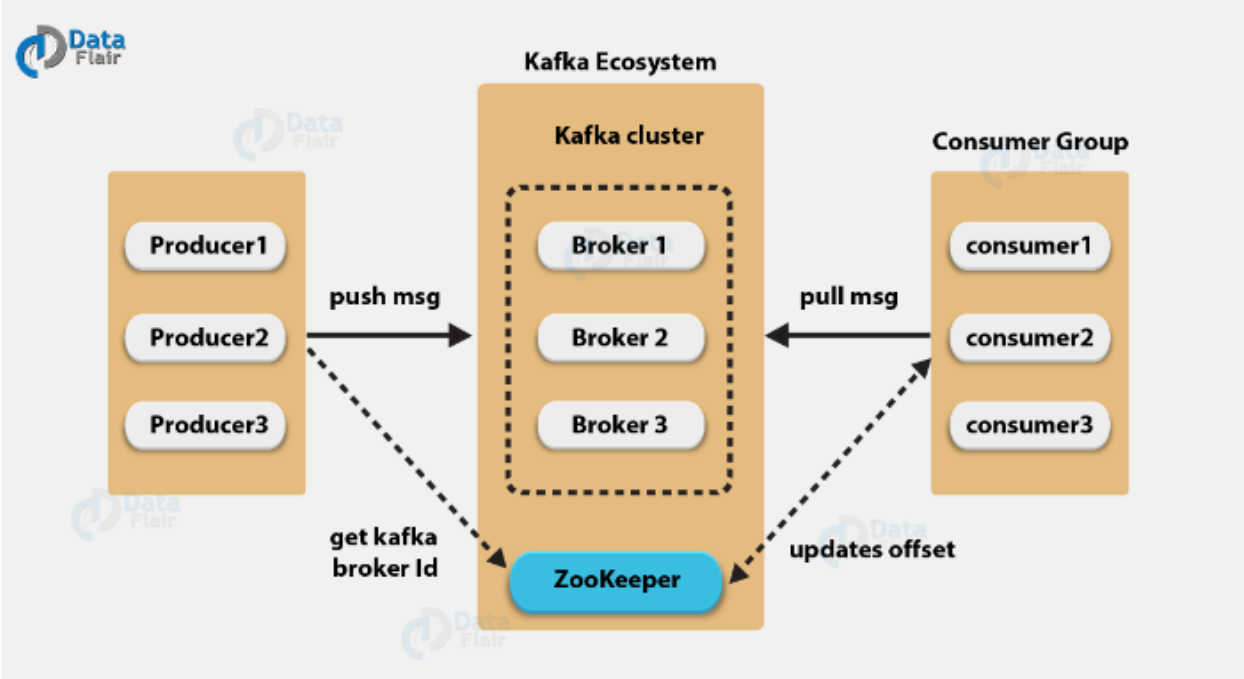
What is Apache Kafka?

Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.



1. Kafka Cluster

Since Kafka is a distributed system, it acts as a cluster. A Kafka cluster consists of a set of brokers. A cluster has a minimum of 3 brokers.

3. Kafka Producer

Producer is an application that sends messages. It does not send messages directly to the recipient. It sends messages only to the Kafka server.

The following diagram shows Producer sends messages directly to Kafka broker:

4. Kafka Consumer

Consumer is an application that reads messages from the Kafka server.

If producers are sending data, they must be sending it to someone, right? The consumers are the recipients. But remember that the producers don't send data to a recipient address. They just send it to the Kafka server.

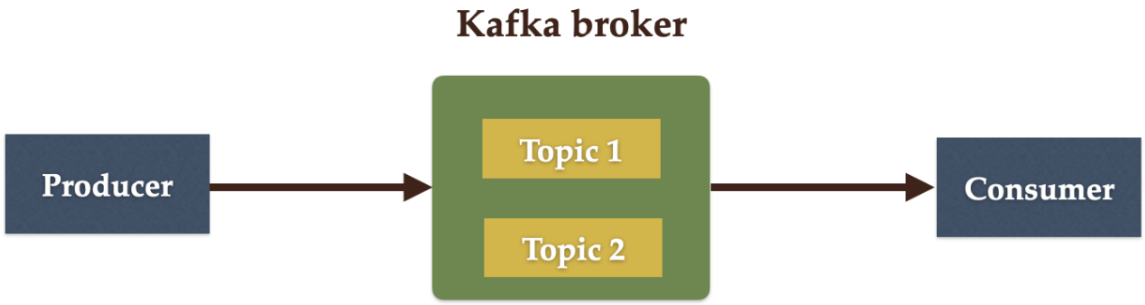
Anyone who is interested in that data can come forward and take it from the Kafka server. So, any application that requests data from a Kafka server is a consumer, and they can ask for data sent by any producer provided they have permission to read it.

5. Kafka Topic

We learned that 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.

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

The following diagram shows two Topics are created in a Kafka broker:



5. Kafka Zookeeper

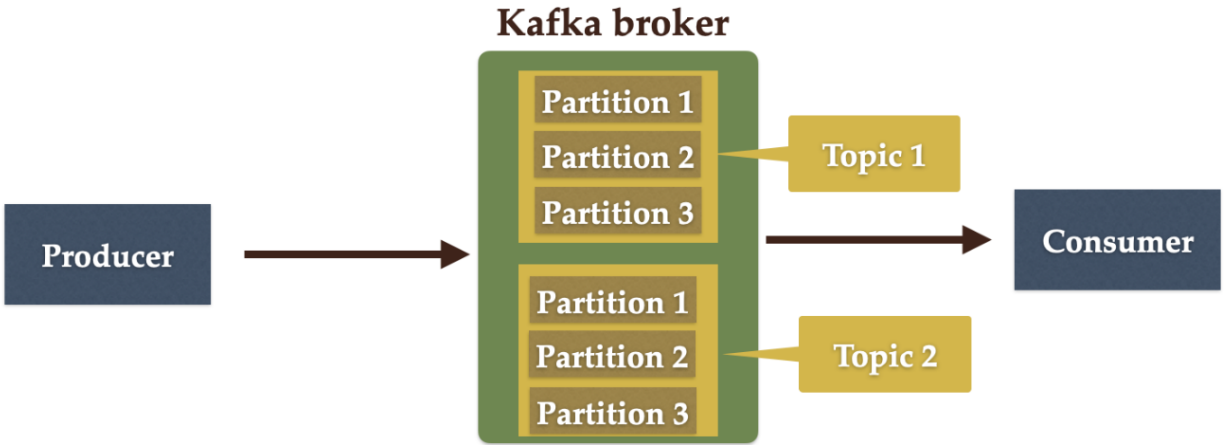
Zookeeper is used by Kafka brokers to determine which broker is the leader of a given partition and topic and perform leader elections. Zookeeper stores configurations for topics and permissions. Zookeeper sends notifications to Kafka in case of changes (e.g. new topic, broker dies, broker comes up, delete topics, etc. ...

6. Kafka Partitions

Kafka topics are divided into a number of partitions, which contain records in an unchangeable sequence.

Kafka Brokers will store messages for a topic. But the capacity of data can be enormous and it may not be possible to store in a single computer. Therefore it will be partitioned into multiple parts and distributed among multiple computers since Kafka is a distributed system.

The following diagram shows Kafka's topic is further divided into a number of partitions:



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.

```
# Start the ZooKeeper service
# Note: Soon, ZooKeeper will no longer be required by Apache Kafka. $
bin/zookeeper-server-start.sh config/zookeeper.properties
```

Note have to install inside the extract kafka file
Got to below directive and hit enter

```
C:\Users\user\Downloads\kafka>.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
```

4. Start Kafka Broker

```
# Start the Kafka broker service
$ bin/kafka-server-start.sh config/server.properties
```

Note have to install inside the extract kafka file

Got to below directive and hit enter

```
C:\Users\user\Downloads\kafka>.\bin\windows\kafka-server-start
.bat .\config\server.properties
```

5. How to create a Topic

```
.\bin\windows\kafka-topics.bat --create --topic topic_demo
--bootstrap-server localhost:9092
```

6. How to write some event or produce the topic

```
.\bin\windows\kafka-console-producer.bat --topic topic_demo
--bootstrap-server localhost:9092
```

7. How to read the event or producer data

```
.\bin\windows\kafka-console-consumer.bat --topic topic_demo
--from-beginning --bootstrap-server localhost:9092
```