Kafka can connect to an external system through Kaffka connect and provides Kafka streams (Java Streaming Library)

Kafka has 5 Core API for Java and Scala

1. Admin API
2. Producer API
3. Consumer API
4. Kafka Streams API
5. Kafka Connect API

Why Kafka?

1. It supports integration with different languages and concepts like spark,scala,Hadoop and big data.
2. It has cluster design to transfer data between multiple complex systems.
3. Integrates with non-java applications as well
4. It supports protocol TCP,FTP,HTTP
5. Supports multiple messaging brokers(Horizontal scaling of broker software is possible
6. It takes support of Zookeeper for load balancing( like Netflix Eureka in Microservices)

Benefits?

1. Kafka offers fast delivery. Single kafka broker can serve thousands of clients by handling MB of reads and writes per second.
2. Messages are replicated in clusters
3. There is provision of data partitioning and streamlining over cluster of machines to handle large data.
4. Provides fault tolerance mechanisam and durability
5. Kafka server can receive is 1000000 bytes

Components of Kafka

Producer

Ppublishes messages to topic

Producers helps to serialize , compress ,load balance data amoung brokers through partitioning

Topic

Broker

Consumer

Zookeeper :

It tracks the status of nodes in the kafka cluster and also maintains kafka list of kafka topic and messages.

Zookeeper also accesses how much data each client is allowed to read /write.

Message Broker

* Also known as Integration Broker or Interface Engine
* Translates message from formal messaging protocol to sender to the formal messaging protocol of the receiver.

Kafka offers much higher performance than message brokers like RabbitMQ.

* Uses sequential disk I/O to boost performance
* Millions of messages per second with limited resources .

Use Case 1

Diagram

Description automatically generated

One topic and multiple partitions

Single consumer taking messages from these 3 partitions

Use Case 2 :

Diagram

Description automatically generated

Use case 3 :

Diagram

Description automatically generated

User case 4

Diagram

Description automatically generated

Kafka Rule : We can have maximum consumers in the consumer group to the number of consumers with respect to the maximum number of partitions you have.

If one consumer is down it immediately transfer over to the remaining consumers

Use case 5

A picture containing diagram

Description automatically generated

If we open another consumer group to the same topic. First consumer group has 3 consumers and second consumer group has 2 consumers. So each consumer has each partition. So how consumer knows where do I take the messages, so the Kafka support retention. So there is a mechanism they maintain called consumer offset

**What is Consumer Offset?**

The consumer offset is a way of tracking the sequential order in which messages are received by Kafka topics. Keeping track of the offset, or position, is important for nearly all Kafka use cases and can be an absolute necessity in certain instances, such as financial services.

Based on the consumer offset, Kafka knows particular consumer need to go to assigned partition respectively.

*Rule: Number of consumers in the consumer group should be equal or less than to the number of partition in the topic*