Kafka

Day 1

Day 1 - Overview

- Kafka Introduction
- Kafka Features
- Kafka Use Cases
- Kafka Architectures and Elements[Consumer, Producer, Broker, Zookeeper, Cluster]
- Kafka Overall Messaging Architecture
- Kafka Streaming Architecture
- Kafka vs AMQP
- Understanding Topics
- Understanding Partitions
- Understanding Replications

What is Kafka?

Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.

Kafka is used for real-time streams of data, to collect big data, or to do real time analysis (or both). Kafka is used with microservices to provide durability and it can be used to feed events to CEP (complex event streaming systems) and IoT-style automation systems.

Kafka's growth is exploding. More than one-third of all Fortune 500 companies use Kafka. These companies include the top ten travel companies, seven of the top ten banks, eight of the top ten insurance companies, nine of the top ten telecom companies, and much more. LinkedIn, Microsoft, and Netflix process four-comma messages a day with Kafka (1,000,000,000,000).

Kafka Features

Scalability: can handle scalability in all the four dimensions, i.e. event producers, event processors, event consumers and event connectors. In other words, Kafka scales easily without downtime.

High-Volume: Can work with the huge volume of data streams, easily.

Data Transformations: Kafka offers provision for deriving new data streams using the data streams from producers.

Fault Tolerance: The Kafka cluster can handle failures with the masters and databases.

Reliability: Since Kafka is distributed, partitioned, replicated and fault tolerant, it is very Reliable.

Durability: It is durable because Kafka uses Distributed commit log, that means messages persists on disk as fast as possible.

Performance: For both publishing and subscribing messages, Kafka has high throughput. Even if many TB of messages is stored, it maintains stable performance.

Replication: By using ingest pipelines, it can replicate the events.

Kafka Use Cases

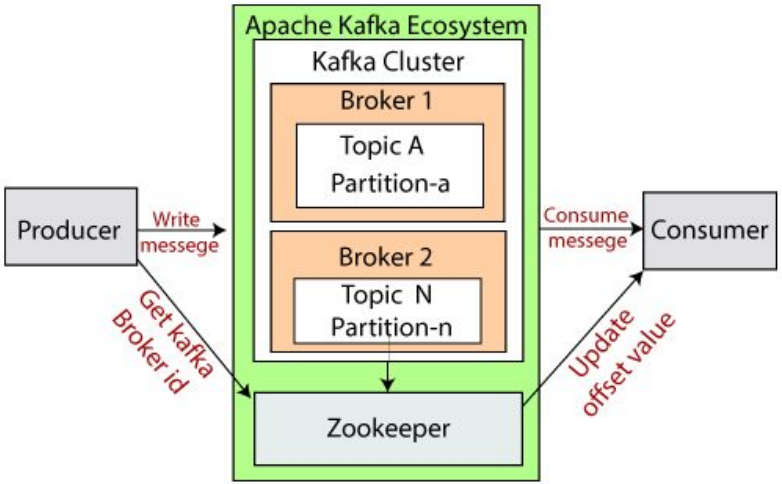
Activity tracking This was the original use case for Kafka. LinkedIn needed to rebuild its user activity tracking pipeline as a set of real-time publish-subscribe feeds.

Real-time data processing: Many systems require data to be processed as soon as it becomes available. Kafka transmits data from producers to consumers with very low latency (5 milliseconds, for instance). IOT

Messaging: Traditional Messaging platform

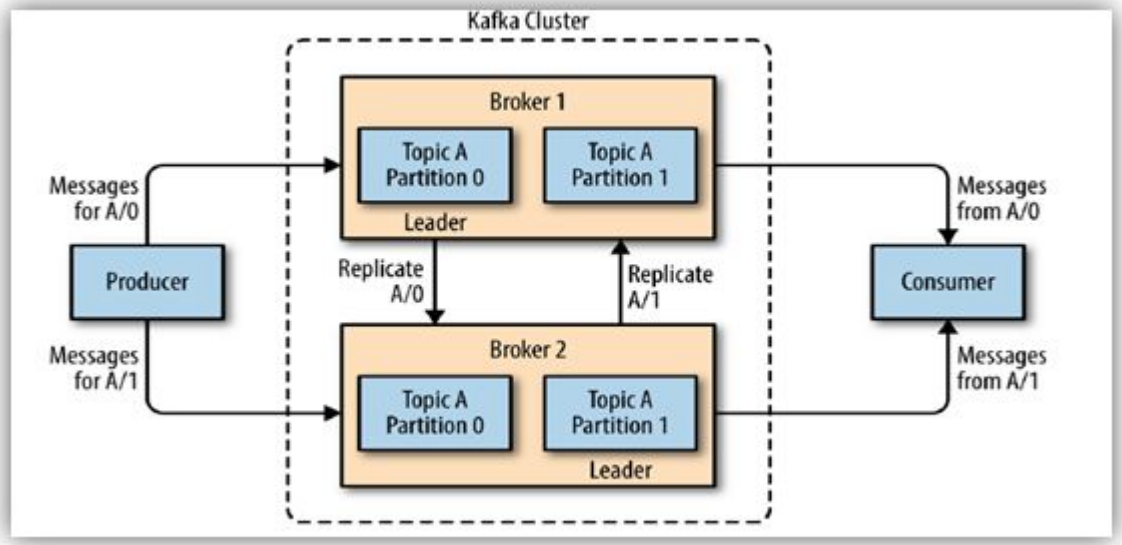
Log Aggregation: Log aggregation for distributed microservices platforms.

Kafka Architecture and Components



Apache Kafka Architecture

Overall Messaging Architecture



Kafka vs ActiveMQ

Active MQ is one of the traditional message brokers, whose goal is to ensure data exchange between applications in a safe and reliable way. It deals with a small amount of data and is therefore specialized for well-defined message formats and transactional messaging.

Kafka is a distributed system meant for processing a huge amount of data.

Kafka vs ActiveMQ

Message Format and Processing

Active MQ supports a JMS standard message format consisting of headers, properties, and the body

Kafka does not define any message format — this is entirely the responsibility of the producer.

Push vs Pull

Active MQ is push based. Producer has to make sure message is delivered

Kafka is pull based. Consumer has to poll the topic for new message

Q and A