★ Kafka

- Open source stream-processing software platform which is used to handle the real-time data storage.
- Works as a broker b/w two parties i-e sender & receive.
- Publish - subscribe messaging system which lets exchanging data b/w app^n, servers & processors as well.

• Messaging system

- Simple exchange of msg. b/w two or more persons, devices etc.
- Publish - subscribe mess. sys. allows sender to send/ write the message & a receiver to read that msg.
- sender is known as producer
  Receiver is " " consumer.

• Streaming Process

- Processing of data in parallely connected systems.
- Process allows diff. app. to limit ||^{el} execution of data, where one record executes w/o waiting for o/p of previous record.

- It stores streams of records in a fault-tolerant durable way.

• Apache Kafka Core API's

1. Producer API
- Allows / permits an app to publish streams of records to one or more topics.

2. Consumer API
- Allows an app^n to subscribe one or more topics & process the stream of records produced by them.

3. Streams API
- Allows an app^n to effectively transform i/p streams to o/p streams.
- It permits an app^n to act as a stream processor which consumes an i/p stream from one or more topics, & produce o/p stream to one or more o/p topics.

4. Connector API
- Executes the reusable producer & consumer APIs with existing data sys. & app^n.

• Topics

→ Refers to a particular heading or name given to some specific inter-related ideas.
- Refers to a category or a common name used to store & publish a particular stream of data.

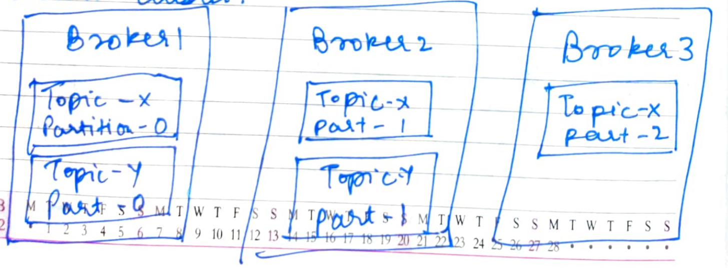
→ Topics are similar to tables in DBS.
→ A ~~subscriber~~ producer publishes data to the topic & a consumer reads that data from the topic by subscribing it.

- Partitions

- topic is split into several parts which are known as partitions. And are seperated in an order.
- while creating topic we need to specify the no. of partitions.
- Each msg. gets stored into partitions with an incremental id known as its offset value.
- Offsets of a partition are infinite.

- Broker

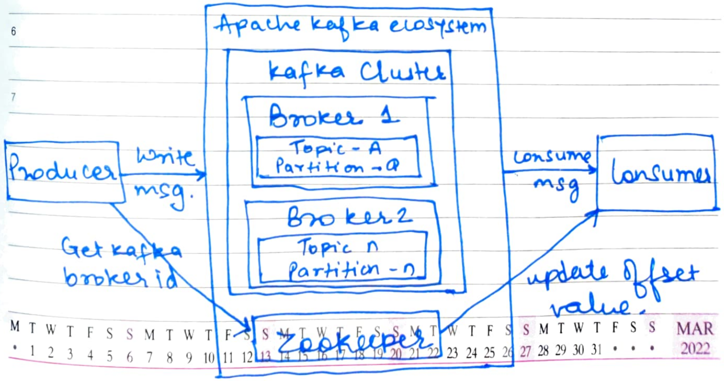
- kafka cluster comprises of one or more servers known as brokers.
- It is a container that holds several topics with their multiple partitions.
- Identified with an integer id only
- Also known as Bootstrap broker as connection with any one broker means connection with entire cluster.



- Replication factor value should be always greater than 1.
  - stores copies of Topics & partitions in other brokers
  - To choose topic for client request it chooses one of the broker partition as leader & others as followers.
  - If leader fails then one of the follower takes the lead.

- Zookeeper

- store info abt kafka clusters & details of consumer clients.
- manages brokers by maintaining a list of them.
- Responsible for choosing leader.
- If any thing fails it sends msg. to kafka.
- operates with an odd no. of kafka servers
- users interact with zookeeper via brokers


Apache kafka ecosystem
kafka Cluster
Broker 1
Topic - A
Partition - 0
Broker 2
Topic n
Partition - n
Producer
write msg.
Get kafka broker id
Zookeeper
consume msg
Consumer
update offset value.