Apache Kafka

Apache kafka is like a communication system that helps different parts of a computer system exchange data by publishing and subscribing to topics.

**Overview of Apache Kafka**

Apache Kafka is a **distributed event streaming platform** designed for high-performance data pipelines, streaming analytics, and data integration. Developed by the Apache Software Foundation, it is an open-source system primarily written in Java and Scala. Kafka provides a unified, high-throughput, low-latency platform for handling real-time data feeds, making it suitable for various applications, including mission-critical systems

**Key Features**

* **High Throughput and Low Latency**: Kafka is optimized for ingesting and processing streaming data in real-time, allowing it to handle large volumes of data efficiently
* **Scalability**: Kafka can scale horizontally, meaning you can add more brokers to handle increased loads without significant changes to the existing architecture.
* **Durability and Reliability**: Kafka stores streams of records in a fault-tolerant manner, ensuring that data is not lost even in the event of failures.
* **Stream Processing**: Kafka supports stream processing through its Kafka Streams API, enabling developers to perform complex transformations and analytics on data as it flows through the system.

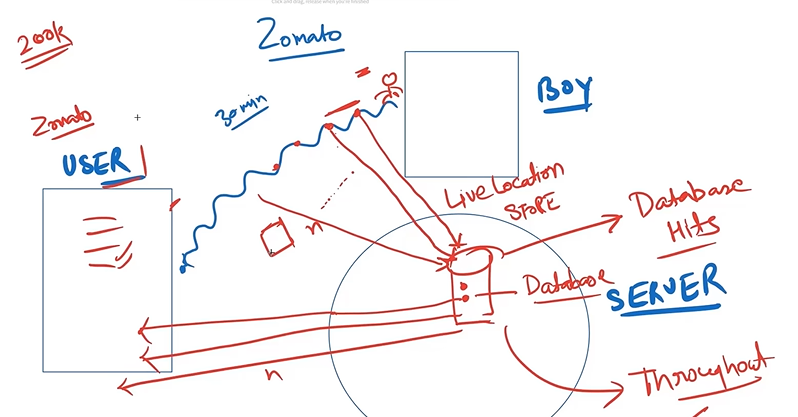
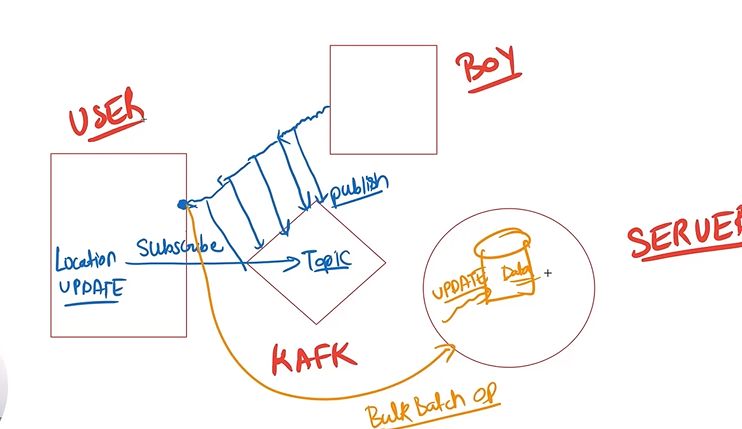
**Use Cases**

Kafka is widely used across various industries for applications such as:

* **Real-Time Analytics**: Companies use Kafka to analyze data in real-time, enabling immediate insights and decision-making.
* **Data Integration**: Kafka acts as a central hub for integrating data from multiple sources, making it easier to manage and process data across different systems.
* **Event Sourcing**: Kafka can be used to implement event sourcing architectures, where state changes are captured as a series of events.

In summary, Apache Kafka is a robust platform that facilitates the handling of real-time data streams, making it an essential tool for modern data-driven applications.  
  
A yellow rectangle with black text

Description automatically generated  
  
A white background with black text

Description automatically generated  
  
Take Zomato without kafka example:-  
  
For sending bulk notifications. We are not allowing user to direct hit to the database multiple times times so we are using kafka.  
  
  
  
Take Zomato with kafka example:-  
  
user and boy both have to subscribe the topic for receiving the notifications.  
  
  
**IMPORTANCE features of Kafka:-**  
  
1. High Throughput:-

We can retrieve million of data within the sec.  
ex. Live tracking

2. Falut Tolerance (Replication)

Data store in multiple copies of data

3.Durable

4.Scalable

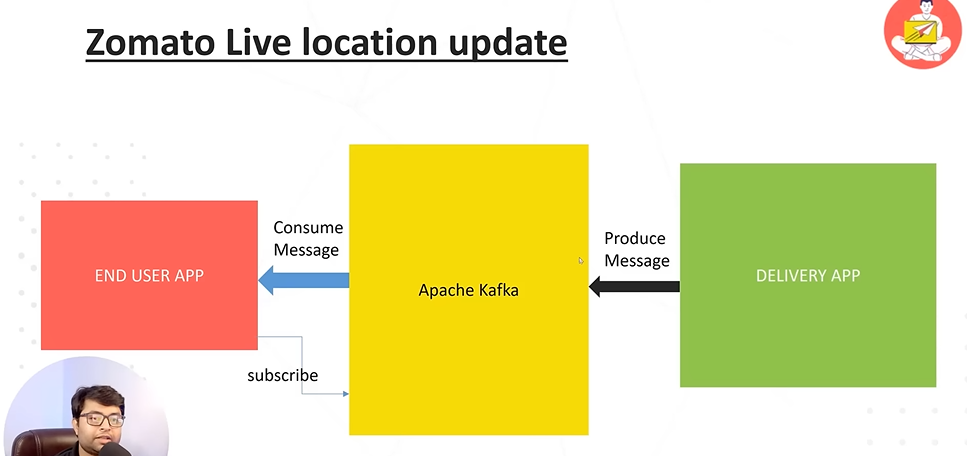
**Kafka Architecture:-**Zookeeper :- manage the state of Kafka, manage the state of broker(server)

Inside the broker there are multiple topics. Topics is used to put data in category wise.

Inside the topic there are partition , and partition have offset.  
  
**A diagram of a structure

Description automatically generated**

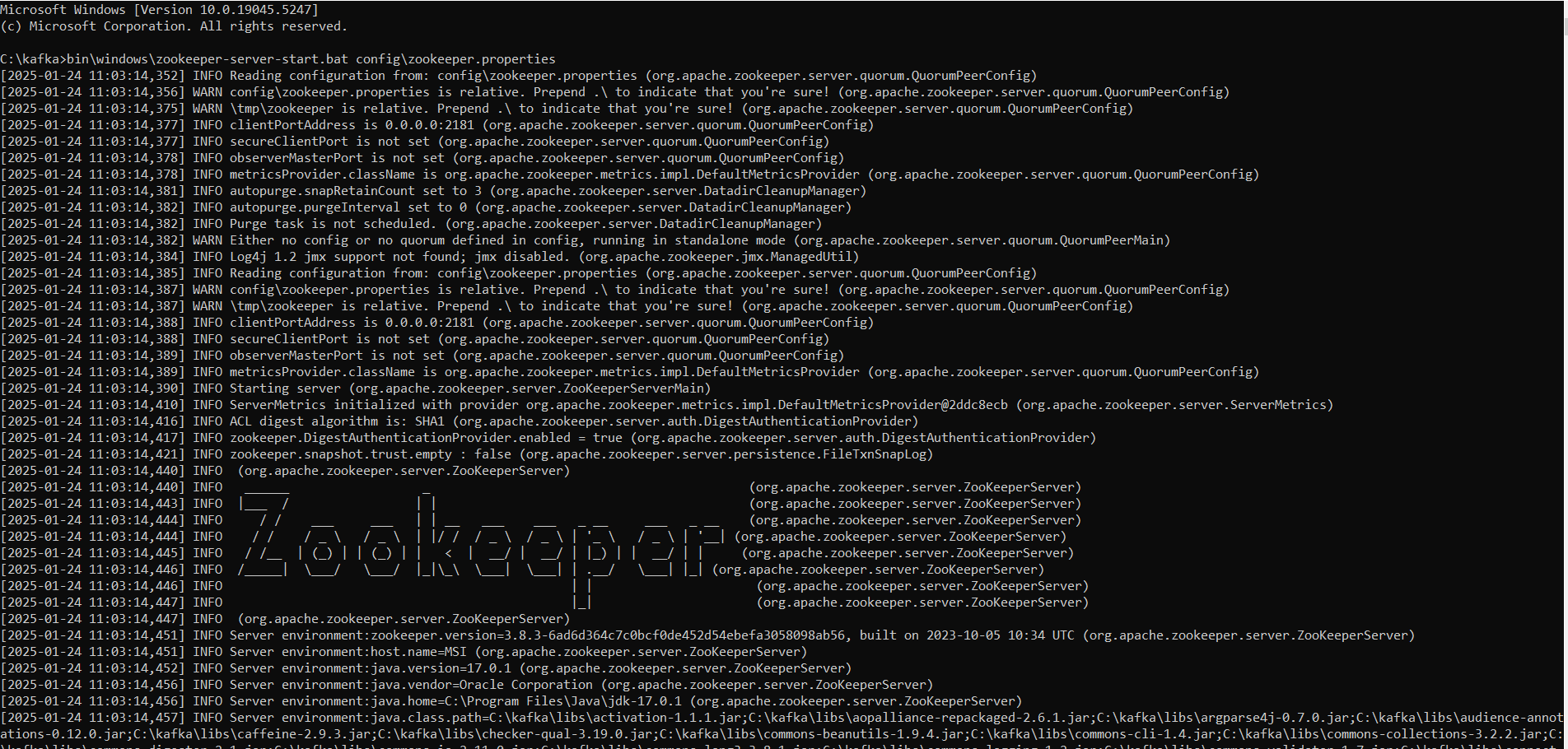
We are creating the 2 microservices

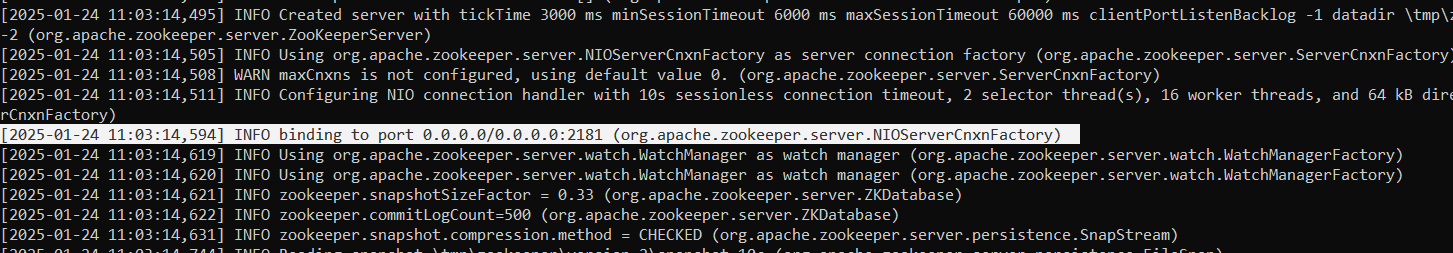
****

**Kafka related commands**

**1. command for starting zookeeper:-**

C:\kafka>bin\windows\zookeeper-server-start.bat config\zookeeper.properties





**2.command for starting server:-**

C:\kafka>bin\windows\kafka-server-start.bat config\server.properties

**3. To see the info related to command.**

C:\kafka\bin>windows\kafka-topics.bat

Create, delete, describe, or change a topic.

**4. To create topic**

C:\kafka\bin>windows\kafka-topics.bat --create --topic USER-TOPIC --bootstrap-server localhost:9092

**5. To see the topic**

C:\kafka\bin>windows\kafka-topics.bat --list --bootstrap-server localhost:9092

USER-TOPIC

**6. to publish message in topic**

C:\kafka\bin>windows\kafka-console-producer.bat --topic USER-TOPIC --bootstrap-server localhost:9092

>hi this is my msg

**7. To consume the message from topic**

C:\kafka>bin\windows\kafka-console-consumer.bat --topic USER-TOPIC --from-beginning --bootstrap-server localhost:9092

