Tutorial 7

Apache Cassandra

To get started with Cassandra NoSQL database, we will step through a single-node, local installation on VM.

1) The following points are the requirements to run Cassandra locally: Move to the Apache download site for the Cassandra project (http://cassandra.apache.org/download/), choose 3.11.16, and select a mirror to download the latest version of Cassandra. When complete, copy the .tar or .gzip file to a location that your user has read and write permissions for. This example will assume that this is going to be the ~/Downloads/ directory on ubuntu VM.

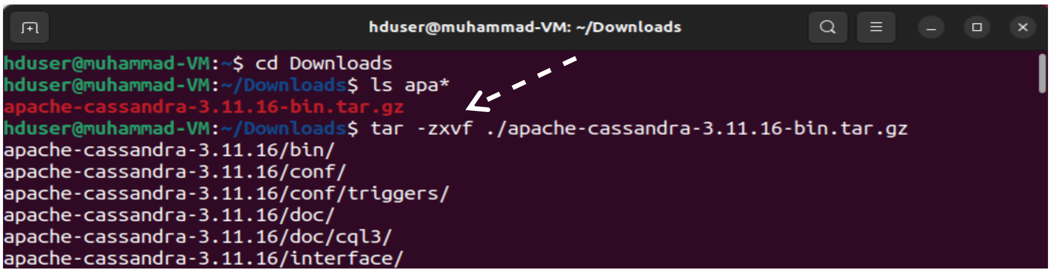
Note: If you could not understand from the command instructions, then please check the screenshot for better understanding. If you like to explore the details of each command along with examples, then check the documentation of Cassandara on website: https://cassandra.apache.org/doc/latest/

Download Apache Cassandra from the following link as mentioned below

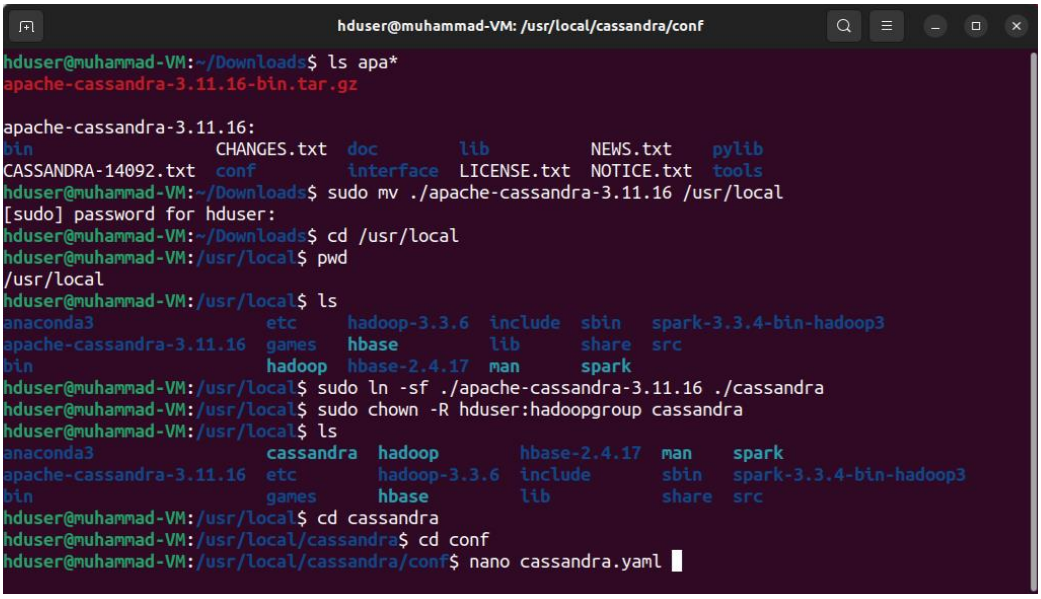
https://www.apache.org/dyn/closer.lua/cassandra/3.11.16/apache-cassandra-3.11.16-bin.tar.gz

2) **`$cd Downloads`**

Follow the instructions to unzip on the below screenshots and change the name of the folder as you did during the Hadoop, HBase and Spark installations (Tutorials, 2, 4, 6).



```
hduser@muhammad-VM:~$ cd Downloads
hduser@muhammad-VM:~/Downloads$ ls apa*
apache-cassandra-3.11.16-bin.tar.gz
hduser@muhammad-VM:~/Downloads$ tar -zxvf ./apache-cassandra-3.11.16-bin.tar.gz
apache-cassandra-3.11.16/bin/
apache-cassandra-3.11.16/conf/
apache-cassandra-3.11.16/conf/triggers/
apache-cassandra-3.11.16/doc/
apache-cassandra-3.11.16/doc/cql3/
apache-cassandra-3.11.16/interface/
```



```
hduser@muhammad-VM:~/Downloads$ ls apa*
apache-cassandra-3.11.16-bin.tar.gz

apache-cassandra-3.11.16:
bin                CHANGES.txt  doc        lib          NEWS.txt    pylib
CASSANDRA-14092.txt  conf       interface  LICENSE.txt  NOTICE.txt  tools
hduser@muhammad-VM:~/Downloads$ sudo mv ./apache-cassandra-3.11.16 /usr/local
[sudo] password for hduser:
hduser@muhammad-VM:~/Downloads$ cd /usr/local
hduser@muhammad-VM:/usr/local$ pwd
/usr/local
hduser@muhammad-VM:/usr/local$ ls
anaconda3                 etc     hadoop-3.3.6  include  sbin   spark-3.3.4-bin-hadoop3
apache-cassandra-3.11.16  games   hbase         lib      share  src
bin                       hadoop  hbase-2.4.17  man      spark
hduser@muhammad-VM:/usr/local$ sudo ln -sf ./apache-cassandra-3.11.16 ./cassandra
hduser@muhammad-VM:/usr/local$ sudo chown -R hduser:hadoopgroup cassandra
hduser@muhammad-VM:/usr/local$ ls
anaconda3                 cassandra  hadoop        hbase-2.4.17  man    spark
apache-cassandra-3.11.16  etc        hadoop-3.3.6  include       sbin   spark-3.3.4-bin-hadoop3
bin                       games      hbase         lib           share  src
hduser@muhammad-VM:/usr/local$ cd cassandra
hduser@muhammad-VM:/usr/local/cassandra$ cd conf
hduser@muhammad-VM:/usr/local/cassandra/conf$ nano cassandra.yaml
```

3) Configuration: At this point, you could start your node with no further configuration. However, it is good to get into the habit of checking and adjusting the properties that are indicated as follows using instructions as shown in the screenshot in step no. 2.

```
$cd /usr/local
$cd cassandra
$cd conf
$nano cassandra.yaml
```

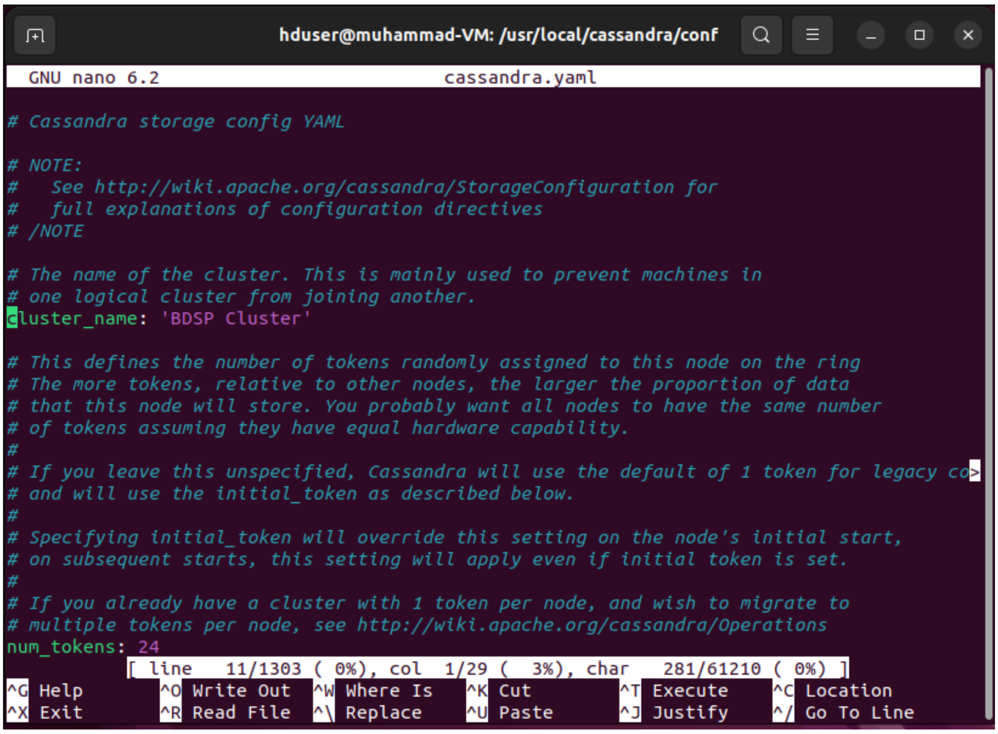
It is usually a good idea to rename your cluster. Inside the conf/cassandra.yaml file, specify a new cluster_name property, overwriting the default Test Cluster as shown below in screenshot:

```
cluster_name: 'BDSP Cluster'
```

The num_tokens property default of 256 has proven to be too high for the newer, 3.x versions of Cassandra. Go ahead and set that to 24:

```
num_tokens: 24
```

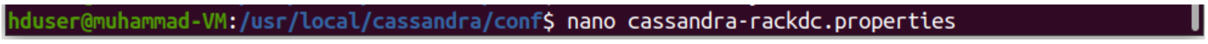
save the file using nano/ gedit editor after updating above parameters.



```
  GNU nano 6.2                          cassandra.yaml

# Cassandra storage config YAML

# NOTE:
#   See http://wiki.apache.org/cassandra/StorageConfiguration for
#   full explanations of configuration directives
# /NOTE

# The name of the cluster. This is mainly used to prevent machines in
# one logical cluster from joining another.
cluster_name: 'BDSP Cluster'

# This defines the number of tokens randomly assigned to this node on the ring
# The more tokens, relative to other nodes, the larger the proportion of data
# that this node will store. You probably want all nodes to have the same number
# of tokens assuming they have equal hardware capability.
#
# If you leave this unspecified, Cassandra will use the default of 1 token for legacy co>
# and will use the initial_token as described below.
#
# Specifying initial_token will override this setting on the node's initial start,
# on subsequent starts, this setting will apply even if initial token is set.
#
# If you already have a cluster with 1 token per node, and wish to migrate to
# multiple tokens per node, see http://wiki.apache.org/cassandra/Operations
num_tokens: 24
              [ line   11/1303 ( 0%), col  1/29 (  3%), char   281/61210 ( 0%) ]
^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
```

Press Alt+C to display the line number using nano editor.

4) By default, Cassandra will come up bound to localhost or 127.0.0.1. For your own local development machine



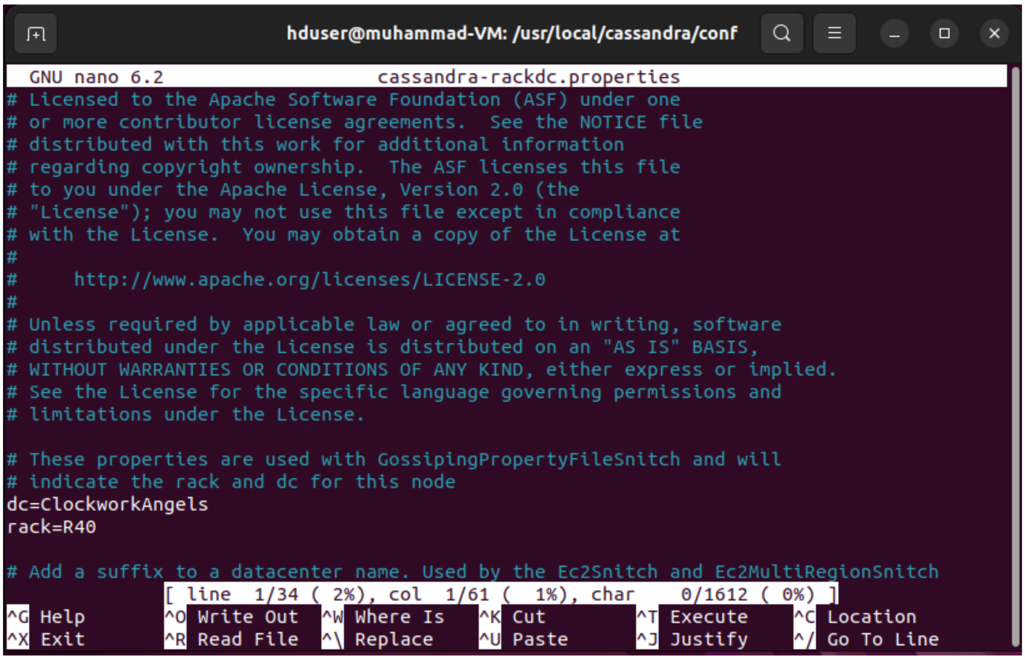
```
hduser@muhammad-VM:/usr/local/cassandra/conf$ nano cassandra-rackdc.properties
```

```
cassandra-rackdc.properties
```

In terms of NoSQL databases, Apache Cassandra handles multi-data center awareness better than any other. To configure this, each node must use **GossipingPropertyFileSnitch** (as previously mentioned in the preceding **cassandra.yaml** configuration process) and must have its local data center (and rack) settings defined. Therefore, we set the dc and rack properties in the **conf/cassandra-rackdc.properties** file:

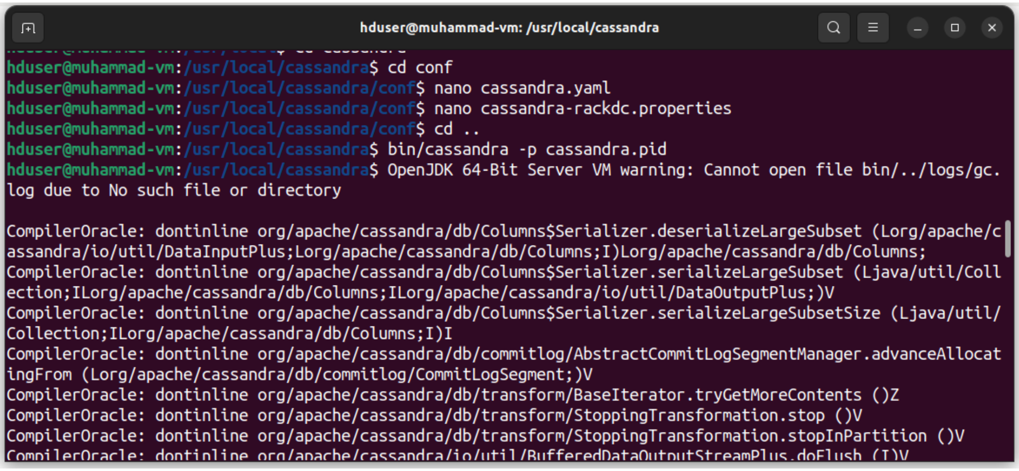
```
dc=ClockworkAngels
rack=R40
```

If these properties are already set as shown below in the screenshot. You can exit from the editor.



```
  GNU nano 6.2                  cassandra-rackdc.properties
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements.  See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership.  The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# These properties are used with GossipingPropertyFileSnitch and will
# indicate the rack and dc for this node
dc=ClockworkAngels
rack=R40

# Add a suffix to a datacenter name. Used by the Ec2Snitch and Ec2MultiRegionSnitch
                [ line   1/34 ( 2%), col   1/61 (  1%), char     0/1612 ( 0%) ]
^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
```

5) Starting Cassandra: To start Cassandra locally, execute the Cassandra script. If no arguments are passed, it will run in the foreground. To have it run in the background, send the **-p flag** with a destination file for the **Process ID** (**PID**):



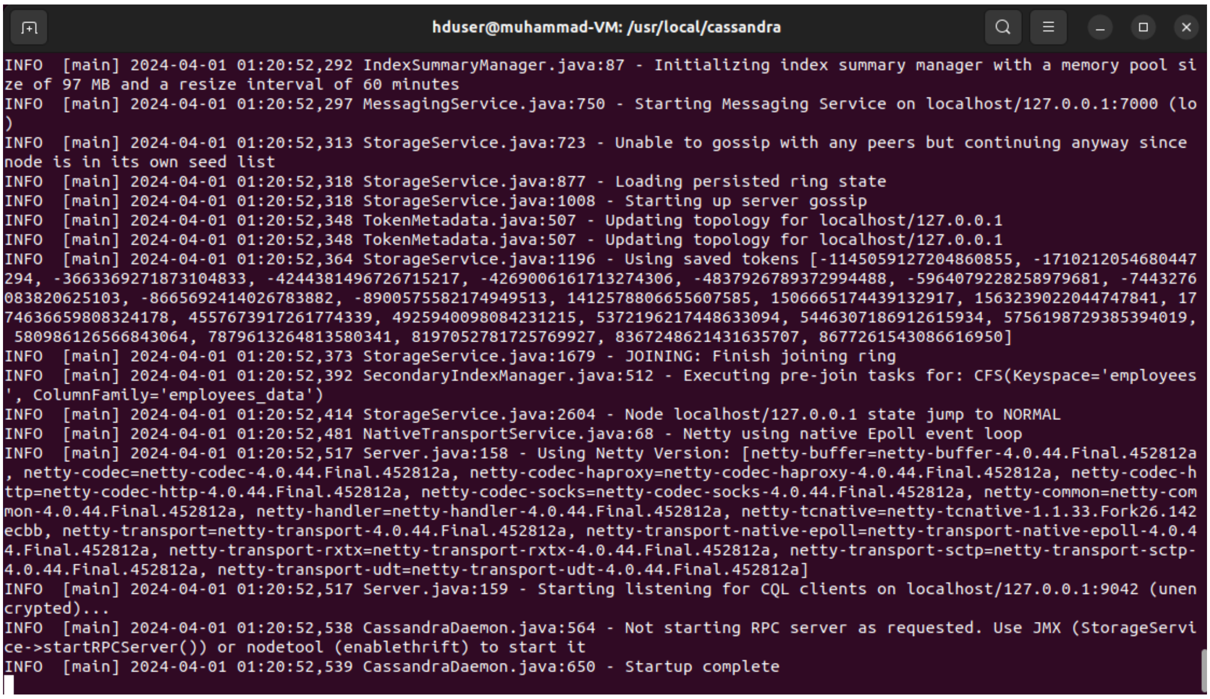
```
hduser@muhammad-vm:/usr/local/cassandra$ cd conf
hduser@muhammad-vm:/usr/local/cassandra/conf$ nano cassandra.yaml
hduser@muhammad-vm:/usr/local/cassandra/conf$ nano cassandra-rackdc.properties
hduser@muhammad-vm:/usr/local/cassandra/conf$ cd ..
hduser@muhammad-vm:/usr/local/cassandra$ bin/cassandra -p cassandra.pid
hduser@muhammad-vm:/usr/local/cassandra$ OpenJDK 64-Bit Server VM warning: Cannot open file bin/../logs/gc.
log due to No such file or directory

CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.deserializeLargeSubset (Lorg/apache/c
assandra/io/util/DataInputPlus;Lorg/apache/cassandra/db/Columns;I)Lorg/apache/cassandra/db/Columns;
CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubset (Ljava/util/Coll
ection;ILorg/apache/cassandra/db/Columns;ILorg/apache/cassandra/io/util/DataOutputPlus;)V
CompilerOracle: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubsetSize (Ljava/util/
Collection;ILorg/apache/cassandra/db/Columns;I)I
CompilerOracle: dontinline org/apache/cassandra/db/commitlog/AbstractCommitLogSegmentManager.advanceAllocat
ingFrom (Lorg/apache/cassandra/db/commitlog/CommitLogSegment;)V
CompilerOracle: dontinline org/apache/cassandra/db/transform/BaseIterator.tryGetMoreContents ()Z
CompilerOracle: dontinline org/apache/cassandra/db/transform/StoppingTransformation.stop ()V
CompilerOracle: dontinline org/apache/cassandra/db/transform/StoppingTransformation.stopInPartition ()V
CompilerOracle: dontinline org/apache/cassandra/io/util/BufferedDataOutputStreamPlus.doFlush (I)V
```

```
$bin/cassandra -p cassandra.pid
```

Or

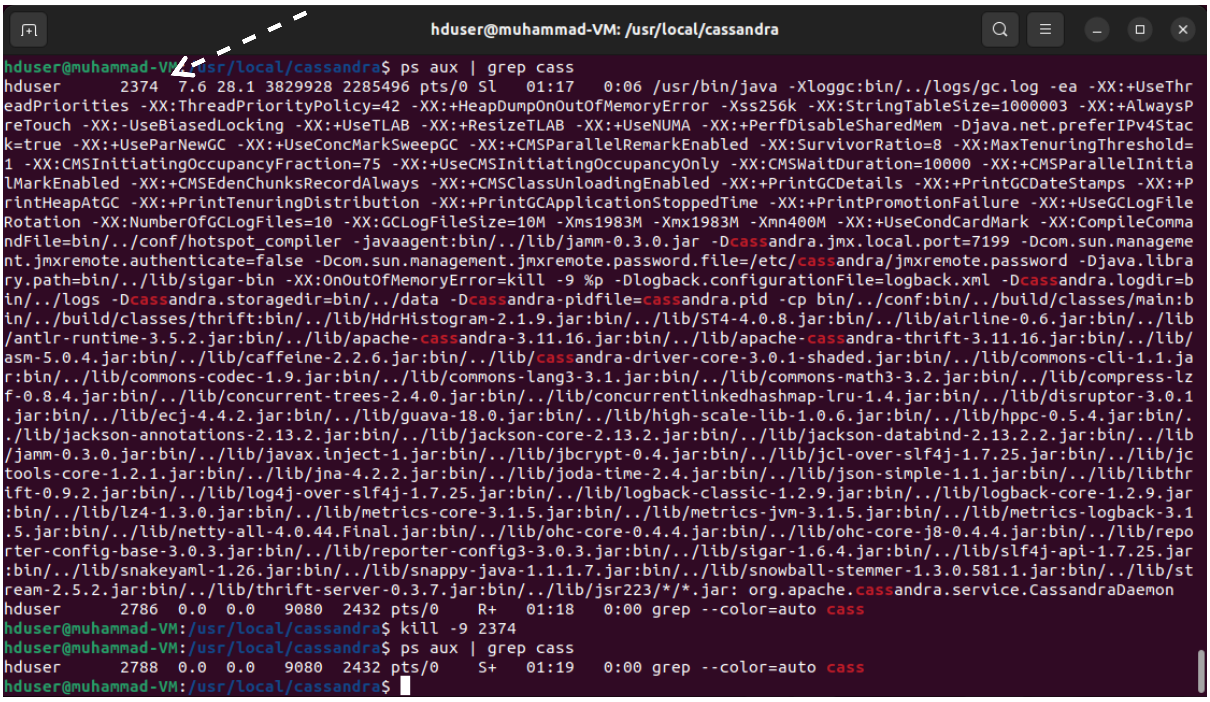
```
$bin/cassandra -f
```

When the Cassandra started, then leave this terminal as shown below



```
INFO  [main] 2024-04-01 01:20:52,292 IndexSummaryManager.java:87 - Initializing index summary manager with a memory pool si
ze of 97 MB and a resize interval of 60 minutes
INFO  [main] 2024-04-01 01:20:52,297 MessagingService.java:750 - Starting Messaging Service on localhost/127.0.0.1:7000 (lo
)
INFO  [main] 2024-04-01 01:20:52,313 StorageService.java:723 - Unable to gossip with any peers but continuing anyway since
node is in its own seed list
INFO  [main] 2024-04-01 01:20:52,318 StorageService.java:877 - Loading persisted ring state
INFO  [main] 2024-04-01 01:20:52,318 StorageService.java:1008 - Starting up server gossip
INFO  [main] 2024-04-01 01:20:52,348 TokenMetadata.java:507 - Updating topology for localhost/127.0.0.1
INFO  [main] 2024-04-01 01:20:52,348 TokenMetadata.java:507 - Updating topology for localhost/127.0.0.1
INFO  [main] 2024-04-01 01:20:52,364 StorageService.java:1196 - Using saved tokens [-1145059127204860855, -1710212054680447
294, -3663369271873104833, -4244381496726715217, -4269006161713274306, -4837926789372994488, -5964079228258979681, -7443276
083820625103, -8665692414026783882, -8900575582174949513, 1412578806655607585, 1506665174439132917, 1563239022044747841, 17
74636659808324178, 4557673917261774339, 4925940098084231215, 5372196217448633094, 5446307186912615934, 5756198729385394019,
 5809861265668430064, 7879613264813580341, 8197052781725769927, 8367248621431635707, 8677261543086616950]
INFO  [main] 2024-04-01 01:20:52,373 StorageService.java:1679 - JOINING: Finish joining ring
INFO  [main] 2024-04-01 01:20:52,392 SecondaryIndexManager.java:512 - Executing pre-join tasks for: CFS(Keyspace='employees
', ColumnFamily='employees_data')
INFO  [main] 2024-04-01 01:20:52,414 StorageService.java:2604 - Node localhost/127.0.0.1 state jump to NORMAL
INFO  [main] 2024-04-01 01:20:52,481 NativeTransportService.java:68 - Netty using native Epoll event loop
INFO  [main] 2024-04-01 01:20:52,517 Server.java:158 - Using Netty Version: [netty-buffer=netty-buffer-4.0.44.Final.452812a
, netty-codec=netty-codec-4.0.44.Final.452812a, netty-codec-haproxy=netty-codec-haproxy-4.0.44.Final.452812a, netty-codec-h
ttp=netty-codec-http-4.0.44.Final.452812a, netty-codec-socks=netty-codec-socks-4.0.44.Final.452812a, netty-common=netty-com
mon-4.0.44.Final.452812a, netty-handler=netty-handler-4.0.44.Final.452812a, netty-tcnative=netty-tcnative-1.1.33.Fork26.142
ecbb, netty-transport=netty-transport-4.0.44.Final.452812a, netty-transport-native-epoll=netty-transport-native-epoll-4.0.4
4.Final.452812a, netty-transport-rxtx=netty-transport-rxtx-4.0.44.Final.452812a, netty-transport-sctp=netty-transport-sctp-
4.0.44.Final.452812a, netty-transport-udt=netty-transport-udt-4.0.44.Final.452812a]
INFO  [main] 2024-04-01 01:20:52,517 Server.java:159 - Starting listening for CQL clients on localhost/127.0.0.1:9042 (unen
crypted)...
INFO  [main] 2024-04-01 01:20:52,538 CassandraDaemon.java:564 - Not starting RPC server as requested. Use JMX (StorageServi
ce->startRPCServer()) or nodetool (enablethrift) to start it
INFO  [main] 2024-04-01 01:20:52,539 CassandraDaemon.java:650 - Startup complete
```

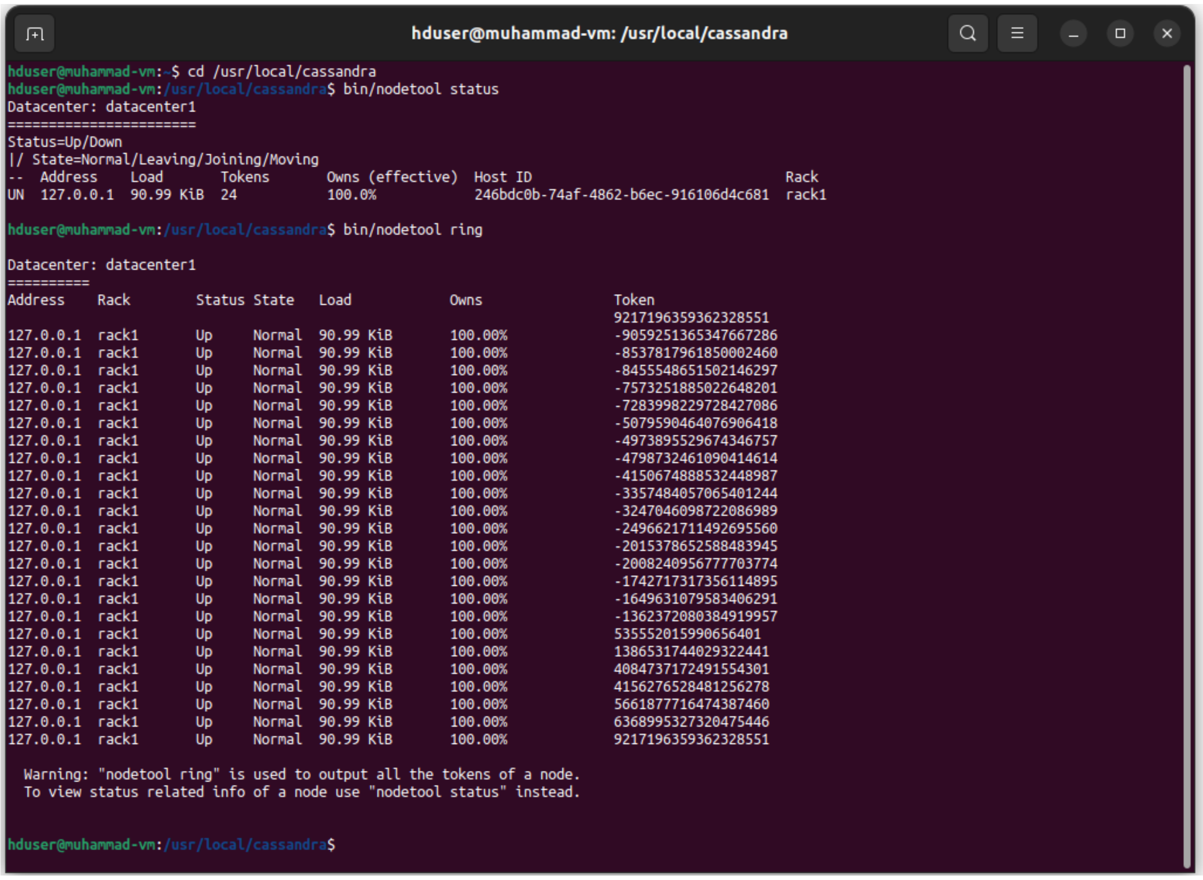
Let this window stay as is for the moment, don't close it. However, if you would like to stop Cassandra, press Ctrl + C to stop the Cassandra process or you can use the command to kill the process as mentioned below on the screen. **You must perform these steps if the Cassandra showed an error in the case of start-up using this command (`bin/cassandra -p cassandra.pid` or `bin/cassandra -f`).**



```
hduser@muhammad-VM:/usr/local/cassandra$ ps aux | grep cass
hduser      2374  7.6 28.1 3829928 2285496 pts/0 Sl   01:17   0:06 /usr/bin/java -Xloggc:bin/../logs/gc.log -ea -XX:+UseThr
eadPriorities -XX:ThreadPriorityPolicy=42 -XX:+HeapDumpOnOutOfMemoryError -Xss256k -XX:StringTableSize=1000003 -XX:+AlwaysP
reTouch -XX:-UseBiasedLocking -XX:+UseTLAB -XX:+ResizeTLAB -XX:+UseNUMA -XX:+PerfDisableSharedMem -Djava.net.preferIPv4Stac
k=true -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:SurvivorRatio=8 -XX:MaxTenuringThreshold=
1 -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSWaitDuration=10000 -XX:+CMSParallelInitia
lMarkEnabled -XX:+CMSEdenChunksRecordAlways -XX:+CMSClassUnloadingEnabled -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+P
rintHeapAtGC -XX:+PrintTenuringDistribution -XX:+PrintGCApplicationStoppedTime -XX:+PrintPromotionFailure -XX:+UseGCLogFile
Rotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=10M -Xms1983M -Xmx1983M -Xmn400M -XX:+UseCondCardMark -XX:CompileComma
ndFile=bin/../conf/hotspot_compiler -javaagent:bin/../lib/jamm-0.3.0.jar -Dcassandra.jmx.local.port=7199 -Dcom.sun.manageme
nt.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.password.file=/etc/cassandra/jmxremote.password -Djava.libra
ry.path=bin/../lib/sigar-bin -XX:OnOutOfMemoryError=kill -9 %p -Dlogback.configurationFile=logback.xml -Dcassandra.logdir=b
in/../logs -Dcassandra.storagedir=bin/../data -Dcassandra-pidfile=cassandra.pid -cp bin/../conf:bin/../build/classes/main:b
in/../build/classes/thrift:bin/../lib/HdrHistogram-2.1.9.jar:bin/../lib/ST4-4.0.8.jar:bin/../lib/airline-0.6.jar:bin/../lib
/antlr-runtime-3.5.2.jar:bin/../lib/apache-cassandra-3.11.16.jar:bin/../lib/apache-cassandra-thrift-3.11.16.jar:bin/../lib/
asm-5.0.4.jar:bin/../lib/caffeine-2.2.6.jar:bin/../lib/cassandra-driver-core-3.0.1-shaded.jar:bin/../lib/commons-cli-1.1.ja
r:bin/../lib/commons-codec-1.9.jar:bin/../lib/commons-lang3-3.1.jar:bin/../lib/commons-math3-3.2.jar:bin/../lib/compress-lz
f-0.8.4.jar:bin/../lib/concurrent-trees-2.4.0.jar:bin/../lib/concurrentlinkedhashmap-lru-1.4.jar:bin/../lib/disruptor-3.0.1
.jar:bin/../lib/ecj-4.4.2.jar:bin/../lib/guava-18.0.jar:bin/../lib/high-scale-lib-1.0.6.jar:bin/../lib/hppc-0.5.4.jar:bin/.
./lib/jackson-annotations-2.13.2.jar:bin/../lib/jackson-core-2.13.2.jar:bin/../lib/jackson-databind-2.13.2.2.jar:bin/../lib
/jamm-0.3.0.jar:bin/../lib/javax.inject-1.jar:bin/../lib/jbcrypt-0.4.jar:bin/../lib/jcl-over-slf4j-1.7.25.jar:bin/../lib/jc
tools-core-1.2.1.jar:bin/../lib/jna-4.2.2.jar:bin/../lib/joda-time-2.4.jar:bin/../lib/json-simple-1.1.jar:bin/../lib/libthr
ift-0.9.2.jar:bin/../lib/log4j-over-slf4j-1.7.25.jar:bin/../lib/logback-classic-1.2.9.jar:bin/../lib/logback-core-1.2.9.jar
:bin/../lib/lz4-1.3.0.jar:bin/../lib/metrics-core-3.1.5.jar:bin/../lib/metrics-jvm-3.1.5.jar:bin/../lib/metrics-logback-3.1
.5.jar:bin/../lib/netty-all-4.0.44.Final.jar:bin/../lib/ohc-core-0.4.4.jar:bin/../lib/ohc-core-j8-0.4.4.jar:bin/../lib/repo
rter-config-base-3.0.3.jar:bin/../lib/reporter-config3-3.0.3.jar:bin/../lib/sigar-1.6.4.jar:bin/../lib/slf4j-api-1.7.25.jar
:bin/../lib/snakeyaml-1.26.jar:bin/../lib/snappy-java-1.1.1.7.jar:bin/../lib/snowball-stemmer-1.3.0.581.1.jar:bin/../lib/st
ream-2.5.2.jar:bin/../lib/thrift-server-0.3.7.jar:bin/../lib/jsr223/*/*.jar: org.apache.cassandra.service.CassandraDaemon
hduser      2786  0.0  0.0   9080  2432 pts/0    R+   01:18   0:00 grep --color=auto cass
hduser@muhammad-VM:/usr/local/cassandra$ kill -9 2374
hduser@muhammad-VM:/usr/local/cassandra$ ps aux | grep cass
hduser      2788  0.0  0.0   9080  2432 pts/0    S+   01:19   0:00 grep --color=auto cass
hduser@muhammad-VM:/usr/local/cassandra$
```

This will store the PID of the Cassandra process in a file named **`cassandra.pid`** in the **`local/cassandra`** directory. Several messages will be dumped to the screen.

Open a new terminal by pressing **ctrl+ Alt + t**, Check the status of Cassandra by using the following commands as mentioned below in the screenshot



hduser@muhammad-vm: /usr/local/cassandra

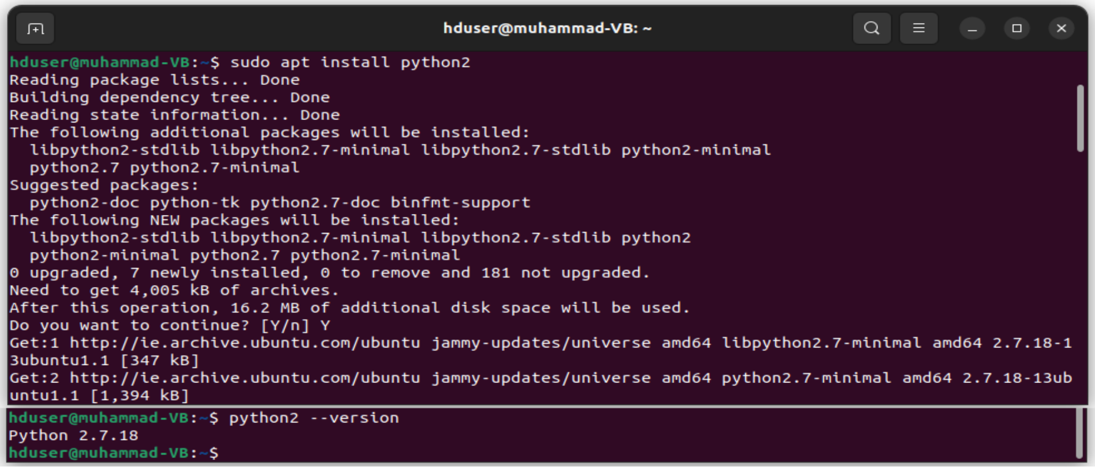
```
hduser@muhammad-vm:~$ cd /usr/local/cassandra
hduser@muhammad-vm:/usr/local/cassandra$ bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address    Load       Tokens       Owns (effective)  Host ID                               Rack
UN  127.0.0.1  90.99 KiB  24           100.0%            246bdc0b-74af-4862-b6ec-916106d4c681  rack1

hduser@muhammad-vm:/usr/local/cassandra$ bin/nodetool ring

Datacenter: datacenter1
==========
Address    Rack        Status State   Load            Owns                Token
                                                                          9217196359362328551
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -9059251365347667286
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -8537817961850002460
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -8455548651502146297
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -7573251885022648201
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -7283998229728427086
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -5079590464076906418
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -4973895529674346757
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -4798732461090414614
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -4150674888532448987
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -3357484057065401244
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -3247046098722086989
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -2496621711492695560
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -2015378652588483945
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -2008240956777703774
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -1742717317356114895
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -1649631079583406291
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             -1362372080384919957
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             535552015990656401
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             1386531744029322441
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             4084737172491554301
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             4156276528481256278
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             5661877716474387460
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             6368995327320475446
127.0.0.1  rack1       Up     Normal  90.99 KiB       100.00%             9217196359362328551

  Warning: "nodetool ring" is used to output all the tokens of a node.
  To view status related info of a node use "nodetool status" instead.


hduser@muhammad-vm:/usr/local/cassandra$
```

6) Cassandra Cluster Manager: If you want an even faster way to install Cassandra, you can use an open-source tool called **CCM**. **CCM** installs Cassandra for you, with very minimal configuration. In addition to ease of installation, CCM also allows you to run multiple Cassandra nodes locally.

Open a new terminal by pressing (ctrl + Alt + t). Install python 2.7 before execution of next commands.

```
$sudo apt install python2
```

Press Y for the installation.



hduser@muhammad-VB: ~

```
hduser@muhammad-VB:~$ sudo apt install python2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python2-minimal
  python2.7 python2.7-minimal
Suggested packages:
  python2-doc python-tk python2.7-doc binfmt-support
The following NEW packages will be installed:
  libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib python2
  python2-minimal python2.7 python2.7-minimal
0 upgraded, 7 newly installed, 0 to remove and 181 not upgraded.
Need to get 4,005 kB of archives.
After this operation, 16.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ie.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libpython2.7-minimal amd64 2.7.18-1
3ubuntu1.1 [347 kB]
Get:2 http://ie.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python2.7-minimal amd64 2.7.18-13ub
untu1.1 [1,394 kB]
hduser@muhammad-VB:~$ python2 --version
Python 2.7.18
hduser@muhammad-VB:~$
```

First, let's clone the CCM repository from GitHub, and cd into the directory:

```
$cd
```

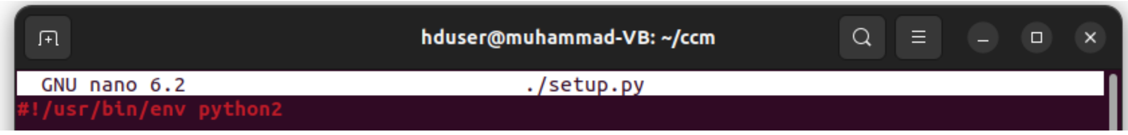
<- Press enter Key

```
$sudo apt install git
$git clone https://github.com/riptano/ccm.git
$cd ccm
```

Next, we will run the setup program to install CCM:

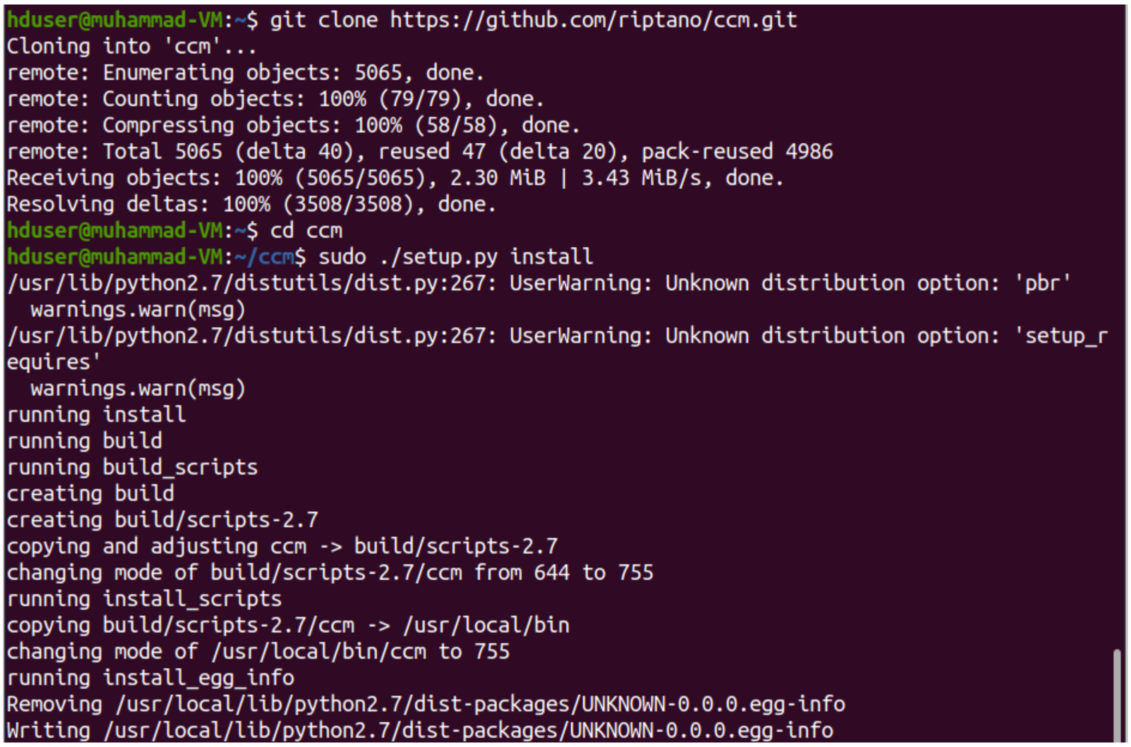
```
$nano ./setup.py
```

And change the first line word "python" to "python2"as mentioned below



```
  GNU nano 6.2                                    ./setup.py
#!/usr/bin/env python2
```

```
$sudo ./setup.py install
```



```
hduser@muhammad-VM:~$ git clone https://github.com/riptano/ccm.git
Cloning into 'ccm'...
remote: Enumerating objects: 5065, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (58/58), done.
remote: Total 5065 (delta 40), reused 47 (delta 20), pack-reused 4986
Receiving objects: 100% (5065/5065), 2.30 MiB | 3.43 MiB/s, done.
Resolving deltas: 100% (3508/3508), done.
hduser@muhammad-VM:~$ cd ccm
hduser@muhammad-VM:~/ccm$ sudo ./setup.py install
/usr/lib/python2.7/distutils/dist.py:267: UserWarning: Unknown distribution option: 'pbr'
  warnings.warn(msg)
/usr/lib/python2.7/distutils/dist.py:267: UserWarning: Unknown distribution option: 'setup_r
equires'
  warnings.warn(msg)
running install
running build
running build_scripts
creating build
creating build/scripts-2.7
copying and adjusting ccm -> build/scripts-2.7
changing mode of build/scripts-2.7/ccm from 644 to 755
running install_scripts
copying build/scripts-2.7/ccm -> /usr/local/bin
changing mode of /usr/local/bin/ccm to 755
running install_egg_info
Removing /usr/local/lib/python2.7/dist-packages/UNKNOWN-0.0.0.egg-info
Writing /usr/local/lib/python2.7/dist-packages/UNKNOWN-0.0.0.egg-info
```

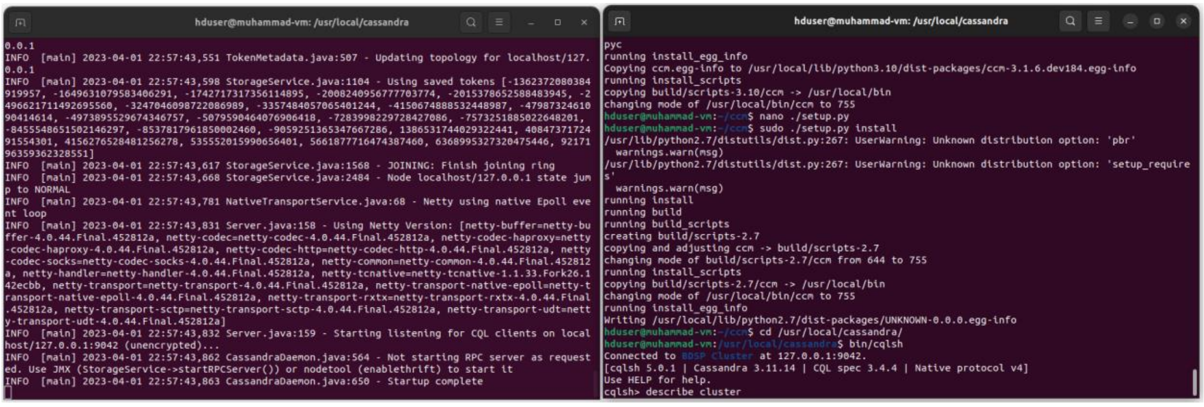
A quick introduction to the data model

Now that we have a Cassandra cluster running on our local machine, we will demonstrate its use with some quick examples. We will start with **cqlsh**, and use that as our primary means of working with the Cassandra data model.

7) Using Cassandra with **cqlsh**: To start working with Cassandra, let's start the **Cassandra Query Language** (**CQL**) shell. The shell interface will allow us to execute CQL commands to define, query, and modify our data. As this is a new cluster and we have turned on authentication and authorization, we will use the default cassandra and cassandra username and password, as follows:

```
$cd /usr/local/cassandra/
```

`$bin/cqlsh`

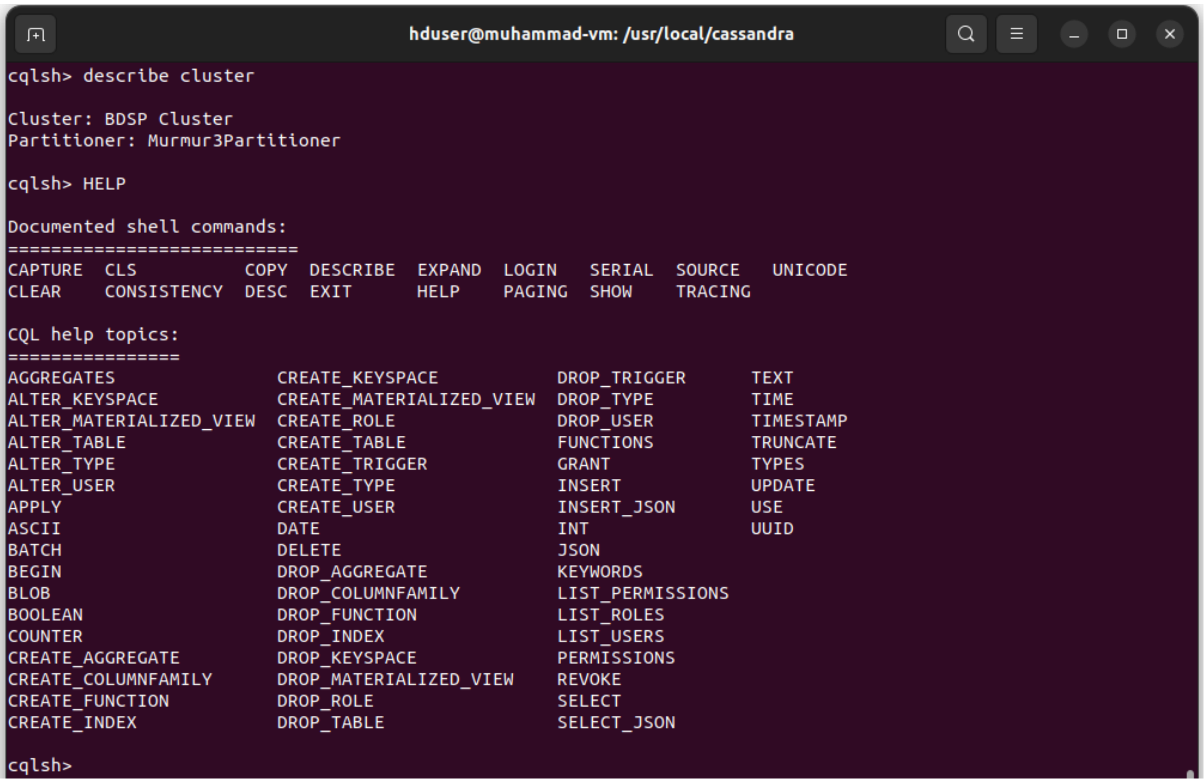

hduser@muhammad-vm: /usr/local/cassandra
hduser@muhammad-vm: /usr/local/cassandra

```
0.0.1
INFO  [main] 2023-04-01 22:57:43,551 TokenMetadata.java:507 - Updating topology for localhost/127.
0.0.1
INFO  [main] 2023-04-01 22:57:43,598 StorageService.java:1104 - Using saved tokens [-1362372080384
919957, -1649631079583406291, -1742717317356114895, -2008240956777703774, -2015378652588483945, -2
496621711492695560, -3247046098722086989, -3357484057065401244, -4150674888532448987, -47987324610
90414614, -4973895529674346757, -5079590464076906418, -7283998229728427086, -7573251885022648201,
-8455548651502146297, -8537817961850002460, -9059251365347667286, 1386531744029322441, 40847371724
91554301, 4156276528481256278, 535552015990656401, 5661877716474387460, 6368995327320475446, 92171
96359362328551]
INFO  [main] 2023-04-01 22:57:43,617 StorageService.java:1568 - JOINING: Finish joining ring
INFO  [main] 2023-04-01 22:57:43,668 StorageService.java:2484 - Node localhost/127.0.0.1 state jum
p to NORMAL
INFO  [main] 2023-04-01 22:57:43,781 NativeTransportService.java:68 - Netty using native Epoll eve
nt loop
INFO  [main] 2023-04-01 22:57:43,831 Server.java:158 - Using Netty Version: [netty-buffer=netty-bu
ffer-4.0.44.Final.452812a, netty-codec=netty-codec-4.0.44.Final.452812a, netty-codec-haproxy=netty
-codec-haproxy-4.0.44.Final.452812a, netty-codec-http=netty-codec-http-4.0.44.Final.452812a, netty
-codec-socks=netty-codec-socks-4.0.44.Final.452812a, netty-common=netty-common-4.0.44.Final.452812
a, netty-handler=netty-handler-4.0.44.Final.452812a, netty-tcnative=netty-tcnative-1.1.33.Fork26.1
42ecbb, netty-transport=netty-transport-4.0.44.Final.452812a, netty-transport-native-epoll=netty-t
ransport-native-epoll-4.0.44.Final.452812a, netty-transport-rxtx=netty-transport-rxtx-4.0.44.Final
.452812a, netty-transport-sctp=netty-transport-sctp-4.0.44.Final.452812a, netty-transport-udt=nett
y-transport-udt-4.0.44.Final.452812a]
INFO  [main] 2023-04-01 22:57:43,832 Server.java:159 - Starting listening for CQL clients on local
host/127.0.0.1:9042 (unencrypted)...
INFO  [main] 2023-04-01 22:57:43,862 CassandraDaemon.java:564 - Not starting RPC server as request
ed. Use JMX (StorageService->startRPCServer()) or nodetool (enablethrift) to start it
INFO  [main] 2023-04-01 22:57:43,863 CassandraDaemon.java:650 - Startup complete
```

```
pyc
running install_egg_info
Copying ccm.egg-info to /usr/local/lib/python3.10/dist-packages/ccm-3.1.6.dev184.egg-info
running install_scripts
copying build/scripts-3.10/ccm -> /usr/local/bin
changing mode of /usr/local/bin/ccm to 755
hduser@muhammad-vm:~/ccm$ nano ./setup.py
hduser@muhammad-vm:~/ccm$ sudo ./setup.py install
/usr/lib/python2.7/distutils/dist.py:267: UserWarning: Unknown distribution option: 'pbr'
  warnings.warn(msg)
/usr/lib/python2.7/distutils/dist.py:267: UserWarning: Unknown distribution option: 'setup_require
s'
  warnings.warn(msg)
running install
running build
running build_scripts
creating build/scripts-2.7
copying and adjusting ccm -> build/scripts-2.7
changing mode of build/scripts-2.7/ccm from 644 to 755
running install_scripts
copying build/scripts-2.7/ccm -> /usr/local/bin
changing mode of /usr/local/bin/ccm to 755
running install_egg_info
Writing /usr/local/lib/python2.7/dist-packages/UNKNOWN-0.0.0.egg-info
hduser@muhammad-vm:~/ccm$ cd /usr/local/cassandra/
hduser@muhammad-vm:/usr/local/cassandra$ bin/cqlsh
Connected to BDSP Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.14 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> describe cluster
```

One terminal showed that the Cassandra is running, and you can execute the Cassandra database commands on the other terminal.

`cassandra@cqlsh> describe cluster;`



```
cqlsh> describe cluster

Cluster: BDSP Cluster
Partitioner: Murmur3Partitioner

cqlsh> HELP

Documented shell commands:
===========================
CAPTURE  CLS          COPY  DESCRIBE  EXPAND  LOGIN   SERIAL  SOURCE   UNICODE
CLEAR    CONSISTENCY  DESC  EXIT      HELP    PAGING  SHOW    TRACING

CQL help topics:
================
AGGREGATES               CREATE_KEYSPACE           DROP_TRIGGER      TEXT
ALTER_KEYSPACE           CREATE_MATERIALIZED_VIEW  DROP_TYPE         TIME
ALTER_MATERIALIZED_VIEW  CREATE_ROLE               DROP_USER         TIMESTAMP
ALTER_TABLE              CREATE_TABLE              FUNCTIONS         TRUNCATE
ALTER_TYPE               CREATE_TRIGGER            GRANT             TYPES
ALTER_USER               CREATE_TYPE               INSERT            UPDATE
APPLY                    CREATE_USER               INSERT_JSON       USE
ASCII                    DATE                      INT               UUID
BATCH                    DELETE                    JSON
BEGIN                    DROP_AGGREGATE            KEYWORDS
BLOB                     DROP_COLUMNFAMILY         LIST_PERMISSIONS
BOOLEAN                  DROP_FUNCTION             LIST_ROLES
COUNTER                  DROP_INDEX                LIST_USERS
CREATE_AGGREGATE         DROP_KEYSPACE             PERMISSIONS
CREATE_COLUMNFAMILY      DROP_MATERIALIZED_VIEW    REVOKE
CREATE_FUNCTION          DROP_ROLE                 SELECT
CREATE_INDEX             DROP_TABLE                SELECT_JSON

cqlsh>
```

`cqlsh>DESCRIBE KEYSPACES;`

Check all the tables that are defined in the keyspace.

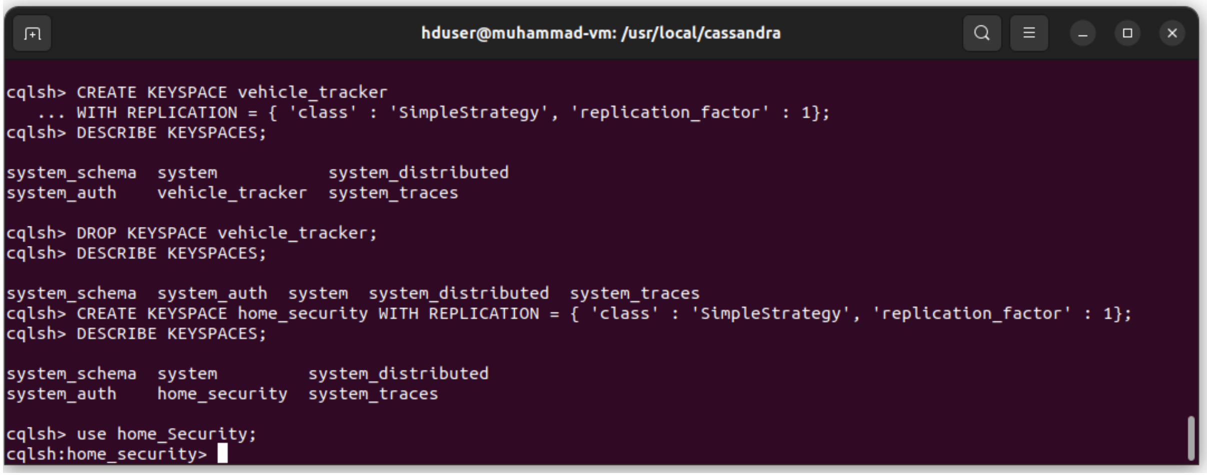
`cqlsh>DESCRIBE KEYSPACE system;`

`cqlsh>CREATE KEYSPACE vehicle_tracker WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };`

`cqlsh>DESCRIBE KEYSPACES;`

Check the screenshot of this command on the next page of the tutorial. If you like to drop the keyspace

`cqlsh>DROP KEYSPACE vehicle_tracker;`

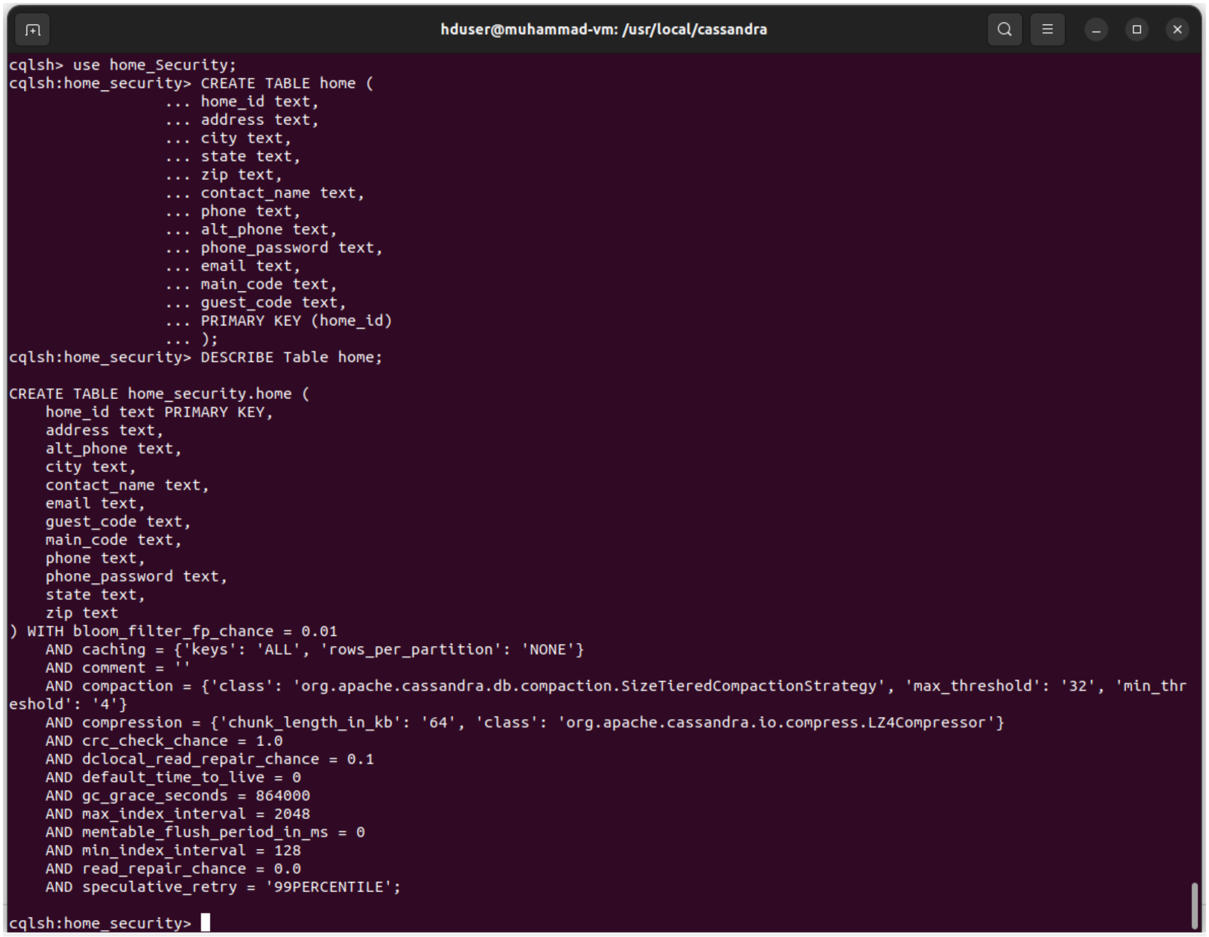


```
cqlsh> CREATE KEYSPACE vehicle_tracker
   ... WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1};
cqlsh> DESCRIBE KEYSPACES;

system_schema  system           system_distributed
system_auth    vehicle_tracker  system_traces

cqlsh> DROP KEYSPACE vehicle_tracker;
cqlsh> DESCRIBE KEYSPACES;

system_schema  system_auth  system  system_distributed  system_traces
cqlsh> CREATE KEYSPACE home_security WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1};
cqlsh> DESCRIBE KEYSPACES;

system_schema  system         system_distributed
system_auth    home_security  system_traces

cqlsh> use home_Security;
cqlsh:home_security> 
```

If you would like to know the details of the commands, please check the website: https://cassandra.apache.org/doc/latest/cassandra/developing/cql/ddl.html

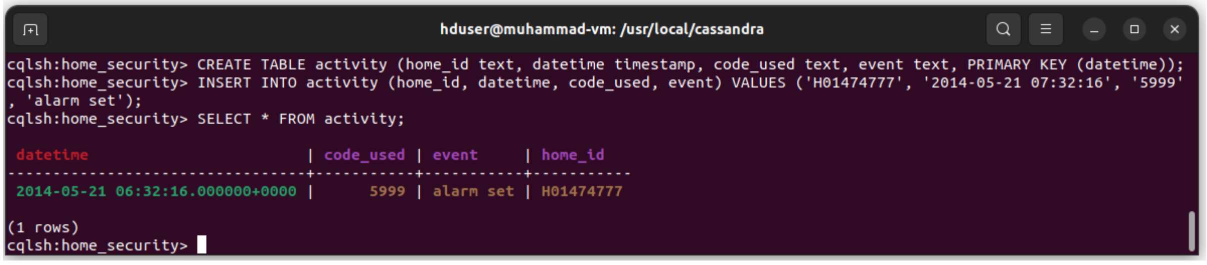
`cqlsh>USE home_Security;`

Follow the screenshot to create the Table in the collection **`'home_security'`**



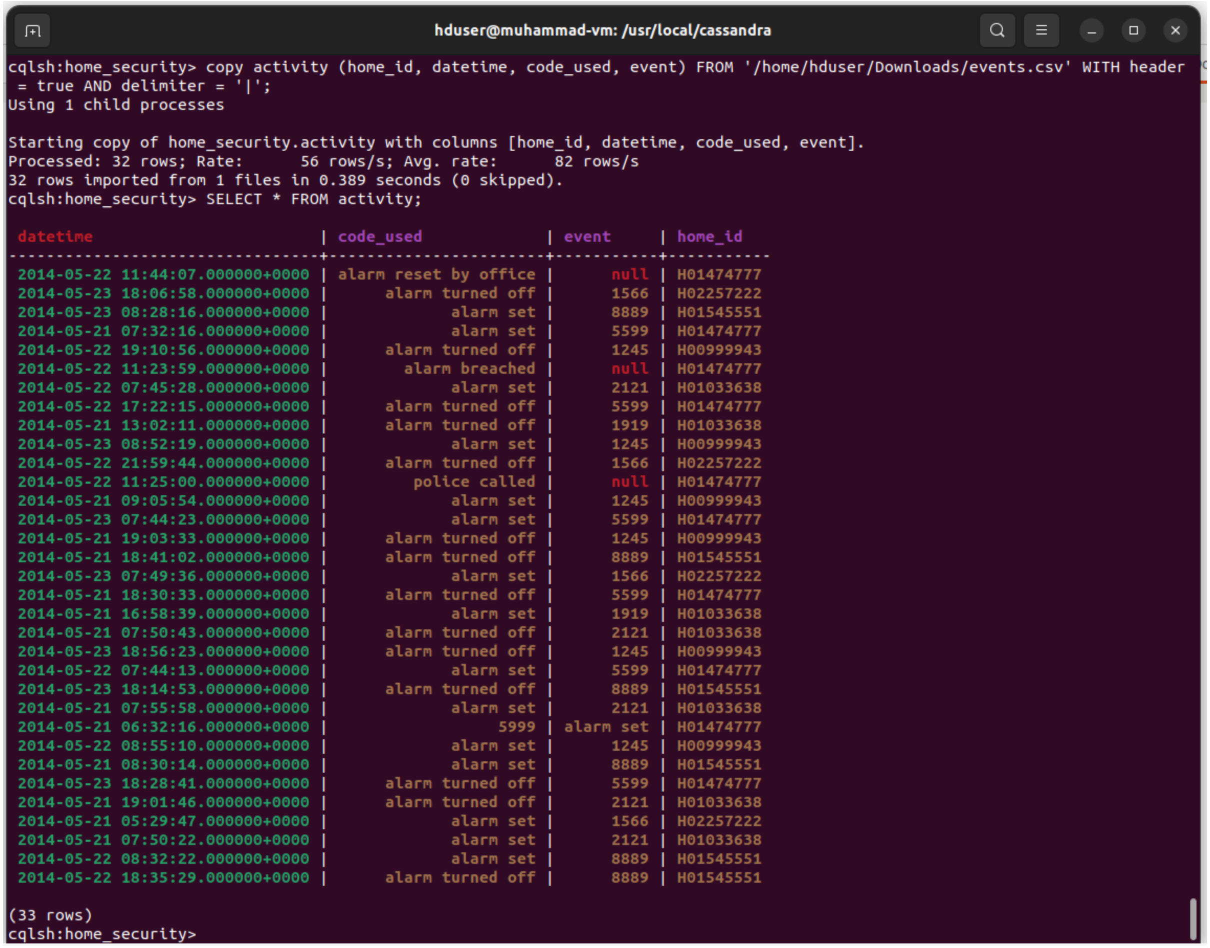
```
cqlsh> use home_Security;
cqlsh:home_security> CREATE TABLE home (
                 ... home_id text,
                 ... address text,
                 ... city text,
                 ... state text,
                 ... zip text,
                 ... contact_name text,
                 ... phone text,
                 ... alt_phone text,
                 ... phone_password text,
                 ... email text,
                 ... main_code text,
                 ... guest_code text,
                 ... PRIMARY KEY (home_id)
                 ... );
cqlsh:home_security> DESCRIBE Table home;

CREATE TABLE home_security.home (
    home_id text PRIMARY KEY,
    address text,
    alt_phone text,
    city text,
    contact_name text,
    email text,
    guest_code text,
    main_code text,
    phone text,
    phone_password text,
    state text,
    zip text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_thr
eshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';

cqlsh:home_security> 
```

Create another table named as 'activity' inside the collection 'home_security' and the screenshots are mentioned below

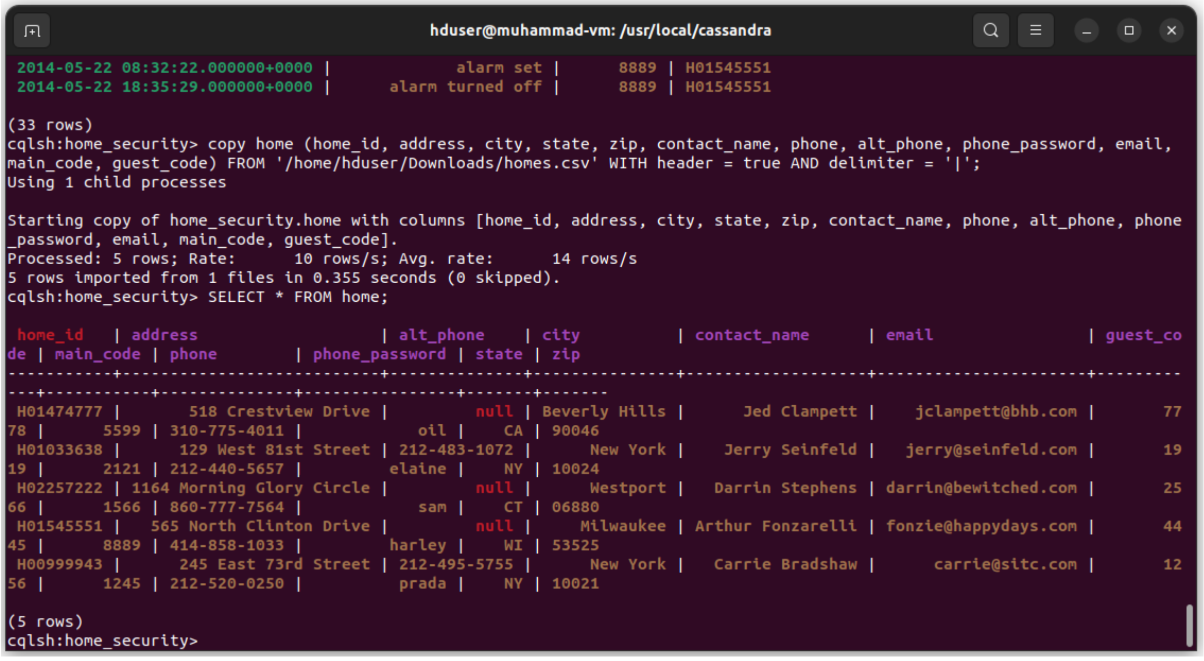


```
cqlsh:home_security> CREATE TABLE activity (home_id text, datetime timestamp, code_used text, event text, PRIMARY KEY (datetime));
cqlsh:home_security> INSERT INTO activity (home_id, datetime, code_used, event) VALUES ('H01474777', '2014-05-21 07:32:16', '5999'
, 'alarm set');
cqlsh:home_security> SELECT * FROM activity;

 datetime                        | code_used | event     | home_id
---------------------------------+-----------+-----------+-----------
 2014-05-21 06:32:16.000000+0000 |      5999 | alarm set | H01474777

(1 rows)
cqlsh:home_security> 
```

8) Copy the data from **csv** file. Download the file **'events.csv'** and **'homes.csv'** from Moodle in the **'Downloads'** folder on VM (This is not Hadoop and it is your local Ubuntu machine) and write the command as mentioned in the screenshot.

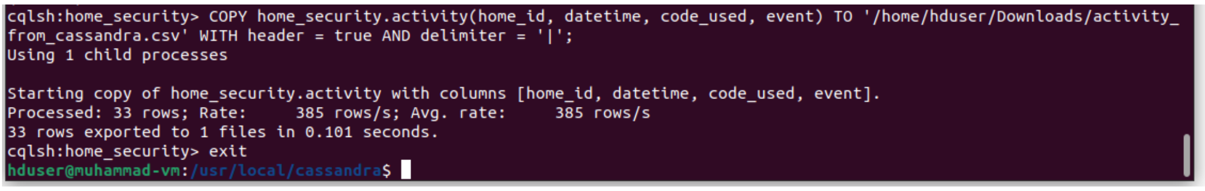


```
cqlsh:home_security> copy activity (home_id, datetime, code_used, event) FROM '/home/hduser/Downloads/events.csv' WITH header
 = true AND delimiter = '|';
Using 1 child processes

Starting copy of home_security.activity with columns [home_id, datetime, code_used, event].
Processed: 32 rows; Rate:      56 rows/s; Avg. rate:      82 rows/s
32 rows imported from 1 files in 0.389 seconds (0 skipped).
cqlsh:home_security> SELECT * FROM activity;

 datetime                        | code_used             | event     | home_id
---------------------------------+-----------------------+-----------+-----------
 2014-05-22 11:44:07.000000+0000 | alarm reset by office |      null | H01474777
 2014-05-23 18:06:58.000000+0000 |      alarm turned off |      1566 | H02257222
 2014-05-23 08:28:16.000000+0000 |             alarm set |      8889 | H01545551
 2014-05-21 07:32:16.000000+0000 |             alarm set |      5599 | H01474777
 2014-05-22 19:10:56.000000+0000 |      alarm turned off |      1245 | H00999943
 2014-05-22 11:23:59.000000+0000 |        alarm breached |      null | H01474777
 2014-05-22 07:45:28.000000+0000 |             alarm set |      2121 | H01033638
 2014-05-22 17:22:15.000000+0000 |      alarm turned off |      5599 | H01474777
 2014-05-21 13:02:11.000000+0000 |      alarm turned off |      1919 | H01033638
 2014-05-23 08:52:19.000000+0000 |             alarm set |      1245 | H00999943
 2014-05-22 21:59:44.000000+0000 |      alarm turned off |      1566 | H02257222
 2014-05-22 11:25:00.000000+0000 |         police called |      null | H01474777
 2014-05-21 09:05:54.000000+0000 |             alarm set |      1245 | H00999943
 2014-05-23 07:44:23.000000+0000 |             alarm set |      5599 | H01474777
 2014-05-21 19:03:33.000000+0000 |      alarm turned off |      1245 | H00999943
 2014-05-21 18:41:02.000000+0000 |      alarm turned off |      8889 | H01545551
 2014-05-23 07:49:36.000000+0000 |             alarm set |      1566 | H02257222
 2014-05-21 18:30:33.000000+0000 |      alarm turned off |      5599 | H01474777
 2014-05-21 16:58:39.000000+0000 |             alarm set |      1919 | H01033638
 2014-05-21 07:50:43.000000+0000 |      alarm turned off |      2121 | H01033638
 2014-05-23 18:56:23.000000+0000 |      alarm turned off |      1245 | H00999943
 2014-05-22 07:44:13.000000+0000 |             alarm set |      5599 | H01474777
 2014-05-23 18:14:53.000000+0000 |      alarm turned off |      8889 | H01545551
 2014-05-21 07:55:58.000000+0000 |             alarm set |      2121 | H01033638
 2014-05-21 06:32:16.000000+0000 |                  5999 | alarm set | H01474777
 2014-05-22 08:55:10.000000+0000 |             alarm set |      1245 | H00999943
 2014-05-21 08:30:14.000000+0000 |             alarm set |      8889 | H01545551
 2014-05-23 18:28:41.000000+0000 |      alarm turned off |      5599 | H01474777
 2014-05-21 19:01:46.000000+0000 |      alarm turned off |      2121 | H01033638
 2014-05-21 05:29:47.000000+0000 |             alarm set |      1566 | H02257222
 2014-05-21 07:50:22.000000+0000 |             alarm set |      2121 | H01033638
 2014-05-22 08:32:22.000000+0000 |             alarm set |      8889 | H01545551
 2014-05-22 18:35:29.000000+0000 |      alarm turned off |      8889 | H01545551

(33 rows)
cqlsh:home_security>
```



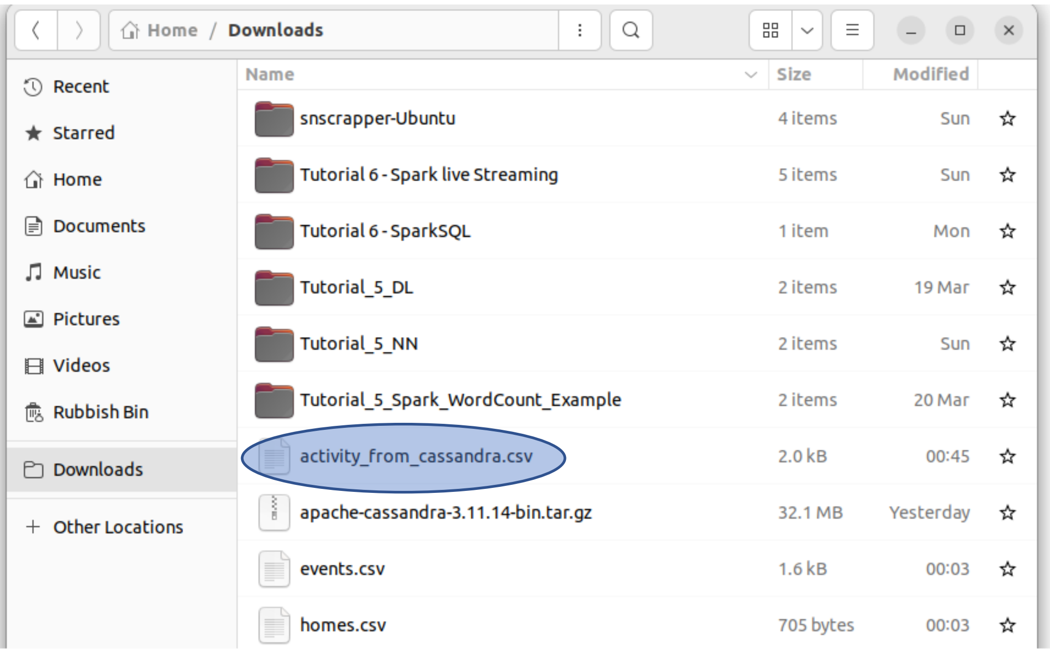
```
 2014-05-22 08:32:22.000000+0000 |          alarm set |      8889 | H01545551
 2014-05-22 18:35:29.000000+0000 |      alarm turned off |      8889 | H01545551

(33 rows)
cqlsh:home_security> copy home (home_id, address, city, state, zip, contact_name, phone, alt_phone, phone_password, email,
main_code, guest_code) FROM '/home/hduser/Downloads/homes.csv' WITH header = true AND delimiter = '|';
Using 1 child processes

Starting copy of home_security.home with columns [home_id, address, city, state, zip, contact_name, phone, alt_phone, phone
_password, email, main_code, guest_code].
Processed: 5 rows; Rate:      10 rows/s; Avg. rate:      14 rows/s
5 rows imported from 1 files in 0.355 seconds (0 skipped).
cqlsh:home_security> SELECT * FROM home;

 home_id   | address                   | alt_phone    | city          | contact_name      | email                | guest_co
de | main_code | phone        | phone_password | state | zip
-----------+---------------------------+--------------+---------------+-------------------+----------------------+---------
---+-----------+--------------+----------------+-------+-------
 H01474777 |       518 Crestview Drive |         null | Beverly Hills |      Jed Clampett |    jclampett@bhb.com |       77
78 |      5599 | 310-775-4011 |            oil |    CA | 90046
 H01033638 |      129 West 81st Street | 212-483-1072 |      New York |    Jerry Seinfeld |   jerry@seinfeld.com |       19
19 |      2121 | 212-440-5657 |         elaine |    NY | 10024
 H02257222 | 1164 Morning Glory Circle |         null |      Westport |   Darrin Stephens | darrin@bewitched.com |       25
66 |      1566 | 860-777-7564 |            sam |    CT | 06880
 H01545551 |   565 North Clinton Drive |         null |     Milwaukee | Arthur Fonzarelli | fonzie@happydays.com |       44
45 |      8889 | 414-858-1033 |         harley |    WI | 53525
 H00999943 |      245 East 73rd Street | 212-495-5755 |      New York |   Carrie Bradshaw |      carrie@sitc.com |       12
56 |      1245 | 212-520-0250 |          prada |    NY | 10021

(5 rows)
cqlsh:home_security>
```

9) Export the data from the Cassandra table to ‘csv’ file on your local Ubuntu machine.



```
cqlsh:home_security> COPY home_security.activity(home_id, datetime, code_used, event) TO '/home/hduser/Downloads/activity_
from_cassandra.csv' WITH header = true AND delimiter = '|';
Using 1 child processes

Starting copy of home_security.activity with columns [home_id, datetime, code_used, event].
Processed: 33 rows; Rate:      385 rows/s; Avg. rate:      385 rows/s
33 rows exported to 1 files in 0.101 seconds.
cqlsh:home_security> exit
hduser@muhammad-vm:/usr/local/cassandra$ 
```

The output file will be stored in ‘Downloads’ folder as shown below on Ubuntu VM. You might see some other files in the Download folder than this screenshot. Make sure that activity_from_cassandra.csv must be present.



Name	Size	Modified
snscrapper-Ubuntu	4 items	Sun
Tutorial 6 - Spark live Streaming	5 items	Sun
Tutorial 6 - SparkSQL	1 item	Mon
Tutorial_5_DL	2 items	19 Mar
Tutorial_5_NN	2 items	Sun
Tutorial_5_Spark_WordCount_Example	2 items	20 Mar
activity_from_cassandra.csv	2.0 kB	00:45
apache-cassandra-3.11.14-bin.tar.gz	32.1 MB	Yesterday
events.csv	1.6 kB	00:03
homes.csv	705 bytes	00:03

Steps to load data file from local ubuntu system to Cassandra Table

1) Create a file named as “employees_data.csv” and insert the records as mentioned below

```
$nano employees_data.csv
```

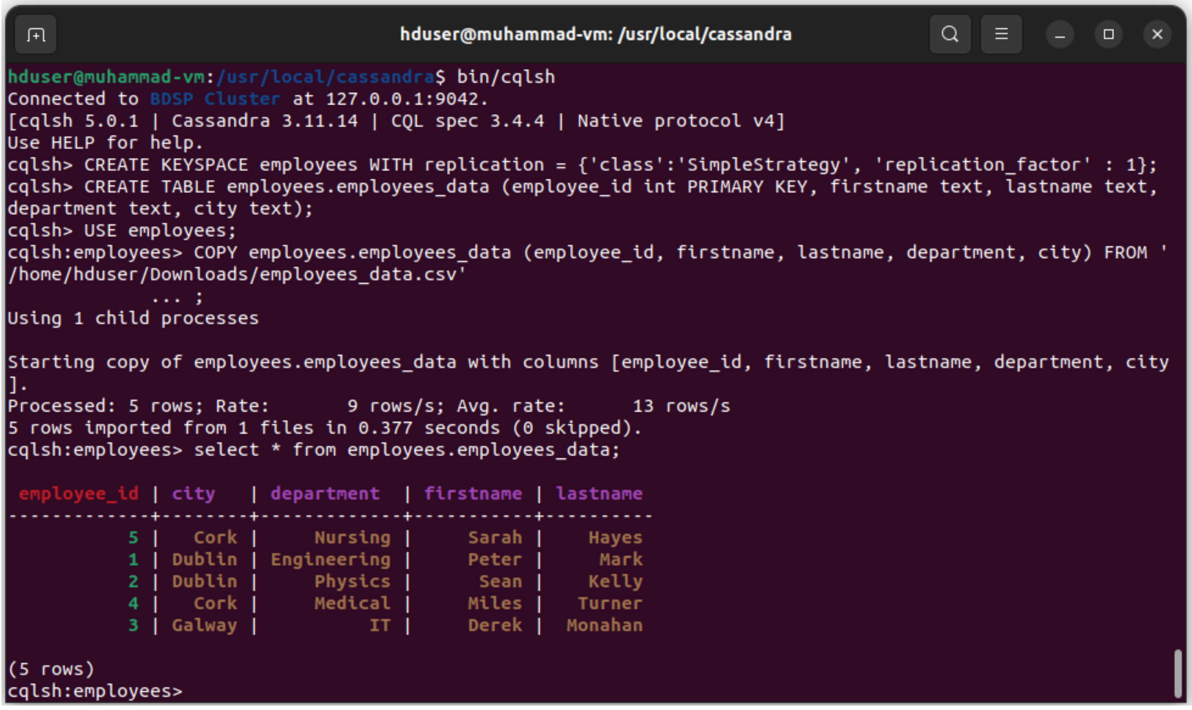
```
employee_id,firstname,lastname,department,city
1,Peter,Mark,Engineering,Dublin
2,Sean,Kelly,Physics,Dublin
3,Derek,Monahan,IT,Galway
4,Miles,Turner,Medical,Cork
5,Sarah,Hayes,Nursing,Cork
```

Or download the file “employees_data.csv” from Moodle.

2) Follow the sequence of commands to load data into Table “employees_data” and Keyspace named as “employees”.

```
$cd /usr/local/cassandra
hduser@muhammad-VM:/usr/local/cassandra$ bin/cqlsh
Connected to BDSP Cluster at 127.0.0.1:9042.
cqlsh> CREATE KEYSPACE employees WITH replication = {'class':'
SimpleStrategy', 'replication_factor' : 1};
cqlsh> CREATE TABLE employees.employees_data (employee_id int PRIMARY KEY,
firstname text, lastname text, department text, city text);
cqlsh> USE employees;
cqlsh:employees> COPY employees.employees_data (employee_id, firstname,
lastname, department, city) FROM '/home/hduser/Downloads/
employees_data.csv' WITH HEADER = true;
cqlsh:employees> select * from employees.employees_data;
```

3) The screenshot showed the sequence of commands as shown below.



```
hduser@muhammad-vm:/usr/local/cassandra$ bin/cqlsh
Connected to BDSP Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.14 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE employees WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 1};
cqlsh> CREATE TABLE employees.employees_data (employee_id int PRIMARY KEY, firstname text, lastname text,
department text, city text);
cqlsh> USE employees;
cqlsh:employees> COPY employees.employees_data (employee_id, firstname, lastname, department, city) FROM '
/home/hduser/Downloads/employees_data.csv'
             ... ;
Using 1 child processes

Starting copy of employees.employees_data with columns [employee_id, firstname, lastname, department, city
].
Processed: 5 rows; Rate:       9 rows/s; Avg. rate:      13 rows/s
5 rows imported from 1 files in 0.377 seconds (0 skipped).
cqlsh:employees> select * from employees.employees_data;

 employee_id | city   | department  | firstname | lastname
-------------+--------+-------------+-----------+----------
           5 |   Cork |     Nursing |     Sarah |    Hayes
           1 | Dublin | Engineering |     Peter |     Mark
           2 | Dublin |     Physics |      Sean |    Kelly
           4 |   Cork |     Medical |     Miles |   Turner
           3 | Galway |          IT |     Derek |  Monahan

(5 rows)
cqlsh:employees>
```

4) Learn from the book reference provided in references for further understanding of Cassandra query language and perform queries on the datasets of your choice.

References:

- https://cassandra.apache.org/doc/latest/cassandra/cql/ddl.html
- Cassandra: The Definitive Guide, (Revised) Third Edition, 3rd Edition, Jeff Carpenter, Eben Hewitt, O'Reilly Media, Inc., January 2022.
- Installation instructions: https://www.cloudduggu.com/cassandra/installation/