## Why use kafka or why KaFka is better than RabbitMQ?

Kafka also clearly outshines RabbitMQ in performance. A single Kafka instance can handle 100K messages per second, versus closer to 20K messages per second for RabbitMQ. Kafka is also a good choice when you want to transmit messages at low latency to support batch consumers, assuming that the consumers could be either online or offline.

## Also Kafka is better because it make backup when partition is damaged it was make another backup fot it so which means “Fault tolerance”

## What is Kafka ? kafka is an open source by linked in that you can reuse its in messaging queue

## Kafka is light and its performance is high

## What are the Kafka components or main components for Kafka ?

## We have producers ,and consumers ,and topics which contains the portions ,and Kafka servers or broker “the main component”

**Topics**

A stream of messages belonging to a particular category is called a topic. Data is stored in topics.

Topics are split into partitions. For each topic, Kafka keeps a mini-mum of one partition. Each such partition contains messages in an immutable ordered sequence. A partition is implemented as a set of segment files of equal sizes.

**Partition**

Topics may have many partitions, so it can handle an arbitrary amount of data.

|  |  |
| --- | --- |
| **Partition offset**  Each partitioned message has a unique sequence id called as offset. | |
|  | |
| **Replicas of partition**  Replicas are nothing but backups of a partition. Replicas are never read or write data. They are used to prevent data loss. |
|  | |

**Brokers**

* Brokers are simple system responsible for maintaining the pub-lished data. Each broker may have zero or more partitions per topic

## 

**Kafka Cluster**

Kafka’s having more than one broker are called as Kafka cluster. A Kafka cluster can be expanded without downtime. These clusters are used to manage the persistence and replication of message data.

**Producers**

Producers are the publisher of messages to one or more Kafka topics. Producers send data to Kafka brokers. Every time a producer pub-lishes a message to a broker, the broker simply appends the message to the last segment file. Actually, the message will be appended to a partition. Producer can also send messages to a partition of their choice.

**Consumers**

Consumers read data from brokers. Consumers subscribes to one or more topics and consume published messages by pulling data from the brokers.

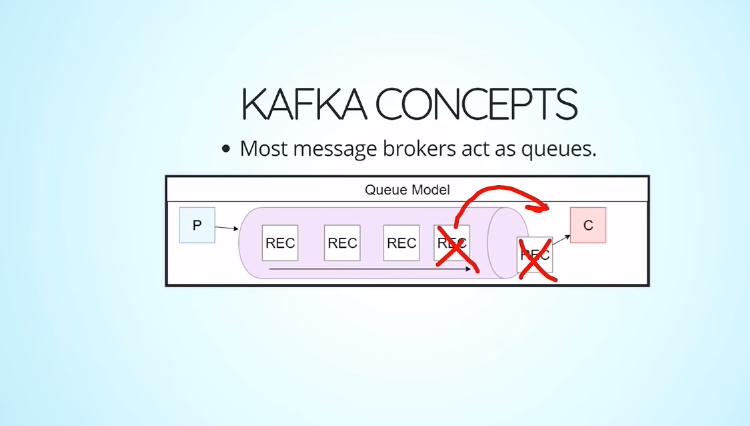
**Leader**

Leader is the node responsible for all reads and writes for the given partition. Every partition has one server acting as a leader.

**Follower**

Node which follows leader instructions are called as follower. If the leader fails, one of the followers will automatically become the new leader. A follower acts as normal consumer, pulls messages and up-dates its own data store

Once the kafka message has been consumed it disappear from queue “FIFO”

First in first out 

Also there is another concept which name consumer group => group of consumers which perform as an one consumer?

What are the brothers or compittors of kafka ?

There are 1) MSMQ from Microsoft

2) RabbitMQ

3) Celery :distributed queue built for python

**How to start ?**

First you require to download Kafka project and start zookeeper ?

What is the zookeeper ?

Zookeeper is an connector or updateable connector which get the new partitions from topics and send it to consumer so its role like bridge

## And follow this link<https://kafka.apache.org/quickstart>