

## PRACTICAL-8

**AIM:** Install the Apache Kafka (Single Node) on Windows and perform practical on Kafka.

**Apache Kafka is** a open source distributed data store optimized for ingesting and processing streaming data in real-time. Streaming data is data that is continuously generated by thousands of data sources, which typically send the data records in simultaneously. A streaming platform needs to handle this constant influx of data, and process the data sequentially and incrementally.

Kafka provides three main functions to its users:

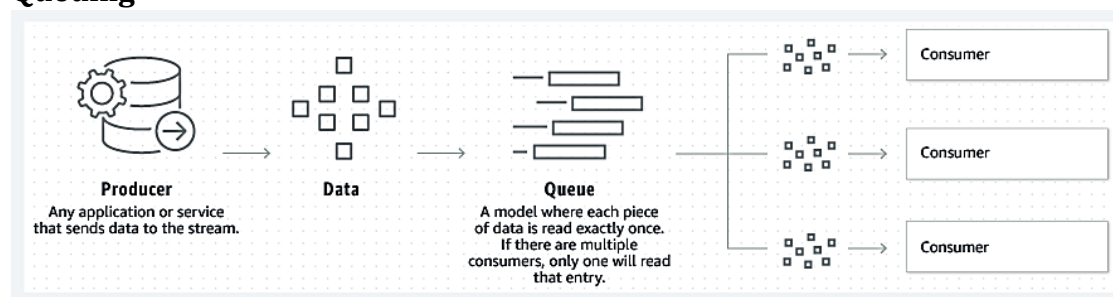
- Publish and subscribe to streams of records
- Effectively store streams of records in the order in which records were generated
- Process streams of records in real time

Kafka is primarily used to build real-time streaming data pipelines and applications that adapt to the data streams. It combines messaging, storage, and stream processing to allow storage and analysis of both historical and real-time data.

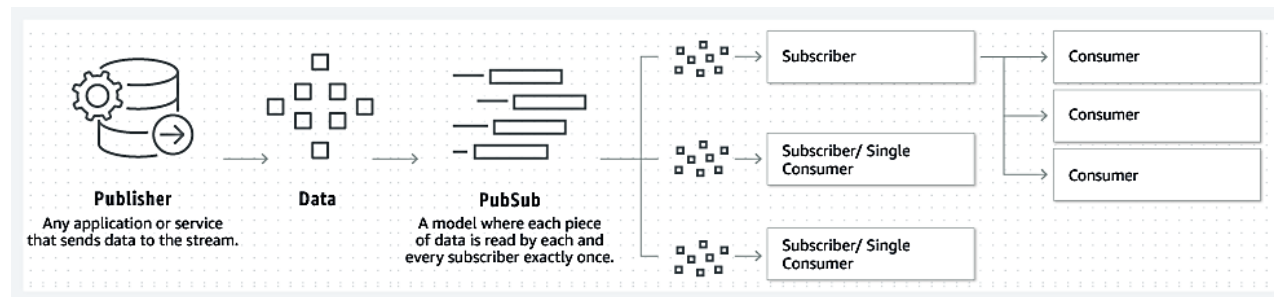
**Kafka is used to** build real-time streaming data pipelines and real-time streaming applications. A data pipeline reliably processes and moves data from one system to another, and a streaming application is an application that consumes streams of data. For example, if you want to create a data pipeline that takes in user activity data to track how people use your website in real-time, Kafka would be used to ingest and store streaming data while serving reads for the applications powering the data pipeline. Kafka is also often used as a message broker solution, which is a platform that processes and mediates communication between two applications.

**Kafka combines two messaging models**, queuing and publish-subscribe, to provide the key benefits of each to consumers. Queuing allows for data processing to be distributed across many consumer instances, making it highly scalable. However, traditional queues aren't multi-subscriber. The publish-subscribe approach is multi-subscriber, but because every message goes to every subscriber it cannot be used to distribute work across multiple worker processes. Kafka uses a partitioned log model to stitch together these two solutions. A log is an ordered sequence of records, and these logs are broken up into segments, or partitions, that correspond to different subscribers. This means that there can be multiple subscribers to the same topic and each is assigned a partition to allow for higher scalability. Finally, Kafka's model provides replayability, which allows multiple independent applications reading from data streams to work independently at their own rate.

### Queuing



## Publish-Subscribe



## Benefits of Kafka

**Scalable:** Kafka's partitioned log model allows data to be distributed across multiple servers, making it scalable beyond what would fit on a single server.

**Fast:** Kafka decouples data streams so there is very low latency, making it extremely fast.

**Durable:** Partitions are distributed and replicated across many servers, and the data is all written to disk. This helps protect against server failure, making the data very fault-tolerant and durable.

Perform the following steps to install the Kafka on Windows.

1.	Download Kafka from <a href="https://kafka.apache.org/downloads">https://kafka.apache.org/downloads</a> (Version: 2.13)
2.	Extract the file using 7ZIP application. (Extract at C:\kafkaV2-13)
3.	Create folder "data" inside C:\kafkaV2-13
4.	Inside "data" folder creates two folders named "kafka" and "zookeeper"
5.	Go to C:\kafkaV2-13\config Open the "Zookeeper.properties" file with any text editor. In the opened file, replace your "datadir" location with the copied Zookeeper folder path (dataDir=C:/kafkaV2-13/data/zookeeper) (Note: Make sure you change the path with forwarding slashes instead of backward slashes)
6.	Go to C:\kafkaV2-13\config Open the server.properties file with any text editor. In the server.properties file, replace the "log.dirs" location with the copied Kafka folder path (log.dirs=C:/kafkaV2-13/data/kafka)(Note:Make sure you change the path with forwarding slashes instead of backward slashes)  Now, you have made the necessary changes and configurations to the Kafka files and are ready to set up and start Kafka on your computer. After configuring Zookeeper and Kafka, you have to start and run Zookeeper and Kafka separately from the command prompt window.
7.	A) Starting Zookeeper Open the command prompt Goto C:\kafkaV2-13\bin\windows and run following command:

	<p>.\zookeeper-server-start.bat ../../config/zookeeper.properties (Note: If Zookeeper is not started then set the java path JAVA_HOME=C:\Progra~1\Java\jdk-22)</p> <p>(You can see from the output that Zookeeper was initiated and bound to port 2181. By this, you can confirm that the Zookeeper Server is started successfully. Do not close the command prompt to keep the Zookeeper running.)</p>
8.	<p><b>B) Starting Kafka</b> Open another command prompt window and type the below command. Goto C:\kafkaV2-13\bin\windows and run following command: .\kafka-server-start.bat ../../config/server.properties</p> <p>(The Kafka Server has started successfully and is ready for Streaming Data. Now, both Zookeeper and Kafka have started and are running successfully. To confirm that, navigate to the newly created Kafka and Zookeeper folders. When you open the respective Zookeeper and Kafka folders, you can notice that certain new files have been created inside the folders.)</p>
9.	<p><b>Perform following in Kafka:</b></p> <p><u>Testing Kafka by Creating a Topic:</u> As you have successfully started Kafka and Zookeeper, you can test them by creating new Topics and then Publishing and Consuming messages using the topic name. Topics are the virtual containers that store and organize a stream of messages under several categories called Partitions. Each Kafka topic is always identified by an arbitrary and unique name across the entire Kafka cluster.</p> <p>In the below steps, you will learn how to create topics:</p> <p>For creating a topic, open a new command prompt and write the below command: goto C:\kafkaV2-13\bin\windows and run following command: kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic test</p> <p>(Output: Created topic test.)</p>
10.	<p>Open another CMD terminal Start sending messages to the topic. Run this command to launch the producer: Run following command: C:\kafkaV2-13\bin\windows&gt;kafka-console-producer.bat --topic test --bootstrap-server localhost:9092 Type in your messages (enter atleast five messages), and hit Enter. (Note: Don't close the terminal (CMD prompt))</p>
11.	<p>Open another CMD terminal consume and see the messages which you/producer sent: Run the following command:</p>

	C:\kafkaV2-13\bin\windows>kafka-console-consumer.bat --topic test --from-beginning --bootstrap-server localhost:9092
12.	List the topics Command: C:\kafkaV2-13\bin\windows>kafka-topics.bat --list --bootstrap-server localhost:9092

**Perform following:**

1. Create a topic "Mobile" with 2 Partitions and 3 Replication Factors.
2. Produce the messages: "Samsung", "Nokia", "Vivo", "Sony", "Xiaomi"
3. Consume the messages