Work Integrated Learning Programmes

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Computer Science & Information Systems

Real Time Analytics / Stream Processing & Analytics

Kafka Lab Sheet 2

Kafka Command line Operations

---

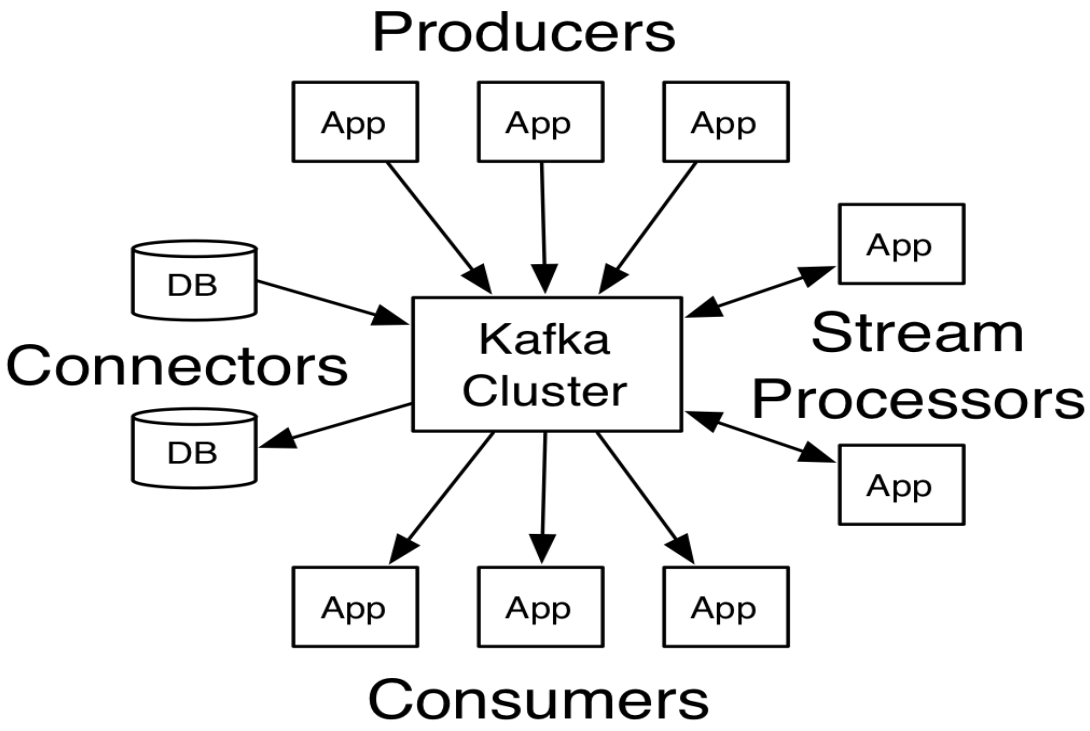
1. Objective:

Students should be able to

A. Get familiarity with the Kafka Cluster

B. Get hands-on experience with Kafka Command line utilities

Apache Kafka® is a distributed streaming platform. What exactly does that mean?



A streaming platform has three key capabilities:

- Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system.
- Store streams of records in a fault-tolerant durable way.
- Process streams of records as they occur.

Kafka is generally used for two broad classes of applications:

- Building real-time streaming data pipelines that reliably get data between systems or applications
- Building real-time streaming applications that transform or react to the streams of data

To understand how Kafka does these things, let's dive in and explore Kafka's capabilities from the bottom up.

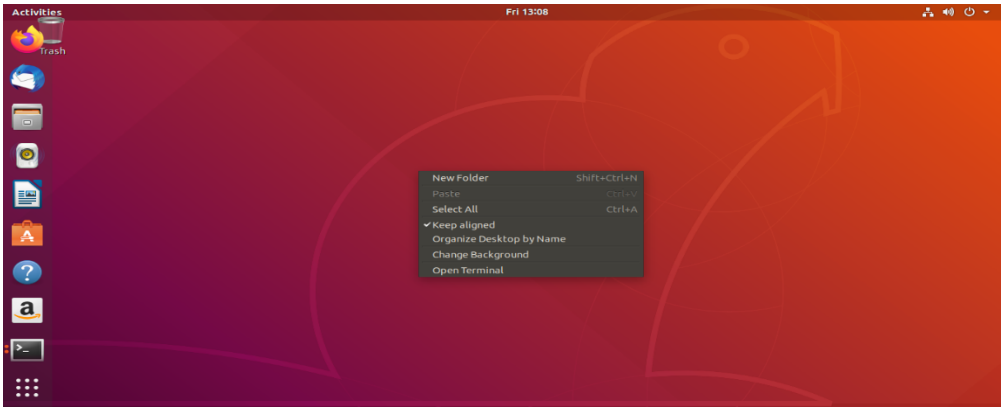
Kafka is run as a cluster on one or more servers that can span multiple datacenters. The Kafka cluster stores streams of records in categories called topics. Each record consists of a key, a value, and a timestamp. In Kafka the communication between the clients and the servers is done with a simple, high-performance, language agnostic TCP protocol. It also provide a Java client for Kafka, but clients are available in many languages.

2. Steps to be performed:

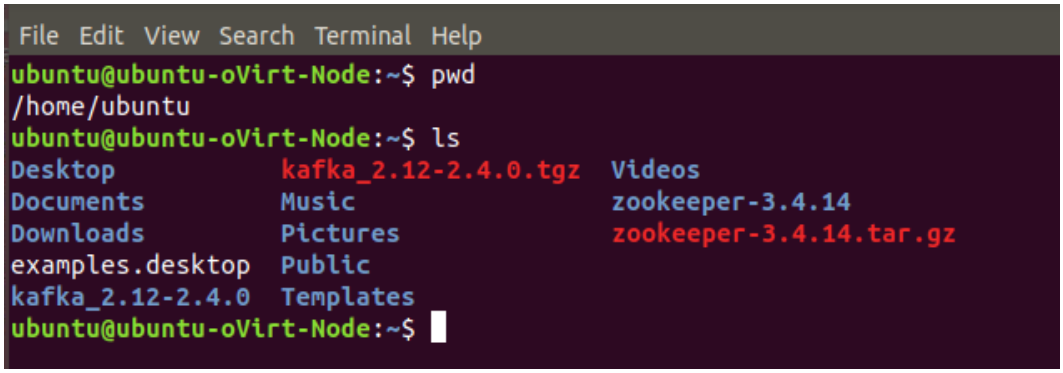
Note - It's assumed that student has made a slot reservation using the slot booking interface where Apache Spark framework was selected. The details of the Apache Spark systems to be used is received through an email. If not, please contact the administrators for the same.

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.

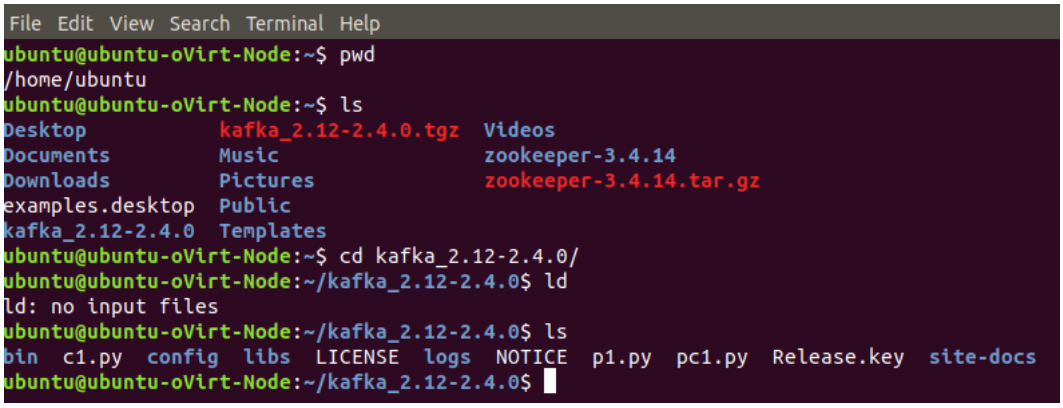
A. Open the terminal by right clicking on the desktop of the virtual machine.



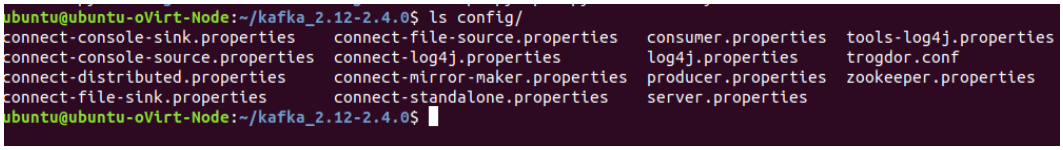
B. Look at the current directory and also file listings in it. It must have a Kafka installation directory present in it. Commands like pwd, ls can be used for it.

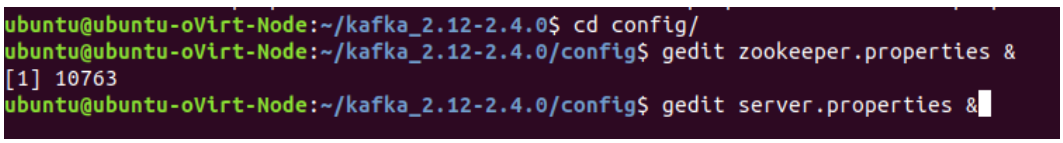


C. Change to the Kafka installation directory using the command and have a look at the files present in the directory.



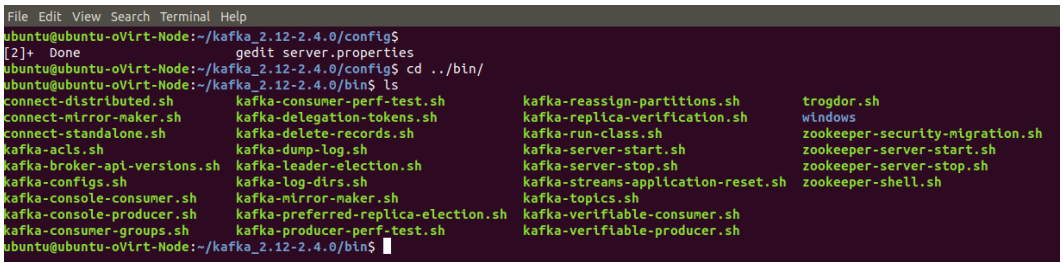
D. Change to the “config” directory present in the Kafka installation directory and have a look at the zookeeper and Kafka server properties files in it.

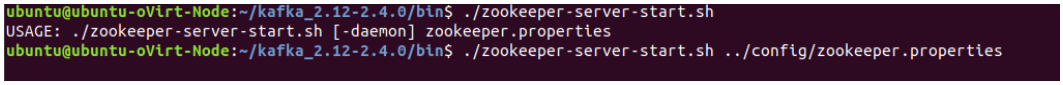




E. Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it. Will be using the following scripts to start and stop the Zookeeper ensemble

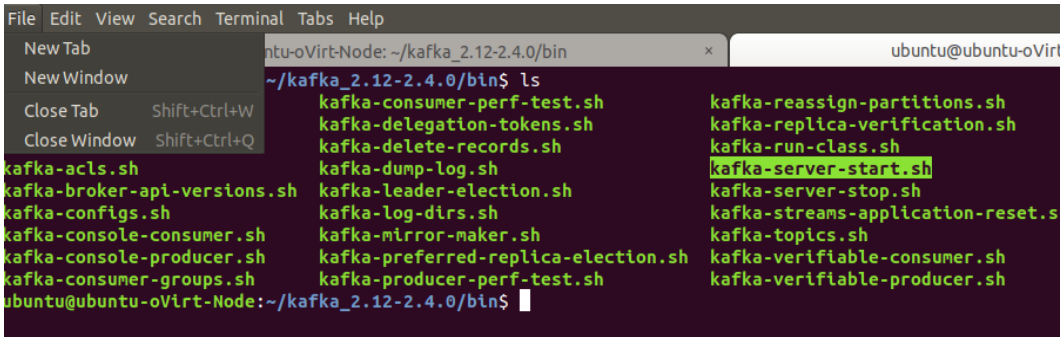
- zookeeper-server-start.sh
- zookeeper-server-stop.sh

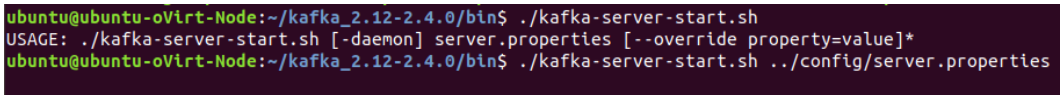




F. Open up another terminal and Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it. Will be using the following scripts to start and stop the Kafka server

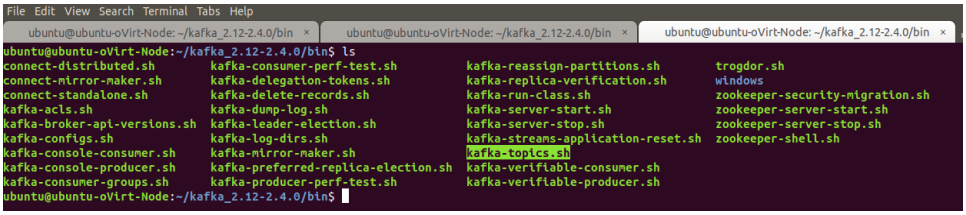
- kafka-server-start.sh
- kafka-server-stop.sh





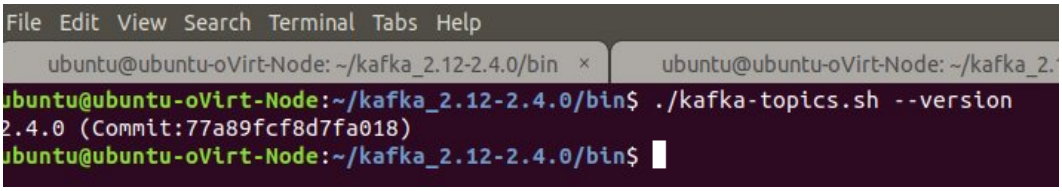
G. Open up another terminal and Change to the “bin” directory present in the Kafka installation directory and have a look at the script files present in it. Will be using the following script to deal with the Kafka topics.

- kafka-topics.sh



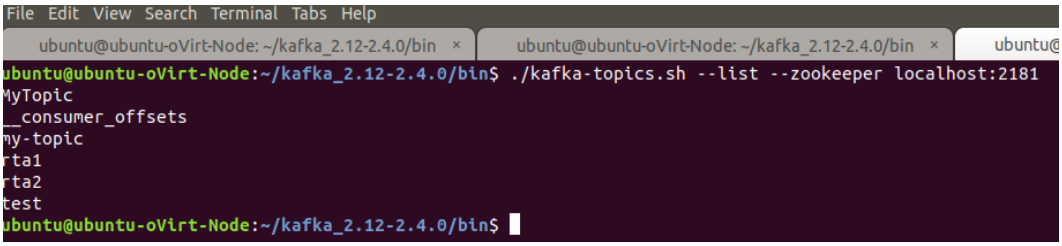
H. Use the –version option to see the version of Kafka.

- ./kafka-topics.sh –version



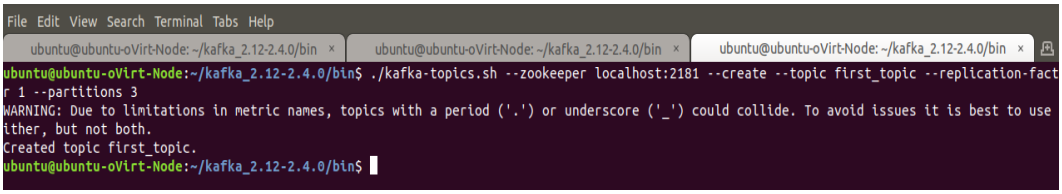
I. Use the –list option in it to list the topics present in the Kafka setup. Note – it will be empty for you as you don't have created any topics as such but if the Kafka setup is shared you may see the topics created in earlier usages of the system.

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



J. 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



K. List the Kafka topics again. Now the "first_topic" should appear in the topics list.

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

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

L. 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 first_topic --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$
```

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

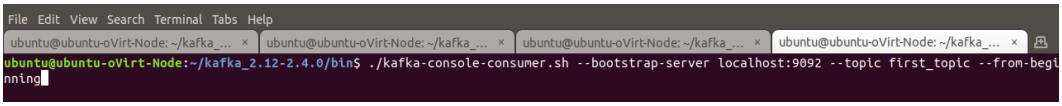
```
File  Edit  View  Search  Terminal  Tabs  Help
ubuntu@ubuntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin   ubuntu@ubuntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin   ubuntu@ubuntu-oVirt-Node: ~/kafka_2.12-2.4.0/bin
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$ ./kafka-console-producer.sh --topic first_topic --broker-list localhost:9092
>this is first message
```

N. 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.

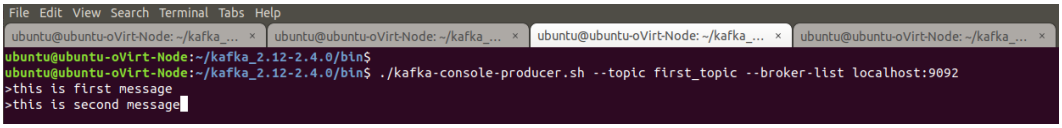
```
File  Edit  View  Search  Terminal  Tabs  Help
ubuntu@ubuntu-oVirt-Node: ~/kafka_...   ubuntu@ubuntu-oVirt-Node: ~/kafka_...   ubuntu@ubuntu-oVirt-Node: ~/kafka_...   ubuntu@ubuntu-oVirt-Node: ~/kafka_...
ubuntu@ubuntu-oVirt-Node:~/kafka_2.12-2.4.0/bin$ ls
connect-distributed.sh          kafka-consumer-perf-test.sh          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$
```

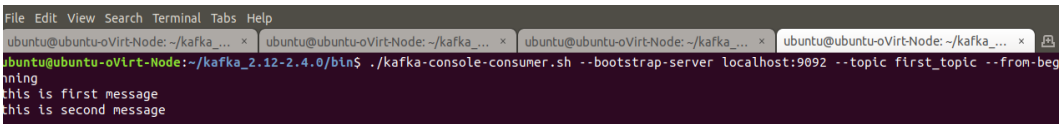
O. The Kafka distribution provides a command utility to see messages from the command line. It displays the messages in various modes. Kafka provides the utility kafka-console-consumer.sh which is located at /bin/kafka-console-producer.sh to receive messages from a topic on the command line.

- ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic first_topic --from-beginning



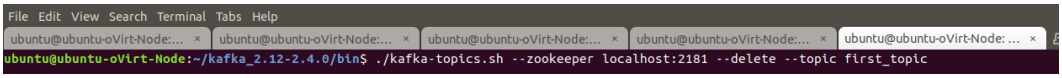
P. Enter the message in the producer terminal and switch to the consumer terminal to see if the message is received or not.





Q. 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



3. Outputs/Results:

Students should be able to use Kafka Command line for

- Starting / stopping the Zookeeper ensemble
- Starting / stopping the Kafka server
- Creating / Deleting the Kafka topics
- Listing the Kafka topics
- Feeding the Kafka topics with messages
- Reading the messages from Kafka topics

4. Observations:

Students carefully needs to observe the syntax and the options available with commands that helps to deal with the Kafka topics.

5. References:

a. Kafka Quickstart

b. Kafka website

c. KAFKA TUTORIAL: USING KAFKA FROM THE COMMAND LINE