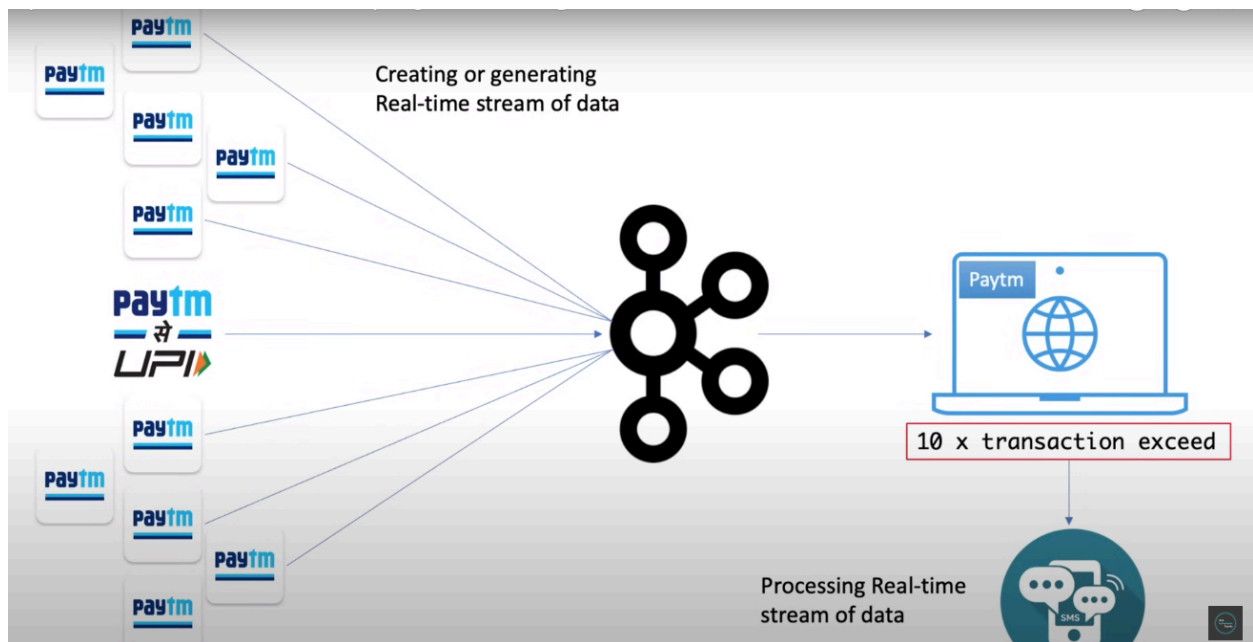**What is Apache Kafka?**
 Apache Kafka is an open-source distributed event streaming platform.

1. Creating Real-time Stream
2. Processing Real-time Stream

==Creating Real-time Stream==

Suppose we are purchasing a fligh ticket by Paytm. Because Paytm provides multiple types of transactions. Like flight ticket booking. Movie ticket booking etc.When we will do any transaction then event will go to the kafka server but here I am not the only one Paytm user who is using the transaction there are millions of user and they are performing millions of transactions into milliseconds. **Kafa server receives millions or billions of event in each minutes or each seconds or even in each milliseconds. Sending the stream of continuos data from Paytm to Kafka. It's called Creating or generating Real-time stream of data.**

==Processing Real-time Stream==

When Kafa server received the data then need to process the data for the operations.
Like: Suppose Paytm want to restrict mark 10x transaction to users per day.if they exceed the limit then paytm wants to send a notification to user. In such this scenarios paytm needs to read the data from the kafka server continuously for validating to the user transaction limitation.
This type of data processing is called **Processing Real-time stream of data.**

What is the meaning of distributed?
To the distributing of kafka server is called the distribution.
Suppose there are three kafka server running the different origin to perform event operations.
In case if any server is goes down another server will come and pick-up the traffic to avoid the application downtime.

Where does kafka come from?
Kafka was originally developed at LinkedIn, and now it is open source sine 2011.
Now it is under of **Apache Kafka Foundation.**

Why do need the Kafka?

Real life Example.
Suppose I hava some parcel on my name and delivery man comes to my home for delivery but unfortunalttly I was not at my home and delivery man return from my home and They repeated same attempt three times and every time I was not at my home and finally delivery man return my parcel to main office. But I donot know who came to my home for parcel delivering may be in the parcel there are some important data. Like papers or money related paper but here I did lost the data because I was not available during the period when the postman came to my door. This could be huge lost for me.so how we can overcome this issue.

If I have installed latter box at my home so when delivery man will come and in my absence they put items into letter box. Wherever I will be back then I can collect the my order. In this case I will not lose the data. Here **Letter Box** acts as middle layer between me and delivery man.

Real time Example.
Suppose there are two applications and application1 wants to send the some data to application2 but application2 was not available to take the data then application2 is lost the data and which might be impact business logic or furthure processing. So overcome this issue same Letter box we have to build a middle layer. In Technical terms we can use messaging system as Middle layer between two applications.
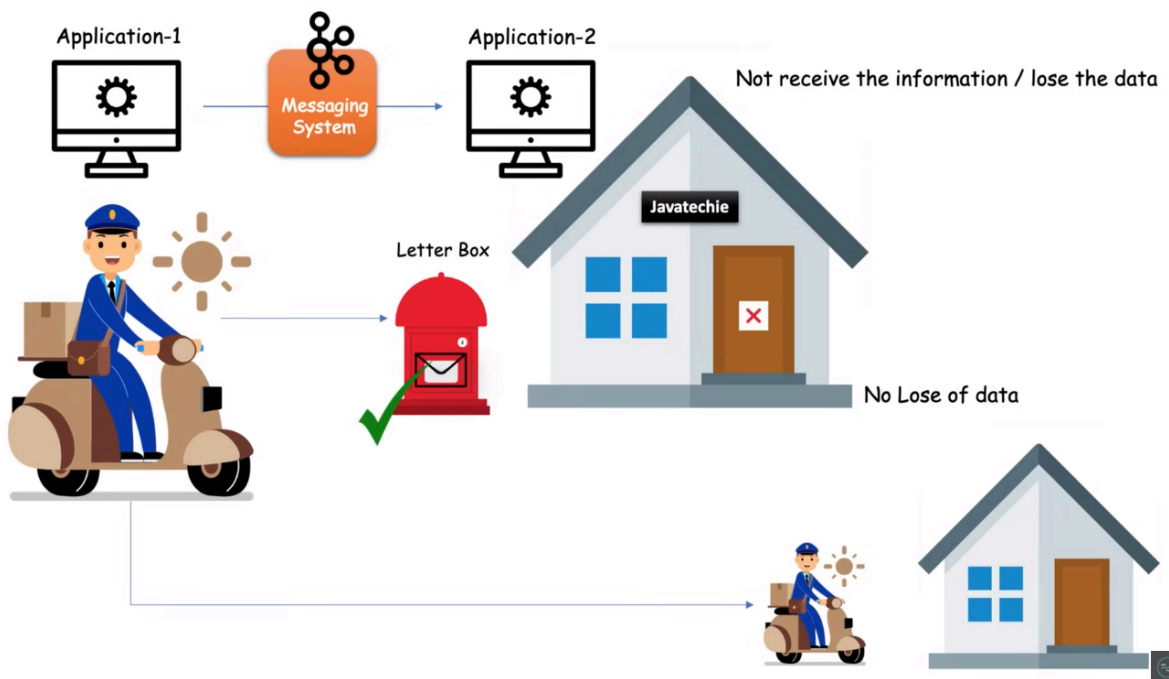
How does kafka internally work?

It is specially works on Pub/Sub Model. (Publisher and subscriber model)

How does it work (High-level overview)

Pub/Sub Model

Message /event

Message Broker

Publisher

Subscriber
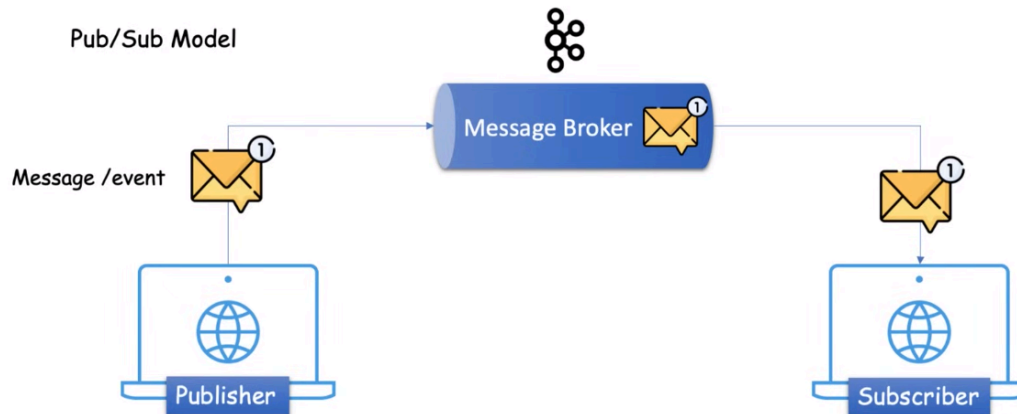
Kafka Architecture and Components.

Below core concept of Kafka.
1. Producer
2. Consumer
3. Broker
4. Cluster
5. Topic
6. Partitions
7. Offset
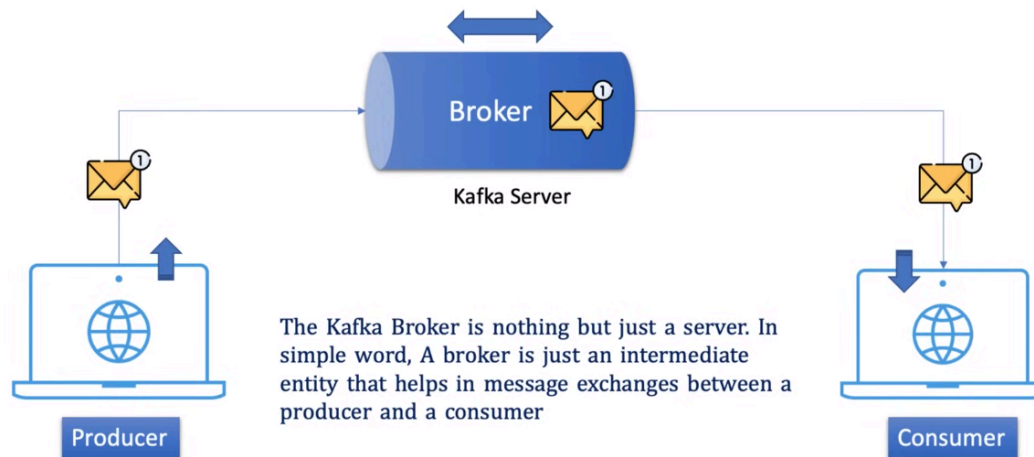8. Consumer Groups
9. Zookeeper

Producer.
It is just responsible to produce the data to kafka server of Broker.

Consumer.
It is just responsible to consume the data from kafka Broker.

Kafka Broker/Server.(Default port = 9092)
It is just a server and works as intermediate between Producer and consumer to exchange the data

The Kafka Broker is nothing but just a server. In simple word, A broker is just an intermediate entity that helps in message exchanges between a producer and a consumer

Cluster.
Cluster is nothing.It is a common terminology in the distributed computing system. It is nothing just a group of computer or server that are working for a common purpose. A Kafka cluster consists of a set of brokers. A cluster has a minimum of 3 brokers.
Suppose your producer publishing huge volume of the data so a single Kafka Broker may not handle the load then he might to need additional kafka server or broker. Then Kafka cluster adding group of brokers.

Topic.
When the 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 to have some identification mechanism to request data from a broker. There comes the Kafka topic.

- The topic is like a table in a database or folder in a file system.
- The topic is identified by a name.
- You can have any number of topics.
- Topic acts like database tables.

The following diagram shows two Topics are created in a Kafka broker:
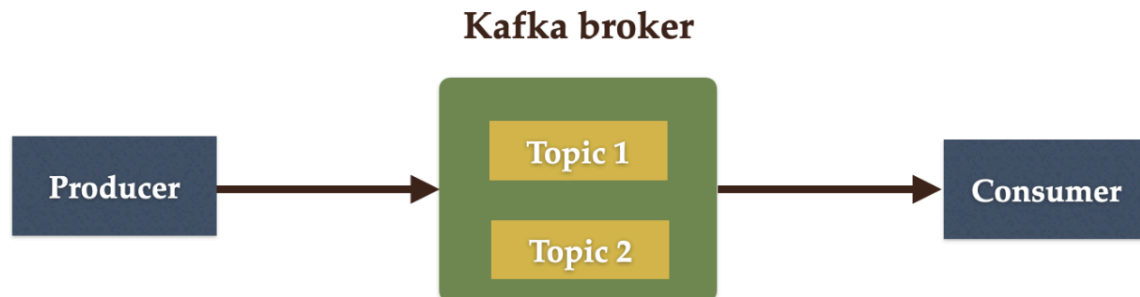
Partition.
Partitioning in Kafka is the process of dividing a topic into multiple partitions, where each partition is an ordered, immutable sequence of records. Partitioning serves several purposes:

1. **Scalability**: Multiple consumers can read from different partitions in parallel, allowing Kafka to handle a high throughput of data.

2. **Distributed Processing**: Partitioning enables distributed processing by distributing the data across multiple brokers.

Offset.

The data which is stored into Partition and a position or id is assigned that id or number or indexes is called offset.
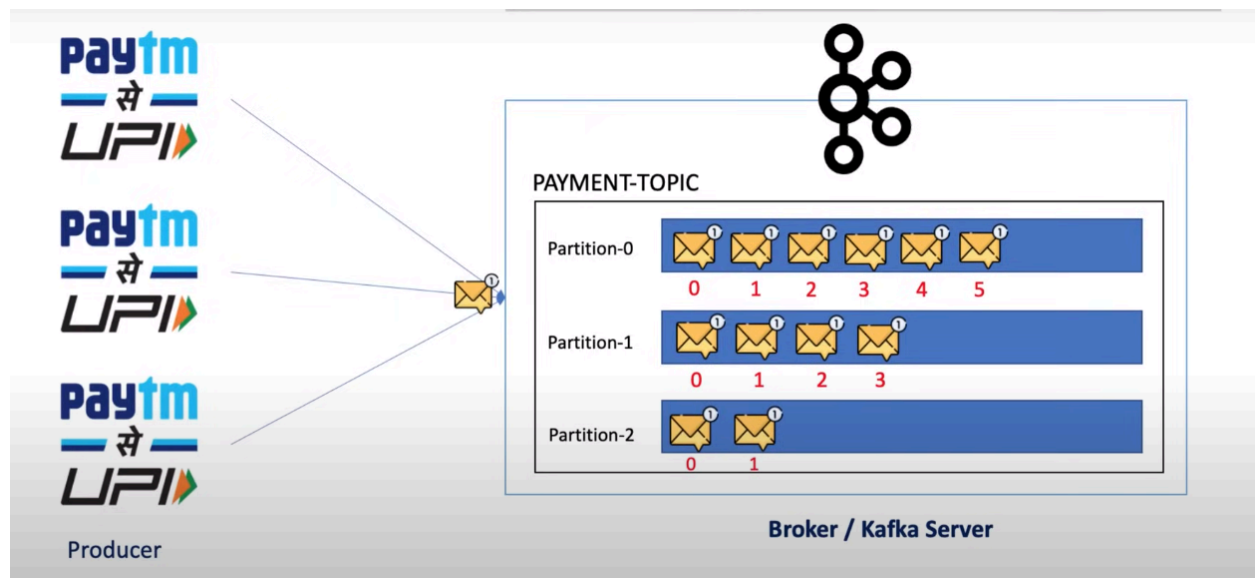
1. Offset is used for tracking like which message is consumed by consumer.
2. Suppose we have 5 offset in kafka partition and consumer consume all 3 offset data but after that consumer is down. Then consumer will back to the online so here offset value will help to the consumer exactly where consumer has to start consuming the messages.
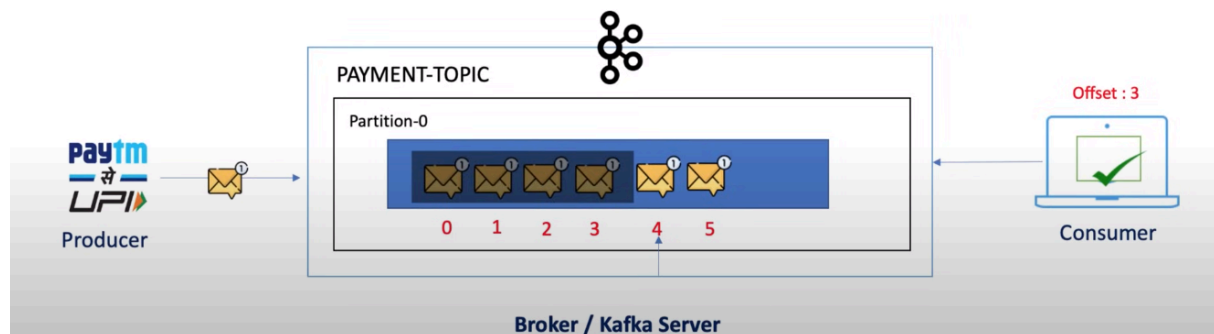
Consumer Group.

As below image suppose we have three partitions if we have one consumer so here performance is not fast because one consumer will read the data from all partitions but if we make a consumer group and make three consumer this time performance is fast because randomly each consumer consume the data from each partition and we do not have control that consumer can read the data from partition 1 because it is decided by coordinator.

In Below image we have 4 consumer so what is the behaviour of this consumer group 4?

The consumer 4 will sit on bench and there is no work for him but if any consumer instance is rejected or goes off then will get chance of the consumer 4 to connect any that is called the consumer rebalancing.
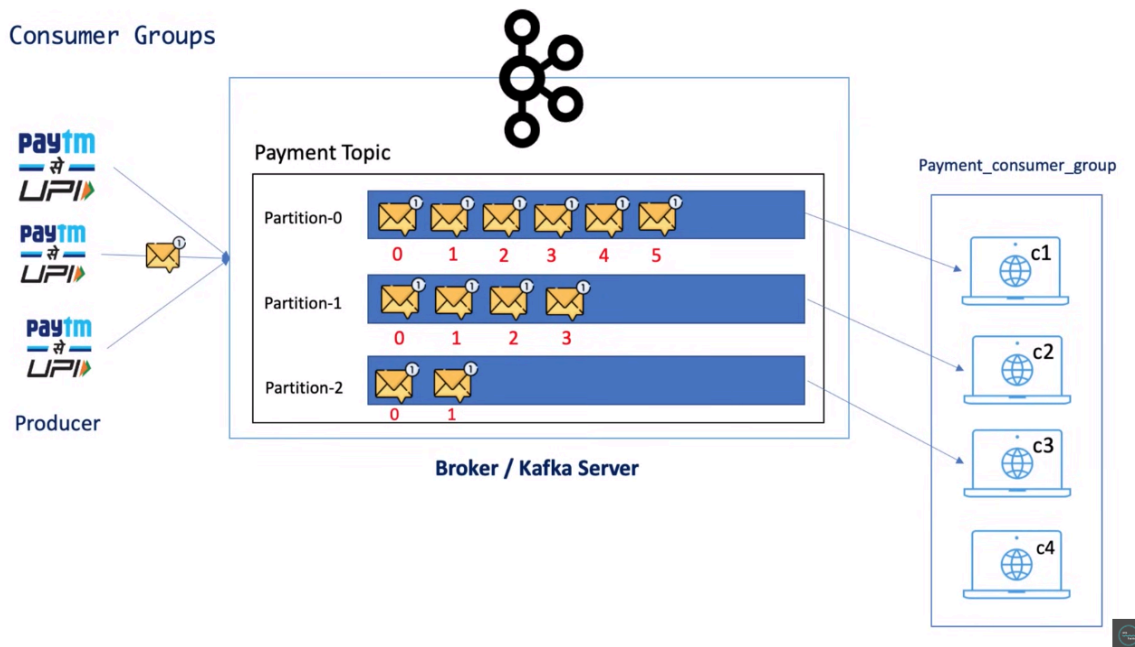
Zookeeper (Default port = 2181)
Zookeeper acts as manager for tracking the status of Kafka cluster nodes. It is also track of the kafka topics, partitions, offset, etc.
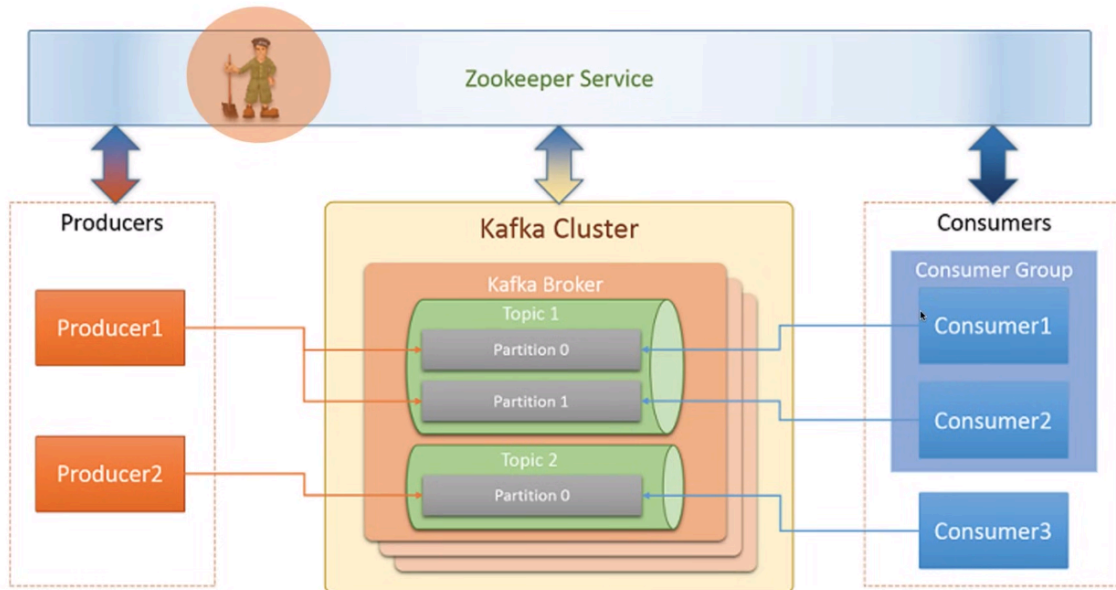
Open Source : Apache Kafka

Commercial distribution : Confluent Kafka

Managed Kafka service : confluent & AWS

1. Install and Setup Apache Kafka

1. Download Kafka from the official website at **https://kafka.apache.org/downloads**
->get satart -> quickstart -> download

2. Extract Kafka zip in the local file system by 7-zip application
3. Start Zookeeper service.
Note have to install it inside the extract Kafka file
Got to the below directive and hit enter

Start Zookeeper `= zookeeper-server-start.bat`
`..\..\config\zookeeper.properties`

```
D:\Application\kafka\bin\windows>zookeeper-server-start.bat ..\..\config\zookeeper.properties
[2024-07-13 14:19:13,038] INFO Reading configuration from: ..\..\config\zookeeper.properties (org.apa
[2024-07-13 14:19:13,042] WARN ..\..\config\zookeeper.properties is relative. Prepend .\ to indicate
```

4. Start Kafka Server/Broker
Start Kafka Server `= kafka-server-start.bat`
`..\..\config\server.properties`

```
D:\Application\kafka\bin\windows>kafka-server-start.bat ..\..\config\server.properties
[2024-07-13 14:24:50,337] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2024-07-13 14:24:50,743] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.apache.zoo
9Util)
[2024-07-13 14:24:50,898] INFO starting (kafka.server.KafkaServer)
[2024-07-13 14:24:50,898] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
[2024-07-13 14:24:50,920] INFO [ZooKeeperClient Kafka server] Initializing a new session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2024-07-13 14:24:50,935] INFO Client environment:zookeeper.version=3.6.3--6401e4ad2087061bc6b9f80dec2d69f2e3c8660a, built on 04/08/2021 16:35 GMT (org
```

5. How to create a Topic

```
D:\Application\kafka\bin\windows>kafka-topics.bat --create --topic my-topic --bootstrap-server localhost:9092 --replication-factor 1 --partitions 3
Created topic my-topic.

D:\Application\kafka\bin\windows>
```

Creating Kafka Topic `= kafka-topics.bat --create --topic my-topic`
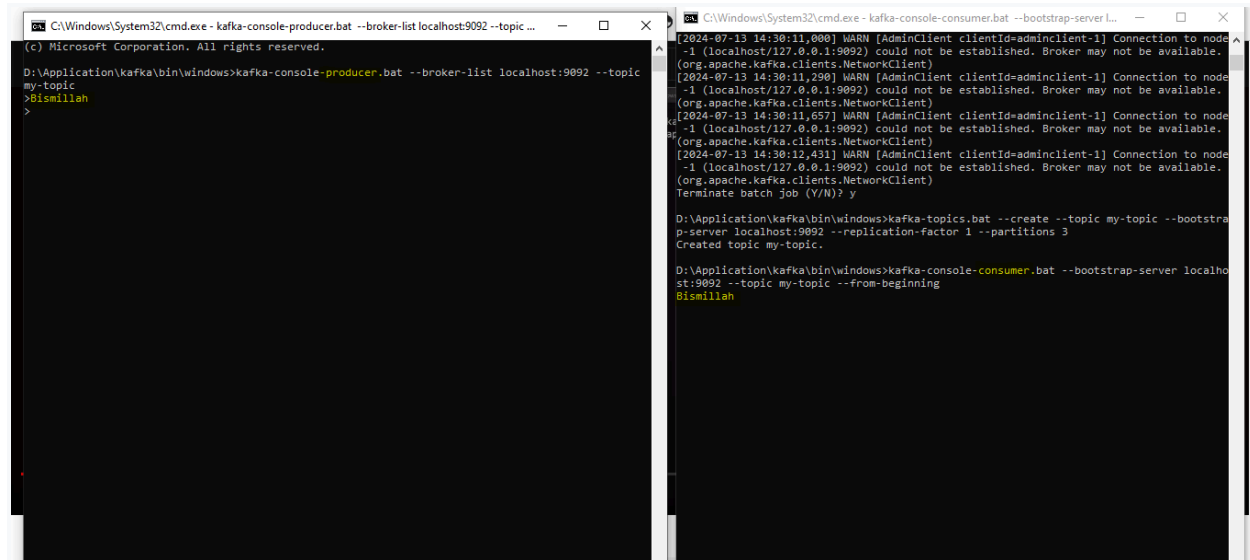`--bootstrap-server localhost:9092 --replication-factor 1`
`--partitions 3`

6. How to write some event or produce the topic

Creating Kafka Producer = `kafka-console-producer.bat --broker-list localhost:9092 --topic my-topic`

```
D:\Application\kafka\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic my-topic
>
```

7. How to read the event or producer data

Creating kafka consumer = `kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic my-topic --from-beginning`


```
D:\Application\kafka\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic my-topic --from-beginning
```

Installing confluent Kafka.

1. Open confluent.io
2. Fill the form and submit but be care full do not choose **cloud**

Installing Kafka Offset Explorer.

This is use for monitoring kafka messaging system.

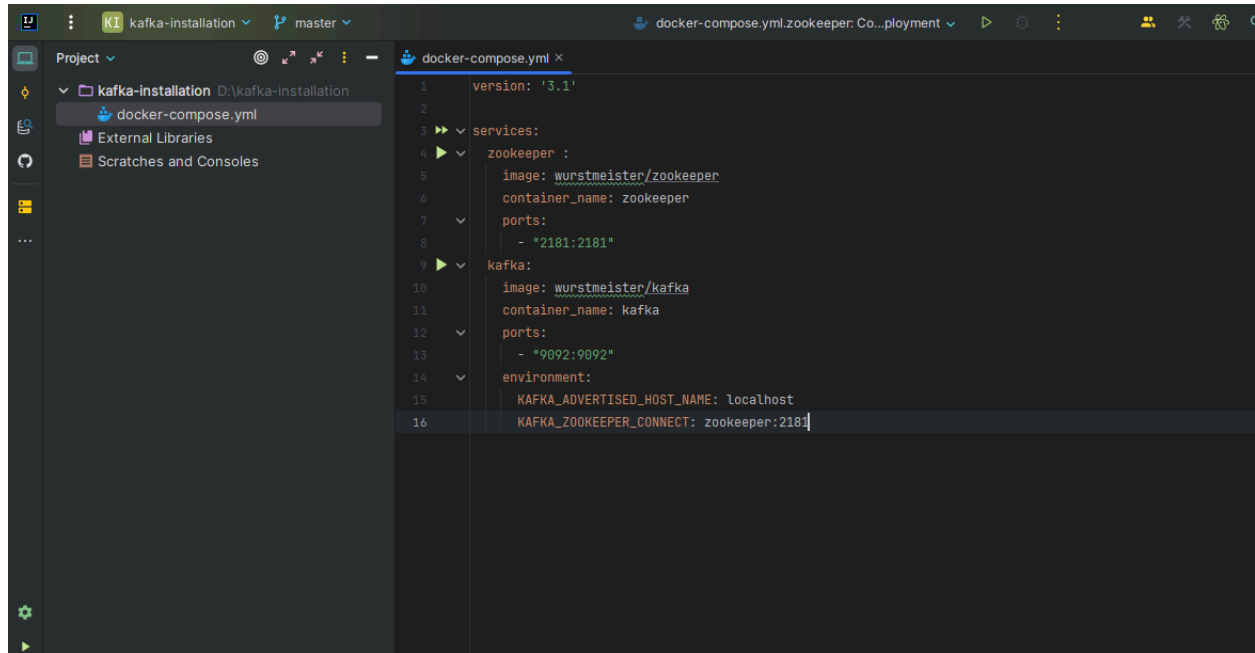Below image how to add new connection in Kafka Offset explorer.
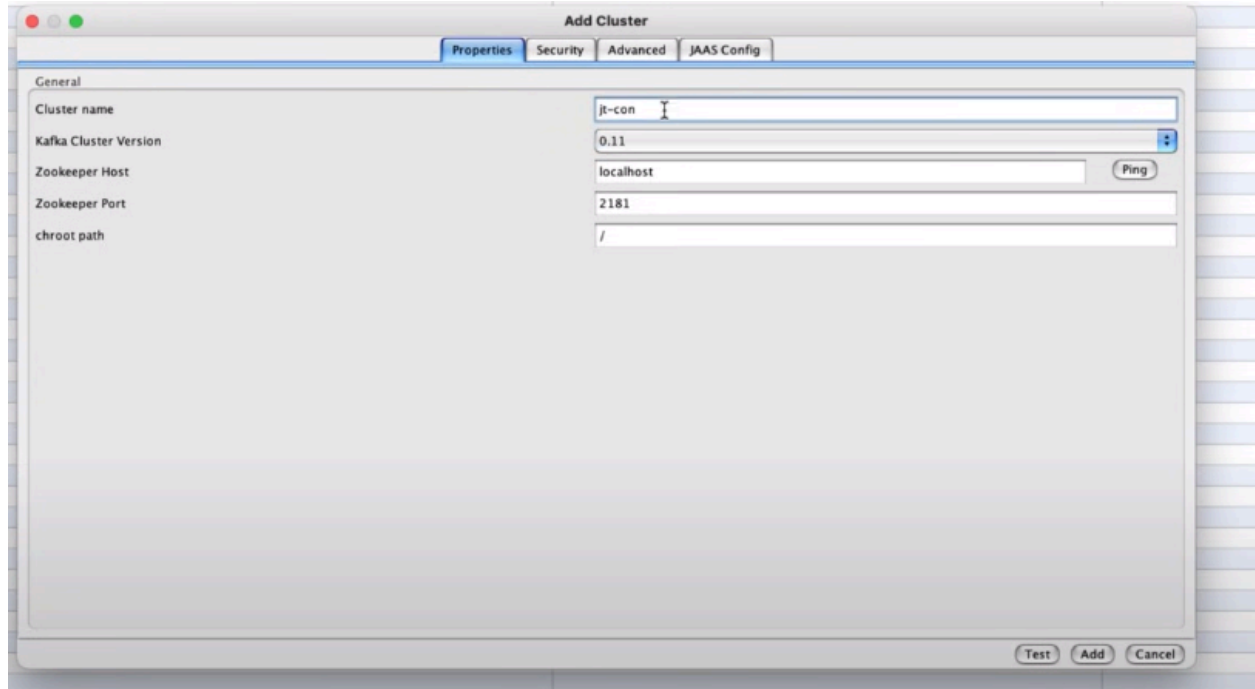
Installing Kafka on Docker.

For installing Kafka on docker container We need two things.

1. Instance of zookeper.
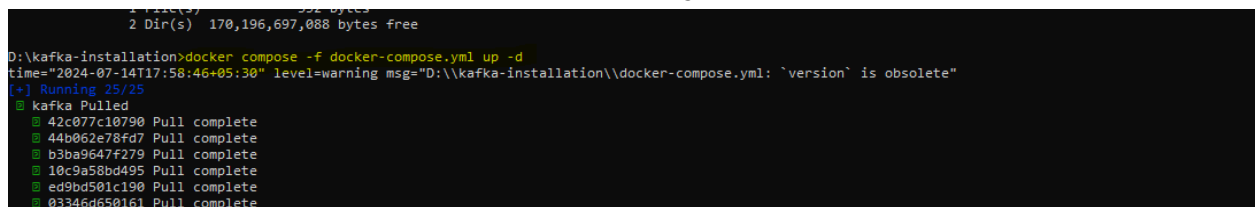2. Instance of Kafka

Create normal project as below image and follow docker-compose.yml

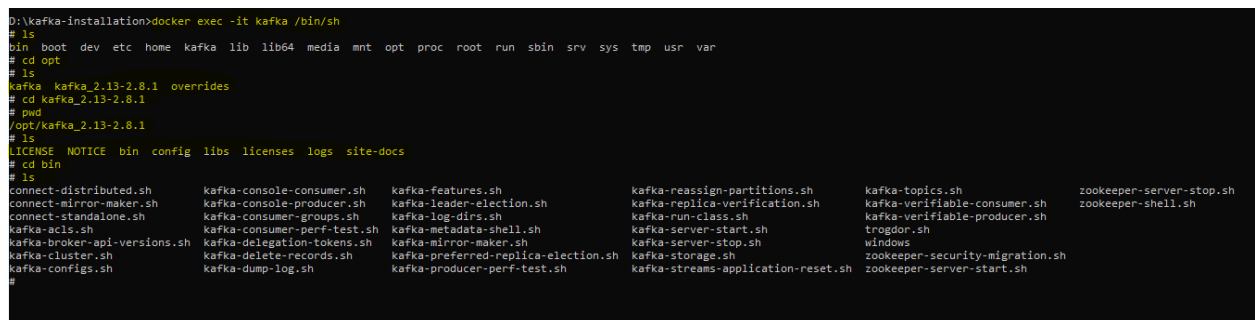Below image to pull the docker image by commands

Note:= Docker desktop should be open.

Command = **docker compose -f docker-compose.yml up -d**

To ensure kafka is installed or not follow below steps.

1. docker exec -it kafka /bin/sh

Command = **kafka-topics.sh --create --zookeeper zookeeper:2181 --replication-factor 1 --partitions 1 --topic test-topic**

```
#
#
# kafka-topics.sh --create --zookeeper zookeeper:2181 --replication-factor 1 --partitions 1 --topic test-topic
Created topic test-topic.
# pwd
/opt/kafka_2.13-2.8.1
#
```