Kafka

To get data from so many prodeucers and to publish data to so many consumers, we use kafka. It is a streaming service.

Kafka is used to handle live streaming data pipeline.

It is used mainly to:

Publish streams of events, continuous import/export of data.

Store stream events.

Process stream of events to occur.

Why we are using kafka, instead we can use mongoDB ?

To receive live streaming data we are using kafka because directly receiving data to mongoDB is not possible. To receive streaming data in database we need to make connection to the database, which will take time again and again.

In [ ]:

```
1
```

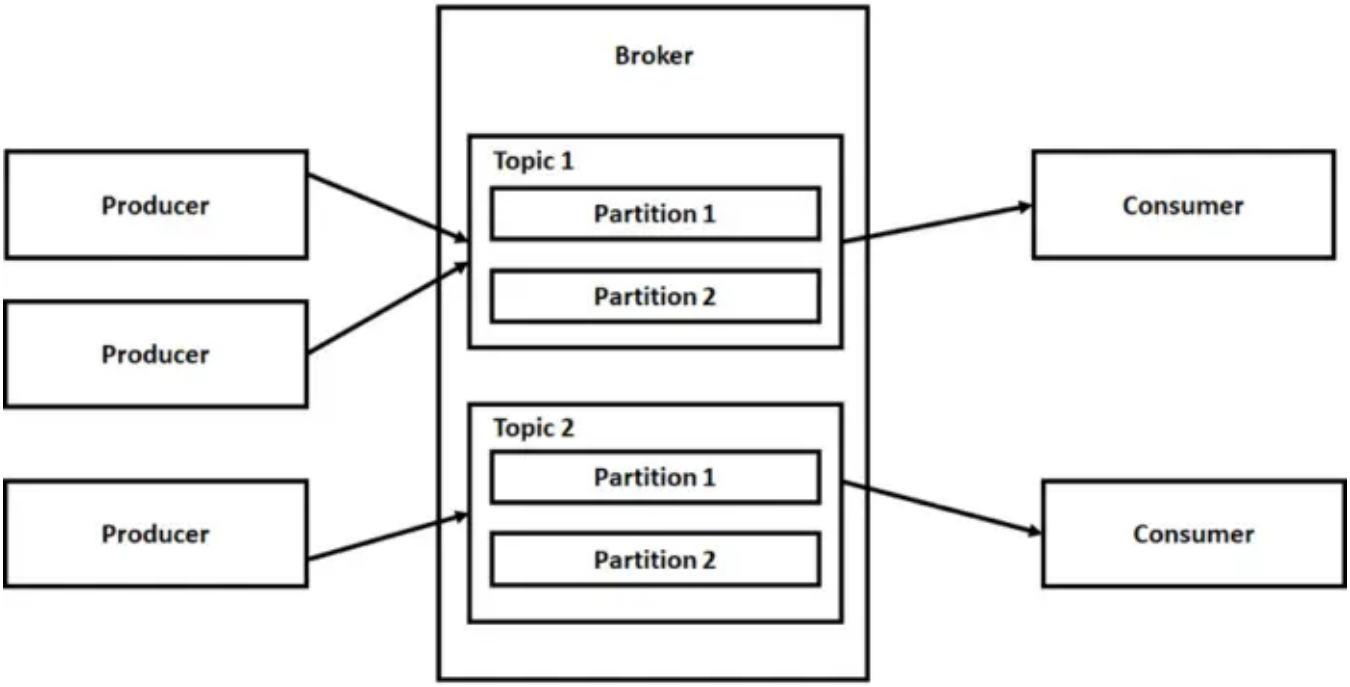
Can we store data in SQL based database and use it for analysis ?

No, because SQL or tabular structured databases run little slow. They are used in tranctional operations. And from nosql databases like mongodb we get data quickly.

No-SQL databases: mongoDB, Cassandra, S3 bucket.

In [ ]:

```
1
```



For pre-processing the data producer write a script and send the produced data in required format to streaming platform(Kafka). Kafka stores the streaming data for some amount of time. Consumer will fetch the data from kafka database by writing a script and stores the data to a database.

By-default kafka stores data for **1 week**. And we can extend the retaintion time. We need to consume the data within that period of time, otherwise kafka automatically deletes the data.

From that database **data Science** or **data analytics** team will collect the data and do some analysis.

Using **kafka** we can setup the **ETL pipeline**. For the above steps from collecting data to storing it to a database in required manner, **big data** team or **data engineering** team is responcible.

In [ ]:

```
1
```

Both producer and consumer interact with kafka topics to produce and consume data. Because we need to define the the topic name before above operations, and we need topic name to get the structure of the data.

Topics : Kafka topics are virtual groups or logs that hold messages and events in a logical order, allowing users to send and receive data between kafka servers with ease.

Topics are dedicated to every single operation and kafka can have multiple topics with-in a cluster. Each Topic has some schema to receive data with device id.

There are **partitions** in kafka topics to receive data faster in realtime. While receiving and storing data inside kafka-topics there is **no order across the partitions but inside the partitions there is**, so we use timestamp to save data.

Partitions are mainly used to make work kafka in distributed environment for **big data** with **large number of systems**. Because when we will send **huge amount of data to kafka server**, it will create **huge load**. And the partition inside kafka will load data **quickly**.

A group of machine working together is **cluster**.

While receiving data from Kafka we need to define the **group name**, so that kafka keeps the **track** of consumer that upto how much it has sent the data. So from next time onwards it'll resume from there and data won't get **over ridden**.

Because of this property of kafka which keeps the track of read and unread status of consumer(**partition number and row number**), users won't get duplicate data.

In [ ]:

```
1
```

In kafka we need to define the **schema** to get particular data in a proper format.

Every topic has unique schema as per data provided. Hence one opic is dedicated to an individual operation.

In [ ]:

```
1
```

To store data to kafka we need to convert the records to **BSON format** to use json producer. Then we will serialize the json data using json serializer.

Serialize : Converting data to a certain format so that it can be stored in a disk. (eg: Pickle is a kind of serialization where we save our model to a file)

In [ ]:

```
1
```

The reverse operations we follow while consuming the data from kafka. Where we are **deseializing** to get **BSON data**. And then convert it to **file object** to store in database.

End point schema url is used to maintain different different schemas.

In [ ]:

```
1
```

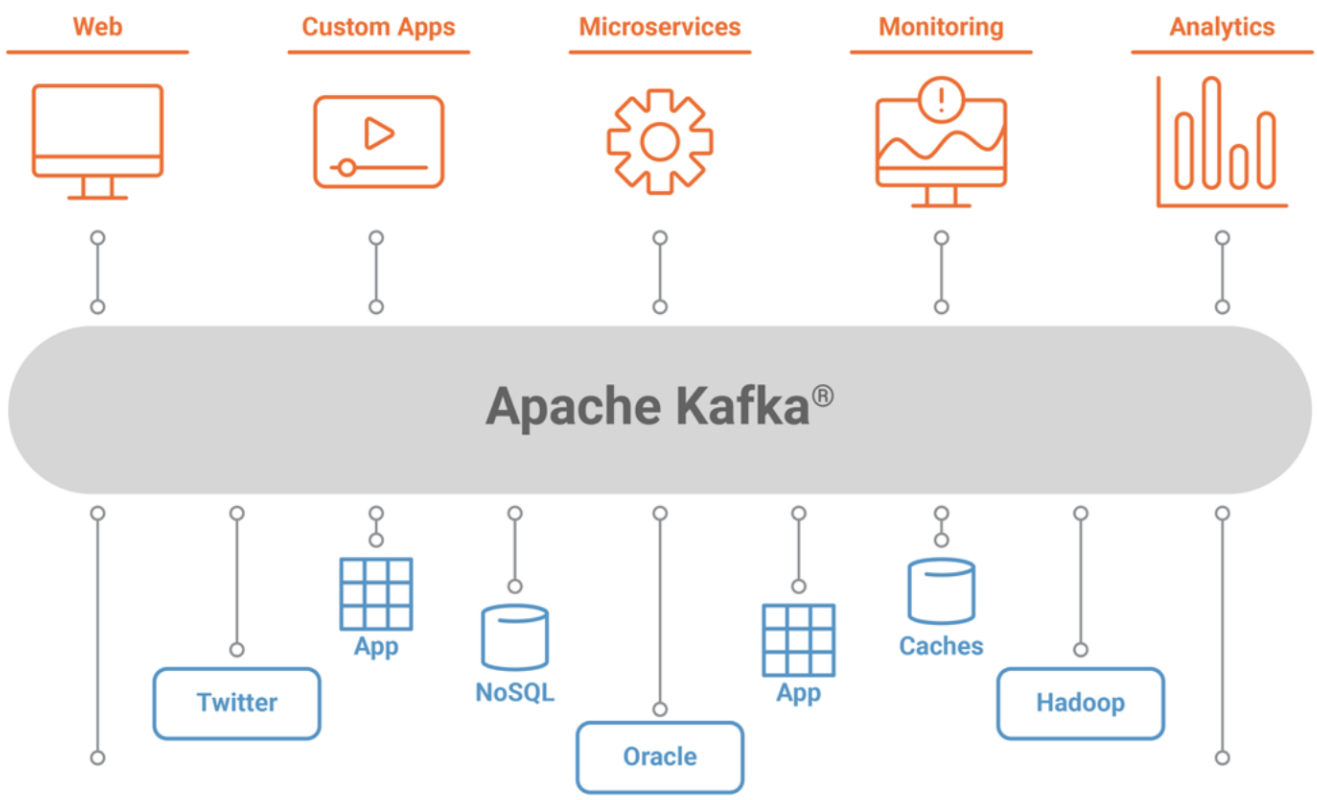
We are using **Confluent** virtual machine to use streaming service of kafka, which is a vendor to manage the operations of kafka with high performance. Confluent is a SAAS, which provide kafka as a service.

Steps in confluent:

1. Create a cluster. (API ket, API Secret, Bootstrap server)
2. Create topics
3. Create Schema API key, API Secret and take Schema End point url -->> To maintain the structure of the data centrally

Kafka can be used directly to create pipeline

Source(devices) >>> Kafka >>> mongoDB



github: https://github.com/saisubhasish (https://github.com/saisubhasish)

In [ ]:

```
1
```