TY B.Tech. (CSE) – II [2022-23]

5CS372 : Advanced Database System Lab.

Assignment No. 10

Name: Khushi Nitinkumar Patel

PRN: 2020BTECS00037

Batch: T2

Title: Cassandra Clustering

Aim: To setup a multi-node Cassandra Cluster on single windows machine.

Introduction:

A graph database stores nodes and relationships instead of tables, or documents. Data is stored just like you might sketch ideas on a whiteboard. Your data is stored without restricting it to a pre-defined model, allowing a very flexible way of thinking about and using it.

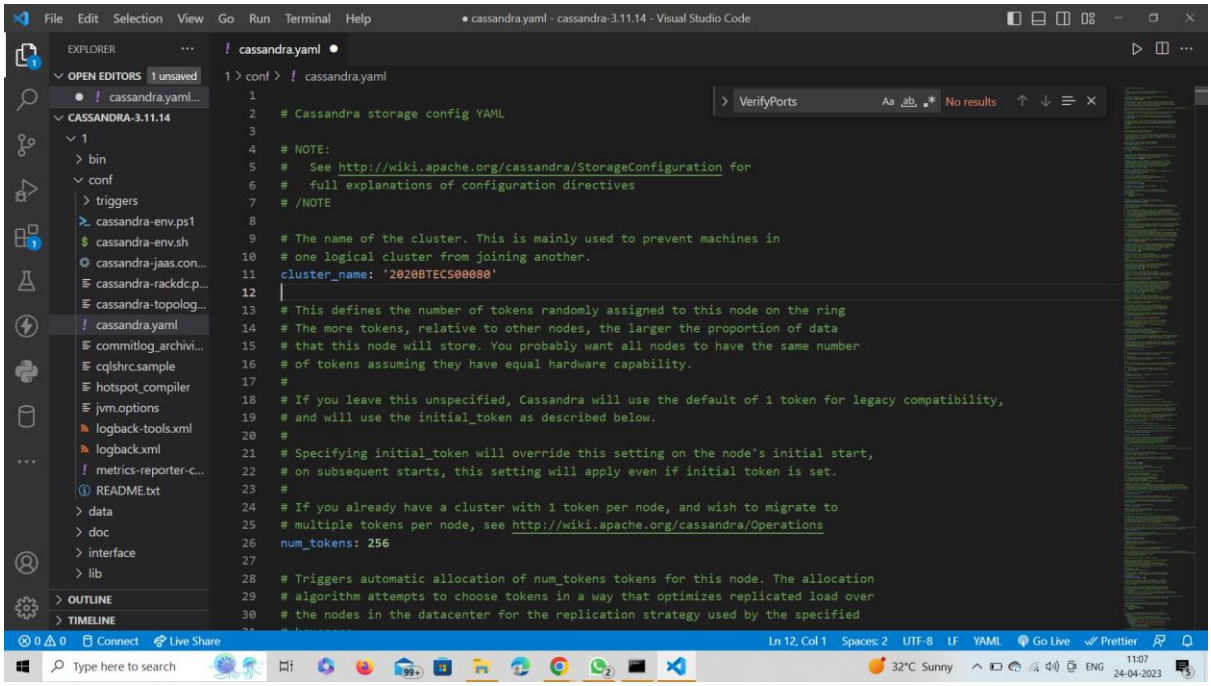
Theory:

Cassandra Clustering: In Cassandra, the data is distributed across a cluster. Additionally, a cluster may consist of a ring of nodes arranged in racks installed in data centers across geographical regions. At a more granular level, virtual nodes known as vnodes assign the data ownership to a physical machine.

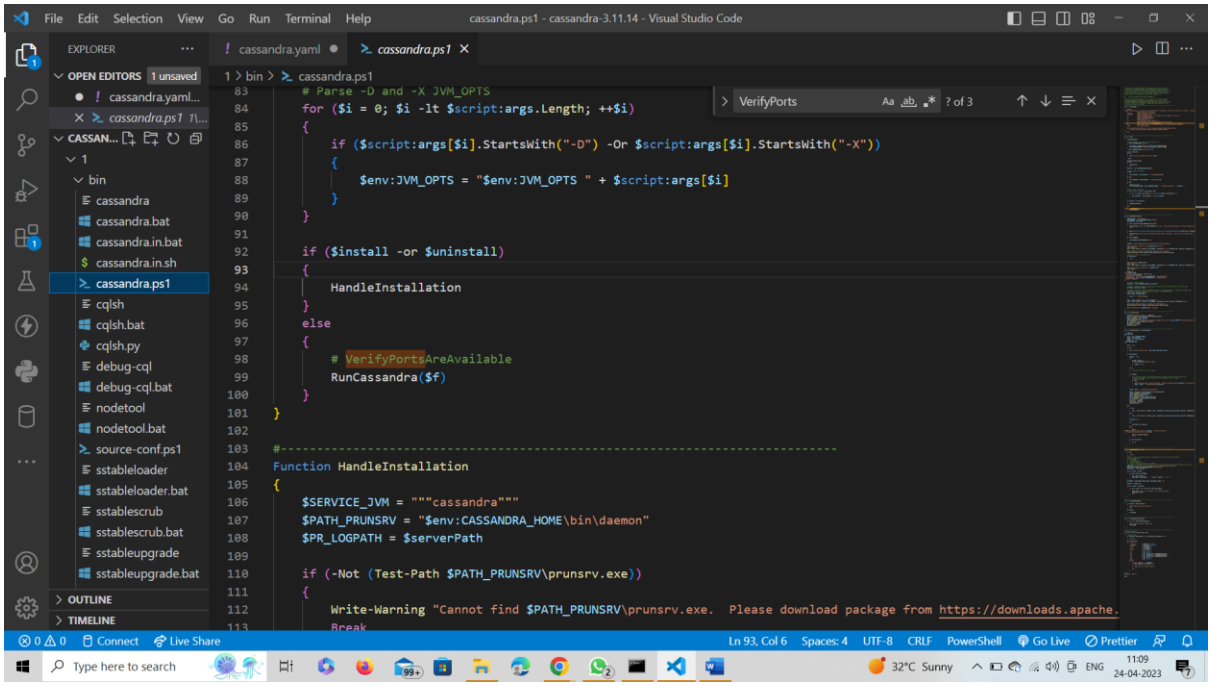
Cassandra **uses either a simple partition key or a composite partition key for a table with a compound primary key, and defines clustering column(s)**. A storage engine process, clustering sorts data based on the definition of the clustering columns within each partition.

Procedure:

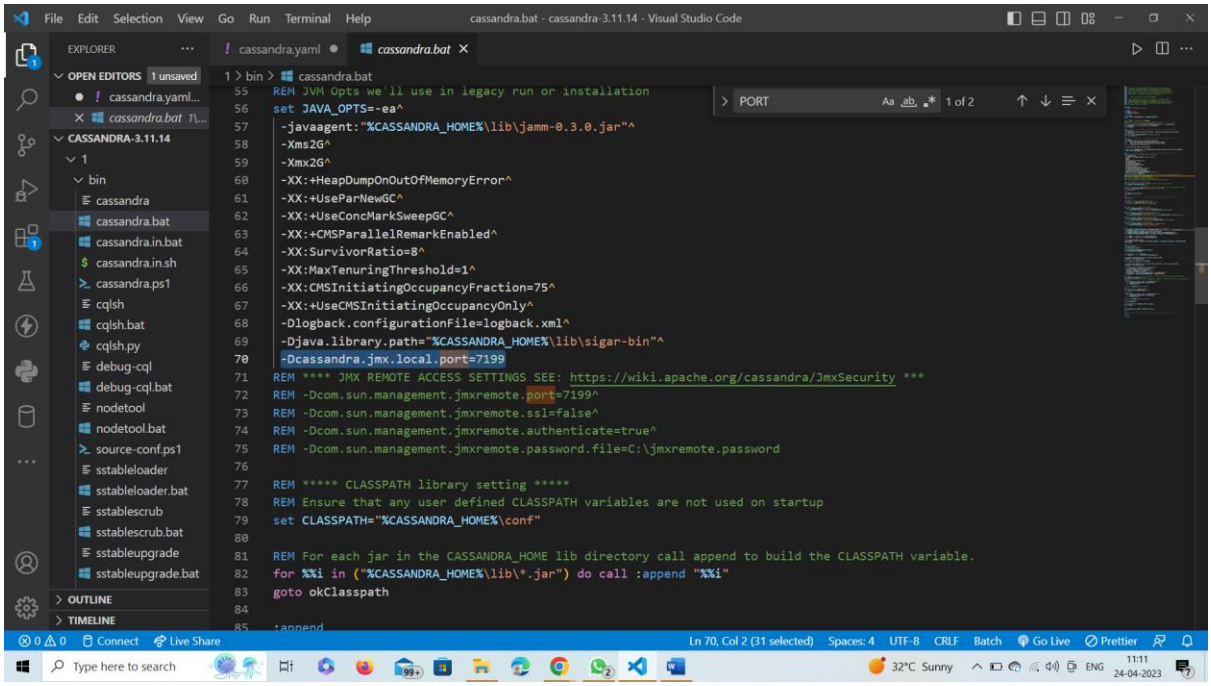
Name cluster:



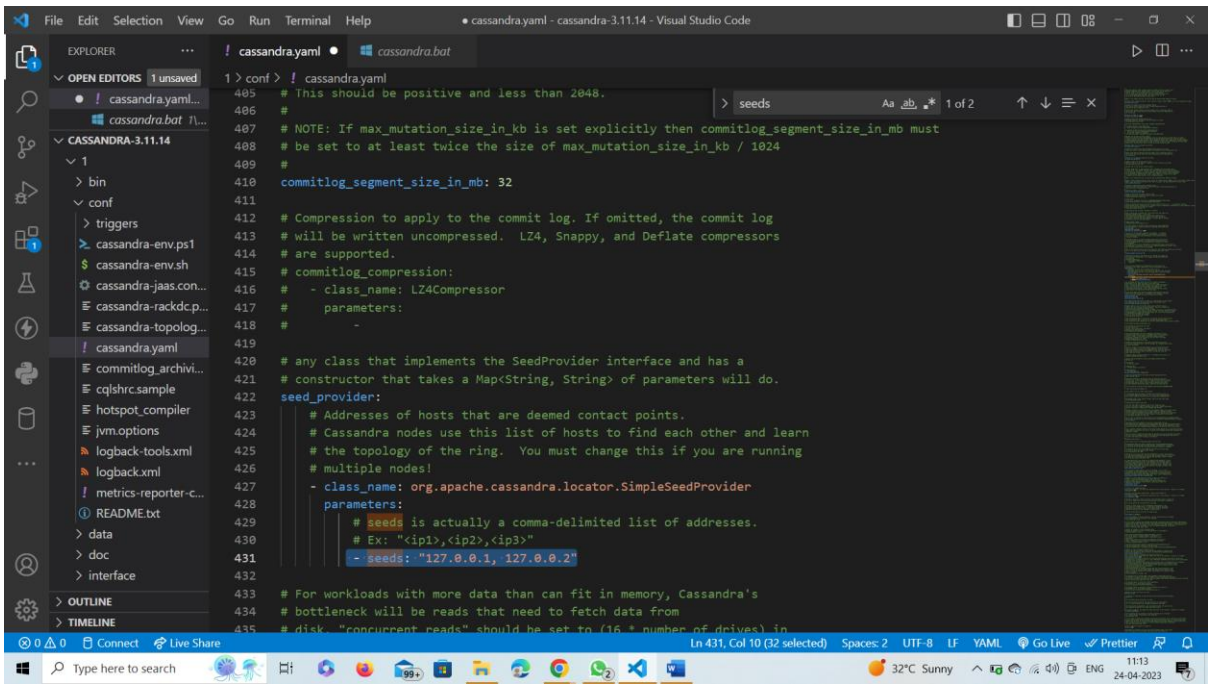
Comment VerifyPortsAreAvailable:



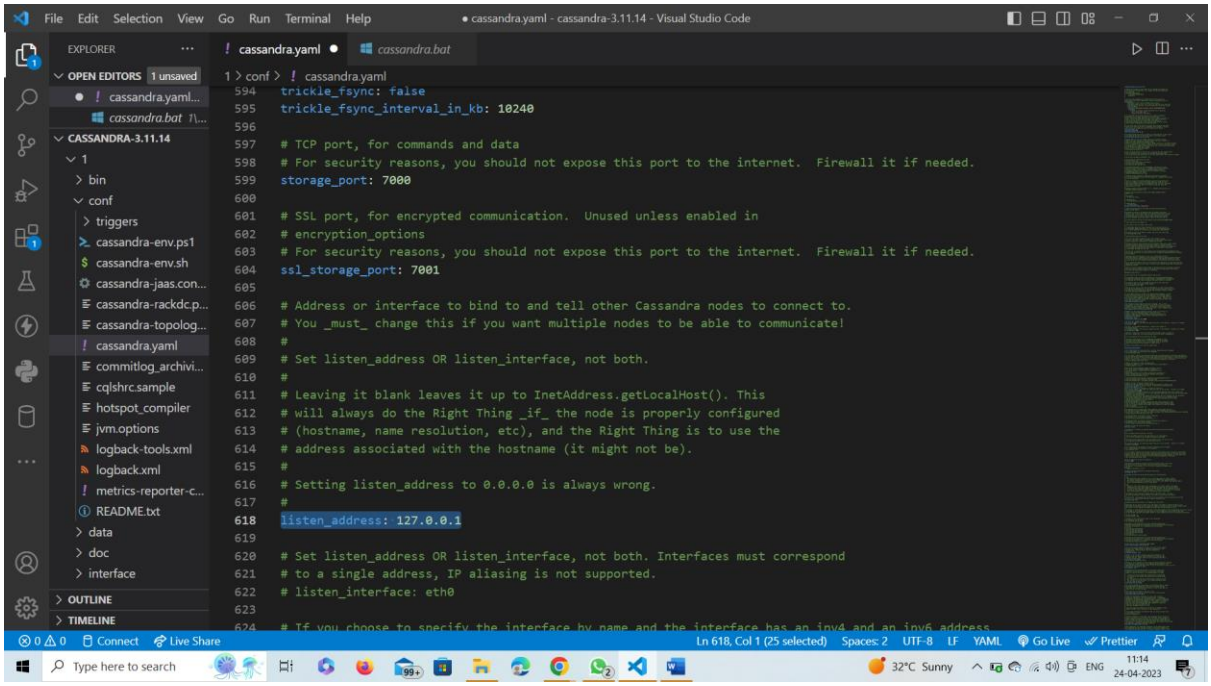
Port for 1st node is 7199:



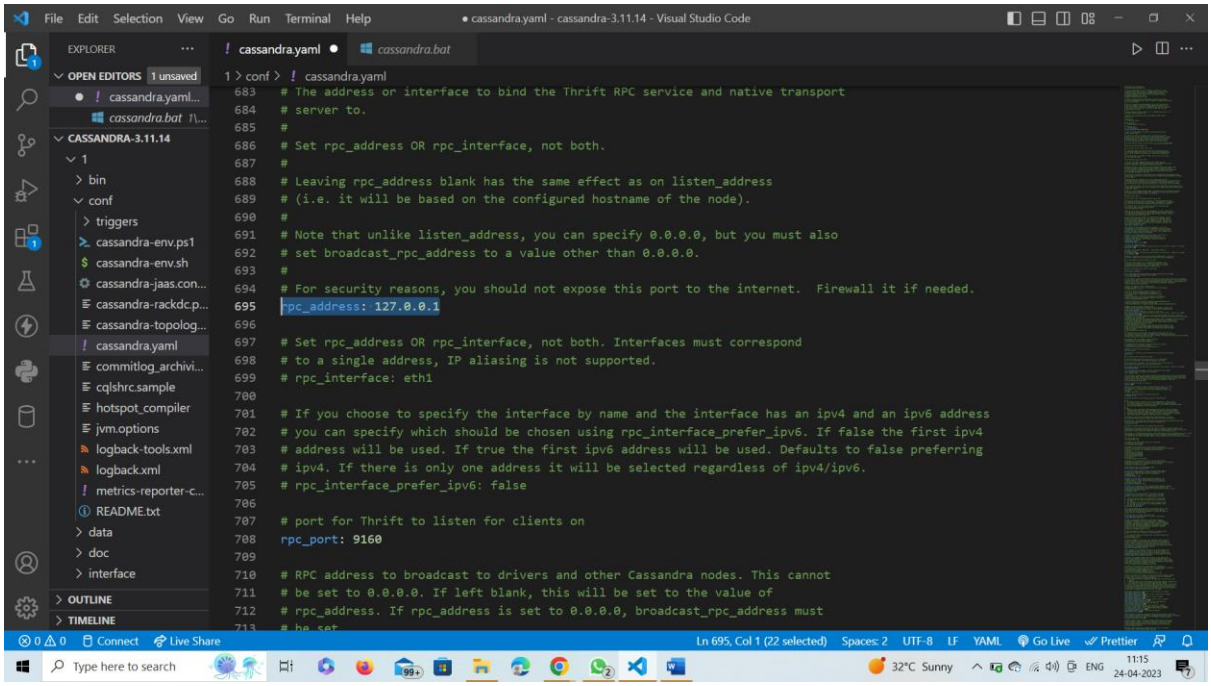
Seeds : 127.0.0.1, 127.0.0.2



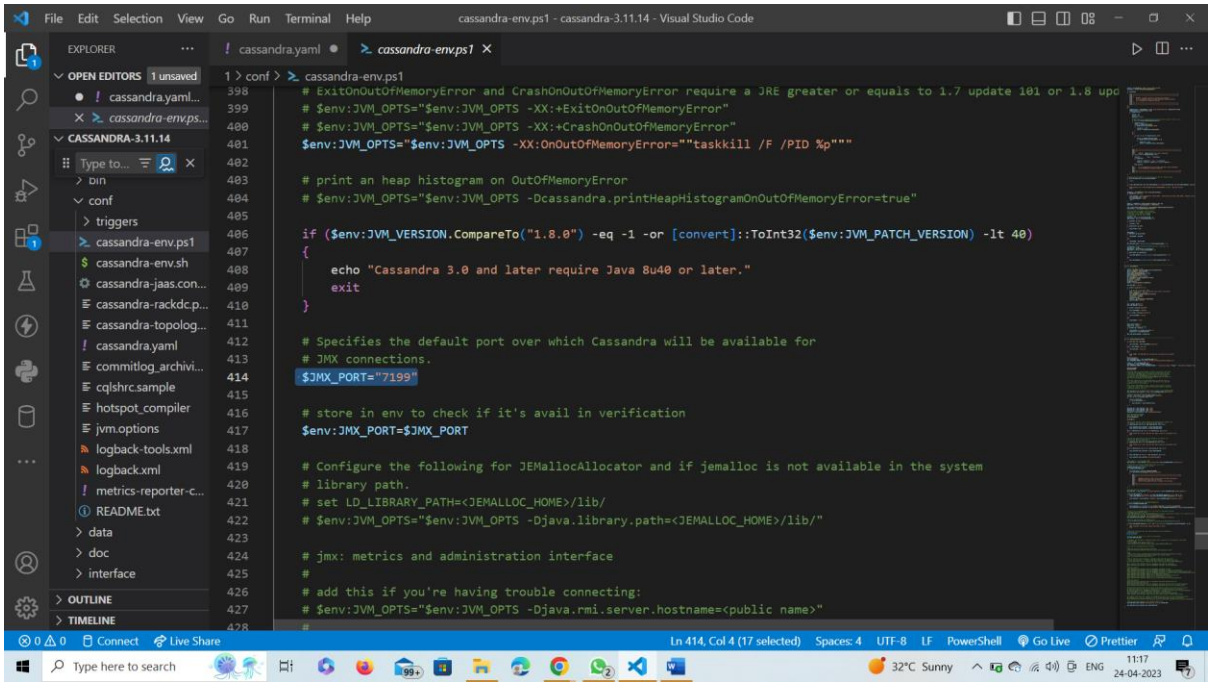
Listen address for 1st node: 127.0.0.1



Rpc address of 1st node: 127.0.0.1



$JMX_PORT="7199"



Make same changes in other 2 nodes.

Just following are node specific changes.

2nd node

:

Port for 2nd node is 7299:

Listen address for 2nd node: 127.0.0.2

Rpc address of 2nd node: 127.0.0.2

$JMX_PORT="7299"

3rd node:

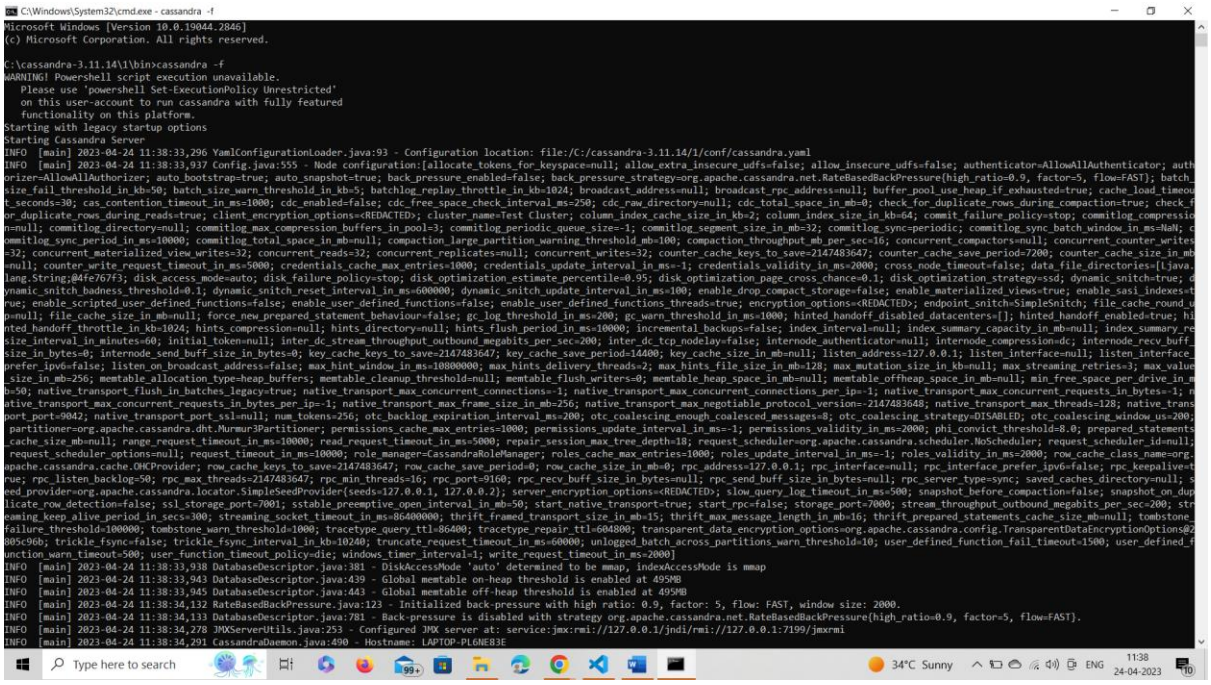
Port for 3rd node is 7399:

Listen address for 3rd node: 127.0.0.3

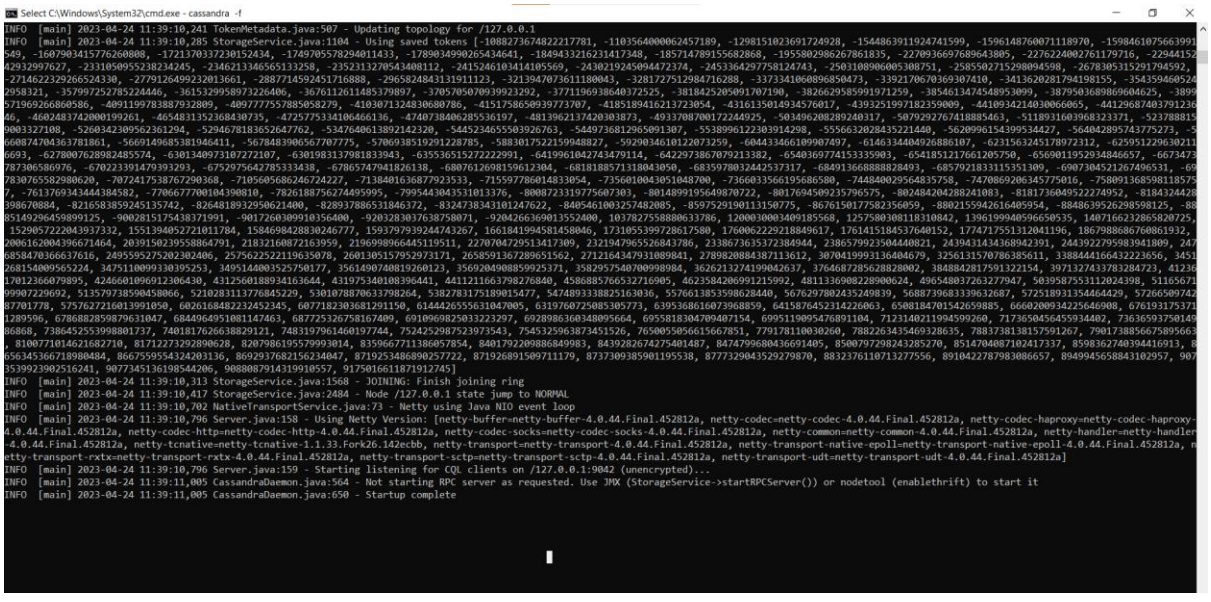
Rpc address of 3rd node: 127.0.0.3

$JMX_PORT="7399"

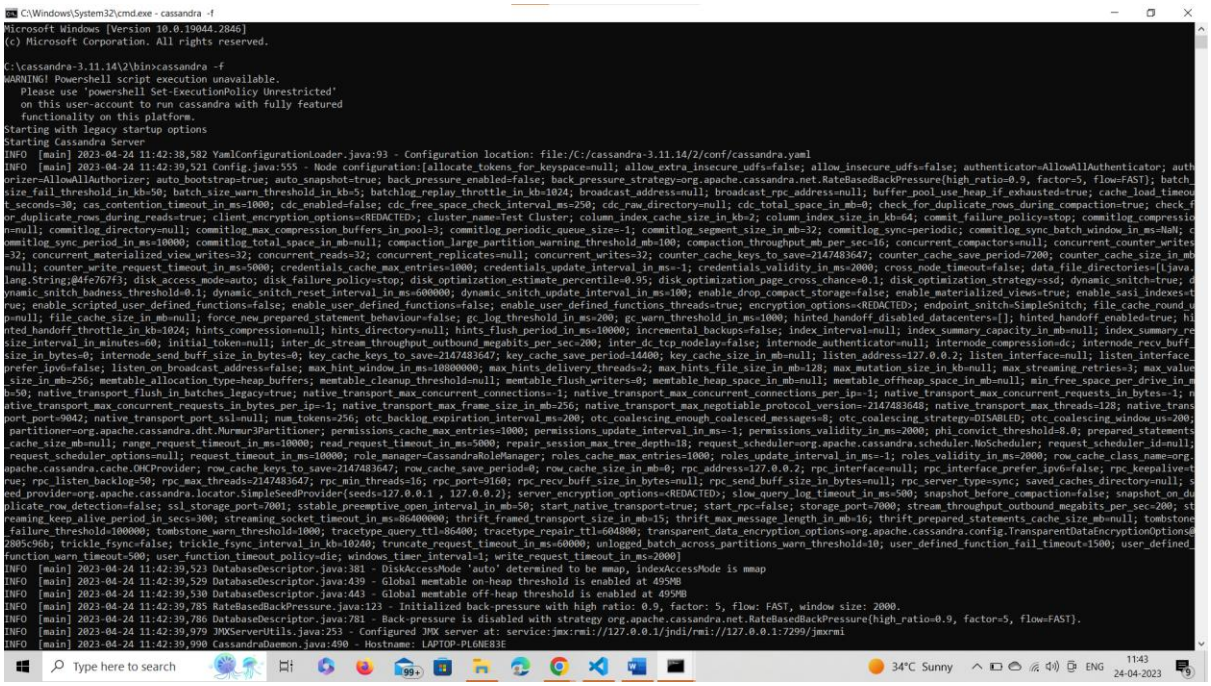
Starting node 1:



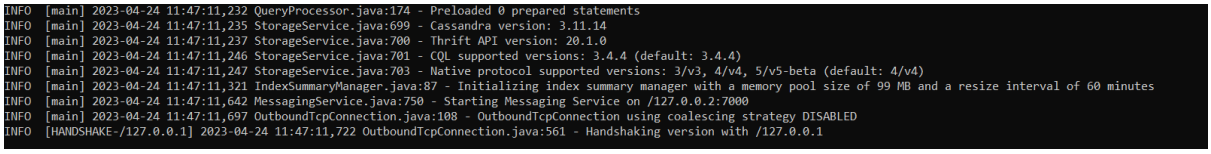
Node 1 started



Starting node 2:



Node 2 started:



Node 3 started:

3 Node cluster formed:

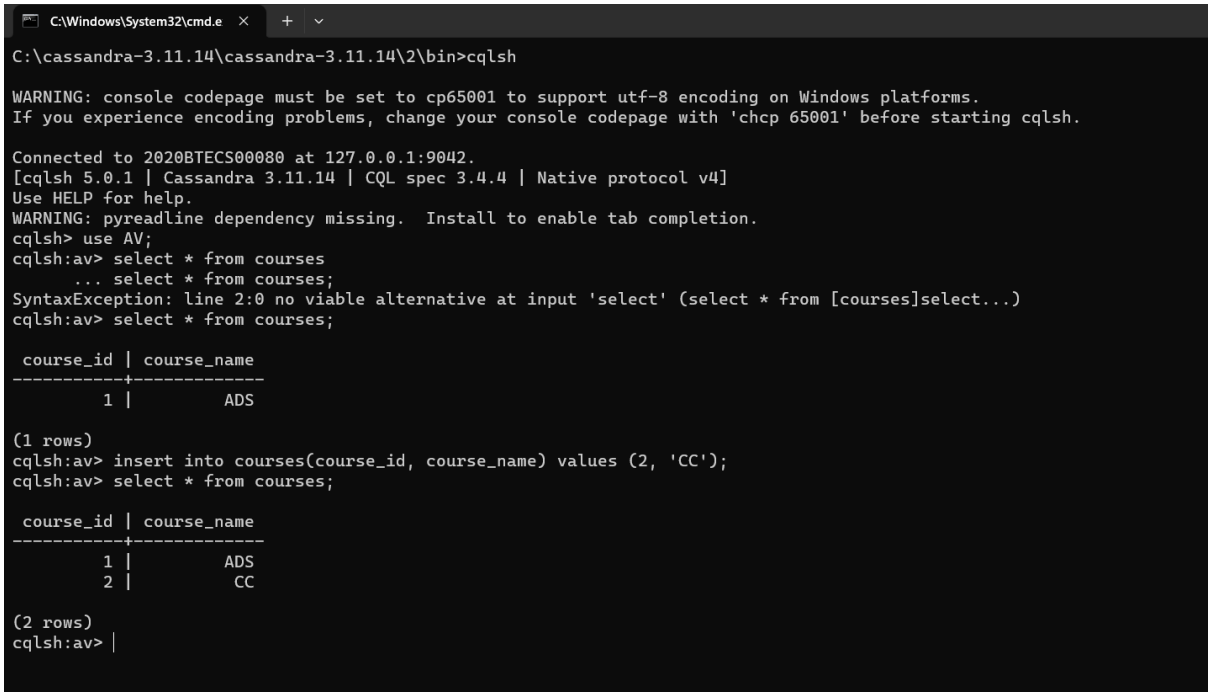
```
C:\cassandra-3.11.14\1\bin>nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address    Load       Tokens       Owns (effective)  Host ID                               Rack
UN  127.0.0.1  76.01 KiB  256          65.6%             47395b58-e7ad-42a4-9bd8-bff541e6f51c  rack1
UN  127.0.0.2  70.95 KiB  256          69.4%             3adcad71-f600-44bc-8541-6cc0fcf1d1e9  rack1
UN  127.0.0.3  70.94 KiB  256          65.0%             476c5b61-685f-499d-9e31-4daf8e5f0d8c  rack1
```

Creating Tables:

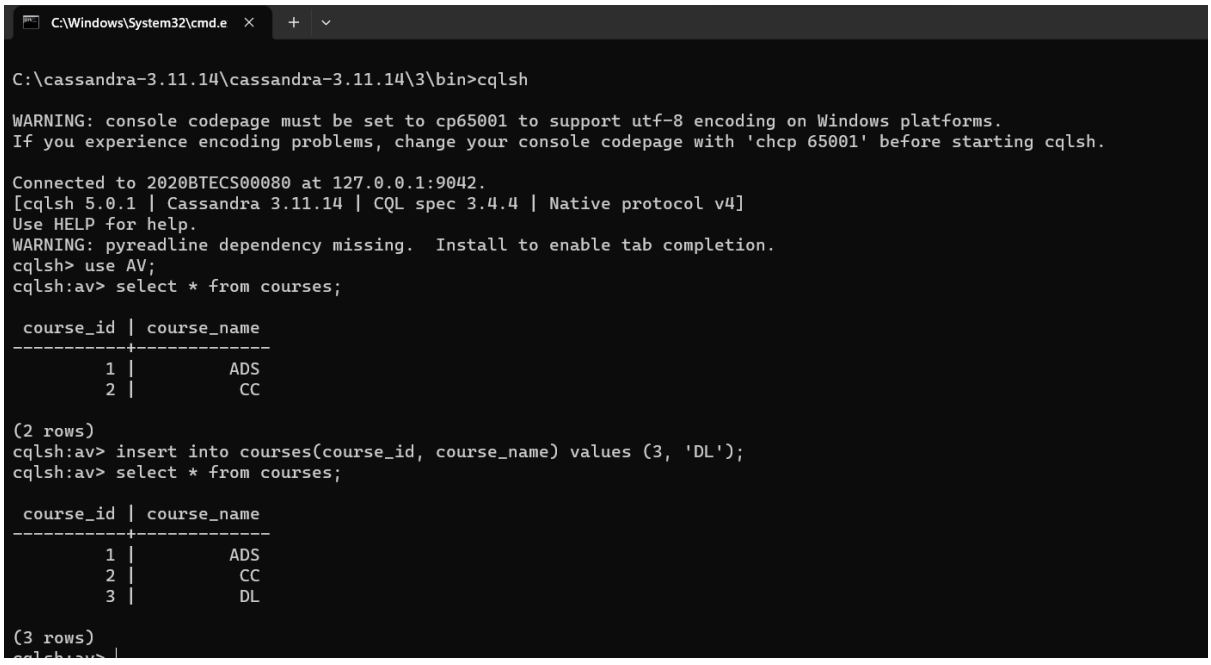
Node1

```
C:\Windows\System32\cmd.e

C:\cassandra-3.11.14\cassandra-3.11.14\1\bin>cqlsh

WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to 2020BTECS00080 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.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh> create keyspace AV with replication = {'class' : 'SimpleStrategy', 'replication_factor': 1};
cqlsh> use AV;
cqlsh:av> create table courses{ course_id int PRIMARY KEY, course_name text};
SyntaxException: line 1:20 no viable alternative at input '{' (create table [courses]{...)
cqlsh:av> create table courses( course_id int PRIMARY KEY, course_name text);
cqlsh:av> insert into courses(course_id, course_name) values (1, 'ADS');
cqlsh:av> select * from courses;

 course_id | course_name
-----------+-------------
         1 |         ADS

(1 rows)
cqlsh:av>
```

Accessible from node 2



Accessible from node 3



Conclusion: Clustering of 3 nodes in Cassandra successfully connected and implemented. Data is visible and accessible across all 3 nodes.