**Apache Kafka**

**Defination**: Apache Kafka is a software platform which is based on a distributed streaming process.

Kafka Producers:

A producer is the one which publishes or writes data to the topics within different partitions

**Message Keys**: to send Data to particullar partitions . If key is not

mention send it send to all in orderly manner

**Acknowledgment** : to the producer can get a confirmation of its data

writes by receiving the following acknowledgments

Kafka Consumer:

A consumer is the one that consumes or reads data from the Kafka cluster via a topic

consumer is not supposed to read data from offset 1 before reading from offset 0

Also have **group of consumer** that can divided based on partition available and if number of consumer is more than partition to read than some of them in **Inactive**

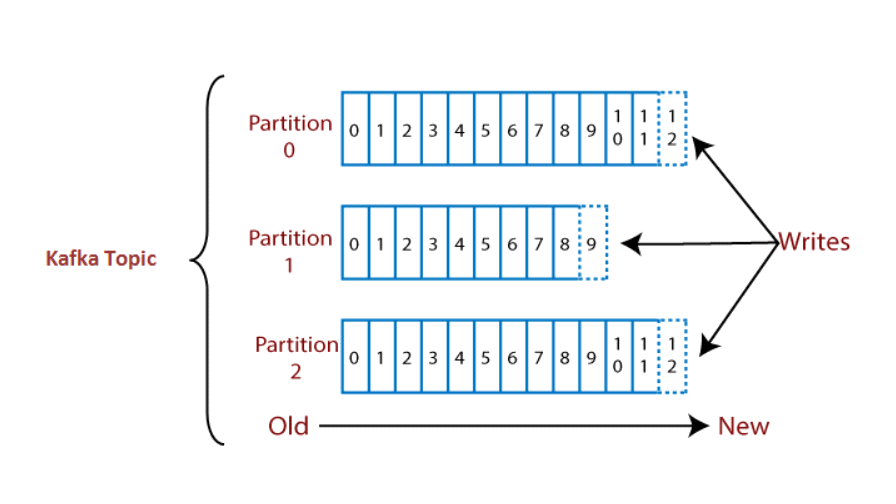
terms:

**Topic :**

In Kafka, the word topic refers to a category or a common name used to store and publish a particular stream of data. Basically, topics in Kafka are similar to **tables** in the database, but not containing all constraints

**Partitions**:

A topic is split into several parts which are known as the partitions of the topic. have order



**OffSet**:

nothing but indexing of partitions.

**Brokers**:

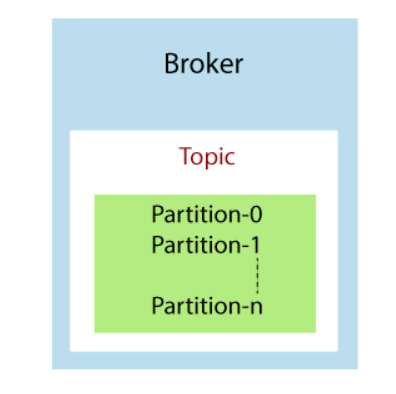
A Kafka cluster is comprised of one or more servers which are known as brokers or Kafka brokers. A broker is a **container** that holds several topics with their multiple partitions.

**Hierarchy**

Broker

--->Topic

--->Partitions



Kafka Topic Replication:

replications allow if any case failure of any Broker then copy of that broker

to other to overcome data loss

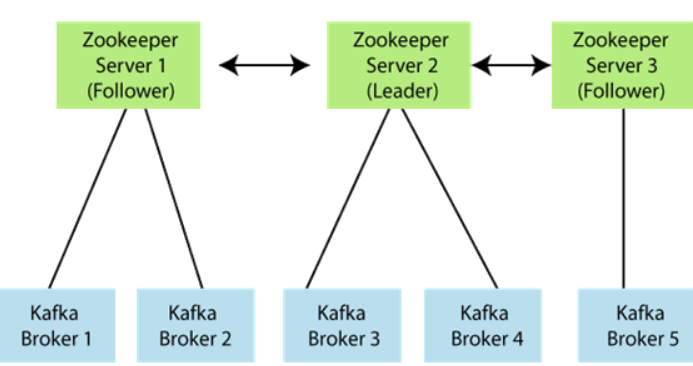
**ZooKeeper**:

A ZooKeeper is used to store information about the Kafka cluster and details of the consumer clients. It manages brokers by maintaining a list of them.

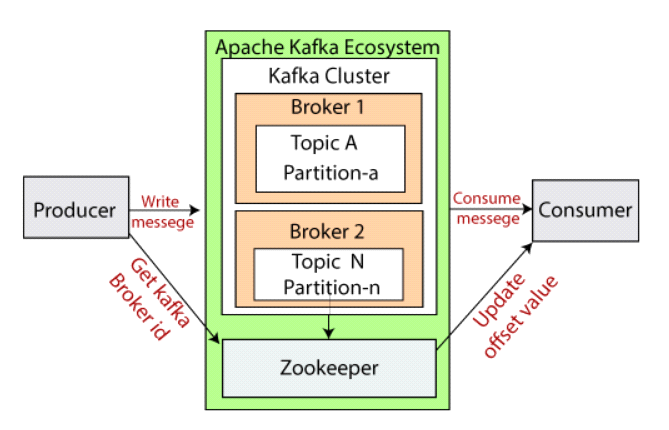
User(Producer) cannot diretly interact with Zookeeper ,

via Broker

for multiple Zookers have leader



Complete Apache Kafka Architecture



Implementation:

Start with commands:

Installing Apache Kafka on Windows

Step 1:

To install Apache Kafka, the java8 JDK kit should be installed on the system. If it is already installed, but with other versions, the user needs to reinstall Java.

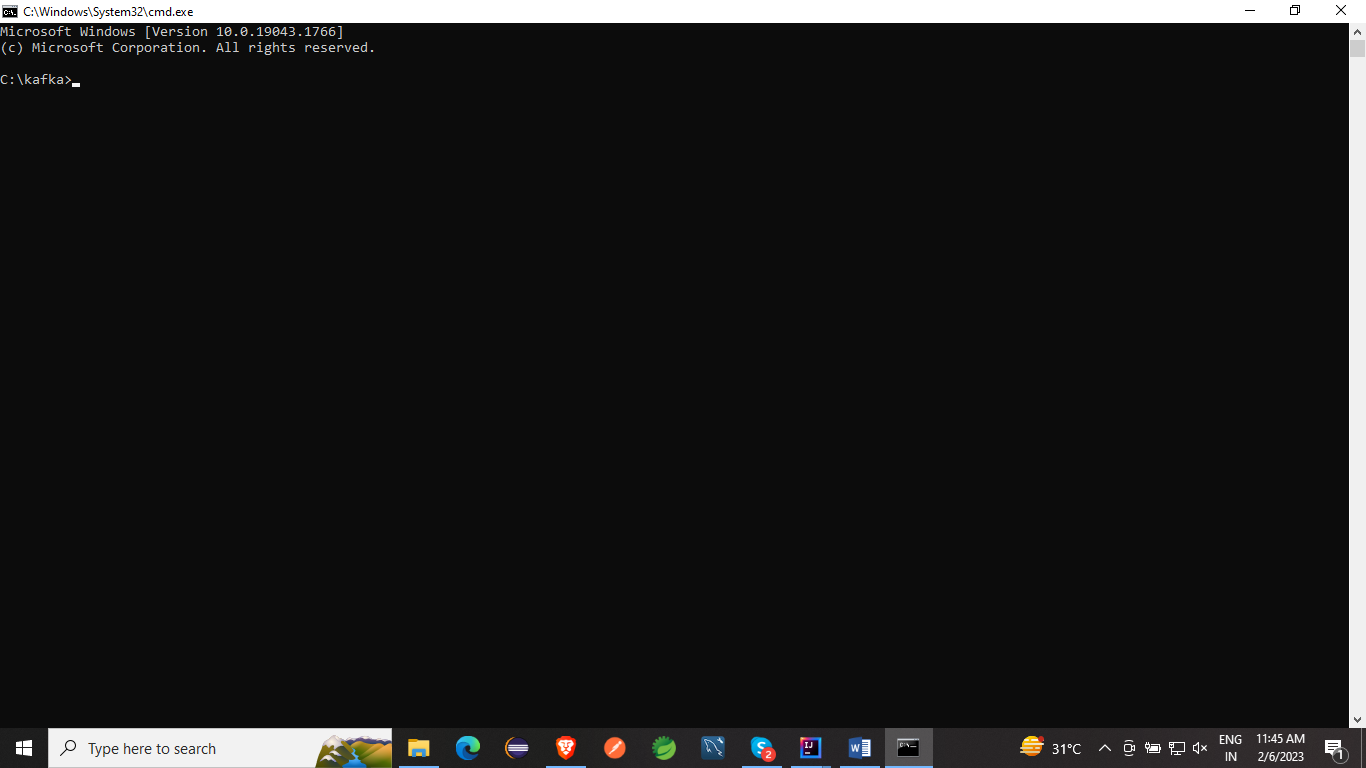
Step2:

Now, download Apache Kafka on the system

Step3:

Open Command Prompt, move to C:\ drive using 'cd..' command. Now, move to the Kafka directory with the help of the following command:

cd <kafka folder name>. Press enter. After entering, it will look like:



Step4:

Check the java version. If correctly installed, the below output would be shown after applying the **'java -version'**

Apache Kafka is unable to run without installing the zookeeper. Therefore, to work with Kafka, the user need to start the zookeeper on the system.

There are following steps to start the zookeeper:

Open the config folder and go to the zookeeper.properties

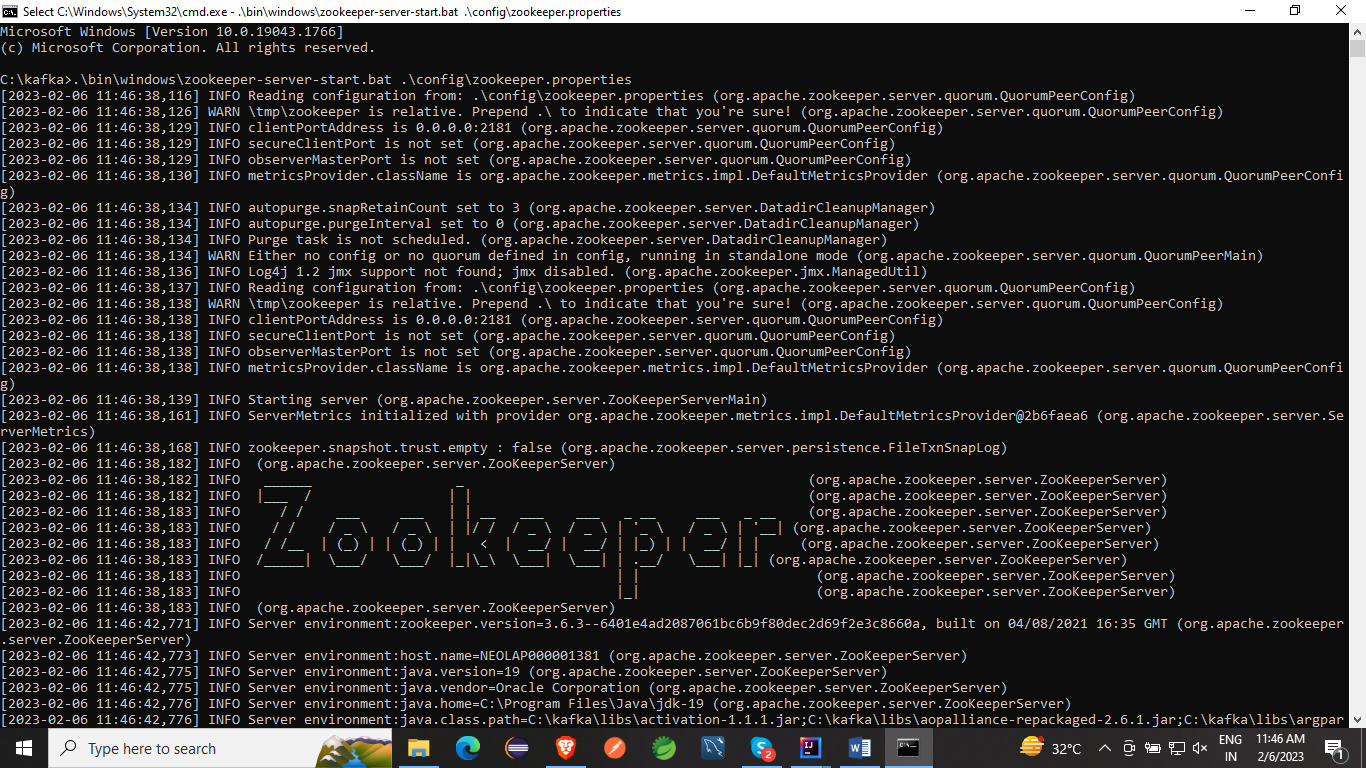
Edit the value of 'datadir' by pasting the zookeeper folder address. Save it.

The Zookeeper server is ready to start. Open the command prompt and go to the Kafka directory.

Then, type the command:

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

Starting Zookeeper Server:



As the zookeeper server is started, go to the kafka directory>data>kafka. Copy the address of the kafka folder.

Move back to the Kafka directory>config>server.properties.

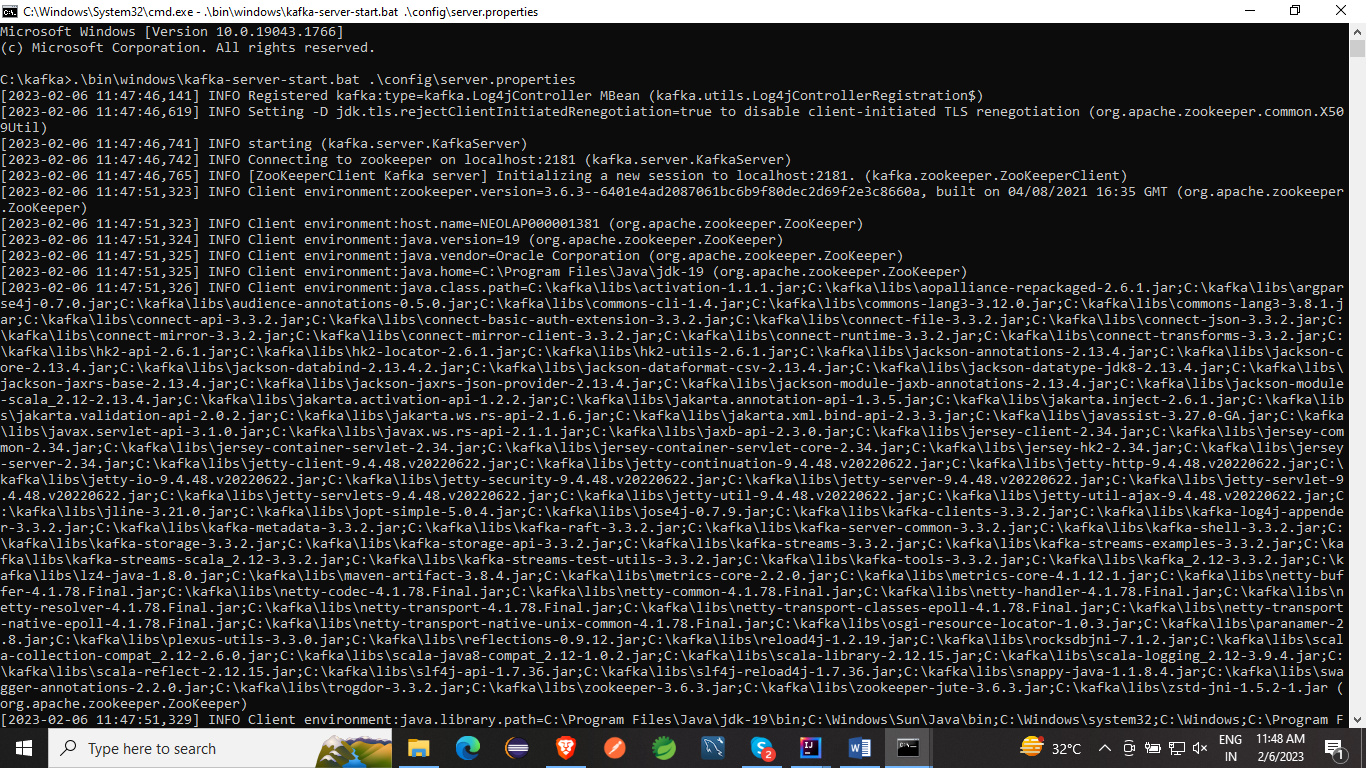
Paste the copied address of the kafka folder as a value of log.dirs.

Open the command prompt and move to the Kafka directory.

Type the command:

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

kafka-server(Broker)



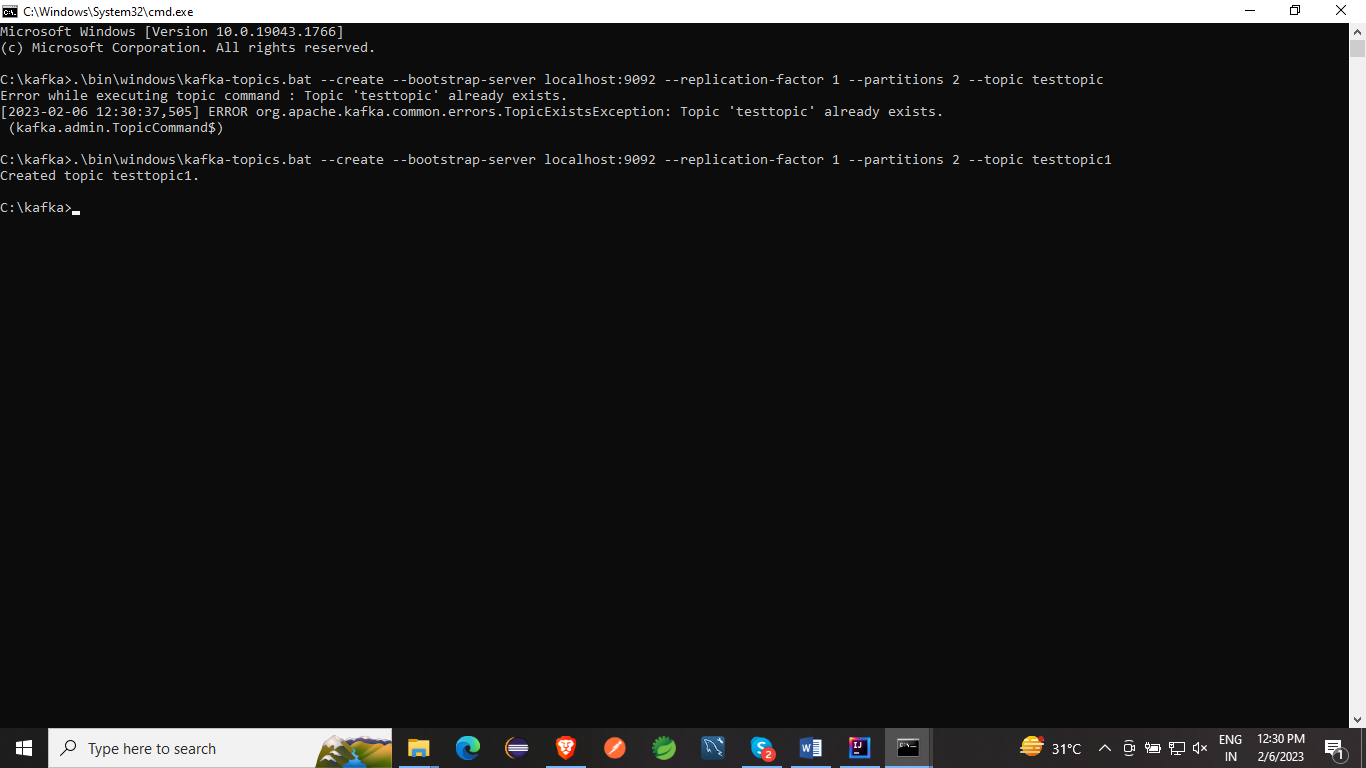
There are following steps used to create a topic:

Step1: Initially, make sure that both zookeeper, as well as the Kafka server, should be started.

Step2: Type 'kafka-topics -zookeeper localhost:9092 -topic -create' on the console and press enter. Here, the name of the topic is 'tested'

**kafka-topics.bat --create --bootstrap-server localhost:9092 --topic tested**

Creating a producer



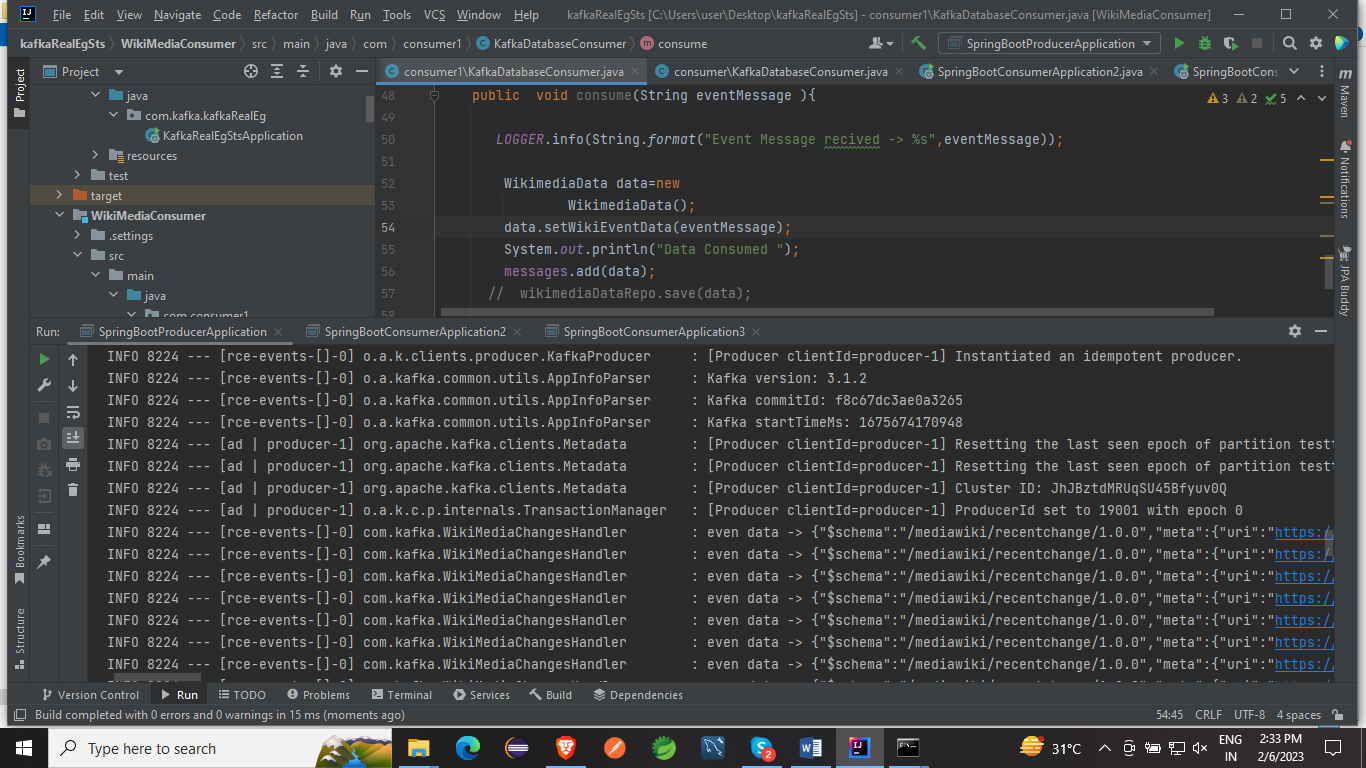
After knowing all the requirements, try to produce a message to a topic using the command:

'kafka-console-producer -broker-list localhost:9092 -topic <topic\_name>'.

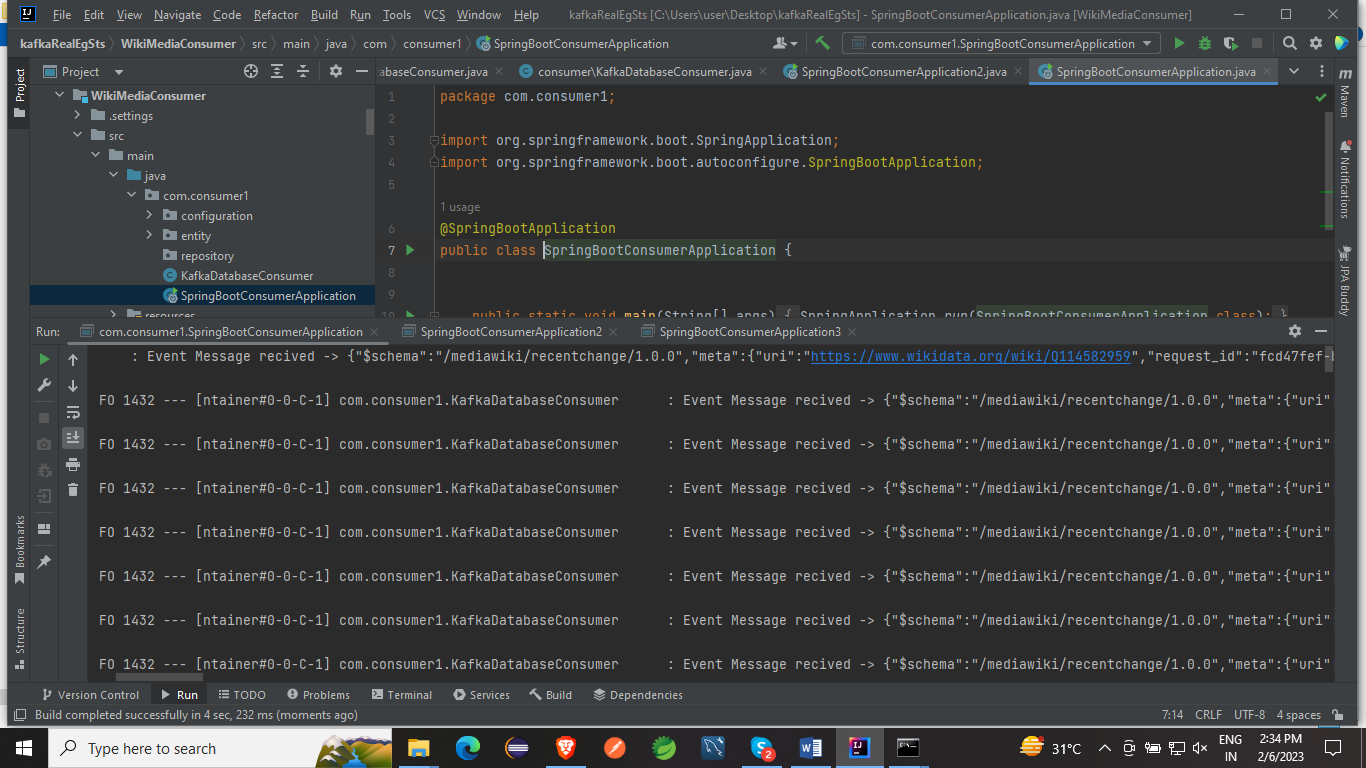
Command:

**.\bin\windows\kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 2 --topic testtopic**

sending message on producer:

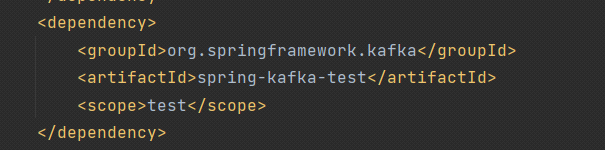


consumer listening

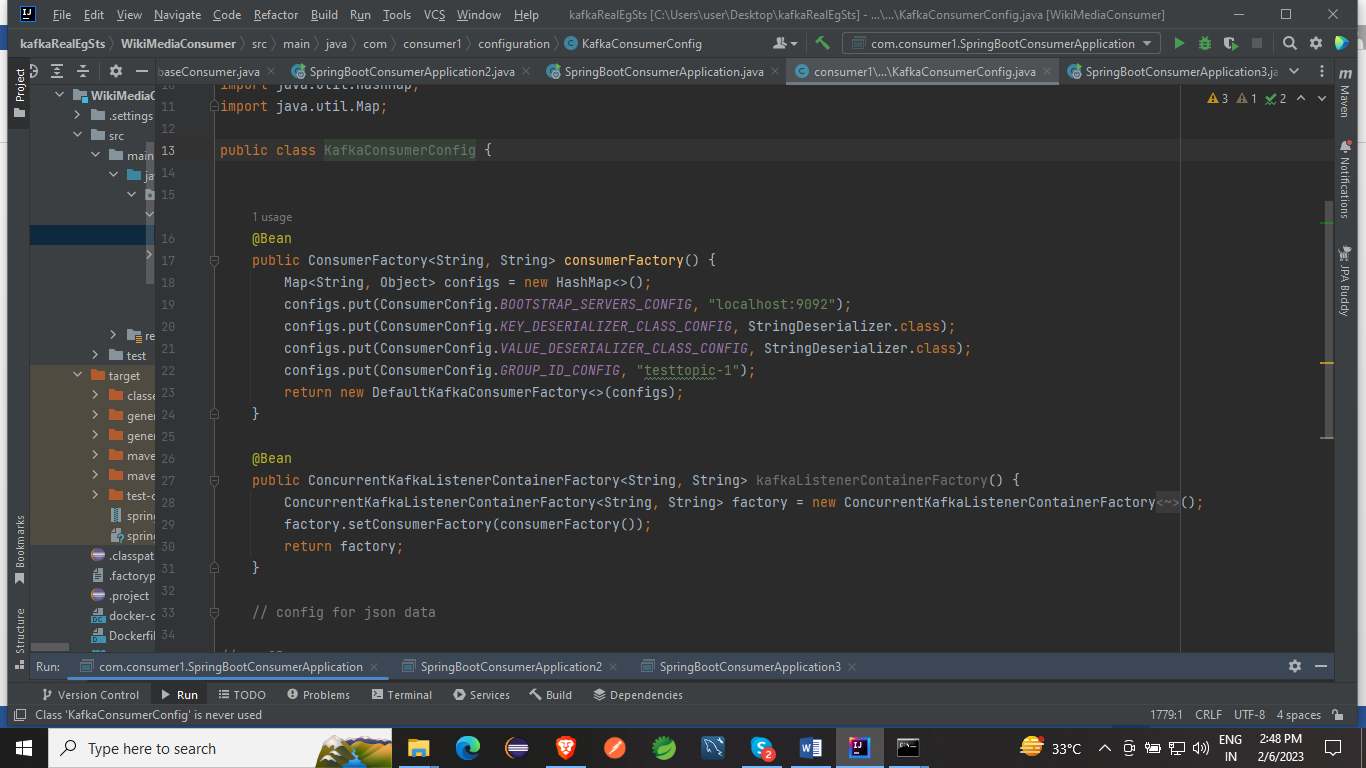


Implementation with Spring boot:

Now with Spring boot:



Configuration:



Service Class for Producer and Consumer

