Kafka

What is streaming Data?

Streaming data denotes a continuous influx of information originating from diverse sources, which is ingested, processed, and analyzed in real time.

In contrast to batch processing, where data is accumulated and handled in discrete intervals, streaming data enables on-the-fly computation and decision-making, facilitating instantaneous insights and responsive actions.

Characteristics of Streaming Data

Data streams are continuously generated from various sources. Data is often treated as a series of events. Data is processed as it is generated, enabling immediate analysis and allowing for real-time analysis and decision-making.

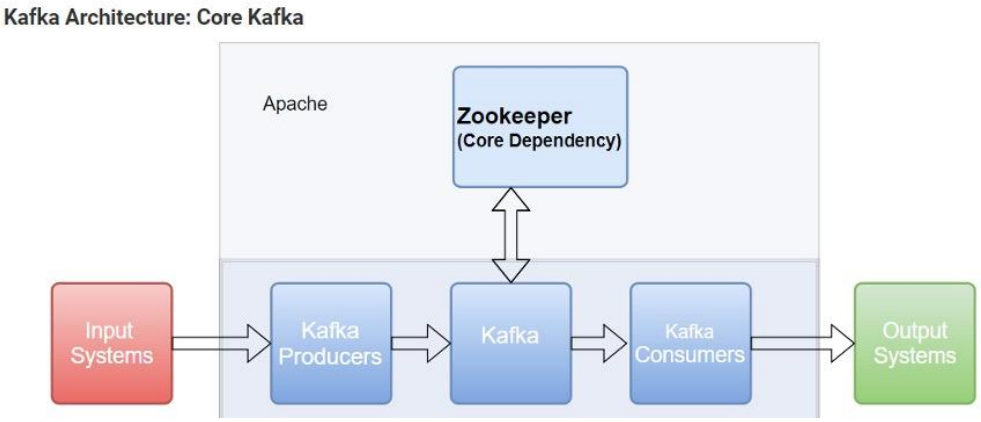
Examples of Streaming Data:

Financial transactions,stock market data, social media sentiment, data from sensors and other IoT devices, network traffic.

What is Kafka?

Kafka is a tool that helps move large amounts of data quickly and reliably between systems. It works like a messenger that collects and delivers data in real-time or later, making it useful for both live and stored data processing.

Kafka was developed by Linkedin and open-sourced under the Apache software foundation.



Why Kafka?

Every day, huge amounts of data are created from user actions like logins, clicks, likes, and from machines. This data is important not just for analysis later, but also for real-time tasks like search, recommendations, ads, and security.

Old systems saved logs on each machine, which was slow and only useful for offline analysis. Tools like Facebook's Scribe and Yahoo's Data Highway helped with that, but they were built mainly for large data storage systems like Hadoop.

Kafka was created to handle this data faster and better, especially for real-time use, with delays of just a few seconds.

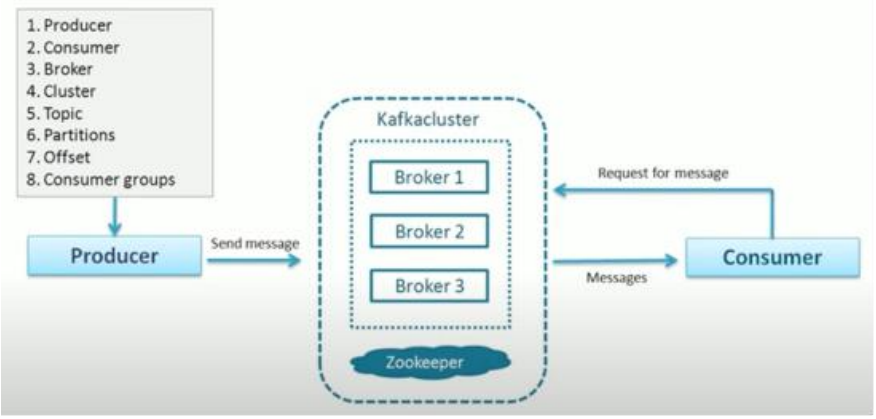
What Things a Kapka can do for us?

- It can be used as an enterprise messaging system, we can use it to simplify complex pipeline that are made up a vast number of consumers and producers.
- It can be used for stream processing which consists two parts: stream and processing framework. Kapka gives us stream and we can plug-in processing framework.
- It also provides connectors to import and export bulk of data from databases and other systems.

General Terms used in Kafka

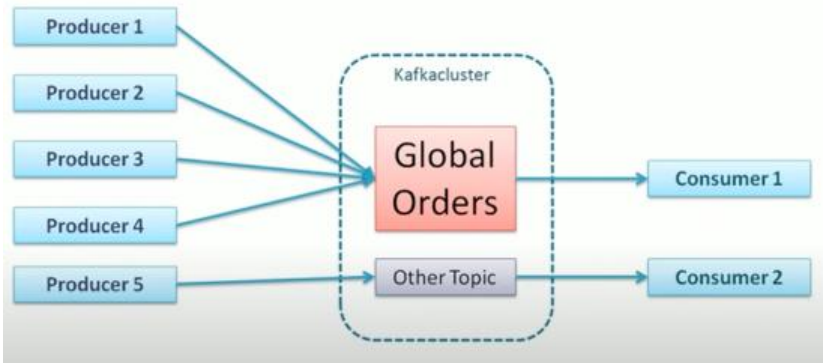
Producer: The producer is an application that sends messages / data to Kafka. Message may have different meaning of schema for us but for Kapka it is simple array of bytes.

Cluster: Cluster is a group of computers sharing workload for a common purpose. Since Kapka is a distributed system each computer executes one instance of Kapka Broker, so that instead of cluster we can denote Kapka as cluster.

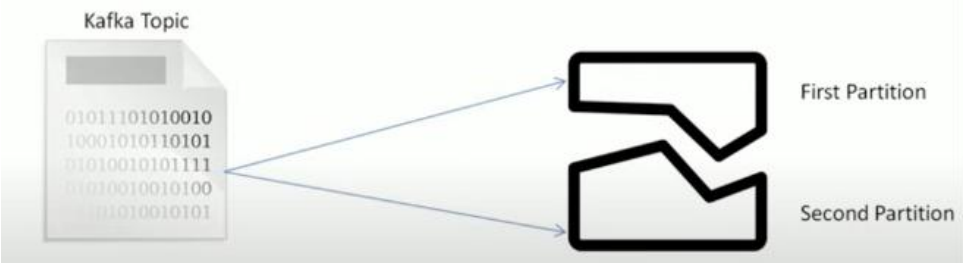


Topic: A topic is a unique name for Kapka stream. Producer send data to Kapka broker. Consumer can ask for data from Kapka broker. But which data set consumer is needed? Topic is a labeled set of data for which consumer may ask for.

For example, 5 producers send data to Kapka Cluster. Kapka Cluster arranges all the data in groups or summarized data in groups according to the demand of consumers. Each defined group is called stream and name of each stream is called topic. Global orders is a topic in the picture. Each consumer may registered for more than one topics and will get data set topic-wise automatically.



Partitions: Data may be huge and sometimes it may the larger than the capacity of a single computer. So the main challenge is to store the data. One solution is to break the data into small parts and distributed to multiple computers so that it will be fitted with the capacity of a computer. Each partition is stored one machine. Each part is called partition. Number of partition have to fix by the designer in advance an it can not be change later.

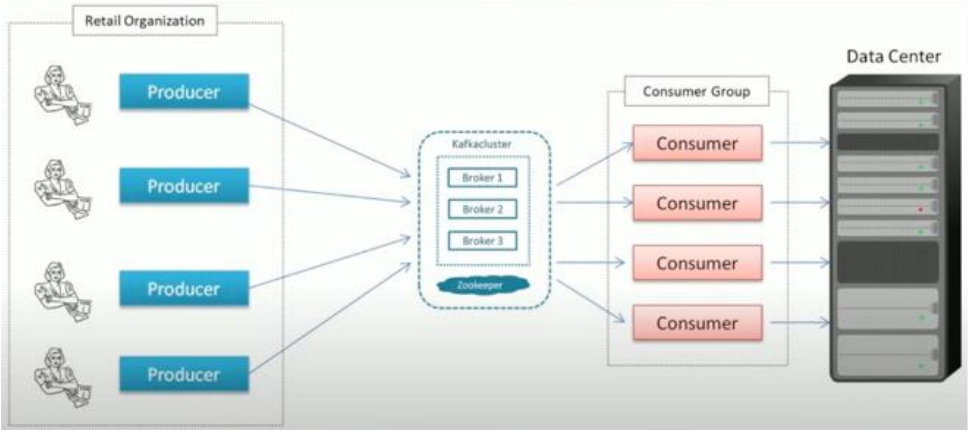


Offset

Offset is a sequence id which are given to messages as they arrived in a partition. Once a number assigned to a message it cannot be changed. First message gets an offset 0 and the second message gets an offset 2 and so on. These offset numbers are local. So to find a message we need topic id, partition id and offset id.

Consumer Group:

A group of consumers acting as a single logical unit. Members of the same group share the values.



What is Zookeeper?

Apache ZooKeeper was first developed by Yahoo! in the mid-2000s. Yahoo created it to help manage distributed applications in a reliable and scalable way. The main goal was to build a central coordination service that could keep different parts of a system in sync, detect failures, and manage settings easily. Later, Yahoo gave ZooKeeper to the Apache Software Foundation, and it became a top-level Apache project in 2010. Since then, ZooKeeper has been used widely in the big data world. It has become an important part of many systems like Apache Kafka, Apache HBase, Apache Hadoop, and Apache Solr.

In Apache Kafka, ZooKeeper has a key role. It works as a central service that stores configuration data, helps manage the system, and supports tasks like leader election and synchronization between different parts of the Kafka cluster.

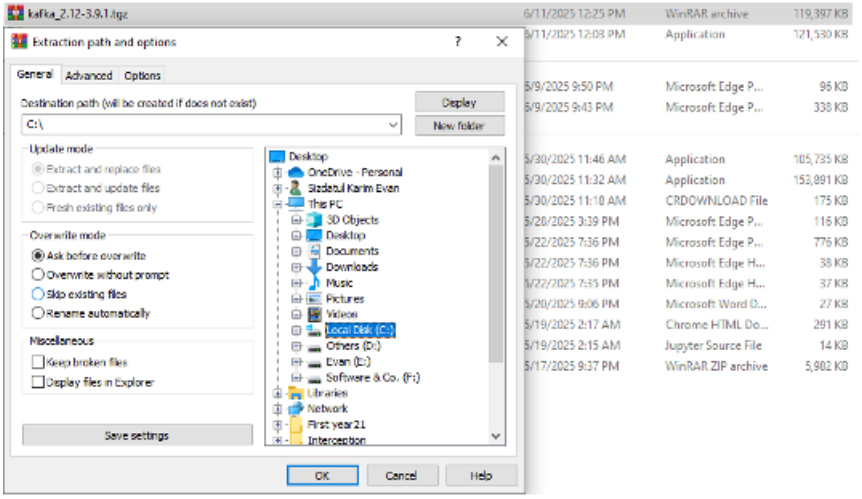
What are the functions of ZooKeeeper in Apache Kapka?

ZooKeeper plays a crucial role in managing and coordinating the Kafka cluster. The following are the main functions of ZooKeeper in Apache Kafka:

- ZooKeeper maintains information about Kafka brokers (nodes) in the cluster. It helps keep track of controller broker and brokers that are currently alive
- ZooKeeper helps in electing a controller broker in case of failure of any leader broker.
- ZooKeeper stores metadata such as Topics and partitions, Broker IDs, Access Control Lists (ACLs)
- All the configuration of Kafka nodes has written in the configuration settings of ZooKeeper, ensuring consistent configurations across the cluster.
- ZooKeeper provides a watcher mechanism when a change occurs in the total system, it notifies the interested components like brokers or the controller.

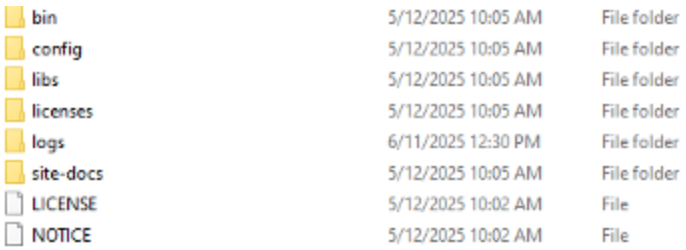
1. Kapka Installation

Download Kafka from: https://kafka.apache.org/quickstart and extract it in your c drive.



2. Checking to the folder

Go to the folder



The bin folder contains some command line of Kapka tools:

The config folder contains all the configuration settings

The libs folder contains all the ".ger" files

3. Starting Kafka

Since Kapka uses Zookeeper we have to activate Zookeeper first. It is available in the bin folder. To start Zookeeper open a command terminal:

.*bin**windows**zookeeper-server-start.bat* .*config**zookeeper.properties*

The above command has one parameter only ".\config\zookeeper.properties" this is default configuration.

Here we will see the port number where bind it information, normally it is 2181

4. Starting Kafka Broker.

To start Kafka Broker with default configuration open a command terminal:

.\bin\windows\kafka-server-start.bat .\config\server.properties

A lot of information will appear and an Apache server has been started. Make sure that there are no errors.

5. Creating Kafka topic

Kafka creates topic automatically but we want to give a name here. When a producer creates a message when there is no existence of any topic, kafka creates a topic and accept the message. Let us create a topic using topic manage tool:

.\bin\windows\kafka-topics.bat --bootstrap-server localhost:2181

--create --topic patwary --partitions 2 --replication-factor 1

1. First parameter is zookeeper server address and port number
2. Second parameter is After create command with topic name, partition number, and replication factors.

Here we have a single broker but how is it possible to create 2 partitions? This is not a problem at all. . Kafka will distribute partitions among all of its brokers. So both the partions will be created into one broker.

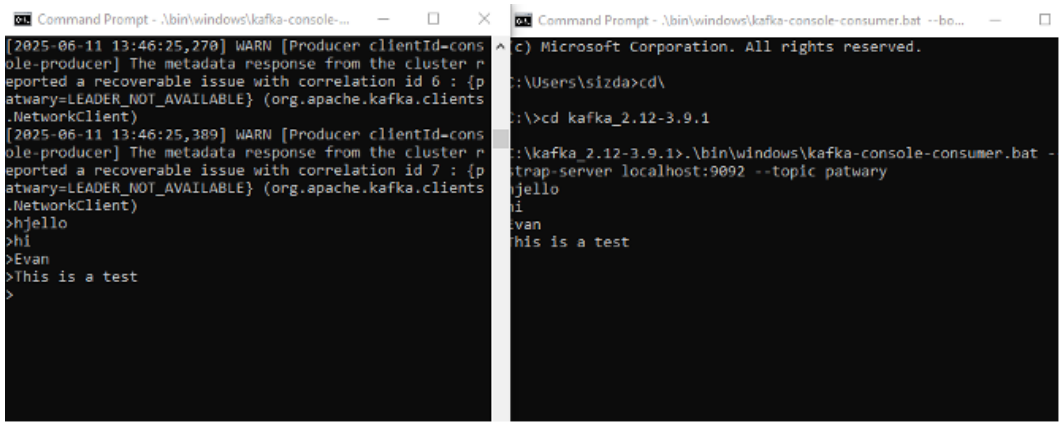
6. Open Two Terminals

Now we need to open two cmd terminals one for producer and another for consumer using default port 9091 or 9092.

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

In another terminal

.\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic patwary


Command Prompt - .\bin\windows\kafka-console-...
[2025-06-11 13:46:25,270] WARN [Producer clientId=cons
ole-producer] The metadata response from the cluster r
eported a recoverable issue with correlation id 6 : {p
atwary=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients
.NetworkClient)
[2025-06-11 13:46:25,389] WARN [Producer clientId=cons
ole-producer] The metadata response from the cluster r
eported a recoverable issue with correlation id 7 : {p
atwary=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients
.NetworkClient)
>hjello
>hi
>Evan
>This is a test
>
Command Prompt - .\bin\windows\kafka-console-consumer.bat --bo...
c) Microsoft Corporation. All rights reserved.
:\Users\sizda>cd\
:\>cd kafka_2.12-3.9.1
:\kafka_2.12-3.9.1>.\bin\windows\kafka-console-consumer.bat -
trap-server localhost:9092 --topic patwary
jello
i
van
his is a test