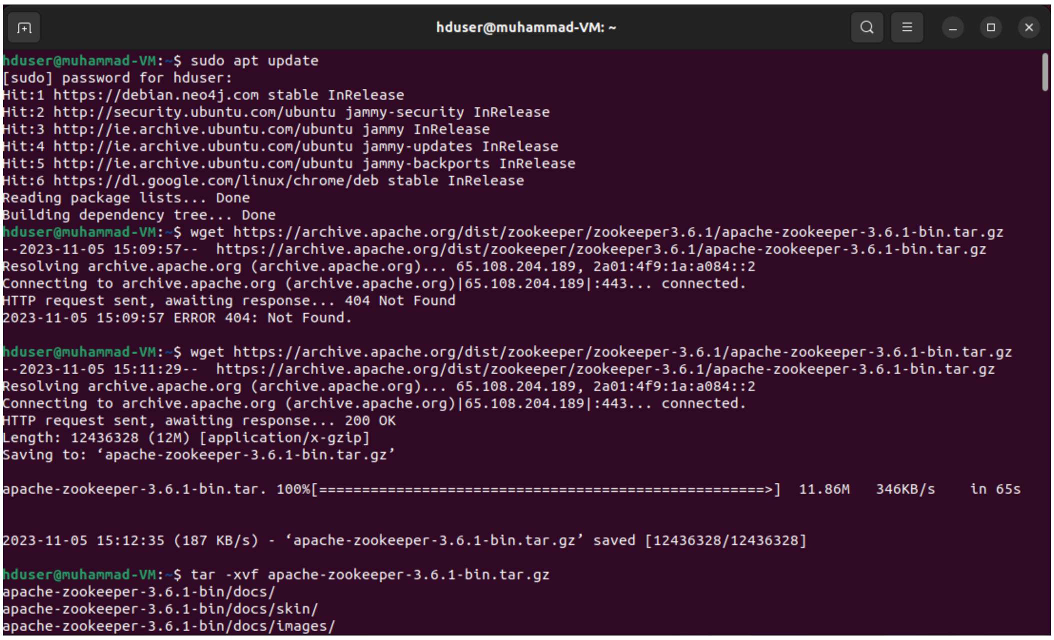
Tutorial Saturday

Apache Kafka

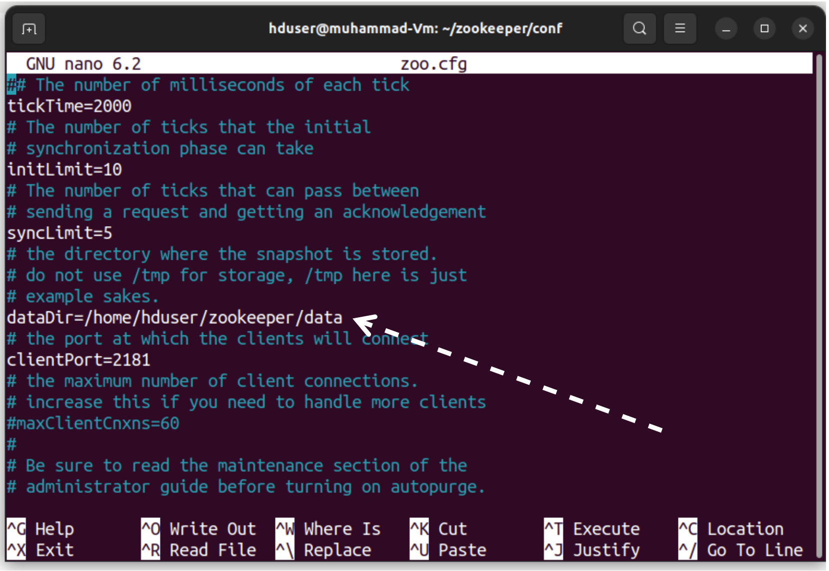
All commands are case sensitive on Ubuntu operating system

1) Start the terminal by writing on the search box or Press **Ctrl + Alt + t** together to open a terminal as shown below
   - **Note: $ sign shows the cursor on the ubuntu shell, do not write with commands**
2) Update the repository in Ubuntu by using the following command
```
$sudo apt update
```
3) Download Apache Zookeeper by using the following command as mentioned below
```
$wget https://archive.apache.org/dist/zookeeper/zookeeper-3.6.1/apache-zookeeper-3.6.1-bin.tar.gz
$tar -xzf apache-zookeeper-3.6.1-bin.tar.gz
```



```
hduser@muhammad-VM:~$ sudo apt update
[sudo] password for hduser:
Hit:1 https://debian.neo4j.com stable InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://ie.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://ie.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://ie.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 https://dl.google.com/linux/chrome/deb stable InRelease
Reading package lists... Done
Building dependency tree... Done
hduser@muhammad-VM:~$ wget https://archive.apache.org/dist/zookeeper/zookeeper3.6.1/apache-zookeeper-3.6.1-bin.tar.gz
--2023-11-05 15:09:57--  https://archive.apache.org/dist/zookeeper/zookeeper3.6.1/apache-zookeeper-3.6.1-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2023-11-05 15:09:57 ERROR 404: Not Found.

hduser@muhammad-VM:~$ wget https://archive.apache.org/dist/zookeeper/zookeeper-3.6.1/apache-zookeeper-3.6.1-bin.tar.gz
--2023-11-05 15:11:29--  https://archive.apache.org/dist/zookeeper/zookeeper-3.6.1/apache-zookeeper-3.6.1-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12436328 (12M) [application/x-gzip]
Saving to: ‘apache-zookeeper-3.6.1-bin.tar.gz’

apache-zookeeper-3.6.1-bin.tar. 100%[====================================================>]  11.86M   346KB/s    in 65s


2023-11-05 15:12:35 (187 KB/s) - ‘apache-zookeeper-3.6.1-bin.tar.gz’ saved [12436328/12436328]

hduser@muhammad-VM:~$ tar -xvf apache-zookeeper-3.6.1-bin.tar.gz
apache-zookeeper-3.6.1-bin/docs/
apache-zookeeper-3.6.1-bin/docs/skin/
apache-zookeeper-3.6.1-bin/docs/images/
```

4) Rename the folder to zookeeper
```
$mv apache-zookeeper-3.6.1-bin zookeeper
```
5) Create a folder named as “data” inside the zookeeper by using the following commands as
```
$cd zookeeper
$mkdir data
$cd conf
$cp ./zoo_sample.cfg ./zoo.cfg
```
6) Open zoo.cfg file and update one line as mentioned started with dataDir and set as mentioned below screenshot
```
$nano zoo.cfg
```



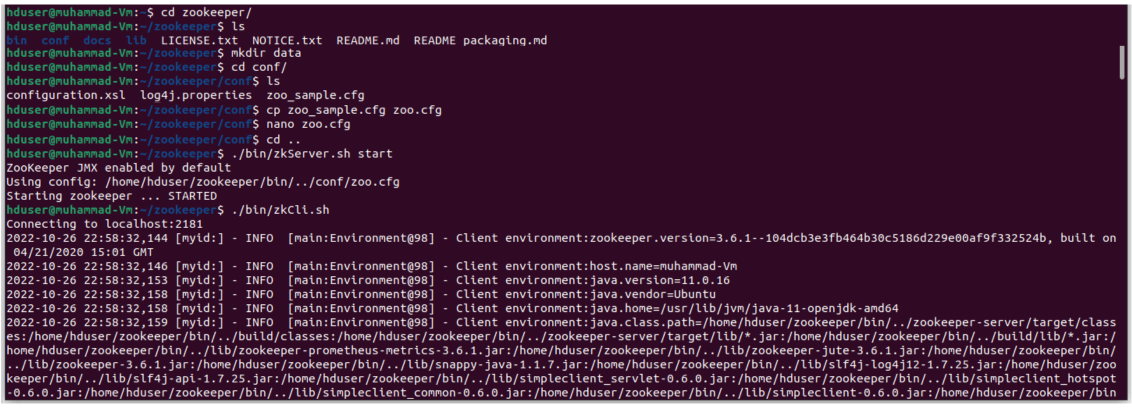
hduser@muhammad-Vm: ~/zookeeper/conf

```
  GNU nano 6.2                              zoo.cfg
# 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/hduser/zookeeper/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
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line
```

`dataDir=/home/hduser/zookeeper/data`

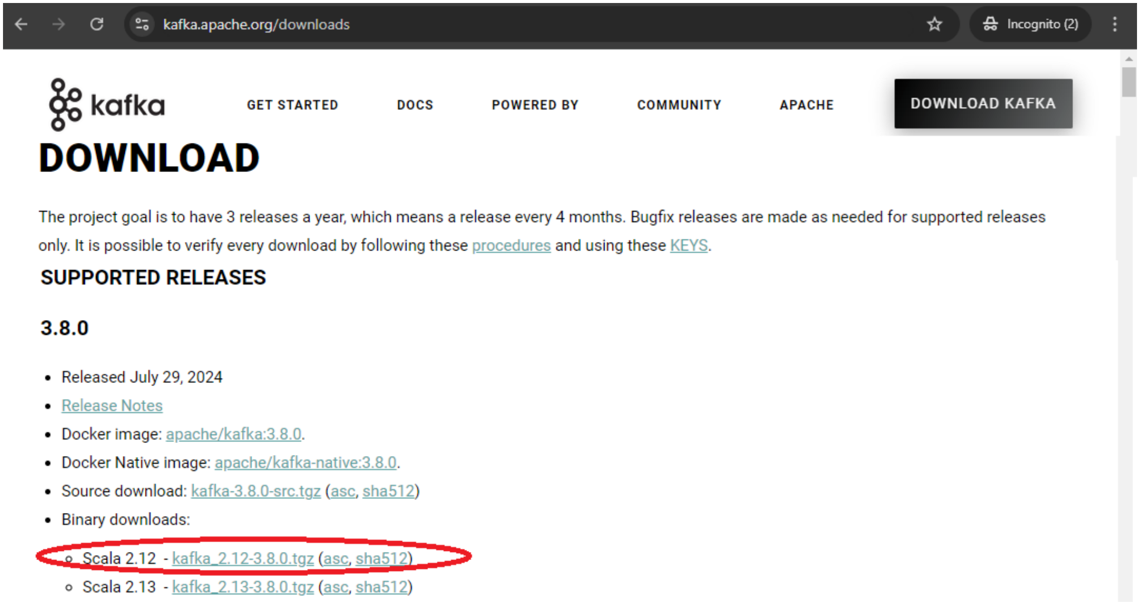
7) `$cd ..`

`$./bin/zkServer.sh start`



```
hduser@muhammad-Vm:~$ cd zookeeper/
hduser@muhammad-Vm:~/zookeeper$ ls
bin  conf  docs  lib  LICENSE.txt  NOTICE.txt  README.md  README packaging.md
hduser@muhammad-Vm:~/zookeeper$ mkdir data
hduser@muhammad-Vm:~/zookeeper$ cd conf/
hduser@muhammad-Vm:~/zookeeper/conf$ ls
configuration.xsl  log4j.properties  zoo_sample.cfg
hduser@muhammad-Vm:~/zookeeper/conf$ cp zoo_sample.cfg zoo.cfg
hduser@muhammad-Vm:~/zookeeper/conf$ nano zoo.cfg
hduser@muhammad-Vm:~/zookeeper/conf$ cd ..
hduser@muhammad-Vm:~/zookeeper$ ./bin/zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /home/hduser/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
hduser@muhammad-Vm:~/zookeeper$ ./bin/zkCli.sh
Connecting to localhost:2181
2022-10-26 22:58:32,144 [myid:] - INFO  [main:Environment@98] - Client environment:zookeeper.version=3.6.1--104dcb3e3fb464b30c5186d229e00af9f332524b, built on
 04/21/2020 15:01 GMT
2022-10-26 22:58:32,146 [myid:] - INFO  [main:Environment@98] - Client environment:host.name=muhammad-Vm
2022-10-26 22:58:32,153 [myid:] - INFO  [main:Environment@98] - Client environment:java.version=11.0.16
2022-10-26 22:58:32,158 [myid:] - INFO  [main:Environment@98] - Client environment:java.vendor=Ubuntu
2022-10-26 22:58:32,158 [myid:] - INFO  [main:Environment@98] - Client environment:java.home=/usr/lib/jvm/java-11-openjdk-amd64
2022-10-26 22:58:32,159 [myid:] - INFO  [main:Environment@98] - Client environment:java.class.path=/home/hduser/zookeeper/bin/../zookeeper-server/target/class
es:/home/hduser/zookeeper/bin/../build/classes:/home/hduser/zookeeper/bin/../zookeeper-server/target/lib/*.jar:/home/hduser/zookeeper/bin/../build/lib/*.jar:/
home/hduser/zookeeper/bin/../lib/zookeeper-prometheus-metrics-3.6.1.jar:/home/hduser/zookeeper/bin/../lib/zookeeper-jute-3.6.1.jar:/home/hduser/zookeeper/bin/
../lib/zookeeper-3.6.1.jar:/home/hduser/zookeeper/bin/../lib/snappy-java-1.1.7.jar:/home/hduser/zookeeper/bin/../lib/slf4j-log4j12-1.7.25.jar:/home/hduser/zoo
keeper/bin/../lib/slf4j-api-1.7.25.jar:/home/hduser/zookeeper/bin/../lib/simpleclient_servlet-0.6.0.jar:/home/hduser/zookeeper/bin/../lib/simpleclient_hotspot
-0.6.0.jar:/home/hduser/zookeeper/bin/../lib/simpleclient_common-0.6.0.jar:/home/hduser/zookeeper/bin/../lib/simpleclient-0.6.0.jar:/home/hduser/zookeeper/bin
```

8) Open a new terminal and download Kafka (Scala 2.12 - kafka_2.12-3.3.1.tgz (asc, sha512)) from the following link, https://kafka.apache.org/downloads



DOWNLOAD

The project goal is to have 3 releases a year, which means a release every 4 months. Bugfix releases are made as needed for supported releases only. It is possible to verify every download by following these procedures and using these KEYS.

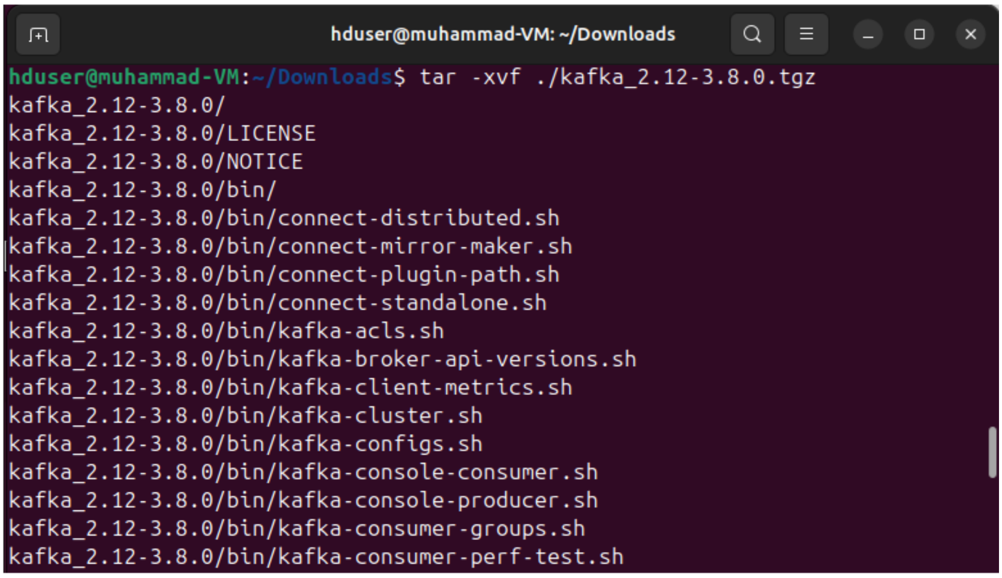
SUPPORTED RELEASES

3.8.0

- Released July 29, 2024
- Release Notes
- Docker image: apache/kafka:3.8.0.
- Docker Native image: apache/kafka-native:3.8.0.
- Source download: kafka-3.8.0-src.tgz (asc, sha512)
- Binary downloads:
  - Scala 2.12 - kafka_2.12-3.8.0.tgz (asc, sha512)
  - Scala 2.13 - kafka_2.13-3.8.0.tgz (asc, sha512)

9) Move to the download folder

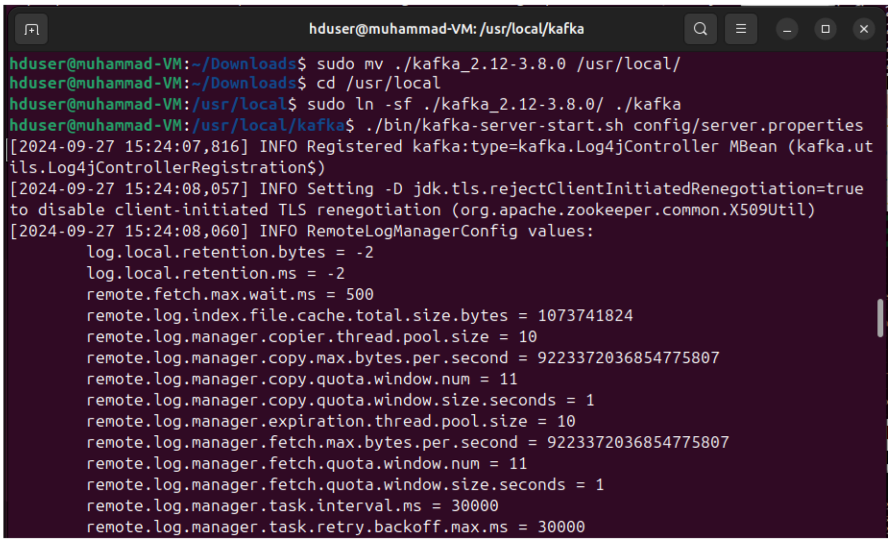
```
$cd Downloads
$tar -xvf ./kafka_2.12-3.8.0.tgz
```



```
hduser@muhammad-VM:~/Downloads$ tar -xvf ./kafka_2.12-3.8.0.tgz
kafka_2.12-3.8.0/
kafka_2.12-3.8.0/LICENSE
kafka_2.12-3.8.0/NOTICE
kafka_2.12-3.8.0/bin/
kafka_2.12-3.8.0/bin/connect-distributed.sh
kafka_2.12-3.8.0/bin/connect-mirror-maker.sh
kafka_2.12-3.8.0/bin/connect-plugin-path.sh
kafka_2.12-3.8.0/bin/connect-standalone.sh
kafka_2.12-3.8.0/bin/kafka-acls.sh
kafka_2.12-3.8.0/bin/kafka-broker-api-versions.sh
kafka_2.12-3.8.0/bin/kafka-client-metrics.sh
kafka_2.12-3.8.0/bin/kafka-cluster.sh
kafka_2.12-3.8.0/bin/kafka-configs.sh
kafka_2.12-3.8.0/bin/kafka-console-consumer.sh
kafka_2.12-3.8.0/bin/kafka-console-producer.sh
kafka_2.12-3.8.0/bin/kafka-consumer-groups.sh
kafka_2.12-3.8.0/bin/kafka-consumer-perf-test.sh
```

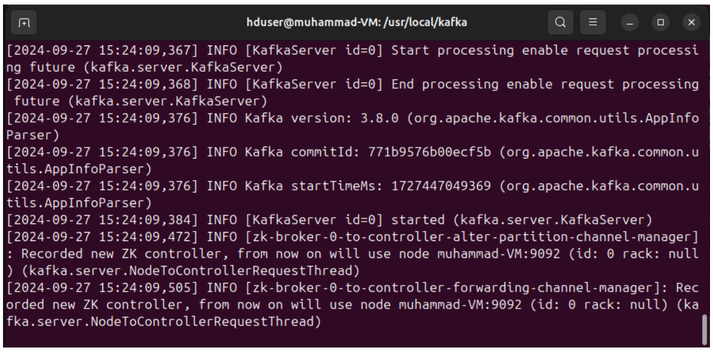
10) Move this folder to /usr/local folder

```
$sudo mv ./kafka_2.12-3.8.0 /usr/local/
$cd /usr/local
$sudo ln -sf ./kafka_2.12-3.8.0/ ./kafka
$cd kafka
$./bin/kafka-server-start.sh config/server.properties
```



```
hduser@muhammad-VM:~/Downloads$ sudo mv ./kafka_2.12-3.8.0 /usr/local/
hduser@muhammad-VM:~/Downloads$ cd /usr/local
hduser@muhammad-VM:/usr/local$ sudo ln -sf ./kafka_2.12-3.8.0/ ./kafka
hduser@muhammad-VM:/usr/local/kafka$ ./bin/kafka-server-start.sh config/server.properties
[2024-09-27 15:24:07,816] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.ut
ils.Log4jControllerRegistration$)
[2024-09-27 15:24:08,057] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true
to disable client-initiated TLS renegotiation (org.apache.zookeeper.common.X509Util)
[2024-09-27 15:24:08,060] INFO RemoteLogManagerConfig values:
        log.local.retention.bytes = -2
        log.local.retention.ms = -2
        remote.fetch.max.wait.ms = 500
        remote.log.index.file.cache.total.size.bytes = 1073741824
        remote.log.manager.copier.thread.pool.size = 10
        remote.log.manager.copy.max.bytes.per.second = 9223372036854775807
        remote.log.manager.copy.quota.window.num = 11
        remote.log.manager.copy.quota.window.size.seconds = 1
        remote.log.manager.expiration.thread.pool.size = 10
        remote.log.manager.fetch.max.bytes.per.second = 9223372036854775807
        remote.log.manager.fetch.quota.window.num = 11
        remote.log.manager.fetch.quota.window.size.seconds = 1
        remote.log.manager.task.interval.ms = 30000
        remote.log.manager.task.retry.backoff.max.ms = 30000
```

Leave this terminal as it is and open a new terminal and you can check all the processing that the Kafka will do on this terminal.



```
[2024-09-27 15:24:09,367] INFO [KafkaServer id=0] Start processing enable request processi
ng future (kafka.server.KafkaServer)
[2024-09-27 15:24:09,368] INFO [KafkaServer id=0] End processing enable request processing
 future (kafka.server.KafkaServer)
[2024-09-27 15:24:09,376] INFO Kafka version: 3.8.0 (org.apache.kafka.common.utils.AppInfo
Parser)
[2024-09-27 15:24:09,376] INFO Kafka commitId: 771b9576b00ecf5b (org.apache.kafka.common.u
tils.AppInfoParser)
[2024-09-27 15:24:09,376] INFO Kafka startTimeMs: 1727447049369 (org.apache.kafka.common.u
tils.AppInfoParser)
[2024-09-27 15:24:09,384] INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
[2024-09-27 15:24:09,472] INFO [zk-broker-0-to-controller-alter-partition-channel-manager]
: Recorded new ZK controller, from now on will use node muhammad-VM:9092 (id: 0 rack: null
) (kafka.server.NodeToControllerRequestThread)
[2024-09-27 15:24:09,505] INFO [zk-broker-0-to-controller-forwarding-channel-manager]: Rec
orded new ZK controller, from now on will use node muhammad-VM:9092 (id: 0 rack: null) (ka
fka.server.NodeToControllerRequestThread)
```

Kafka installation is completed, and the server is running. Leave this window open and start another terminal window. ™We now check the wordcount example for Kafka streaming

11) Now we test Apache Kafka installation that is working properly or not using WordCountdemo program. A client library used to build real-time and mission-critical applications is known as Apache Kafka Streams. The input and output data for Kafka Stream operations are kept in a Kafka cluster.
12) Make sure that you have started Kafka and Zookeeper Server as we did in step 7 and step 10.
    Or
    Use the following commands again if you have not started yet

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

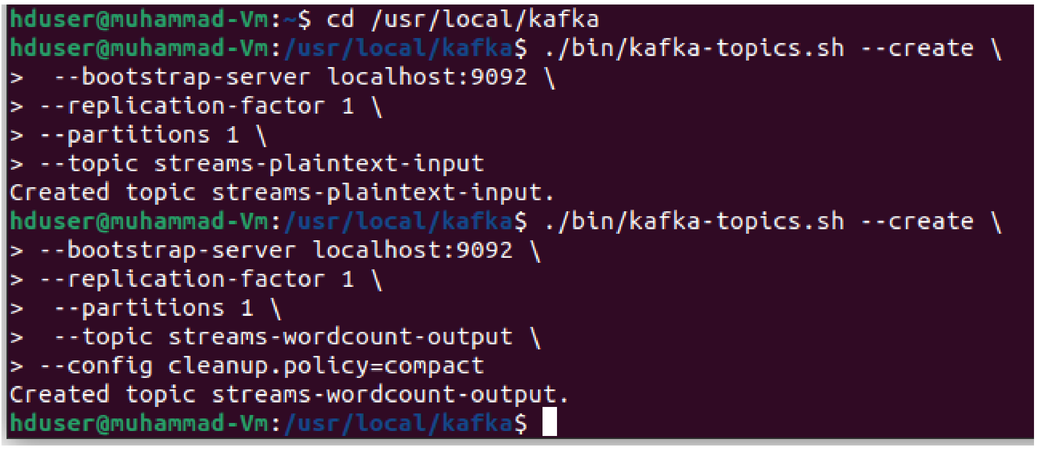
13) Open a new terminal. Create Input and Output Topic: Now we will create an input topic named streams-plaintext-input and the output topic named streams-wordcount-output.
    a) Command for input topic
       Move the folder

```
$cd /usr/local/kafka
$./bin/kafka-topics.sh --create \
--bootstrap-server localhost:9092 \
--replication-factor 1 \
--partitions 1 \
--topic streams-plaintext-input
```

    b) Command for output topic

```
$./bin/kafka-topics.sh --create \
--bootstrap-server localhost:9092 \
--replication-factor 1 \
--partitions 1 \
--topic streams-wordcount-output \
--config cleanup.policy=compact
```

The screenshot for the execution of the above commands is shown below

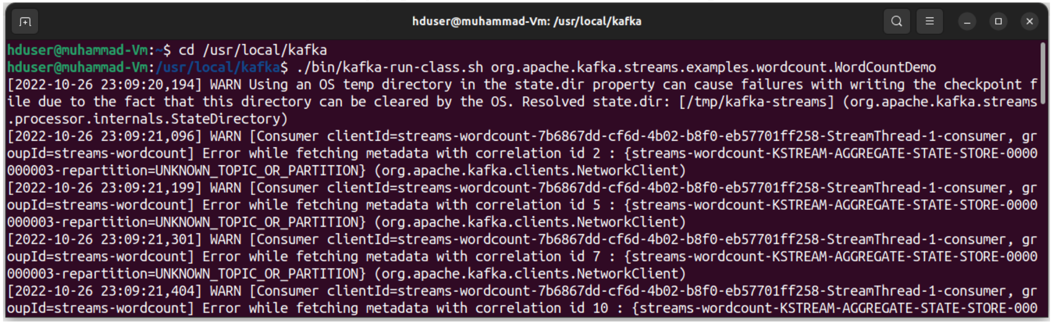


```
hduser@muhammad-Vm:~$ cd /usr/local/kafka
hduser@muhammad-Vm:/usr/local/kafka$ ./bin/kafka-topics.sh --create \
>  --bootstrap-server localhost:9092 \
> --replication-factor 1 \
> --partitions 1 \
> --topic streams-plaintext-input
Created topic streams-plaintext-input.
hduser@muhammad-Vm:/usr/local/kafka$ ./bin/kafka-topics.sh --create \
> --bootstrap-server localhost:9092 \
> --replication-factor 1 \
>  --partitions 1 \
>  --topic streams-wordcount-output \
> --config cleanup.policy=compact
Created topic streams-wordcount-output.
hduser@muhammad-Vm:/usr/local/kafka$
```

14) Open a new terminal and start the wordcount application using the below command.

The WordCount application program will read data from the input topic streams-plaintext-input and perform the computations of the WordCount algorithm on each of the read messages, and continuously write its current results to the output topic streams-wordcount-output.

```
$cd /usr/local/kafka
$./bin/kafka-run-class.sh org.apache.kafka.streams.examples.wordcount.WordCountDemo
```

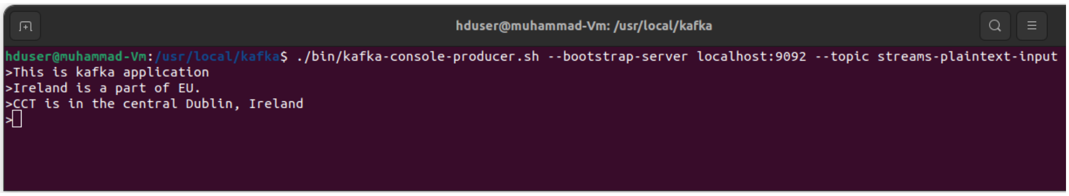


```
hduser@muhammad-Vm:~$ cd /usr/local/kafka
hduser@muhammad-Vm:/usr/local/kafka$ ./bin/kafka-run-class.sh org.apache.kafka.streams.examples.wordcount.WordCountDemo
[2022-10-26 23:09:20,194] WARN Using an OS temp directory in the state.dir property can cause failures with writing the checkpoint f
ile due to the fact that this directory can be cleared by the OS. Resolved state.dir: [/tmp/kafka-streams] (org.apache.kafka.streams
.processor.internals.StateDirectory)
[2022-10-26 23:09:21,096] WARN [Consumer clientId=streams-wordcount-7b6867dd-cf6d-4b02-b8f0-eb57701ff258-StreamThread-1-consumer, gr
oupId=streams-wordcount] Error while fetching metadata with correlation id 2 : {streams-wordcount-KSTREAM-AGGREGATE-STATE-STORE-0000
000003-repartition=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2022-10-26 23:09:21,199] WARN [Consumer clientId=streams-wordcount-7b6867dd-cf6d-4b02-b8f0-eb57701ff258-StreamThread-1-consumer, gr
oupId=streams-wordcount] Error while fetching metadata with correlation id 5 : {streams-wordcount-KSTREAM-AGGREGATE-STATE-STORE-0000
000003-repartition=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2022-10-26 23:09:21,301] WARN [Consumer clientId=streams-wordcount-7b6867dd-cf6d-4b02-b8f0-eb57701ff258-StreamThread-1-consumer, gr
oupId=streams-wordcount] Error while fetching metadata with correlation id 7 : {streams-wordcount-KSTREAM-AGGREGATE-STATE-STORE-0000
000003-repartition=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2022-10-26 23:09:21,404] WARN [Consumer clientId=streams-wordcount-7b6867dd-cf6d-4b02-b8f0-eb57701ff258-StreamThread-1-consumer, gr
oupId=streams-wordcount] Error while fetching metadata with correlation id 10 : {streams-wordcount-KSTREAM-AGGREGATE-STATE-STORE-000
```

Leave this terminal open as it is and open two new terminals for input and output topics.

15) **Start Producer:** Open a new terminal and start Kafka producer using tool “kafka-console-producer.sh” and write some input.

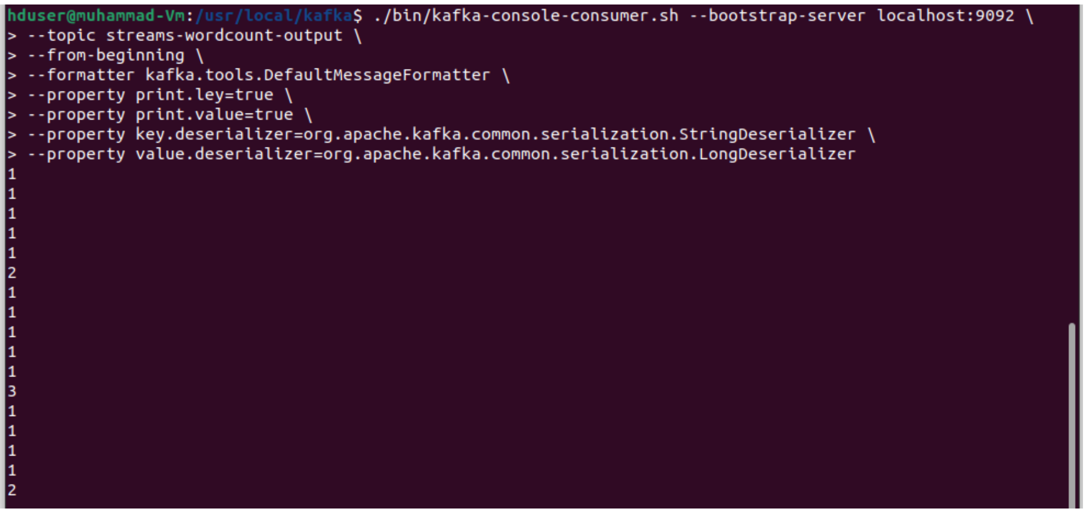
```
$cd /usr/local/kafka
$./bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topic streams-plaintext-input
```



```
hduser@muhammad-Vm:/usr/local/kafka$ ./bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topic streams-plaintext-input
>This is kafka application
>Ireland is a part of EU.
>CCT is in the central Dublin, Ireland
>
```

16) **Start Producer:** Open another terminal and run the consumer tool to read the input, process it, and print word count output.

```
$cd /usr/local/kafka
$./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 \
--topic streams-wordcount-output \
--from-beginning \
--formatter kafka.tools.DefaultMessageFormatter \
--property print.ley=true \
--property print.value=true \
--property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer \
--property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer
```



```
hduser@muhammad-Vm:/usr/local/kafka$ ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 \
> --topic streams-wordcount-output \
> --from-beginning \
> --formatter kafka.tools.DefaultMessageFormatter \
> --property print.ley=true \
> --property print.value=true \
> --property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer \
> --property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer
1
1
1
1
1
2
1
1
1
1
1
3
1
1
1
1
2
```

We can view the Wordcount application's output, which is essentially a continuous stream of updates with each output record representing an updated word count.

If you would like to explore further, the following websites might be useful as mentioned below

Task

Complete the Web Application chat and the instructions are available at the following link (https://www.cloudduggu.com/kafka/messaging-project/).

References:

- https://www.cloudduggu.com/kafka/
- https://www.cloudduggu.com/kafka/installation/
- https://www.cloudduggu.com/kafka/streams-application/