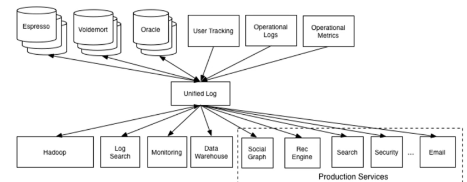
# **Apache Kafka**

Kafka was originated at linked in, to solve the data integration problem

Kafka was developed by jay creeps

Kafka can be used as streaming platform



## What is apache Kafka?

Apache Kafka is highly scalable (each broker can be replicated easily like replication factor of 3.. and once ur connected to any broker in a cluster means you are connected to the entire cluster) and it’s a distributed platform for creating and processing streams

Its Iike a Blood circulatory systems which will pump blood to all organs similarly it will push data to all listeners. Messages can be sent from producers to consumers in milli seconds

Example of processing streams

1. To know the lorries real time positions the machines kept in lorries will continuously send their latitude and longitude data to the kafka broker, this is called stream of data because these lorries will send continuous data/ data stream to the broker

Stream is nothing but continuous flow of data

How it evolved as streaming platform from data integration platform

Components of Kafka

|  |
| --- |
| 1. Kafka Broker - |
| 1. Kafka Client |
| 1. Kafka connect – for integration between 2 components / multiple components solving above problem- like 1dstr appn is used for integration as we don’t get stream of data()because its not continuous flow of data |
| 1. Kafka Streams |
| 1. Kafka SQL/ Ksql – with this kafka wanted to become real time database. – all above 4 are open sourced whereas this is licensed |
|  |

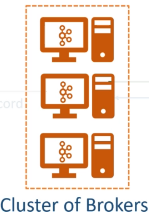
Ideas

If u don’t have access to source code use kafka connect api(source connector for producer and sink connector for consumer api)

Broker

Broker is nothing but a kafka server

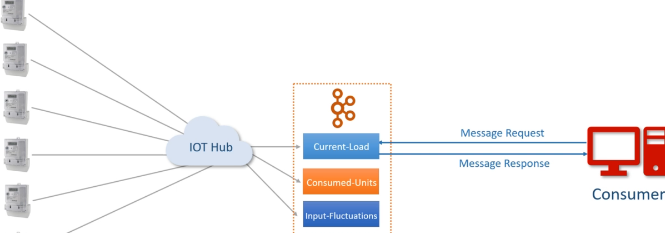
A kafka cluster is nothing but collection of brokers



TOPIC

It is the unique name given for the data stream, its like a small data base table

Like all sakshi related news in sakshi related topic , and etv news in etv topic



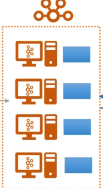
Partitions

Topic is divided into many partitions and partition is divided into many segment

Think if every current meter is sending data (called stream (flow of data)) for every minute to a topic it can’t store in a single computer the topic would have been divided into many partitions **, each partition would get stored in separate computer** , but each partition will further be divided into multiple segments, based upon the load we should decide how many partitions we want, in case of huge data prefer more partitions

Huge load== huge partitions(20,000 partitions)== we may need huge consumers(20,000 consumers in same consumer group, therefore each consumer reads from 1 single group)

So its our responsibility to decide the number of partitions we want



Partition offsets

Offset number /sequence number=in cinema theater it’s our seat number=message arrival order number

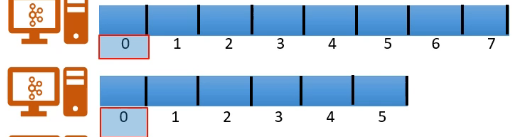
It same like in cinema theater how we get seat number , when we came to cinema theater the theater management will give us that number so that we can sit in that seat

It is the unique sequence id that is given to each arrived message in to the partitions topic by the kafka broker

Its clearly an arrival order number, when a message came broker will give

But these offset numbers are local to that partition, like another partition can also have same partition number , within each partition offset is going to start by 0 and increase by 1.

These provided numbers are immutable



Therefore if u want to locate a message then u have to know the topic number, partition number (this partition number is mandatory because 1 topic may have many partitions),offset number

What if an big message came ?

Will it be broken and stored across many partitions and each part will have some offset number??

Consumer group

If multiple consumers want to share the work, then they will form as a group and share the load, its same like podupu group