Guía de inicio de Apache Kafka: Arquitecturas de diseño de sistemas: sistema de notificación, rastreador de actividad web, canalización ELT, sistema de almacenamiento

Arneesh Aima Seguir
12 de junio de 2020 · 9 min de lectura ★

¿Qué es Kafka?

Kafka es una **plataforma de procesamiento de flujo** distribuida de código abierto a través de la cual podemos **publicar** , **suscribirnos al** flujo de registros, **almacenar** estos registros y procesar / extraer este flujo de registros sobre la marcha. Kafka es mantenido por **Apache Software Foundation** . Kafka se puede utilizar de múltiples formas para extraer varios casos de uso de acuerdo con nuestras necesidades. Algunos de los casos de uso más famosos que se pueden manejar desde Kafka se explican y demuestran en forma de diagrama en las secciones posteriores de este blog. Los casos de uso más comunes de Kafka son construir canalizaciones de datos de transmisión en tiempo real y crear marcos ETL / ETL de transmisión en tiempo real.

Estructura y funcionamiento

Kafka siempre se configura como un clúster que se ejecuta en uno o más servidores, estos servidores se denominan "intermediarios" en la terminología de Kafka. El flujo de registros se almacena en este grupo dentro de categorías llamadas **temas** y cada entrada en un tema se llama **registro** . Cada registro del tema de Kafka contiene una clave, un valor y una marca de tiempo asociada.

Kafka tiene 5 API principales asociadas:

1. API de productor
2. API de consumidor
3. API de Streams
4. API de conector
5. API de administración.

Las API de Producer, Consumer, Streams y Connector están relacionadas con componentes clave del ecosistema de Kafka que se utilizan para operaciones de entrada / salida según varios escenarios. Estos componentes se explican con más detalle en este blog. Mientras que la API de administración permite administrar e inspeccionar temas, agentes (servidores) y otros objetos de Kafka.

En Kafka, la **comunicación entre los clientes y los servidores / intermediarios** se realiza con un **protocolo TCP** simple, de alto rendimiento e independiente del lenguaje (Transmission Control Protocol).

Características de Kafka

Las características más destacadas de Apache Kafka son:

Alto rendimiento

Kafka proporciona un rendimiento relativamente alto que otros sistemas de mensajería como RabbitMQ y ActiveMQ. Una gran cantidad de mensajes se pueden procesar fácilmente mediante el despliegue de tamaño medio en Kafka. Otros sistemas de mensajería requerirían implementaciones mucho más grandes y muchos nodos para lograr la misma capacidad de procesamiento de mensajes.

Durabilidad

Kafka ofrece la posibilidad de que el mensaje persista en el disco. Esta propiedad proporciona una tremenda tolerancia a fallos y funcionalidades de durabilidad. Incluso si algo sale mal en medio de una transmisión, ninguno de nuestros mensajes se perderá y se guardará de forma segura en el disco. Podemos comenzar el consumo de mensajes exactamente desde el mismo lugar donde lo dejamos en caso de que el servidor falle.

Replicación de datos

Kafka proporciona la funcionalidad de replicación de datos. Esta propiedad la proporcionan muchas herramientas en estos días para permitir el procesamiento paralelo al más alto nivel. Más replicación significa que más suscriptores pueden consumir temas en paralelo. La replicación de datos también se realiza para evitar la pérdida de datos en caso de un bloqueo.

Procesamiento de flujo

Kafka proporciona la funcionalidad para construir canales de transmisión de datos en tiempo real que permiten un intercambio de datos confiable entre sistemas y aplicaciones. Estas canalizaciones de procesamiento de flujos se utilizan mucho para transformar un flujo de datos.

Escalabilidad

Las capacidades de escalabilidad de Kafka son una de las mejores entre todos los servicios de mensajería / plataformas de transmisión en el mundo en este momento. Proporciona una funcionalidad de escalabilidad sencilla con un gran rendimiento.

Componentes / conceptos clave de Kafka

Los componentes / conceptos clave de Apache Kafka son:

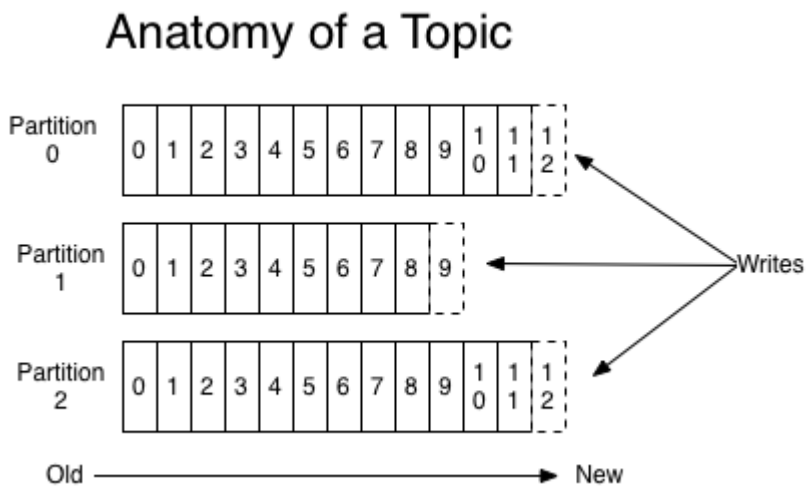
Mensajes / Registros

La entidad más pequeña del ecosistema de Kafka se llama mensaje o registro. Un registro es una entrada única dentro de un tema de Kafka. Esto se puede considerar como un documento dentro de una colección de MongoDB.

```
{"schema":{"type":"string","optional":false},"payload":"foo"}
{"schema":{"type":"string","optional":false},"payload":"bar"}
...
```

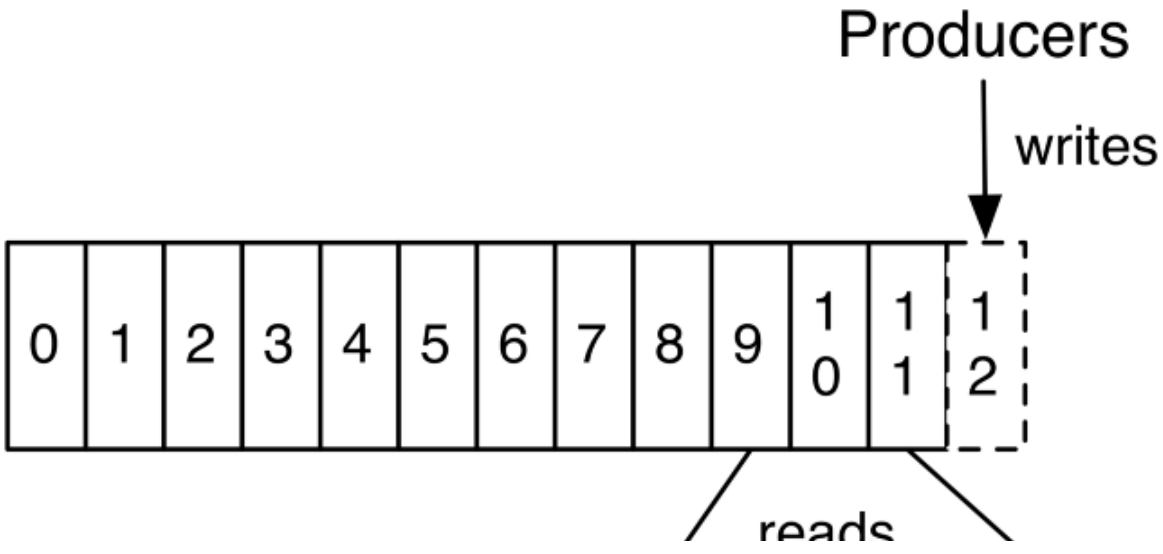
Temas

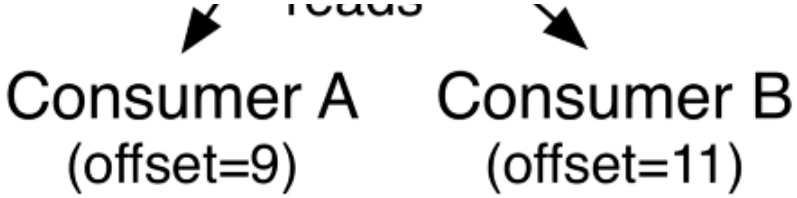
Un tema es un nombre de fuente en el que se publican los registros; se puede considerar similar a una colección MongoDB o una tabla MySQL. Una colección de MongoDB contiene documentos, de manera similar, un tema de Kafka contiene registros. Los temas en Kafka son siempre de múltiples suscriptores; es decir, un tema puede tener cero, uno o muchos consumidores que se suscriban a los datos escritos en él.



El clúster de Kafka mantiene un registro particionado para cada tema de Kafka. Cada partición es una secuencia ordenada e inmutable de registros que se agrega continuamente a: un registro de confirmación estructurado. A cada uno de los registros de las particiones se les asigna un número de identificación secuencial llamado desplazamiento que identifica de forma única a cada registro dentro de la partición.

El clúster de Kafka conserva de forma duradera todos los registros publicados, se hayan consumido o no, mediante un período de retención configurable. Por ejemplo, si la política de retención se establece en dos días, durante los dos días posteriores a la publicación de un registro, estará disponible para el consumo, después de lo cual se descartará para liberar espacio. El rendimiento de Kafka es efectivamente constante con respecto al tamaño de los datos, por lo que almacenar datos durante mucho tiempo no es un problema.





De hecho, los únicos metadatos retenidos por consumidor es el desplazamiento o la posición de ese consumidor en el registro. Este desplazamiento lo controla el consumidor: normalmente un consumidor avanzará su desplazamiento linealmente a medida que lee los registros, pero, de hecho, dado que la posición está controlada por el consumidor, puede consumir registros en el orden que desee. Por ejemplo, un consumidor puede restablecer un desplazamiento anterior para reprocesar datos del pasado o saltar al registro más reciente y comenzar a consumir desde "ahora".

Esta combinación de características significa que los consumidores de Kafka son muy económicos: pueden entrar y salir sin mucho impacto en el clúster o en otros consumidores. Por ejemplo, puede utilizar nuestras herramientas de línea de comandos para "seguir" el contenido de cualquier tema sin cambiar lo que consumen los consumidores existentes.

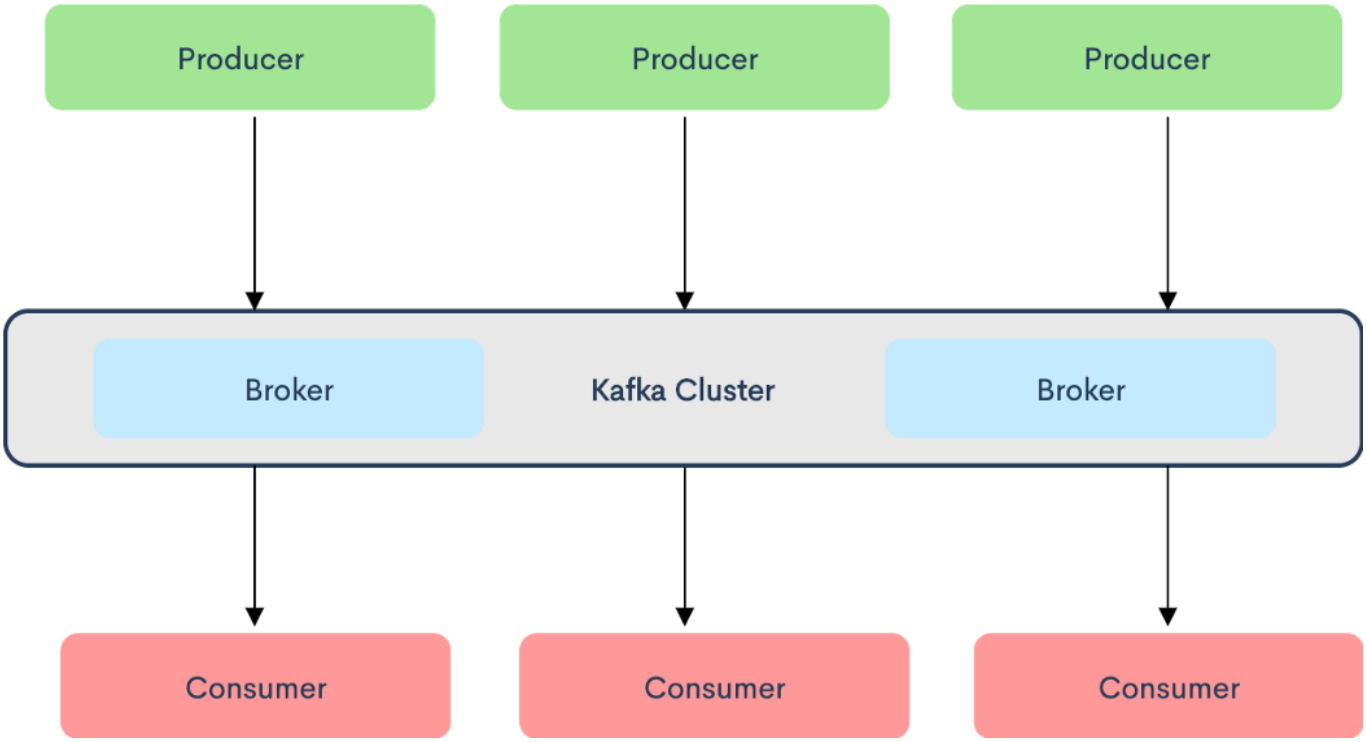
Las particiones en el registro sirven para varios propósitos. Primero, permiten que el registro se amplíe más allá de un tamaño que cabe en un solo servidor. Cada partición individual debe caber en los servidores que la alojan, pero un tema puede tener muchas particiones para que pueda manejar una cantidad arbitraria de datos. En segundo lugar, actúan como la unidad de paralelismo, más sobre eso en un momento. *(Fuente de la explicación de los temas de Kafka: Documentación oficial de Kafka)*

Corredores y Distribución

Un servidor de Kafka se llama corredor. Las particiones del registro se distribuyen a través de los intermediarios (servidores) en el clúster de Kafka y cada servidor maneja los datos y las solicitudes para compartir las particiones. Cada partición se replica en una cantidad configurable de servidores para tolerancia a fallas.

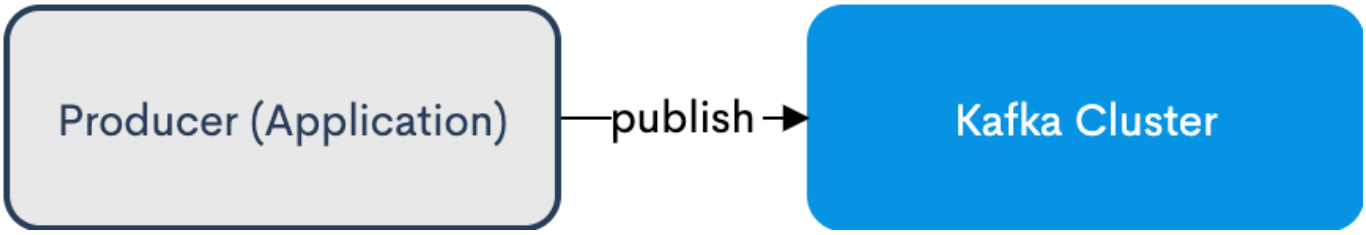
Cada partición tiene un servidor que actúa como "**líder**" y cero o más servidores que actúan como "**seguidores**" . El líder maneja todas las solicitudes de lectura y escritura para la partición, mientras que los seguidores replican pasivamente al líder. Si el líder

falla, uno de los seguidores se convertirá automáticamente en el nuevo líder. Cada servidor actúa como líder para algunas de sus particiones y seguidor para otras, por lo que la carga está bien equilibrada dentro del clúster.



Productores

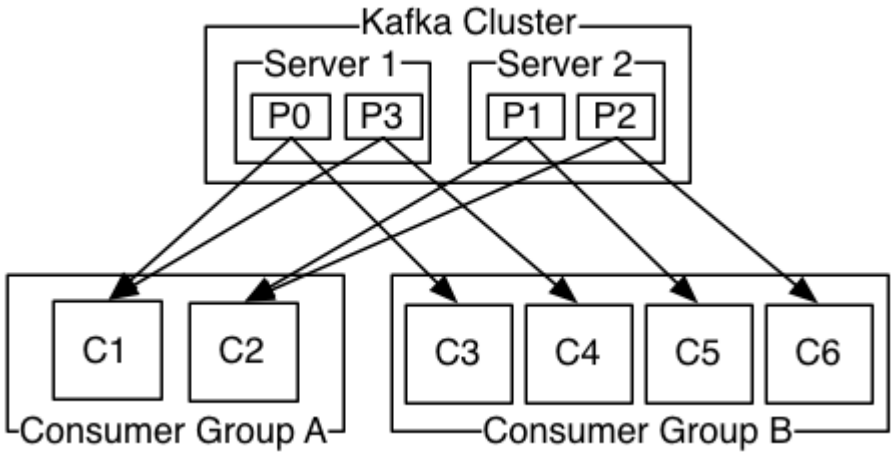
Los productores son la aplicación que publica un flujo de datos sobre el tema de Kafka deseado. La asignación de registros dentro de una partición de tema es manejada por el productor, lo que se puede hacer en función de una función de partición semántica o la asignación puede realizarse en el patrón de operación por turnos predeterminado en el que los registros se asignan 1… .ny luego nuevamente desde 1… n de manera exhaustiva.



Consumidores

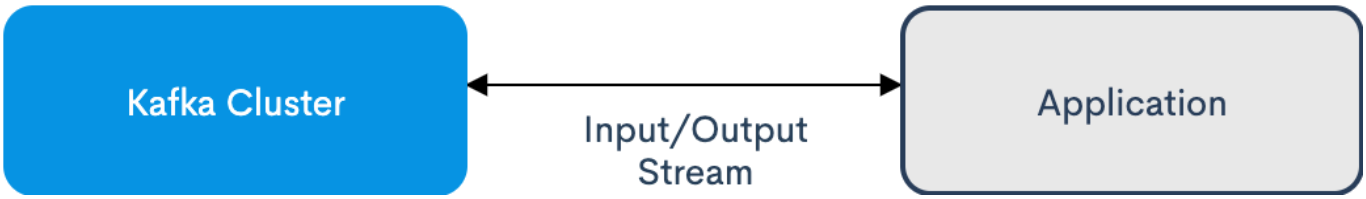
Los consumidores son los suscriptores de los temas de Kafka. Se suscriben a un tema y extraen los datos deseados.

Los datos se extraen de un tema de Kafka mediante una **instancia de consumidor** que se encapsula dentro de un **grupo de consumidores** . *"Si todas las instancias de consumidores tienen el mismo grupo de consumidores, entonces los registros se equilibrarán de manera efectiva en las instancias de consumidores". "Si todas las instancias de consumidores tienen diferentes grupos de consumidores, entonces cada registro se transmitirá a todos los procesos de consumidores".*



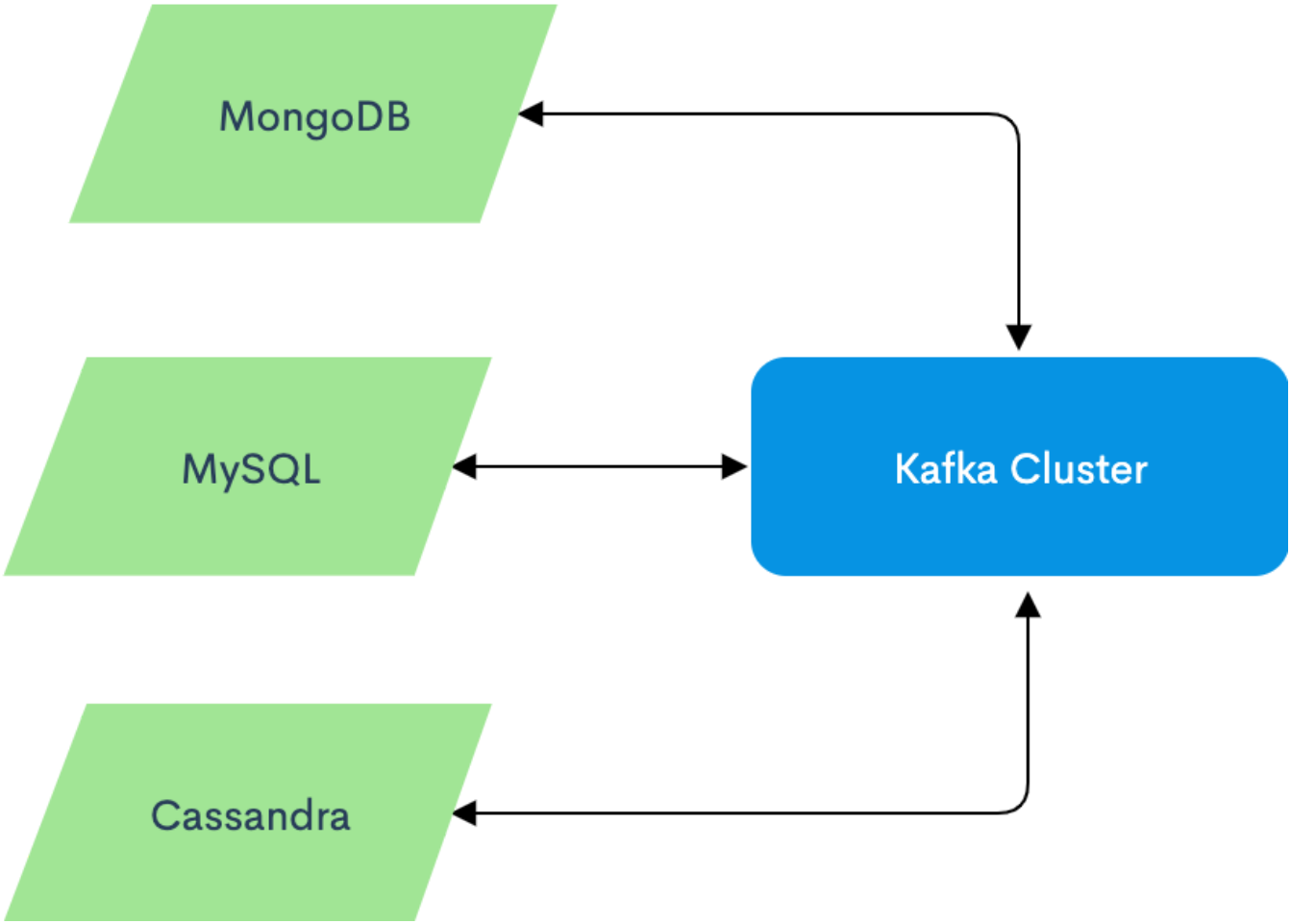
Corrientes

Los flujos consumen un flujo de entrada de uno o más temas y producen un flujo de salida para uno o más temas de salida, transformando efectivamente los flujos de entrada en flujos de salida. **Esto se utiliza principalmente para completar o transformar datos principalmente en canalizaciones ELT / ETL.**



Conector

Los conectores son los productores o consumidores reutilizables que conectan los temas de Kafka con aplicaciones o sistemas de datos existentes. Por ejemplo, un conector a una base de datos relacional puede capturar todos los cambios en una tabla.



Kafka como infraestructura: diseño de sistemas: varios casos de uso

Varios casos de uso de Kafka:

1. Kafka como canalización ELT / ETL de Stream Processing

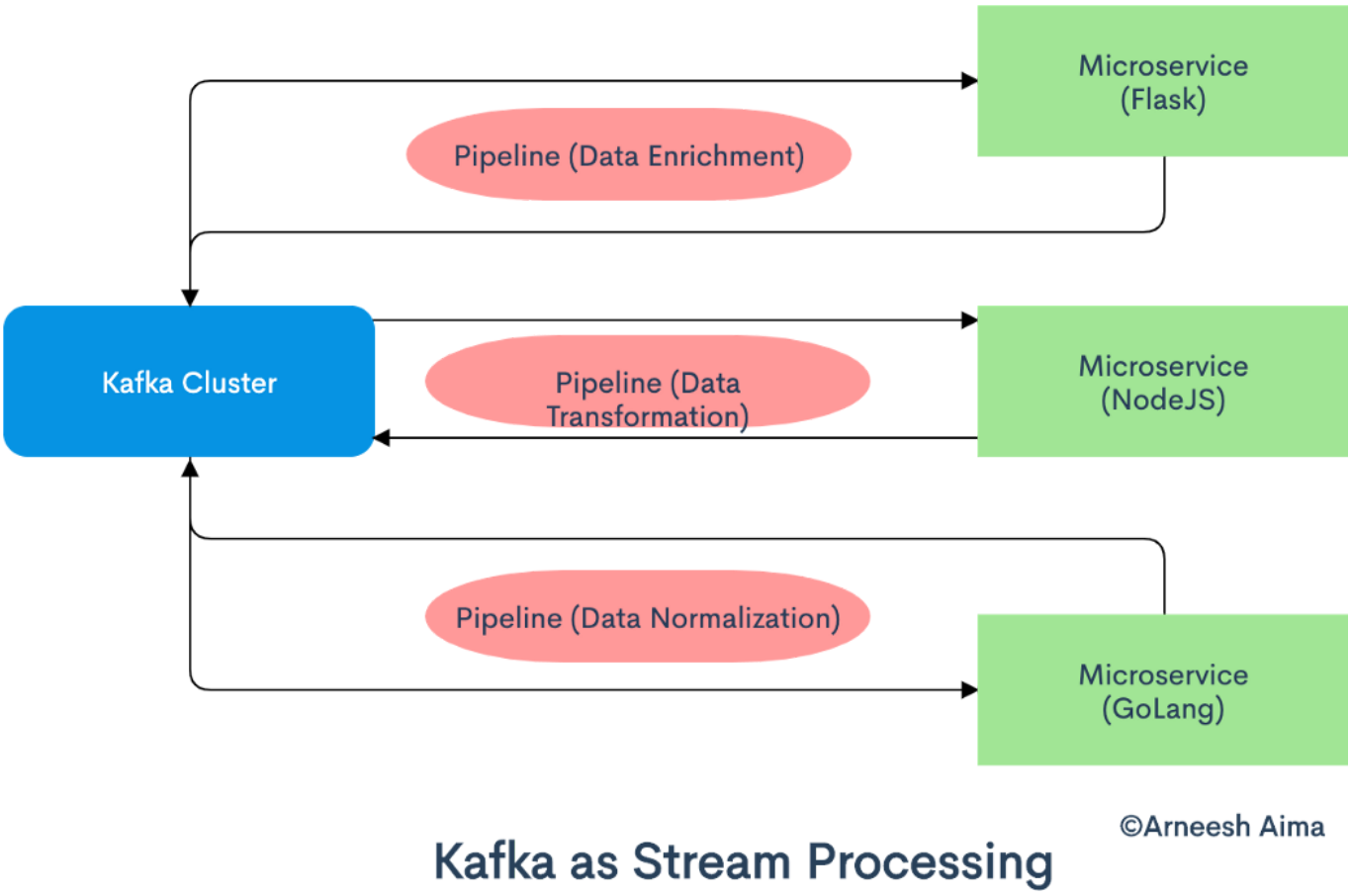
Empresas que utilizan Kafka para este fin

Honeywell, Uber.

como lo usan?

Las empresas suelen utilizar Kafka para enriquecer o transformar sus datos sin procesar. En este escenario, se usa Kafka Stream API y los datos se consumen de un tema, se someten a una transformación y luego se publican en otro tema de Kafka. Los datos agregados al segundo tema pueden enriquecerse aún más utilizando otra aplicación /

microservicio y enviarse de vuelta a otro tema de Kafka. Dicho ecosistema actúa como una infraestructura de canalización completa, ya que las operaciones de lectura / escritura se realizan con respecto a una única fuente / entidad de datos, es decir, Kafka, y no se requieren integraciones adicionales. Sus microservicios pueden estar en una amplia variedad de lenguajes que establecen las reglas para el enriquecimiento y la transformación de datos. Algunos de los lenguajes / frameworks más utilizados son el microservicio Javascript basado en entornos de nodo, el microservicio Python basado en Flask,



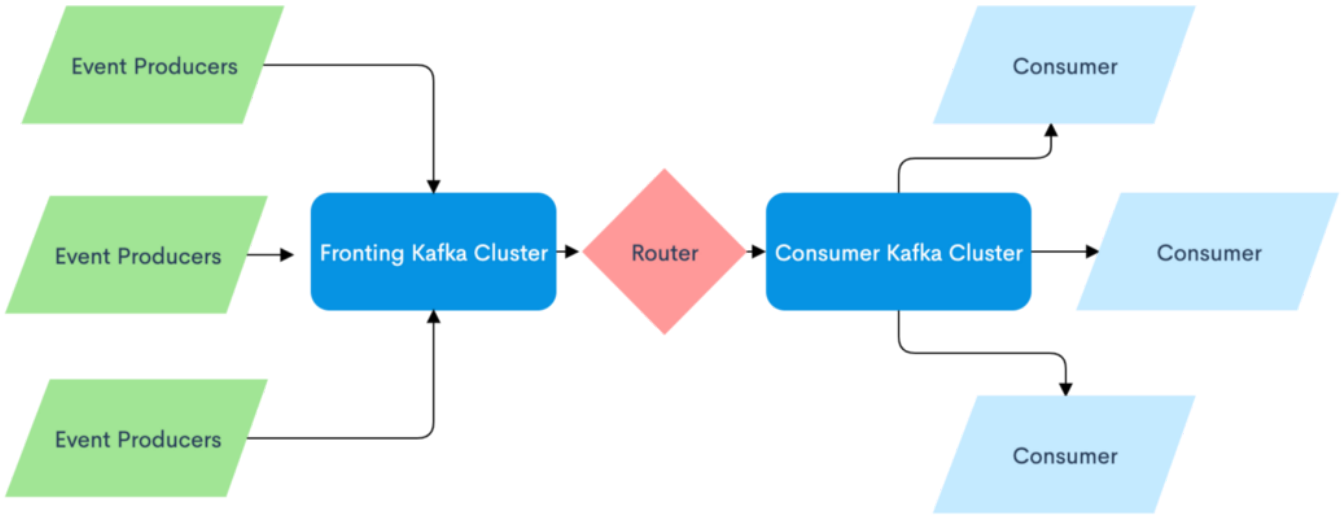
2. Kafka como sistema de almacenamiento

Empresas que utilizan Kafka para este fin

Netflix. Netflix usa Kafka como su bus de transmisión de mensajes / datos como parte de su infame canalización Keystone .

como lo usan?

Un sistema de almacenamiento activo es cualquier sistema que permita la publicación de datos desvinculada de su consumo. Kafka actúa como un excelente sistema de almacenamiento debido a su durabilidad y capacidad de tolerancia a fallas. Cada registro se escribe en el disco y se replica para tolerancia a fallas. Una escritura / publicación de un productor solo se completa cuando también se ha creado la replicación para los mismos datos. Esto se hace para garantizar que haya una copia de seguridad disponible en caso de que algo salga mal con un tema. El **almacenamiento de** datos **en Kafa es** independiente del **tamaño, es** decir, no importa cuál sea la cantidad de datos, ya sea un par de MB o en TB, la fluidez de la recuperación de información no se verá afectada siempre que se asignen buenos recursos físicos al sistema.



Kafka as Storage System

3. Kafka como sistema de mensajería / notificación

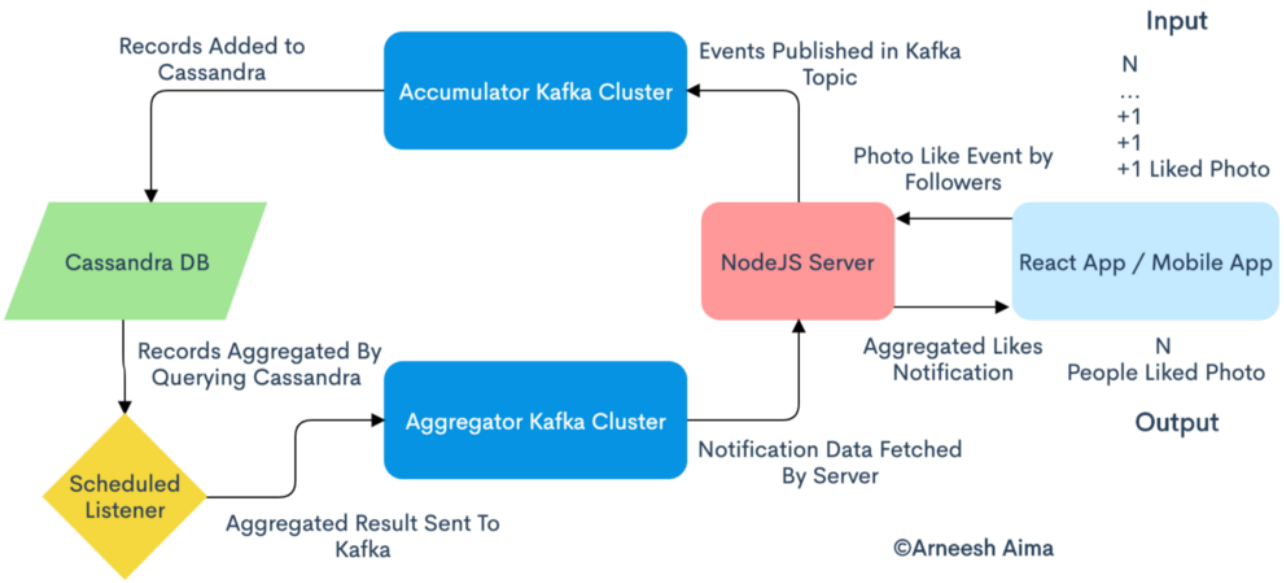
Empresas que utilizan Kafka para este fin

Instagram (Facebook), Twitter

como lo usan?

Muchas de estas empresas utilizan intermediarios de mensajes para generar solicitudes de Fan-out desde el servidor a sus clientes. El modelo Fanout sugiere la entrega de información a uno o varios usuarios en paralelo sin esperar un token ACK (Reconocimiento) de los usuarios. En este caso, Kafka se utiliza como intermediario de mensajes de acuerdo con una base de datos para almacenar feeds de actividades como

InfluxDB o Cassandra. Redis se usó mucho antes, pero ya no se recomienda, ya que cuando su proyecto crece en tamaño, mantener Redis se vuelve una tarea agitada y muy costosa. Incluso Facebook se mudó a Cassandra desde Redis para ahorrar dinero. Además, Redis no es tan robusto como Cassandra y tiene muchas limitaciones.



Kafka as Messaging/Notification System

4. Kafka como rastreador de actividad del sitio web

Empresas que utilizan Kafka para este fin

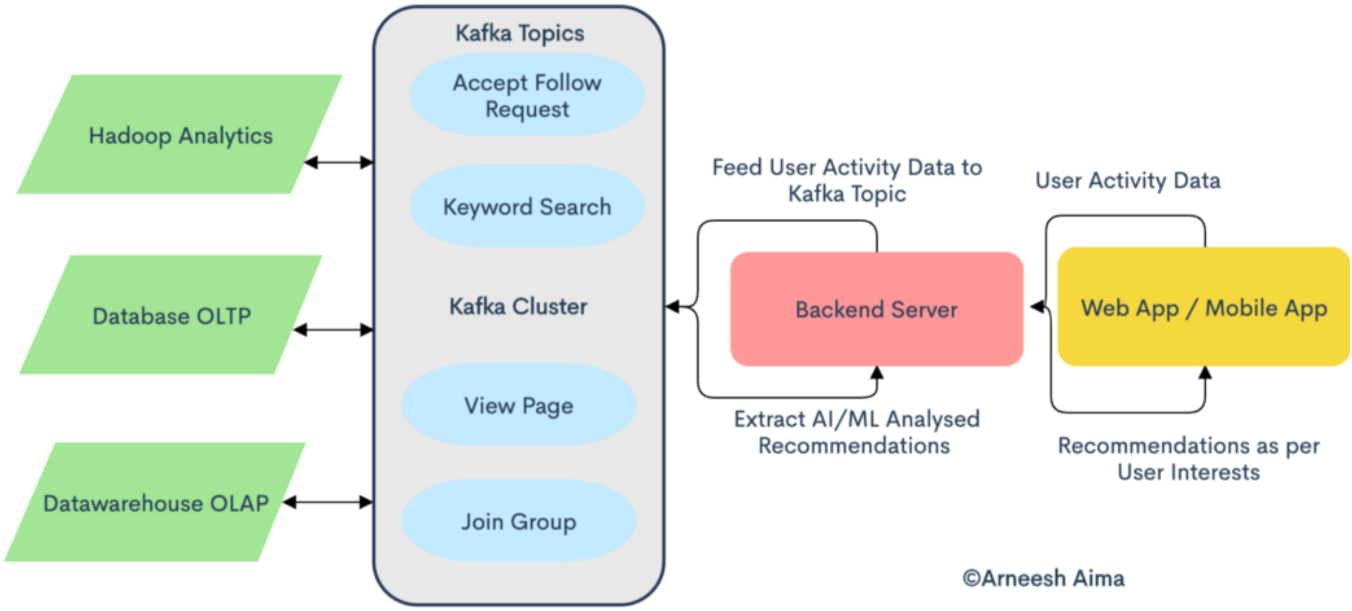
LinkedIn. Kafka fue creado inicialmente para este propósito por LinkedIn antes de que fuera entregado a Apache.

como lo usan?

Kafka se utiliza para el seguimiento de la actividad del sitio web mediante la configuración de feeds de publicación y suscripción en tiempo real. Se crea un clúster central de Kafka y los temas dentro de él se generan en función de la actividad, como "aceptar solicitud de seguimiento", "búsqueda de palabras clave", etc. Se crea un tema para todas y cada una de las actividades que se deben rastrear para el análisis de datos y el reconocimiento de patrones.

Kafka se creó cuando se requería un intermediario de mensajes que pudiera manejar la inmensa cantidad de seguimiento de actividades que comprende docenas de actividades

por parte de millones de usuarios. Un corredor de mensajes regular como RabbitMQ generalmente no puede escalar tanto y brindar una experiencia eficiente / fluida.



Kafka as Web Activity Tracker

Gracias !
Mi LinkedIn: Visítame en LinkedIn

Kafka Corrientes de Kafka Netflix Facebook LinkedIn