Week 6

streaming _ kafka
(Messaging System)
[. Trasportation System]

. Integration between producer & consumer

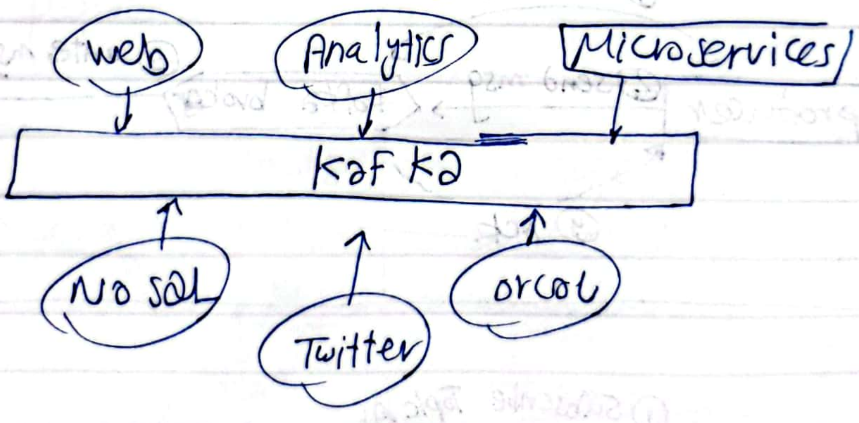
. kafka Handel :
- protocol - How data transported (TCP, HTTP, REST..)
- Data format (Binary, CSV, JSON)
- Data schema How data is shaped and might change.

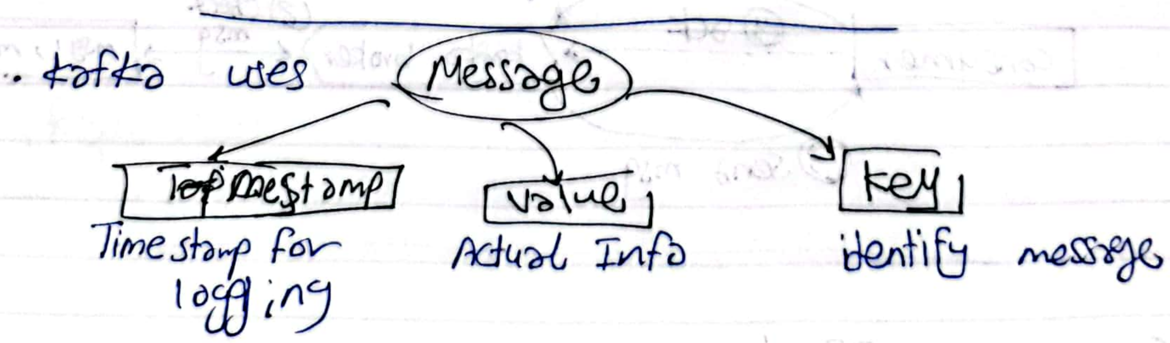
. latency less 10ms [Real Time]

- consumers : Those that consume data
  web pages , micro services
- produces : who supply data to consumes

- connecting consumes direct to producers can lead to amorphous and hard to maintain architecture.

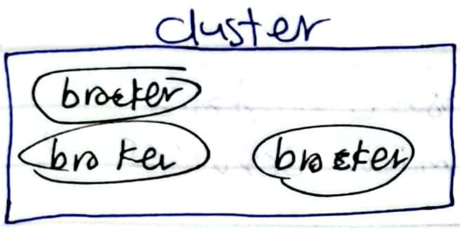
. kafka is intermediary component





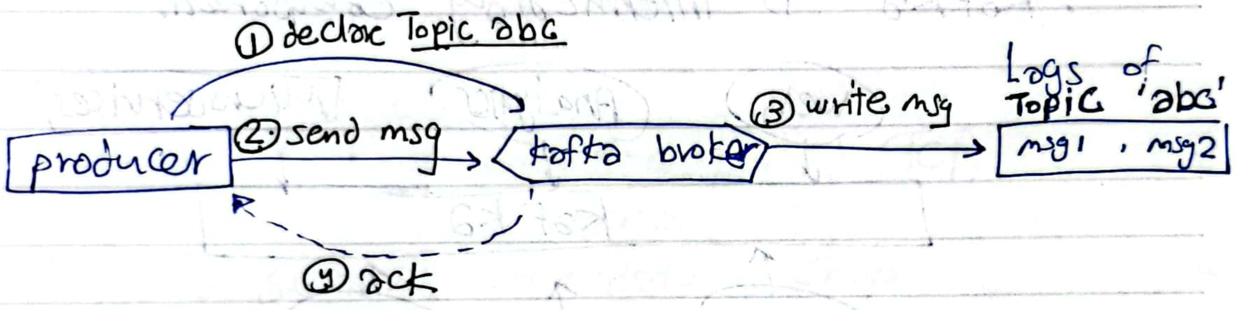
Topic

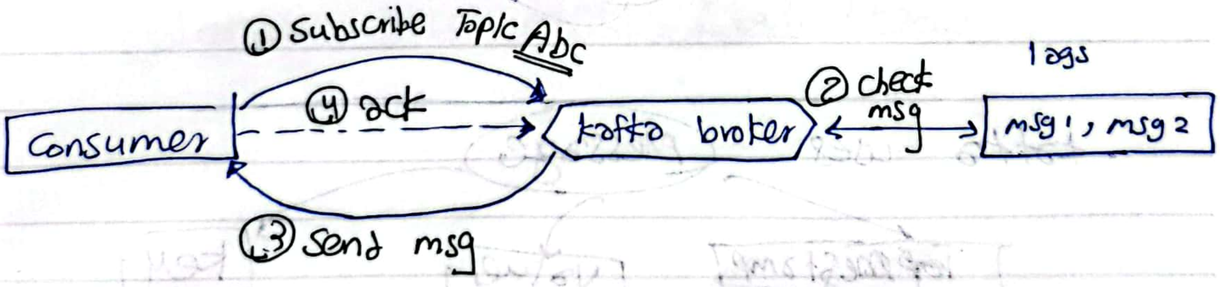
Topic which set a group of msgs.



<u>Broker</u> : pyshical or virtual Machine where kafka work.
سمسار

<u>logs</u>
kafka store and assige each msg ID then store it in logs





<u>Consumer_offset</u>
special topic keep track of each consumer and topic
keep track of what consumer do.

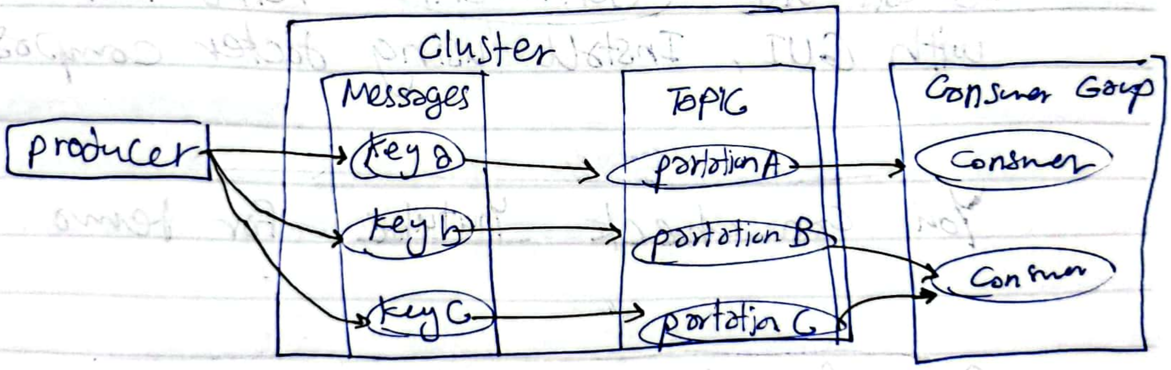
<u>e.x</u> If consumers dies or what ever happen and get back consumer_offset will continue from past state.

Consumer Group

- a set of consumers with an ID for Group & ID for each Consumer But that help with:
  - Duplicated
  - redundant msgs.

Partitions

Logs contains msgs ID, we call Topic logs as partitions.



Replication

partitions are replicated across multible brokers in kafka. fault tolerance.

Kafka Configurations

retention (ms): Time specific topic will be available before deleting.

replication : Replication factor, Number of times partitions will be replicated.

consumer configuration
offset: msgs ID which have been read
by consumers.

producer configuration
acks :
0 → fire forget
1 → wait for leader

Installing kafka:

we will use confluent kafka tool
with GUI, Install using docker compose.

You can check Youtube for demo.

Avro & schema Registry:

---

msgs can be any thing plain txt - - - binary
this give Flexibility to kafka.

But we must take care of compatibility

$$\begin{tikzcd}
\text{PHP producer} \arrow[r, "\text{PHP}", "\text{object}"'] & \text{kafka} \arrow[r, "\text{WTF}", "??"'] & \text{CSV consumer}
\end{tikzcd}$$

Avro

Data serialization system, unlike JSON Avro stores schema separated from record.

- smaller record size.
- Schema evolution : Change schema over time.
- Automatic validation

schema registry

contains info about schema, we validate from schema reg.

Schema Evolution

for ex Backward check against last version, upgrade consumers first.

forward upgrade producers first.

Kafka streams

- Kafka streams used when we work with input & output stored in Kafka clusters. stream & produce data ~~From Kafka to Kafka~~
- streams process events with latency of milliseconds, deal with msgs ASA they arrive

. streams works with DSL
(Domain specific Language), which simplify creation of stream.

. Kafka stream is powerful and simple
Spark and flink more powerful, But hard.

> Kafka streams work with Kafka produce and consumer only.

~~state~~ . Stream vs state
- ~~stream changelog~~ - ~~individual msg~~
- ~~KTable~~ - ~~read sequential~~

- stream (K streams)
  - individual msg read sequentially.

- state (K-state)
  - stream changelog
  - view of stream at certin time.

Topology
How input data transfoned to output.

stream processor
Transform data.

• Native language for kafka stream is Scala. in $\boxed{\text{JVM}}$
we will use faust python library

Windowing
Time reference in which series of events happen.

• Time based window
- window has predetermined size "S"

• Session - based
- window based on key to start and end not Absolute time.

• Kafka support Threads, And control number of threads.

• Global K-Table
- Brodcast variable for
• more convenient
• efective Joints
• But increase local storage

• Interactive queries
Allows External apps to query your Stream apps directly.

- kafka connect
  - stream data between external application

$$\begin{tikzcd}\text{Sources} \arrow[r] & \boxed{\text{connect}} \arrow[r] & \text{sinks}\end{tikzcd}$$

KSQL

KSQL offers consumers such as Data science a tool for analyzing kafka streams.