Work Integrated Learning Programmes

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Computer Science & Information Systems

Real Time Analytics / Stream Processing & Analytics

Apache Storm Lab Sheet 4

Storm – Kafka Integration

1. Objective:

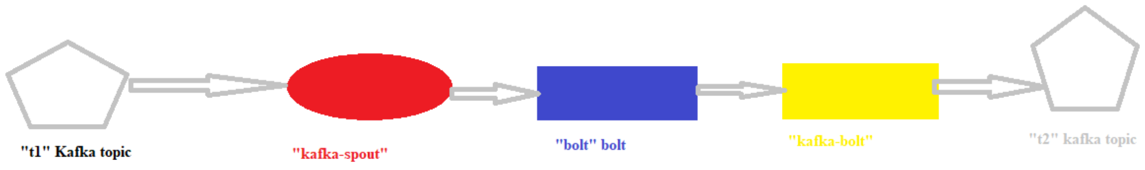
Students should be able to

A. Get hands-on experience of Storm – Kafka integration using the Java programs

Kafka and Storm naturally complement each other, and their powerful cooperation enables real-time streaming analytics for fast-moving big data. Kafka and Storm integration is to make easier for developers to ingest and publish data streams from Storm topologies.

A spout is a source of streams. For example, a spout may read tuples off a Kafka Topic and emit them as a stream. A bolt consumes input streams, process and possibly emits new streams. Bolts can do anything from running functions, filtering tuples, do streaming aggregations, streaming joins, talk to databases, and more. Each node in a Storm topology executes in parallel. A topology runs indefinitely until you terminate it. Storm will automatically reassign any failed tasks. Additionally, Storm guarantees that there will be no data loss, even if the machines go down and messages are dropped.

Let us go through the Kafka-Storm integration API's in detail. This lab sheet will introduce students with usage of Storm Topology with Java. The application that will be taken as example is word processing application. The topology will consist of a Kafka spout and two bolts – one of them being Kafka bolt. The spout will receive the words from the Kafka Topic. Its responsibility is to provide a word received from Kafka topic to the next bolt in topology. The first bolt is the topology will be responsible for enriching the words by adding some extra characters in it and forwarding the words to the next Kafka bolt. The Kafka bolt will output the enriched word to the Kafka topic. The topology of this Kafka – Storm integration application can be visualized as shown below.

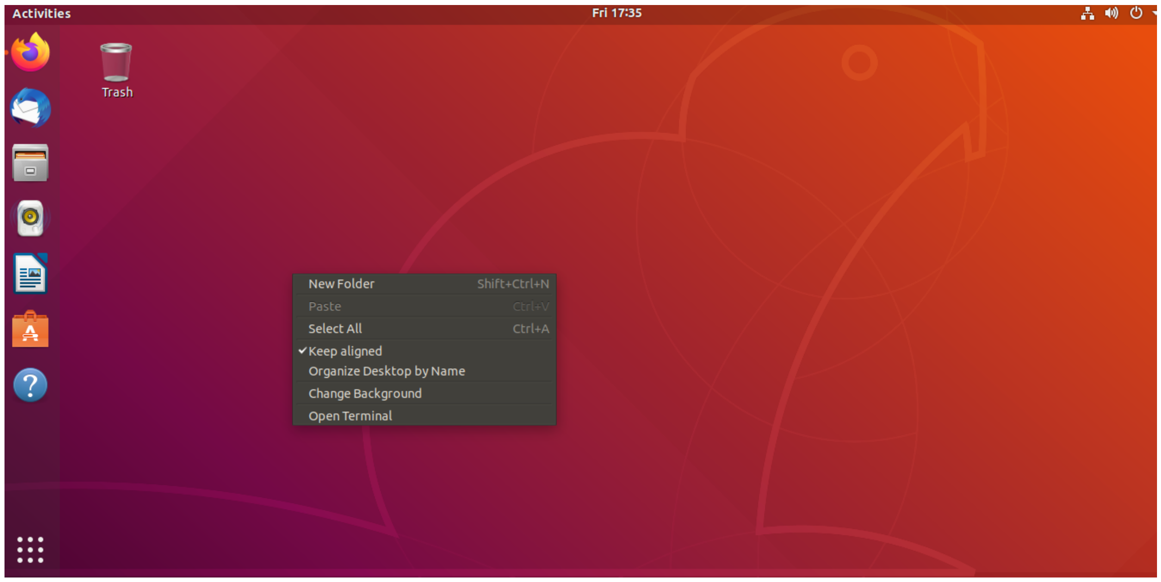


2. Steps to be performed:

Note - It's assumed that student has made a slot reservation using the slot booking interface where Apache Storm framework was selected. The details of the Apache Strom 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.

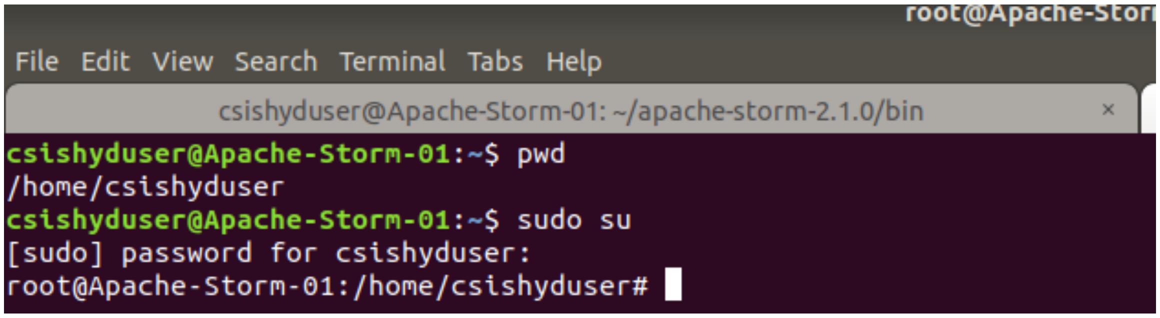
A. Open the terminal by right clicking on the desktop of the virtual machine.



B. Login as sudo user.

```
>>> sudo su
```

Provide the password provided in the email received from BITS remote lab team.



File Edit View Search Terminal Tabs Help

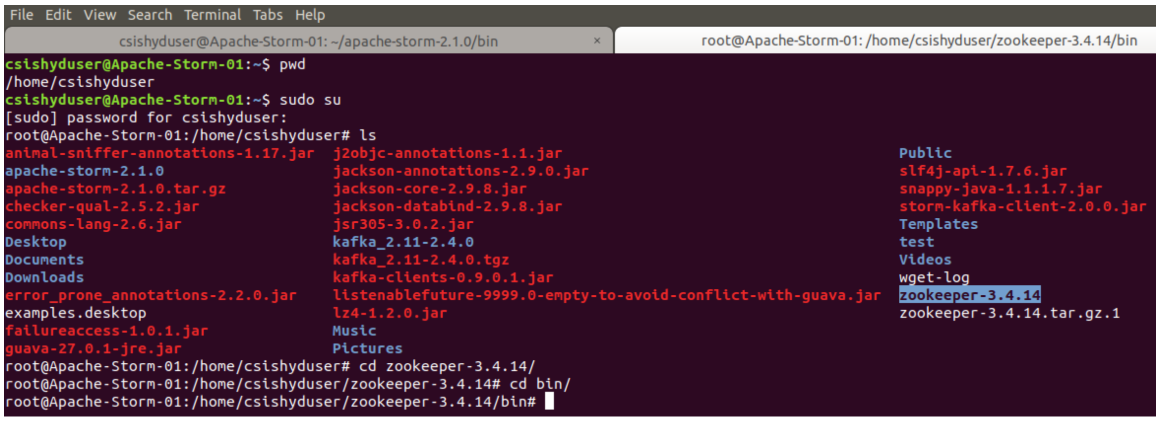
csishyduser@Apache-Storm-01: ~/apache-storm-2.1.0/bin

```
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ sudo su
[sudo] password for csishyduser:
root@Apache-Storm-01:/home/csishyduser#
```

C. Look at the current working directory using the “pwd” command. Then change the directory to the Zookeepers bin directory.

>>> pwd

>>> cd zookeeper-3.4.14/bin/



File Edit View Search Terminal Tabs Help

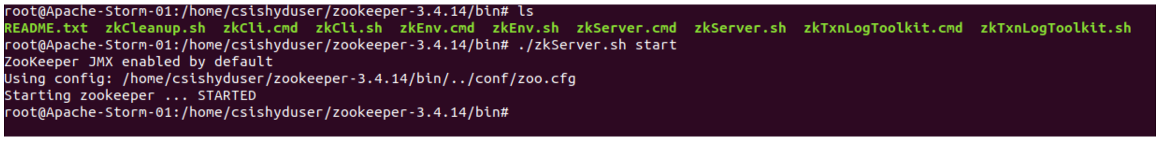
csishyduser@Apache-Storm-01: ~/apache-storm-2.1.0/bin

root@Apache-Storm-01: /home/csishyduser/zookeeper-3.4.14/bin

```
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ sudo su
[sudo] password for csishyduser:
root@Apache-Storm-01:/home/csishyduser# ls
animal-sniffer-annotations-1.17.jar  j2objc-annotations-1.1.jar                                Public
apache-storm-2.1.0                   jackson-annotations-2.9.0.jar                             slf4j-api-1.7.6.jar
apache-storm-2.1.0.tar.gz            jackson-core-2.9.8.jar                                    snappy-java-1.1.1.7.jar
checker-qual-2.5.2.jar               jackson-databind-2.9.8.jar                                storm-kafka-client-2.0.0.jar
commons-lang-2.6.jar                 jsr305-3.0.2.jar                                          Templates
Desktop                              kafka_2.11-2.4.0                                          test
Documents                            kafka_2.11-2.4.0.tgz                                      Videos
Downloads                            kafka-clients-0.9.0.1.jar                                 wget-log
error_prone_annotations-2.2.0.jar    listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar  zookeeper-3.4.14
examples.desktop                     lz4-1.2.0.jar                                             zookeeper-3.4.14.tar.gz.1
failureaccess-1.0.1.jar              Music
guava-27.0.1-jre.jar                 Pictures
root@Apache-Storm-01:/home/csishyduser# cd zookeeper-3.4.14/
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14# cd bin/
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14/bin#
```

D. Start the zookeeper.

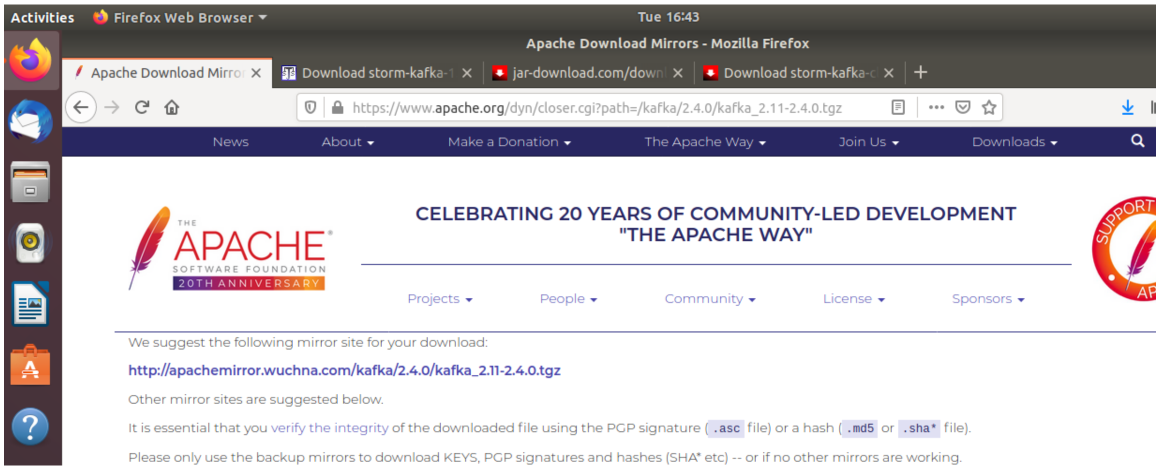
>>> ./zkServer.sh start



```
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14/bin# ls
README.txt  zkCleanup.sh  zkCli.cmd  zkCli.sh  zkEnv.cmd  zkEnv.sh  zkServer.cmd  zkServer.sh  zkTxnLogToolkit.cmd  zkTxnLogToolkit.sh
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14/bin# ./zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /home/csishyduser/zookeeper-3.4.14/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
root@Apache-Storm-01:/home/csishyduser/zookeeper-3.4.14/bin#
```

E. Open Firefox or any other browser application. Visit the following link in it to download the Apache Kafka tarball.

https://kafka.apache.org/downloads


We suggest the following mirror site for your download:

http://apachemirror.wuchna.com/kafka/2.4.0/kafka_2.11-2.4.0.tgz

Other mirror sites are suggested below.

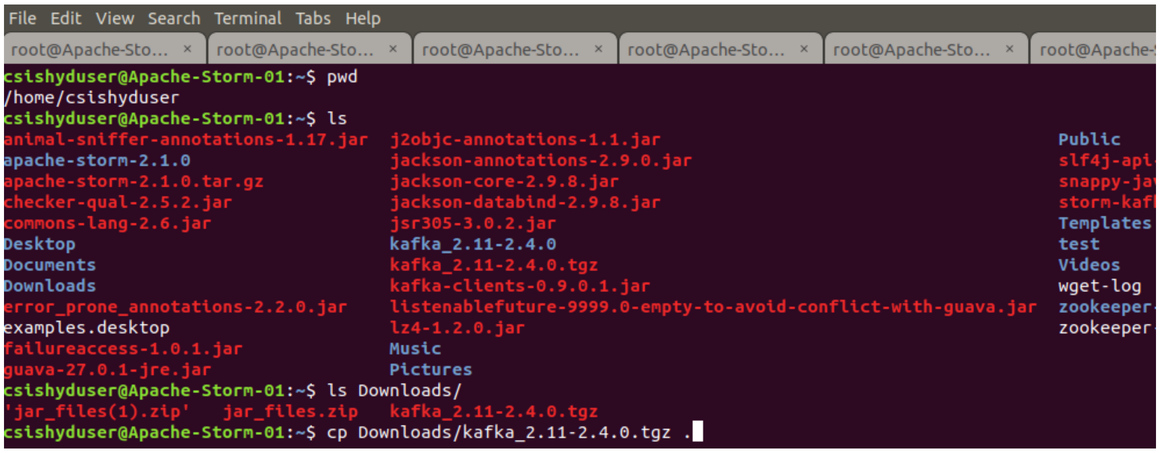
It is essential that you verify the integrity of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

Activities
Firefox Web Browser
Tue 16:43
Apache Download Mirrors - Mozilla Firefox
https://www.apache.org/dyn/closer.cgi?path=/kafka/2.4.0/kafka_2.11-2.4.0.tgz
News
About
Make a Donation
The Apache Way
Join Us
Downloads
CELEBRATING 20 YEARS OF COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"
Projects
People
Community
License
Sponsors

F. The tarball will get download into "Downloads" directory. Copy it into the home directory or the directory into which you want to do its installation.

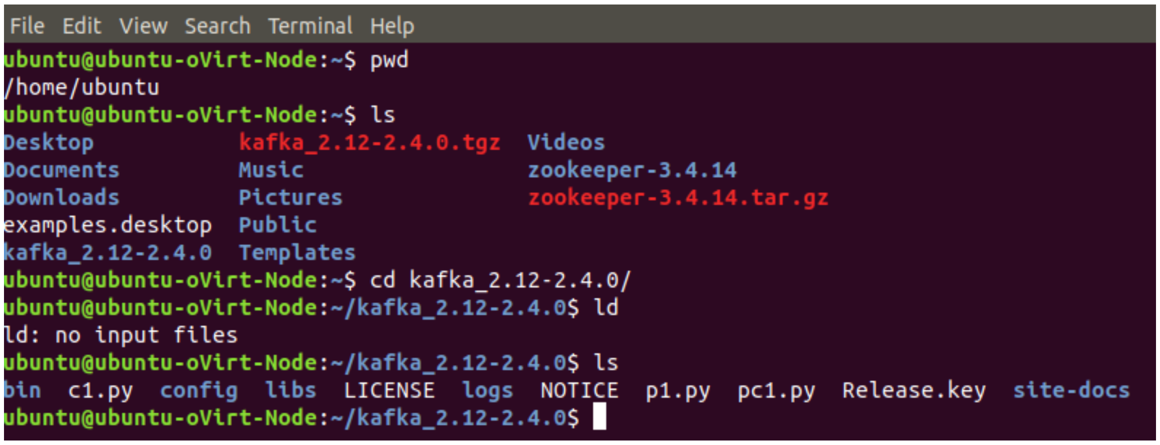
>>> cp Downloads/kafka_2.11-2.4.0.tgz .


```
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ ls
animal-sniffer-annotations-1.17.jar  j2objc-annotations-1.1.jar                                    Public
apache-storm-2.1.0                   jackson-annotations-2.9.0.jar                                 slf4j-api
apache-storm-2.1.0.tar.gz            jackson-core-2.9.8.jar                                        snappy-ja
checker-qual-2.5.2.jar               jackson-databind-2.9.8.jar                                    storm-kaf
commons-lang-2.6.jar                 jsr305-3.0.2.jar                                              Templates
Desktop                              kafka_2.11-2.4.0                                              test
Documents                            kafka_2.11-2.4.0.tgz                                          Videos
Downloads                            kafka-clients-0.9.0.1.jar                                     wget-log
error_prone_annotations-2.2.0.jar    listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar zookeeper
examples.desktop                     lz4-1.2.0.jar                                                 zookeeper
failureaccess-1.0.1.jar              Music
guava-27.0.1-jre.jar                 Pictures
csishyduser@Apache-Storm-01:~$ ls Downloads/
'jar_files(1).zip'   jar_files.zip   kafka_2.11-2.4.0.tgz
csishyduser@Apache-Storm-01:~$ cp Downloads/kafka_2.11-2.4.0.tgz .
```

G. Unzip the tarball in the current directory.

>>> tar -xf kafka_2.11-2.4.0.tgz

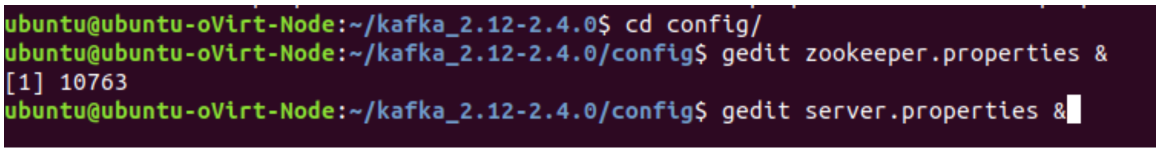
H. Change to the Kafka installation directory using the command and have a look at the files present in the directory.



```
ubuntu@ubuntu-oVirt-Node:~$ pwd
/home/ubuntu
ubuntu@ubuntu-oVirt-Node:~$ ls
Desktop           kafka_2.12-2.4.0.tgz  Videos
Documents         Music                 zookeeper-3.4.14
Downloads         Pictures              zookeeper-3.4.14.tar.gz
examples.desktop  Public
kafka_2.12-2.4.0  Templates
ubuntu@ubuntu-oVirt-Node:~$ cd kafka_2.12-2.4.0/
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0$ ld
ld: no input files
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0$ ls
bin  c1.py  config  libs  LICENSE  logs  NOTICE  p1.py  pc1.py  Release.key  site-docs
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0$
```

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

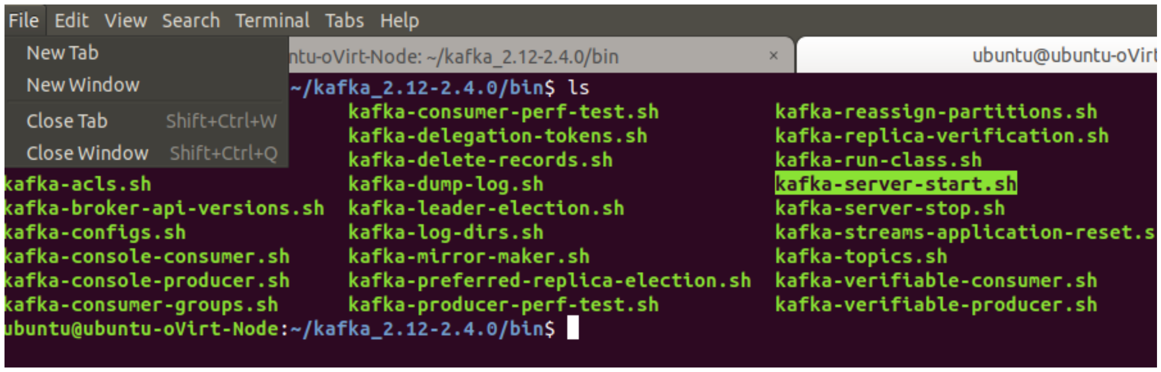
```
>>> cd config

>>>gedit server.properties&
```



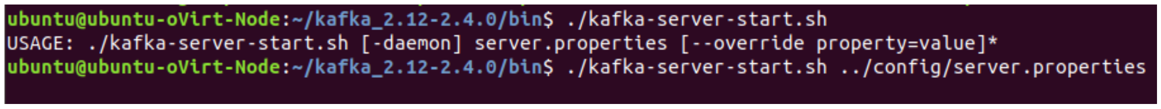
```
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0$ cd config/
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/config$ gedit zookeeper.properties &
[1] 10763
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/config$ gedit server.properties &
```

J. 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


File Edit View Search Terminal Tabs Help
New Tab
New Window
Close Tab Shift+Ctrl+W
Close Window Shift+Ctrl+Q
ntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin
ubuntu@ubuntu-oVirt

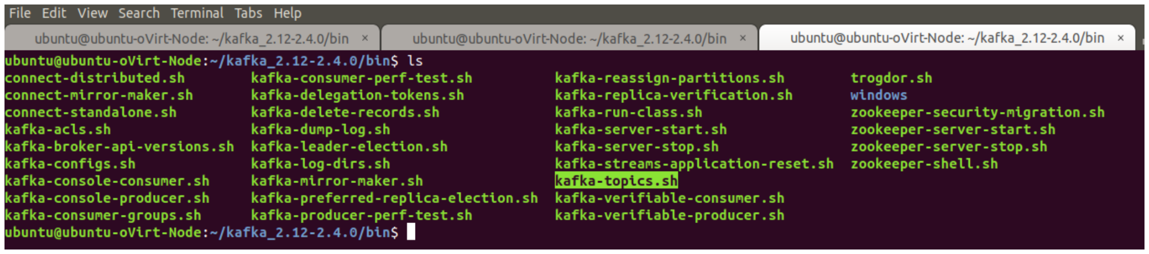
```
~/kafka_2.12-2.4.0/bin$ ls
          kafka-consumer-perf-test.sh          kafka-reassign-partitions.sh
          kafka-delegation-tokens.sh           kafka-replica-verification.sh
          kafka-delete-records.sh              kafka-run-class.sh
kafka-acls.sh                  kafka-dump-log.sh                    kafka-server-start.sh
kafka-broker-api-versions.sh   kafka-leader-election.sh             kafka-server-stop.sh
kafka-configs.sh               kafka-log-dirs.sh                    kafka-streams-application-reset.s
kafka-console-consumer.sh      kafka-mirror-maker.sh                kafka-topics.sh
kafka-console-producer.sh      kafka-preferred-replica-election.sh  kafka-verifiable-consumer.sh
kafka-consumer-groups.sh       kafka-producer-perf-test.sh          kafka-verifiable-producer.sh
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$
```



```
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$ ./kafka-server-start.sh
USAGE: ./kafka-server-start.sh [-daemon] server.properties [--override property=value]*
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$ ./kafka-server-start.sh ../config/server.properties
```

K. 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.

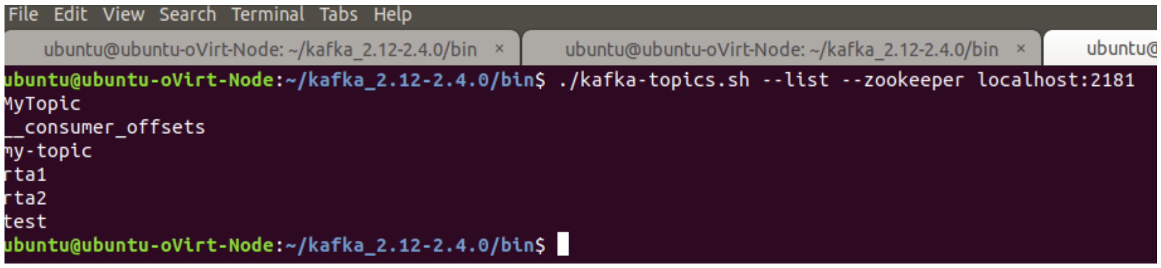
- kafka-topics.sh


File Edit View Search Terminal Tabs Help
ubuntu@ubuntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin
ubuntu@ubuntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin
ubuntu@ubuntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin

```
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$ ls
connect-distributed.sh        kafka-consumer-perf-test.sh          kafka-reassign-partitions.sh        trogdor.sh
connect-mirror-maker.sh       kafka-delegation-tokens.sh           kafka-replica-verification.sh       windows
connect-standalone.sh         kafka-delete-records.sh              kafka-run-class.sh                  zookeeper-security-migration.sh
kafka-acls.sh                 kafka-dump-log.sh                    kafka-server-start.sh               zookeeper-server-start.sh
kafka-broker-api-versions.sh  kafka-leader-election.sh             kafka-server-stop.sh                zookeeper-server-stop.sh
kafka-configs.sh              kafka-log-dirs.sh                    kafka-streams-application-reset.sh  zookeeper-shell.sh
kafka-console-consumer.sh     kafka-mirror-maker.sh                kafka-topics.sh
kafka-console-producer.sh     kafka-preferred-replica-election.sh  kafka-verifiable-consumer.sh
kafka-consumer-groups.sh      kafka-producer-perf-test.sh          kafka-verifiable-producer.sh
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$
```

L. 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


File Edit View Search Terminal Tabs Help
ubuntu@ubuntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin
ubuntu@ubuntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin
ubuntu@

```
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$ ./kafka-topics.sh --list --zookeeper localhost:2181
MyTopic
__consumer_offsets
my-topic
rta1
rta2
test
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$
```

M. Let’s try to create a Kafka topic named “t1” and “t2” using the –create option.

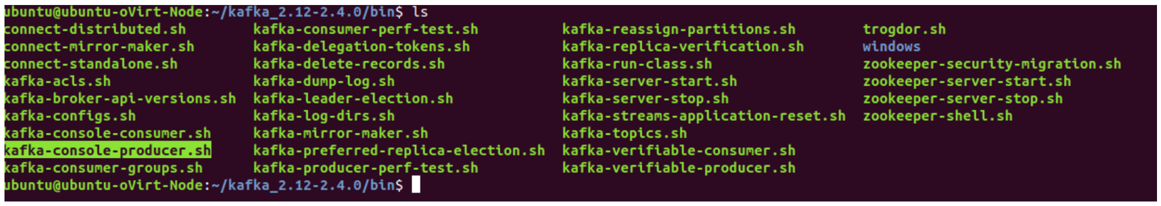
- ./kafka-topics.sh --zookeeper localhost:2181 --create --topic t1 --replication-factor 1 --partitions 3
- ./kafka-topics.sh --zookeeper localhost:2181 --create --topic t2 --replication-factor 1 --partitions 3

N. List the Kafka topics again. Now the “t1” and “t2” should appear in the topics list.

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

O. The Kafka distribution provides a command utility to send messages from the command line. It start up a terminal window where everything you type is sent to the Kafka topic. Kafka provides the utility kafka-console-producer.sh to send messages to a topic on the command line.

- ./kafka-console-producer.sh --topic t1 --broker-list localhost:9092

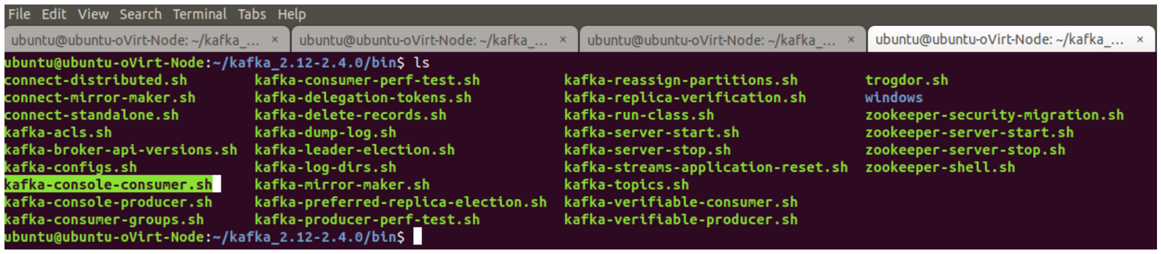


```
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$ ls
connect-distributed.sh        kafka-consumer-perf-test.sh          kafka-reassign-partitions.sh         trogdor.sh
connect-mirror-maker.sh       kafka-delegation-tokens.sh           kafka-replica-verification.sh        windows
connect-standalone.sh         kafka-delete-records.sh              kafka-run-class.sh                   zookeeper-security-migration.sh
kafka-acls.sh                 kafka-dump-log.sh                    kafka-server-start.sh                zookeeper-server-start.sh
kafka-broker-api-versions.sh  kafka-leader-election.sh             kafka-server-stop.sh                 zookeeper-server-stop.sh
kafka-configs.sh              kafka-log-dirs.sh                    kafka-streams-application-reset.sh   zookeeper-shell.sh
kafka-console-consumer.sh     kafka-mirror-maker.sh                kafka-topics.sh
kafka-console-producer.sh     kafka-preferred-replica-election.sh  kafka-verifiable-consumer.sh
kafka-consumer-groups.sh      kafka-producer-perf-test.sh          kafka-verifiable-producer.sh
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$
```

P. Try inserting some message in the command prompt provided by producer utility.

>>> ./kafka-console-producer.sh --broker-list localhost:9092 --topic t1

Q. 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.



```
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$ ls
connect-distributed.sh        kafka-consumer-perf-test.sh          kafka-reassign-partitions.sh         trogdor.sh
connect-mirror-maker.sh       kafka-delegation-tokens.sh           kafka-replica-verification.sh        windows
connect-standalone.sh         kafka-delete-records.sh              kafka-run-class.sh                   zookeeper-security-migration.sh
kafka-acls.sh                 kafka-dump-log.sh                    kafka-server-start.sh                zookeeper-server-start.sh
kafka-broker-api-versions.sh  kafka-leader-election.sh             kafka-server-stop.sh                 zookeeper-server-stop.sh
kafka-configs.sh              kafka-log-dirs.sh                    kafka-streams-application-reset.sh   zookeeper-shell.sh
kafka-console-consumer.sh     kafka-mirror-maker.sh                kafka-topics.sh
kafka-console-producer.sh     kafka-preferred-replica-election.sh  kafka-verifiable-consumer.sh
kafka-consumer-groups.sh      kafka-producer-perf-test.sh          kafka-verifiable-producer.sh
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$
```

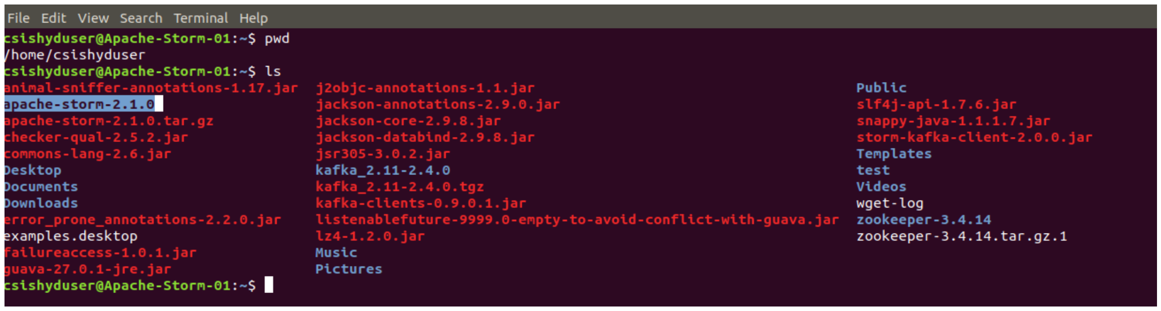
R. The Kafka distribution provides a command utility to see messages from the command line. It displays the messages in various modes. Kafka provides the utility kafka-console-consumer.sh which is located at /bin/kafka-console-producer.sh to receive messages from a topic on the command line.

- ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic t2

S. Open another terminal. Look at the current working directory using the “pwd” command. Then change the directory to the Storms directory.

>>> pwd

>>> cd apache-storm-2.1.0/

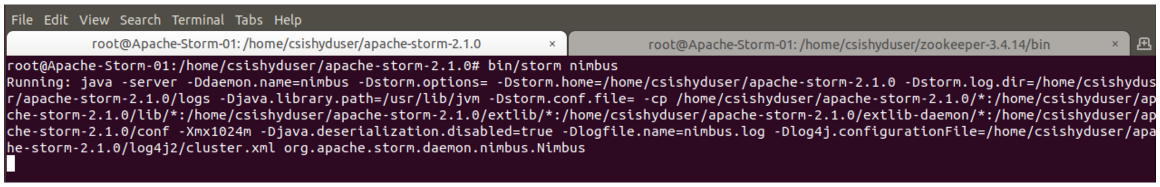


```
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ ls
animal-sniffer-annotations-1.17.jar  j2objc-annotations-1.1.jar                                Public
apache-storm-2.1.0                   jackson-annotations-2.9.0.jar                             slf4j-api-1.7.6.jar
apache-storm-2.1.0.tar.gz            jackson-core-2.9.8.jar                                    snappy-java-1.1.1.7.jar
checker-qual-2.5.2.jar               jackson-databind-2.9.8.jar                                storm-kafka-client-2.0.0.jar
commons-lang-2.6.jar                 jsr305-3.0.2.jar                                          Templates
Desktop                              kafka_2.11-2.4.0                                          test
Documents                            kafka_2.11-2.4.0.tgz                                      Videos
Downloads                            kafka-clients-0.9.0.1.jar                                 wget-log
error_prone_annotations-2.2.0.jar    listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar  zookeeper-3.4.14
examples.desktop                     lz4-1.2.0.jar                                             zookeeper-3.4.14.tar.gz.1
failureaccess-1.0.1.jar              Music
guava-27.0.1-jre.jar                 Pictures
csishyduser@Apache-Storm-01:~$
```

T. Start the nimbus node using the storm command.

>>> bin/strom nimbus

Note – if the command fails, then login as sudo su and then try again.

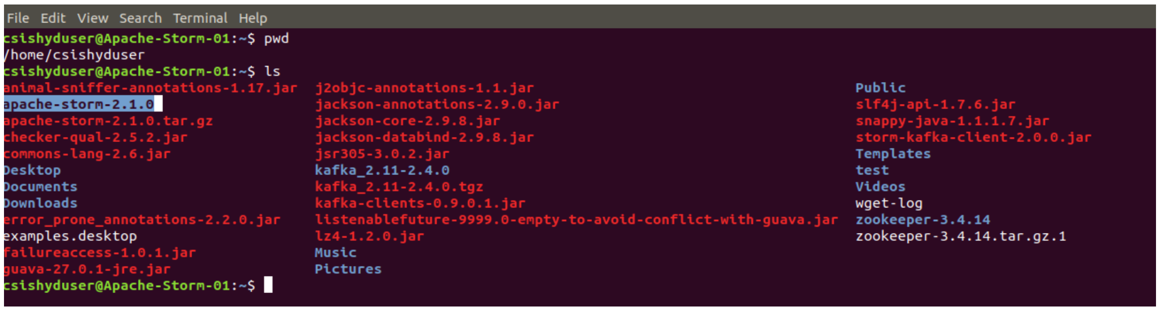


```
File  Edit  View  Search  Terminal  Tabs  Help
root@Apache-Storm-01: /home/csishyduser/apache-storm-2.1.0    ×    root@Apache-Storm-01: /home/csishyduser/zookeeper-3.4.14/bin    ×
root@Apache-Storm-01:/home/csishyduser/apache-storm-2.1.0# bin/storm nimbus
Running: java -server -Ddaemon.name=nimbus -Dstorm.options= -Dstorm.home=/home/csishyduser/apache-storm-2.1.0 -Dstorm.log.dir=/home/csishydus
r/apache-storm-2.1.0/logs -Djava.library.path=/usr/lib/jvm -Dstorm.conf.file= -cp /home/csishyduser/apache-storm-2.1.0/*:/home/csishyduser/ap
che-storm-2.1.0/lib/*:/home/csishyduser/apache-storm-2.1.0/extlib/*:/home/csishyduser/apache-storm-2.1.0/extlib-daemon/*:/home/csishyduser/ap
che-storm-2.1.0/conf -Xmx1024m -Djava.deserialization.disabled=true -Dlogfile.name=nimbus.log -Dlog4j.configurationFile=/home/csishyduser/apa
he-storm-2.1.0/log4j2/cluster.xml org.apache.storm.daemon.nimbus.Nimbus
```

U. Open another terminal. Look at the current working directory using the "pwd" command. Then change the directory to the Storms directory.

>>> pwd

>>> cd apache-storm-2.1.0/

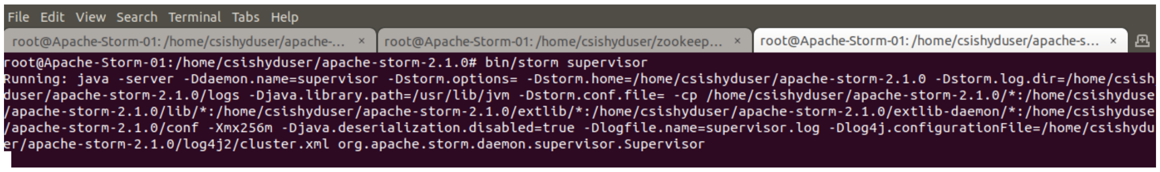


```
File  Edit  View  Search  Terminal  Help
csishyduser@Apache-Storm-01:~$ pwd
/home/csishyduser
csishyduser@Apache-Storm-01:~$ ls
animal-sniffer-annotations-1.17.jar  j2objc-annotations-1.1.jar                                Public
apache-storm-2.1.0                   jackson-annotations-2.9.0.jar                             slf4j-api-1.7.6.jar
apache-storm-2.1.0.tar.gz            jackson-core-2.9.8.jar                                    snappy-java-1.1.1.7.jar
checker-qual-2.5.2.jar               jackson-databind-2.9.8.jar                                storm-kafka-client-2.0.0.jar
commons-lang-2.6.jar                 jsr305-3.0.2.jar                                          Templates
Desktop                              kafka_2.11-2.4.0                                          test
Documents                            kafka_2.11-2.4.0.tgz                                      Videos
Downloads                            kafka-clients-0.9.0.1.jar                                 wget-log
error_prone_annotations-2.2.0.jar    listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar  zookeeper-3.4.14
examples.desktop                     lz4-1.2.0.jar                                             zookeeper-3.4.14.tar.gz.1
failureaccess-1.0.1.jar              Music
guava-27.0.1-jre.jar                 Pictures
csishyduser@Apache-Storm-01:~$
```

V. Start the supervisor node using the storm command.

>>> bin/strom suporvisor

Note – if the command fails, then login as sudo su and then try again.



```
File  Edit  View  Search  Terminal  Tabs  Help
root@Apache-Storm-01: /home/csishyduser/apache-...  ×  root@Apache-Storm-01: /home/csishyduser/zookeep...  ×  root@Apache-Storm-01: /home/csishyduser/apache-s...  ×
root@Apache-Storm-01:/home/csishyduser/apache-storm-2.1.0# bin/storm supervisor
Running: java -server -Ddaemon.name=supervisor -Dstorm.options= -Dstorm.home=/home/csishyduser/apache-storm-2.1.0 -Dstorm.log.dir=/home/csish
duser/apache-storm-2.1.0/logs -Djava.library.path=/usr/lib/jvm -Dstorm.conf.file= -cp /home/csishyduser/apache-storm-2.1.0/*:/home/csishyduse
/apache-storm-2.1.0/lib/*:/home/csishyduser/apache-storm-2.1.0/extlib/*:/home/csishyduser/apache-storm-2.1.0/extlib-daemon/*:/home/csishyduse
/apache-storm-2.1.0/conf -Xmx256m -Djava.deserialization.disabled=true -Dlogfile.name=supervisor.log -Dlog4j.configurationFile=/home/csishydu
er/apache-storm-2.1.0/log4j2/cluster.xml org.apache.storm.daemon.supervisor.Supervisor
```

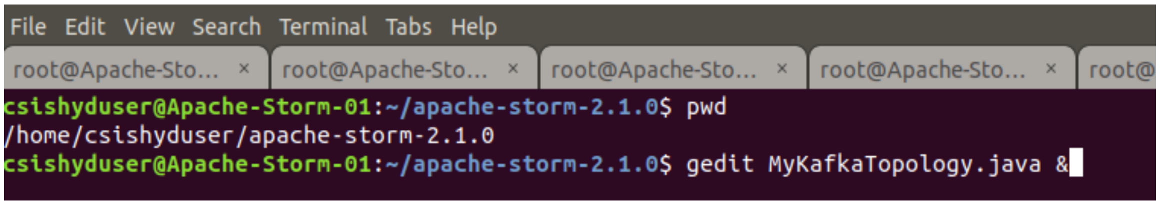
W. Open another terminal. Look at the current working directory using the "pwd" command. Then change the directory to the Storms directory.

>>> pwd

>>> cd apache-storm-2.1.0/

X. Open up gedit editor for writing the Java code.

>>> gedit MyKafkaTopology.java&



```
csishyduser@Apache-Storm-01:~/apache-storm-2.1.0$ pwd
/home/csishyduser/apache-storm-2.1.0
csishyduser@Apache-Storm-01:~/apache-storm-2.1.0$ gedit MyKafkaTopology.java &
```

Y. Copy paste the content of attached MyKafkaTopology.java file into the file opened in the geditor.

Z. Repeat the step X and Y for two other java files namely

- MyKafkaBolt.java

AA. Compile the MyKafkaTopology.java class which has the topology definition.

>>> javac -cp .:lib/* MyKafkaTopology.java

Note – Make sure following jars are present inside the "lib" directory of your storm installation. (Attached with this lab).

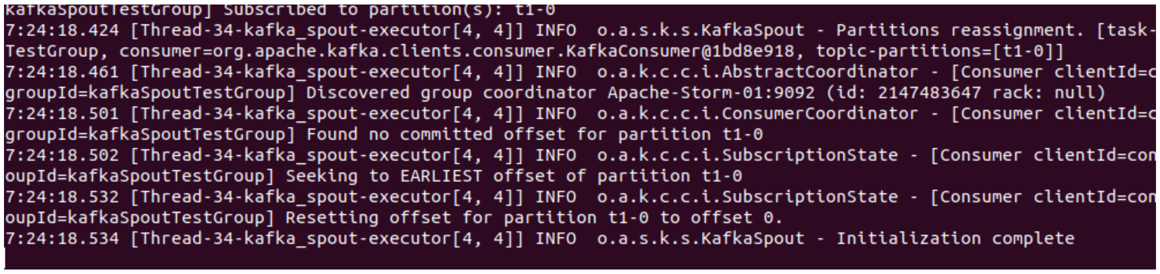
- kafka_2.10-0.10.2.2.jar
- kafka-clients-1.1.0.jar
- storm-kafka-1.0.1.jar
- storm-kafka-client-2.0.0.jar

BB. Run the MyKafkaTopology Storm application and observe the output.

>>> java -cp .:lib/*:/home/csishyduser/kafka_2.11-2.4.0/libs/* MyKafkaTopology

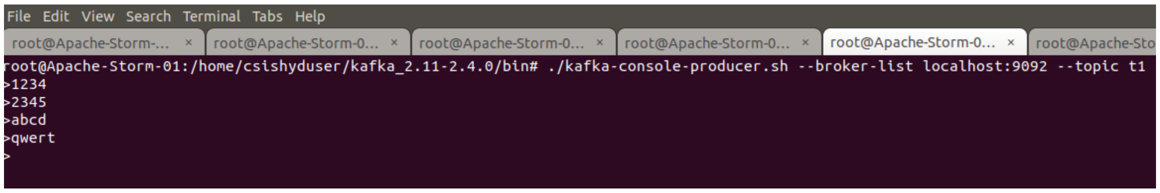
Note – The classpath also needs to include the "libs" directory of your kafka-installation.

CC. In the output you must be seeing the lines shown below which shows that program is executing.



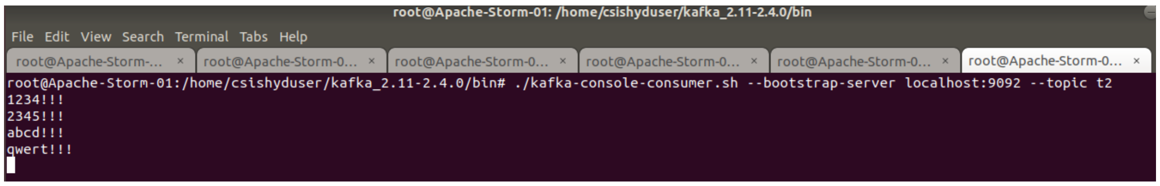
```
kafkaSpoutTestGroup] Subscribed to partition(s): t1-0
7:24:18.424 [Thread-34-kafka_spout-executor[4, 4]] INFO  o.a.s.k.s.KafkaSpout - Partitions reassignment. [task-
TestGroup, consumer=org.apache.kafka.clients.consumer.KafkaConsumer@1bd8e918, topic-partitions=[t1-0]]
7:24:18.461 [Thread-34-kafka_spout-executor[4, 4]] INFO  o.a.k.c.c.i.AbstractCoordinator - [Consumer clientId=c
groupId=kafkaSpoutTestGroup] Discovered group coordinator Apache-Storm-01:9092 (id: 2147483647 rack: null)
7:24:18.501 [Thread-34-kafka_spout-executor[4, 4]] INFO  o.a.k.c.c.i.ConsumerCoordinator - [Consumer clientId=c
groupId=kafkaSpoutTestGroup] Found no committed offset for partition t1-0
7:24:18.502 [Thread-34-kafka_spout-executor[4, 4]] INFO  o.a.k.c.c.i.SubscriptionState - [Consumer clientId=con
oupId=kafkaSpoutTestGroup] Seeking to EARLIEST offset of partition t1-0
7:24:18.532 [Thread-34-kafka_spout-executor[4, 4]] INFO  o.a.k.c.c.i.SubscriptionState - [Consumer clientId=con
oupId=kafkaSpoutTestGroup] Resetting offset for partition t1-0 to offset 0.
7:24:18.534 [Thread-34-kafka_spout-executor[4, 4]] INFO  o.a.s.k.s.KafkaSpout - Initialization complete
```

DD. Now enter the messages into the console producer screen for topic "t1"



```
root@Apache-Storm-01:/home/csishyduser/kafka_2.11-2.4.0/bin# ./kafka-console-producer.sh --broker-list localhost:9092 --topic t1
>1234
>2345
>abcd
>qwert
>
```

EE. You can see the enriched messages coming in the console consumer screen opened for topic "t2".



```
root@Apache-Storm-01:/home/csishyduser/kafka_2.11-2.4.0/bin# ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic t2
1234!!!
2345!!!
abcd!!!
qwert!!!
```

3. Outputs/Results:

Students should be able to write a Storm application

- To read the tuples / records from the Kafka topic through Spout
- To do data processing on the tuples through the Bolt logic
- To execute the Storm Topology on the local cluster
- To write enriched output back to Kafka Topic

4. Observations:

Students carefully needs to observe the steps followed for

- Setting up Kafka Cluster and starting it
- Creating the Kafka topic and writing messages into it
- Building the topology with Spout and Bolt and executing it on cluster
- Consuming the enriched outcomes from Kafka Topics

5. References:

a. Storm Documentation

b. Kafka Documentation

c. Data-flair training tutorial