APACHE KAFKA

- Apache Kafka: - Apache Kafka is a distributed data store optimized for ingesting and processing stream data in real-time.
  Streaming data is data that is continuously generated by thousands of data sources, which typically send the data records simultaneously.

  Kafka Queue in Apache Kafka Queueing System, message is saved in queue fashion. This allows messages in the Queue to be ingested by one or more consumer, but one consumer can only consume each message at a time.

Advantages of Apache Kafka: - Apache Kafka's message broker system can sequentially and incrementally process a massive inflow of continuous data streams that are simultaneously produced by thousands of data sources with high throughput and durability. The data integration benefits are:

Scalability
By dividing a topic into multiple partitions, Apache Kafka provides load balancing over a pool of servers. This allows you to scale production clusters up or down to fit your needs and to spread clusters across geographic regions or availability zones.

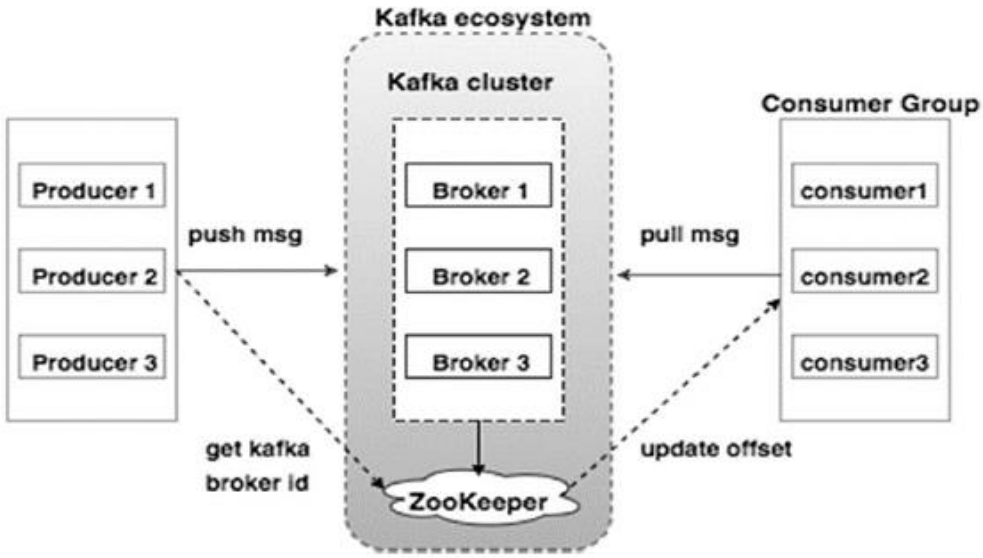
Speed
By decoupling data streams, Apache Kafka can deliver messages at network limited throughput using a cluster of servers with extremely low latency (as low as 2ms).

Durability
Apache Kafka makes the data highly fault-tolerant and durable in two main ways. First, it protects against server failure by distributing storage of data streams in a fault-tolerant cluster. Second, it provides intra-cluster replication because it persists the messages to disk.

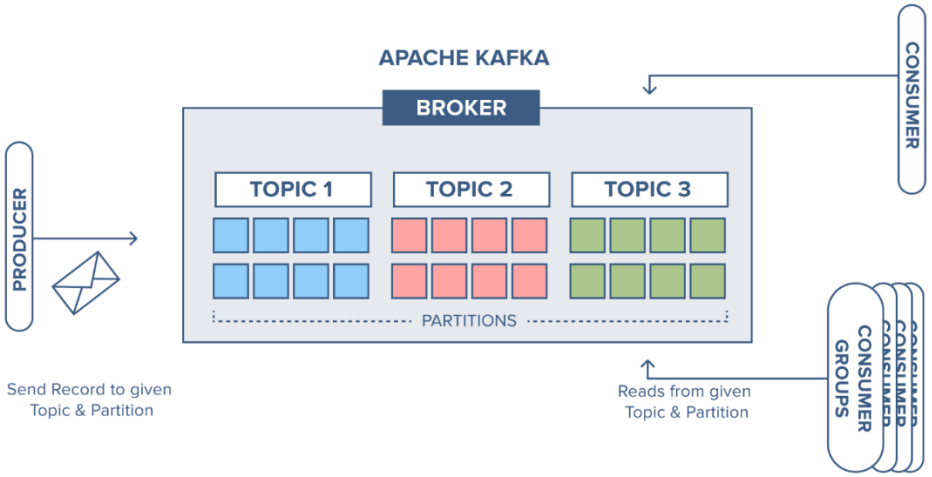
Before starting let us discuss some terms used in Kafka

Terms in Kafka:

- **Kafka Cluster:** Since Kafka is a distributed system, it act as a cluster. A Kafka cluster consists of a set of **brokers**. A cluster has a minimum of three brokers.



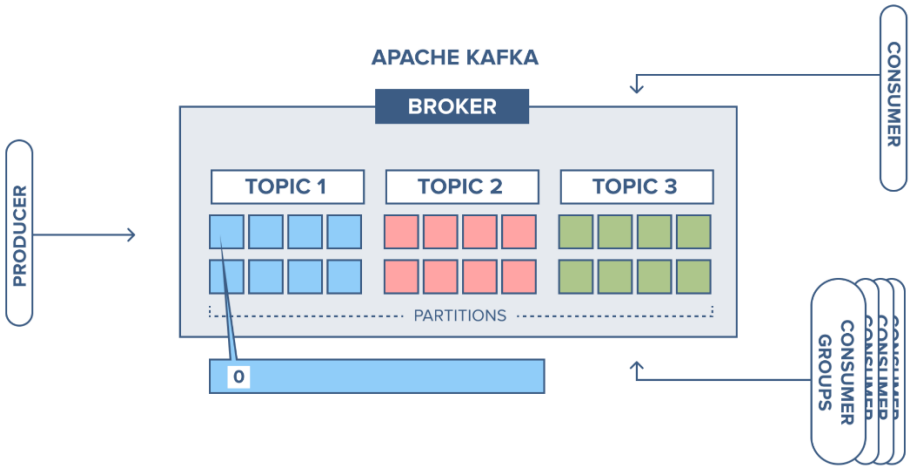
- **Kafka Broker:** The Broker is the Kafka Server. It is just a meaningful name given to the Kafka server. And this name make sense as well because all that Kafka does it act as a message broker between producer and consumer. The producer and consumer do not interact directly. They use Kafka server as an agent or broker to exchange messages.



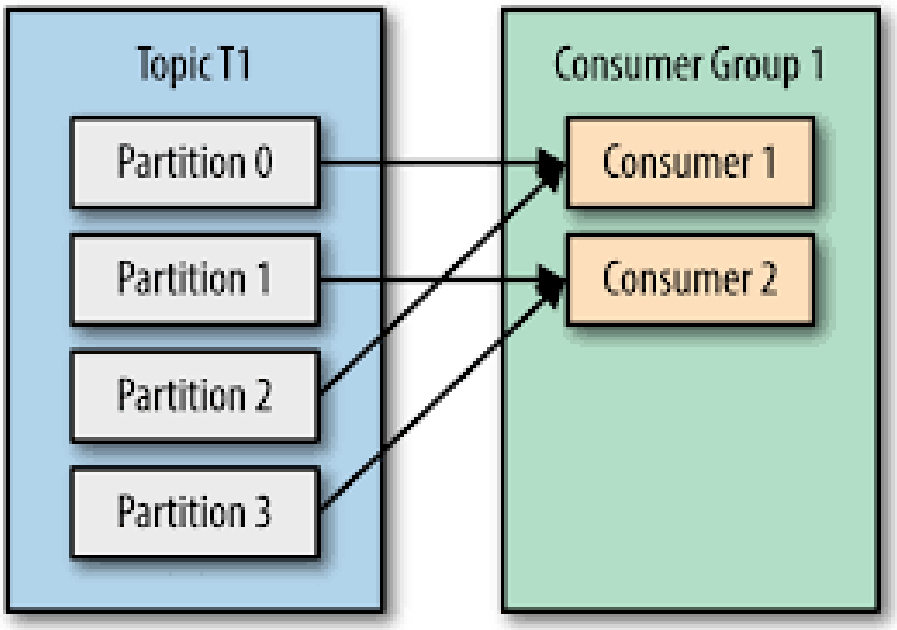
- **Producer:** Producer is an application that sends messages. It does not send messages directly to the recipient. It sends messages only to the Kafka Server.
- **Consumer:** Consumer is an application that reads messages from the Kafka server.
  If producers are sending data, they must be sending it to someone, right? The consumer are the recipients, but remember that the producer does not send data to a recipient address. They just send it to Kafka server, and anyone who is interested in that data can come forward and take it from Kafka server. So, any application that request data from Kafka server is a consumer, and they can ask for data send b by any producer provided they have permission to read it.

NOTE: we learned that producer sends data to the Kafka Broker. Then a consumer can ask for data from the Kafka broker. But the question is which data? We need have some identification mechanism to request data from a broker. There comes with the ideas of TOPIC.

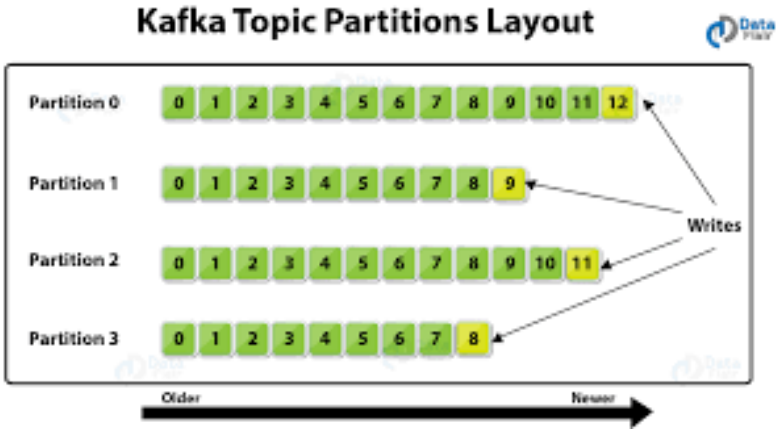
- **Kafka Topic:** It categorize the message or data according to its type. Ex- String, Integer, JSON etc. etc.
  -Topic is like a table in database or folder in a file system.
  -Topic is identified by the name.
  -You can have any number of topics.



- Kafka Partitions: Kafka Topics are divided into a number of partitions, which contain records in an Unchangeable Sequence. Kafka broker will store messages in a Topic, but the capacity of data can be enormous and it may not be possible to store in a single computer. Therefore, it will be partitioned into multiple parts and distributed among multiple computers, since Kafka is a distributed System.



- Offsets: Offsets is a sequence of IDs given to messages as they arrive at a partition. Once the offset is assigned it will never be changed. The first message will get the offset number zero, the next message will receive an offset one and so on.



- Consumer Group: A consumer group contain one or more consumer working together to process the message.