación, recibir predicciones en tiempo real basadas en los datos pasados. Esto le permite mantener el diseño y el desarrollo de sus modelos de machine learning independientes del código que realiza acciones en los resultados inferidos.

De este modo, los científicos de datos se pueden centrar en el machine learning y los desarrolladores que utilizan el modelo de aprendizaje automático, en cómo lo usan en su código. Para obtener más información sobre cómo invocar un punto de conexión en SageMaker, consulte InvokeEnpoint en la referencia de la API de Amazon SageMaker.

Inferir información de datos en tiempo real

El diagrama de arquitectura anterior muestra que la aplicación web existente de Fast Sneakers ha agregado una secuencia de datos de Kinesis que contiene eventos de secuencias de clics, lo que proporciona datos de tráfico y eventos del sitio web. El catálogo de productos, que contiene información como la categorización, los atributos de producto y los precios, y la tabla de pedidos, que contiene datos como los artículos solicitados, la facturación, el envío, etc., son tablas de DynamoDB independientes. El origen de la secuencia de datos y las tablas apropiadas de DynamoDB tienen sus metadatos y esquemas definidos en AWS Glue Data Catalog para que los utilice el trabajo ETL de streaming de AWS Glue.

Al utilizar Apache Spark, Spark Streaming y `DynamicFrames` en su trabajo ETL de streaming de AWS Glue, Fast Sneakers puede extraer datos de la secuencia de datos y transformarlos, mediante la combinación de los datos de las tablas de productos y pedidos. Con los datos obtenidos de la transformación, los conjuntos de datos de los que se obtienen los resultados de inferencia se envían a una tabla de DynamoDB.

La secuencia de DynamoDB para la tabla desencadena una función Lambda para cada registro nuevo escrito. La función Lambda envía los registros transformados previamente a un punto de conexión de SageMaker con AWS SDK para inferir qué ajustes de precios, si los hubiera, son necesarios para un producto. Si el modelo de machine learning identifica que se requiere un ajuste del precio, la función Lambda escribe el cambio de precio en el producto en la tabla de DynamoDB del catálogo.

Resumen

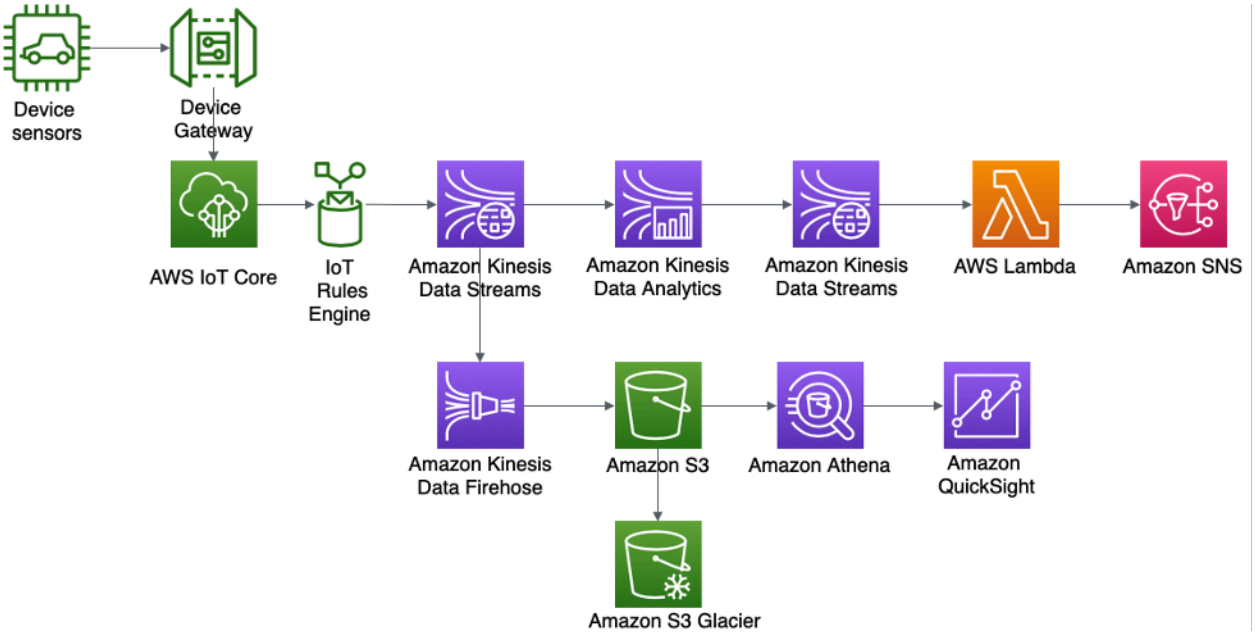
Amazon Kinesis Data Streams facilita la recopilación, el procesamiento y el análisis de datos de streaming generados en tiempo real para obtener información nueva, a tiempo y tomar acciones rápidamente basándose en ella. En combinación con el servicio de integración de datos sin servidor de AWS Glue, puede crear aplicaciones de secuencias de eventos en tiempo real que preparen y combinen datos para el machine learning.

Ya que tanto Kinesis Data Streams como los servicios de AWS Glue están completamente administrados, AWS elimina el trabajo pesado e indiferenciado de administrar la infraestructura para su plataforma de macrodatos, lo que le permite centrarse en generar análisis basados en sus datos.

Fast Sneakers puede utilizar el procesamiento de eventos en tiempo real y el machine learning para permitir que su sitio web realice ajustes de precios en tiempo real totalmente automatizados, para maximizar su stock de productos. Esto aporta el mayor valor a su empresa y evita la necesidad de crear y mantener una plataforma de macrodatos.

Escenario 4: detección de anomalías y notificaciones en tiempo real de sensores de dispositivo

La empresa ABC4Logistics transporta productos petrolíferos muy inflamables como gasolina, propano líquido (GLP) y nafta desde el puerto a varias ciudades. Hay centenares de vehículos que tienen varios sensores instalados para supervisar aspectos como la ubicación, la temperatura del motor, la temperatura en el contenedor, la velocidad de conducción, la ubicación en el estacionamiento, las condiciones de la carretera, etc. Uno de los requisitos que tiene ABC4Logistics es supervisar las temperaturas del motor y del contenedor en tiempo real y avisar al conductor y al equipo de supervisión de la flota en caso de cualquier anomalía. Para detectar dichas condiciones y generar alertas en tiempo real, ABC4Logistics implementó la siguiente arquitectura en AWS.



Arquitectura de notificaciones y detección de anomalías en tiempo real de sensores de dispositivo de ABC4Logistics

AWS IoT Gateway incorpora los datos de los sensores de dispositivo, donde el motor dereglas de AWS IoT hará que los datos de streaming estén disponibles en Amazon Kinesis Data Streams. Con Kinesis Data Analytics, ABC4Logistics puede realizar los análisis en tiempo real de los datos de streaming en Kinesis Data Streams.

Con Kinesis Data Analytics, ABC4Logistics puede detectar si las lecturas de temperatura de los sensores se desvían de las lecturas normales durante un período de diez segundos e incorporar el registro en otra instancia de Kinesis Data Streams, por lo que se pueden identificar los registros anómalos. Después, Amazon Kinesis Data Streams invoca las funciones Lambda, que pueden enviar las alertas al conductor y al equipo de supervisión de flotas a través de Amazon SNS.

Los datos de Kinesis Data Streams también se envían a Amazon Kinesis Data Firehose. Amazon Kinesis Data Firehose conserva estos datos en Amazon S3, lo que permite a ABC4Logistics realizar análisis por lotes o casi en tiempo real de los datos de sensor. ABC4Logistics utiliza Amazon Athena para consultar datos en S3 y Amazon QuickSight para las visualizaciones. Para la retención de datos a largo plazo, se utiliza la política de ciclo de vida de S3 para archivar datos en Amazon S3 Glacier.

A continuación, se detallan los componentes importantes de esta arquitectura.

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics le permite transformar y analizar los datos de streaming y responder a las anomalías en tiempo real. Es un servicio sin servidor en AWS, lo que significa que Kinesis Data Analytics se ocupa del aprovisionamiento y escala elásticamente toda la infraestructura para gestionar cualquier rendimiento de datos. De este modo, se elimina el trabajo pesado indiferenciado de configurar y administrar la infraestructura de streaming, y le permite dedicar más tiempo a escribir aplicaciones de streaming.

Con Amazon Kinesis Data Analytics, puede consultar datos de streaming de forma interactiva con múltiples opciones, como SQL estándar, aplicaciones de Apache Flink en Java, Python y Scala, y crear aplicaciones de Apache Beam mediante Java para analizar las secuencias de datos.

Estas opciones le proporcionan la flexibilidad de usar un enfoque específico según el nivel de complejidad de la aplicación de streaming y la compatibilidad de origen/destino. En la siguiente sección se describe la opción Kinesis Data Analytics para aplicaciones de Flink.

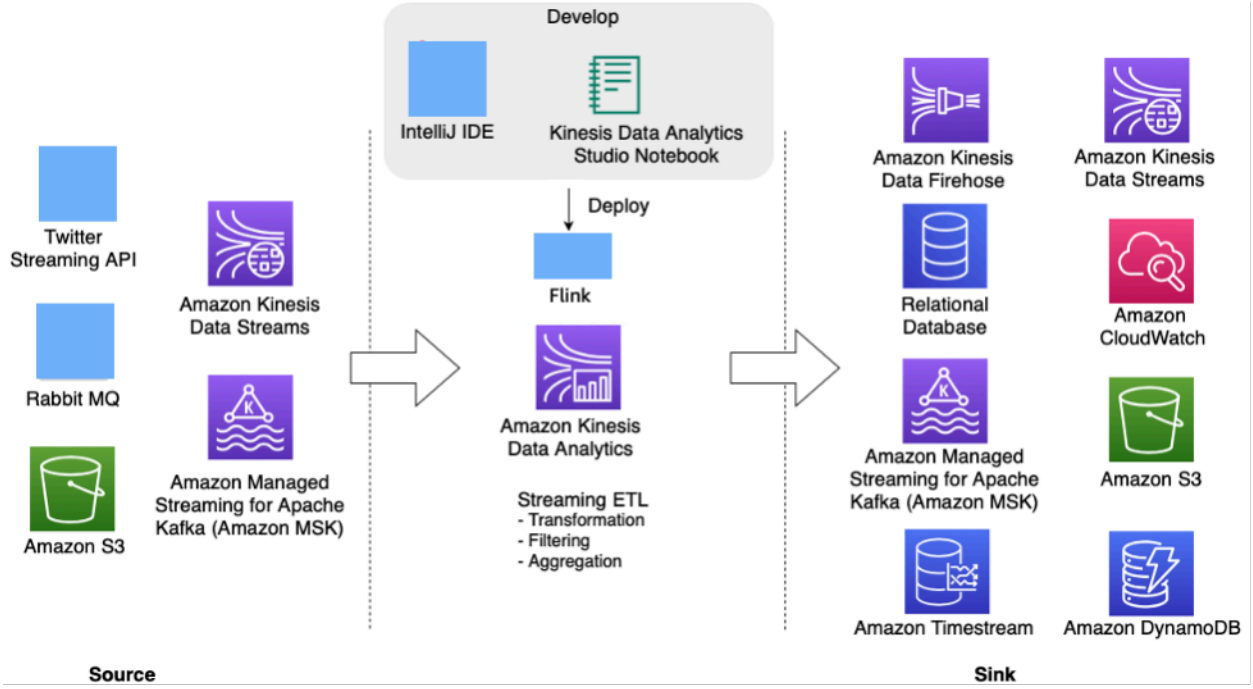
Amazon Kinesis Data Analytics para aplicaciones de Apache Flink

Apache Flink es un conocido marco de código abierto y un motor de procesamiento distribuido para cálculos con estado en secuencias de datos ilimitados y limitados. Apache Flink se ha diseñado para realizar cálculos a velocidad en memoria y a escala que admite exactamente una sola semántica. Con las aplicaciones basadas en Apache Flink se puede obtener una baja latencia con un alto rendimiento con tolerancia a errores.

Con Amazon Kinesis Data Analytics for Apache Flink, puede crear y ejecutar código en orígenes de transmisión para realizar análisis de series temporales, alimentar paneles en tiempo real y crear métricas en tiempo real sin administrar el complejo entorno distribuido de Apache Flink. Puede usar las funciones de programación de Flink de alto nivel de la misma manera que las usa cuando aloja la infraestructura de Flink por su cuenta.

Kinesis Data Analytics for Apache Flink le permite crear aplicaciones en Java, Scala, Python o SQL para procesar y analizar datos de streaming. Una aplicación Flink típica lee los datos del flujo de entrada o la ubicación o el origen de datos, transforma/filtra o une los datos mediante operadores o funciones, y almacena los datos en la secuencia de salida o la ubicación de los datos, o receptor.

El siguiente diagrama de arquitectura muestra algunas de los orígenes y los receptores compatibles para la aplicación Flink de Kinesis Data Analytics. Además de los conectores preintegrados para origen/receptor, también puede incorporar conectores personalizados a diferentes orígenes/receptores para las aplicaciones Flink en Kinesis Data Analytics.



Aplicación Apache Flink en Kinesis Data Analytics para el procesamiento de secuencias en tiempo real

Los desarrolladores pueden usar su IDE preferido para desarrollar aplicaciones Flink e implementarlas en Kinesis Data Analytics desde AWS Management Console o en herramientas de DevOps.

Amazon Kinesis Data Analytics Studio

Como parte del servicio Kinesis Data Analytics, ya está disponible Kinesis Data Analytics Studio, para que los clientes consulten de forma interactiva secuencias de datos en tiempo real y creen y ejecuten fácilmente aplicaciones de procesamiento de secuencias con SQL, Python y Scala. Los cuadernos portátiles Studio cuenta con tecnología de Apache Zeppelin.

Con el cuaderno de Studio, tiene la capacidad de desarrollar el código de la aplicación Flink en un entorno de cuadernos, ver los resultados de su código en tiempo real y visualizarlo en su cuaderno. Puede crear un cuaderno de Studio con tecnología de Apache Zeppelin y Apache Flink con un solo clic desde la consola de Kinesis Data Streams y Amazon MSK, o lanzarlo desde la consola de Kinesis Data Analytics.

Una vez que desarrolle el código de forma iterativa como parte de Kinesis Data Analytics Studio, puede implementar un cuaderno como una aplicación de análisis de datos de Kinesis para que se ejecute en modo streaming de forma continua, lea los datos de sus fuentes, escriba en sus destinos,

mantenga el estado de la aplicación de larga ejecución y escale automáticamente en función del rendimiento de las secuencias de origen. Anteriormente, los clientes utilizaban Kinesis Data Analytics for SQL Applications para realizar análisis interactivos de los datos de streaming en tiempo real en AWS.

Kinesis Data Analytics para aplicaciones SQL aún está disponible, pero para proyectos nuevos, AWS recomienda utilizar el nuevo Kinesis Data Analytics Studio. Kinesis Data Analytics Studio combina la facilidad de uso con capacidades analíticas avanzadas, lo que permite crear sofisticadas aplicaciones de procesamiento de secuencias en cuestión de minutos.

Para hacer que la aplicación de Kinesis Data Analytics Flink sea tolerante a errores, puede utilizar puntos de control e instantáneas, como se describe en Implementación de tolerancia a errores en Kinesis Data Analytics para Apache Flink.

Las aplicaciones Flink de Kinesis Data Analytics son útiles para escribir aplicaciones de análisis de streaming complejas, como aplicaciones con exactamente una sola semántica de procesamiento de datos, capacidades de puntos de control y procesamiento de datos de orígenes de datos como Kinesis Data Streams, Kinesis Data Firehose, Amazon MSK, Rabbit MQ y Apache Cassandra, incluidos los conectores personalizados.

Después de procesar los datos de streaming en la aplicación Flink, puede conservar los datos en varios receptores o destinos, como Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon DynamoDB, Amazon OpenSearch Service, Amazon Timestream, Amazon S3, etc. La aplicación Flink de Kinesis Data Analytics también ofrece garantías de rendimiento en menos de un segundo.

Aplicaciones Apache Beam para Kinesis Data Analytics

Apache Beam es un modelo de programación para procesar datos de streaming. Apache Beam proporciona una capa de API portátil para crear canalizaciones sofisticadas de procesamiento de datos en paralelo que se pueden ejecutar en distintos motores o ejecutores como Flink, Spark Streaming, Apache Samza, etc.

Puede utilizar el marco de Apache Beam con su aplicación de análisis de datos de Kinesis para procesar los datos de streaming. Las aplicaciones de Kinesis Data Analytics que usan Apache Beam utilizan el sistema de ejecución de Apache Flink para ejecutar canalizaciones Beam.

Resumen

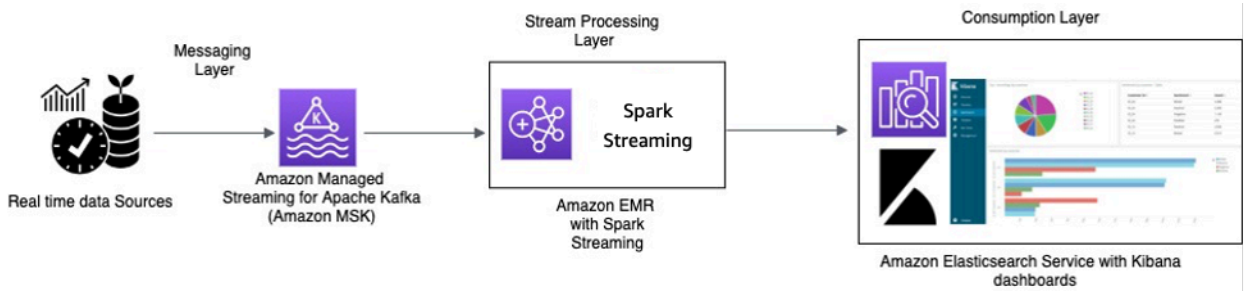
Al utilizar los servicios de streaming de AWS Amazon Kinesis Data Streams, Amazon Kinesis Data Analytics y Amazon Kinesis Data Firehose,

ABC4Logistics puede detectar patrones anómalos en las lecturas de temperatura y notificar al conductor y al equipo de administración de la flota en tiempo real, lo que evita accidentes graves como averías completas del vehículo o incendios.

Escenario 5: supervisión de datos de telemetría en tiempo real con Apache Kafka

ABC1Cabs es una empresa de servicios de reserva de taxis en línea. Todos los taxis tienen dispositivos IoT que recopilan datos de telemetría de los vehículos. Actualmente, ABC1Cabs ejecuta clústeres de Apache Kafka que están diseñados para el consumo de eventos en tiempo real, recopilan métricas de estado del sistema, realizan el seguimiento de la actividad e incorporan datos en la plataforma Apache Spark Streaming, basada en un clúster de Hadoop local.

ABC1Cabs utiliza OpenSearch Dashboards para métricas empresariales, depuración, alertas y creación de otros paneles. Tiene interés en Amazon MSK, Amazon EMR con Spark Streaming y OpenSearch Service con OpenSearch Dashboards. Su requisito es reducir la sobrecarga administrativa para mantener los clústeres de Apache Kafka y Hadoop, al tiempo que utilizan API y software de código abierto conocidos para orquestar su canalización de datos. El siguiente diagrama de arquitectura muestra su solución en AWS.



Procesamiento en tiempo real con Amazon MSK y procesamiento de secuencias mediante Apache Spark Streaming en Amazon EMR y Amazon OpenSearch Service con OpenSearch Dashboards

Los dispositivos IoT del taxi recopilan datos de telemetría y los envían a un concentrador de origen. El concentrador de origen está configurado para enviar datos en tiempo real a Amazon MSK. Con las API de la biblioteca del productor de Apache Kafka, Amazon MSK se ha configurado para transmitir

los datos en un clúster de Amazon EMR. El clúster de Amazon EMR tiene un cliente de Kafka y Spark Streaming instalados para poder consumir y procesar las secuencias de datos.

Spark Streaming tiene conectores receptores que pueden escribir datos directamente en índices definidos de Elasticsearch. Los clústeres de Elasticsearch con OpenSearch Dashboards se pueden usar para las métricas y los paneles. Amazon MSK, Amazon EMR con Spark Streaming y OpenSearch Service con OpenSearch Dashboards son servicios administrados, en los que AWS administra el pesado trabajo indiferenciado de la administración de la infraestructura de diferentes clústeres, lo que le permite crear su aplicación utilizando un software de código abierto familiar con unos pocos clics. La siguiente sección analiza en detalle estos servicios.

Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Apache Kafka es una plataforma de código abierto que permite a los clientes capturar datos de streaming como eventos de streaming de clics, transacciones, eventos de IoT y registros de aplicaciones y máquinas. Con esta información, puede desarrollar aplicaciones que realicen análisis en tiempo real, ejecuten transformaciones continuas y distribuyan estos datos a lagos de datos y bases de datos en tiempo real.

Puede usar Kafka como almacén de datos de streaming para desacoplar las aplicaciones del productor y los consumidores, y permitir una transferencia de datos fiable entre los dos componentes. Si bien Kafka es una conocida plataforma de streaming de datos y mensajería empresarial, puede resultar difícil de configurar, escalar y administrar en producción.

Amazon MSK se encarga de estas tareas de administración y facilita la preparación, la configuración y la ejecución de Kafka, junto con Apache Zookeeper, en un entorno que sigue las prácticas recomendadas para ofrecer alta disponibilidad y seguridad. Aún puede usar las operaciones del plano de control y las operaciones del plano de datos de Kafka para administrar la producción y el consumo de datos.

Como Amazon MSK ejecuta y administra Apache Kafka de código abierto, facilita a los clientes migrar y ejecutar aplicaciones Apache Kafka existentes en AWS sin necesidad de realizar cambios en su código de aplicación.

Escalado

Amazon MSK ofrece operaciones de escalado para que el usuario pueda escalar el clúster de forma activa mientras está en ejecución. Al crear un clúster de Amazon MSK, puede especificar el tipo de instancia de los agentes en el lanzamiento del clúster. Puede empezar con unos pocos agentes en

un clúster de Amazon MSK y, a continuación, con AWS Management Console o AWS CLI, puede escalar hasta cientos de agentes por clúster.

De manera alternativa, puede escalar sus clústeres al cambiar el tamaño o la familia de sus agentes de Apache Kafka. Cambiar el tamaño o la familia de los agentes proporciona la flexibilidad de ajustar la capacidad de computación de los clústeres de Amazon MSK en función de los cambios en las cargas de trabajo. Utilice la hoja de cálculo de tamaños y precios de Amazon MSK (descarga de archivo) para determinar el número correcto de agentes para su clúster de Amazon MSK. Esta hoja de cálculo proporciona una estimación del tamaño de un clúster de Amazon MSK y los costes asociados de Amazon MSK en comparación con un clúster de Apache Kafka similar, autoadministrado, basado en EC2.

Después de crear el clúster de Amazon MSK, puede aumentar la cantidad de almacenamiento de EBS por agente, con la excepción de reducir el almacenamiento. Los volúmenes de almacenamiento siguen estando disponibles durante esta operación de ampliación. Ofrece dos tipos de operaciones de escalado: automático y manual.

Amazon MSK admite la expansión automática del almacenamiento de su clúster para responder al aumento del uso mediante políticas de escalado automático de aplicaciones. La política de escalado automático establece el uso de disco de destino y la capacidad de escalado máxima.

El umbral de utilización de almacenamiento ayuda a Amazon MSK a desencadenar una operación de escalado automático. Para aumentar el almacenamiento mediante el escalado manual, espere a que el clúster se encuentre en el estado `ACTIVE`. El escalado del almacenamiento tiene un período de recuperación de al menos seis horas entre eventos. Aunque la operación hace que haya almacenamiento adicional disponible de inmediato, el servicio realiza optimizaciones en el clúster que pueden tardar 24 horas o más.

La duración de estas optimizaciones es proporcional al tamaño del almacenamiento. Además, también ofrece replicación de múltiples zonas de disponibilidad en una región de AWS para proporcionar alta disponibilidad.

Configuración

Amazon MSK ofrece una configuración predeterminada para agentes, temas y nodos de Apache Zookeeper. También puede crear configuraciones personalizadas y utilizarlas para crear nuevos clústeres de Amazon MSK o actualizar clústeres existentes. Al crear un clúster de MSK sin especificar una configuración de Amazon MSK personalizada, Amazon MSK crea y utiliza una configuración predeterminada. Para obtener una lista de valores predeterminados, consulte esta configuración de Apache Kafka.

Con fines de supervisión, Amazon MSK recopila métricas de Apache Kafka y las envía a Amazon CloudWatch, donde puede consultarlas. Las métricas que configure para su clúster de MSK se recopilan y envían automáticamente a CloudWatch. La supervisión del retraso del consumidor le permite identificar a los consumidores lentos o atascados que no se mantienen actualizados con los datos más recientes disponibles en un tema. Cuando sea necesario, puede tomar medidas correctivas, como escalar o reiniciar esos consumidores.

Migración a Amazon MSK

La migración desde un entorno local a Amazon MSK se puede lograr mediante uno de los siguientes métodos.

- MirrorMaker2.0: MirrorMaker2.0 (MM2) MM2 es un motor de replicación de datos de varios clústeres basado en el marco Apache Kafka Connect. MM2 es una combinación de un conector de origen Apache Kafka y un conector de receptor. Puede usar un único clúster MM2 para migrar datos entre varios clústeres. MM2 detecta automáticamente los nuevos temas y particiones, a la vez que garantiza que las configuraciones de los temas se sincronicen de un clúster a otro. MM2 admite migraciones de ACL, configuraciones de temas y traducción de desplazamiento. Para obtener más detalles relacionados con la migración, consulte Migración de clústeres mediante MirrorMaker de Apache Kafka. MM2 se utiliza para casos de uso relacionados con la replicación de configuraciones de temas y la traslación compensada automáticamente.
- Apache Flink: MM2 admite al menos una vez semántica. Los registros se pueden duplicar en el destino y se espera que los consumidores sean idempotentes para gestionar los registros duplicados. En escenarios de exactamente una vez, se requiere semántica y los clientes pueden usar Apache Flink. Proporciona una alternativa para lograr exactamente una sola semántica.

  Apache Flink también se puede usar para escenarios en los que los datos requieren acciones de asignación o transformación antes de enviarlos al clúster de destino. Apache Flink proporciona conectores para Apache Kafka con orígenes y receptores que pueden leer datos de un clúster de Apache Kafka y escribir en otro. Apache Flink se puede ejecutar en AWS mediante el lanzamiento de un clúster de Amazon EMR o mediante la ejecución de Apache Flink como una aplicación mediante Amazon Kinesis Data Analytics.
- AWS Lambda: con compatibilidad con Apache Kafka como origen de eventos para AWS Lambda, los clientes ahora pueden consumir mensajes de un tema mediante una función Lambda. El servicio AWS Lambda sondea internamente en busca de nuevos registros o mensajes del origen de eventos y, a continuación, invoca de forma sincrónica la función Lambda de destino para consumir estos mensajes. Lambda lee los mensajes en lotes y proporciona los lotes de mensajes

a su función en la carga del evento para su procesamiento. Los mensajes consumidos se pueden transformar o escribir directamente en el clúster de Amazon MSK de destino.

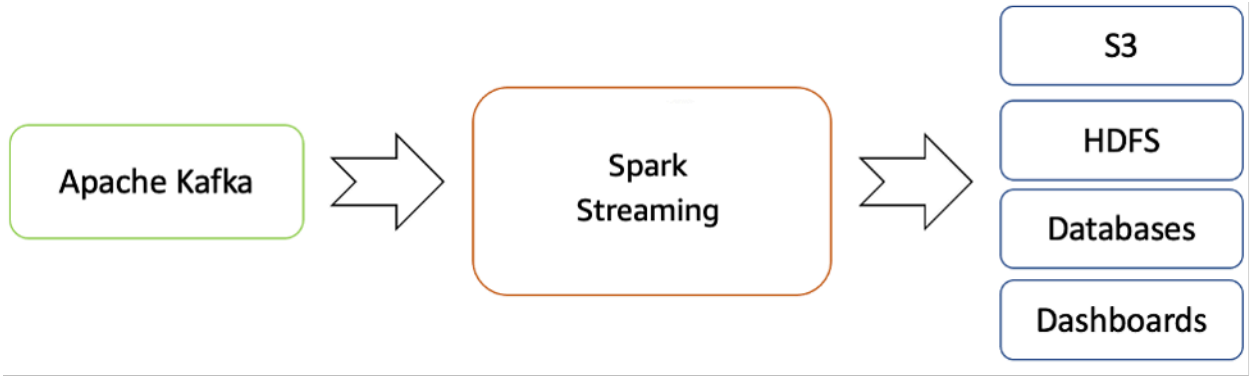
Amazon EMR con Spark Streaming

Amazon EMR es una plataforma de clúster administrado que simplifica la ejecución de los marcos de trabajo de macrodatos, tales como Apache Hadoop y Apache Spark en AWS para procesar y analizar grandes cantidades de datos.

Amazon EMR proporciona las capacidades de Spark y se puede usar para iniciar Spark Streaming para consumir datos de Kafka. Spark Streaming es una extensión de la API principal de Spark que permite el procesamiento de secuencias de datos en directo escalable, de alto rendimiento y con tolerancia a errores.

Puede crear un clúster de Amazon EMR con AWS Command Line Interface (AWS CLI) o en AWS Management Console y seleccionar Spark y Zeppelin en configuraciones avanzadas durante la creación del clúster. Como se muestra en el siguiente diagrama de arquitectura, los datos se pueden ingerir de muchas fuentes, como Apache Kafka y Kinesis Data Streams, y se pueden procesar con algoritmos complejos expresados con funciones de alto nivel, como «map», «reduce», «join» y «window». Para obtener más información, consulte Transformaciones en DStreams.

Los datos procesados se pueden enviar a sistemas de archivos, bases de datos y paneles en directo.



Flujo de streaming en tiempo real desde el ecosistema de Apache Kafka a Hadoop

De forma predeterminada, Apache Spark Streaming tiene un modelo de ejecución de microlotes. Sin embargo, desde que se lanzó Spark 2.3, Apache ha incorporado un nuevo modo de procesamiento de baja latencia llamado procesamiento continuo, que puede lograr latencias de extremo a extremo de hasta un milisegundo con garantías de «al menos una vez».

Sin cambiar las operaciones de Dataset/DataFrames en sus consultas, puede elegir el modo en función de los requisitos de su aplicación. Algunas de los beneficios de Spark Streaming son:

- Lleva la API integrada en el lenguaje de Apache Spark al procesamiento de secuencias, lo que le permite escribir trabajos de streaming de la misma manera que escribe trabajos por lotes.
- Es compatible con Java, Scala y Python.
- Puede recuperar tanto el trabajo perdido como el estado del operador (como ventanas deslizantes) desde el primer momento, sin código adicional de su parte.
- Al ejecutarse en Spark, Spark Streaming te permite reutilizar el mismo código para el procesamiento por lotes, unir transmisiones con datos históricos o ejecutar consultas ad hoc sobre el estado de la secuencia y crear aplicaciones interactivas potentes, no solo análisis.
- Después de procesar el flujo de datos con Spark Streaming, se puede usar OpenSearch Sink Connector para escribir datos en el clúster de OpenSearch Service y, a su vez, OpenSearch Service con OpenSearch Dashboards se puede usar como capa de consumo.

Amazon OpenSearch Service con OpenSearch Dashboards

OpenSearch Service es un servicio administrado que facilita las tareas de implementación, operación y escalado de clústeres OpenSearch en la nube de AWS. OpenSearch es un conocido motor de búsqueda y análisis, y de código abierto para casos de uso como análisis de registros, supervisión de aplicaciones en tiempo real y análisis de secuencias de clics.

OpenSearch Dashboards es una herramienta de código abierto de visualización y exploración de datos utilizada para el análisis de registros y series temporales, la supervisión de aplicaciones y los casos de uso de inteligencia operativa. Ofrece características potentes y fáciles de usar, como histogramas, gráficos de líneas, gráficos circulares, mapas de calor y asistencia geoespacial integrada.

OpenSearch Dashboards proporciona una estrecha integración con OpenSearch, un conocido motor de análisis y búsqueda, lo que hace que OpenSearch Dashboards sea la opción predeterminada para visualizar los datos almacenados en OpenSearch. OpenSearch Service proporciona una instalación de paneles de OpenSearch Dashboards con cada dominio de OpenSearch Service. Puede encontrar un enlace a OpenSearch Dashboards en el panel del dominio en la consola de OpenSearch Service.

Resumen

Con Apache Kafka ofrecido como un servicio administrado en AWS, puede centrarse en el consumo en lugar de en administrar la coordinación entre los agentes, lo que generalmente requiere una comprensión detallada de Apache Kafka. La plataforma Amazon MSK administra funciones como la alta disponibilidad, la escalabilidad de los agentes y el control de acceso detallado.

ABC1Cabs utilizó estos servicios para crear aplicaciones de producción sin necesidad de tener experiencia en administración de la infraestructura. Podía centrarse en la capa de procesamiento para consumir datos de Amazon MSK y propagarlos a la capa de visualización.

Spark Streaming en Amazon EMR puede ayudar a realizar análisis en tiempo real de datos de transmisión y publicar en OpenSearch Dashboards en Amazon OpenSearch Service para la capa de visualización.

Conclusión y colaboradores

Conclusión

En este documento se han analizado varios escenarios para flujos de trabajo de streaming. En estos escenarios, el procesamiento de datos de streaming proporcionó a las empresas de ejemplo la capacidad de agregar nuevas características y funcionalidades.

Al analizar los datos a medida que se crean, obtendrá información sobre lo que su empresa hace en ese momento. Los servicios de streaming de AWS le permiten centrarse en su aplicación para tomar decisiones empresariales urgentes, en lugar de implementar y administrar la infraestructura

Colaboradores

- Amalia Rabinovitch, arquitecta de soluciones sénior, AWS
- Priyanka Chaudhary, lago de datos, arquitecta de datos, AWS
- Zohair Nasimi, arquitecto de soluciones, AWS
- Rob Kuhr, arquitecto de soluciones, AWS
- Ejaz Sayyed, arquitecto de soluciones sénior para socios, AWS
- Allan MacInnis, arquitecto de soluciones, AWS
- Chander Matrubhutam, director de marketing de productos, AWS

Revisiones del documento

Para recibir notificaciones sobre las actualizaciones de este documento técnico, suscríbase a la fuente RSS.

update-history-change	update-history-description	update-history-date
Actualizado	Revisado para garantizar la precisión técnica	1 de septiembre de 2021
Publicación inicial	Documento técnico publicado por primera vez	1 de julio de 2017