

# **KAFKA**

## **What is Kafka?**

*Apache Kafka is a distributed publish-subscribe messaging system and a robust queue that can handle a high volume of data and enables you to pass messages from one end-point to another. Kafka is suitable for both offline and online message consumption. Kafka messages are persisted on the disk and replicated within the cluster to prevent data loss. Kafka is built on top of the ZooKeeper synchronization service. It integrates very well with Apache Storm and Spark for real-time streaming data analysis.*

## **Benefits:**

*Reliability – Kafka is distributed, partitioned, replicated and fault tolerance.*

*Scalability – Kafka messaging system scales easily without down time..*

*Durability – Kafka uses Distributed commit log which means messages persists on disk as fast as possible, hence it is durable..*

*Performance – Kafka has high throughput for both publishing and subscribing messages. It maintains stable performance even many TB of messages are stored.*

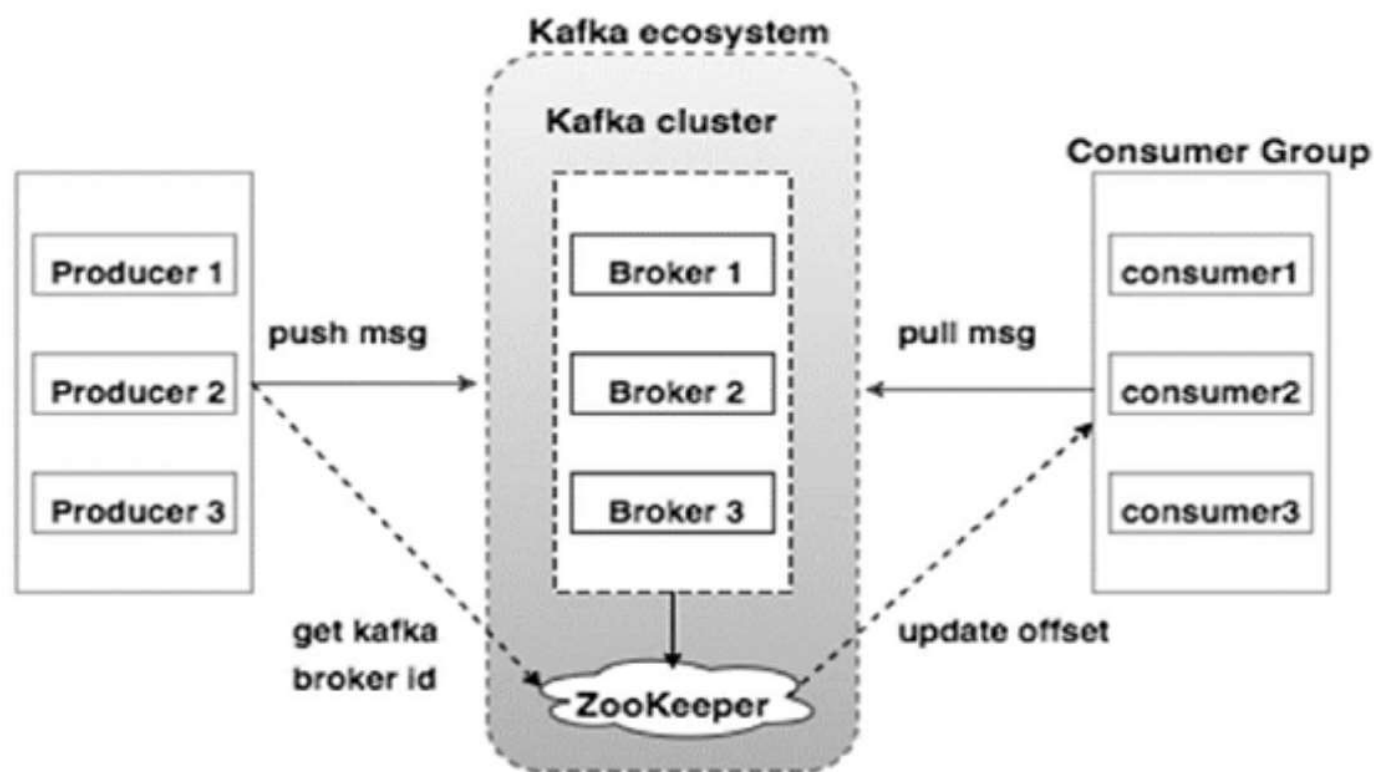
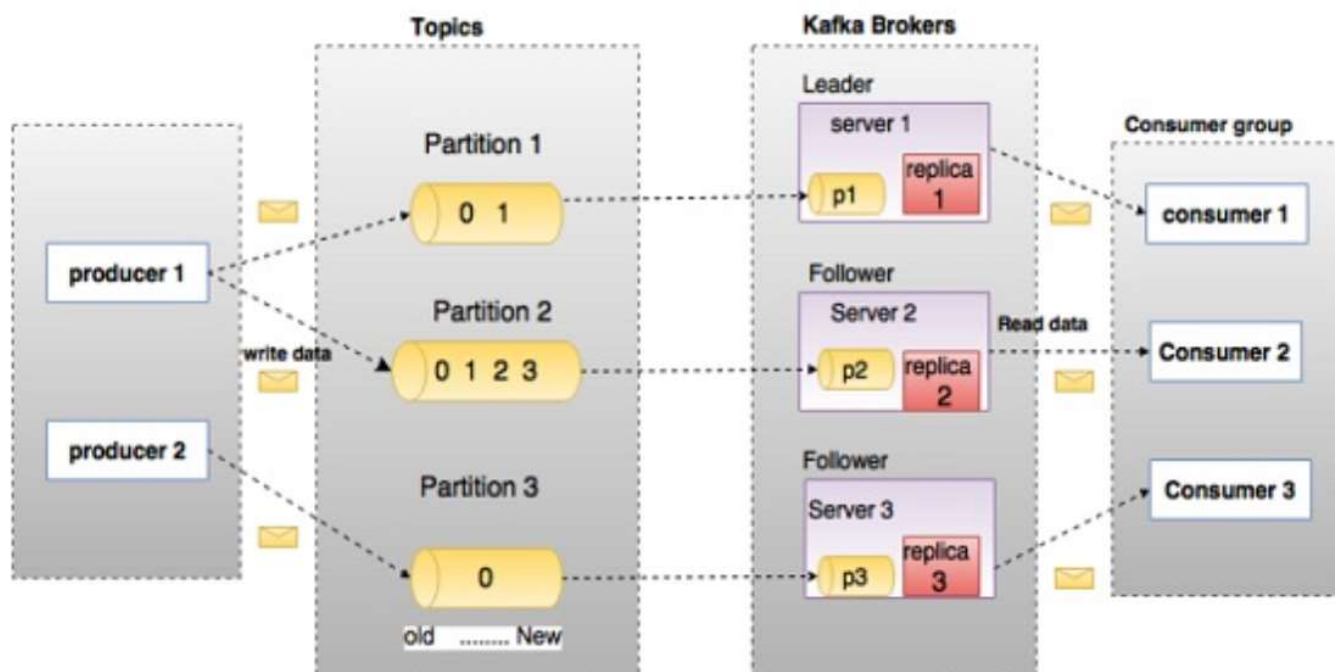
*Kafka is very fast and guarantees zero downtime and zero data loss.*

## **Use Cases :**

**Metrics** – *Kafka is often used for operational monitoring data. This involves aggregating statistics from distributed applications to produce centralized feeds of operational data.*

**Log Aggregation Solution** – *Kafka can be used across an organization to collect logs from multiple services and make them available in a standard format to multiple consumers.*

**Stream Processing** – *Popular frameworks such as Storm and Spark Streaming read data from a topic, processes it, and write processed data to a new topic where it becomes available for users and applications. Kafka's strong durability is also very useful in the context of stream processing.*



| S.No | Components and Description   |
|------|--|
| 1    | <p><b>Broker</b></p> <p>Kafka cluster typically consists of multiple brokers to maintain load balance. Kafka brokers are stateless, so they use ZooKeeper for maintaining their cluster state. One Kafka broker instance can handle hundreds of thousands of reads and writes per second and each broker can handle TB of messages without performance impact. Kafka broker leader election can be done by ZooKeeper.</p>  |
| 2    | <p><b>ZooKeeper</b></p> <p>ZooKeeper is used for managing and coordinating Kafka broker. ZooKeeper service is mainly used to notify producer and consumer about the presence of any new broker in the Kafka system or failure of the broker in the Kafka system. As per the notification received by the Zookeeper regarding presence or failure of the broker then producer and consumer takes decision and starts coordinating their task with some other broker.</p>  |
| 3    | <p><b>Producers</b></p> <p>Producers push data to brokers. When the new broker is started, all the producers search it and automatically send a message to that new broker. Kafka producer doesn't wait for acknowledgements from the broker and sends messages as fast as the broker can handle.</p>  |
| 4    | <p><b>Consumers</b></p> <p>Since Kafka brokers are stateless, which means that the consumer has to maintain how many messages have been consumed by using partition offset. If the consumer acknowledges a particular message offset, it implies that the consumer has consumed all prior messages. The consumer issues an asynchronous pull request to the broker to have a buffer of bytes ready to consume. The consumers can rewind or skip to any point in a partition simply by supplying an offset value. Consumer offset value is notified by ZooKeeper.</p> |

## ***To Run Zoo Keeper :***

### ***Step1 :Start the zookeeper server***

*C:\kafka\_2.13-3.5.1\bin\windows\zookeeper-server-start.bat C:\kafka\_2.13-3.5.1\config\zookeeper.properties*

### ***Step2 : Starting Kafka Broker***

*C:\kafka\_2.13-3.5.1\bin\windows\kafka-server-start.bat C:\kafka\_2.13-3.5.1\config\server.properties*

### ***Step 3: Creating my topic***

*C:\kafka\_2.13-3.5.1\bin\windows\kafka-topics.bat --bootstrap-server localhost:9092 --topic kaninitopic --partitions 3 --replication-factor 1 --create*

```
Command Prompt - C:\kafka_ x + v
[2023-08-16 12:15:33,117] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-08-16 12:15:33,117] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-08-16 12:15:33,117] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-08-16 12:15:33,118] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-08-16 12:15:33,118] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2023-08-16 12:15:33,144] INFO ServerMetrics initialized with provider org.apache.zookeeper.metrics.impl.DefaultMetricsProvider@4b14c583 (org.apache.zookeeper.server.ServerMetrics)
[2023-08-16 12:15:33,148] INFO zookeeper.snapshot.trust.empty : false (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2023-08-16 12:15:33,162] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO ----- (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO |___ / | | (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO / / ___ ___ | | __ ___ ___ ___ ___ (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO / / / _ \ / _ \ | / / / _ \ / _ \ | ' _ \ / _ \ | ' _ \ (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO / / __ | ( ) | ( ) | | < | __ / | __ / | ( ) | | __ / | | (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO / ____ | \___/ \___/ | _\|_ \ \___/ \___/ | _ \ / \___/ \___/ | _ \ (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO | | (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO |_ | (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:33,163] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:38,182] INFO Server environment:zookeeper.version=3.6.4--d65253dcf68e9097c6e95a126463fd5fdeb4521c, built on 12/18/2022 18:10 GMT (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:38,183] INFO Server environment:host.name=LAPTOP-2633JJUA (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:38,183] INFO Server environment:java.version=20.0.2 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-08-16 12:15:38,183] INFO Server environment:java.vendor=Oracle Corporation (org.apache.zookeeper.server.ZooKeeperServer)
```