



GeeksHubs
academy _

Kafka

Herramientas de monitorización

Introducción

- ¿Qué es la monitorización?
- Monitorización con Xinfra-Monitor
- Herramientas de gestión
- API REST de Monitorización
- Introducción a Nagios
 - Monitorización de Kafka con Nagios
- Otras herramientas de monitorización



Monitorización



Monitorización

Es una actividad crítica que nos permite evaluar, analizar, gestionar y reportar tanto la disponibilidad como el rendimiento de servidores y aplicaciones. Esto nos permite obtener una imagen de la salud de nuestra infraestructura como identificar y analizar tendencias a corto y largo plazo.

Otras ventajas son el disponer de datos que nos permitan evaluar la capacidad de nuestra infraestructura, planificar la expansión de recursos o realizar auditorías.

Para poder monitorizar de forma fiable necesitamos contar con herramientas especializadas que cumplan con cuatro funciones básicas:

- Recopilar datos de forma fiable.
- Analizar esos datos en tiempo real.
- Un sistema de alertas en caso de desvío sobre ciertos parámetros.
- Ejecutar acciones automatizadas basadas en criterios definidos por el responsable de la infraestructura.



Monitorización

Hay dos tipos de monitorización de infraestructura: reactivo y proactivo.

- La monitorización reactiva nos alerta de fallos y errores. Aunque esto implica caídas en el servicio, la monitorización nos permite actuar rápidamente para reducirlas al mínimo.
- La monitorización proactiva, además de vigilar el rendimiento del sistema, comprueba si existen anomalías y condiciones que conduzcan a errores, para poder prevenirlos.



Monitorización

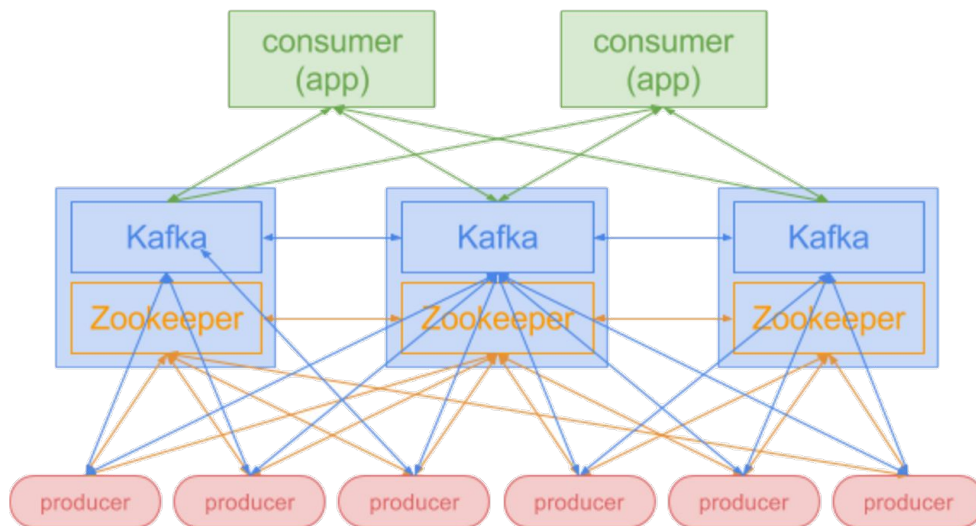
Antes de profundizar en el tema, le presentamos una terminología útil:

- Topic / Tema: Conjunto de mensajes o paquetes.
- Partition / Partición: Grupo de temas divididos por escalabilidad y redundancia.
- Producer / Productor: Proceso que introduce mensajes en la cola.
- Consumer / Consumidor: Proceso que se suscribe a varios temas y procesos a partir de una fuente de mensajes publicados.
- Broker: Nodo que forma parte del clúster Kafka



Monitorización

A continuación se muestra un diagrama de un clúster Kafka junto al conjunto Zookeeper necesario: 3 brokers Kafka más 3 servidores Zookeeper (2n+1 de redundancia) con 6 productores escribiendo en 2 particiones por redundancia.



Monitorización

Métricas del Sistema

Métrico	Comentarios	Alerta sugerida
Uso de memoria	Kafka debería ejecutarse completamente en RAM. El tamaño del montón de JVM no debería ser mayor que la RAM disponible. Eso es para evitar intercambiar.	Ninguno
Uso de intercambio	Esté atento al uso de swap, ya que degradará el rendimiento en Kafka y provocará que las operaciones se agoten (establezca <code>vm.swappiness = 0</code>).	Cuando se usa, el intercambio es > 128 MB.
Ancho de banda de la red	Los servidores de Kafka pueden incurrir en un uso elevado de la red. Esté atento a esto, especialmente si nota alguna degradación del rendimiento. También busque errores de paquetes descartados.	Ninguno
Uso del disco	Asegúrese de tener siempre espacio libre para nuevos datos, archivos temporales, instantáneas o copias de seguridad.	Cuando el disco tiene un uso > 85%.
E / S de disco	Las particiones de Kafka se almacenan de forma asincrónica como un registro de escritura anticipada secuencial. Por lo tanto, las lecturas y escrituras de disco en Kafka son secuenciales, con muy pocas búsquedas aleatorias.	Ninguno

Así es como, por ejemplo, Server Density va haciendo gráficas de algunas métricas de Kafka:



Monitorización

Métricas de Kafka

Estas son algunas de las muchas métricas de Kafka:

Métrica	Comentarios	Alerta sugerida
UnderReplicatedPartitions	kafka.server: type=ReplicaManager, name=UnderReplicatedPartitions - Número de particiones subreplicadas.	Cuando UnderReplicatedPartitions > 0.
OfflinePartitionsCount	kafka.controller: type=KafkaController, name=OfflinePartitionsCount - Número de particiones sin líder activo, por lo que no se pueden leer ni escribir.	Cuando OfflinePartitiosCount > 0.
ActiveControllerCount	kafka.controller: type=KafkaController, name=ActiveControllerCount - Número de brokers controladores activos.	Cuando ActiveControllerCount != 1.



Monitorización

Métricas del consumidor de Kafka

Estas son algunas de las métricas:

Métrico	Comentarios	Alerta sugerida
MaxLag	kafka.consumer: type = ConsumerFetcherManager, name = MaxLag, clientId = ([-. \ w] +) Número de mensajes por los cuales el consumidor va por detrás del productor.	Cuando MaxLag > 50.
MinFetchRate	kafka.consumer: type = ConsumerFetcherManager, name = MinFetchRate, clientId = ([-. \ w] +) Tasa mínima a la que el consumidor envía solicitudes al corredor de Kafka. Si está estancado o muerto, esto se reduce a 0.	Cuando MinFetchRate < 0.5.
MessagesPerSec	kafka.consumer: type = ConsumerTopicMetrics, name = MessagesPerSec, clientId = ([-. \ w] +) Mensajes consumidos por segundo.	Ninguno
BytesPerSec	kafka.consumer: type = ConsumerTopicMetrics, name = BytesPerSec, clientId = ([-. \ w] +) Bytes consumidos por segundo.	Ninguno
KafkaCommitsPerSec	kafka.consumer: type = ZookeeperConsumerConnector, name = KafkaCommitsPerSec, clientId = ([-. \ w] +) Tasa a la que el consumidor compromete compensaciones con Kafka.	Ninguno



Monitorización

En Kafka se utiliza Yammer Metrics para los informes de métricas en el servidor. Los clientes de Java utilizan Kafka Metrics, un registro de métricas integrado que minimiza las dependencias transitivas que se introducen en las aplicaciones cliente. Ambos exponen las métricas a través de JMX y pueden configurarse para informar de las estadísticas mediante reporteros de estadísticas enchufables para conectarse a su sistema de supervisión.

Todas las métricas de tasa de Kafka tienen su correspondiente métrica de recuento acumulado con el sufijo `-total`. Por ejemplo, `records-consumed-rate` tiene una métrica correspondiente llamada `records-consumed-total`.

La forma más fácil de ver las métricas disponibles es encender `jconsole` y apuntar a un cliente o servidor kafka en ejecución; esto permitirá navegar por todas las métricas con JMX.



Monitorización con Xinfra Monitor



Monitorización con Xinfra Monitor

Xinfra Monitor (antes Kafka Monitor) es un framework para implementar y ejecutar pruebas de sistema de Kafka de larga duración en un clúster real. Complementa las pruebas de sistema existentes de Kafka mediante la captura de posibles errores o regresiones que sólo es probable que ocurran después de un período prolongado de tiempo o con baja probabilidad. Además, permite monitorizar el clúster de Kafka utilizando pipelines de extremo a extremo para obtener una serie de estadísticas vitales derivadas como:

- Latencia de extremo a extremo
- Disponibilidad del servicio
- Disponibilidad de producción y consumo
- Disponibilidad de la confirmación del consumidor
- Latencia de confirmación del consumidor
- Tasa de pérdida de mensajes de Kafka
- Y muchos, muchos más.



Monitorización con Xinfra Monitor

Puede desplegar fácilmente Xinfra Monitor para probar y monitorear su cluster Kafka sin requerir ningún cambio en su aplicación.

Xinfra Monitor puede crear automáticamente el tema de monitorización con la configuración especificada y aumentar el recuento de particiones del tema de monitorización para garantizar que `partition# >= broker#`.

También puede reasignar la partición y activar la elección del líder preferido para asegurar que cada broker actúe como líder de al menos una partición del tema de monitorización.

Esto permite a Xinfra Monitor detectar problemas de rendimiento en cada corredor sin necesidad de que los usuarios gestionen manualmente la asignación de particiones del tema de monitorización.



Monitorización con Xinfra Monitor

Xinfra Monitor se utiliza junto con diferentes servicios de capa intermedia como li-apache-kafka-clients para monitorizar clústeres individuales, clústeres de destino de tuberías y otros tipos de clústeres como se hace en la ingeniería de LinkedIn para la comprobación de la salud del clúster en tiempo real.



Herramientas de gestión



Herramientas de Gestión

Kafka UI

UI for Apache Kafka es una interfaz web gratuita y de código abierto para supervisar y gestionar los clústeres de Apache Kafka.

Es una herramienta sencilla que hace que sus flujos de datos sean observables, ayuda a encontrar y solucionar problemas más rápidamente y ofrece un rendimiento óptimo. Su ligero panel de control facilita el seguimiento de las métricas clave de sus clústeres de Kafka: corredores, temas, particiones, producción y consumo.

Se configura la interfaz de usuario para Apache Kafka con sólo un par de sencillos comandos para visualizar sus datos de Kafka de forma comprensible. Se puede ejecutar la herramienta localmente o en la nube



Herramientas de Gestión

Kafka UI

The screenshot displays the 'UI for Apache Kafka' dashboard. On the left, a sidebar menu includes 'GENERAL' (with 'Dashboard' selected) and 'CLUSTERS' (with sub-items for 'secondLocal' and 'local', each containing 'Brokers', 'Topics', and 'Consumers'). The main content area, titled 'Clusters', shows a summary of 'ONLINE CLUSTERS' (1) and 'OFFLINE CLUSTERS' (0), with a toggle for 'HIDE ONLINE CLUSTERS'. Below this, two panels provide detailed metrics for the 'secondLocal' and 'local' clusters.

Cluster	Brokers	Partitions	Topics	Production	Consumption
secondLocal	1	1	1	NaN	NaN
local	1	2	2	NaN	NaN

Herramientas de Gestión

Conduktor

Conduktor es una herramienta que ofrece algo más que una interfaz sobre Apache Kafka, **proporciona el control de todo su pipeline de datos**, gracias a su integración con la mayoría de las tecnologías en torno a Apache Kafka.

Conduktor ayuda a las empresas y a los equipos a trabajar con fluidez con sus clusters de Apache Kafka y proporciona:

- Aplicación de escritorio fácil de usar.
- Inspección de datos en tiempo real.
- Seguro de que sus aplicaciones están funcionando bien.
- Portal empresarial para supervisar su uso.



Herramientas de Gestión

Conductor

Desarrolladores: Configurar temas, encontrar datos, gestionar consumidores, aplicaciones Kafka Streams, Conectores, ejecutar consultas ksqlDB.

Operadores: Configurar y supervisar sus corredores, reasignar particiones de temas, asegurar que las métricas son buenas.

Garantía de calidad: Ejecutar escenarios mediante el envío de mensajes personalizados para ejecutar pruebas funcionales. Probar la resistencia enviando datos aleatorios.

Empresas: Asegura que sus equipos trabajan con productos útiles. Permite que trabajen rápidamente con todo su ecosistema de Apache Kafka.



API REST de Monitorización



API REST de Monitorización

Kafka Connect

Es una herramienta para transmitir datos de forma escalable y fiable entre Apache Kafka y otros sistemas. Facilita la definición rápida de conectores que mueven grandes colecciones de datos dentro y fuera de Kafka. Puede ingerir bases de datos enteras o recopilar métricas de todos sus servidores de aplicaciones en temas de Kafka, haciendo que los datos estén disponibles para el procesamiento de flujos con baja latencia. Un trabajo de exportación puede entregar los datos de los temas de Kafka en sistemas de almacenamiento y consulta secundarios o en sistemas por lotes para el análisis fuera de línea.

Las características de Kafka Connect incluyen:

- Marco común para los conectores Kafka.
- Modos distribuidos y autónomos.
- Interfaz REST.
- Gestión automática de la compensación.



API REST de Monitorización

Kafka Connect

Una vez que un clúster de Kafka Connect está en funcionamiento, puede monitorizarlo y modificarlo. Vamos a ver algunas tareas de gestión comunes realizadas a través de la API REST.

Dado que Kafka Connect está pensado para ejecutarse como un servicio, también admite una API REST para gestionar los conectores. Por defecto, este servicio se ejecuta en el puerto 8083. Cuando se ejecuta en modo distribuido, la API REST será la interfaz principal del clúster. Puede realizar solicitudes a cualquier miembro del clúster; la API REST reenvía automáticamente las solicitudes si es necesario.

Aunque puede utilizar el modo autónomo simplemente enviando un conector en la línea de comandos, también ejecuta la interfaz REST. Esto es útil para obtener información de estado, añadir y eliminar conectores sin detener el proceso, y mucho más.



API REST de Monitorización

Kafka Connect

Uso de JMX para monitorizar Connect

Connect informa de una serie de métricas a través de Java Management Extensions (JMX). Connect puede configurarse para que informe de las estadísticas mediante reporteros de estadísticas enchufables adicionales utilizando la opción de configuración `metric.reporters`.

En la documentación [Usando JMX para monitorizar Connect](#) hay secciones donde se pueden encontrar MBeans, métricas y descripciones.



Introducción a Nagios



Introducción a Nagios

Originalmente llamado NetSaint y lanzado en 1999, Nagios fue desarrollado por Ethan Galstad y posteriormente perfeccionado por numerosos colaboradores como un proyecto de código abierto. Nagios Enterprises, una empresa basada en la tecnología Nagios Core, ofrece múltiples productos, como XI, Log Server, Network Analyzer y Fusion.

Nagios es un sistema de monitorización de código abierto para sistemas informáticos. Fue diseñado para ejecutarse en el sistema operativo Linux y puede supervisar dispositivos que ejecutan sistemas operativos (SO) Linux, Windows y Unix.

El software Nagios **realiza comprobaciones periódicas de los parámetros críticos de los recursos de la aplicación, la red y el servidor**. Por ejemplo, Nagios puede supervisar el uso de la memoria, el uso del disco, la carga del microprocesador, el número de procesos que se están ejecutando y los archivos de registro.



Introducción a Nagios

Un usuario puede elegir trabajar en la interfaz de línea de comandos (CLI) o seleccionar una interfaz gráfica de usuario (GUI) basada en la web en algunas versiones de Nagios y de terceros. El panel de control de Nagios proporciona una visión general de los parámetros críticos monitorizados en los activos.

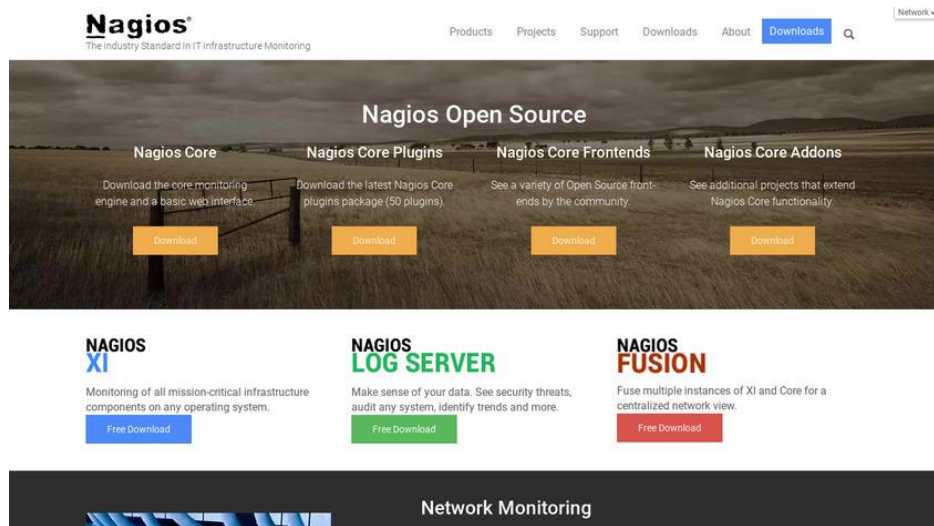
Basándose en los parámetros y umbrales definidos, Nagios puede enviar alertas si se alcanzan niveles críticos. Estas notificaciones se pueden enviar de diferentes maneras, incluyendo el correo electrónico y los mensajes de texto. Un sistema de autorización permite al administrador restringir el acceso.

Nagios funciona tanto con configuraciones basadas en agentes como sin agentes. Los agentes independientes se instalan en cualquier sistema de hardware o software para recopilar datos que luego se informan al servidor de gestión. La monitorización sin agente utiliza protocolos existentes para emular a un agente. Ambos enfoques pueden supervisar el uso del sistema de archivos, las métricas del sistema operativo, el servicio y los estados de los procesos y más.



Introducción a Nagios

Nagios también puede supervisar servicios, como el Protocolo Simple de Transferencia de Correo (SMTP), el Protocolo de Oficina Postal 3 (POP3), el Protocolo de Transferencia de Hipertexto (HTTP) y otros protocolos de red comunes. Las comprobaciones activas son iniciadas por Nagios, mientras que las pasivas provienen de aplicaciones externas conectadas a la herramienta de monitorización.



Introducción a Nagios

Nagios tiene varios productos, algunos conocidos, como:

- Nagios XI: Ofrece lo más potente de monitorización de la infraestructura de IT y de alertas de software de monitorización de IT para los exigentes requisitos organizativos de hoy en día. También proporciona a las organizaciones una visión ampliada de su infraestructura de IT antes de que los problemas afecten a los procesos críticos del negocio.
- Nagios Log Server: Potente aplicación de monitorización, gestión y análisis de registros de clase empresarial que permite a las organizaciones ver, consultar y analizar rápida y fácilmente los registros de todos los datos de registro generados por las máquinas.
- Nagios Core: Muy premiado motor que ha sido el estándar en la monitorización de la infraestructura de red durante más de una década, proporcionando un rendimiento y una flexibilidad sin precedentes.



Monitorización con Nagios

Nagios Core, uno de los productos que hemos visto anteriormente, originalmente se conocía como Nagios y es un software de monitorización de código abierto para sistemas de IT, redes e infraestructuras.

Nagios es el más popular para monitorización pues contiene una **amplia gama de monitorización de la infraestructura a través de permitir plug-ins para ampliar sus capacidades de monitoreo**. Además el personal de IT puede configurar Nagios para **supervisar los componentes críticos de la infraestructura**, incluidas las métricas del sistema, protocolos de red, aplicaciones, servicios, servidores e infraestructura de red.

Nagios envía alertas cuando los componentes críticos de la infraestructura fallan y se recuperan, proporcionando a los administradores un aviso de eventos importantes, y así el personal de IT puede reconocer las alertas y comenzar a resolver las interrupciones e investigar las alertas de seguridad inmediatamente.



Monitorización con Nagios

Los informes proporcionan un registro histórico de las interrupciones, los eventos, las notificaciones y la respuesta a las alertas para su posterior revisión. Los informes de disponibilidad ayudan a garantizar el cumplimiento de sus SLA.

El tiempo de inactividad programado evita las alertas durante las ventanas de mantenimiento y actualización programadas.

Los gráficos e informes de tendencias y planificación de la capacidad le permiten identificar las actualizaciones necesarias de la infraestructura antes de que se produzcan los fallos.

Además cuenta con varias API que se utilizan para ampliar sus capacidades para realizar tareas adicionales, se implementa como un servicio escrito en C por razones de rendimiento, y está diseñado para ejecutarse de forma nativa en los sistemas Linux / *nix.



Monitorización de Kafka con Nagios



Monitorización con Nagios

La implementación de una monitorización JMX efectiva con Nagios ofrece los siguientes beneficios:

- Mayor disponibilidad de servidores, servicios y aplicaciones.
- Detección rápida de cortes de red y fallos de protocolo.
- Detección rápida de procesos, servicios y trabajos por lotes fallidos.

A continuación veremos distintas herramientas con las que se suele monitorizar Kafka con Nagios.



Monitorización con Nagios

JMXEval

Es un plugin de Nagios realmente flexible para la monitorización de aplicaciones Java mediante JMX. En lugar de comprobar sólo un atributo de un MBean, JMXEval le permite consultar múltiples atributos, y también realizar cálculos matemáticos para obtener cifras mucho más significativas para la supervisión.

JMXEval también puede proporcionar toda la información capturada de los atributos, así como cualquier cifra calculada como datos de rendimiento, lo que le permite capturar y visualizar las tendencias en Nagios.



Monitorización con Nagios

Nagios XI

Como hemos visto anteriormente, Nagios XI es de las herramientas de supervisión de infraestructuras más potentes y fiables del mercado por lo que ofrece lo más potente de monitorización y de alertas de software de monitorización.

Proporciona la supervisión de todos los componentes de la infraestructura de misión crítica, incluyendo aplicaciones, servicios, sistemas operativos, protocolos de red, métricas de sistemas e infraestructura de red. Cientos de complementos de terceros permiten supervisar prácticamente todas las aplicaciones, servicios y sistemas internos.

La interfaz de configuración integrada basada en la web permite a los administradores ceder el control de la gestión de la configuración de la monitorización, los ajustes del sistema, etc., a los usuarios finales y a los miembros del equipo con facilidad. Los asistentes de configuración guían a los usuarios a través del proceso de monitorización de nuevos dispositivos, servicios y aplicaciones, todo ello sin tener que entender complejos conceptos de monitorización.



Monitorización con Nagios

Nagios Core

El personal de TI configura Nagios para supervisar los componentes críticos de la infraestructura de TI, incluyendo las métricas del sistema, los protocolos de red, las aplicaciones, los servicios, los servidores y la infraestructura de red.

Además envía alertas cuando los componentes críticos de la infraestructura fallan y se recuperan, proporcionando a los administradores un aviso de eventos importantes. Las alertas se pueden enviar por correo electrónico, SMS o script personalizado.

Los informes proporcionan un registro histórico de las interrupciones, los eventos, las notificaciones y la respuesta a las alertas para su posterior revisión.



Otras herramientas de Monitorización



Herramientas de Monitorización

JmxTool

Es una aplicación independiente que imprime las métricas JMX de un endpoint dado (por ejemplo, el broker Kafka) a la salida estándar cada intervalo de informe (a menos que se especifique la opción de una sola vez).

JmxTool puede ser ejecutado usando el script de shell `kafka-run-class`.

En general, JmxTool es un gadget que entra en la comunidad. Para escenarios de monitoreo simples generales también puede hacer frente, pero tiene con una funcionalidad limitada, pues ofrece soluciones generales en problemas de monitoreo complejo.

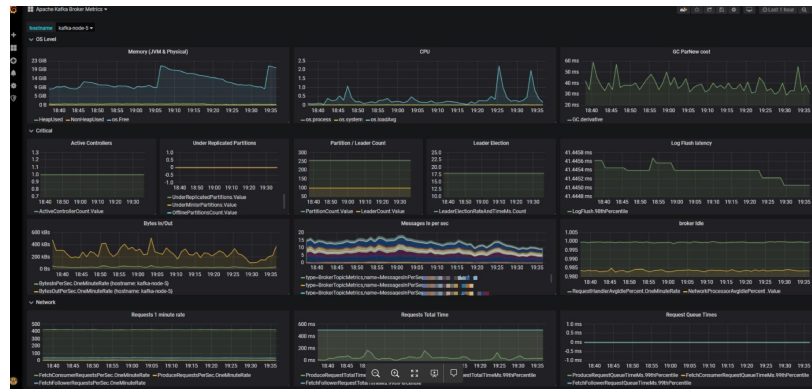


Herramientas de Monitorización

JMXTrans + InfluxDB + Grafana

Actualmente es una práctica popular monitorear en un conjunto de marcos de monitoreo universales, como la combinación de JMXTRANS + influnsp + grafana. Dado que Grafana admite el monitoreo de los indicadores JMX, es fácil integrar los distintos indicadores JMX de Kafka.

En comparación con el administrador de Kafka, la ventaja de este marco de monitoreo es que puede monitorear múltiples componentes técnicos clave en un marco de vigilancia. En particular para las empresas que han construido la cartera de monitoreo, reutilizar directamente este marco pueden ahorrar en gran medida los costos de operación y mantenimiento, y es una buena opción.



Herramientas de Monitorización

Otras herramientas

La primera es **check_kafka.pl** de Hari Sekhon. Realiza una prueba completa de extremo a extremo, es decir, inserta un mensaje en Kafka como productor y luego lo extrae como consumidor. Esto nos facilita la vida a la hora de medir los tiempos de servicio.

Otra herramienta útil es **KafkaOffsetMonitor** para monitorizar los consumidores de Kafka y su posición (offset) en la cola. Nos ayuda a entender cómo crece nuestra cola y qué grupos de consumidores se están quedando atrás.

Por último, pero no menos importante, la gente de LinkedIn ha desarrollado lo que se cree que es la herramienta más inteligente que existe: **Burrow**. Analiza los desplazamientos y retrasos de los consumidores durante un periodo de tiempo y determina el estado del consumidor. Puedes recuperar este estado a través de un punto final HTTP y luego conectarlo a tu herramienta de monitorización favorita.

