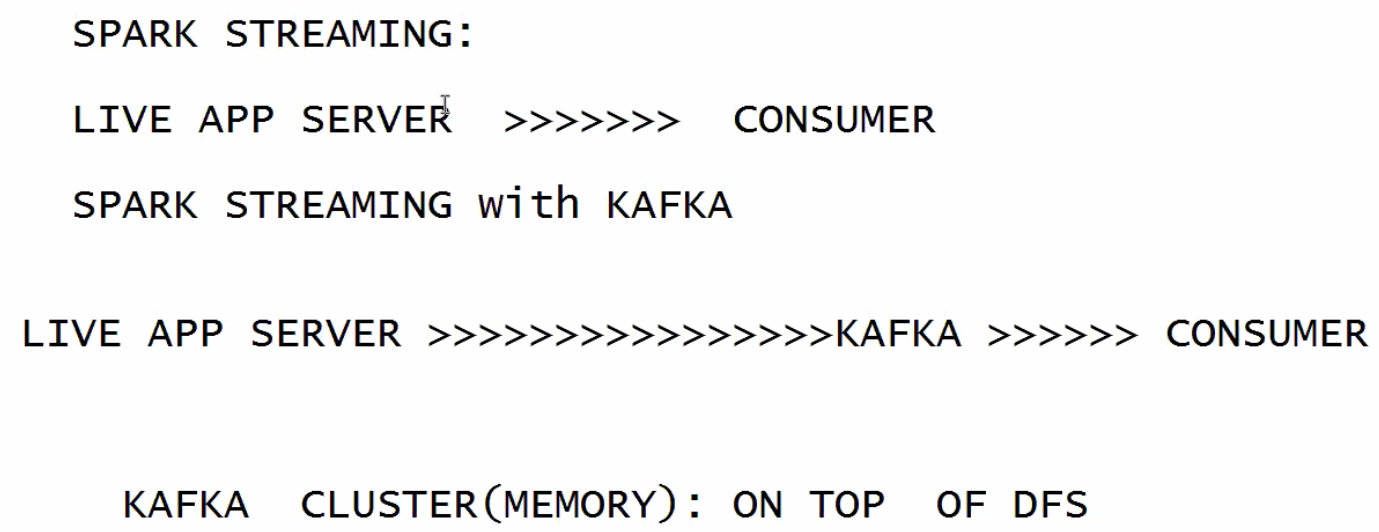
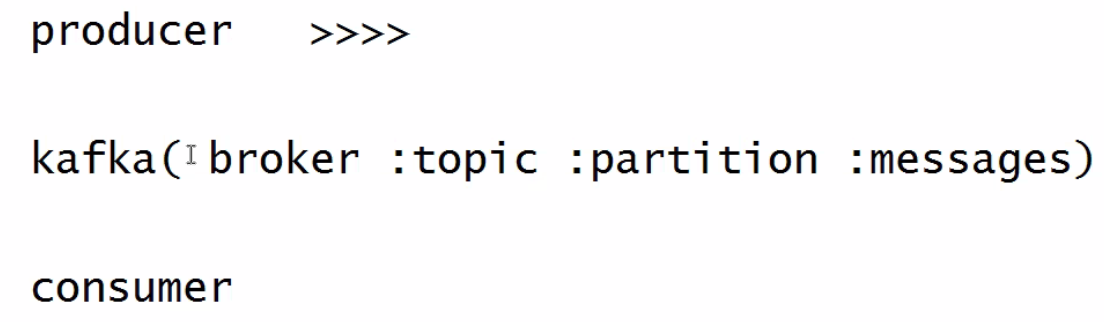
**22/11/2021**

KAFKA is a tool given by LinkedIn in 2011.



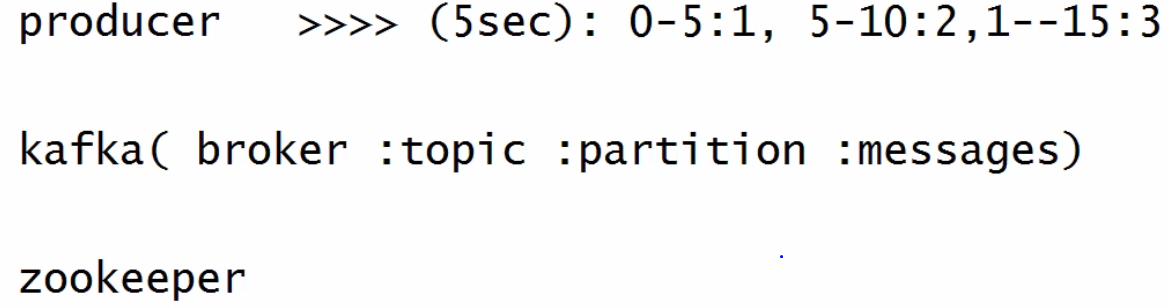
How KAFKA is managing data?

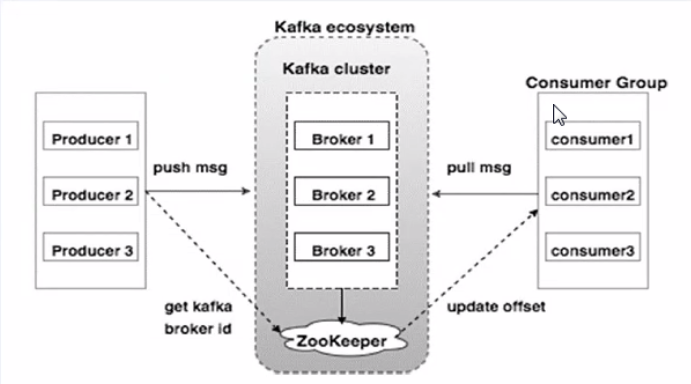
KAFKA connects with producer and consumer. KAFKA is a cluster.



Every partition has a data, defined as a set of messages. Batch interval time is messages.

All component names in big data are animal names.

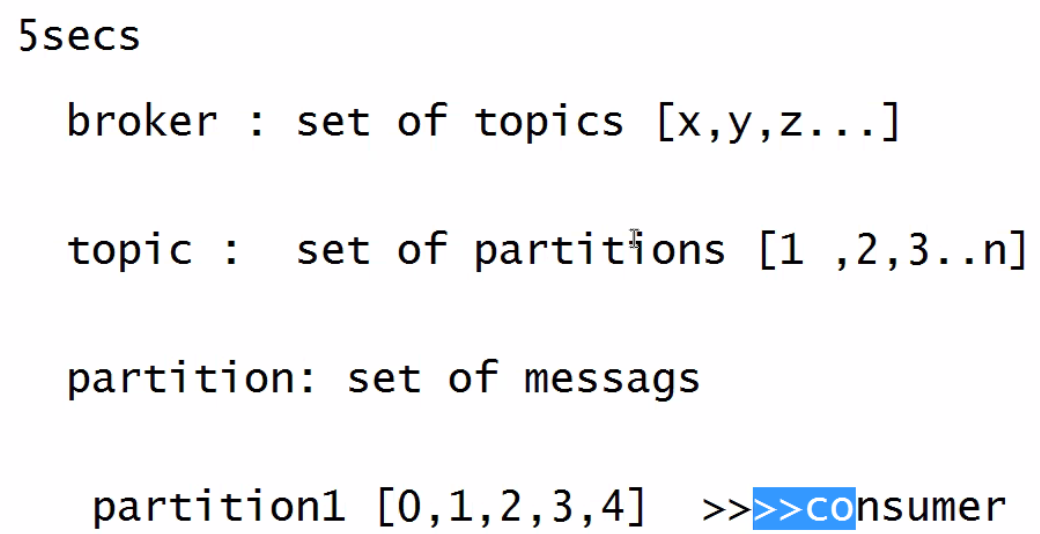




Producer-live application server from where we are fetching the data

Data is going to the broker. Broker is maintaining data in the topic. Broker is a set of topics. Many topics per broker-depends on the volume of data. Partitions inside the topic. We can create n no of partitions. Partition defined data as messages.

Consumer-processing server



How much duration of data is there if 5 sec per partition? - 20 seconds

Consumer acknowledges to zookeeper.

Kafka will handle multiple producers and multiple consumers. Kafka cluster should be ready before running producer and consumer.

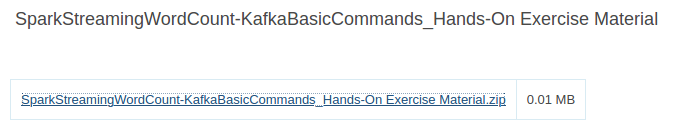
We are working on single node cluster so replication factor would be 1. We need to start zookeeper server. If zookeeper service is down, we can’t do streaming analysis. Zookeeper is managing all Hadoop components in streaming analysis.

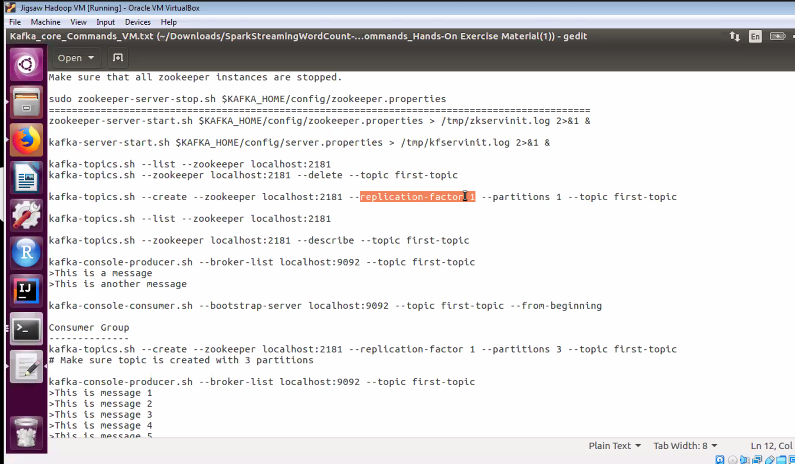
Kafka gives us default brokers and we use them to create topics directly with zoopkeeper connection.

**Hands-on**

Start Hadoop cluster

Run pyspark





sudo zookeeper-server-start.sh $KAFKA\_HOME/config/zookeeper.properties

Server is running with port no 2181.

kafka-server-start.sh $KAFKA\_HOME/config/server.properties

List-to list out the topics in kafka

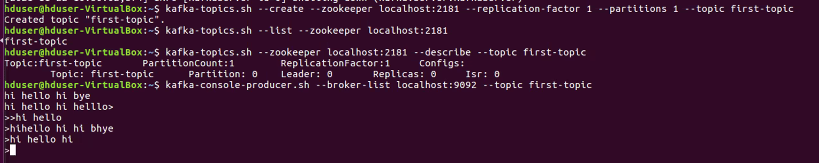
kafka-topics.sh --list --zookeeper localhost:2181

Delete-to delete if there are any topics, not compulsory to execute always

kafka-topics.sh --zookeeper localhost:2181 --delete --topic first-topic

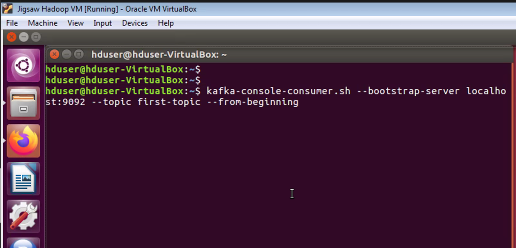
kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic first-topic

Default broker number is 0



Use default port no to pass msg to topic

Open new terminal



Kafka consumes data from topics

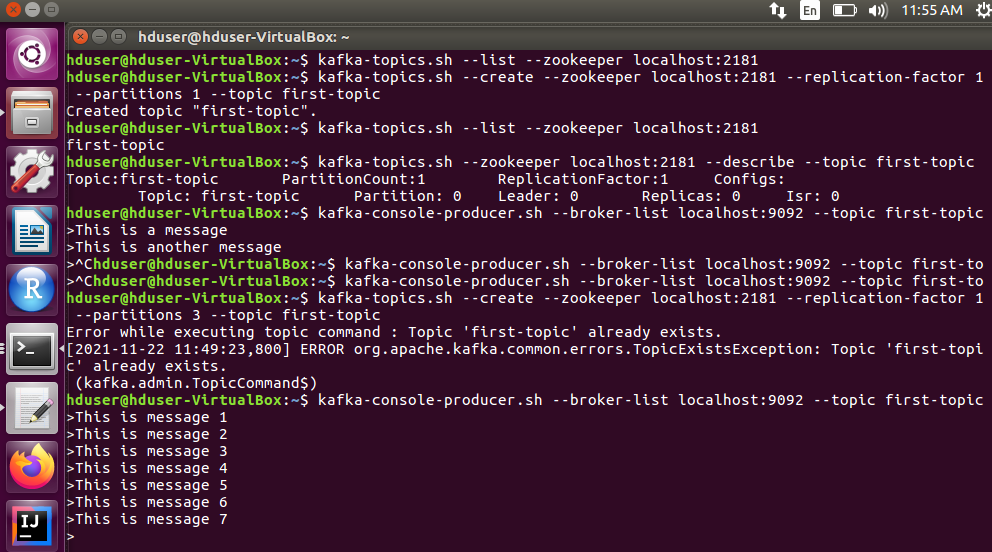
Topic is created with 3 partitions here

Run the producer and pass the data

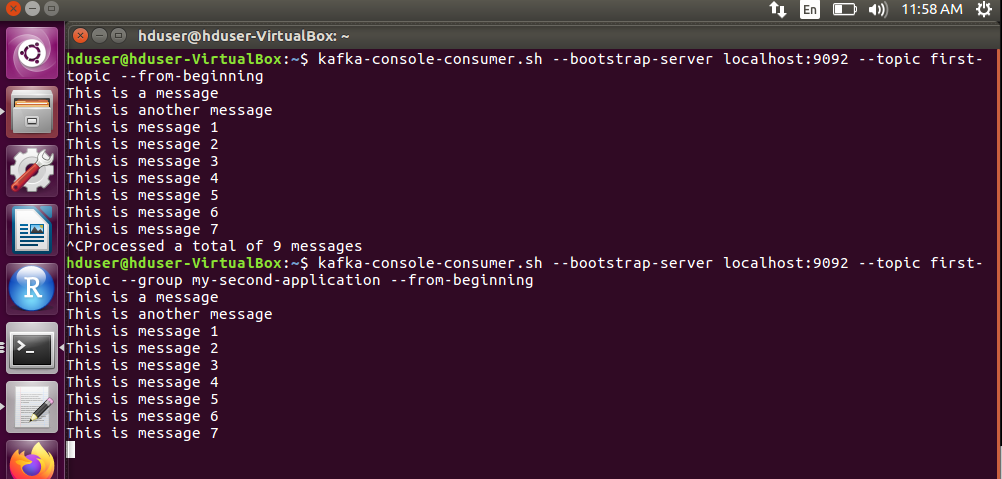
Run 3-4 consumers.

Consume all partition msgs from the beginning

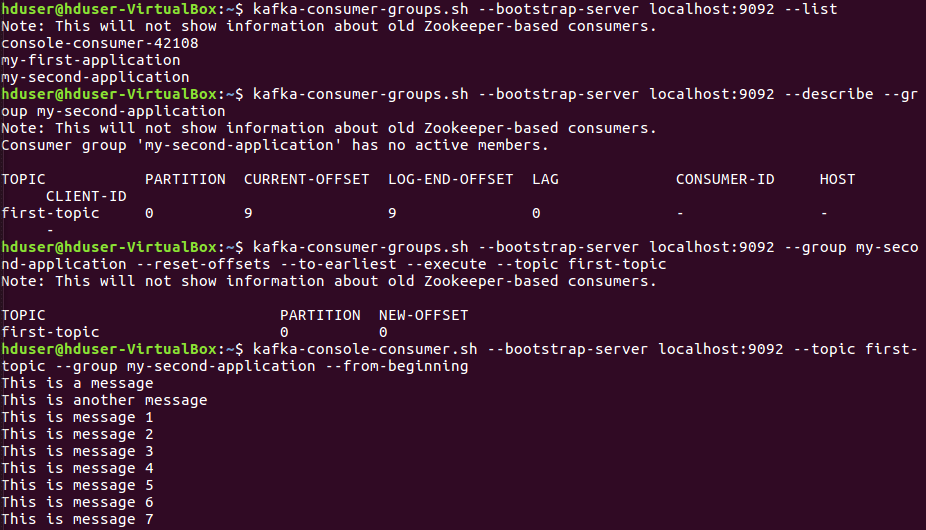
Producer Terminal-



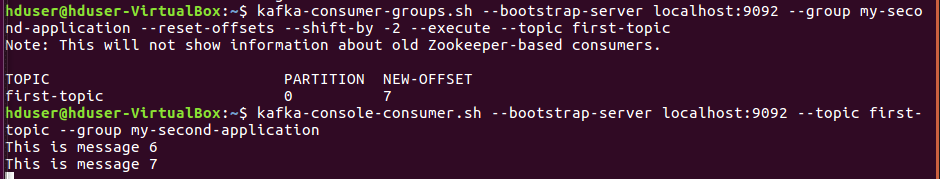
Consumer Terminal-



We are not giving the --from-beginning it will read from the last offset



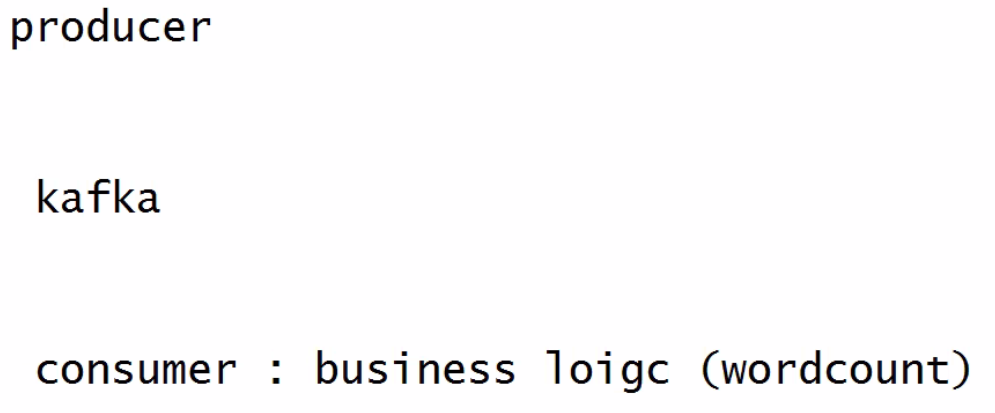
Reads from the last offset-



Here, I haven’t closed the consumers so offset is 7.

With offset =0 no messages will be skipped

**Problem**

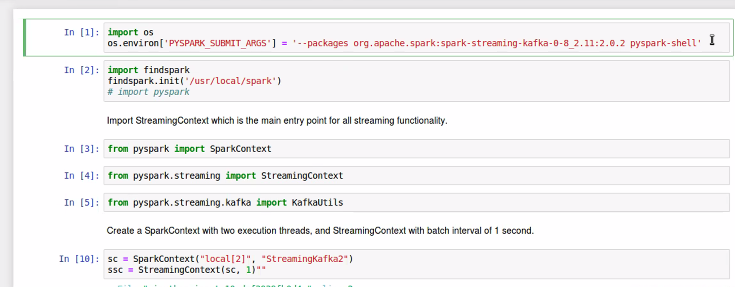


Open Jupyter notebook

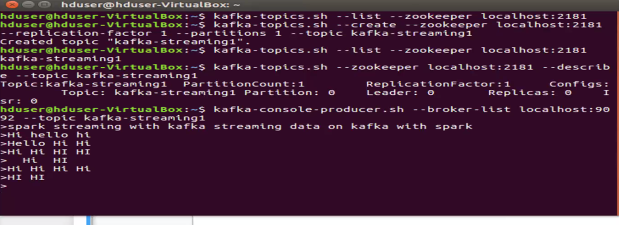
Run Hadoop cluster-start-all.sh

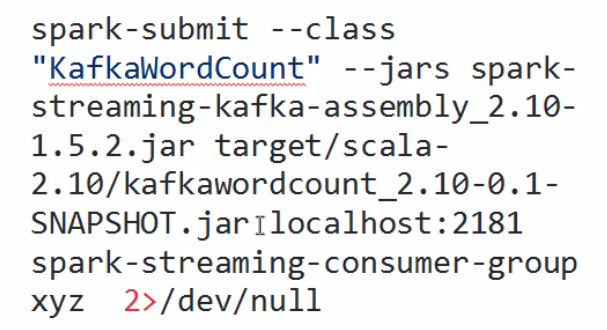
Upload ipynb file to jupyter n/b





Start the zookeeper and kafka server





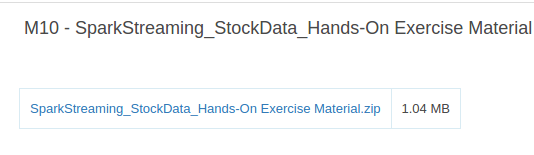
JAR created for execution of code-name given here

2>/dev/null=represents the listener/receiver

21781-zookeeper port no

Error showing while executing .py file in console(hyphen is to be corrected)

**Problem-**



Static-historical data, exists

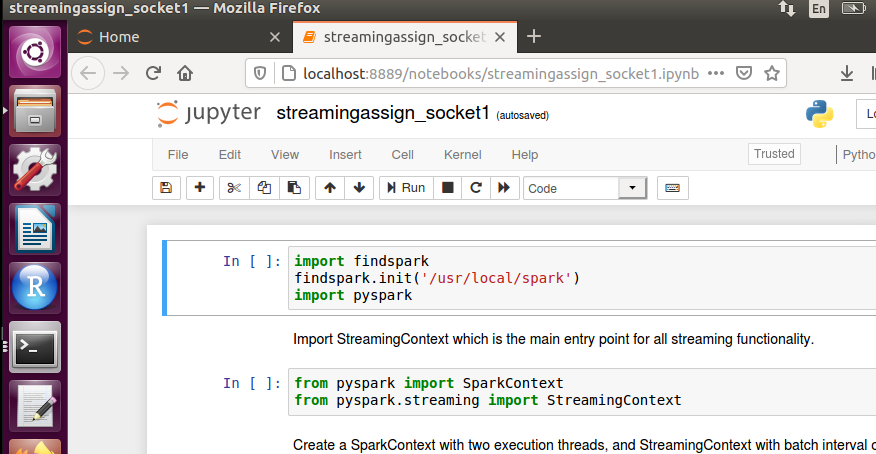
Compare previous stock market price with current, stock symbol wise.

Apply cache function-performance increases

Pass stock symbol and stock price, delimiter comma

Key value is taken as our input is RDD, so divide the data as key values.

2 input files



chmod +x stock\_details\_socket.sh –(not given in pdf)-makes the program executable – change mode (chmod) executable(+x)

./stock\_details\_socket.sh stock\_stream\_data.csv

./stock\_details\_socket.sh stock\_stream\_data.csv | nc –lk 9999 –(not executing)