ASSIGNMENT – 05

Introduction to Distributed Systems IS41243
Apache Kafka

By:
Sayanthiny.R
15APC2383

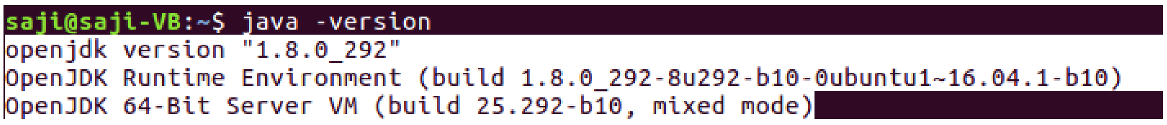
DEPARTMENT OF COMPUTING & INFORMATION SYSTEMS
FACULTY OF APPLIED SCIENCES
SABARAGAMUWA UNIVERSITY OF SRI LANKA

TABLE OF CONTENTS

Table of figures

Step 1 - Verifying Java Installation

Already installed java so no need to install java in my case. Use the following command to verify it.



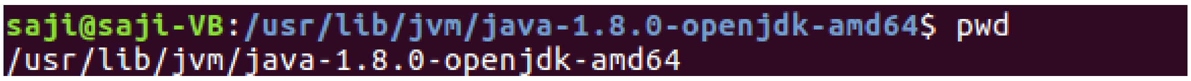
```
saji@saji-VB:~$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~16.04.1-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

Figure 1 Verifying Java Installation

Output shows that successfully installed because we can see the version of the installed java.

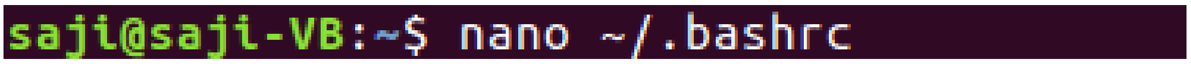
Step 1.1- Set path
Then to set path and JAVA_HOME variables, add the following commands to **~/.bashrc** file.

- Use **"pwd"** command to get java path.
- Use **nano ~/.bashrc** command to edit the bashrc file



```
saji@saji-VB:/usr/lib/jvm/java-1.8.0-openjdk-amd64$ pwd
/usr/lib/jvm/java-1.8.0-openjdk-amd64
```

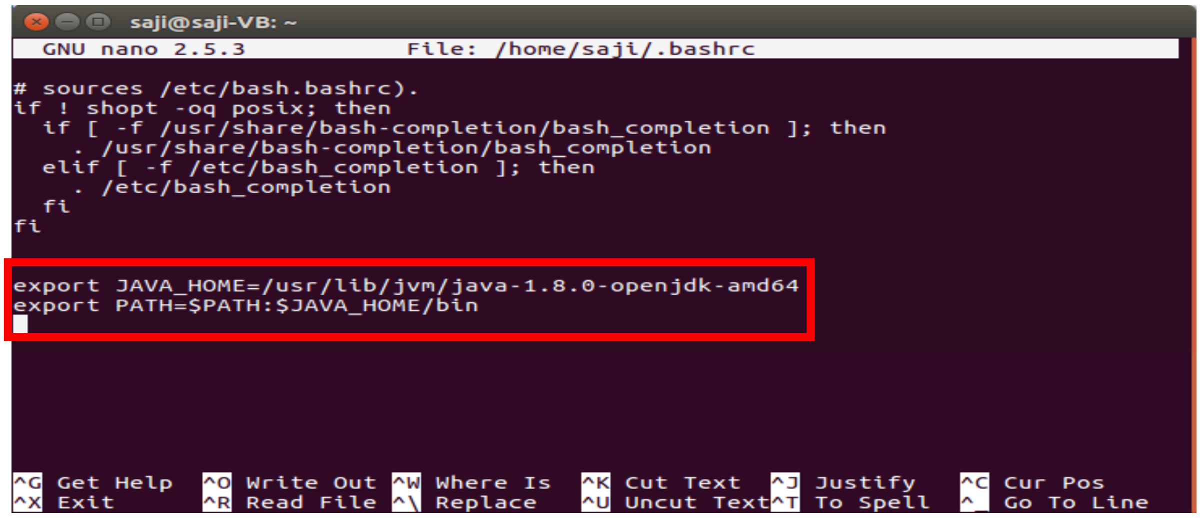
Figure 2 get java path



```
saji@saji-VB:~$ nano ~/.bashrc
```

Figure 3 edit the bashrc file

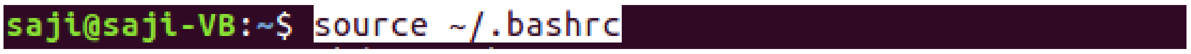
Edit the file as you see,



```
saji@saji-VB: ~
  GNU nano 2.5.3              File: /home/saji/.bashrc

# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin


^G Get Help   ^O Write Out ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Figure 4 Set path and JAVA_HOME variables

Use this command to apply changes into current running system.



```
saji@saji-VB:~$ source ~/.bashrc
```

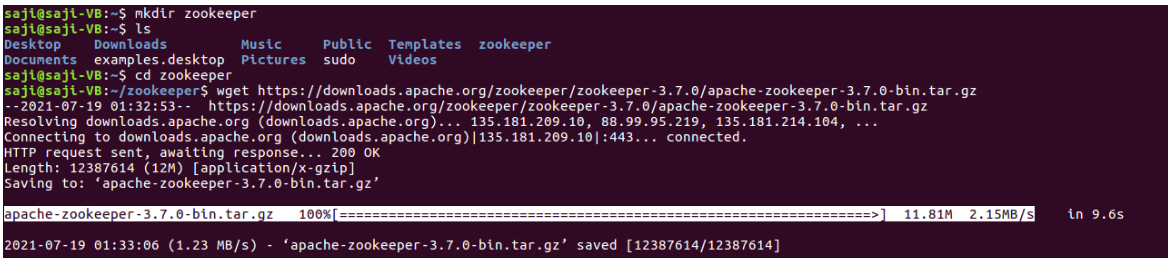
Figure 5 Apply changes into current running system

Step 2 - ZooKeeper Framework Installation

Step 2.1 - Download ZooKeeper

First make directory as Zookeeper then navigate to zookeeper directory then download zookeeper following these commands

```
wget https://downloads.apache.org/zookeeper/zookeeper-3.7.0/apache-zookeeper-3.7.0-bin.tar.gz
```

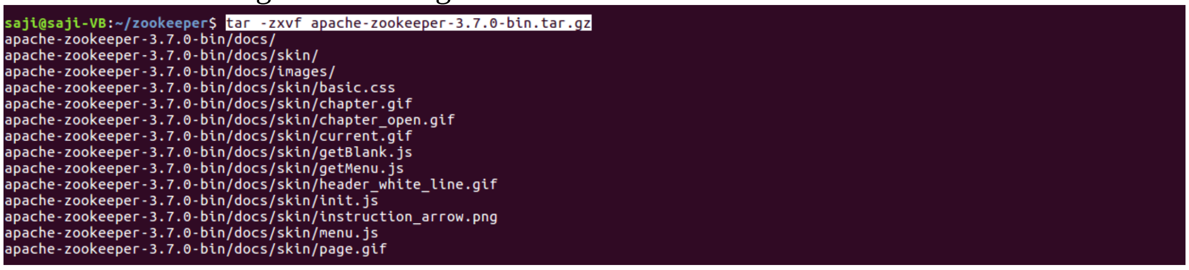


```
saji@saji-VB:~$ mkdir zookeeper
saji@saji-VB:~$ ls
Desktop    Downloads         Music     Public  Templates  zookeeper
Documents  examples.desktop  Pictures  sudo    Videos
saji@saji-VB:~$ cd zookeeper
saji@saji-VB:~/zookeeper$ wget https://downloads.apache.org/zookeeper/zookeeper-3.7.0/apache-zookeeper-3.7.0-bin.tar.gz
--2021-07-19 01:32:53--  https://downloads.apache.org/zookeeper/zookeeper-3.7.0/apache-zookeeper-3.7.0-bin.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 135.181.209.10, 88.99.95.219, 135.181.214.104, ...
Connecting to downloads.apache.org (downloads.apache.org)|135.181.209.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12387614 (12M) [application/x-gzip]
Saving to: ‘apache-zookeeper-3.7.0-bin.tar.gz’

apache-zookeeper-3.7.0-bin.tar.gz   100%[===================================================================>]  11.81M  2.15MB/s    in 9.6s

2021-07-19 01:33:06 (1.23 MB/s) - ‘apache-zookeeper-3.7.0-bin.tar.gz’ saved [12387614/12387614]
```

Figure 6 - Download ZooKeeper

Step 2.2 - Extract tar file

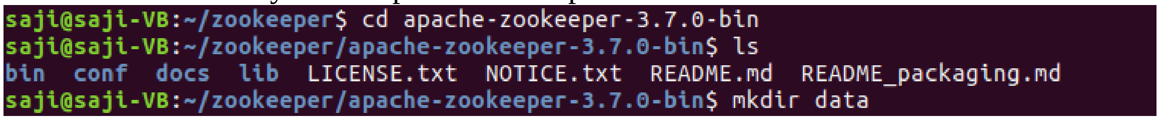
Extract the file using the following command.



```
saji@saji-VB:~/zookeeper$ tar -zxvf apache-zookeeper-3.7.0-bin.tar.gz
apache-zookeeper-3.7.0-bin/docs/
apache-zookeeper-3.7.0-bin/docs/skin/
apache-zookeeper-3.7.0-bin/docs/images/
apache-zookeeper-3.7.0-bin/docs/skin/basic.css
apache-zookeeper-3.7.0-bin/docs/skin/chapter.gif
apache-zookeeper-3.7.0-bin/docs/skin/chapter_open.gif
apache-zookeeper-3.7.0-bin/docs/skin/current.gif
apache-zookeeper-3.7.0-bin/docs/skin/getBlank.js
apache-zookeeper-3.7.0-bin/docs/skin/getMenu.js
apache-zookeeper-3.7.0-bin/docs/skin/header_white_line.gif
apache-zookeeper-3.7.0-bin/docs/skin/init.js
apache-zookeeper-3.7.0-bin/docs/skin/instruction_arrow.png
apache-zookeeper-3.7.0-bin/docs/skin/menu.js
apache-zookeeper-3.7.0-bin/docs/skin/page.gif
```

Figure 7 Extract tar file

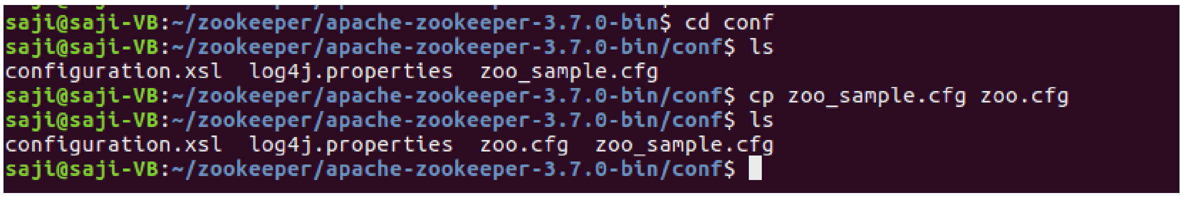
Create **data** directory inside apache-zookeeper



```
saji@saji-VB:~/zookeeper$ cd apache-zookeeper-3.7.0-bin
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin$ ls
bin  conf  docs  lib  LICENSE.txt  NOTICE.txt  README.md  README_packaging.md
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin$ mkdir data
```

Step 2.3 - Create Configuration File

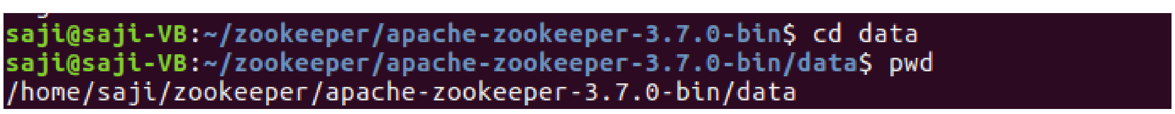
Inside conf directory the file has **zoo_sample.cfg** copy that file and make new file as **zoo.cfg**



```
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin$ cd conf
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin/conf$ ls
configuration.xsl  log4j.properties  zoo_sample.cfg
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin/conf$ cp zoo_sample.cfg zoo.cfg
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin/conf$ ls
configuration.xsl  log4j.properties  zoo.cfg  zoo_sample.cfg
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin/conf$
```

Figure 8 Create Configuration File

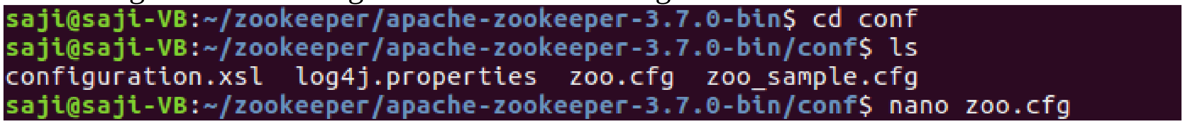
Then edit the zoo.cfg file using “**nano zoo.cfg** ” command. Get the data directory path then edit data dir_path path as you saw and set another parameters as default.



```
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin$ cd data
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin/data$ pwd
/home/saji/zookeeper/apache-zookeeper-3.7.0-bin/data
```

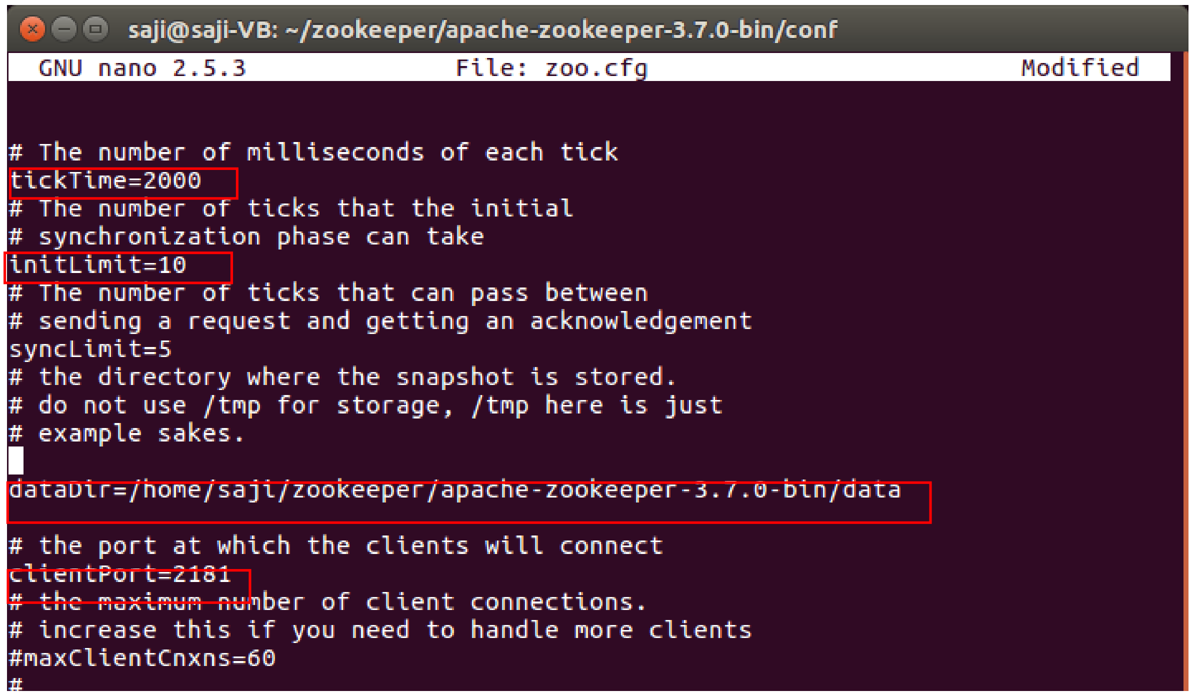
Figure 9 get the data directory path

Edit configuration file using command nano zoo.cfg



```
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin$ cd conf
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin/conf$ ls
configuration.xsl  log4j.properties  zoo.cfg  zoo_sample.cfg
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin/conf$ nano zoo.cfg
```

Figure 10 Edit configuration file



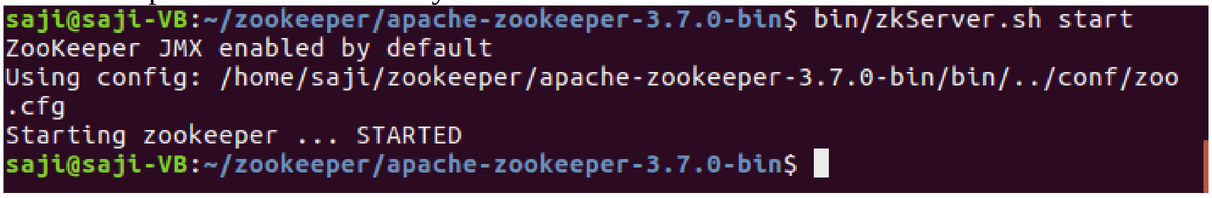
```
saji@saji-VB: ~/zookeeper/apache-zookeeper-3.7.0-bin/conf
  GNU nano 2.5.3              File: zoo.cfg                          Modified


# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.

dataDir=/home/saji/zookeeper/apache-zookeeper-3.7.0-bin/data

# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
```

Figure 11 Edit dataDIR in configuration file

Step 2.4 - Start ZooKeeper Server

Start zookeeper using this command $ bin/zkServer.sh start

After executing this command, we can get a response as shown below. This output shows that zookeeper started successfully



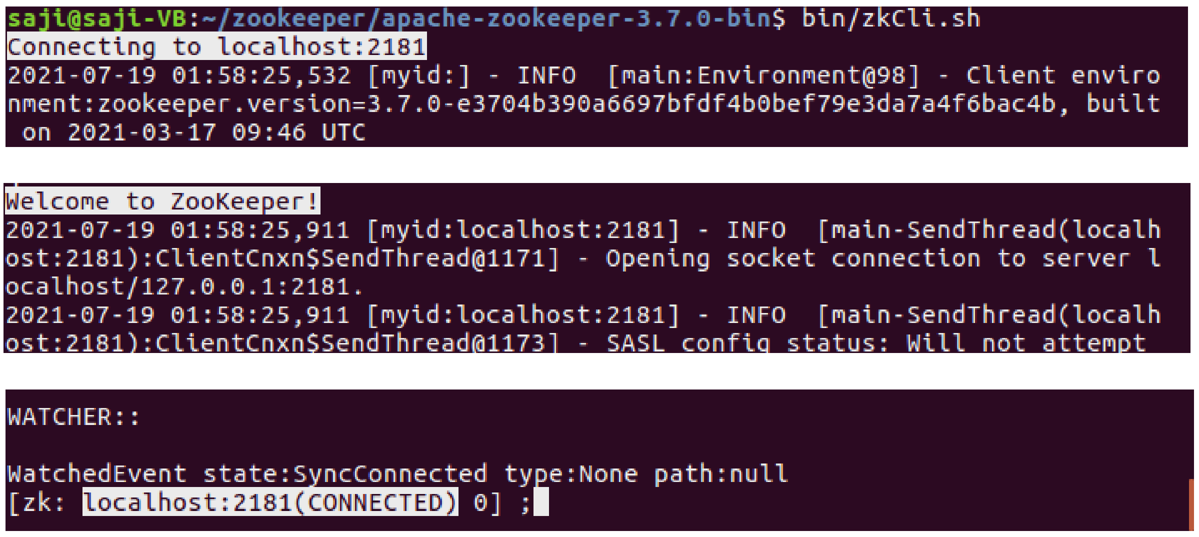
```
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin$ bin/zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /home/saji/zookeeper/apache-zookeeper-3.7.0-bin/bin/../conf/zoo
.cfg
Starting zookeeper ... STARTED
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin$
```

Figure 12 Start ZooKeeper Server

Step 2.5 - Start CLI

```
$ bin/zkCli..sh
```

After typing the above command, it will be connected to the zookeeper server and get the response as you see.



```
saji@saji-VB:~/zookeeper/apache-zookeeper-3.7.0-bin$ bin/zkCli.sh
Connecting to localhost:2181
2021-07-19 01:58:25,532 [myid:] - INFO  [main:Environment@98] - Client enviro
nment:zookeeper.version=3.7.0-e3704b390a6697bfdf4b0bef79e3da7a4f6bac4b, built
 on 2021-03-17 09:46 UTC
```

```
Welcome to ZooKeeper!
2021-07-19 01:58:25,911 [myid:localhost:2181] - INFO  [main-SendThread(localh
ost:2181):ClientCnxn$SendThread@1171] - Opening socket connection to server l
ocalhost/127.0.0.1:2181.
2021-07-19 01:58:25,911 [myid:localhost:2181] - INFO  [main-SendThread(localh
ost:2181):ClientCnxn$SendThread@1173] - SASL config status: Will not attempt
```

```
WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: localhost:2181(CONNECTED) 0] ;
```

Figure 13 connected to the zookeeper server

Step 2.6 - Stop Zookeeper Server

After connecting the server and performing all the operations, you can stop the zookeeper server with the following command.

```
$ bin/zkServer.sh stop
```

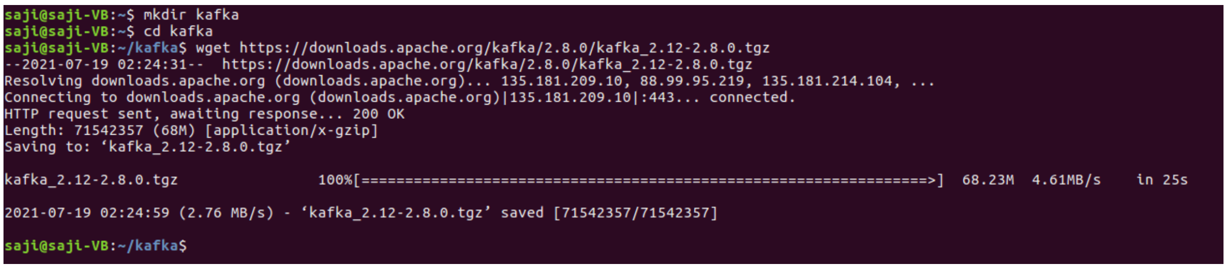
Step 3 - Apache Kafka Installation

Let us continue with the following steps to install Kafka on the machine

Step 3.1 - Download Kafka

Download kafka using the following command

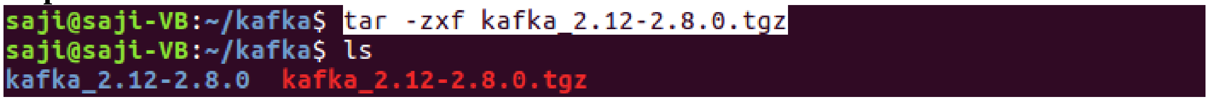
```
wget https://mirrors.estointernet.in/apache/kafka/2.8.0/kafka_2.13-2.8.0.tgz
```



```
saji@saji-VB:~$ mkdir kafka
saji@saji-VB:~$ cd kafka
saji@saji-VB:~/kafka$ wget https://downloads.apache.org/kafka/2.8.0/kafka_2.12-2.8.0.tgz
--2021-07-19 02:24:31--  https://downloads.apache.org/kafka/2.8.0/kafka_2.12-2.8.0.tgz
Resolving downloads.apache.org (downloads.apache.org)... 135.181.209.10, 88.99.95.219, 135.181.214.104, ...
Connecting to downloads.apache.org (downloads.apache.org)|135.181.209.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 71542357 (68M) [application/x-gzip]
Saving to: ‘kafka_2.12-2.8.0.tgz’

kafka_2.12-2.8.0.tgz                100%[=================================================================>]  68.23M  4.61MB/s    in 25s

2021-07-19 02:24:59 (2.76 MB/s) - ‘kafka_2.12-2.8.0.tgz’ saved [71542357/71542357]

saji@saji-VB:~/kafka$
```

Figure 14 Download Kafka

Step 3.2 - Extract the tar file



```
saji@saji-VB:~/kafka$ tar -zxf kafka_2.12-2.8.0.tgz
saji@saji-VB:~/kafka$ ls
kafka_2.12-2.8.0  kafka_2.12-2.8.0.tgz
```

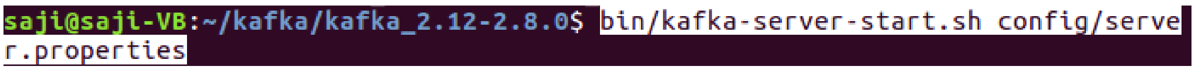
Figure 15 - Extract kafka tar file

Step 3.3 - Start Server

Start the server using this command

```
$ bin/kafka-server-start.sh config/server.properties
```

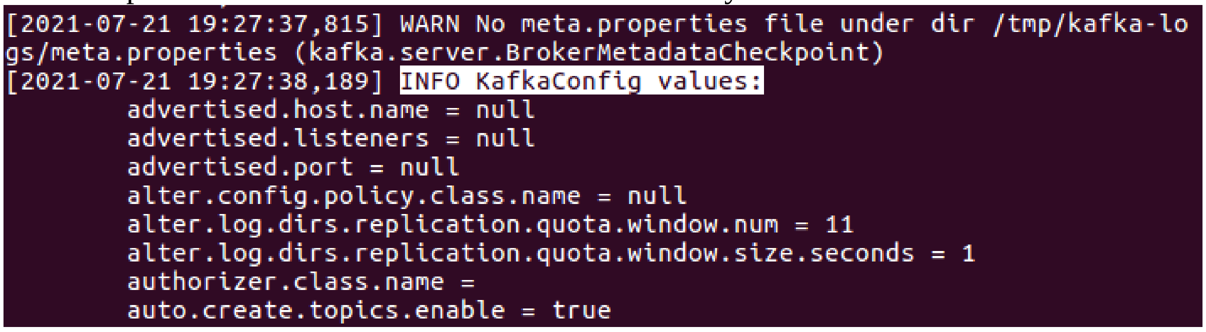
After the server starts, you would see the below response on your screen



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-server-start.sh config/serve
r.properties
```

Figure 16 kafka start command

Below response shows that kafka was started successfully.



```
[2021-07-21 19:27:37,815] WARN No meta.properties file under dir /tmp/kafka-lo
gs/meta.properties (kafka.server.BrokerMetadataCheckpoint)
[2021-07-21 19:27:38,189] INFO KafkaConfig values:
        advertised.host.name = null
        advertised.listeners = null
        advertised.port = null
        alter.config.policy.class.name = null
        alter.log.dirs.replication.quota.window.num = 11
        alter.log.dirs.replication.quota.window.size.seconds = 1
        authorizer.class.name =
        auto.create.topics.enable = true
```

Figure 17 kafka was started

Step 3.4 - Stop the Server

After performing all the operations, you can stop the server using the following command

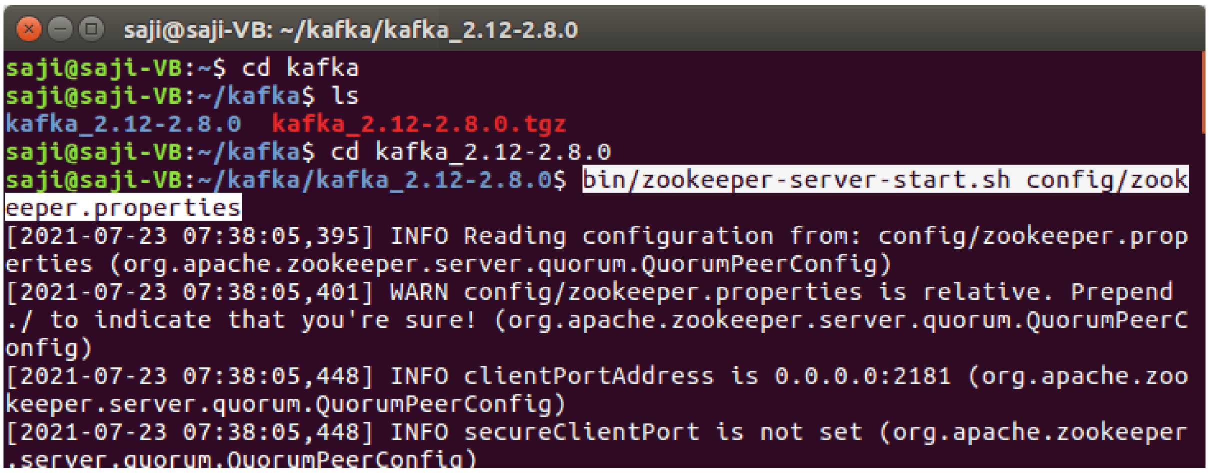
```
$ bin/kafka-server-stop.sh config/server.properties
```

Apache Kafka - Basic Operations

Prerequisites

These are the prerequisite to run Apache Kafka Basic operations

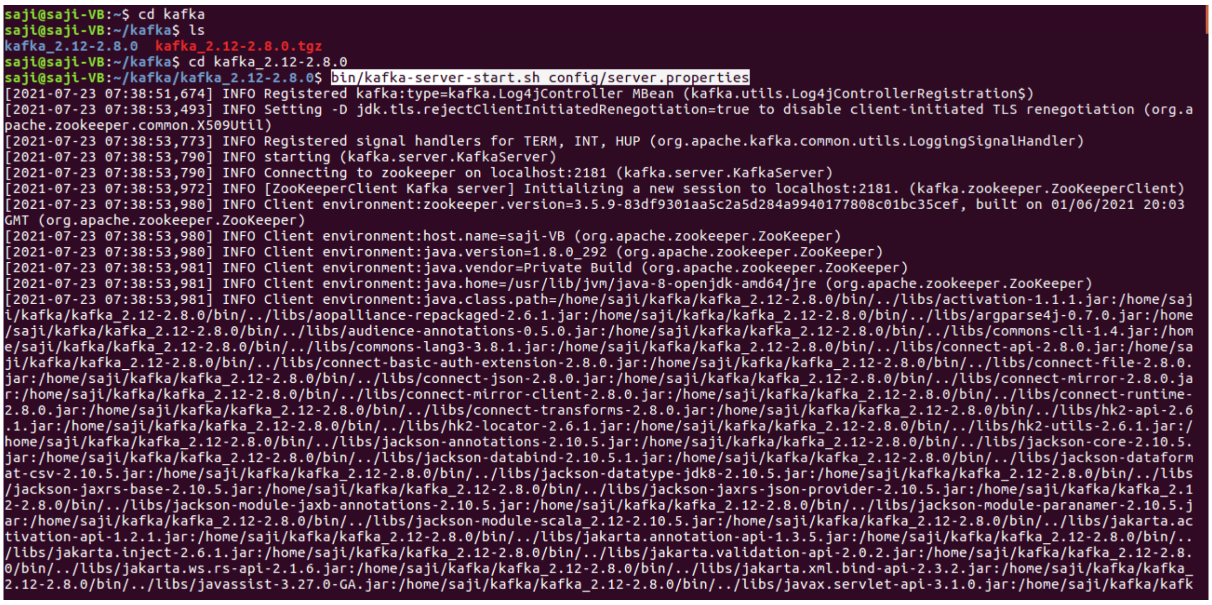
Start ZooKeeper



```
saji@saji-VB: ~/kafka/kafka_2.12-2.8.0
saji@saji-VB:~$ cd kafka
saji@saji-VB:~/kafka$ ls
kafka_2.12-2.8.0  kafka_2.12-2.8.0.tgz
saji@saji-VB:~/kafka$ cd kafka_2.12-2.8.0
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/zookeeper-server-start.sh config/zook
eeper.properties
[2021-07-23 07:38:05,395] INFO Reading configuration from: config/zookeeper.prop
erties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2021-07-23 07:38:05,401] WARN config/zookeeper.properties is relative. Prepend
./ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerC
onfig)
[2021-07-23 07:38:05,448] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zoo
keeper.server.quorum.QuorumPeerConfig)
[2021-07-23 07:38:05,448] INFO secureClientPort is not set (org.apache.zookeeper
.server.quorum.QuorumPeerConfig)
```

Figure 18 start the zookeeper

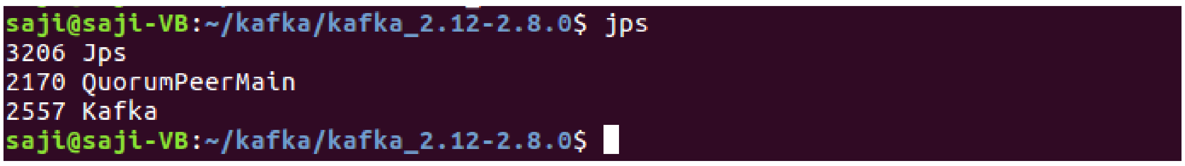
Start Kafka Broker



```
saji@saji-VB:~$ cd kafka
saji@saji-VB:~/kafka$ ls
kafka_2.12-2.8.0  kafka_2.12-2.8.0.tgz
saji@saji-VB:~/kafka$ cd kafka_2.12-2.8.0
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-server-start.sh config/server.properties
[2021-07-23 07:38:51,674] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2021-07-23 07:38:53,493] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.a
pache.zookeeper.common.X509Util)
[2021-07-23 07:38:53,773] INFO Registered signal handlers for TERM, INT, HUP (org.apache.kafka.common.utils.LoggingSignalHandler)
[2021-07-23 07:38:53,790] INFO starting (kafka.server.KafkaServer)
[2021-07-23 07:38:53,790] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
[2021-07-23 07:38:53,972] INFO [ZooKeeperClient Kafka server] Initializing a new session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2021-07-23 07:38:53,980] INFO Client environment:zookeeper.version=3.5.9-83df9301aa5c2a5d284a9940177808c01bc35cef, built on 01/06/2021 20:03
GMT (org.apache.zookeeper.ZooKeeper)
[2021-07-23 07:38:53,980] INFO Client environment:host.name=saji-VB (org.apache.zookeeper.ZooKeeper)
[2021-07-23 07:38:53,980] INFO Client environment:java.version=1.8.0_292 (org.apache.zookeeper.ZooKeeper)
[2021-07-23 07:38:53,981] INFO Client environment:java.vendor=Private Build (org.apache.zookeeper.ZooKeeper)
[2021-07-23 07:38:53,981] INFO Client environment:java.home=/usr/lib/jvm/java-8-openjdk-amd64/jre (org.apache.zookeeper.ZooKeeper)
[2021-07-23 07:38:53,981] INFO Client environment:java.class.path=/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/activation-1.1.1.jar:/home/saj
i/kafka/kafka_2.12-2.8.0/bin/../libs/aopalliance-repackaged-2.6.1.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/argparse4j-0.7.0.jar:/home
/saji/kafka/kafka_2.12-2.8.0/bin/../libs/audience-annotations-0.5.0.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/commons-cli-1.4.jar:/hom
e/saji/kafka/kafka_2.12-2.8.0/bin/../libs/commons-lang3-3.8.1.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/connect-api-2.8.0.jar:/home/sa
ji/kafka/kafka_2.12-2.8.0/bin/../libs/connect-basic-auth-extension-2.8.0.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/connect-file-2.8.0.
jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/connect-json-2.8.0.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/connect-mirror-2.8.0.ja
r:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/connect-mirror-client-2.8.0.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/connect-runtime-
2.8.0.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/connect-transforms-2.8.0.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/hk2-api-2.6
.1.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/hk2-locator-2.6.1.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/hk2-utils-2.6.1.jar:/
home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jackson-annotations-2.10.5.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jackson-core-2.10.5.
jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jackson-databind-2.10.5.1.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jackson-dataform
at-csv-2.10.5.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jackson-datatype-jdk8-2.10.5.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs
/jackson-jaxrs-base-2.10.5.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jackson-jaxrs-json-provider-2.10.5.jar:/home/saji/kafka/kafka_2.1
2-2.8.0/bin/../libs/jackson-module-jaxb-annotations-2.10.5.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jackson-module-paranamer-2.10.5.j
ar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jackson-module-scala_2.12-2.10.5.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jakarta.ac
tivation-api-1.2.1.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jakarta.annotation-api-1.3.5.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/..
/libs/jakarta.inject-2.6.1.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jakarta.validation-api-2.0.2.jar:/home/saji/kafka/kafka_2.12-2.8.
0/bin/../libs/jakarta.ws.rs-api-2.1.6.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/jakarta.xml.bind-api-2.3.2.jar:/home/saji/kafka/kafka_
2.12-2.8.0/bin/../libs/javassist-3.27.0-GA.jar:/home/saji/kafka/kafka_2.12-2.8.0/bin/../libs/javax.servlet-api-3.1.0.jar:/home/saji/kafka/kafk
```

Figure 19start Kafka Broker

After starting Kafka Broker, type the command "jps" on ZooKeeper terminal and two daemons running on the terminal where QuorumPeerMain is ZooKeeper daemon and another one is Kafka daemon



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ jps
3206 Jps
2170 QuorumPeerMain
2557 Kafka
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ 
```

Figure 20 jps

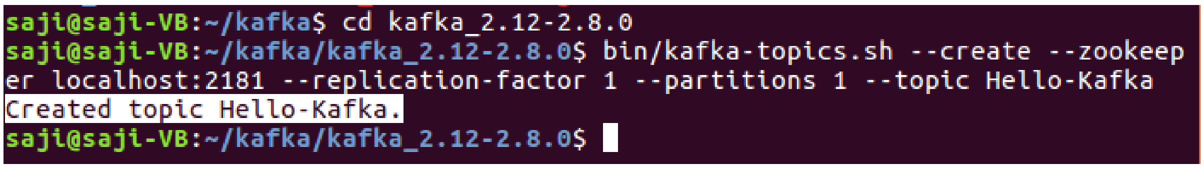
Single Node-Single Broker Configuration

In this configuration have a single ZooKeeper and broker id instance.Following are the steps to configure it.

1. Creating a Kafka Topic

use "**kafka-topics.sh**" to create topics on the server. After type the below command in terminal get the output to created topic **Hello-Kafka.**

```
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic Hello-Kafka
```



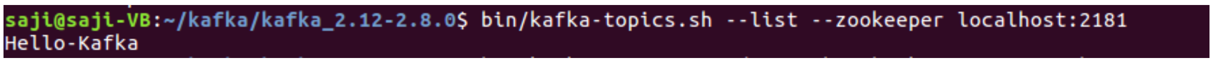
```
saji@saji-VB:~/kafka$ cd kafka_2.12-2.8.0
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-topics.sh --create --zookeep
er localhost:2181 --replication-factor 1 --partitions 1 --topic Hello-Kafka
Created topic Hello-Kafka.
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$
```

Figure 21 create topic in Single Node-Single Broker Configuration

2. List of Topics

To get a list of topics in Kafka server.

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



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-topics.sh --list --zookeeper localhost:2181
Hello-Kafka
```

Figure 22 List of topics

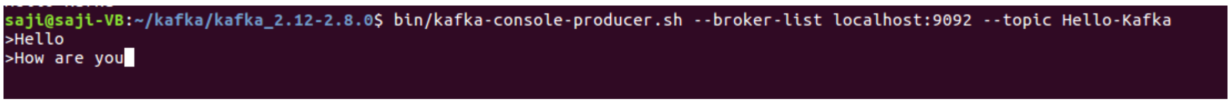
After type the following command, we can see the output **Hello-Kafka.**
Since we have created a topic, it will list out Hello-Kafka only. Suppose, if you create more than one topics, you will get the topic names in the output.

3. Start producer to send messages

In this case we only have one broker. The **Config/server.properties** file contains broker port id, since we know our broker is listening on **port 9092,** so you can specify it directly.

```
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic Hello-Kafka
```

The producer will wait on input and publishes to the Kafka cluster. When type a few lines of messages in the terminal as shown below.



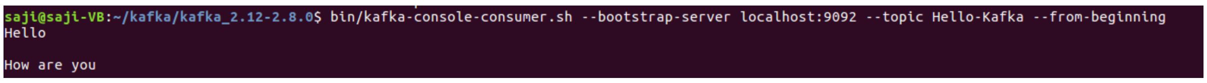
```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic Hello-Kafka
>Hello
>How are you
```

Figure 23 producer send messages

4. Start consumer to receive messages

type the below command for consuming messages.

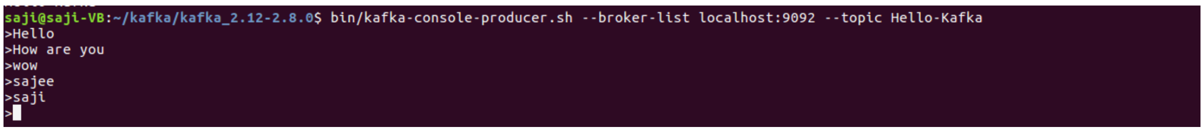
```
bin/kafka-console-consumer.sh –-bootstrap-server localhost:9092 --topic Hello-Kafka –from
beginning
```



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic Hello-Kafka --from-beginning
Hello

How are you
```

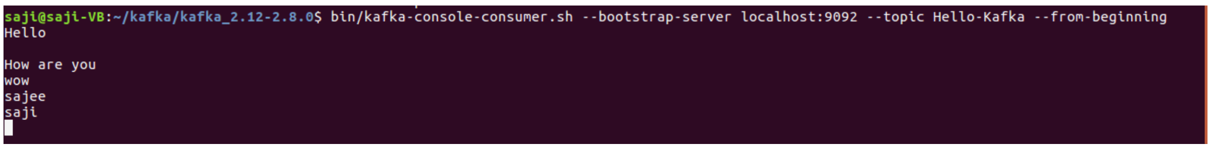
Figure 24 Consumer receive messages

Finally, we should able to enter messages from the producer's terminal and see them appearing in the consumer's terminal.



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic Hello-Kafka
>Hello
>How are you
>wow
>sajee
>saji
>
```

Figure 25 producer send messages



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic Hello-Kafka --from-beginning
Hello

How are you
wow
sajee
saji
```

Figure 26 Consumer receive messages

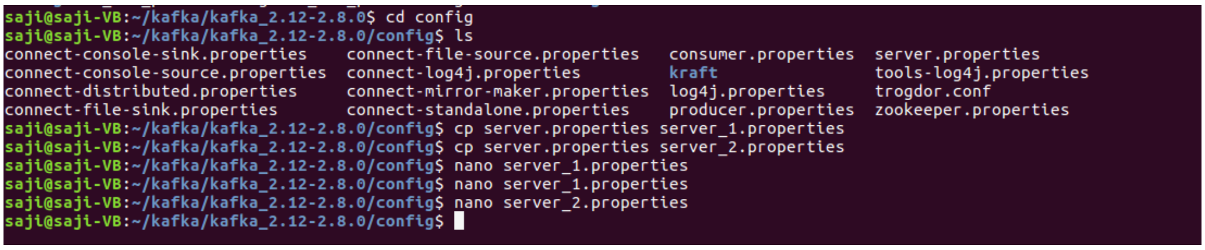
Single node-multiple brokers configuration

Start ZooKeeper server.

Before moving on to the multiple brokers cluster setup, first start your ZooKeeper server.

1.Create Multiple Kafka Brokers

We have one Kafka broker instance already in config/server.properties. Now we need multiple broker instances, so copy the existing server.properties file into two new config files and rename it as server_1.properties and server_2.properties.

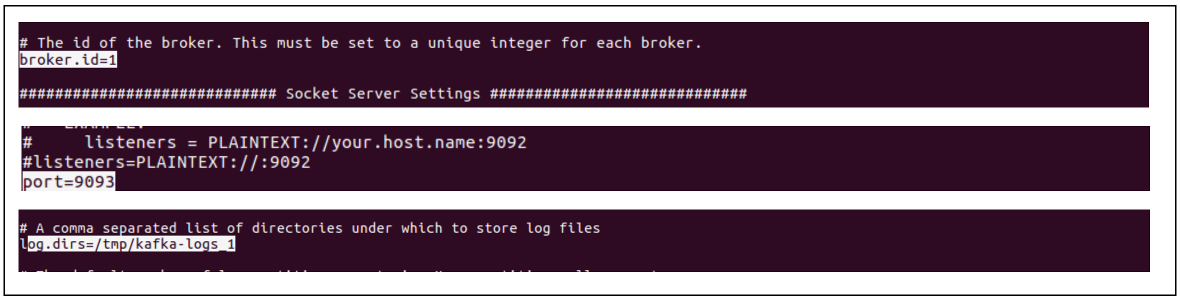


```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ cd config
saji@saji-VB:~/kafka/kafka_2.12-2.8.0/config$ ls
connect-console-sink.properties    connect-file-source.properties   consumer.properties  server.properties
connect-console-source.properties  connect-log4j.properties         kraft                tools-log4j.properties
connect-distributed.properties     connect-mirror-maker.properties  log4j.properties     trogdor.conf
connect-file-sink.properties       connect-standalone.properties    producer.properties  zookeeper.properties
saji@saji-VB:~/kafka/kafka_2.12-2.8.0/config$ cp server.properties server_1.properties
saji@saji-VB:~/kafka/kafka_2.12-2.8.0/config$ cp server.properties server_2.properties
saji@saji-VB:~/kafka/kafka_2.12-2.8.0/config$ nano server_1.properties
saji@saji-VB:~/kafka/kafka_2.12-2.8.0/config$ nano server_1.properties
saji@saji-VB:~/kafka/kafka_2.12-2.8.0/config$ nano server_2.properties
saji@saji-VB:~/kafka/kafka_2.12-2.8.0/config$
```

Figure 27 Create Multiple Kafka Brokers

Then edit both new files and assign the following changes:

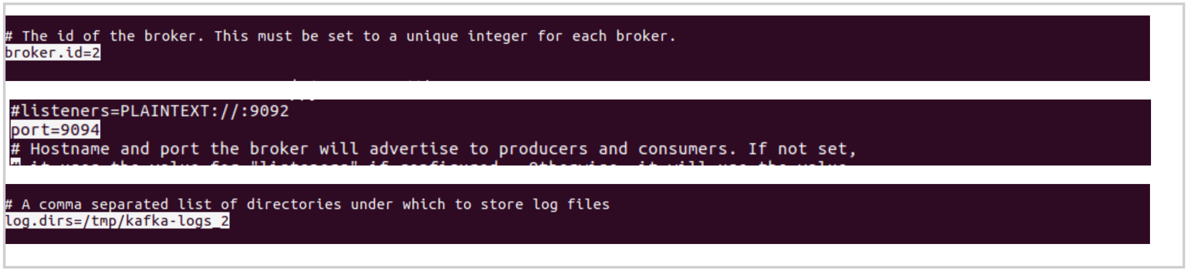
config/server_1.properties



```
# The id of the broker. This must be set to a unique integer for each broker.
broker.id=1

############################# Socket Server Settings #############################

#     listeners = PLAINTEXT://your.host.name:9092
#listeners=PLAINTEXT://:9092
port=9093

# A comma separated list of directories under which to store log files
log.dirs=/tmp/kafka-logs_1
```

Figure 28 config/server_1.properties

config/server_2.properties



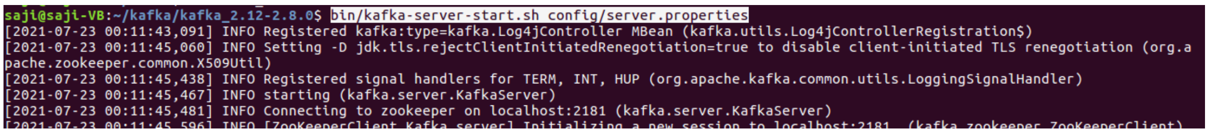
```
# The id of the broker. This must be set to a unique integer for each broker.
broker.id=2

#listeners=PLAINTEXT://:9092
port=9094
# Hostname and port the broker will advertise to producers and consumers. If not set,

# A comma separated list of directories under which to store log files
log.dirs=/tmp/kafka-logs_2
```

Figure 29 config/server_2.properties

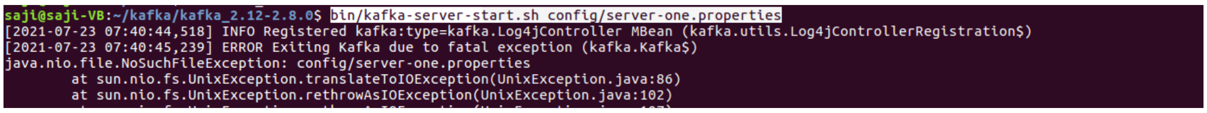
Start Multiple Brokers

After all the changes have been made on 3 servers then open 3 new terminals to start each broker one by one using this commands.

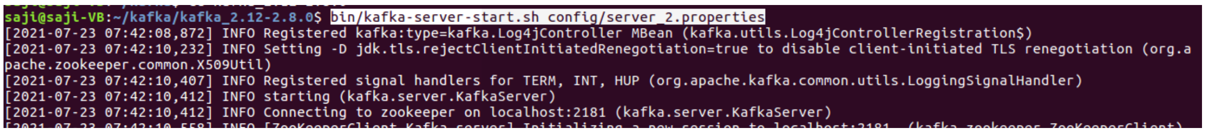
```
$ bin/kafka-server-start.sh config/server.properties
```



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-server-start.sh config/server.properties
[2021-07-23 00:11:43,091] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2021-07-23 00:11:45,060] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.a
pache.zookeeper.common.X509Util)
[2021-07-23 00:11:45,438] INFO Registered signal handlers for TERM, INT, HUP (org.apache.kafka.common.utils.LoggingSignalHandler)
[2021-07-23 00:11:45,467] INFO starting (kafka.server.KafkaServer)
[2021-07-23 00:11:45,481] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
```



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-server-start.sh config/server-one.properties
[2021-07-23 07:40:44,518] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2021-07-23 07:40:45,239] ERROR Exiting Kafka due to fatal exception (kafka.Kafka$)
java.nio.file.NoSuchFileException: config/server-one.properties
        at sun.nio.fs.UnixException.translateToIOException(UnixException.java:86)
        at sun.nio.fs.UnixException.rethrowAsIOException(UnixException.java:102)
```



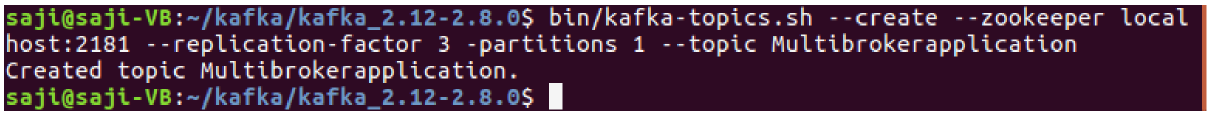
```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-server-start.sh config/server_2.properties
[2021-07-23 07:42:08,872] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2021-07-23 07:42:10,232] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.a
pache.zookeeper.common.X509Util)
[2021-07-23 07:42:10,407] INFO Registered signal handlers for TERM, INT, HUP (org.apache.kafka.common.utils.LoggingSignalHandler)
[2021-07-23 07:42:10,412] INFO starting (kafka.server.KafkaServer)
[2021-07-23 07:42:10,412] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
```

Figure 30 start multiple brokers

2. Creating a Topic

Let us assign the replication factor value as 3 for this topic because we have 3 different brokers running.

```
$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3 -partitions 1 --
topic Multibrokerapplication
```



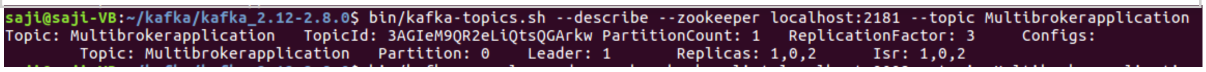
```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-topics.sh --create --zookeeper local
host:2181 --replication-factor 3 -partitions 1 --topic Multibrokerapplication
Created topic Multibrokerapplication.
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ 
```

Figure 31create topic in multibroker Application

3. Describe command

The Describe command is used to check which broker is listening on the current created topic as shown below.

```
$ bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic Multibrokerapplication
```



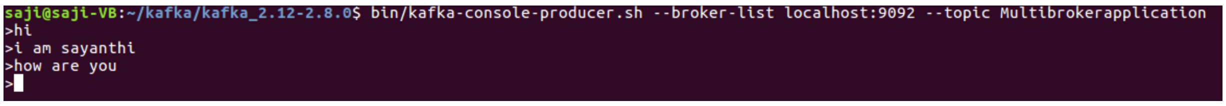
```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic Multibrokerapplication
Topic: Multibrokerapplication   TopicId: 3AGIeM9QR2eLiQtsQGArkw PartitionCount: 1   ReplicationFactor: 3    Configs:
        Topic: Multibrokerapplication   Partition: 0    Leader: 1       Replicas: 1,0,2     Isr: 1,0,2
```

Figure 32 Describe command

4. Start producer to send messages

The producer will wait on input and publishes to the Kafka cluster. When type a few lines of messages in the terminal as shown below.

```
$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic Multibrokerapplication
```



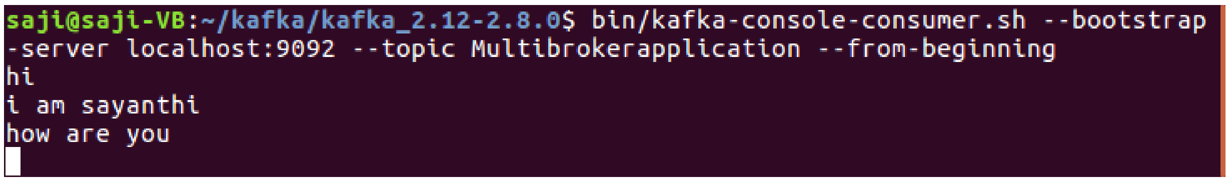
```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic Multibrokerapplication
>hi
>i am sayanthi
>how are you
>
```

Figure 33 producer send message

5. Start consumer to receive messages

Type the below command for consuming messages.

```
$ bin/kafka-console-consumer.sh –bootstrap-server localhost:9092 --topic Multibrokerapplication
--from-beginning
```



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-console-consumer.sh --bootstrap
-server localhost:9092 --topic Multibrokerapplication --from-beginning
hi
i am sayanthi
how are you
```

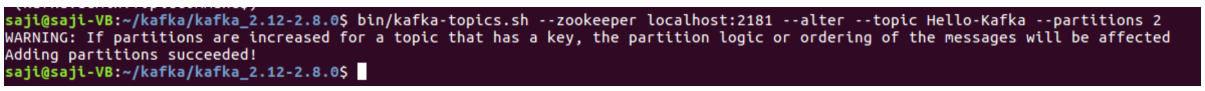
Figure 34 consumer receive messages

Basic topic operations

Modifying a Topic

modify a created topic using the following command

```
$ bin/kafka-topics.sh --zookeeper localhost:2181 --alter --topic Hello-Kafka --partitions 2
```



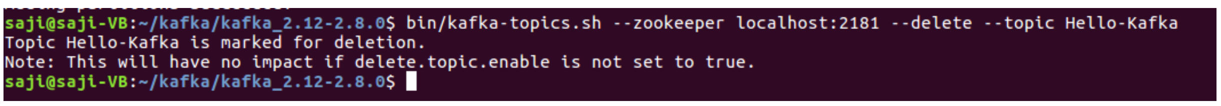
```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-topics.sh --zookeeper localhost:2181 --alter --topic Hello-Kafka --partitions 2
WARNING: If partitions are increased for a topic that has a key, the partition logic or ordering of the messages will be affected
Adding partitions succeeded!
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$
```

Figure 35 Modifying a topic

Delete a topic

To delete a topic, you can use the following command

```
$ bin/kafka-topics.sh --zookeeper localhost:2181 --delete --topic Hello-kafka
```



```
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$ bin/kafka-topics.sh --zookeeper localhost:2181 --delete --topic Hello-Kafka
Topic Hello-Kafka is marked for deletion.
Note: This will have no impact if delete.topic.enable is not set to true.
saji@saji-VB:~/kafka/kafka_2.12-2.8.0$
```

Figure 36 delete a topic

Apache Kafka - Simple Producer Example

These are the following steps to Simple Producer Example.

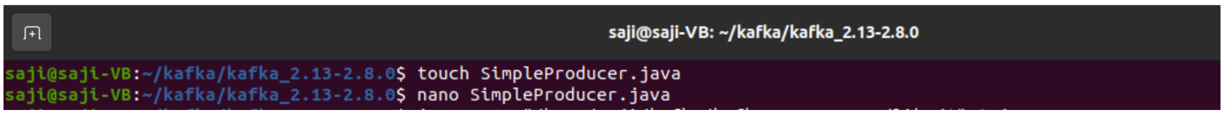
1.Create SimpleProducer.java file

```
$ touch SimpleProducer.java
```

2. Edit the SimpleProducer.java

Edit the SimpleProducer.java file using this command

```
$ nano SimpleProducer.java
```



```
saji@saji-VB: ~/kafka/kafka_2.13-2.8.0
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ touch SimpleProducer.java
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ nano SimpleProducer.java
```

Figure 37 Create SimpleProducer.java file

This is a **simpleProducer.java** file

```
//import util.properties packages
import java.util.Properties;

//import simple producer packages
import org.apache.kafka.clients.producer.Producer;

//import KafkaProducer packages
import org.apache.kafka.clients.producer.KafkaProducer;

//import ProducerRecord packages
import org.apache.kafka.clients.producer.ProducerRecord;

//Create java class named "SimpleProducer"
public class SimpleProducer {

  public static void main(String[] args) throws Exception{

    // Check arguments length value
    if(args.length == 0){
      System.out.println("Hello-Sayanthiny");
      return;
    }

    //Assign topicName to string variable
    String topicName = args[0].toString();

    // create instance for properties to access producer configs
    Properties props = new Properties();
```

```
      //Assign localhost id
      props.put("bootstrap.servers", "localhost:9092");


//Set acknowledgements for producer requests.
      props.put("acks", "all");

      //If the request fails, the producer can automatically retry,
      props.put("retries", 0);

      //Specify buffer size in config
      props.put("batch.size", 16384);

      //Reduce the no of requests less than 0
      props.put("linger.ms", 1);

      //The buffer.memory controls the total amount of memory available to the producer for
buffering.
      props.put("buffer.memory", 33554432);

      props.put("key.serializer",
         "org.apache.kafka.common.serialization.StringSerializer");

      props.put("value.serializer",
         "org.apache.kafka.common.serialization.StringSerializer");

      Producer<String, String> producer = new KafkaProducer
         <String, String>(props);

      for(int i = 0; i < 10; i++)
         producer.send(new ProducerRecord<String, String>(topicName,
            Integer.toString(i), Integer.toString(i)));
               System.out.println("Message sent successfully");
               producer.close();
   }
}
```

3. Compilation

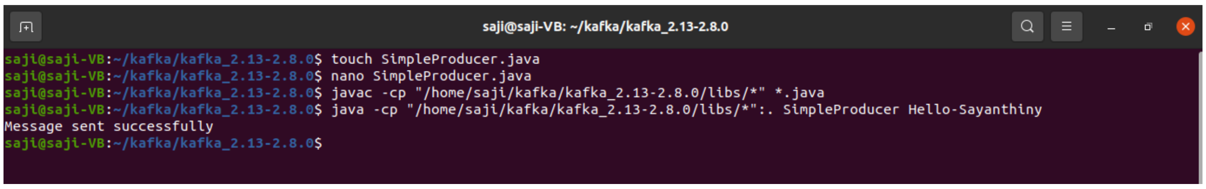
The application can be compiled using the following command.

```
$ javac -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*" *.java
```

4. Execution

The application can be executed using the following command.

```
$ java -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*":. SimpleProducer Hello-Sayanthiny
```

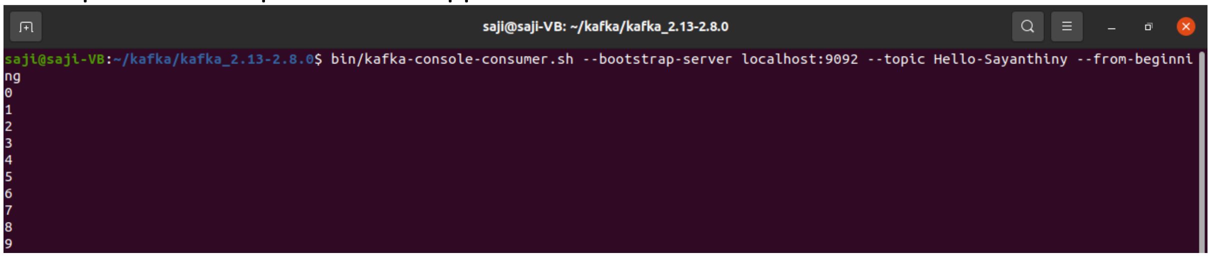


saji@saji-VB: ~/kafka/kafka_2.13-2.8.0

```
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ touch SimpleProducer.java
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ nano SimpleProducer.java
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ javac -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*" *.java
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ java -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*":. SimpleProducer Hello-Sayanthiny
Message sent successfully
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$
```

Figure 38 Application Compilation & Execution

5. Output of the Simple Producer Application



saji@saji-VB: ~/kafka/kafka_2.13-2.8.0

```
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic Hello-Sayanthiny --from-beginni
ng
0
1
2
3
4
5
6
7
8
9
```

Figure 39 Output

Simple Consumer Example

These are the following steps to Simple Producer Example.

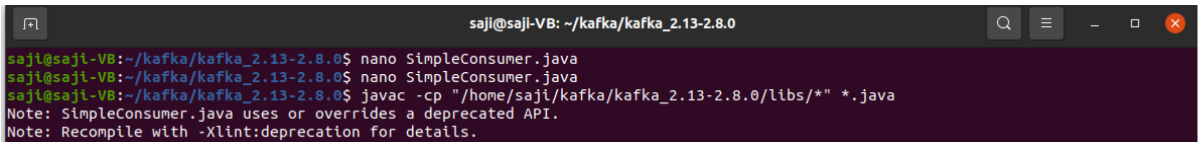
1.Create SimpleConsumer.java file

```
$ touch SimpleConsumer.java
```

2. Edit the SimpleProducer.java

Edit the SimpleProducer.java file using

```
$ nano SimpleConsumer.java
```



```
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ nano SimpleConsumer.java
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ nano SimpleConsumer.java
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ javac -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*" *.java
Note: SimpleConsumer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

Figure 40 Create SimpleConsumer.java file

3. SimpleConsumer.java file

```
SimpleConsumer.java  file

import java.util.Properties;
import java.util.Arrays;
import org.apache.kafka.clients.consumer.KafkaConsumer;
import org.apache.kafka.clients.consumer.ConsumerRecords;
import org.apache.kafka.clients.consumer.ConsumerRecord;

public class SimpleConsumer {
   public static void main(String[] args) throws Exception {
      if(args.length == 0){
         System.out.println("Hello-Sayanthiny");
         return;
      }
      //Kafka consumer configuration settings
      String topicName = args[0].toString();
      Properties props = new Properties();

      props.put("bootstrap.servers", "localhost:9092");
      props.put("group.id", "test");
      props.put("enable.auto.commit", "true");
      props.put("auto.commit.interval.ms", "1000");
      props.put("session.timeout.ms", "30000");
      props.put("key.deserializer",
         "org.apache.kafka.common.serialization.StringDeserializer");
      props.put("value.deserializer",
         "org.apache.kafka.common.serialization.StringDeserializer");
```

```
    KafkaConsumer<String, String> consumer = new KafkaConsumer
      <String, String>(props);


    //Kafka Consumer subscribes list of topics here.
    consumer.subscribe(Arrays.asList(topicName));

    //print the topic name
    System.out.println("Subscribed to topic " + topicName);
    int i = 0;

    while (true) {
      ConsumerRecords<String, String> records = consumer.poll(100);
      for (ConsumerRecord<String, String> record : records)

      // print the offset,key and value for the consumer records.
      System.out.printf("offset = %d, key = %s, value = %s\n",
        record.offset(), record.key(), record.value());
    }
  }
}
```

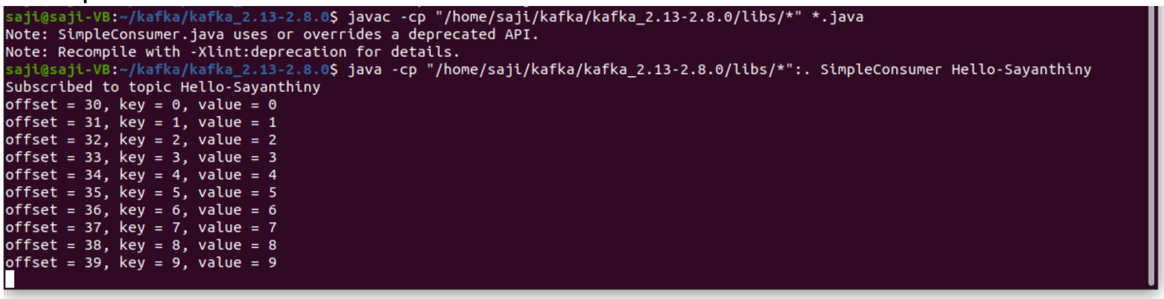
4. Compilation
The application can be compiled using the following command.

```
$ javac -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*" *.java
```

5. Execution
The application can be executed using the following command.

```
$ java -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*":. SimpleConsumer Hello-Sayanthiny
```

6. Output



```
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ javac -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*" *.java
Note: SimpleConsumer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
saji@saji-VB:~/kafka/kafka_2.13-2.8.0$ java -cp "/home/saji/kafka/kafka_2.13-2.8.0/libs/*":. SimpleConsumer Hello-Sayanthiny
Subscribed to topic Hello-Sayanthiny
offset = 30, key = 0, value = 0
offset = 31, key = 1, value = 1
offset = 32, key = 2, value = 2
offset = 33, key = 3, value = 3
offset = 34, key = 4, value = 4
offset = 35, key = 5, value = 5
offset = 36, key = 6, value = 6
offset = 37, key = 7, value = 7
offset = 38, key = 8, value = 8
offset = 39, key = 9, value = 9
```

Figure 41 Execution output

Input − Open the producer producer CLI and send some messages messages to the topic. Following will be the output.