**Cài đặt Kafka trên Ubuntu Server**

|  |
| --- |
| cd /home/hadoopnhutvinh  wget https://downloads.apache.org/kafka/3.7.2/kafka\_2.13-3.7.2.tgz  tar -xzf kafka\_2.13-3.7.2.tgz  mv kafka\_2.13-3.7.2 kafka\_folder |

**Loại bỏ file xung đột: (nếu máy bạn đã cài hive trước đó)**

|  |
| --- |
| mv /home/hadoopnhutvinh/hive/lib/kafka-clients-2.5.0.jar /home/hadoopnhutvinh/hive/lib/kafka-clients-2.5.0.jar.bak  mv /home/hadoopnhutvinh/hive/lib/zookeeper-3.8.3.jar /home/hadoopnhutvinh/hive/lib/zookeeper-3.8.3.jar.bak  mv /home/hadoopnhutvinh/hive/lib/zookeeper-jute-3.8.3.jar /home/hadoopnhutvinh/hive/lib/zookeeper-jute-3.8.3.jar.bak |

**Chỉnh sửa: /home/hadoopnhutvinh/kafka\_folder/config/server.properties**

|  |
| --- |
| broker.id=1  listeners=PLAINTEXT://0.0.0.0:9092  advertised.listeners=PLAINTEXT://192.168.x.x:9092  zookeeper.connect=192.168.x.x:2181 |

**Cấu hình zookeeper:**

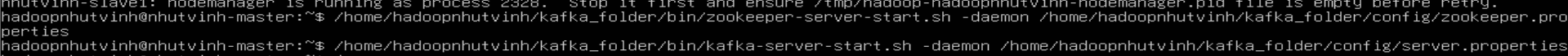
chỉnh sửa trong: /home/hadoopnhutvinh/kafka\_folder/config/zookeeper.properties

|  |
| --- |
| dataDir=/home/hadoopnhutvinh/zookeeper/data  clientPort=2181  maxClientCnxns=0 |

**Khởi động Zookeeper và Kafka**

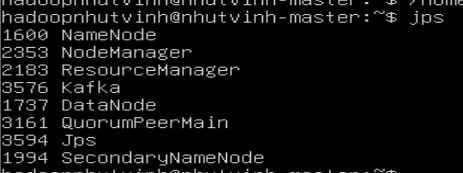
/home/hadoopnhutvinh/kafka\_folder/bin/zookeeper-server-start.sh -daemon /home/hadoopnhutvinh/kafka\_folder/config/zookeeper.properties

/home/hadoopnhutvinh/kafka\_folder/bin/kafka-server-start.sh -daemon /home/hadoopnhutvinh/kafka\_folder/config/server.properties

****

**Kiểm tra xem kafka và zoopkeeper có đang chạy không**

jps

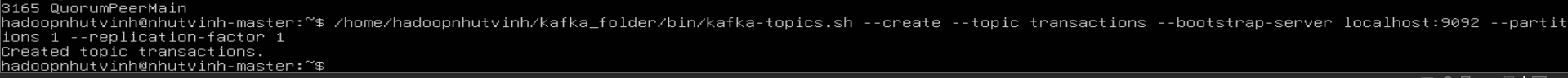
****



**🡪**Có dòng Kafka và QuorumPeerMain(Zookeeper) thì bạn đã thành công

**Tạo Kafka topic có tên là transaction**

|  |
| --- |
| /home/hadoopnhutvinh/kafka\_folder/bin/kafka-topics.sh --create --topic transactions --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1 |





**Producer đẩy dữ liệu vào Kafka**:

Tạo môi trường ảo:

|  |
| --- |
| python3 -m venv ~/myenv  source ~/myenv/bin/activate |

**Tạo dataset**

nano transactions.csv

id,amount,location

T001,1000,HoChiMinh

T002,2500,Hanoi

T003,1500,Danang

T004,2000,CanTho

T005,3000,Haiphong

T006,1800,QuangNinh

T007,2700,NhaTrang

T008,3500,Hue

T009,2200,HoChiMinh

T010,2600,Hanoi

A black screen with a white border

AI-generated content may be incorrect.

**Tạo producer**

nano producer.py

from kafka import KafkaProducer

import json

import time

import csv

producer = KafkaProducer(

bootstrap\_servers=['localhost:9092'],

value\_serializer=lambda v: json.dumps(v).encode('utf-8')

)

topic = "transactions"

with open("transactions.csv", "r") as f:

reader = csv.DictReader(f)

for row in reader:

data = {

"id": row["id"],

"amount": float(row["amount"]),

"location": row["location"]

}

producer.send(topic, value=data)

print(f"Sent: {data}")

time.sleep(1)

producer.flush()

producer.close()

**Tạo file spark streaming**

nano spark\_streaming.py

from pyspark.sql import SparkSession

from pyspark.sql.functions import from\_json, col, sum as \_sum

from pyspark.sql.types import StructType, StringType, DoubleType

spark = SparkSession.builder \

.appName("KafkaSparkStreaming") \

.master("local[\*]") \

.getOrCreate()

spark.sparkContext.setLogLevel("WARN")

schema = StructType() \

.add("id", StringType()) \

.add("amount", DoubleType()) \

.add("location", StringType())

df = spark.readStream \

.format("kafka") \

.option("kafka.bootstrap.servers", "localhost:9092") \

.option("subscribe", "transactions") \

.option("startingOffsets", "earliest") \

.load()

json\_df = df.selectExpr("CAST(value AS STRING) as json\_str")

data\_df = json\_df.select(from\_json(col("json\_str"), schema).alias("data")).select("data.\*")

agg\_df = data\_df.groupBy("location").agg(\_sum("amount").alias("total\_amount"))

query = agg\_df.writeStream \

.outputMode("complete") \

.format("console") \

.option("truncate", "false") \

.start()

query.awaitTermination()

**Kiểm tra topics**

cd kafka\_folder

cd bin

./kafka-topics.sh --list --bootstrap-server localhost:9092

A black screen with white text

AI-generated content may be incorrect.



**Tiến hành chạy file producer**

python3 producer.py

A screenshot of a computer program

AI-generated content may be incorrect.

**Kiểm tra kết quả**

/home/hadoopnhutvinh/kafka\_folder/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic transactions --from-beginning

A black screen with white text

AI-generated content may be incorrect.

**Chạy file Spark Streaming**

spark-submit --packages org.apache.spark:spark-sql-kafka-0-10\_2.12:3.5.3,org.apache.spark:spark-token-provider-kafka-0-10\_2.12:3.5.3 spark\_streaming.py

A black screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Hướng dẫn cách cài đặt thư viện spark-streaming-kafka, cách tạo KafkaUtils.createDirectStreamS**

**Chuẩn bị môi trường:**

* Java (OpenJDK 8 hoặc 11)
* Apache Spark (>=3.3.0) 🡪 Do đây là phiên bản cũ, nên bây giờ thư viện **spark-streaming-kafka** & **KafkaUtils.createDirectStream** không còn được sử dụng nữa **🡪** phải dùng **Structured Streaming** (spark-sql-kafka-0-10), vì DStreams connector không build cho bản mới.
* Apache Kafka (>=3.x)

Kiểm tra phiên bản:

java -version

spark-shell --version

kafka-topics.sh –version

**2. Tải Spark với Kafka connector**

Mặc định Spark không kèm thư viện Kafka, bạn cần **thêm package** khi chạy.

Ví dụ:

spark-shell --packages org.apache.spark:spark-streaming-kafka-0-10\_2.12:3.3.0

hoặc nếu bạn dùng PySpark:

pyspark --packages org.apache.spark:spark-streaming-kafka-0-10\_2.12:3.3.0

Lưu ý:

* 2.12 là phiên bản Scala mà Spark build. (Spark 3.x thường build với Scala 2.12).
* 3.3.0 là phiên bản Spark Streaming Kafka tương ứng với Spark 3.3.0.