

# Apache Kafka Concepts Condensed In A Diagram

 [mycloudreference.com/apache-kafka-concepts-condensed-in-a-diagram/](https://mycloudreference.com/apache-kafka-concepts-condensed-in-a-diagram/)

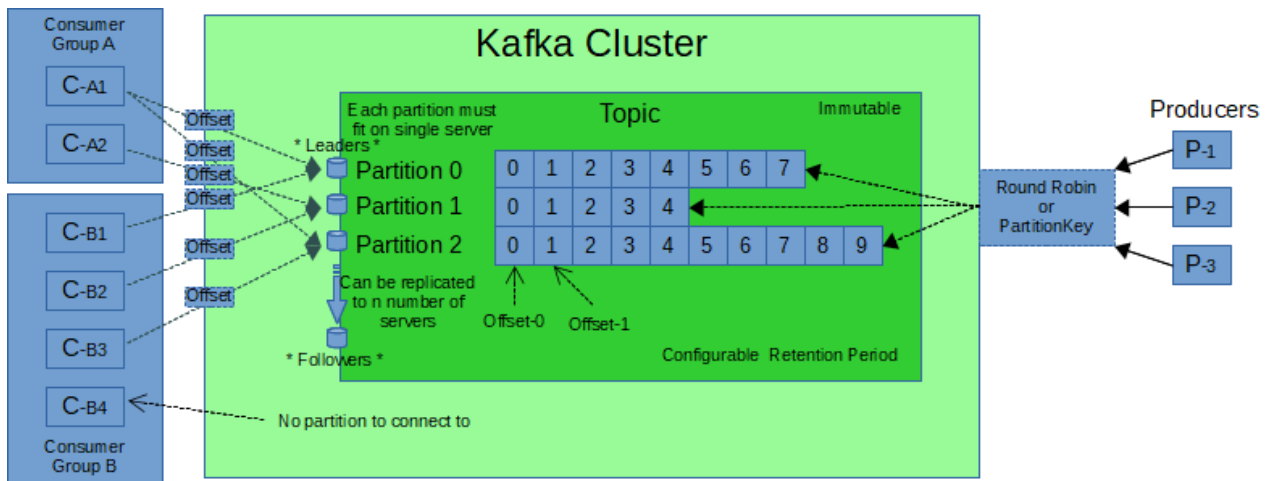
Jerry  
Chong



Apache Kafka is a distributed streaming platform that was first developed by LinkedIn and open-sourced in 2011. The core of the technology is based on the commit log of messages. It allows publishers to send messages to these commit logs and subscribers to read these messages without having any direct impact on each other.

You can read in-depth about this messaging system on the [official website](#). However, if you would like a quick condensed view of the concepts, I have included a single diagram created with [LibreOffice Draw](#) which I used and added on overtime.

The various components and terms are described below.



- The stream of different types of records flows through topics.
- Each topic can be further divided into partitions to increase throughput and/or to provide data sharding.
- Records produced by producers are sent to the topic and distributed among the partitions in a round-robin fashion or through the use of a partition key for sharding.
- Records stored in topics are immutable (cannot be changed or deleted)
- Records are only removed from the topic if they live past the configurable retention period.
- A workaround to remove records is to reduce the retention period significantly, let the system remove the record and then increase the retention period again. This is typically done in a non-production scenario where the topic needs to be drained for various reasons.
- It is possible for partitions in a topic having a varying number of records.
- The partitions can be replicated to n number of servers for redundancy.
- Each set of replicated partitions will have a leader while the others will act as a follower.
- Each partition (replicated or not) must fit on a single physical server.
- Each record stored on a partition has an offset number assigned to it.
- Consumers read records from the topic by connecting as part of a logical consumer group to a partition or multiple partitions.
- A single partition can only have one consumer in a consumer group.
- A consumer in a consumer group can read from more than one partition.
- Consumers in a consumer group can connect to a specific partition to read records. Alternatively, consumers in a consumer group can also allow Kafka to distribute partitions to consumers in a consumer group in a round-robin fashion.
- If there are more consumers in a consumer group than the number of partitions, the additional consumer(s) will have no partition to read from.

Hope this information helps. Drop me a line if you think I should add other information about Kafka here.