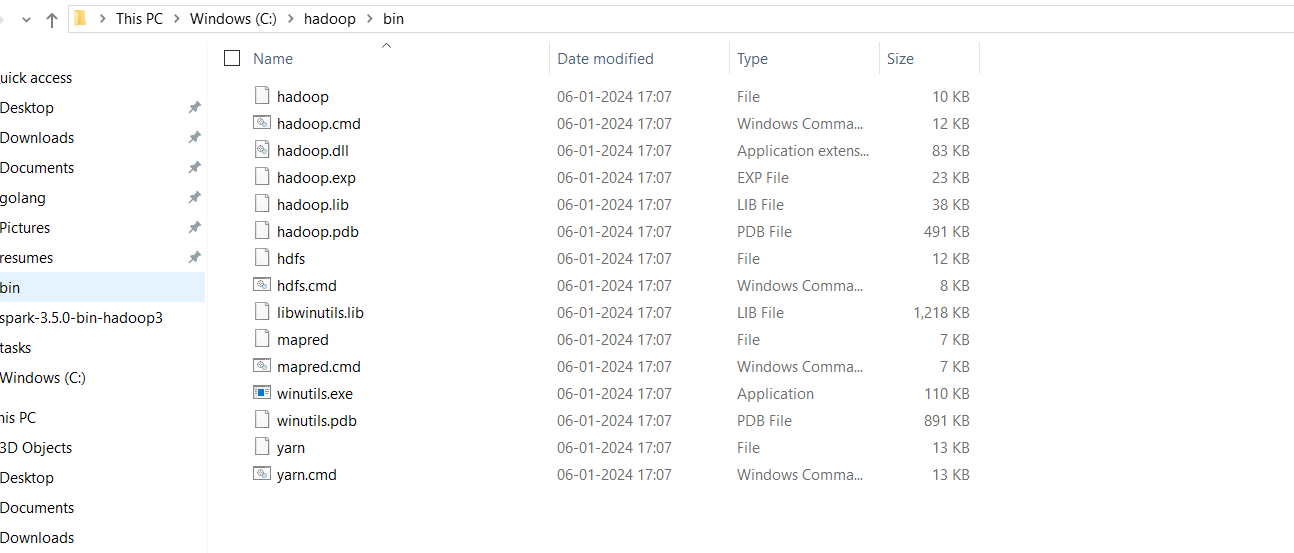
Setup kafka in local

Pre requisites

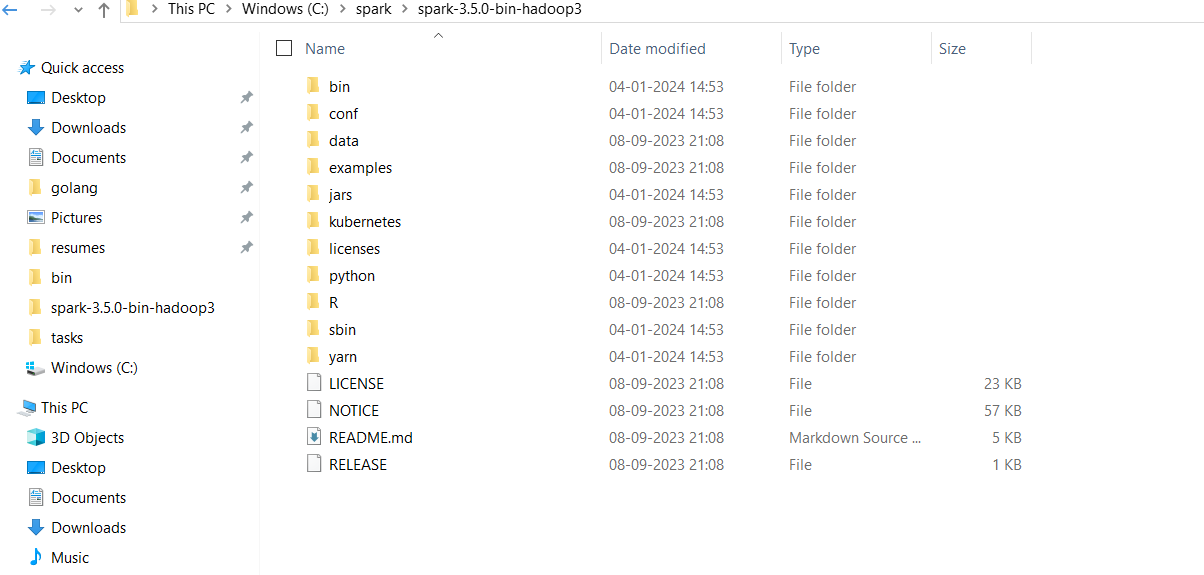
Download hadoop

https://github.com/cdarlint/winutils/tree/master/hadoop-3.3.5



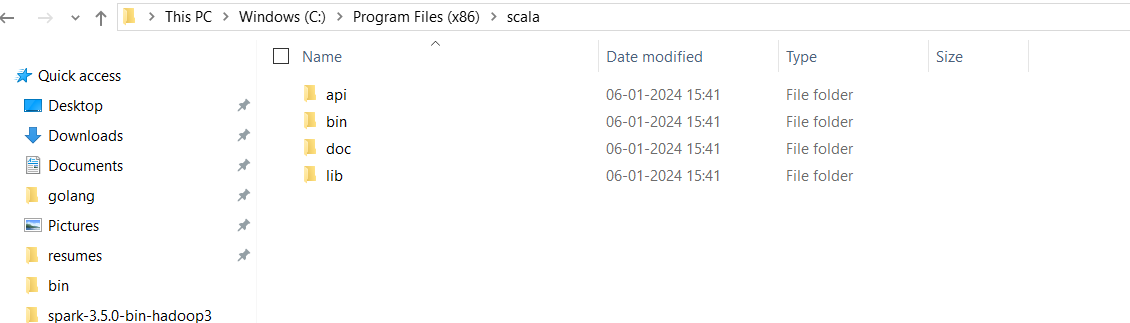
Download spark

https://spark.apache.org/downloads.html



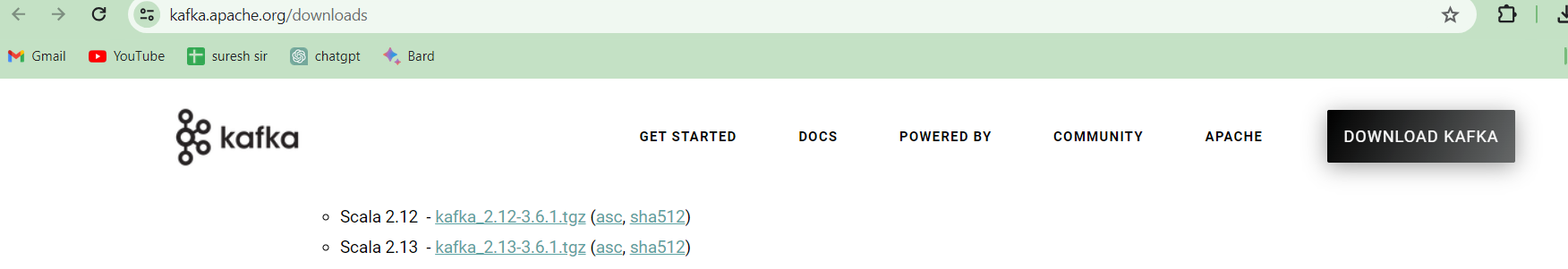
Download scala

<https://kafka.apache.org/downloads>

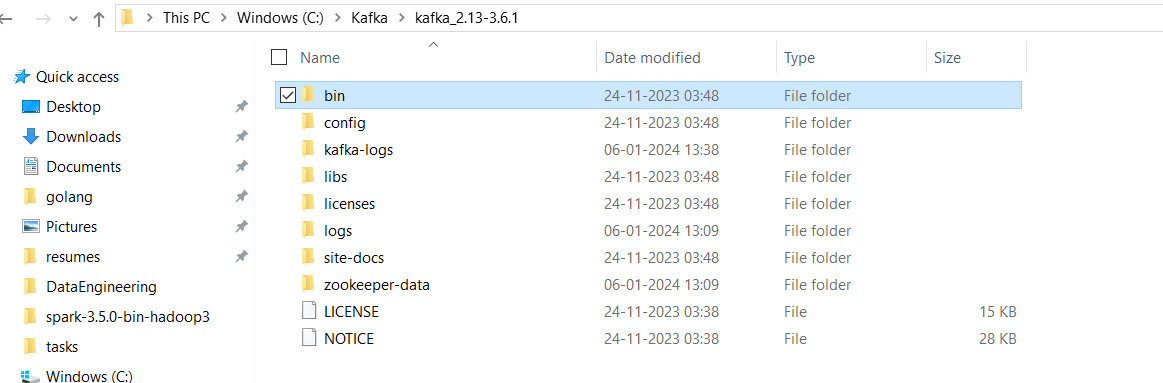


Download Kafka

https://kafka.apache.org/downloads



Extract the folder to C:\Kafka

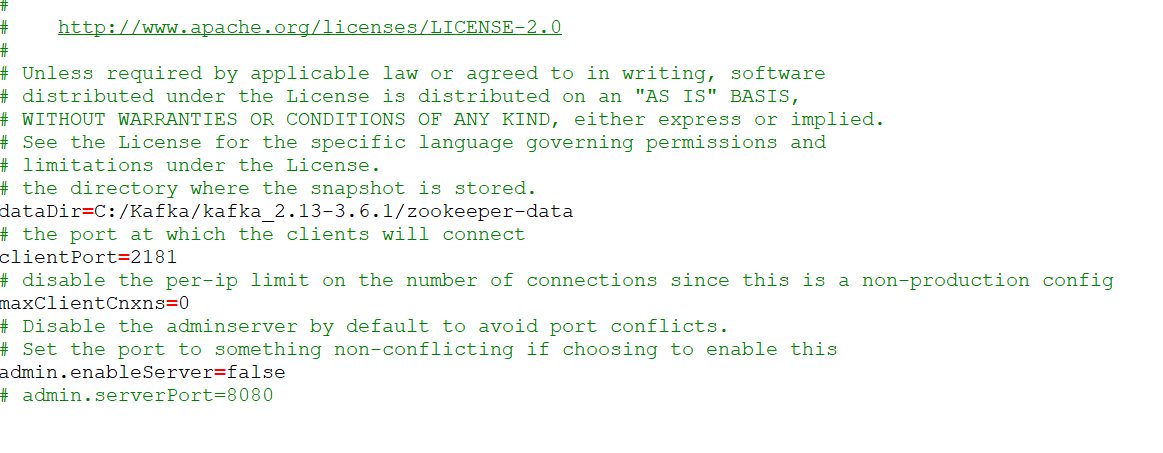


Goto C:\Kafka\kafka\_2.13-3.6.1\config

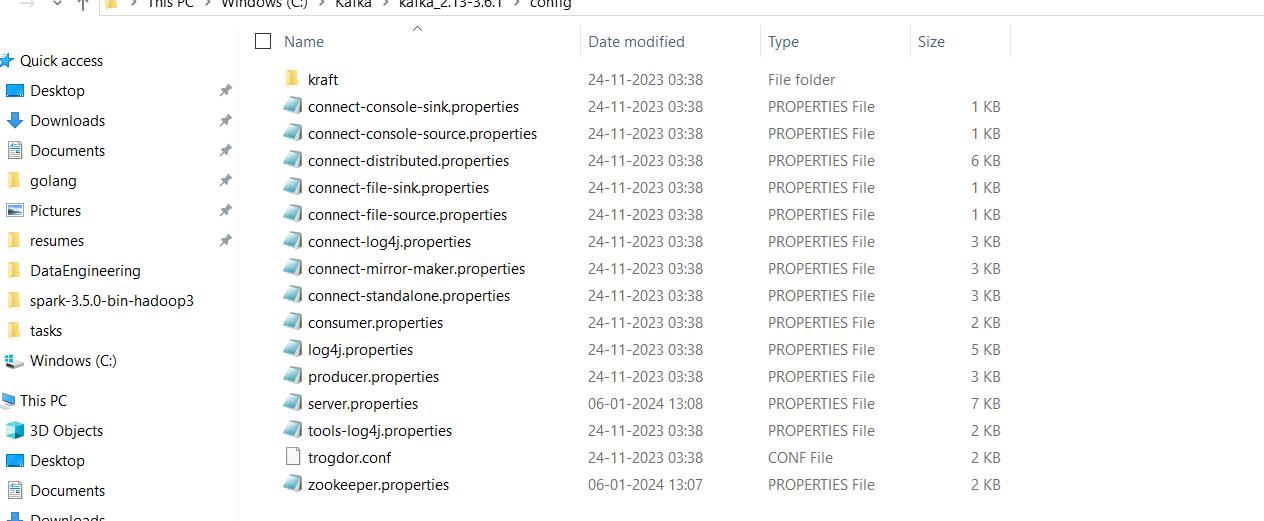
Inside config you will find zookeeper.properties file

Append you folder with zookeeper-data to dataDir

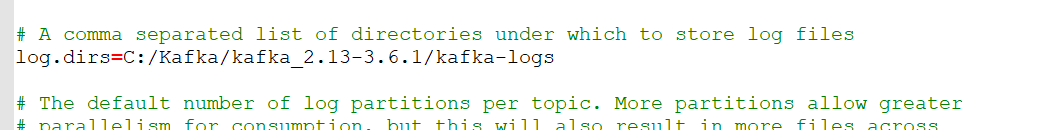
dataDir=C:/Kafka/kafka\_2.13-3.6.1/zookeeper-data



Then again in same config folder there will be a server.properties file



Open it add your logs directory



Now run you zookeeper   
open your command line and run the command

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

And again run kafka server by using below command

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

**So** Your kafka and zookeeper both are running now create kafka topic

**C:\Kafka\kafka\_2.13-3.6.1\bin\windows\kafka-topics.bat --create --topic TutorialTopic3 --bootstrap-server localhost:9092 --partitions 3 --replication-factor 1**

Now if you want to send message to that kafka topic throw command prompt then

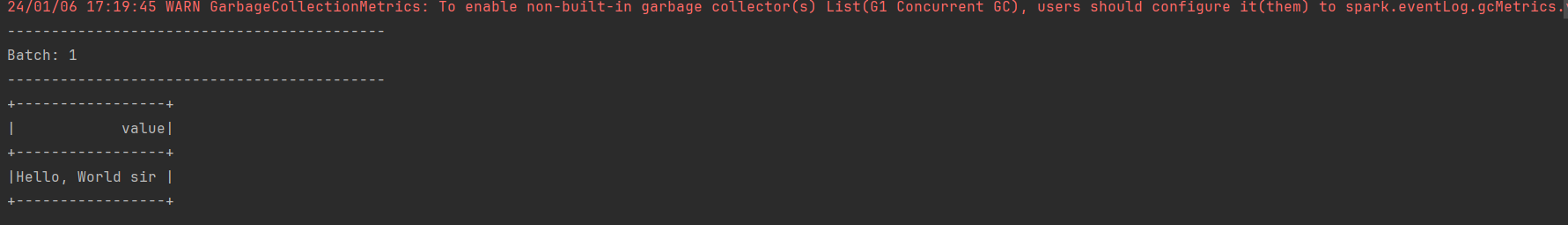
echo Hello, World | C:\Kafka\kafka\_2.13-3.6.1\bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic TutorialTopic3

---------------------

Lets create consumer streaming applications

from pyspark.sql import SparkSession  
spark = SparkSession.builder.config("spark.jars", "C:\\Users\\sudhe\\Downloads\\spark-sql-kafka-0-10\_2.12-3.5.0.jar,C:\\Users\\sudhe\\Downloads\\kafka-clients-3.6.1.jar,C:\\Users\\sudhe\\Downloads\\spark-streaming-kafka-0-10-assembly\_2.12-3.5.0.jar,C:\\Users\\sudhe\\Downloads\\commons-pool2-2.12.0.jar") \  
 .master("local").appName("KafkaStreamingToPostgreSQL").getOrCreate()  
kafka\_params = {  
 "kafka.bootstrap.servers": "localhost:9092",  
 "subscribe": "TutorialTopic3", # Replace with your actual topic name  
 "startingOffsets": "latest" # Adjust if you want to start from the latest offset  
}  
kafka\_df = (spark.readStream  
 .format("kafka")  
 .options(\*\*kafka\_params)  
 .load()  
)  
# Convert the value column from Kafka into a string  
string\_df = kafka\_df.selectExpr("CAST(value AS STRING) AS value")  
# Display the messages to the console  
query = string\_df.writeStream \  
 .outputMode("append") \  
 .format("console") \  
 .start()  
  
# Await termination to keep the streaming application running  
query.awaitTermination()

This will load data to console



Possible error and fixes:

Add the respective jars if you encounter the respective errors

java.lang.NoClassDefFoundError: org/apache/spark/kafka010/KafkaConfigUpdater

https://repo1.maven.org/maven2/org/apache/spark/spark-streaming-kafka-0-10-assembly\_2.12/3.5.0/spark-streaming-kafka-0-10-assembly\_2.12-3.5.0.jar

java.lang.NoClassDefFoundError: org/apache/commons/pool2/impl/GenericKeyedObjectPoolConfig

<https://repo1.maven.org/maven2/org/apache/commons/commons-pool2/2.12.0/commons-pool2-2.12.0.jar>

Check hadoop.dll file is present in your

java.lang.UnsatisfiedLinkError: 'boolean org.apache.hadoop.io.nativeio.NativeIO$Windows.access0(java.lang.String, int)'