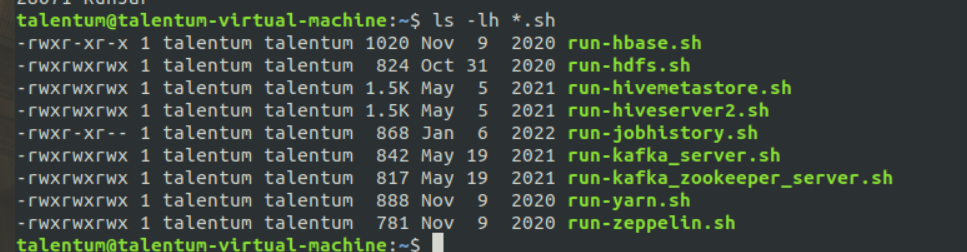
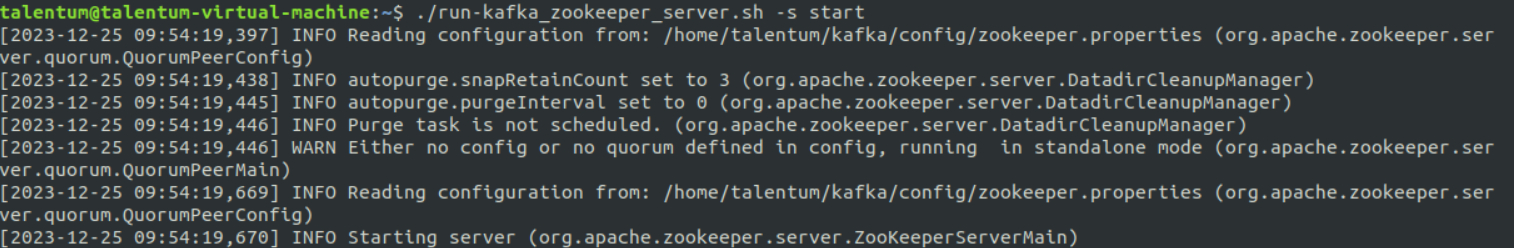
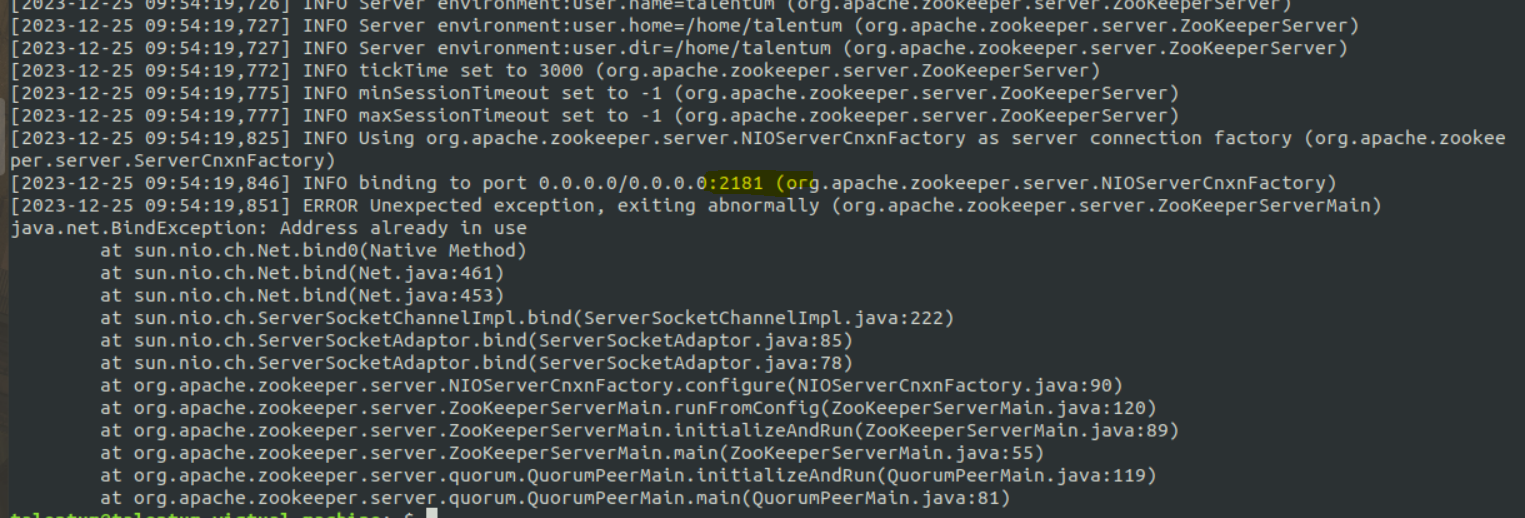
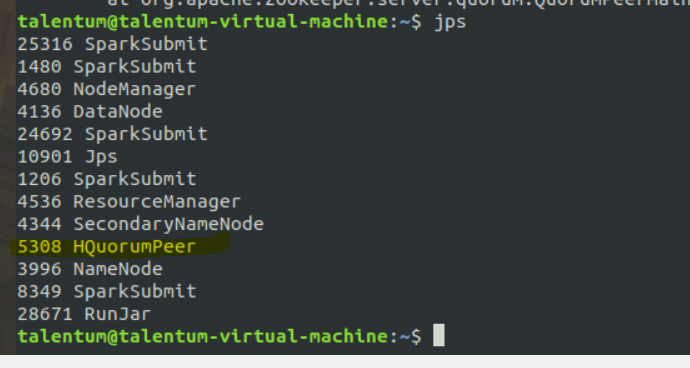
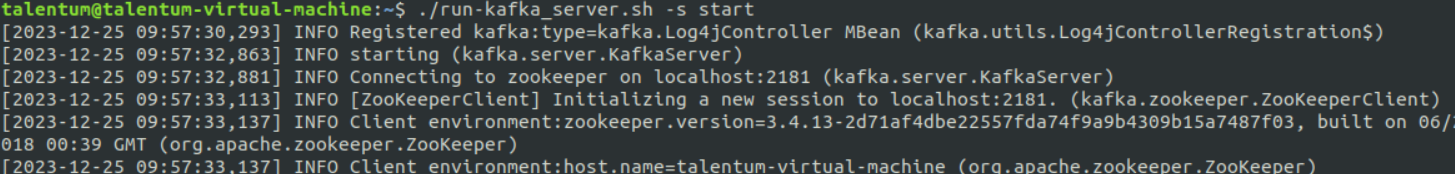
****

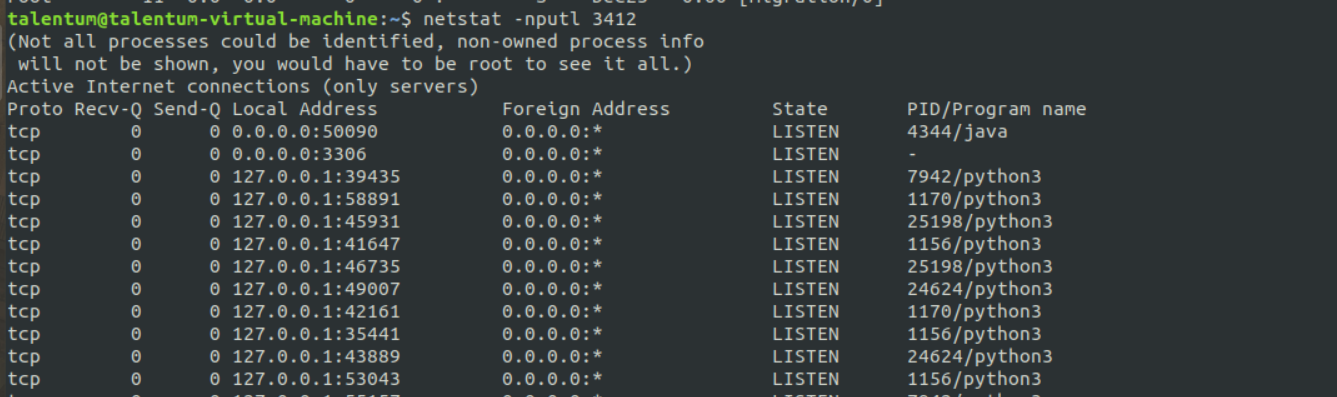
****

****

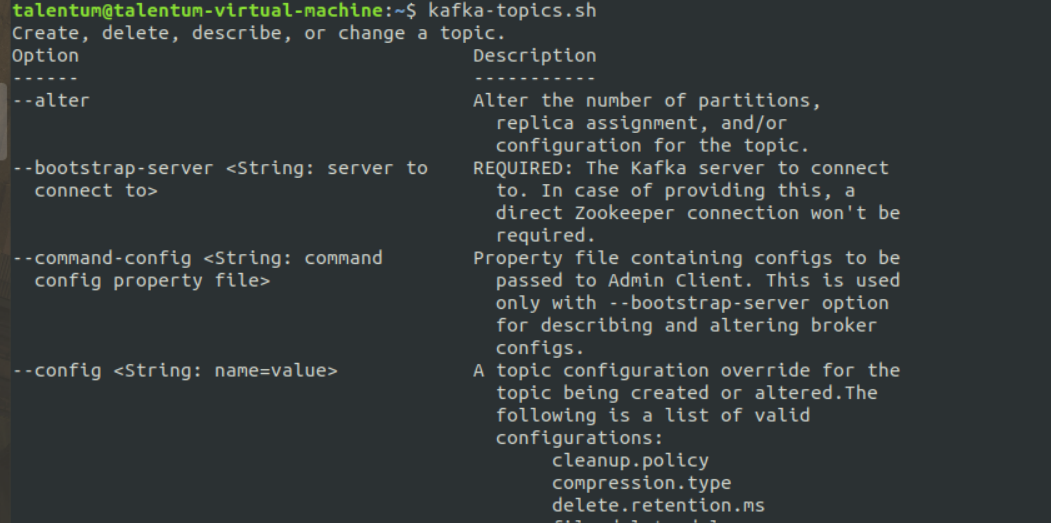
****

**Kill -9 (name of the process want to stop)**

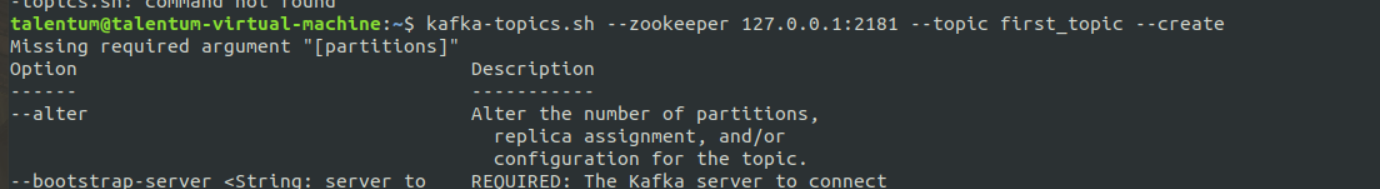
****

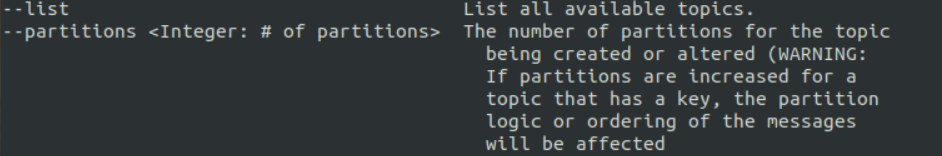
****

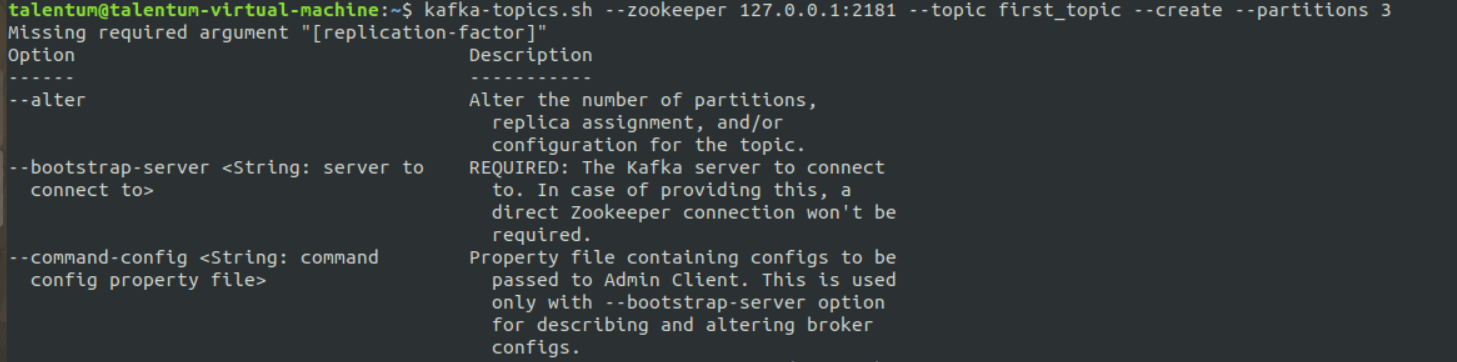
****

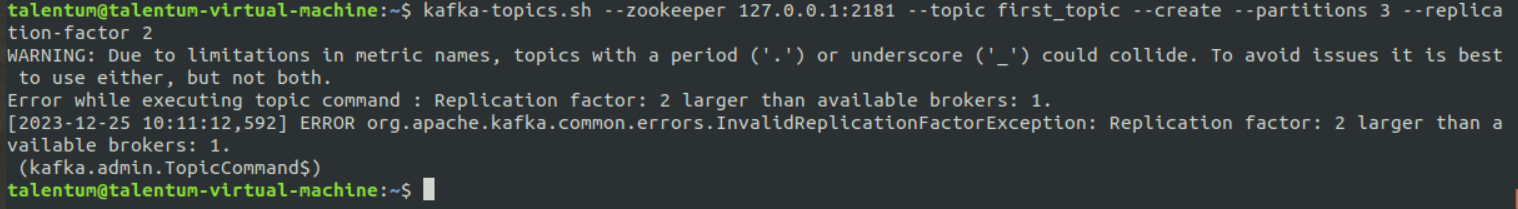
****

**Create topic**

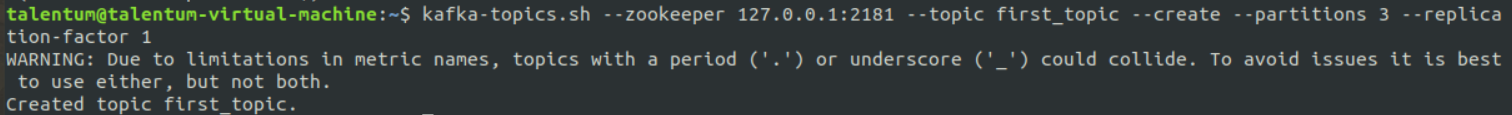
****

****

****

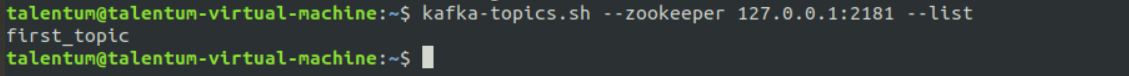
****

**Replication factor cannot be 2 in this example so make it 1 (stand-alone machine)**

****

**To list topics**

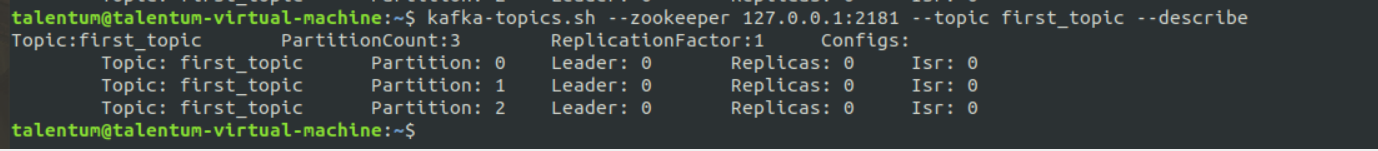
****

****

**To delete topic (don’t do this)**

**kafka-topics.sh --zookeeper 127.0.0.1:2181 - -topic first\_topic - -delete**

**Describe topic**



**Leader : 0 ( 0 is broker id )**

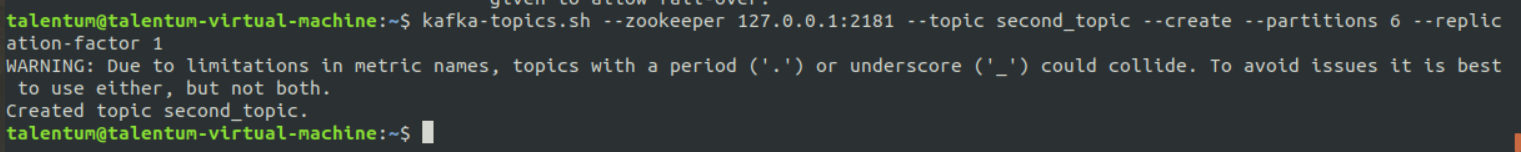
**Isr partition**

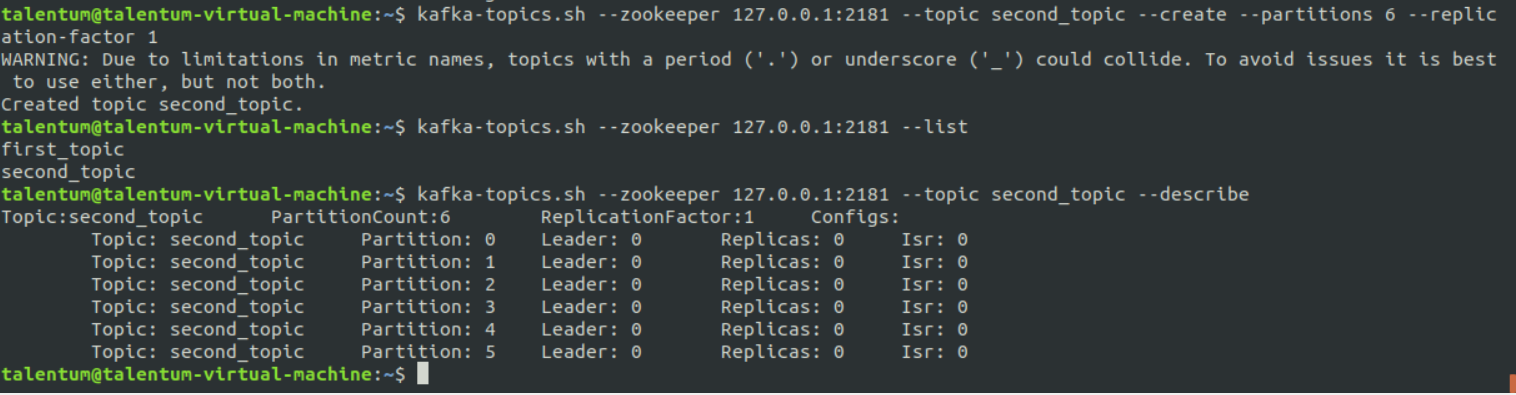
**If we made partitions 4 so there will be 1 leader partition and 3 Isr partitions.**

**If 1 Isr got crashed so, kafka will take care of it.**

**Exercise**

Create a new topic, having name as second\_topic, having six partitions, and then the replication factor, which is going to be one.

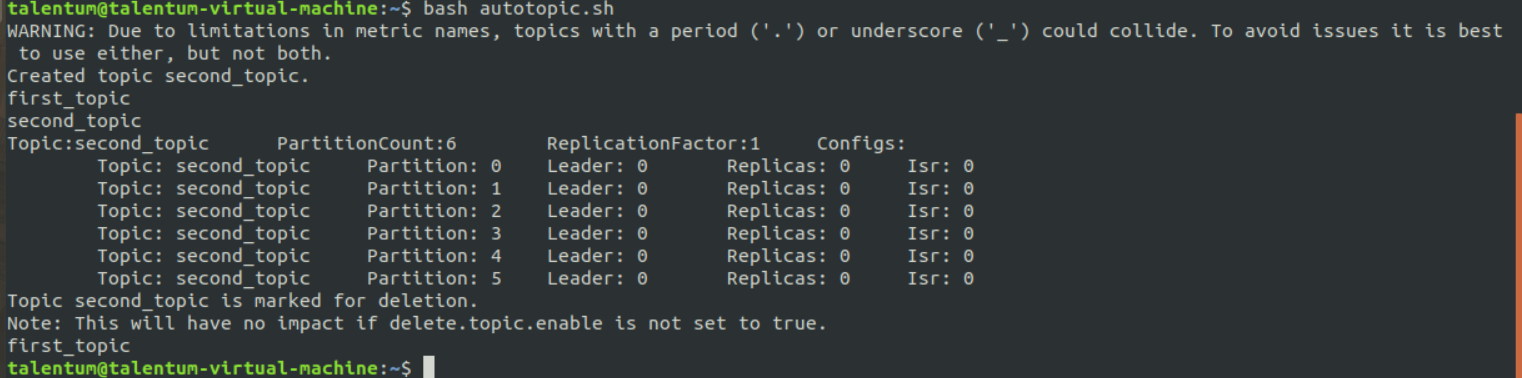






**Automate same**

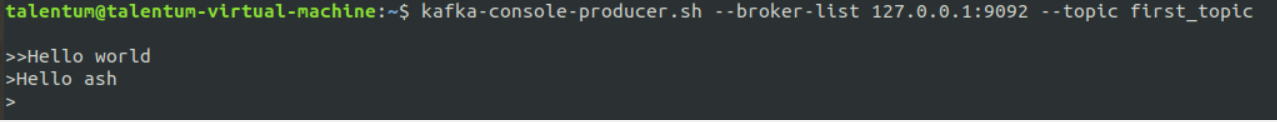




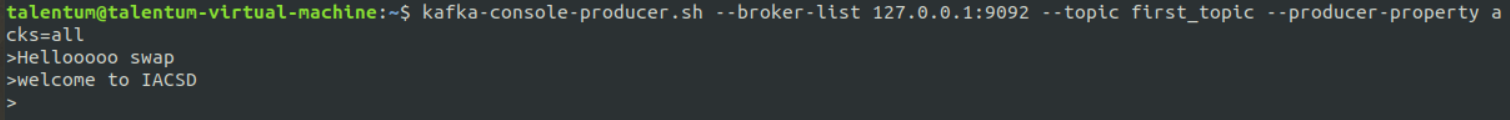
**On a single machine (Host + port = socket) as we cannot run many kafka broker on single machine.**

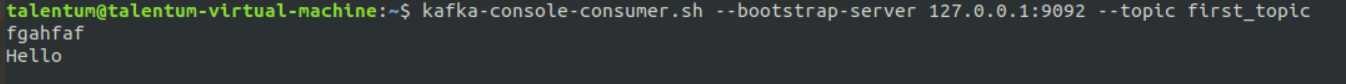
**In Kubernetis we can run.**

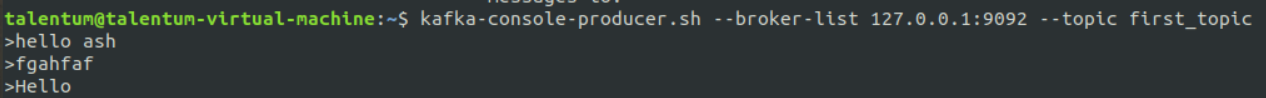
**Producers**

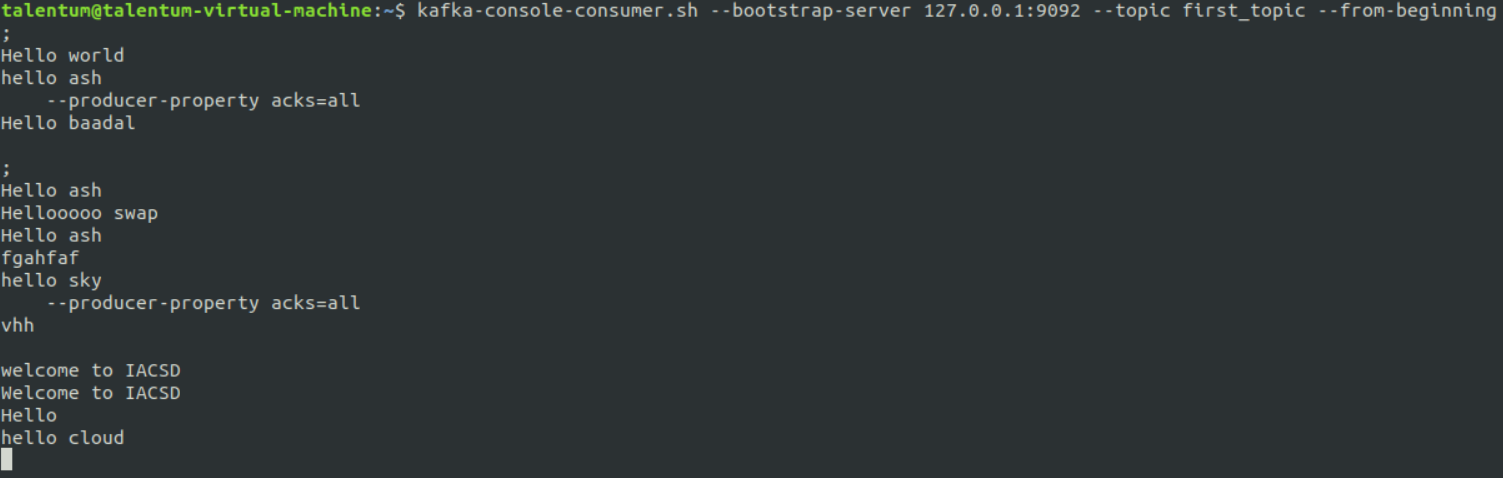


To end producer : (ctr+c)



**Consumers**





<https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html#structured-streaming-programming-guide>

**Structured Streaming Programming Guide**

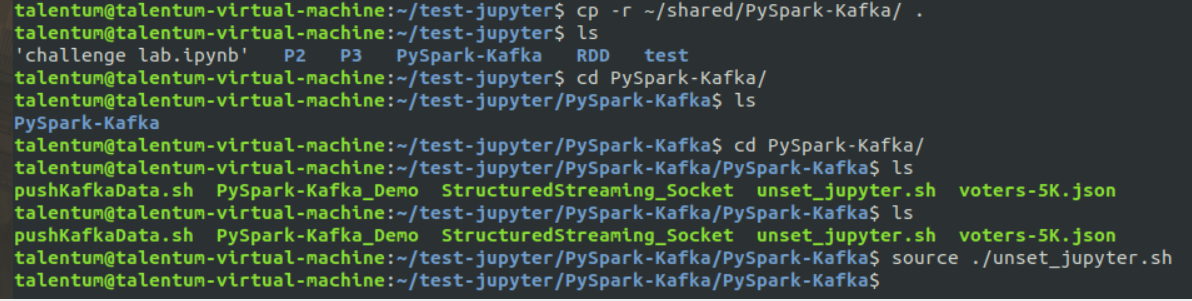
**18] Spark Structured Streaming**

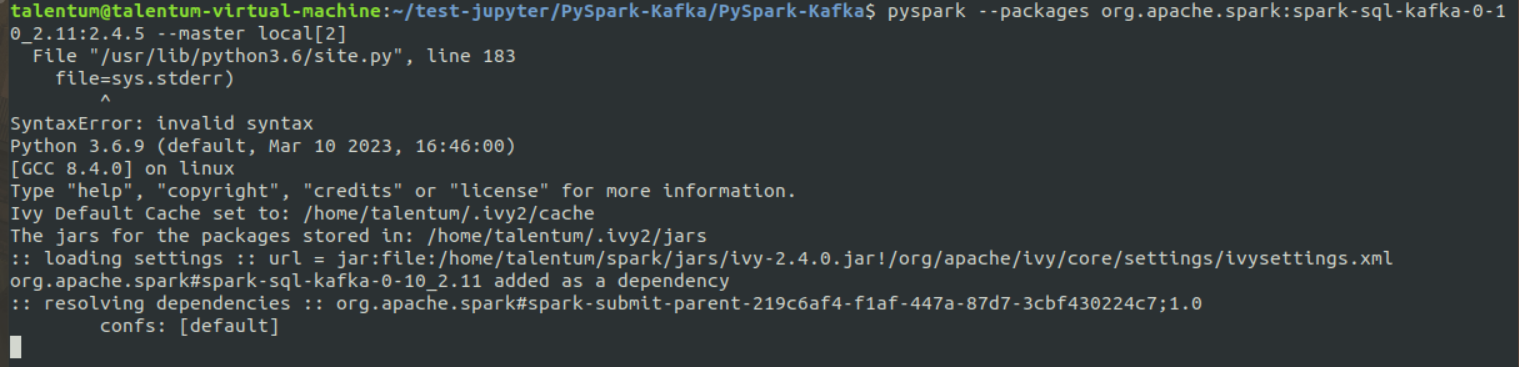
**Download the attached file in STAGING\_AREA**

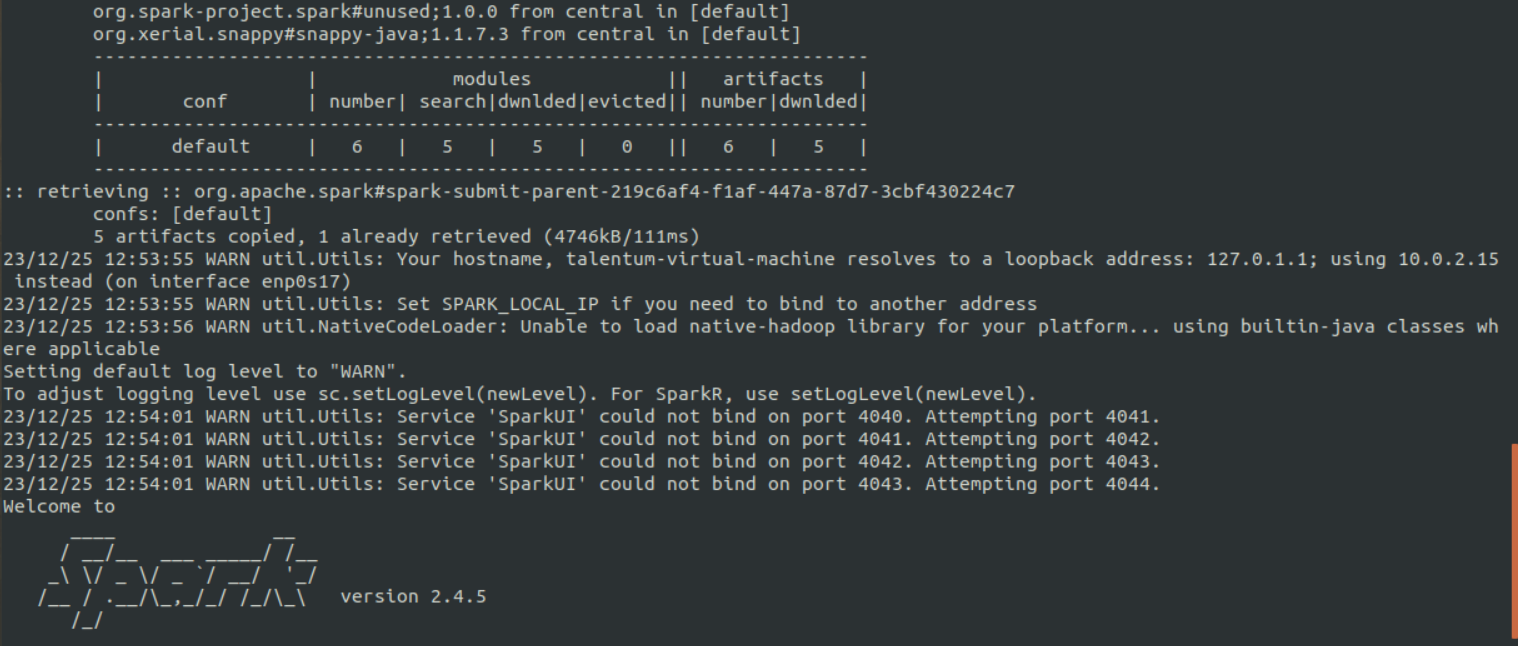
**Pull the file in LABS\_AREA/test-jupyter/pyspark-kafka**

**Code/Dataset**

**data/PySpark-Kafka.zip**

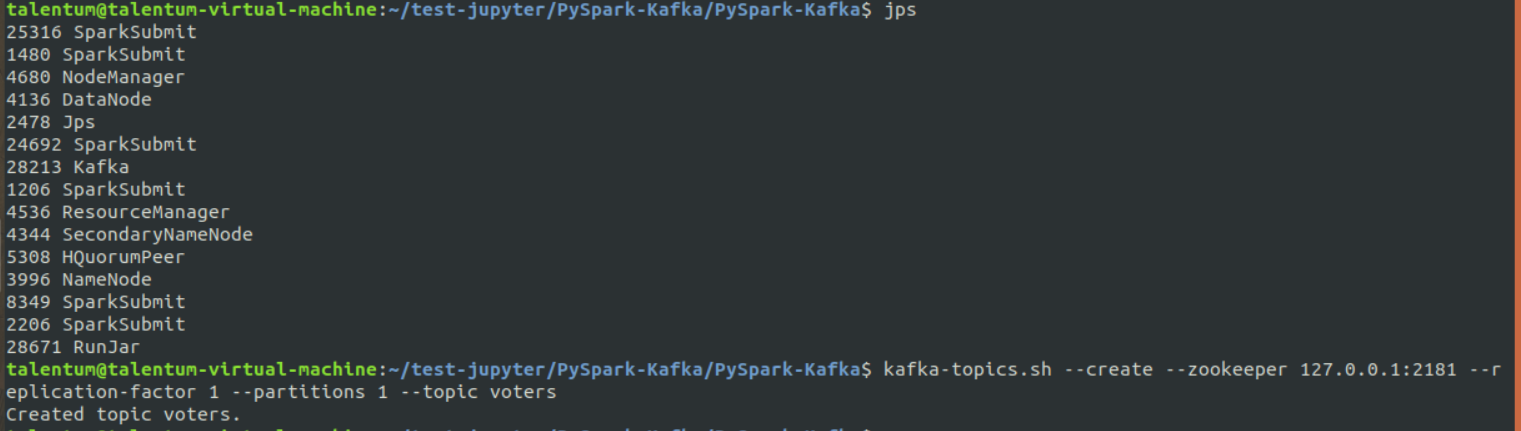


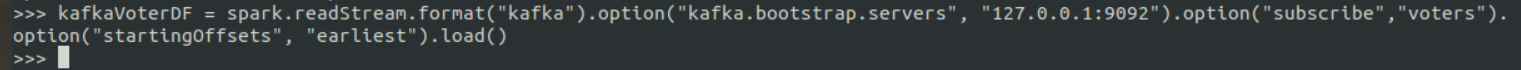


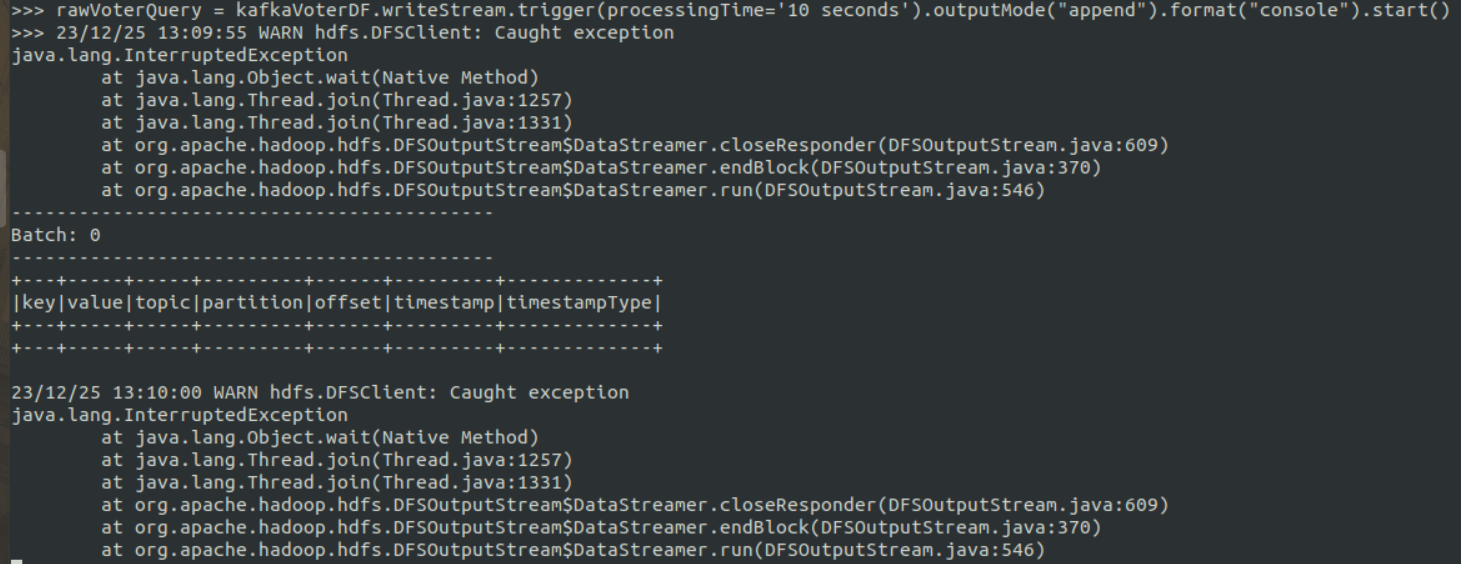


**Producer – linux bash program, Consumer –**

**REPL**

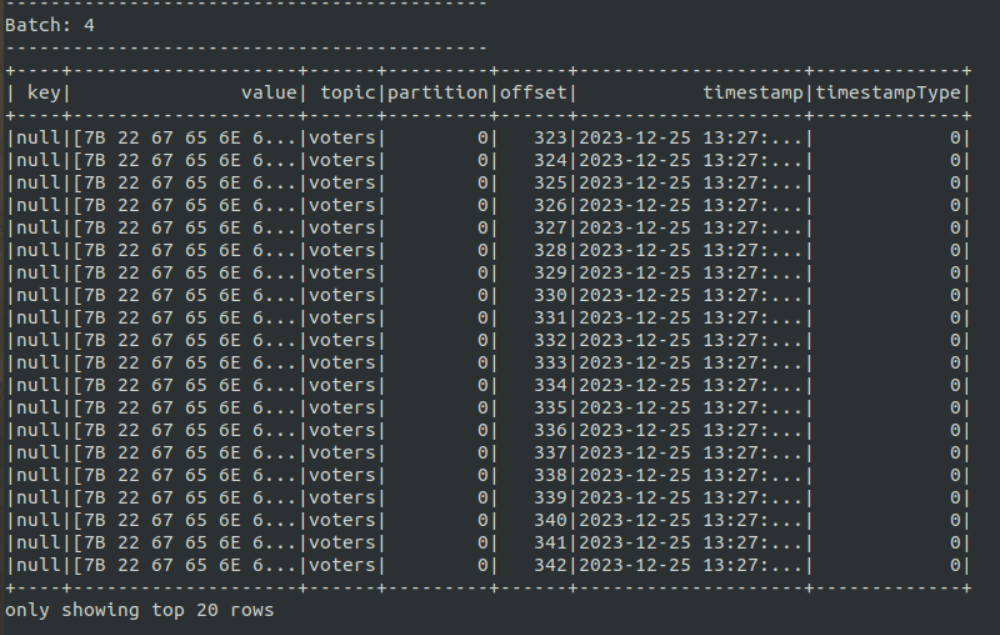
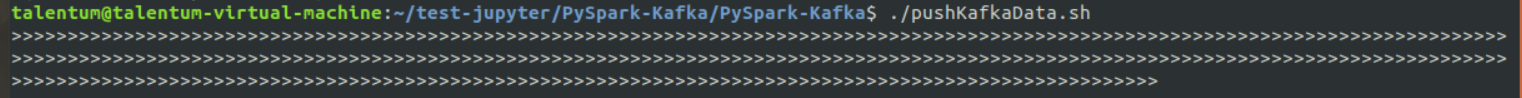


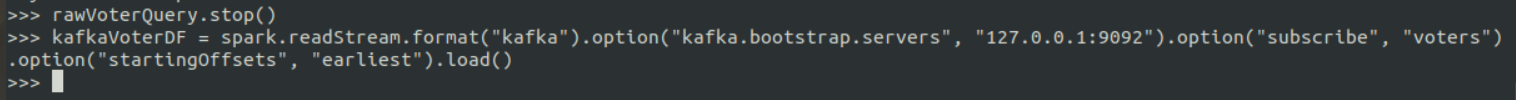


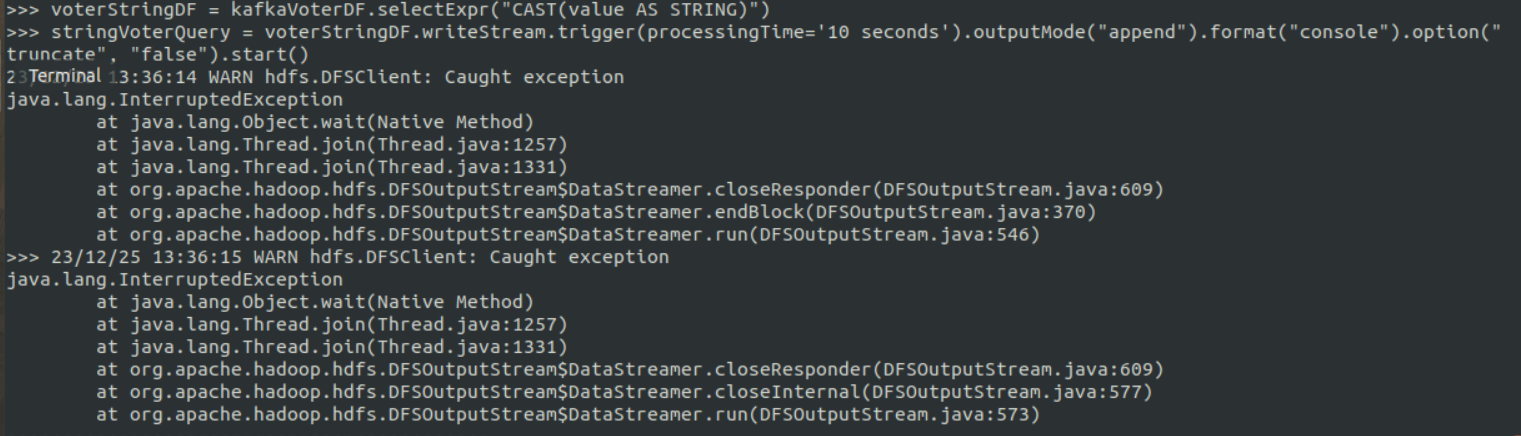


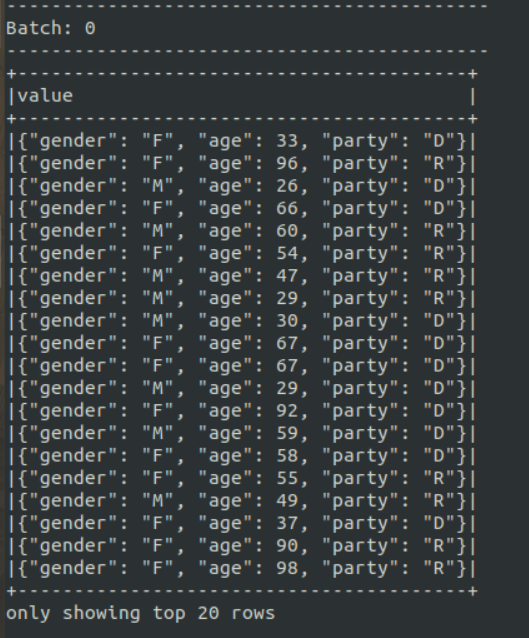
**Consumer is running and now the producer will run**

**After every 10 second message is getting read (Batch 0--------12)**

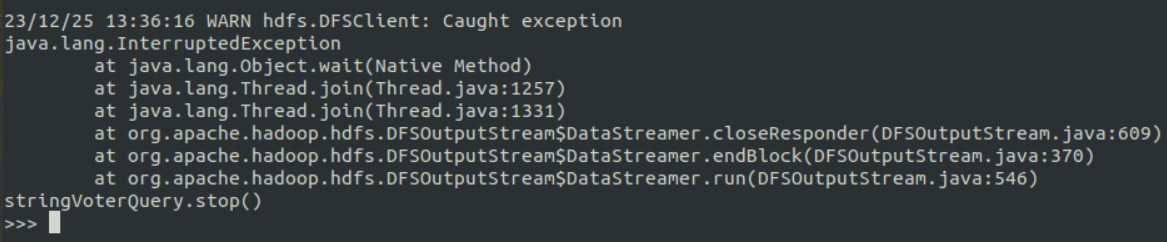




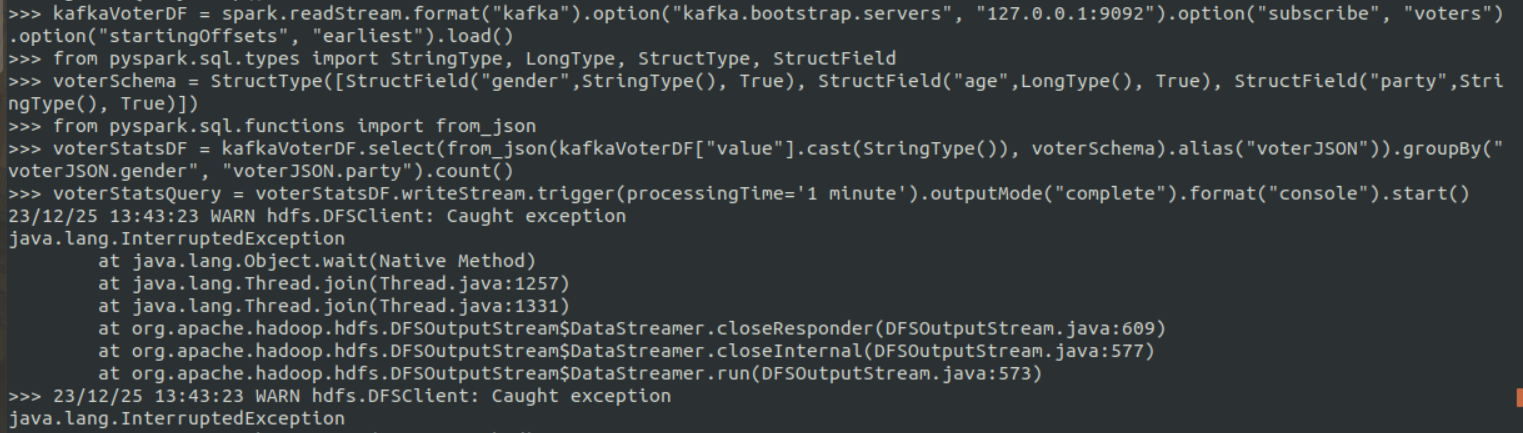




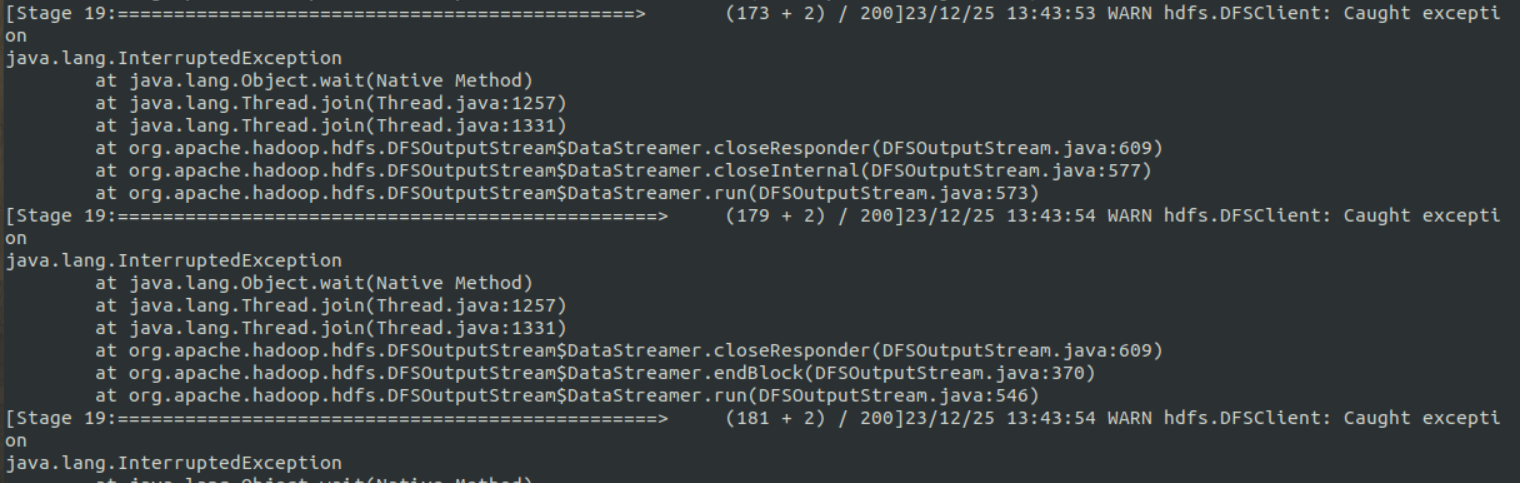
**Now it is in human readable format.**



**voterStatsQuery = voterStatsDF.writeStream.trigger(processingTime='10 seconds').outputMode("complete").format("console").start()**

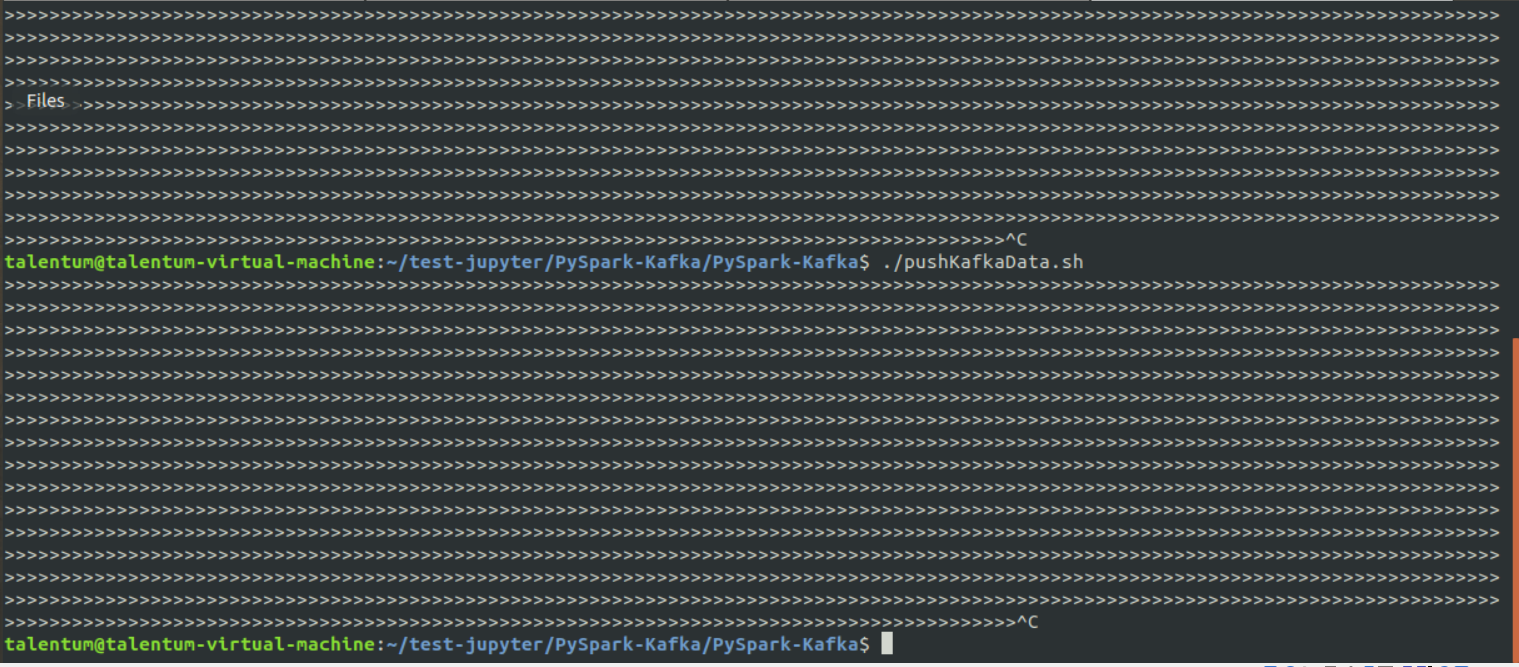


**Will return group data object by using .count()**

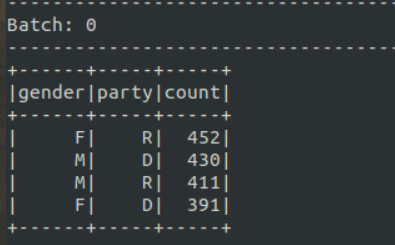


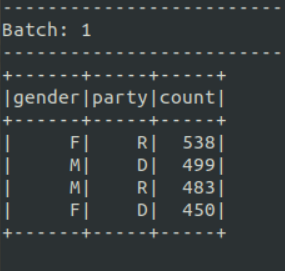
**In other shell run that command again**

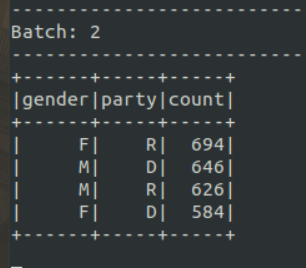
**./pushkafkaData.sh**



**Here after every specified time batches will be changed**







**Now stop repel**

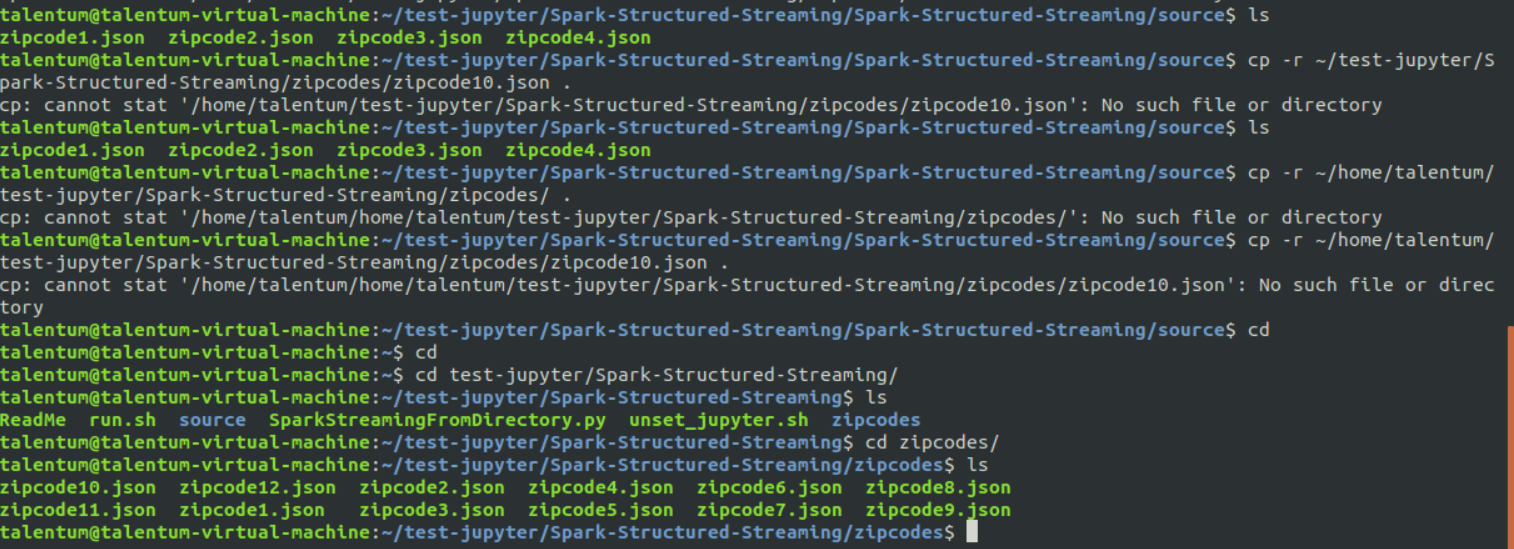
**voterStatsQuery.stop()**

19] SPark Streaming

1. Download the attached file in STAGING\_AREA
2. Pull the contents in LABS\_AREA/test-jupyter/Spark-Structured-Streaming
3. Study and understand the streaming application code written in a .py file, refer the api documentation wherever required
4. Follow the instructions in ReadMe file to run the application
5. What is this streaming application doing?

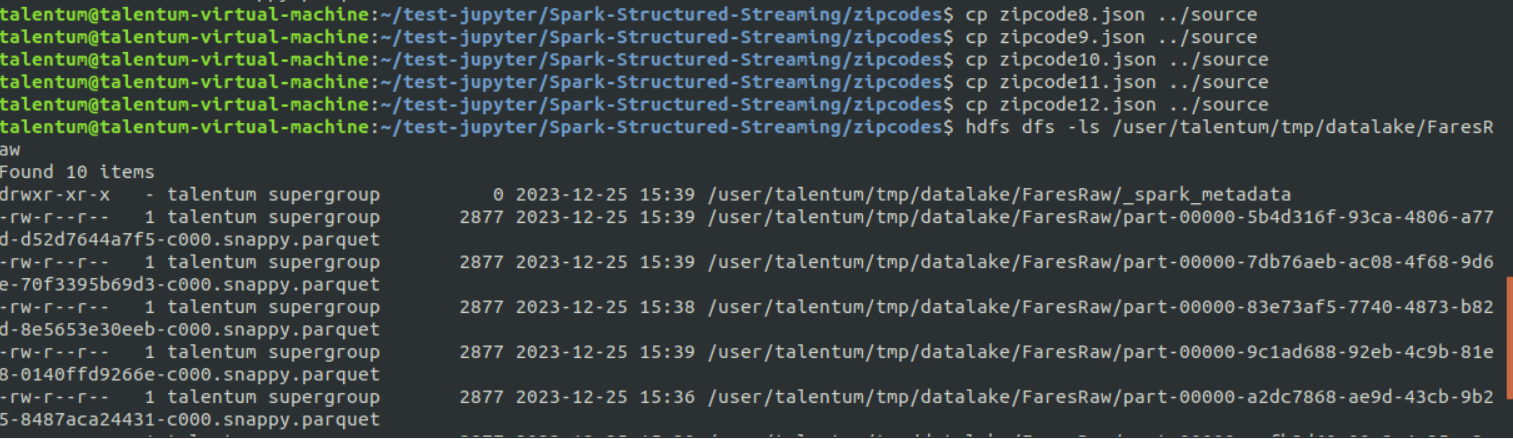
* Code/Dataset

data/Spark-Structured-Streaming.zip



**Copy files one by one from zipcode to source folder**

**And then check kar**



**Done**

