**Step 0:-**

**Install the below packages.**

**Pip install kafka-python**

**pip install configparser**

The above commands are documented in requirements.txt in the project root dir.

**Step 1:-**

**Add the Project directory to the Python path in the .bashrc file.**

1) vi .bashrc

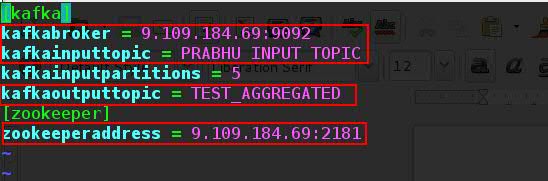
add the below line

export PYTHONPATH="${PYTHONPATH}:/projectrootdirectory"

the above is needed so that the Imports work..

**Step 2:-**

**Edit the following file projectrootdir/config/Configfile.properties**

****

Change the Kafka Broker address / Input topic name if needed / output topic name and also the zookeeper address.

Kafkainputpartitions is set to 5 purposefully since there are 5 unique keys as per the requirement and each of the keys data goes in to specific partitions. This is done to give parallelism to Spark streaming.

**Step3:-**

**Create the necessary topics in Kafka.**

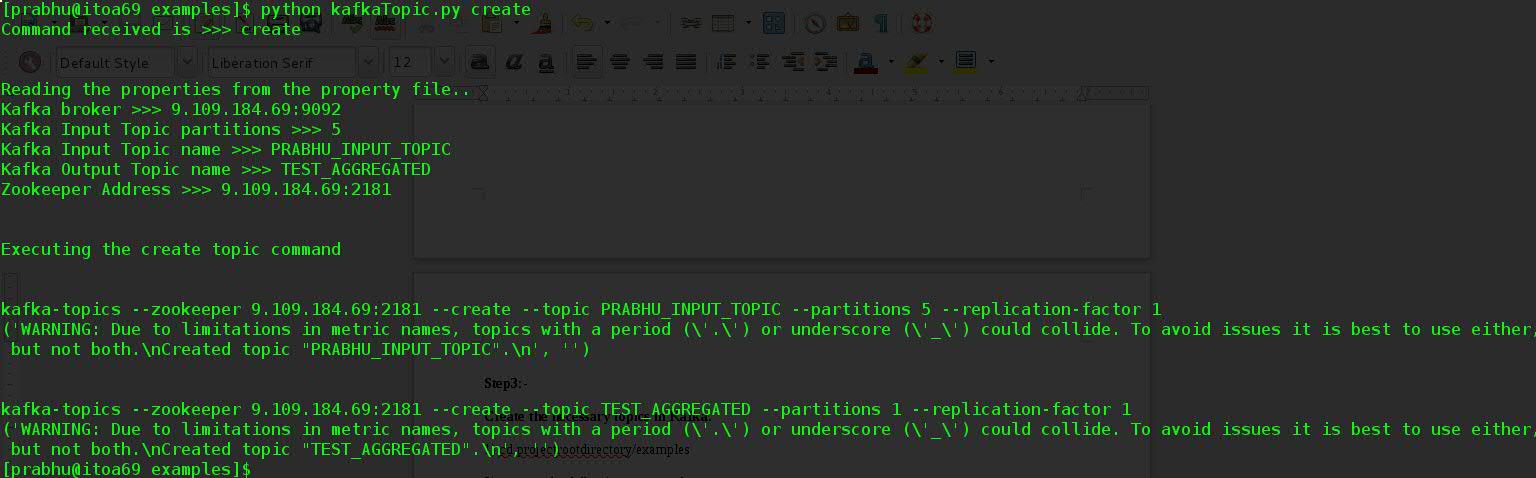
[ It is important that kafka commands are available for the user Id under which the python script is executed. ]

1) cd projectrootdirectory/examples

2) execute the following command

**python kafkaTopic.py create => creates the necessary input and output topics in kafka.**

Below is the screenshot on the command execution output..

****

The script basically executes the below two commands.

kafka-topics --zookeeper 9.109.184.69:2181 --create --topic PRABHU\_INPUT\_TOPIC --partitions 5 --replication-factor 1

kafka-topics --zookeeper 9.109.184.69:2181 --create --topic TEST\_AGGREGATED --partitions 1 --replication-factor 1

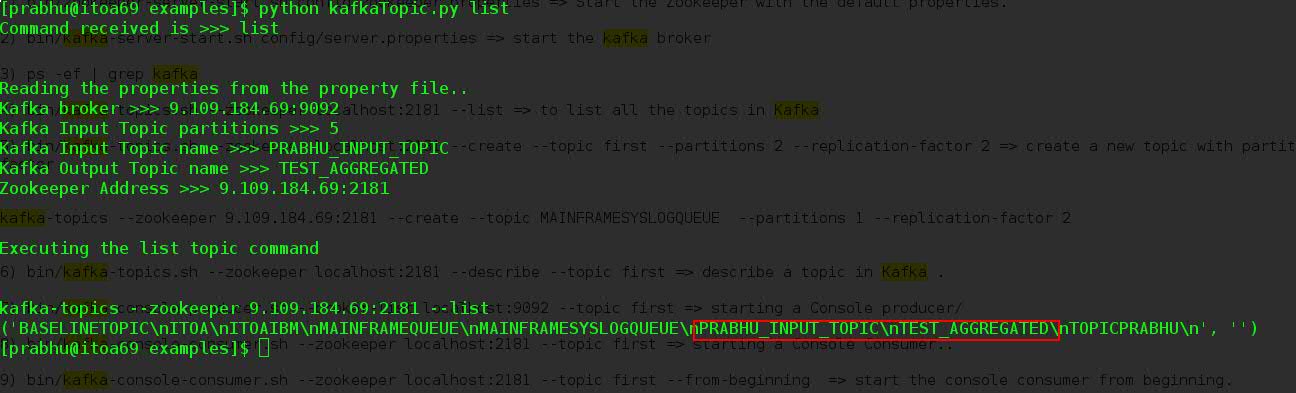
**Step 4:-**

**Verify if the necessary topics are being created ..**

1) cd projectrootdirectory/examples

2) execute the following command

**python kafkaTopic.py list => lists the topics from Kafka.**

****

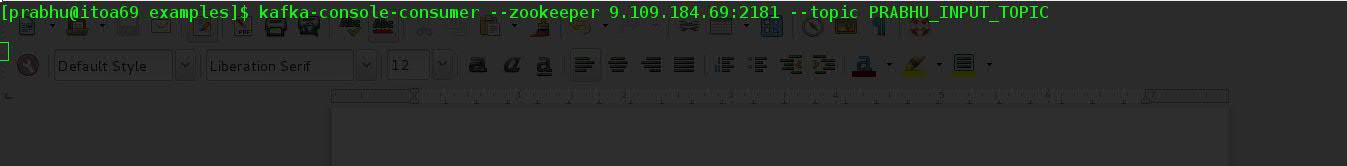
We can see from this output that the Input and Output kafka Topics are being created..

**python kafkaTopic.py delete => delete the topics from kafka.**

Use this just in case If the created topics are to be deleted.

**Step5 :-**

**Start the KafkaConsoleConsumer just to verify that the producer is sending messages..**



below is the command to start the kafka-console-consumer

**kafka-console-consumer --zookeeper 9.109.184.69:2181 --topic PRABHU\_INPUT\_TOPIC**

**Step6:-**

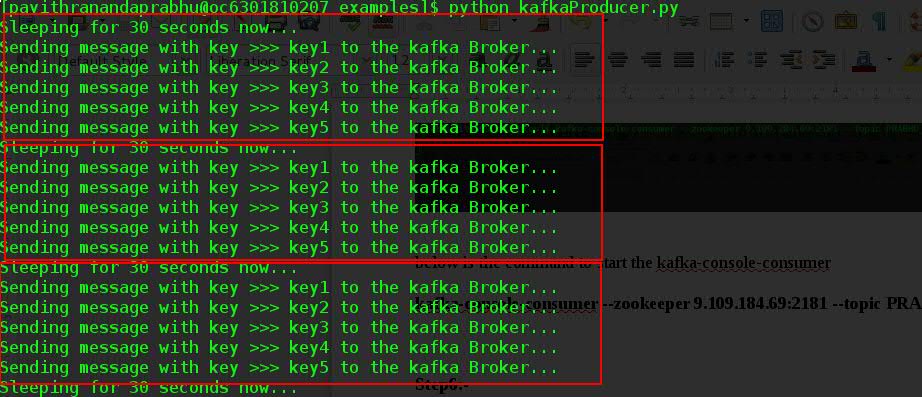
**Start the kafkaProducer .**

1) cd projectrootdirectory/examples

2) execute the following command

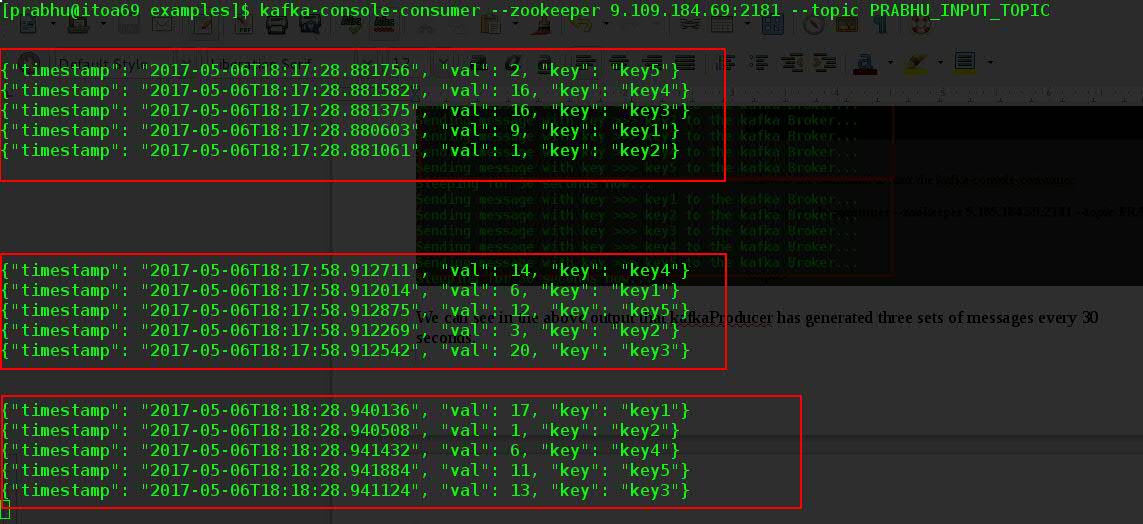
**python kafkaProducer.py**

this is a program which runs forever and generates messages to kafka every 30 seconds.



We can see in the above output that kafkaProducer has generated three sets of messages every 30 seconds.

Verify the output from the kafka-console-consumer If the consumer has received the messages sent by the producer.



We can see that the console consumer has also received three sets of 5 messages every 30 seconds.

Hence verifying that the producer program is working correctly.