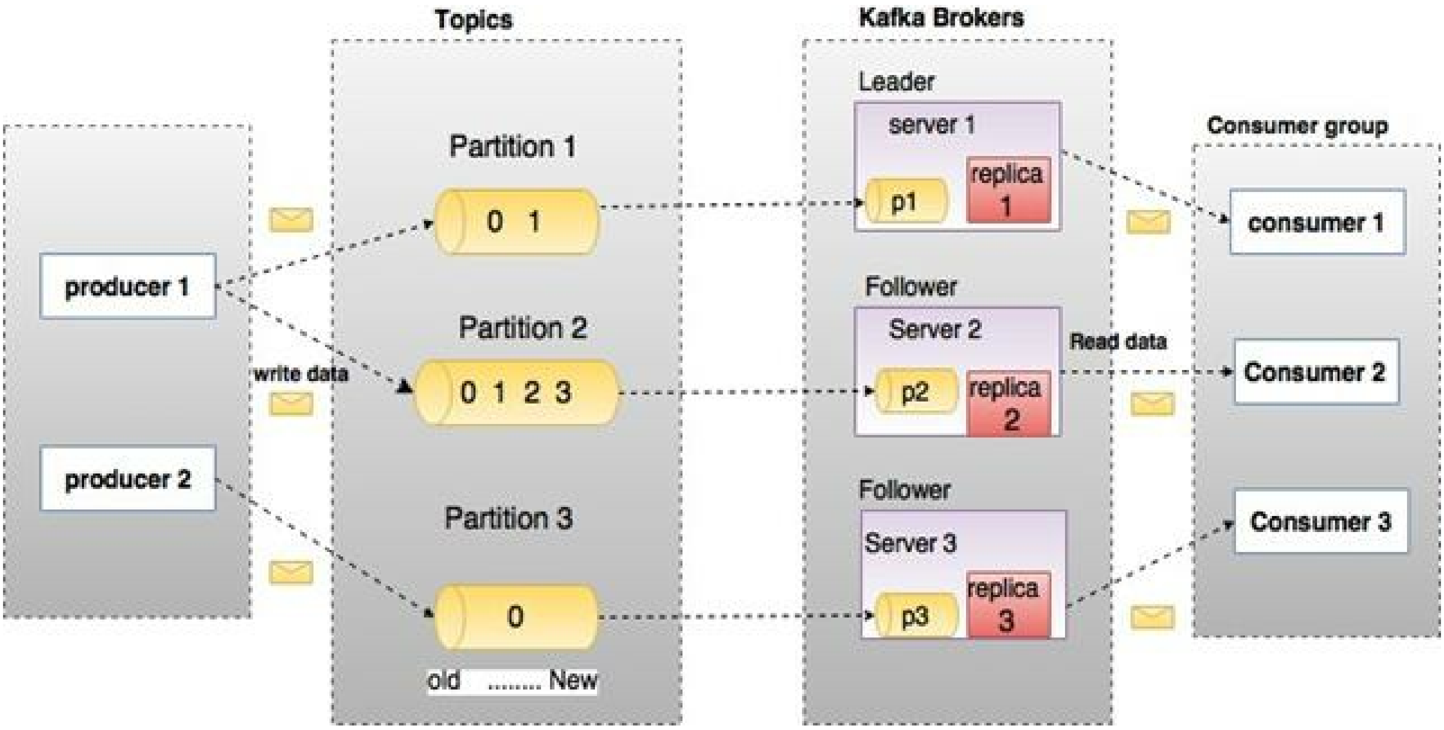
# Kafka Fundamentals

Before moving deep into the Kafka, you must aware of the main terminologies such as topics, brokers, producers and consumers. The following diagram illustrates the main terminologies and the table describes the diagram components in detail.



In the above diagram, a topic is configured into three partitions. Partition 1 has two offset factors 0 and 1. Partition 2 has four offset factors 0, 1, 2, and 3. Partition 3 has one offset factor 0. The id of the replica is same as the id of the server that hosts it.

Assume, if the replication factor of the topic is set to 3, then Kafka will create 3 identical replicas of each partition and place them in the cluster to make available for all its operations. To balance a load in cluster, each broker stores one or more of those partitions. Multiple producers and consumers can publish and retrieve messages at the same time.

| Components | Description |
|---|---|
| Topics | A stream of messages belonging to a particular category is called a topic. Data is stored in topics |
| Partition | Topics are split into partitions. For each topic, Kafka keeps a minimum 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. 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 | * Are simple system responsible for maintaining the published data. Each broker may have zero or more partitions per topic. Assume, if there are N partitions in a topic and N number of brokers, each broker will have one partition. * Assume if there are N partitions in a topic and more than N brokers (n + m), the first N broker will have one partition and the next M broker will not have any partition for that particular topic. * Assume if there are N partitions in a topic and less than N brokers (n-m), each broker will have one or more partition sharing among them. This scenario is not recommended due to unequal load distribution among the broker. |
| Kafka Cluster | 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 | Are the publisher of messages to one or more Kafka |

| | |
|---|---|
| | topics. Producers send data to Kafka brokers. Every time a producer publishes 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 | Read data from brokers. Consumers subscribes to one or more topics and consume published messages by pulling data from the brokers. |
| 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 follower will automatically become the new leader. A follower acts as normal consumer, pulls messages and up- dates its own data store.