Computer Science & Information Systems

**Real Time Analytics / Stream Processing & Analytics**

**Kafka Lab Sheet 3**

**Kafka Producer API**

1. Objective:

Students should be able to

1. Get familiarity with the working of Kafka Producer
2. Get hands-on experience writing Java /Python program involving message producers

Kafka has four core APIs:

* The Producer API allows an application to publish a stream of records to one or more Kafka topics.
* The Consumer API allows an application to subscribe to one or more topics and process the stream of records produced to them.
* The Streams API allows an application to act as a stream processor, consuming an input stream from one or more topics and producing an output stream to one or more output topics, effectively transforming the input streams to output streams.
* The Connector API allows building and running reusable producers or consumers that connect Kafka topics to existing applications or data systems. For example, a connector to a relational database might capture every change to a table.

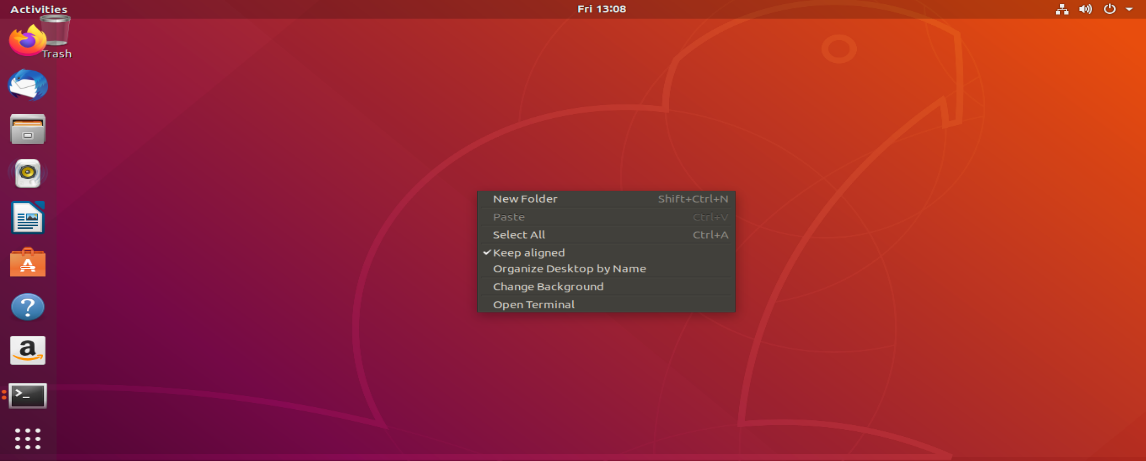
This lab sheet will introduce students with usage of Producer API with Java code and use of kafka-python package to demonstrate in Python.

1. Steps to be performed:

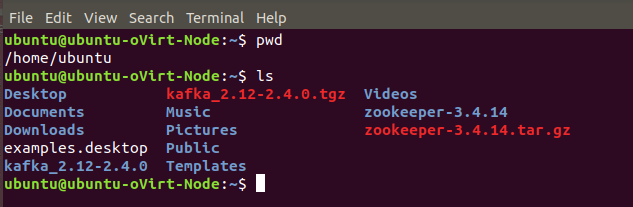
Note - It’s assumed that student has made a slot reservation using the slot booking interface where Apache Spark framework was selected. The details of the Apache Spark systems to be used is received through an email. If not, please contact the administrators for the same.

Also it’s assumed that students are aware of the process of logging into these virtual machines. If not, then get access to the user manual maintained for the usage of remote lab setup.

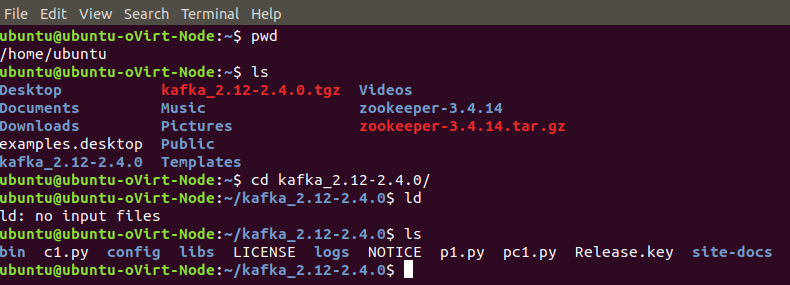
1. Open the terminal by right clicking on the desktop of the virtual machine.



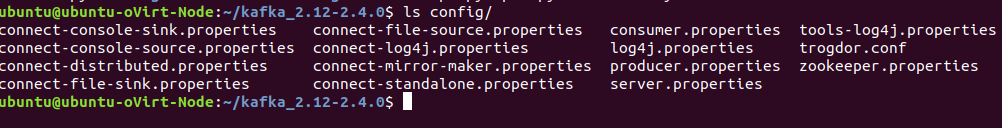
1. Look at the current directory and also file listings in it. It must have a Kafka installation directory present in it. Commands like pwd, ls can be used for it.

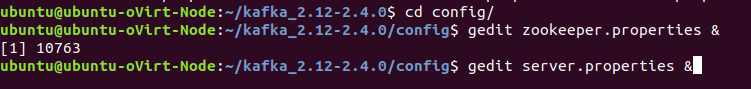


1. Change to the Kafka installation directory using the command and have a look at the files present in the directory.



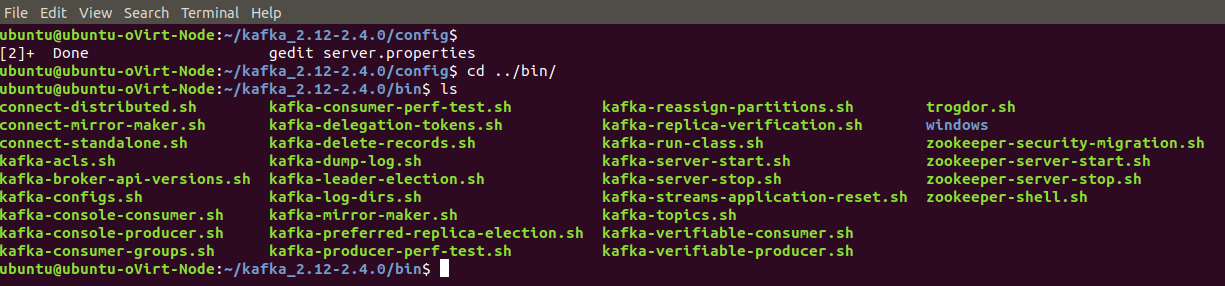
1. Change to the “config” directory present in the Kafka installation directory and have a look at the zookeeper and Kafka server properties files in it.

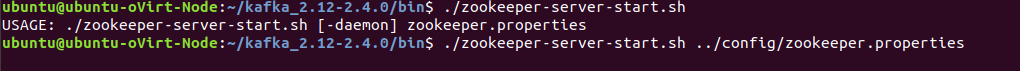




1. Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it. Will be using the following scripts to start and stop the Zookeeper ensemble

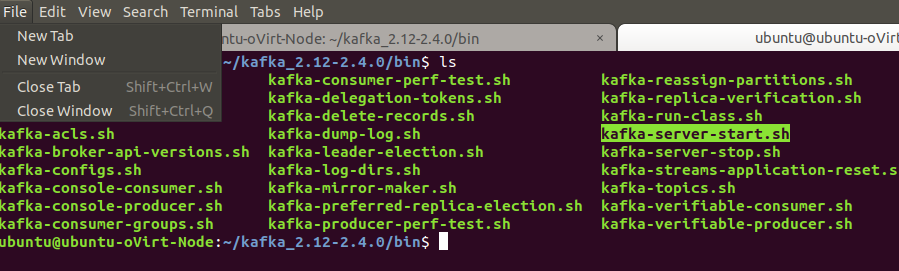
* zookeeper-server-start.sh
* zookeeper-server-stop.sh

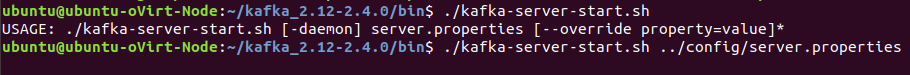




1. Open up another terminal and Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it. Will be using the following scripts to start and stop the Kafka server

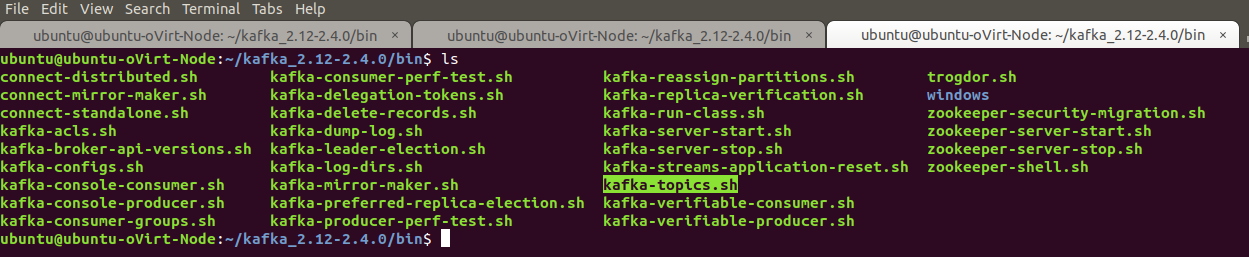
* kafka-server-start.sh
* kafka-server-stop.sh





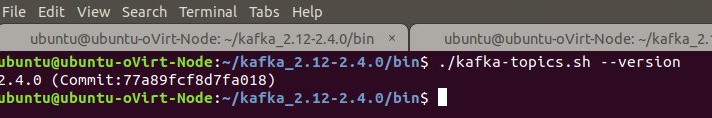
1. Open up another terminal and Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it. Will be using the following script to deal with the Kafka topics.

* kafka-topics.sh



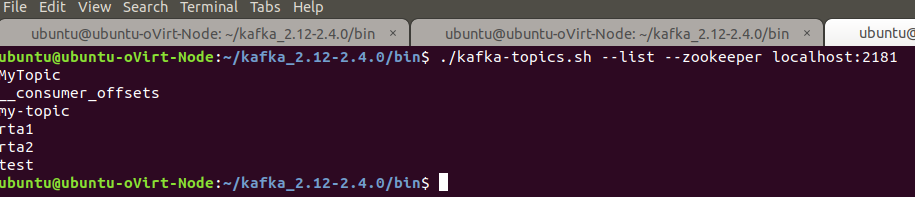
1. Use the –version option to see the version of Kafka.

* ./kafka-topics.sh –version



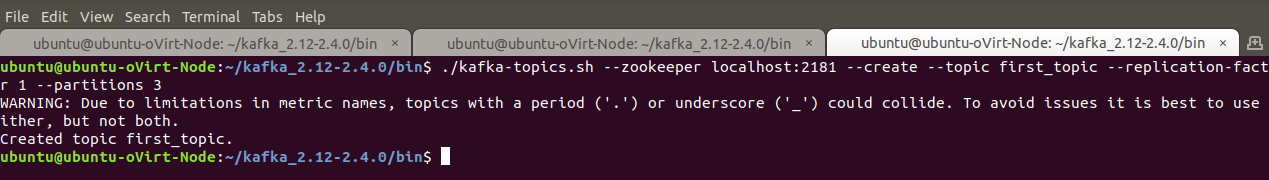
1. Use the –list option in it to list the topics present in the Kafka setup. Note – it will be empty for you as you don’t have created any topics as such but if the Kafka setup is shared you may see the topics created in earlier usages of the system.

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



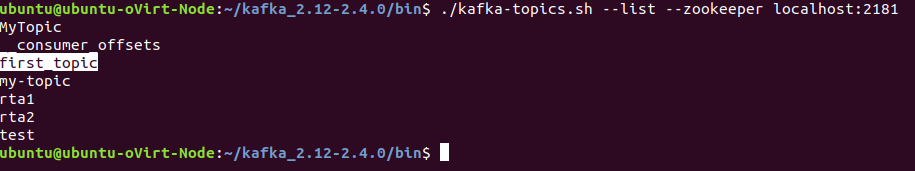
1. Let’s try to create a Kafka topic named “first\_topic” using the –create option.

* ./kafka-topics.sh --zookeeper localhost:2181 --create --topic first\_topic --replication-factor 1 --partitions 3



1. List the Kafka topics again. Now the “first\_topic” should appear in the topics list.

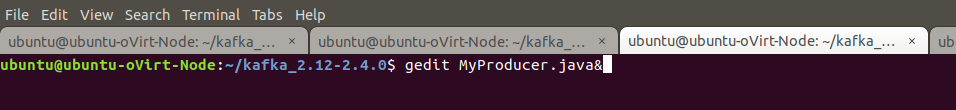
* ./kafka-topics.sh --list --zookeeper localhost:2181



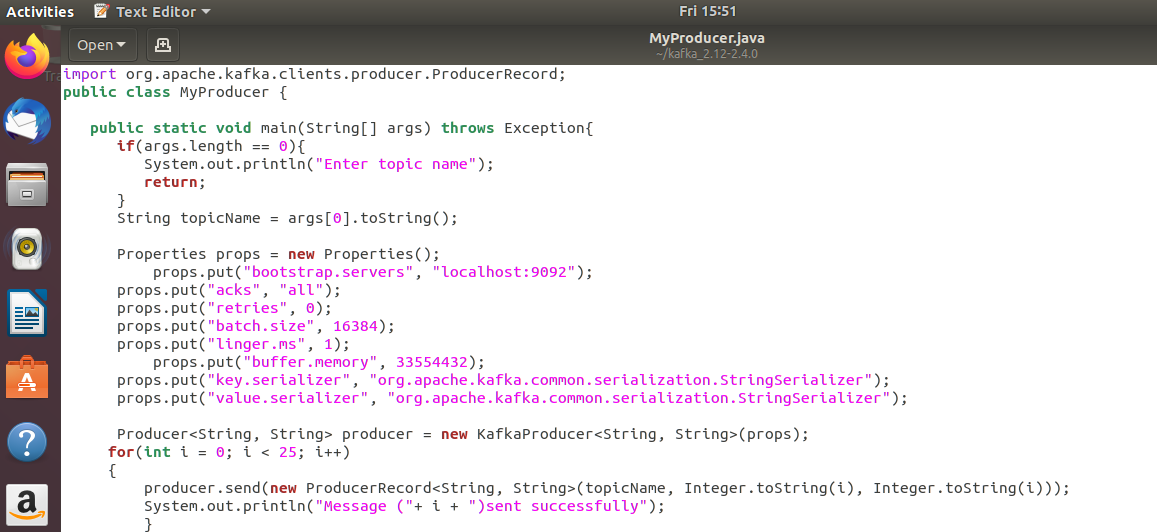
**Using Java Way:**

Now let’s see how we can write a Java program demoing the usage of Producer API.

1. Create a Java file named MyProducer.java.

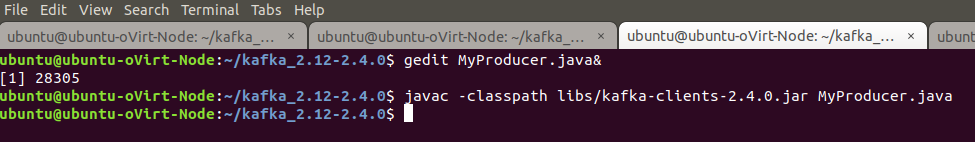


Copy paste the Java code from the attached MyProducer.java file in it.



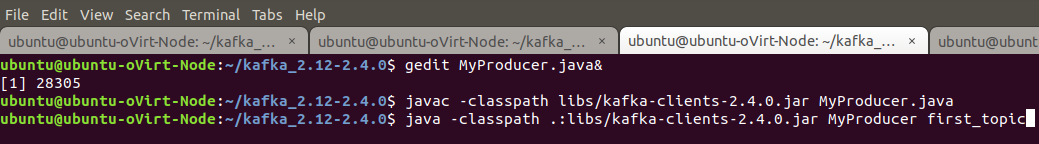
1. Compile the Java code using the following command

* javac -classpath libs/kafka-clients-2.4.0.jar MyProducer.java

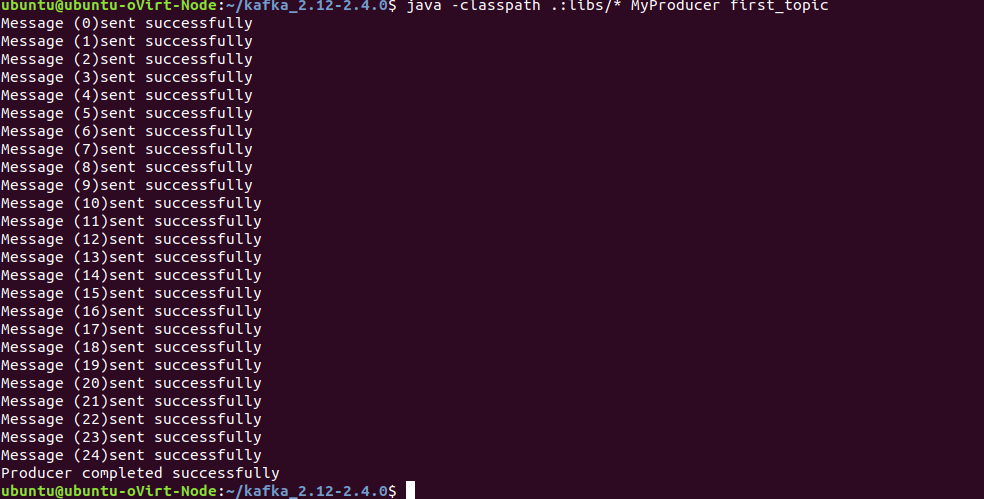


1. Run the Java code using the following command

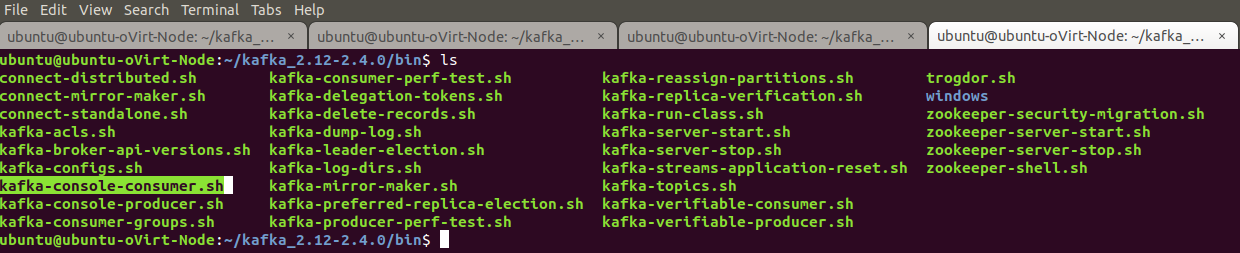
* java -classpath .:libs/\* MyProducer first\_topic



See that it has sent the 25 messages and same can be confirmed from the console output.

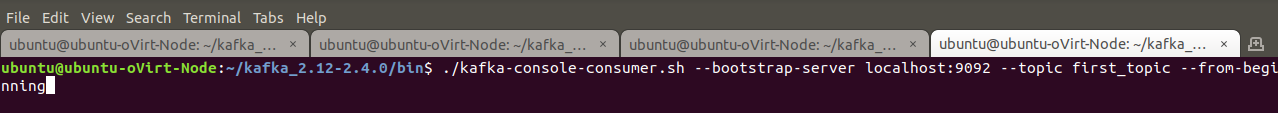


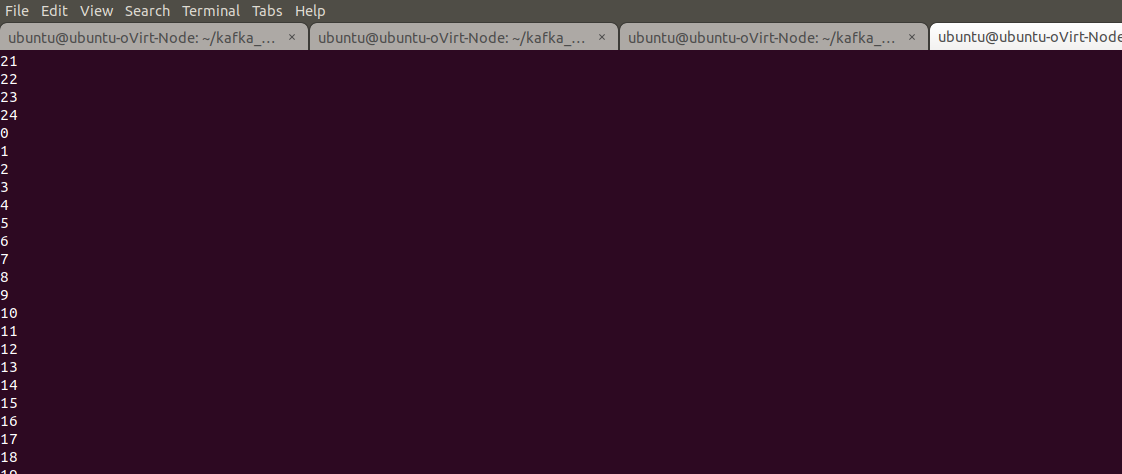
1. Open up another terminal and Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it.



The Kafka distribution provides a command utility to see messages from the command line. It displays the messages in various modes. Kafka provides the utility kafka-console-consumer.sh which is located at /bin/kafka-console-producer.sh to receive messages from a topic on the command line.

* ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic first\_topic --from-beginning



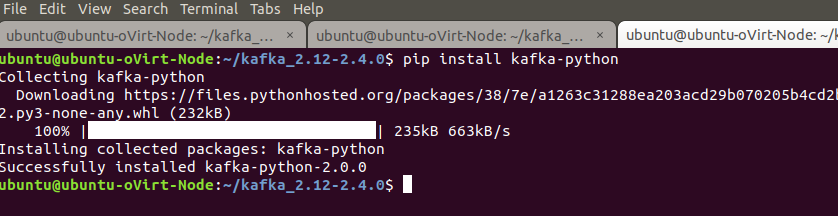


1. Execute the producer programe again and should be able to see that the consumer client is receiving those message.

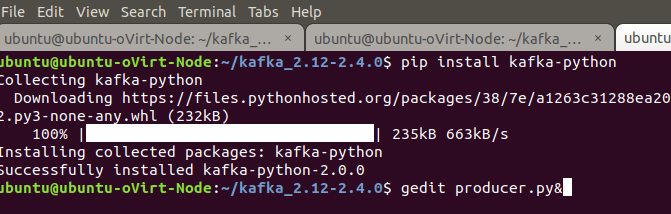
**Using Python Way:**

Now let’s see how we can write a Python program demoing the usage of Producer.

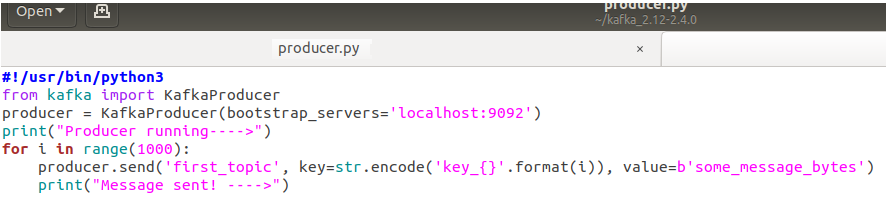
1. First install the kafka-python package.



1. Create a Python file named producer.py.

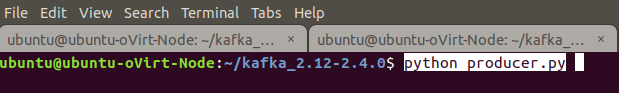


1. Copy paste the Python code from the attached producer.py file in it.

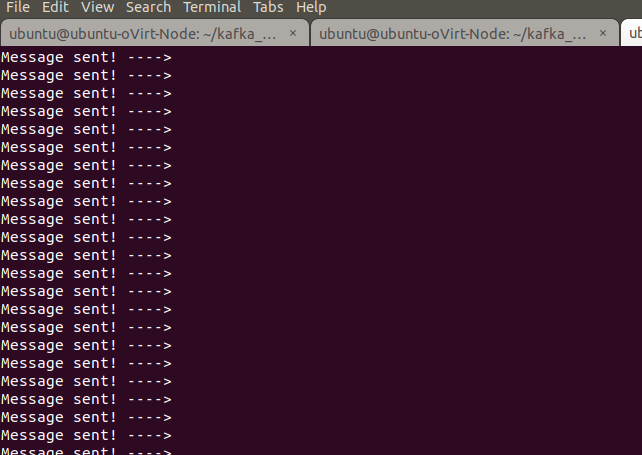


1. Run the Python code using the following command

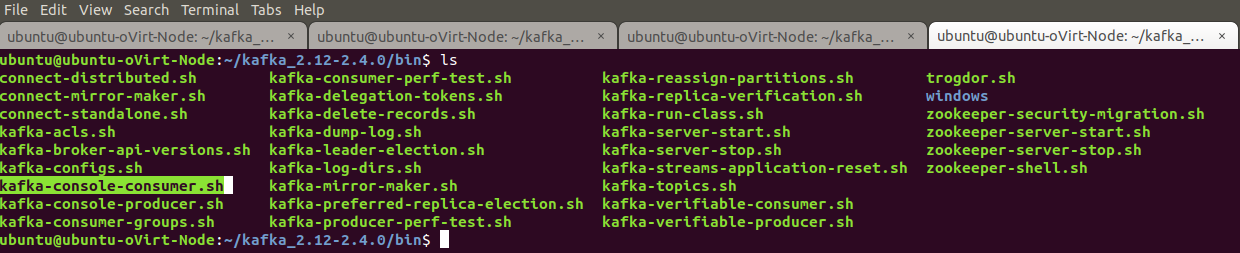
* python producer.py



1. See that it has sent the 1000 messages and same can be confirmed from the console output.

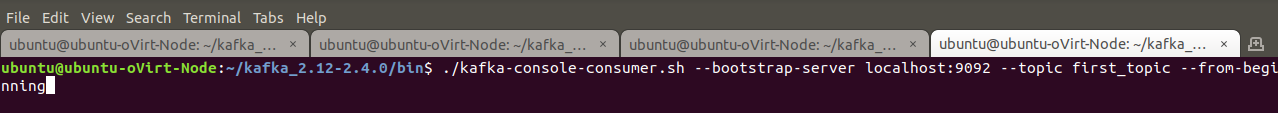


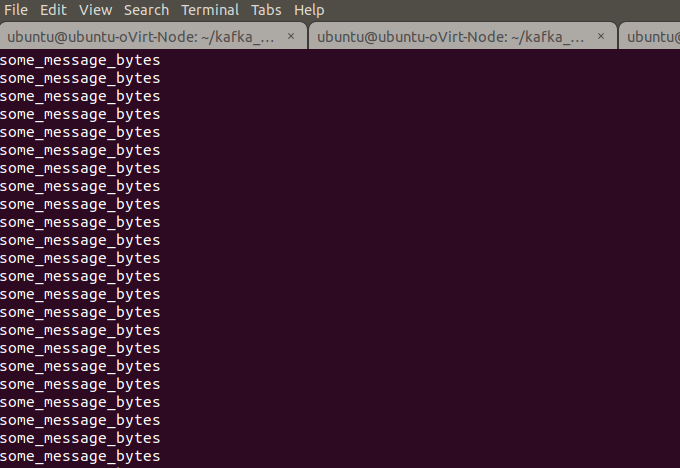
1. Open up another terminal and Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it.



The Kafka distribution provides a command utility to see messages from the command line. It displays the messages in various modes. Kafka provides the utility kafka-console-consumer.sh which is located at /bin/kafka-console-producer.sh to receive messages from a topic on the command line.

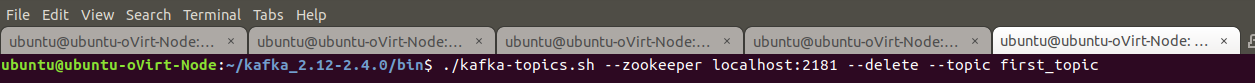
* ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic first\_topic --from-beginning





1. Execute the producer programe again and should be able to see that the consumer client is receiving those message.
2. Now let’s try to delete the topic. Open up another terminal and Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it. Will be using the following script to delete with the Kafka topics.

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



1. Outputs/Results:

Students should be able to use Kafka Producer

* To publish the messages into the Kakfa Topics using the Producer API
* To publish the messages into the Kafka Topics using kafka-python package

1. Observations:

Students carefully needs to observe the code written for the usage of Producer API for

* Specifying the Kafka topic configurations
* Linking to the Kafka topics
* Publishing the messages to the Kafka Topics

1. References:
2. [Kafka Quickstart](https://kafka.apache.org/quickstart)
3. [Kafka website](https://kafka.apache.org/)
4. [Kafka Streams](https://kafka.apache.org/24/documentation/streams/)