

# Soluciones de datos de streaming en AWS con Amazon Kinesis

*Julio de 2017*



© 2017, Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.

## Avisos

Este documento se suministra únicamente con fines informativos. Representa la oferta actual de productos y prácticas de AWS a partir de la fecha de publicación de este documento. Dichas prácticas y productos pueden modificarse sin previo aviso. Los clientes son responsables de realizar sus propias evaluaciones independientes de la información contenida en este documento y de cualquier uso de los productos o servicios de AWS, cada uno de los cuales se ofrece “tal cual”, sin garantía de ningún tipo, ya sea explícita o implícita. Este documento no genera ninguna garantía, declaración, compromiso contractual, condición ni certeza por parte de AWS, sus filiales, proveedores o licenciantes. Las responsabilidades y obligaciones de AWS con respecto a sus clientes se controlan mediante los acuerdos de AWS y este documento no forma parte ni modifica ningún acuerdo entre AWS y sus clientes.

# Contenido

Introducción	1
Situaciones de aplicación en tiempo real	1
Diferencia entre el procesamiento de lotes y de streaming	2
Desafíos del procesamiento de streaming	2
De lotes a tiempo real: un ejemplo	3
Situación de ejemplo: notificación y facturación de carreteras de peaje	4
Requisito 1: datos más recientes en el almacén de datos	5
Amazon Kinesis Firehose	6
Requisito 2: alertas del umbral de facturación	13
Amazon Kinesis Analytics	14
Amazon Kinesis Streams	17
Requisito 3: otras alertas del umbral	23
Arquitectura completa	24
Conclusión	25
Colaboradores	26

# Resumen

Los ingenieros de datos, analistas de datos y desarrolladores de big data desean evolucionar su análisis de lotes a tiempo real para que sus empresas pueden obtener información sobre lo que están haciendo sus clientes, aplicaciones y productos, y tomar decisiones de manera oportuna. Este documento técnico analiza la evolución del análisis de lotes al de tiempo real. Describe cómo servicios como Amazon Kinesis Streams, Amazon Kinesis Firehose y Amazon Kinesis Analytics pueden utilizarse para implementar aplicaciones en tiempo real y proporciona patrones de diseño comunes a través de estos servicios.

# Introducción

Actualmente, las empresas reciben datos a una escala y velocidad masivas debido al crecimiento explosivo de los orígenes de datos que generan continuamente secuencias de datos. Independientemente de que se trate de datos de registro de servidores de aplicaciones, datos de secuencias de clics en sitios web y aplicaciones móviles o datos de telemetría en dispositivos de Internet de las cosas (IoT), todo contiene información que puede ayudarle a saber qué es lo que están haciendo sus clientes, aplicaciones y productos en este momento. Tener la capacidad de procesar y analizar estos datos en tiempo real es fundamental para realizar operaciones como monitorear continuamente sus aplicaciones con el fin de garantizar un alto nivel de tiempo de actividad y personalizar ofertas promocionales y recomendaciones de productos. El procesamiento en tiempo real también puede representar otros casos de usos comunes, como el análisis de sitios web y el aprendizaje automático de forma más precisa y factible al habilitar los datos para estas aplicaciones en solo segundos o minutos en lugar de horas o días.

## Situaciones de aplicación en tiempo real

Existen dos tipos de situaciones de casos de uso que sirven para aplicaciones de datos de streaming:

- **La evolución del análisis de lotes al análisis de streaming**

Puede realizar un análisis de datos en tiempo real que tradicionalmente se analizaban con un procesamiento en lotes en almacenes de datos o con marcos Hadoop. Los casos de uso más comunes en esta categoría incluyen “data lakes”, ciencia de datos y aprendizaje automático. Puede utilizar soluciones de datos de streaming para cargar continuamente datos en tiempo real en sus “data lakes”. También puede actualizar modelos de aprendizaje automático con mayor frecuencia a medida que se ponen a disposición datos nuevos, lo que garantiza la precisión y fiabilidad de los resultados. Por ejemplo, Zillow utiliza Amazon Kinesis Streams para recopilar datos de registros públicos y listas de MLS para luego brindarle a los compradores y vendedores la tasación más actualizada del valor de las propiedades en tiempo real. Zillow también envía los mismos datos a su “data lake” de Amazon Simple Storage Service (S3) mediante Kinesis Streams de modo que todas las aplicaciones trabajan con la información más reciente.

- **La creación de aplicaciones en tiempo real**

Puede utilizar los servicios de datos de streaming para aplicaciones en tiempo real, como el monitoreo de aplicaciones, la detección de fraude y los marcadores en directo. Estos casos de uso requieren latencias de milisegundos de principio a fin; desde la adquisición hasta el procesamiento, hasta llegar a la emisión de los resultados hasta los almacenes de datos de destino y otros sistemas. Por ejemplo, Netflix utiliza Kinesis Streams para monitorear las comunicaciones entre todas sus aplicaciones con el objetivo de detectar y corregir errores rápidamente, lo que garantiza un muy buen tiempo de actividad y disponibilidad para sus clientes. Si bien el caso de uso que aplica con más frecuencia es el monitoreo del desempeño de la aplicación, hay una cantidad cada vez mayor de aplicaciones en tiempo real en la tecnología publicitaria, los juegos y la IoT comprendidas en esta categoría.

## Diferencia entre el procesamiento de lotes y de streaming

Necesita un conjunto diferente de herramientas para recopilar, preparar y procesar datos de streaming en tiempo real que las herramientas que tradicionalmente utiliza para el análisis de lotes. Con el análisis tradicional, recopila los datos, los carga de forma periódica a una base de datos y los analizará horas, días o semanas más tarde. Analizar los datos en tiempo real requiere un enfoque diferente. En lugar de ejecutar consultas a la base de datos sobre los datos almacenados, las aplicaciones de procesamiento de streaming procesan datos constantemente en tiempo real, incluso antes de que se almacenen. Los datos de streaming pueden presentarse a un ritmo veloz y los volúmenes de datos pueden aumentar o disminuir en cualquier momento. Las plataformas de procesamiento de secuencias de datos tienen que ser capaces de gestionar la velocidad y la variabilidad de los datos entrantes y procesarlos a medida que llegan, a menudo van de millones a cientos de millones de eventos por hora.

## Desafíos del procesamiento de streaming

Procesar datos en tiempo real a medida que llegan puede permitirle tomar decisiones mucho más rápido que con las tecnologías de análisis de datos tradicionales. Sin embargo, crear y utilizar sus propias canalizaciones de datos de streaming personalizadas es complicado y conlleva muchos recursos. Debe crear un sistema que pueda recopilar, preparar y transmitir de manera rentable los datos que vienen en simultáneo desde miles de orígenes de datos. Debe ajustar el



almacenamiento y calcular los recursos de modo que los datos se colocan en lotes y se transmiten de forma eficaz para un máximo rendimiento y una baja latencia. Tiene que implementar y administrar una flota de servidores para escalar el sistema para que pueda gestionar las velocidades variables de datos que va a presentarle. Una vez que haya creado esta plataforma, tiene que monitorear el sistema y recuperarse de cualquier error de servidor o red al ponerse al día con el procesamiento desde el punto adecuado de la secuencia, sin crear datos duplicados. Todo esto consume valioso tiempo y dinero y, a fin de cuentas, la mayoría de las empresas nunca llegan allí y deben conformarse con el status quo y administrar sus negocios con información que tiene horas o días de antigüedad.

## De lotes a tiempo real: un ejemplo

Para comprender mejor cómo están evolucionando las organizaciones desde un procesamiento de lotes a uno de streaming con AWS, analicemos un ejemplo. En este ejemplo, analizaremos una situación y analizaremos en detalle de qué manera los servicios de AWS [Amazon Kinesis Streams](#),<sup>1</sup> [Amazon Kinesis Firehose](#)<sup>2</sup> y [Amazon Kinesis Analytics](#)<sup>3</sup> se utilizan para resolver el problema.

El procesamiento de lotes es una práctica común para el procesamiento de datos. Las organizaciones suelen realizar los trabajos habituales para analizar sus datos con una frecuencia aplicable a su caso de uso. Por ejemplo, una organización podría ejecutar un proceso a fin de mes para determinar cuánto facturarle a cada uno de sus clientes. O bien, podrían realizar un trabajo por hora para analizar los registros de sus aplicaciones de TI para determinar qué errores se produjeron en la última hora. Si bien estos procesos por mes o por hora son valiosos, ¿qué sucede si se podrían analizar los mismos datos a medida que se crean? ¿Hay información adicional que se podría recopilar o valor adicional que se podría crear?

Considere nuevamente la situación de facturación mensual. Al analizar los datos de uso de un cliente a medida que se generan, una organización puede habilitar funciones valiosas como notificar a los usuarios que se aproximan al límite de facturación predefinido. Si los registros de aplicaciones de TI se pueden analizar en tiempo real, un administrador de sistema puede recibir una notificación inmediata para investigar y emprender medidas correctivas.

Ahora combinemos estos dos en una única situación y analicemos cómo podemos crear una solución.



## Situación de ejemplo: notificación y facturación de carreteras de peaje

En este ejemplo simplificado, una empresa ficticia, ABC Tolls, opera autopistas de peaje en todo el país. Los clientes que se registran en ABC Tolls reciben un transceptor para su automóvil. Cuando el cliente conduce por el área de peaje, un sensor recibe información del transceptor y registra los detalles de la transacción en una base de datos relacional. ABC Tolls tiene una arquitectura de lote tradicional. Todos los días se ejecuta un proceso programado de extracción, transformación y carga (ETL) que procesa las transacciones diarias y las transforma para que se puedan cargar en su almacén de datos. Al día siguiente, los analistas comerciales de ABC Tolls revisan los datos a través de una herramienta de generación de informes. Además, una vez al mes (al final del ciclo de facturación) otro proceso agrega todas las transacciones para que cada uno de los clientes de ABC Tolls calcule su pago mensual.

ABC Tolls quisiera realizar algunas modificaciones a su sistema. El primer requisito proviene de su equipo de analistas comerciales. Han solicitado la capacidad de ejecutar informes desde su almacén de datos con datos de no más de 30 minutos de antigüedad.

ABC Tolls también está desarrollando una nueva aplicación móvil para sus clientes. Mientras desarrollaban la aplicación, decidieron crear algunas funciones nuevas. Una función le brinda a los clientes la capacidad de establecer un umbral de gasto para su cuenta. Si una factura de peaje acumulado del cliente supera este umbral, ABC Tolls envía un mensaje dentro de la aplicación para informarle al cliente que se ha incumplido el umbral 10 minutos después de que esto suceda.

Por último, el equipo de operaciones de ABC Tolls tiene algunos requisitos adicionales que quieren introducir en el sistema. Mientras monitorean sus estaciones de peaje, quieren recibir una notificación de inmediato cuando el tráfico de vehículos en una estación de peaje cae por debajo de un umbral predefinido para cada período de 30 minutos en un día. Por ejemplo, saben que, a partir de los datos históricos, una de sus estaciones de peaje ve aproximadamente 360 vehículos los miércoles entre las 2:00 p. m. y las 2:30 p. m. En ese período de 30 minutos, el equipo de operaciones quiere recibir una notificación si la estación de peaje ve menos de 100 vehículos. Sus operadores pueden investigar para determinar si el tráfico es normal o si algún otro factor ha contribuido al valor inesperado (por ejemplo, un sensor defectuoso o un accidente de automóvil en la autopista).

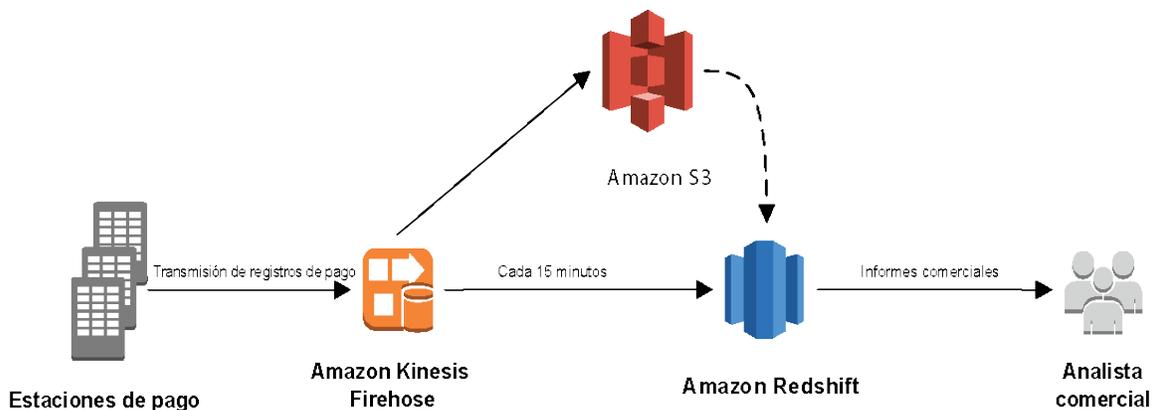


El equipo de ingeniería de ABC Tolls determina que su arquitectura actual necesita algunas modificaciones para admitir estos requisitos. Deciden crear una adquisición de datos de streaming y un sistema de análisis para admitir los requisitos. Analicemos cada requisito y echemos un vistazo a las mejoras en la arquitectura que admitirá cada uno de ellos.

## Requisito 1: datos más recientes en el almacén de datos

En la actualidad, los datos del almacén de datos de ABC Tolls pueden tener hasta 24 horas debido a su proceso por lotes diario. Su solución de almacén de datos actual es Amazon Redshift. Al revisar las funciones de los servicios de Amazon Kinesis, reconocieron que Kinesis Firehose puede recibir una secuencia de registros de datos e insertarlas en Amazon Redshift. Crearon una secuencia de entrega de Kinesis Firehose y la configuraron para que pueda copiar datos a su tabla de Amazon Redshift cada 15 minutos. Su solución actual almacena registros en un sistema de archivos como parte de su proceso por lotes. Como parte de esta nueva solución, utilizaron Amazon Kinesis Agent en sus servidores para reenviar los datos de registro a Kinesis Firehose. Dado que Kinesis Firehose utiliza Amazon S3 para almacenar datos de streaming sin procesar antes de que se copien en Amazon Redshift, ABC Tolls no tuvo que crear otra solución para archivar sus datos sin procesar.

La figura 1 muestra esta solución.



**Figura 1: nueva solución con Amazon Kinesis Firehose**

Para esta parte de la arquitectura, ABC Tolls eligió Kinesis Firehose. Analicemos las funciones de Kinesis Firehose en detalle.



## Amazon Kinesis Firehose

Amazon Kinesis Firehose es la forma más fácil de cargar datos de streaming en AWS. Puede capturar, transformar y cargar datos de streaming en Amazon Kinesis Analytics, Amazon S3, Amazon Redshift y Amazon Elasticsearch Service, lo que permite realizar un análisis casi en tiempo real con las herramientas disponibles de inteligencia empresarial y los paneles que usa actualmente. Se trata de un servicio totalmente administrado que se escala automáticamente para adaptarse al rendimiento de los datos y no precisa de ninguna administración permanente. También 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.

Kinesis Firehose es un servicio completamente administrado. No es necesario escribir aplicaciones ni administrar recursos. Configura los productores de datos para que envíen datos a Kinesis Firehose, que a su vez los envía automáticamente al destino que especifique. También puede configurar Kinesis Firehose para transformar los datos antes de entregarlos.

### Envío de datos a una secuencia de entrega de Amazon Kinesis Firehose

Existen varias opciones para enviar datos a su secuencia de entrega. AWS ofrece SDK para varios idiomas de programación conocidos, cada uno de los cuales proporciona las API para Kinesis Firehose. AWS también ha creado una utilidad para ayudar a enviar datos a su secuencia de entrega.

#### ***Uso de la API***

La API de Kinesis Firehose ofrece dos operaciones para enviar datos a la secuencia de entrega. `PutRecord` envía un registro de datos dentro de una llamada. `PutRecordBatch` puede enviar varios registros de datos dentro de una llamada.

En cada método, debe especificar el nombre de la secuencia de entrega y el registro de datos, o la variedad de registros de datos, cuando se utiliza el método. Cada registro de datos se compone de un blob de datos que puede ser de hasta 1000 KB en tamaño y cualquier tipo de datos.

Para obtener información detallada y código de muestra para las operaciones de la API de Kinesis Firehose, consulte [Cómo escribir en una secuencia de entrega Firehose con AWS SDK](#)<sup>4</sup>.



### ***Uso de Amazon Kinesis Agent***

Amazon Kinesis Agent es una aplicación de software Java independiente que ofrece una forma sencilla de recopilar y enviar datos a Kinesis Streams y Kinesis Firehose. El agente monitorea constantemente un conjunto de archivos y envía nuevos datos a su secuencia. El agente se encarga de rotar los archivos, crear puntos de comprobación e intentarlo nuevamente cuando se producen errores. Entrega todos sus datos de manera fiable, oportuna y sencilla. También emite las métricas de Amazon CloudWatch para ayudarle a monitorear y solucionar mejor los problemas que surjan en el proceso de streaming.

Puede instalar el agente en entornos de servidor basados en Linux, como servidores web, de registro o de base de datos. Después de instalar el agente, configúrelo especificando los archivos que desea monitorear y la secuencia de destino de los datos. Una vez configurado, el agente recopila datos de los archivos de forma duradera y los envía de forma fiable a la secuencia de entrega.

El agente puede monitorear varios directorios de archivos y escribir en varias secuencias. También puede configurarse para procesar previamente los registros de datos antes de que se envíen a su secuencia o secuencia de entrega.

Si está considerando una migración desde un sistema de archivos de lote tradicional hacia los datos de streaming, es posible que sus aplicaciones ya estén registrando eventos para los archivos en los sistemas de archivos de los servidores de aplicaciones. O bien, si la aplicación utiliza una biblioteca de registro conocida (como Log4j), suele ser una tarea sencilla configurarla para escribir en archivos locales. Independientemente de la forma en que los datos se escriben en un archivo de registro, debe tener en cuenta el uso del agente en esta situación. Proporciona una solución sencilla que requiere cambios mínimos o ninguno en su sistema existente. En muchos casos, se puede utilizar de forma simultánea con su solución de lote existente. En este caso, proporciona una secuencia de datos en Kinesis Streams, mediante los archivos de registro como origen de datos para la secuencia.

En nuestra situación de ejemplo, ABC Tolls eligió utilizar el agente para enviar datos de streaming a su secuencia de entrega. Ya estaban creando archivos de registro, por lo que reenviar las entradas de registro a Kinesis Firehose fue una instalación y configuración sencillas del agente. No se necesitó ningún código adicional para comenzar a transmitir sus datos.

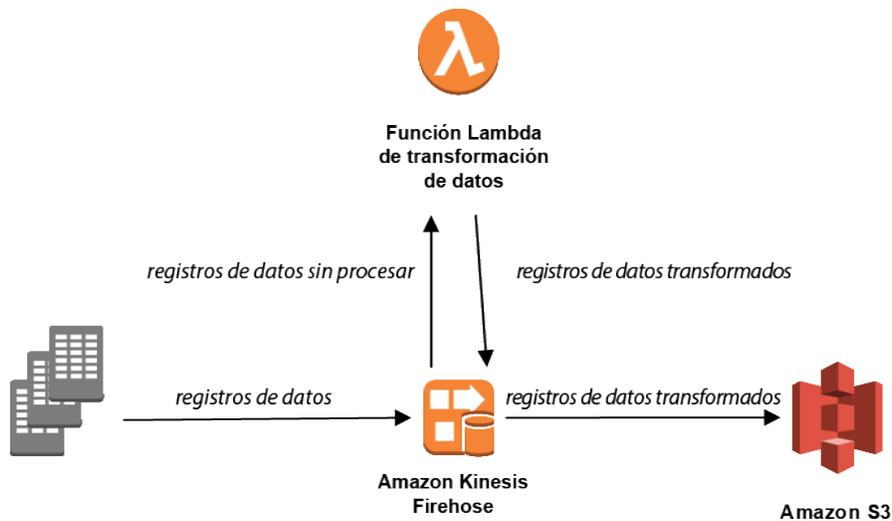
## Transformación de datos

En algunos casos, puede ser conveniente transformar o mejorar los datos de streaming antes de que se entreguen a su destino. Por ejemplo, los productores de datos podrían enviar texto no estructurado en cada registro de datos, y usted necesita transformarlos a JSON antes de entregarlos a Amazon Elasticsearch Service.

Para habilitar las transformaciones de datos de streaming, Kinesis Firehose utiliza una función de [AWS Lambda](#) que crea para transformar sus datos.<sup>5</sup>

### **Flujo de transformación de datos**

Al habilitar la transformación de datos de Kinesis Firehose, Kinesis Firehose almacena datos de entrada de hasta 3 MB o el tamaño de almacenamiento en búfer especificado para la secuencia de entrega, lo que sea menor. Kinesis Firehose luego invoca la función de Lambda especificada con cada lote almacenado en búfer de forma asíncrona. Los datos transformados se envían desde Lambda hacia Kinesis Firehose para almacenarlos en búfer. Los datos transformados se envían en sus destinos tan pronto se alcanza el tamaño o el intervalo de almacenamiento en búfer, lo que ocurra primero. La figura 2 describe este proceso para una secuencia de entrega que envía datos en Amazon S3.



**Figura 2: datos almacenados en búfer con Kinesis Firehose y funciones de Lambda**

## Envío de datos

Una vez que se hayan alcanzado los umbrales de almacenamiento en búfer de la secuencia de entrega, sus datos se envían al destino que ha configurado. Existen algunas diferencias en la forma en que Kinesis Firehose envía datos a cada destino, y eso analizaremos en las siguientes secciones.

### **Amazon Simple Storage Service**

[Amazon S3](#) es un almacenamiento de objetos con una interfaz de servicios web sencilla para almacenar y recuperar cualquier cantidad de datos, desde cualquier parte de la web.<sup>6</sup> Se ha diseñado para proporcionar una durabilidad del 99,999999999 % y escalar billones de objetos a nivel mundial. Puede usar Amazon S3 como almacenamiento principal para las aplicaciones en la nube nativas, como un repositorio masivo o “data lake” para los análisis y como un destino de backup y restauración y recuperación ante desastres.

#### Formato del envío de datos

Para enviar datos a Amazon S3, Kinesis Firehose concatena múltiples registros de entrada a partir de la configuración del búfer de la secuencia de entrega y luego los envía a Amazon S3 como un objeto de S3. Es posible que desee añadir un separador de registros al final de cada uno de los registros antes de enviarlos a Kinesis Firehose, de modo que puede dividir un objeto de S3 enviado a los registros individuales.

#### Frecuencia del envío de datos

La frecuencia de envío de datos en Amazon S3 se determina en función del valor que haya configurado para el tamaño y el intervalo del búfer de S3 para su secuencia de entrega. Kinesis Firehose almacena en búfer los datos de entrada antes de enviarlos a Amazon S3. Puede configurar los valores para el tamaño del búfer de Amazon S3 (de 1 MB a 128 MB) o para el intervalo del búfer (de 60 segundos a 900 segundos). La primera condición que se cumpla desencadena el envío de datos a Amazon S3. Tenga en cuenta que si el envío de datos a destino no alcanza la velocidad de escritura de datos en la secuencia de entrega, Kinesis Firehose incrementa el tamaño del búfer de forma dinámica para alcanzarla y se asegura de que todos los datos lleguen a destino.

#### Flujo de datos

En la figura 3 se muestra el flujo de datos para los destinos de Amazon S3.



**Figura 3: flujo de datos desde Kinesis Firehose hasta los buckets de S3**

### **Amazon Redshift**

[Amazon Redshift](#) es un almacén de datos rápido y totalmente administrado que permite analizar todos los datos empleando de forma sencilla y rentable SQL estándar y las herramientas de inteligencia empresarial (BI) existentes.<sup>7</sup> Permite ejecutar consultas analíticas complejas en petabytes de datos estructurados utilizando una sofisticada optimización de consultas, almacenamiento en columnas en discos locales de alto desempeño y ejecución masiva de consultas paralelas. La mayoría de los resultados se producen en segundos.

En nuestro ejemplo, ABC Tolls ya estaba utilizando Amazon Redshift como su solución de almacén de datos. Cuando implementaron la solución de datos de streaming, configuraron su secuencia de entrega para enviar sus datos de streaming a su clúster de Amazon Redshift existente.

#### Formato del envío de datos

Para el envío de datos a Amazon Redshift, Kinesis Firehose primero envía los datos entrantes al bucket de S3 en el formato descrito anteriormente. Kinesis Firehose luego emite un comando COPY de Amazon Redshift para cargar los datos desde el bucket de S3 a su clúster de Amazon Redshift. Debe asegurarse de que, después de que Kinesis Firehose concatena varios registros de entrada en un objeto de S3, este se pueda copiar en su clúster de Amazon Redshift. Para obtener más información, consulte los [parámetros del formato de datos del comando COPY de Amazon Redshift](#).

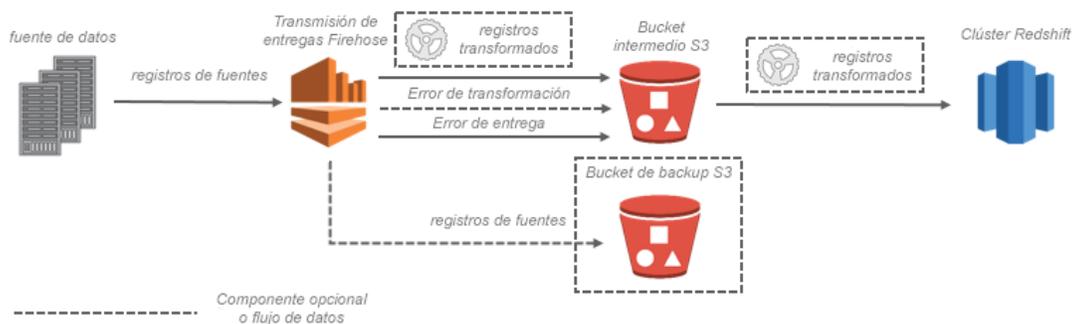


Frecuencia del envío de datos

La frecuencia de las operaciones COPY de datos desde Amazon S3 hacia Amazon Redshift se determina según la velocidad a la que su clúster de Amazon Redshift puede finalizar el comando COPY. Si quedan datos por copiar, Kinesis Firehose emite un nuevo comando COPY tan pronto como Amazon Redshift termine correctamente el comando COPY anterior.

Flujo de datos

La figura 4 muestra el flujo de datos para los destinos de Amazon Redshift.



**Figura 4: flujo de datos desde Kinesis Firehose hacia Amazon Redshift**

### **Amazon Elasticsearch Service**

[Amazon Elasticsearch Service](#) (Amazon ES) es un servicio completamente administrado que pone a disposición las API de uso sencillo y las capacidades en tiempo real de Elasticsearch junto con la disponibilidad, escalabilidad y seguridad requeridas por las cargas de trabajo de producción.<sup>8</sup> Amazon ES le permite implementar, operar y escalar fácilmente Elasticsearch para los análisis de registros, la búsqueda completa de texto, el monitoreo de aplicaciones y otras operaciones.

Formato del envío de datos

Para el envío de datos a Amazon ES, Kinesis Firehose almacena en búfer los registros de entrada en función de la configuración del búfer de la secuencia de entrega y luego genera una solicitud masiva de Elasticsearch para indexar varios registros a su clúster de Elasticsearch. Asegúrese de que el registro esté codificado en UTF-8 y acoplado en un objeto JSON de una sola línea antes de enviarlo a Kinesis Firehose.



### Frecuencia del envío de datos

La frecuencia envío datos en Amazon ES se determina según los valores del tamaño y del intervalo del búfer Elasticsearch que ha configurado para la secuencia de entrega. Kinesis Firehose almacena en búfer los datos de entrada antes de entregarlos a Amazon ES. Puede configurar los valores para el tamaño del búfer Elasticsearch (1 MB a 100 MB) o el intervalo del búfer (60 segundos a 900 segundos). La condición que se cumpla primero activará el envío de datos a Amazon ES. Tenga en cuenta que si el envío de datos a destino no alcanza la velocidad de escritura de datos en la secuencia de entrega, Kinesis Firehose incrementa el tamaño del búfer de forma dinámica para alcanzarla y se asegura de que todos los datos lleguen a destino.

### Flujo de datos

La figura 5 muestra el flujo de datos para los destinos de Amazon ES.



**Figura 5: envío de datos desde Kinesis Firehose hacia el clúster de Amazon ES**

## Resumen

Kinesis Firehose es la forma más sencilla de almacenar sus datos de streaming en un destino compatible. Es una solución completamente administrada que requiere poco o ningún desarrollo. Para ABC Tolls, utilizar Kinesis Firehose era una elección obvia. Ya utilizaban Amazon Redshift como su solución de almacén de datos. Y debido a que sus orígenes de datos se escribían de forma continua en



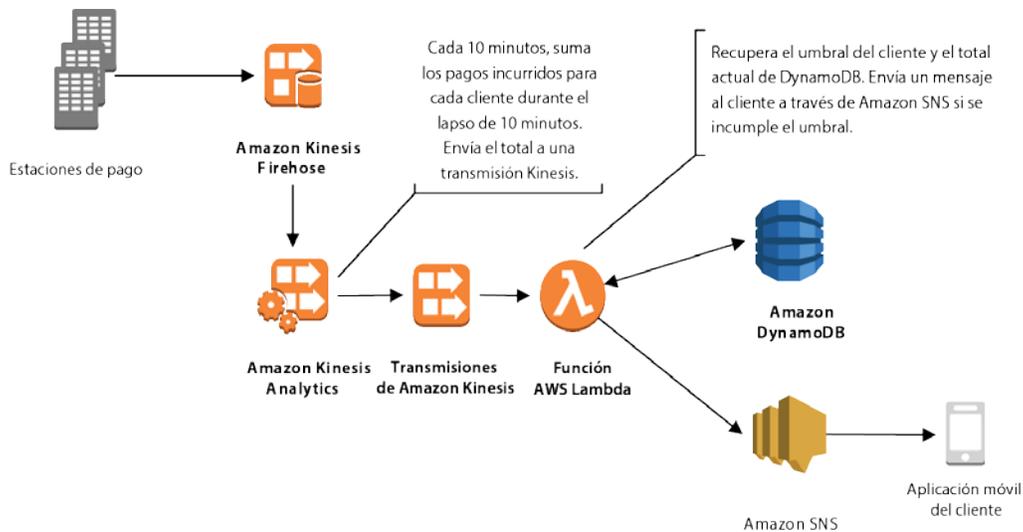
los registros de las transacciones, pudieron aprovechar Amazon Kinesis Agent para transmitir esos datos sin tener que escribir ningún código adicional.

Ahora que ABC Tolls ha creado una secuencia de registros de peaje y reciben estos registros a través de Kinesis Firehose, pueden utilizar esto como la base para sus otros requisitos de datos de streaming.

## Requisito 2: alertas del umbral de facturación

Para admitir la función de envío de notificación cuando se alcanza el umbral de gasto, el equipo de desarrollo de ABC Tolls ha creado una aplicación móvil y una tabla de [Amazon DynamoDB](#).<sup>9</sup> La aplicación permite a los clientes definir su umbral y la tabla almacena ese valor para cada cliente. La tabla también se utiliza para almacenar la cantidad acumulada que gastó cada cliente, cada mes. Para proporcionar notificaciones oportunas, ABC Tolls necesita actualizar el valor acumulativo en esta tabla de manera oportuna, y comparar ese valor con el umbral para determinar si se debe enviar una notificación al cliente. Dado que las transacciones de peaje ya están transmitiéndose mediante Kinesis Firehose, decidieron utilizar estos datos de streaming como el origen de su conjunto y alertas. Y debido a que Kinesis Analytics les permitió utilizar SQL para sumar los datos de streaming, es la solución ideal para el problema. En esta solución, Kinesis Analytics suma el valor total de las transacciones para cada cliente durante un período (lapso) de 10 minutos. Al final de este lapso, envía la suma total a una secuencia de Kinesis. Esta secuencia es el origen de eventos para una función de AWS Lambda. La función de Lambda consulta la tabla de DynamoDB para recuperar los umbrales y la suma total actual que cada cliente gastó descrita en el resultado de Kinesis Analytics. Para cada cliente, la función de Lambda actualiza la suma total actual en DynamoDB y también compara la suma total con el umbral. Si se ha superado el umbral, utiliza AWS SDK para indicarle a Amazon Simple Notification Service (SNS) que debe enviar una notificación a los clientes.

La figura 6 muestra la arquitectura para esta solución.



**Figura 6: arquitectura de las alertas y notificaciones del umbral de facturación**

Con esta solución, ABC Tolls proporciona a sus clientes una notificación oportuna cuando se aproximan a los límites de gasto.

Para extraer información en tiempo real de sus datos de streaming, ABC Tolls decidió utilizar Kinesis Analytics para analizar sus datos de streaming. Con Kinesis Analytics, ABC Tolls utilizó SQL, un idioma que ya conocían, para inspeccionar los datos a medida que se transmitían a lo largo de su secuencia de entrega. Analicemos Kinesis Analytics con más detalle.

## Amazon Kinesis Analytics

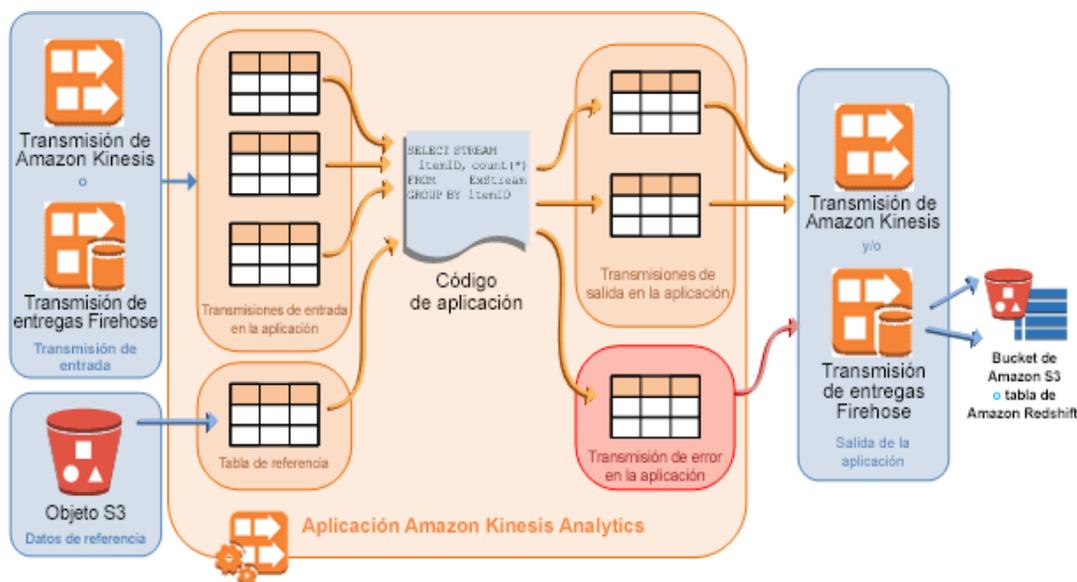
Con Kinesis Analytics, puede procesar y analizar datos de streaming mediante SQL. El servicio le permite rápidamente crear y ejecutar un potente código SQL en los orígenes de streaming para realizar un análisis de serie temporal, alimentar los paneles en tiempo real y crear métricas en tiempo real.

Para comenzar a utilizar Kinesis Analytics, debe crear una aplicación de Kinesis Analytics que lea y procese constantemente los datos de streaming. El servicio admite el ingreso de datos desde los orígenes de streaming de Kinesis Streams y Kinesis Firehose. Luego, crea el código SQL con el editor interactivo y lo prueba con los datos de streaming en directo. También puede configurar los destinos en los que desea que Kinesis Analytics almacene los resultados. Kinesis Analytics admite a Kinesis Firehose (Amazon S3, Amazon Redshift y Amazon Elasticsearch Service) y Kinesis Streams como destinos.



## Conceptos clave

Una *aplicación* es el recurso principal en Kinesis Analytics que puede crear en su cuenta. Las aplicaciones de Kinesis Analytics leen y procesan datos de streaming en tiempo real de forma constante. Usted escribe el código de aplicación con SQL para procesar los datos de streaming de entrada y producir un resultado. Luego, Kinesis Analytics escribe el resultado en un destino configurado. La figura 7 ilustra una arquitectura de aplicaciones típica.



**Figura 7: arquitectura para una aplicación de Kinesis Analytics**

Cada aplicación tiene un nombre, una descripción, un ID de versión y un estado. Al crear una aplicación, solo tiene que configurar la entrada, crear el código de la aplicación y configurar el resultado.

### **Entrada**

La entrada de la aplicación es el origen de streaming para su aplicación. Puede seleccionar una secuencia de Kinesis o una secuencia de entrega como el origen de streaming. Tiene la opción de configurar un origen de datos de referencia para enriquecer su secuencia de datos de entrada en la aplicación. Como consecuencia, se crea una tabla de referencia dentro la aplicación. Debe almacenar sus datos de referencia como un objeto en un bucket de S3. Cuando se inicia la aplicación, Kinesis Analytics lee el objeto de S3 y crea una tabla dentro de la aplicación.

ABC Tolls utilizaba su secuencia de entrega como entrada a su aplicación de Kinesis Analytics.

### ***Cód. aplicación***

Su código de aplicación se compone de una serie de instrucciones SQL que procesan entradas y producen resultados. Puede escribir instrucciones SQL contra las secuencias dentro de la aplicación, consultar tablas y puede escribir consultas JOIN para combinar los datos de ambos orígenes.

En su forma más sencilla, el código de aplicación puede ser una única instrucción SQL que selecciona desde una entrada de streaming e introduce los resultados en una salida de streaming. También puede ser una serie de instrucciones SQL en las que el resultado de una instrucción impulsa el resultado de la siguiente instrucción SQL. Además, puede escribir el código de aplicación para dividir una secuencia de entrada en varias secuencias y aplicar consultas adicionales para procesar estas secuencias.

### ***Resultado***

En su código de la aplicación, los resultados de las consultas van a las secuencias dentro de la aplicación. En su código de aplicación, puede crear una o más secuencias dentro de la aplicación para conservar los resultados intermedios. También tiene la opción de configurar los resultados de la aplicación para almacenar datos en las secuencias dentro de la aplicación, que mantienen el resultado de su aplicación (también conocido como secuencias de resultado dentro la aplicación) para los destinos externos. Los destinos externos pueden ser una secuencia de entrega o una secuencia de Kinesis.

ABC Tolls utilizaba Kinesis Streams como el destino de sus valores agregados.

## **Resumen**

Kinesis Analytics le permite utilizar SQL a fin de deducir información a partir de sus datos mientras se transmiten por el sistema. ABC Tolls escribió sus SQL para realizar conjuntos de 10 minutos de duración para sumar el total de peajes devengados por sus clientes. Los valores del resultado de estos conjuntos de 10 minutos podrían compararse con los umbrales de los clientes.

Como se mencionó anteriormente, Kinesis Analytics genera sus resultados para Kinesis Streams o Kinesis Firehose. En este ejemplo, ABC Tolls eligió enviar el resultado de Kinesis Analytics a una secuencia de Kinesis debido a la integración de Kinesis Streams con AWS Lambda. Conozcamos más sobre Kinesis Streams.



## Amazon Kinesis Streams

Amazon Kinesis Streams le permite crear aplicaciones personalizadas en tiempo real con marcos de procesamiento de secuencias conocidos y cargar los datos de streaming en cualquier almacén de datos. Puede configurar cientos de miles de productores de datos para incluir datos de manera continua en una secuencia de Kinesis, por ejemplo, datos de las secuencias de clics de los sitios web, registros de aplicaciones, sensores de IoT y fuentes de redes sociales. En menos de un segundo, los datos estarán disponibles para que su aplicación los lea y los procese desde la secuencia.

Cuando implementa una solución con Kinesis Streams, creará aplicaciones de procesamiento de datos personalizadas conocidas como aplicaciones *de Kinesis Streams*. Una aplicación típica de Kinesis Streams lee datos de una secuencia de Kinesis como registros de datos.

Si bien puede utilizar Kinesis Streams para resolver diversos problemas relacionados con los datos de streaming, un uso habitual es la suma o el análisis de datos en tiempo real, además de la carga de los datos sumados en un almacén de datos o un clúster de asignar/reducir.

Los datos se insertan en Kinesis Streams, lo que garantiza su durabilidad y elasticidad. El tiempo que transcurre entre el momento en que un registro se inserta en la secuencia y el tiempo en el que se puede recuperar (retraso put-to-get) es normalmente menor a 1 segundo, es decir, una aplicación de Kinesis Streams puede empezar a consumir los datos de la secuencia casi inmediatamente después de agregarlos. Dado que Kinesis Streams es un servicio administrado, lo exime de la carga operativa de crear y ejecutar una canalización de admisión de datos.

### Envío de datos a Amazon Kinesis Streams

Existen varios mecanismos para enviar datos a su secuencia. AWS ofrece SDK para muchos idiomas de programación conocidos, cada uno de los cuales proporciona las API para Kinesis Streams. AWS también ha creado varias utilidades para ayudar a enviar datos a su secuencia. Analicemos cada uno de los métodos que puede utilizar y por qué podría elegir cada uno.

### ***Amazon Kinesis Agent***

Ya hemos analizado a Amazon Kinesis Agent anteriormente como una herramienta que se puede utilizar para enviar datos a Kinesis Firehose. La misma herramienta se puede utilizar para enviar datos a Kinesis Streams. Para obtener más información sobre cómo instalar y configurar Kinesis Agent, consulte [Cómo escribir en Amazon Kinesis Firehose a través de Amazon Kinesis Agent](#).<sup>10</sup>

### ***Amazon Kinesis Producer Library (KPL)***

KPL simplifica el desarrollo de aplicaciones de productores y permite a los desarrolladores alcanzar un mayor desempeño de escritura en una o varias secuencias de Kinesis. KPL es una biblioteca fácil de usar y con gran capacidad de configuración que se instala en los hosts que generan los datos que desea transmitir a Kinesis Streams. Actúa como un intermediario entre su código de aplicación de productor y las acciones API de Kinesis Streams. KPL realiza las siguientes tareas principales:

- Escribe en una o varias secuencias de Kinesis con un mecanismo de volver a intentarlo automático y configurable
- Recopila registros y utiliza `PutRecords` para escribir varios registros en varios fragmentos por solicitud
- Suma los registros de usuario para aumentar el tamaño de la carga y mejorar el desempeño
- Se integra perfectamente con Amazon Kinesis Client Library (KCL) para dividir los registros en lotes sobre el consumidor
- Envía las métricas de Amazon CloudWatch en su nombre para proporcionar información sobre el desempeño del productor

KPL puede utilizarse en casos de uso síncronos o asíncronos. Se recomienda utilizar el mayor desempeño de la interfaz asíncrona a menos que haya una razón específica para utilizar un comportamiento síncrono. Para obtener más información sobre estos dos casos de uso y el código de ejemplo, consulte [Cómo escribir en sus secuencias con KPL](#).<sup>11</sup>

KPL puede ayudarle a crear productores de alto desempeño. Piense en una situación en la que sus instancias de Amazon Elastic Compute Cloud (EC2) sirven como un proxy para recopilar eventos de 100 bytes a partir de cientos o miles de dispositivos de baja potencia y para escribir registros en una



secuencia de Kinesis. Cada una de estas instancias EC2 debe escribir miles de eventos por segundo en su secuencia de Kinesis. Para conseguir el nivel de desempeño necesario, los productores deben implementar una lógica complicada como la creación de lotes o subprocesos, además de la lógica de un nuevo intento y la división de registros en el lado del consumidor. KPL realiza todas estas tareas por usted.

Dado que KPL almacena en búfer sus registros antes de que se envíen a una secuencia de Kinesis, KPL puede ocasionar un retraso adicional en el procesamiento, según el período de tiempo que haya configurado para que KPL almacene en búfer los registros antes de enviarlos a Kinesis. Un tiempo de búfer más prolongado conlleva a una mayor eficacia de empaquetamiento y un mejor desempeño. Las aplicaciones que no pueden tolerar este retraso adicional podrían tener que utilizar AWS SDK directamente.

Si la aplicación no registra los informes en un archivo local y se crea un gran número de pequeños registros por segundo, considere la posibilidad de utilizar el KPL.

Para obtener más información acerca de cómo utilizar el KPL para producir datos, consulte [Cómo desarrollar productores de Amazon Kinesis Streams con Amazon Kinesis Producer Library](#).<sup>12</sup>

### ***API de Amazon Kinesis***

Una vez que se crea una secuencia, puede agregarle los registros de datos. Un registro es una estructura de datos que contiene los datos a procesar en forma de un blob de datos. Después de almacenar los datos en el registro, Kinesis Streams no inspecciona, interpreta ni cambia los datos de ninguna forma.

Existen dos operaciones diferentes en la API de Kinesis Streams que agregan datos a una secuencia: `PutRecords` y `PutRecord`. La operación `PutRecords` envía varios registros a su secuencia por solicitud HTTP y la operación `PutRecord` envía registros a su secuencia, una por vez (se necesita una solicitud HTTP independiente para cada registro). Es preferible utilizar `PutRecords` para la mayoría de las aplicaciones porque conseguirá un mayor rendimiento por productor de datos.

Debido a que las API se exponen en todos los AWS SDK, el uso de la API para escribir registros proporciona la solución más flexible para enviar datos a una secuencia de Kinesis. Si no puede utilizar Kinesis Agent o KPL (por ejemplo, desea escribir mensajes directamente desde una aplicación móvil o desea reducir la latencia total de los mensajes en la medida de lo posible), entonces utilice las API para escribir registros en la secuencia de Kinesis.

Para obtener más información sobre estas API, consulte [Cómo usar la API](#) en la documentación de Kinesis Streams.<sup>13</sup> Puede encontrar los detalles para cada una de las operaciones de API en la [referencia de API de Amazon Kinesis Streams](#).<sup>14</sup>

## Procesamiento de datos en Amazon Kinesis Streams

Un consumidor es una aplicación que lee y procesa datos procedentes de Kinesis Streams. Puede crear consumidores para Kinesis Streams de varias maneras. En esta sección, analizaremos cuatro de los métodos más comunes: con Kinesis Analytics, con KCL, con Amazon Lambda y directamente con la API de Kinesis Streams.

### ***Uso de Amazon Kinesis Analytics***

Anteriormente, hablamos de cómo Kinesis Analytics se puede utilizar para analizar datos de streaming con SQL estándar. Kinesis Analytics puede leer los datos de su secuencia de Kinesis y procesarlos con el SQL que proporcione. Para obtener más información sobre el procesamiento de los datos de streaming con Kinesis Analytics, consulte [Cómo configurar la entrada de la aplicación](#) en la guía para desarrolladores de Kinesis Analytics.

### ***Uso de Amazon Kinesis Client Library (KCL)***

Puede desarrollar una aplicación de consumidor para Kinesis Streams mediante KCL. Si bien puede utilizar la API de Kinesis Streams para obtener datos de una secuencia de Amazon Kinesis, le recomendamos que utilice los patrones de diseño y el código para aplicaciones de consumidor proporcionados por KCL.

KCL le ayuda a consumir y procesar datos de una secuencia de Kinesis. Este tipo de aplicación también se conoce como un consumidor. KCL se encarga de muchas de las tareas complejas asociadas a la informática distribuida, como el balanceador de carga entre varias instancias, la respuesta a errores en las instancias, la creación de puntos de control en registros procesados y la respuesta a cambios en los fragmentos. KCL le permite concentrarse en la lógica de procesamiento de registro de escritura.



KCL es una biblioteca de Java que ofrece compatibilidad para otros idiomas diferentes a través de una interfaz de múltiples idiomas. En el tiempo de ejecución, una aplicación de KCL crea una instancia de un proceso de trabajo con información de configuración y luego utiliza un procesador de registros para procesar los datos recibidos desde una secuencia de Kinesis. Se puede ejecutar una aplicación de KCL en cualquier cantidad de instancias. Las diversas instancias de la misma aplicación se coordinan en cuanto a los errores y balancean la carga de forma dinámica. También puede tener varias aplicaciones de KCL funcionando en la misma secuencia, en función de los límites de rendimiento. KCL actúa como intermediaria entre su lógica de procesamiento de registros y Kinesis Streams.

Para obtener información detallada sobre cómo crear su propia aplicación de KCL, consulte [Cómo desarrollar consumidores de Amazon Kinesis Streams con Amazon Kinesis Client Library](#).<sup>15</sup>

### ***Uso de AWS Lambda***

[AWS Lambda](#) es un servicio informático que permite ejecutar código sin aprovisionar ni administrar servidores.<sup>16</sup> AWS Lambda ejecuta el código solo cuando es necesario y escala de manera automática. Con AWS Lambda, puede ejecutar código sin ningún tipo de administración. AWS Lambda ejecuta el código en una infraestructura informática de alta disponibilidad y realiza todas las tareas de administración de los recursos informáticos, incluido el mantenimiento del servidor y sistema operativo, el aprovisionamiento de capacidad y el escalado automático, y el monitoreo del código y la elaboración de registros. Lo único que tiene que hacer es suministrar su código en uno de los idiomas compatibles con AWS Lambda.

Puede suscribirse a las funciones de Lambda para que lean automáticamente los lotes de registros procedentes de la secuencia de Kinesis y los procesen si se detectan registros en la secuencia. Posteriormente, AWS Lambda sondea la secuencia de forma periódica (una vez por segundo) en busca de registros nuevos. Cuando detecta nuevos registros, invoca su función de Lambda pasando los registros nuevos como parámetro. Si no se detectan nuevos registros, no se invoca la función de Lambda.

Para obtener información detallada sobre cómo utilizar AWS Lambda para consumir datos de Kinesis Streams, consulte [Cómo usar AWS Lambda con Amazon Kinesis](#).<sup>17</sup>



### ***Uso de la API***

Para la mayoría de los casos de uso, debe utilizar KCL o AWS Lambda para recuperar y procesar datos de una secuencia. Sin embargo, si prefiere escribir su propia aplicación de consumidor desde cero, existen varios métodos que lo permiten. La API de Kinesis Streams proporciona los métodos `GetShardIterator` y `GetRecords` para recuperar datos de una secuencia. Se trata de un modelo de extracción, donde el código extrae datos directamente de los fragmentos de la secuencia. Para obtener más información acerca de cómo escribir su propia aplicación de consumidor mediante la API, consulte [Cómo desarrollar consumidores de Amazon Kinesis Streams con la API de Amazon Kinesis Streams](#). Puede encontrar detalles sobre la API en la [referencia de la API de Amazon Kinesis Streams](#).<sup>18</sup>

### ***Elección del mejor modelo de consumidor para su aplicación***

¿Cómo saber cuál es el mejor modelo de consumidor para su caso de uso? Cada método tiene su propio conjunto de ventajas y tendrá que determinar qué es importante para usted. A continuación, se muestran algunas instrucciones generales para ayudarle a elegir el modelo de consumidor adecuado.

En la mayoría de los casos, considere empezar con AWS Lambda. Su facilidad de uso y modelo de implementación sencilla le permitirá crear rápidamente un consumidor de datos. La ventaja de utilizar AWS Lambda es que cada invocación de la función de Lambda debe considerarse sin estado. Es decir, no puede utilizar fácilmente los resultados de invocaciones anteriores de la función (por ejemplo, lotes anteriores de registros de la secuencia). Tenga en cuenta también que el tiempo de ejecución máximo para una única función de Lambda es de 5 minutos. Si un único lote de registros tarda más de 5 minutos en procesarse, AWS Lambda podría no ser el mejor consumidor para su caso de uso.

Si decide que no puede utilizar AWS Lambda, considere la posibilidad de crear su propia aplicación de procesamiento con KCL. Dado que implementa aplicaciones de KCL en las instancias de EC2 dentro de su cuenta de AWS, tiene mucha flexibilidad y control en la persistencia y estado de los datos locales para sus datos.

Su tercera opción es crear su propia aplicación directamente mediante las API. Esto le ofrece el máximo control y flexibilidad, pero también deberá crear su propia lógica para gestionar las funciones de aplicación de consumidor habituales, como puntos de control, escalado y conmutación por error.

## Resumen

Kinesis Streams facilita la recepción de datos de streaming. Puede escalar una secuencia de Kinesis para gestionar solo algunos registros por segundo o millones de registros por segundo. Para ABC Tolls, la tasa de la secuencia de datos no era alta. Sin embargo, se beneficiaron de la integración directa con AWS Lambda, lo que fácilmente les permitió realizar llamadas API a Amazon SNS para las notificaciones de usuario.

## Requisito 3: otras alertas del umbral

El requisito final es similar al anterior, pero presenta un problema adicional. Para resumir este requisito final, los operadores de ABC Tolls quieren recibir notificación inmediata cuando el tráfico de vehículos de una estación de peaje cae por debajo de un umbral predefinido por cada período de 30 minutos en un día. Por ejemplo, saben que, a partir de los datos históricos, una de sus estaciones de peaje ve aproximadamente 360 vehículos los miércoles entre las 2:00 p. m. y las 2:30 p. m. En ese lapso de 30 minutos, si una estación de peaje ve menos de 100 vehículos, desean recibir una notificación.

ABC Tolls quiere comparar la suma total actual de vehículos para cada estación con una tasa promedio conocida para esa estación. Para lograrlo, crearon un archivo que contiene los valores de tráfico del umbral para cada lapso de 30 minutos, para cada estación. Como se describe anteriormente, Kinesis Analytics admite el uso de los datos de referencia. Creará una secuencia dentro de la aplicación (como una tabla) en función de los datos en un archivo almacenado en un bucket de S3. Con esto en marcha, los desarrolladores de ABC Tolls pudieron escribir SQL en su aplicación de Kinesis Analytics para contar el número de vehículos vistos en cada estación durante un lapso superior a 30 minutos y comparar esos valores con los umbrales en el archivo. Si se incumplió con el umbral, Kinesis Analytics genera un registro para una secuencia de Kinesis. Cuando los registros llegan a la secuencia, se ejecuta una función de Lambda, que utiliza Amazon SNS para enviar una notificación a los operadores de AWS Tolls. La figura 8 ilustra la arquitectura para esta situación.

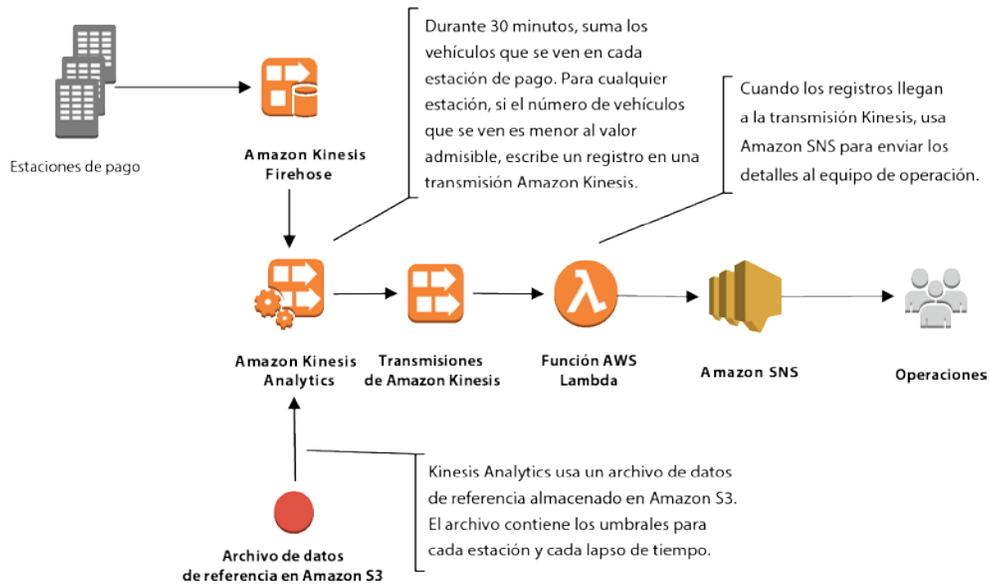


Figura 8: arquitectura para las alertas y notificaciones con períodos de 30 minutos

## Arquitectura completa

Con una solución para cada requisito, ahora tenemos nuestra solución de streaming general, como se muestra en la figura 9.

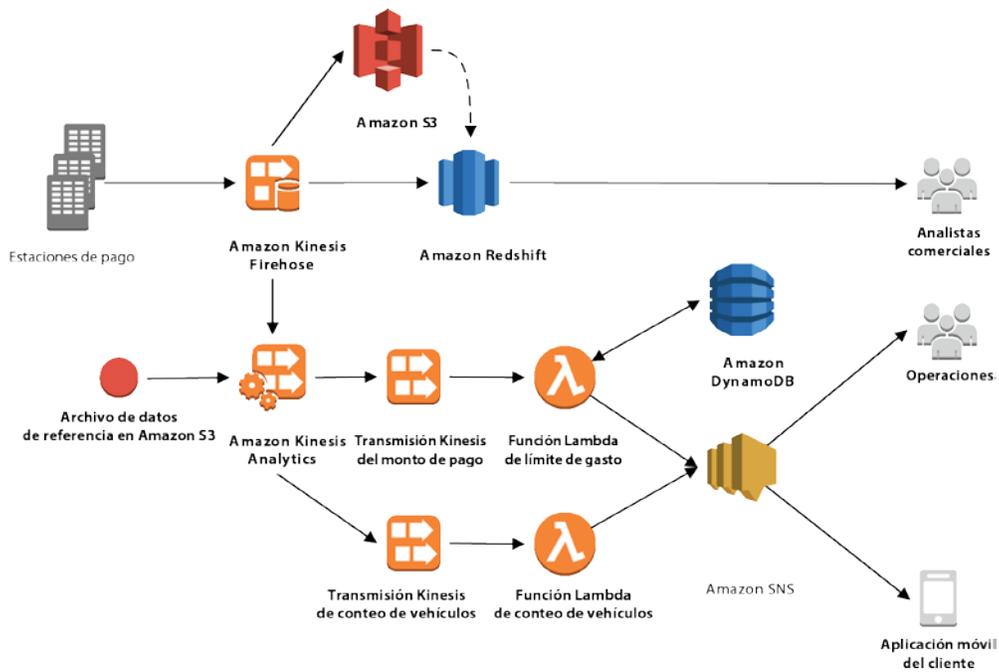


Figura 9: arquitectura de la solución de streaming general

Este diseño le proporciona a ABC Tolls una arquitectura flexible y reactiva. Al transmitir las transacciones de sus clientes en tiempo real, pueden darse cuenta de sus requisitos con muy poco esfuerzo de desarrollo y una infraestructura mínima para administrar.

## Conclusión

En este documento, analizamos cómo la compañía ficticia, ABC Tolls, utilizó los servicios de Amazon Kinesis para pasar de un flujo de trabajo por lotes tradicional a un flujo de trabajo de streaming. Esta migración les dio la posibilidad de añadir nuevas funciones y funcionalidad que no eran posible con su solución de lote tradicional.

Al analizar los datos a medida que se van creando, podrá obtener información sobre lo que está haciendo su negocio en este momento. Los servicios de Amazon Kinesis le permiten centrarse en su aplicación para tomar decisiones empresariales sujetas al tiempo en lugar de implementar y administrar la infraestructura.

## Colaboradores

Las siguientes personas y organizaciones han participado en la redacción de este documento:

- Allan MacInnis, Arquitecto de soluciones, AWS
- Chander Matrubhutam, Gerente de marketing de productos, AWS

## Notes

- <sup>1</sup> <https://aws.amazon.com/kinesis/streams/>
- <sup>2</sup> <https://aws.amazon.com/kinesis/firehose/>
- <sup>3</sup> <https://aws.amazon.com/kinesis/analytics/>
- <sup>4</sup> <http://docs.aws.amazon.com/firehose/latest/dev/writing-with-sdk.html>
- <sup>5</sup> <https://aws.amazon.com/lambda/>
- <sup>6</sup> <https://aws.amazon.com/s3/>
- <sup>7</sup> <https://aws.amazon.com/redshift/>
- <sup>8</sup> <https://aws.amazon.com/elasticsearch-service/>
- <sup>9</sup> <https://aws.amazon.com/dynamodb/>
- <sup>10</sup> <http://docs.aws.amazon.com/firehose/latest/dev/writing-with-agents.html>
- <sup>11</sup> <http://docs.aws.amazon.com/streams/latest/dev/kinesis-kpl-writing.html>
- <sup>12</sup> <http://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-kpl.html>
- <sup>13</sup> <http://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-sdk.html>
- <sup>14</sup> <http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html>
- <sup>15</sup> <http://docs.aws.amazon.com/streams/latest/dev/developing-consumers-with-kcl.html>
- <sup>16</sup> <https://aws.amazon.com/lambda/>
- <sup>17</sup> <http://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html>
- <sup>18</sup> <http://docs.aws.amazon.com/kinesis/latest/APIReference/Welcome.html>