**Cassandra Clustering**

**Aim :** setting up a multi-node Cassandra cluster, installing and configuring DataStax OpsCenter, designing a database schema to store weather station IoT temperature sensor data, and demonstrating cluster operations using OpsCenter

**Theory** :

clustering refers to the way data is distributed and organized across multiple nodes in a cluster. Cassandra is a distributed database system designed for high availability and scalability, and clustering is fundamental to achieving these goals.

**Data Distribution**: In a Cassandra cluster, data is partitioned and distributed across multiple nodes using a technique called partitioning. Each node in the cluster is responsible for storing a portion of the data

**Replication**: To ensure fault tolerance and data availability, Cassandra replicates data across multiple nodes in the cluster. Replication involves making copies of data and storing them on different nodes. Cassandra allows you to configure the replication factor, which determines how many copies of each piece of data are stored and on which nodes they are placed.

**Scalability**: Clustering enables Cassandra to scale horizontally by adding more nodes to the cluster. As the data grows or the workload increases, you can easily expand the cluster by adding additional nodes without downtime

**Multi node cluster on single machine**

**Steps :**

1. Install the Cassandra instance on the machine
2. Create a folder for storing the 3 instances of the Cassandra
3. Create 3 folders ( 3 nodes ) and copy the Cassandra instances in each of the folder
4. For each of the node in the cluster repeat the following process

* Go to config folder of the that node and open Cassandra.yaml file
* Edit the listen address , rpc address as to 127.0.0.1 , 127.0.0.2 and 127.0.0.3 on each of the node
* Edit the seeds with “127.0.0.1 , 127.0.0.2, 127.0.0.3 ”
* Configure the ports on each of the node

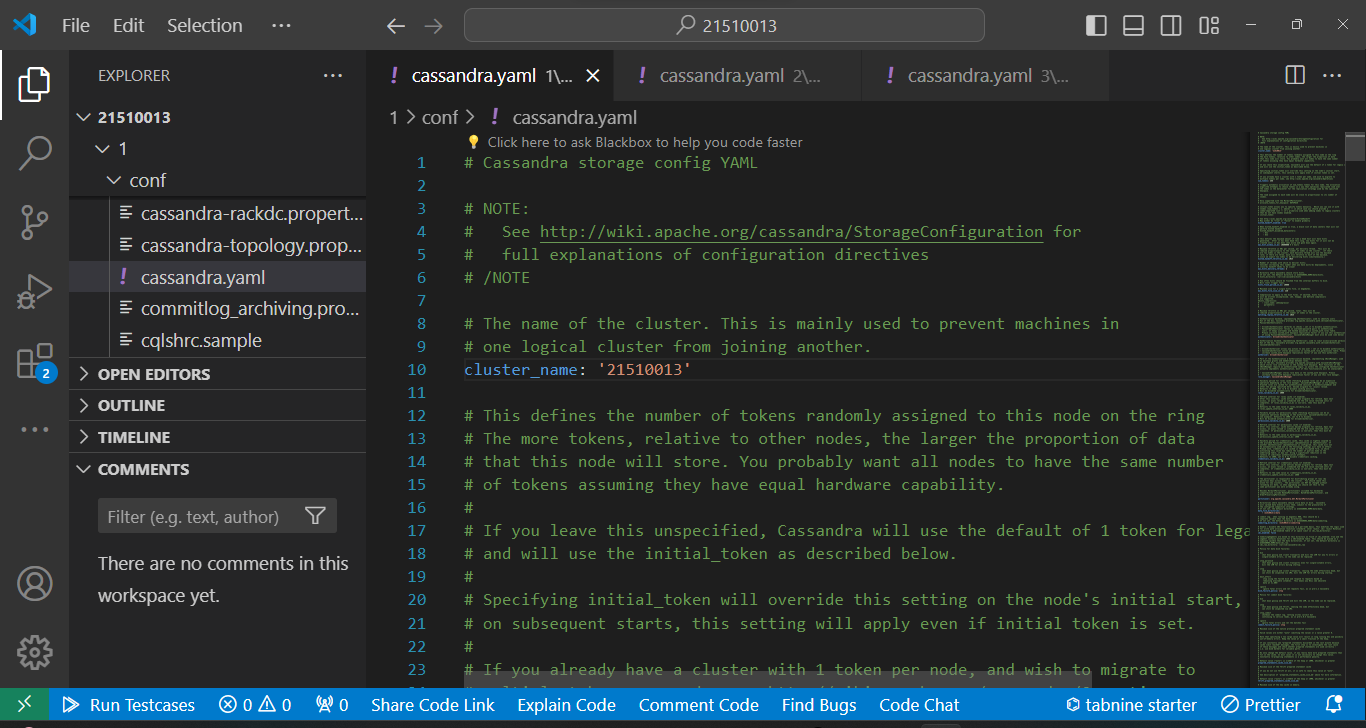
1. Then test the cluster using cassandra.bat -f in 1st node
2. Then start the instances using Cassandra -f in each node
3. To see the status of the cluster run command : nodetool status
4. Now cluster is ready to use

**Setting up cluster on multiple machine**

* Here the most of the process remains the same just we need to put the actual ip addresses of the corresponding nodes in cluster in place of rpc address and listen address for each of node in cluster
* And edit seed with list of all the ip addresses of the nodes in cluster which will be basically a comma separated list

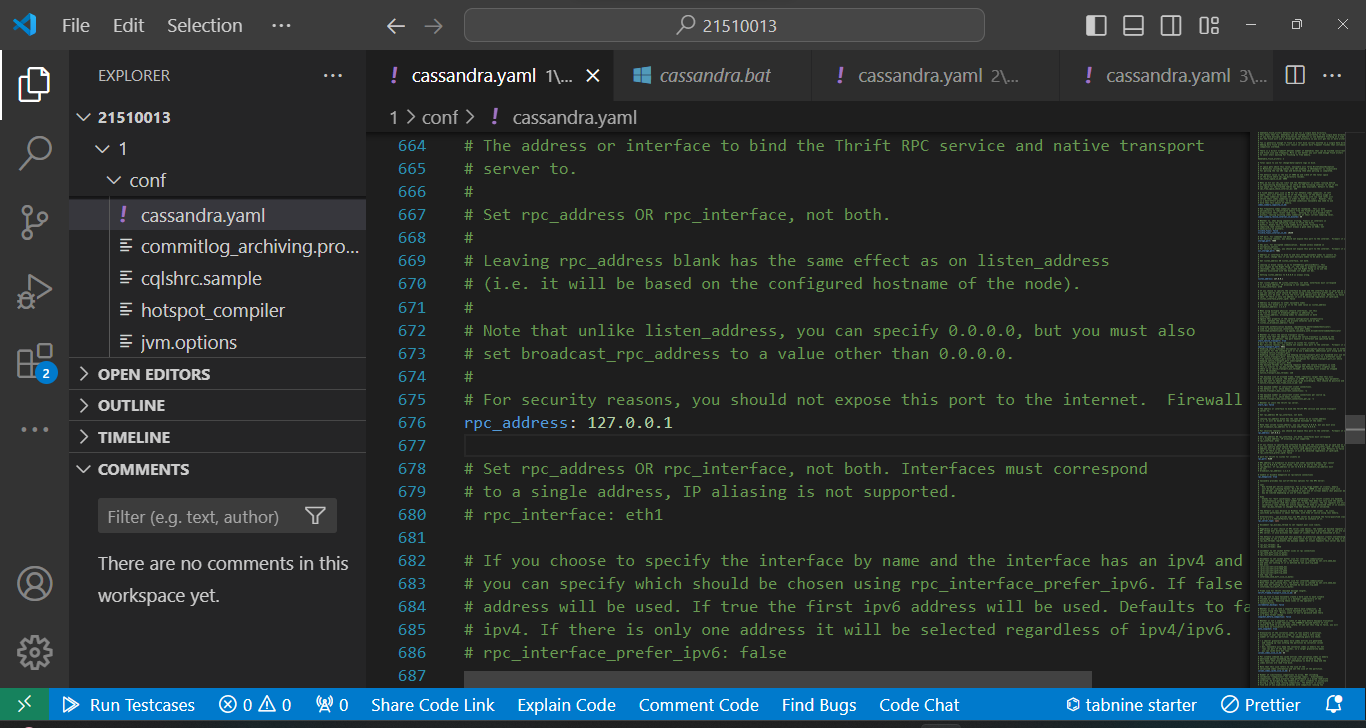
**Before setting up the Cassandra cluster we must have python 2.7.17 version , jdk8 and apache Cassandra 3.11.4 version**

**Setting up cluster name**

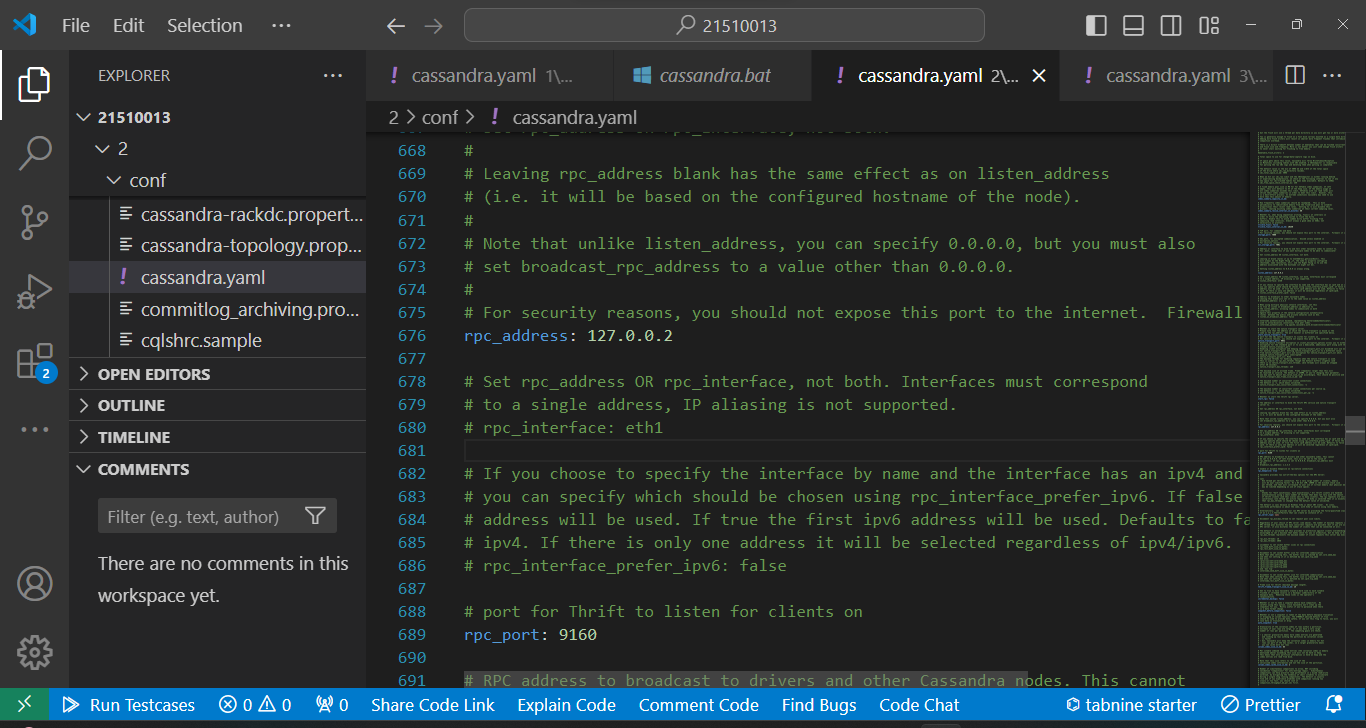
****

**Specifying the rpc\_address**

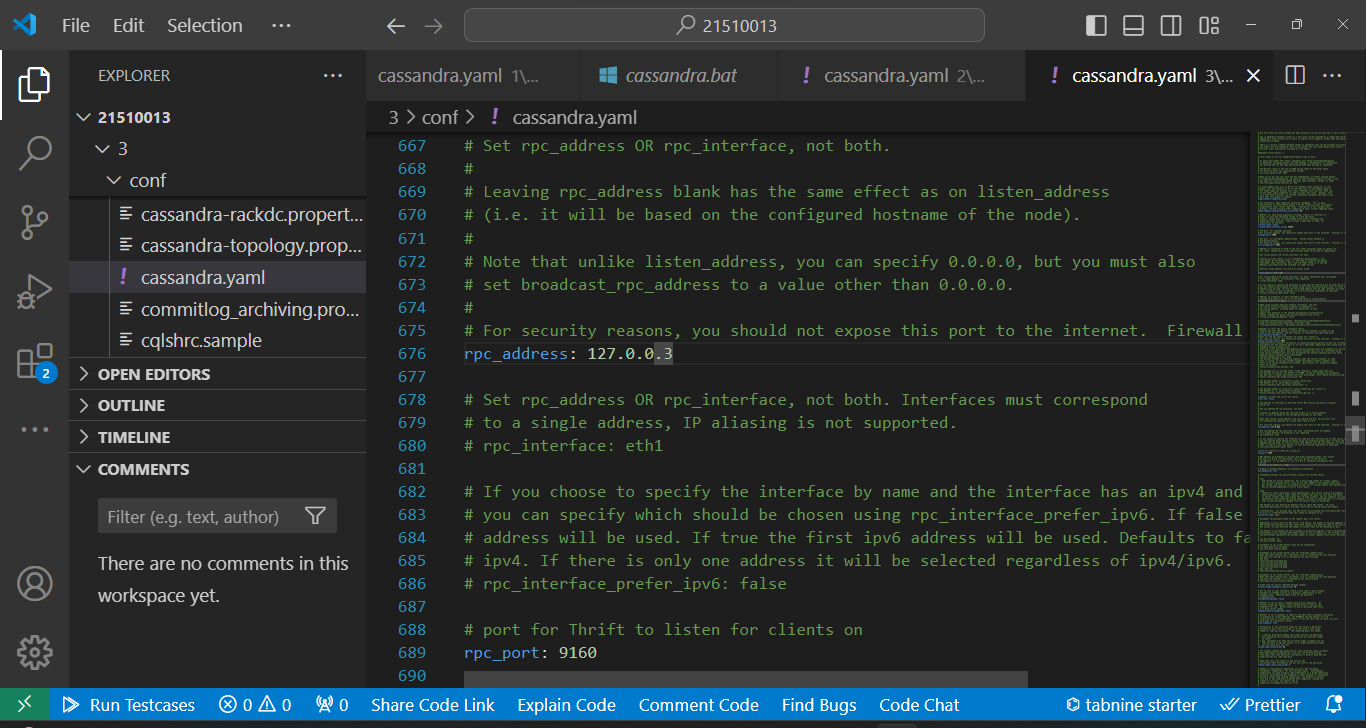
**Node 1**

****

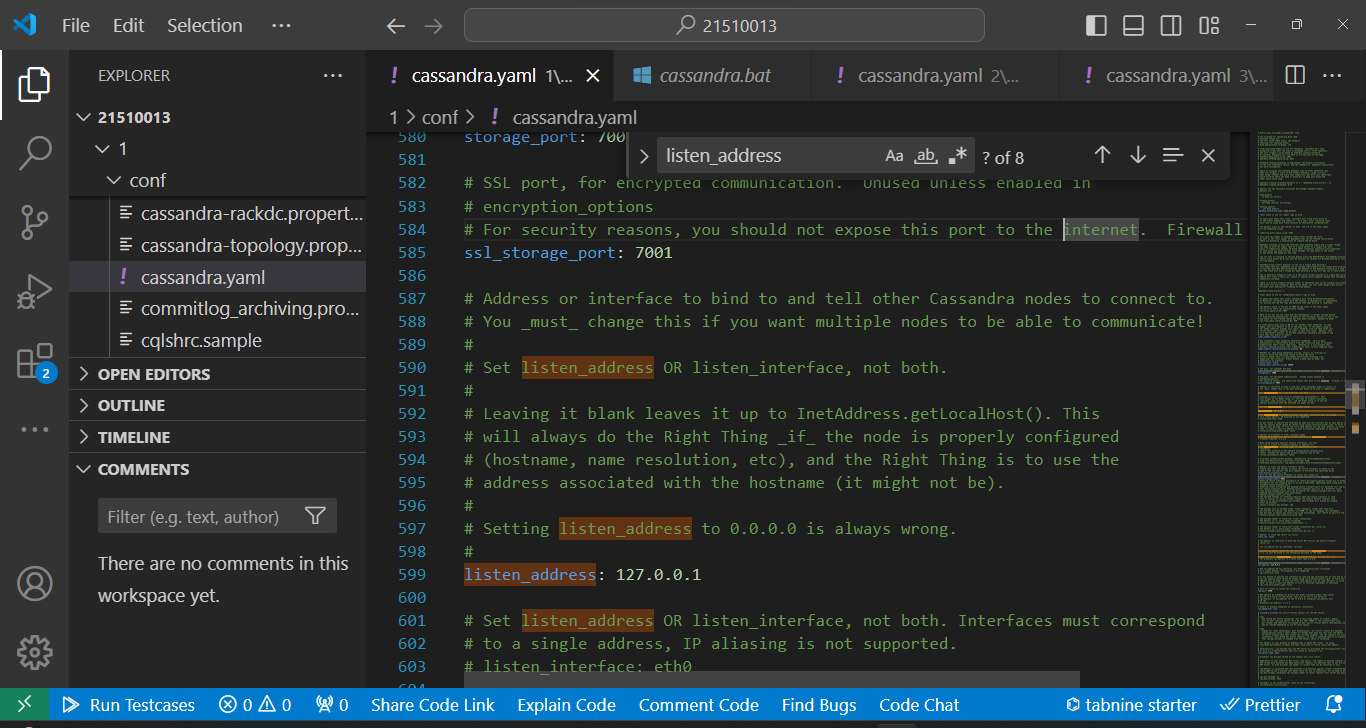
**Node 2**

****

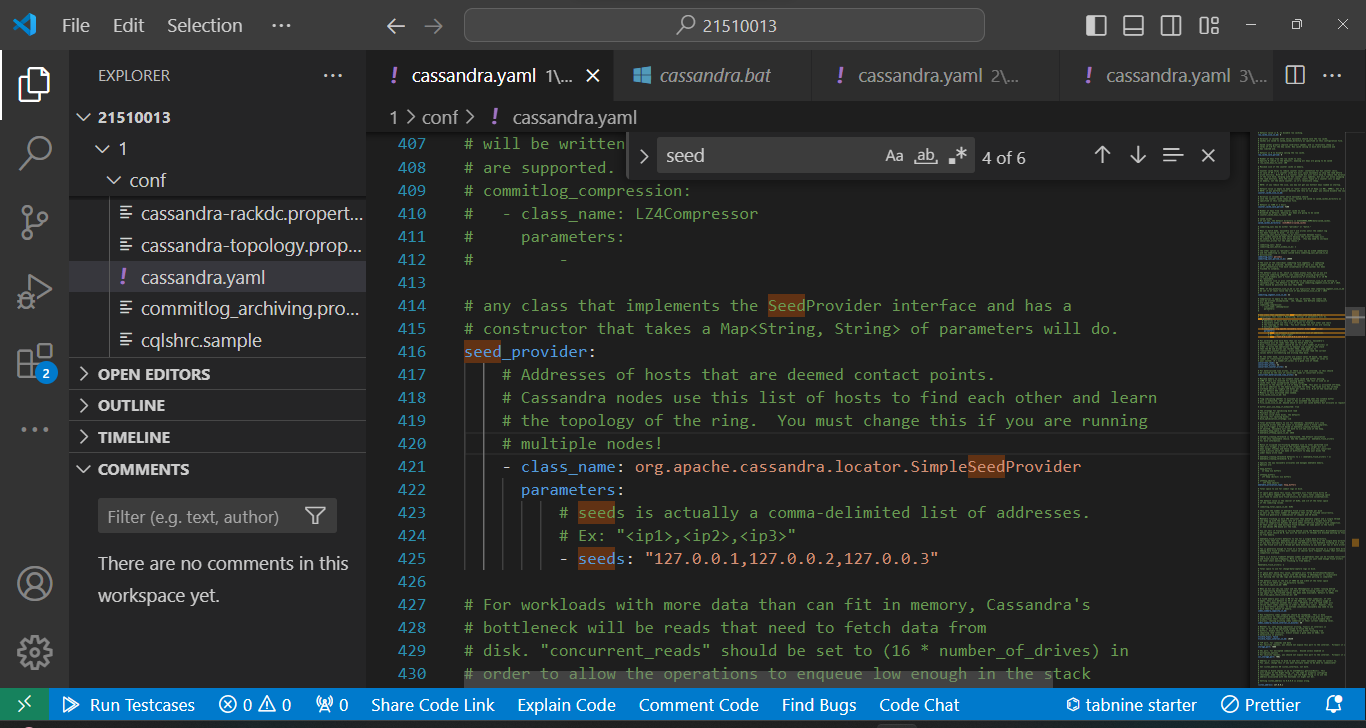
**Node 3**

****

**Setting up listen address**

****

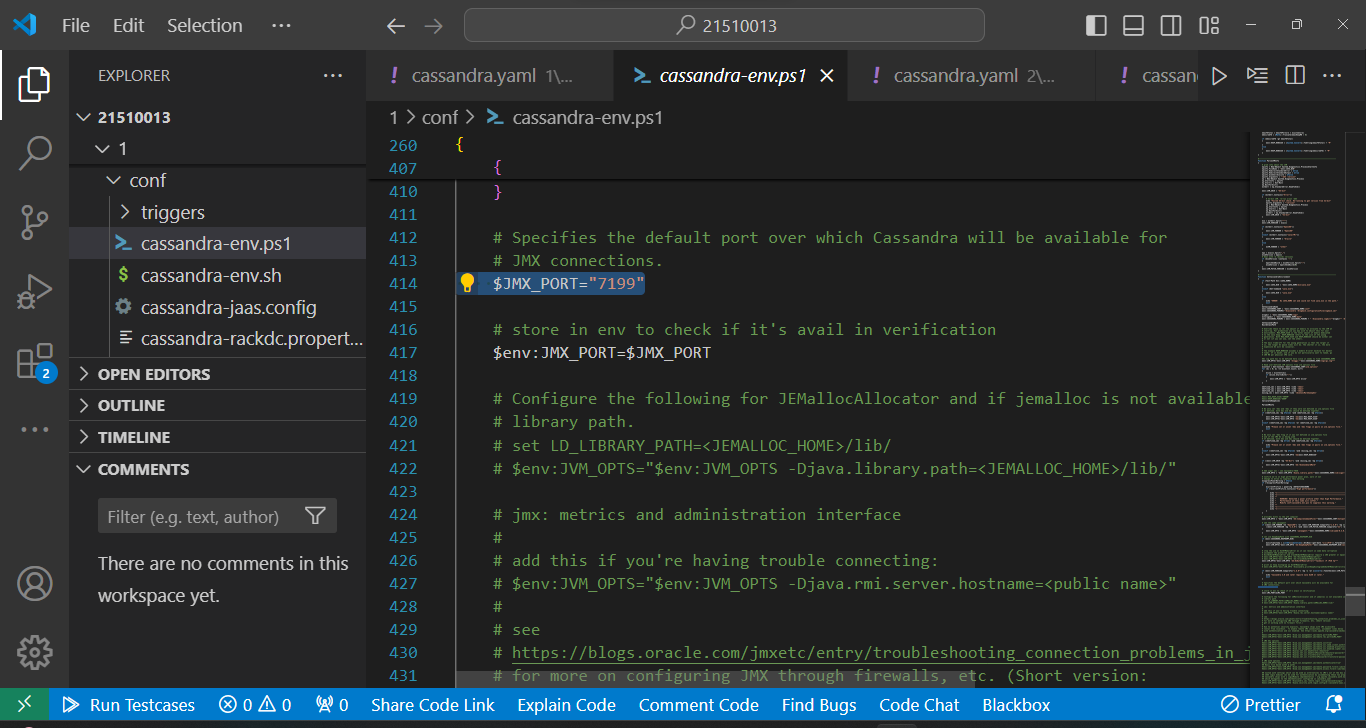
**Setting up the seeds**

****

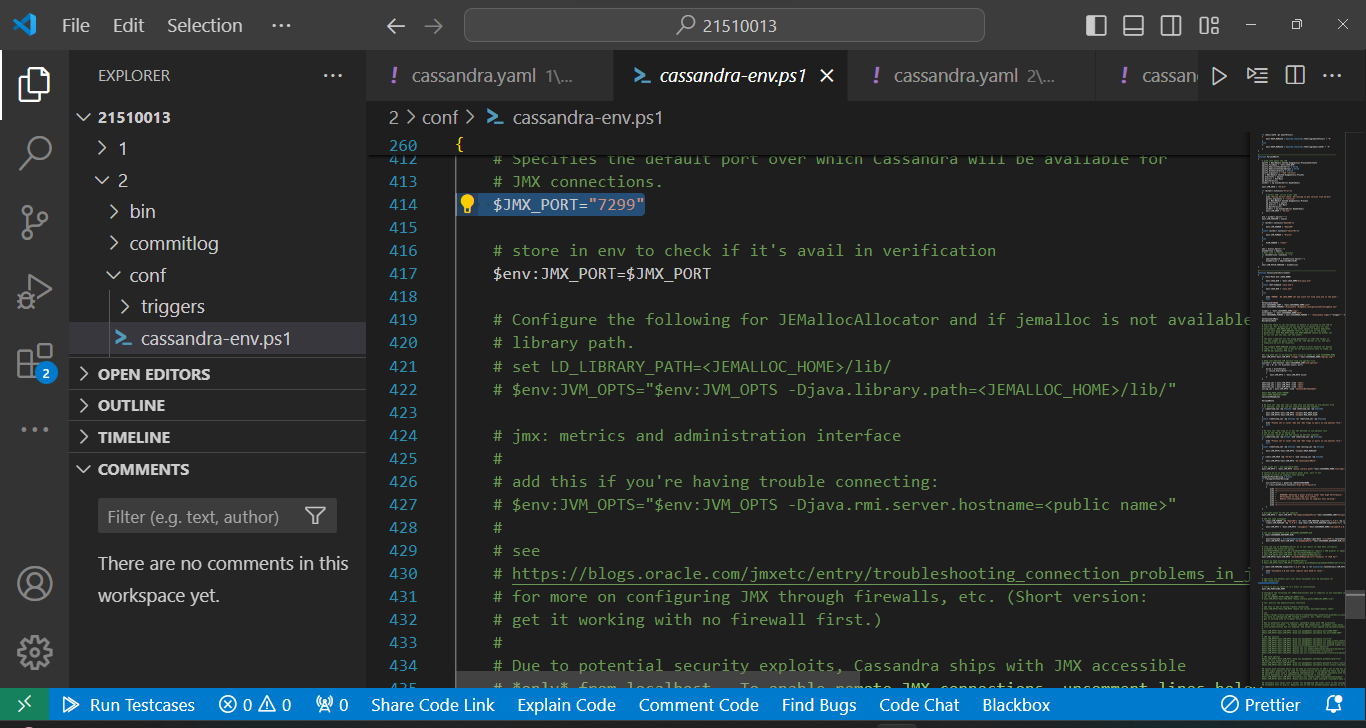
* rpc\_address: Specifies the IP address or hostname where clients can connect to the node over the native protocol (Thrift RPC).
* listen\_address: Defines the IP address or hostname where the node listens for incoming connections from other nodes in the cluster.
* seeds: Specifies a list of nodes in the cluster that serve as initial contact points for the discovery process. These nodes are used by new nodes joining the cluster to find and establish initial connections for cluster communication.

**Setting up the ports**

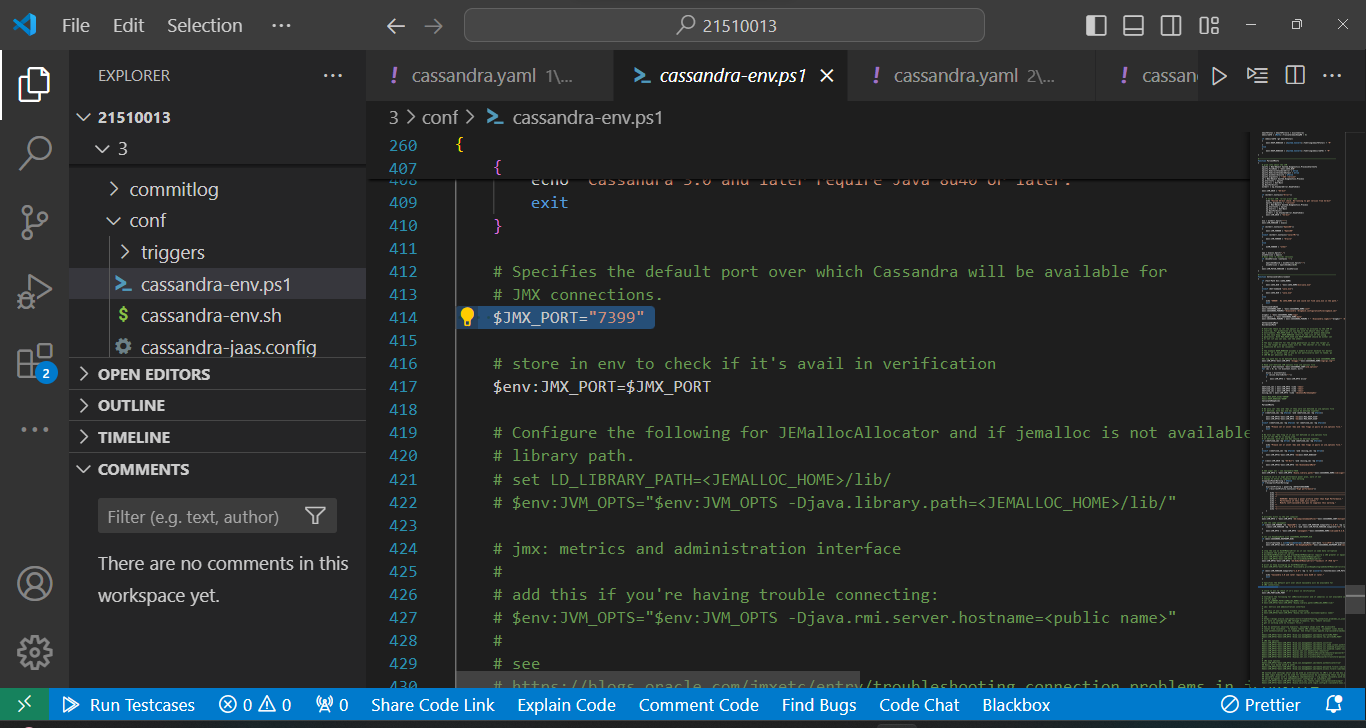
**For node 1 : 7199**

****

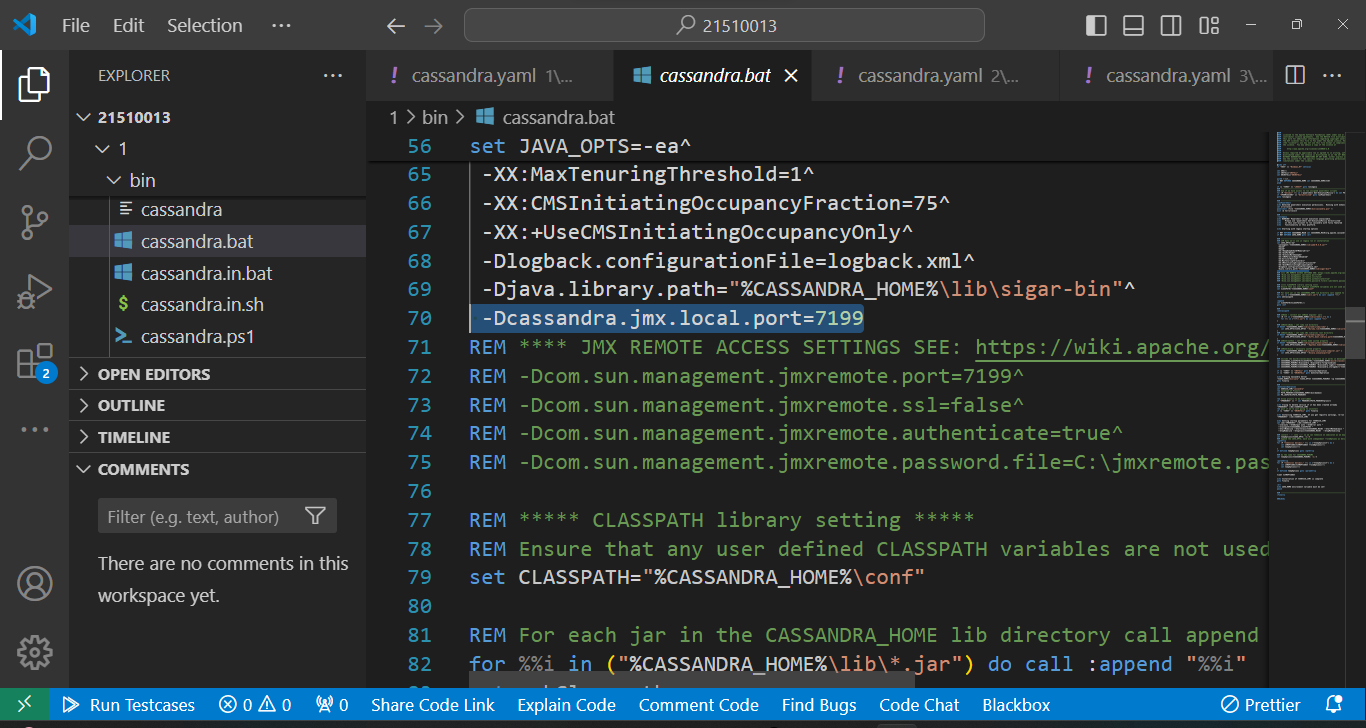
**For node 2 : 7299**

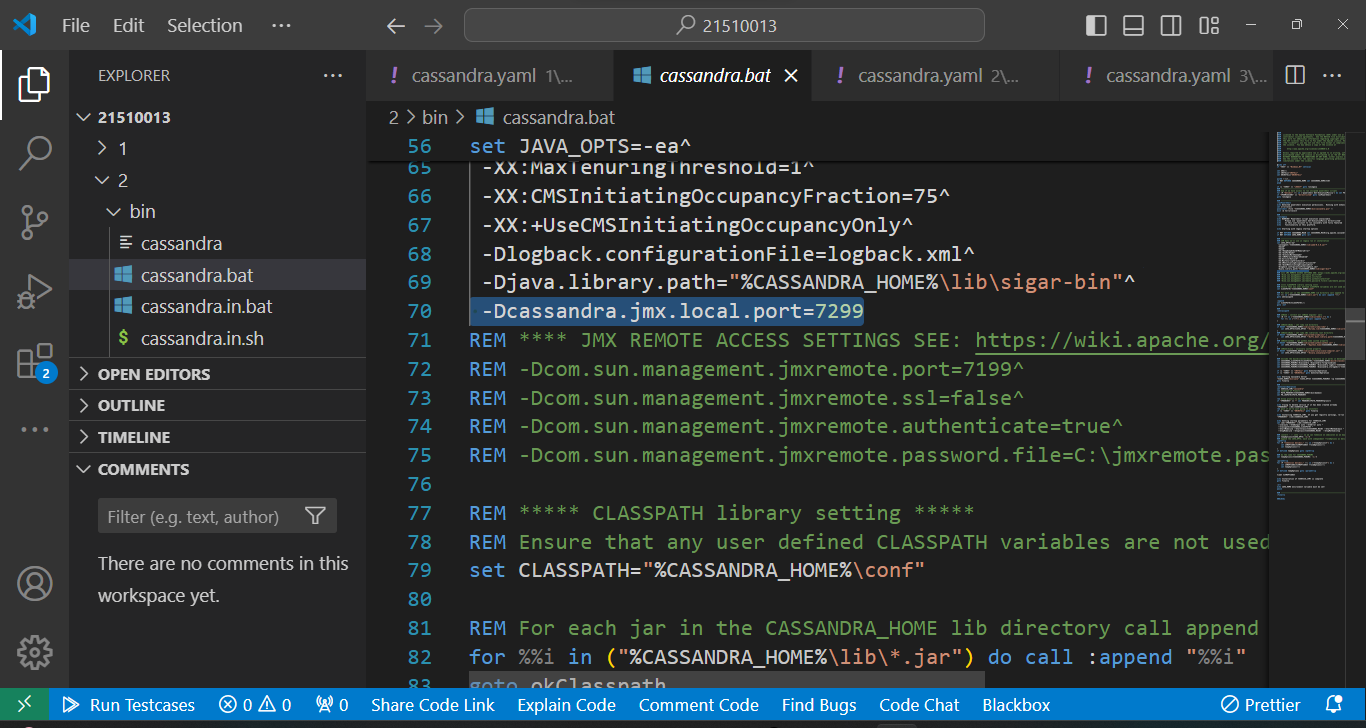
****

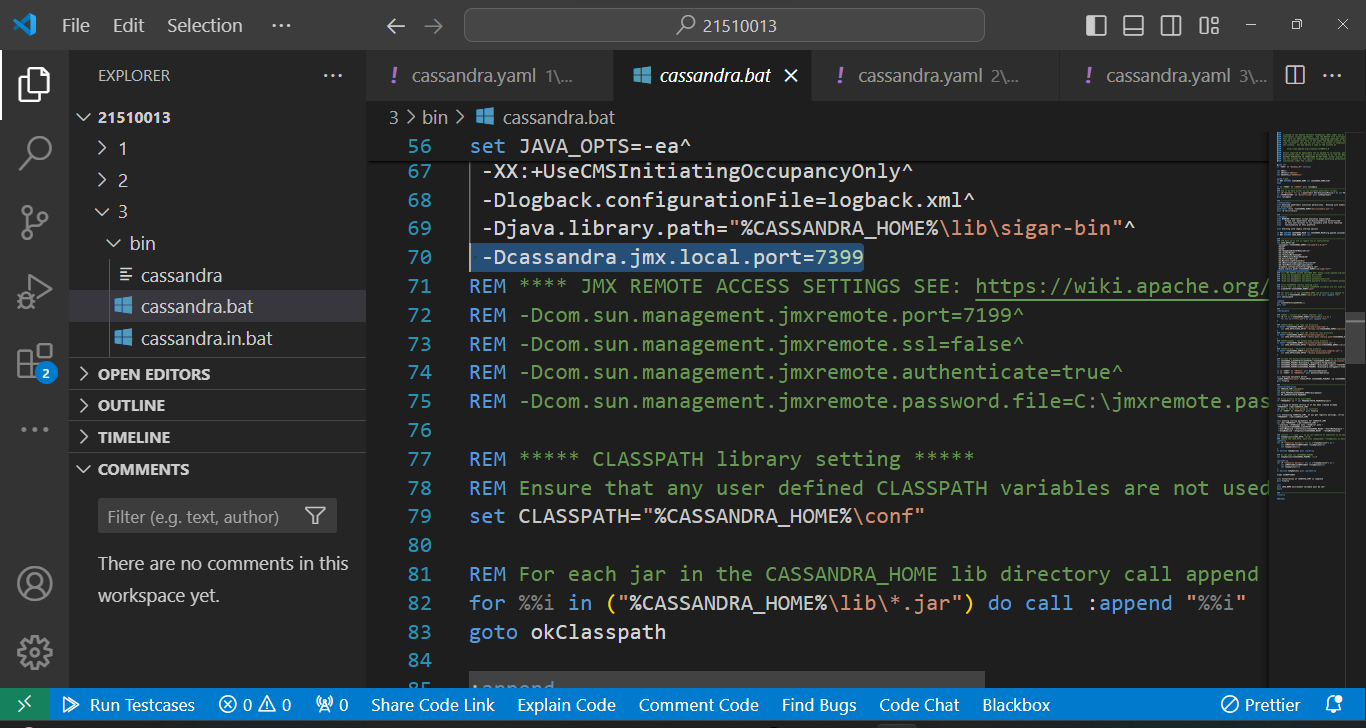
**For node 3 : 7399**

****

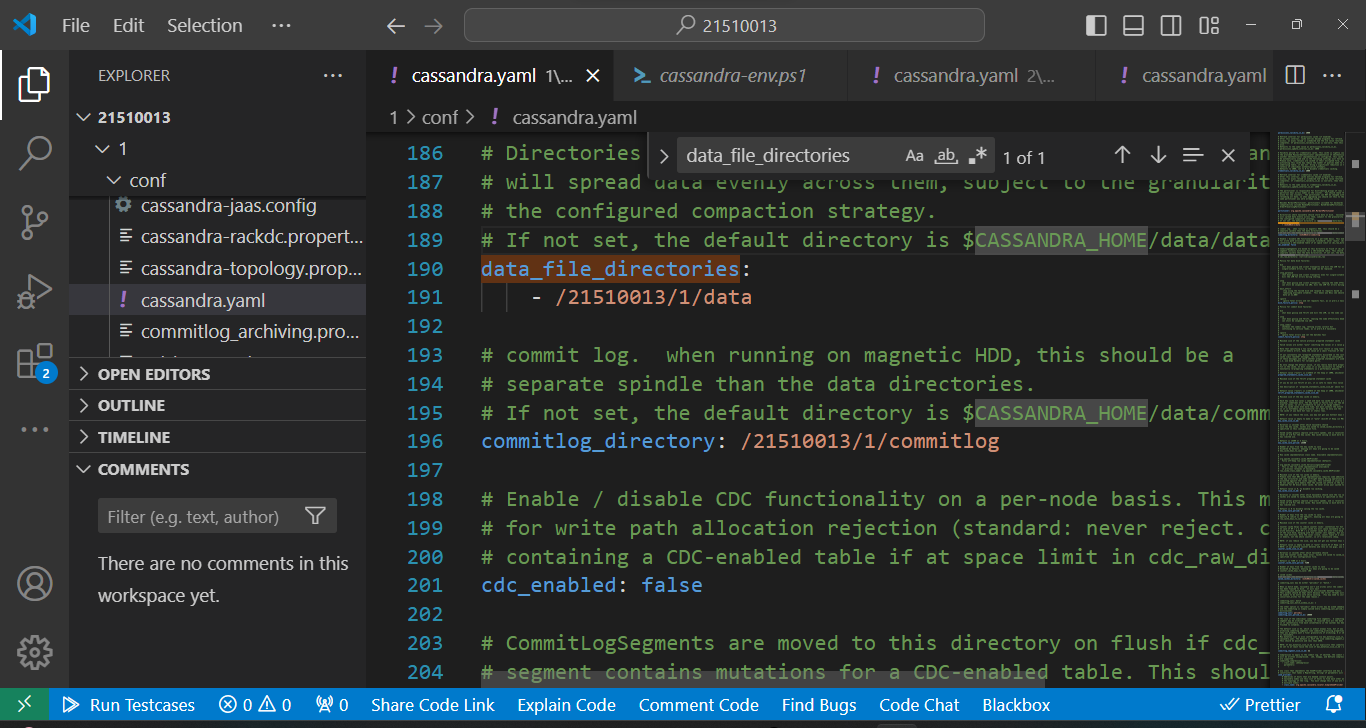
**And similarly we need to edit ports in cassandra.bat file for each node**

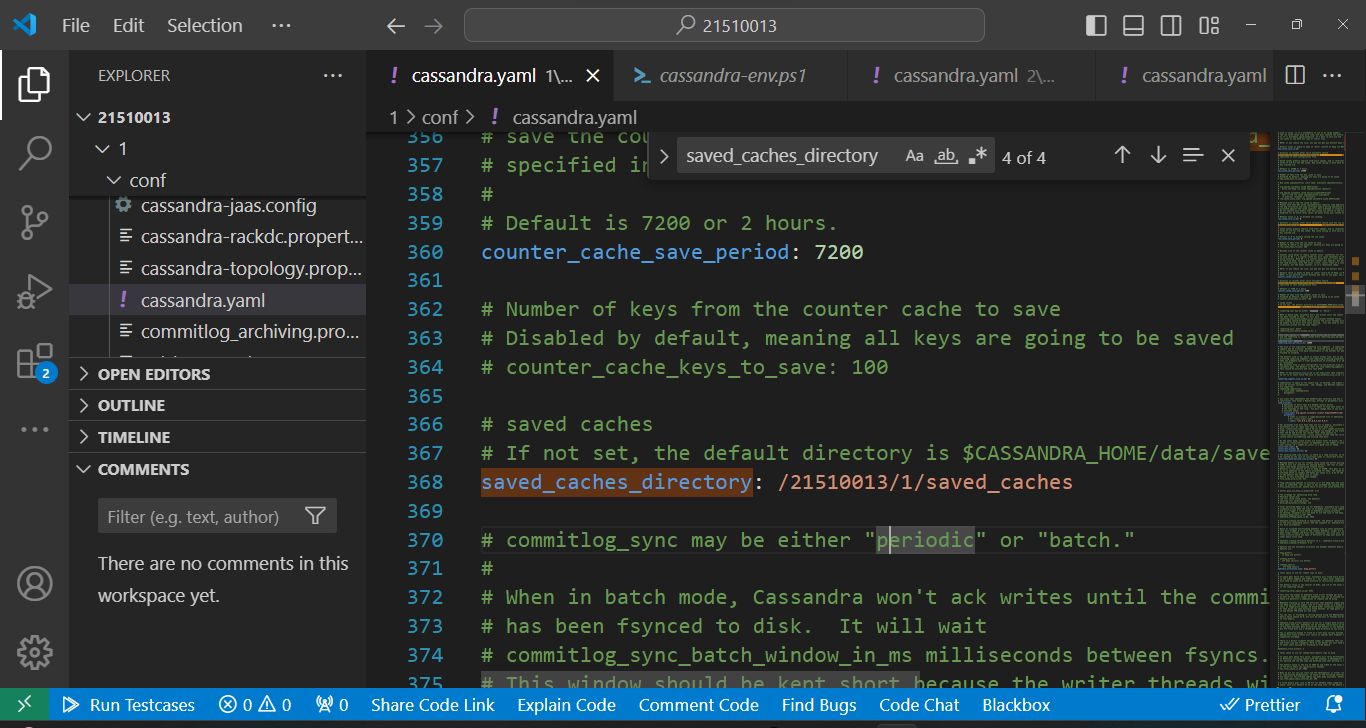
****

****

****

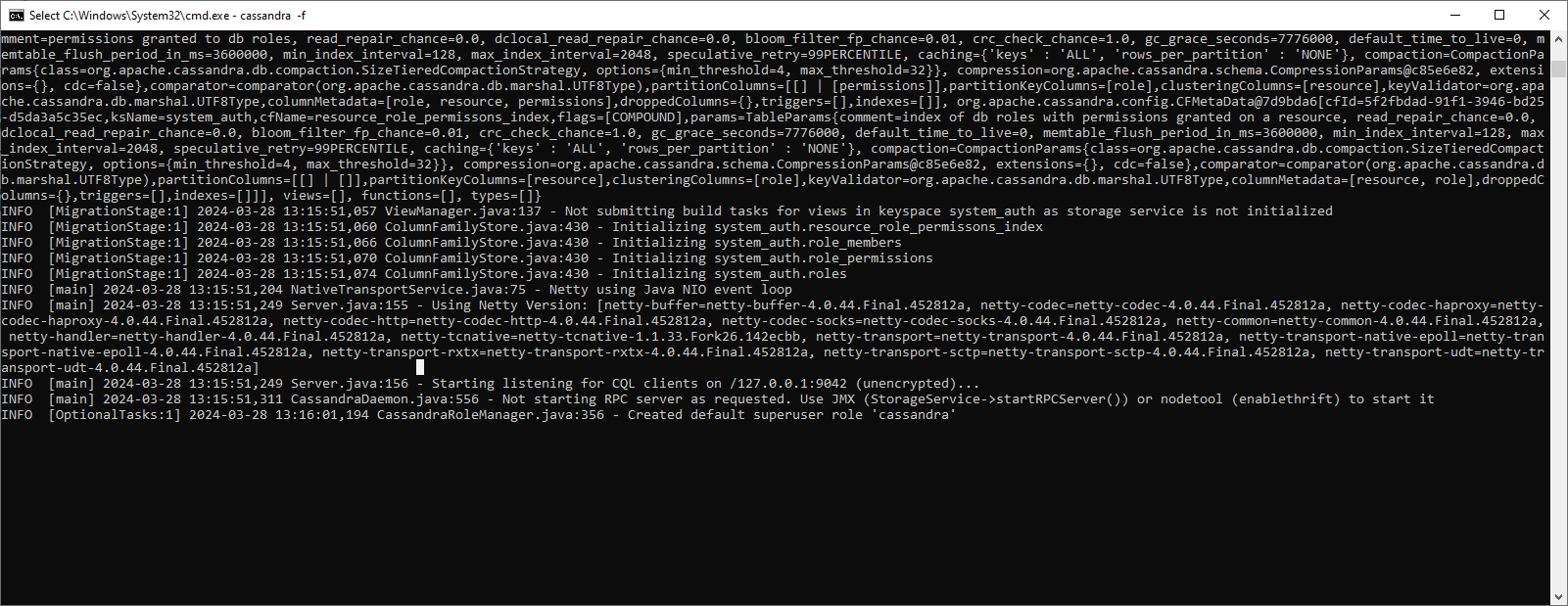
**Now we need to update the commit log directory , data file directory , saved cache directory of each of the node according to the path of individual node**

**From Cassandra.yaml file**

****

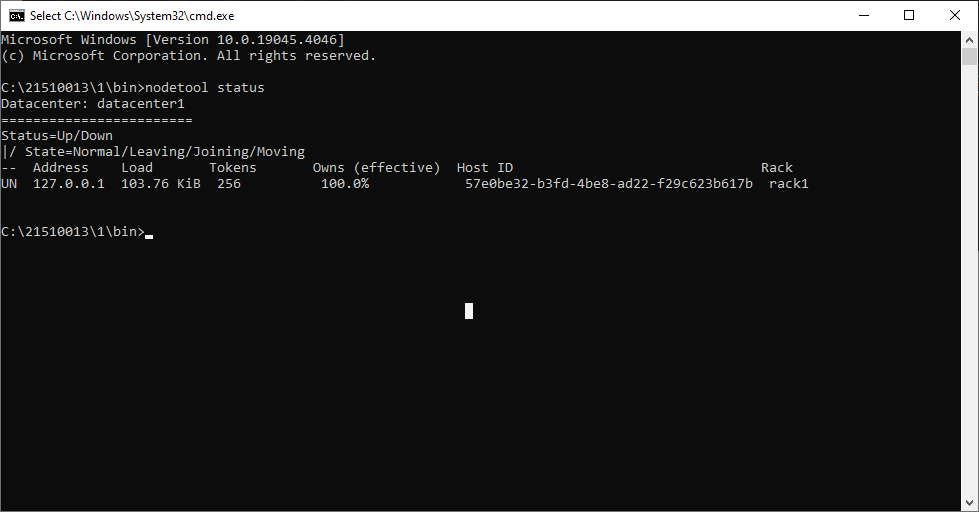
**Starting the node 1**

**Command : Cassandra -f**

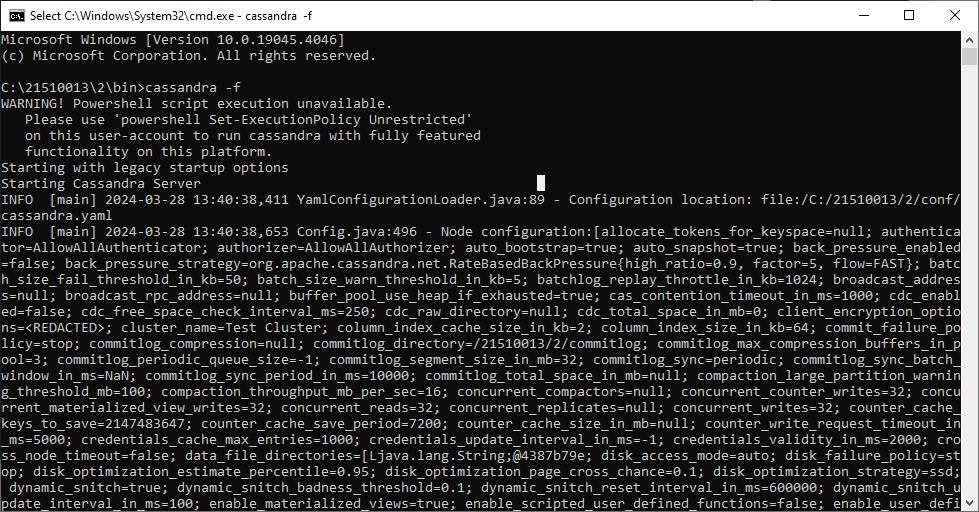


**Checking status after node 1 started :**

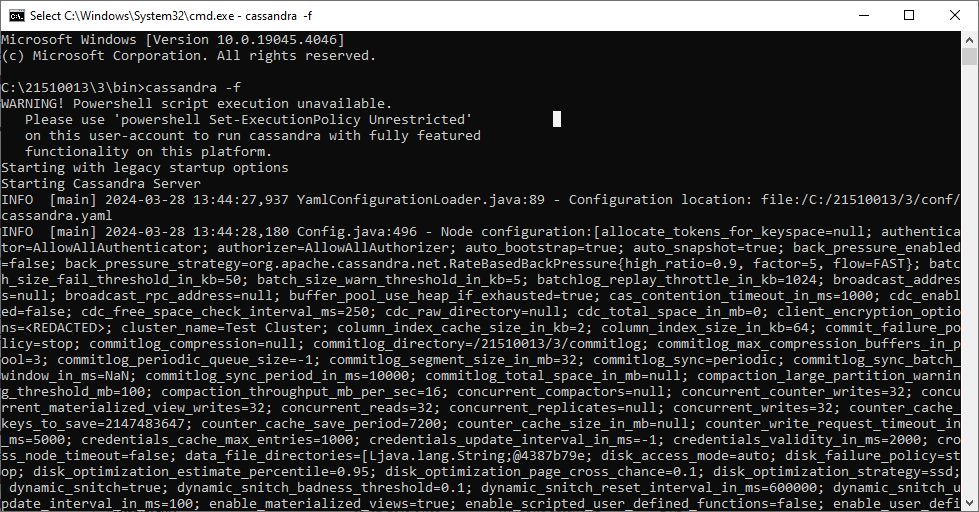
**Command : nodetool status**



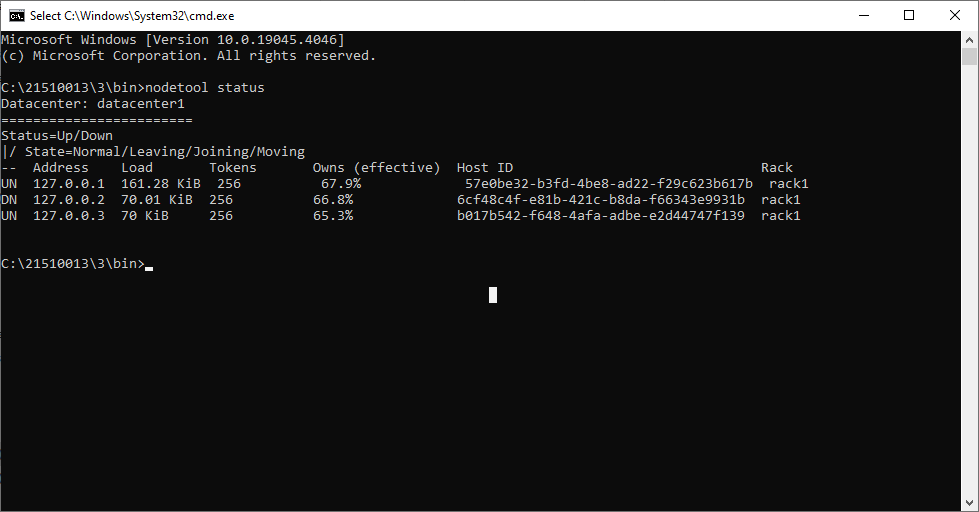
**Node 2 started**



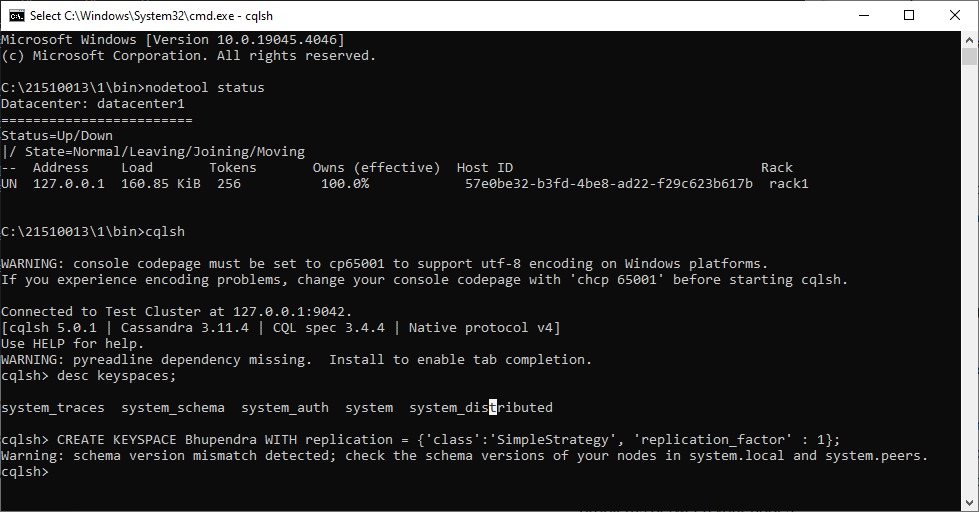
**Node 3 started**



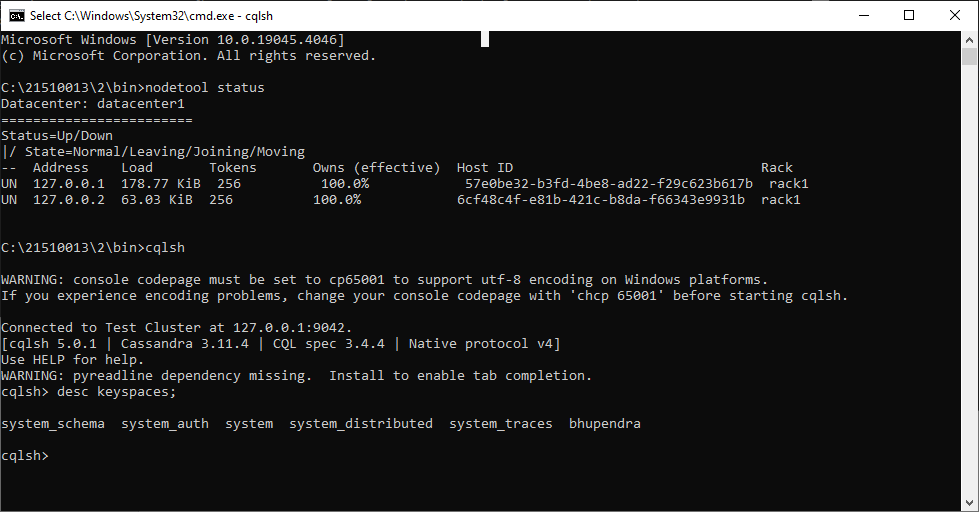
**Status after starting all nodes**



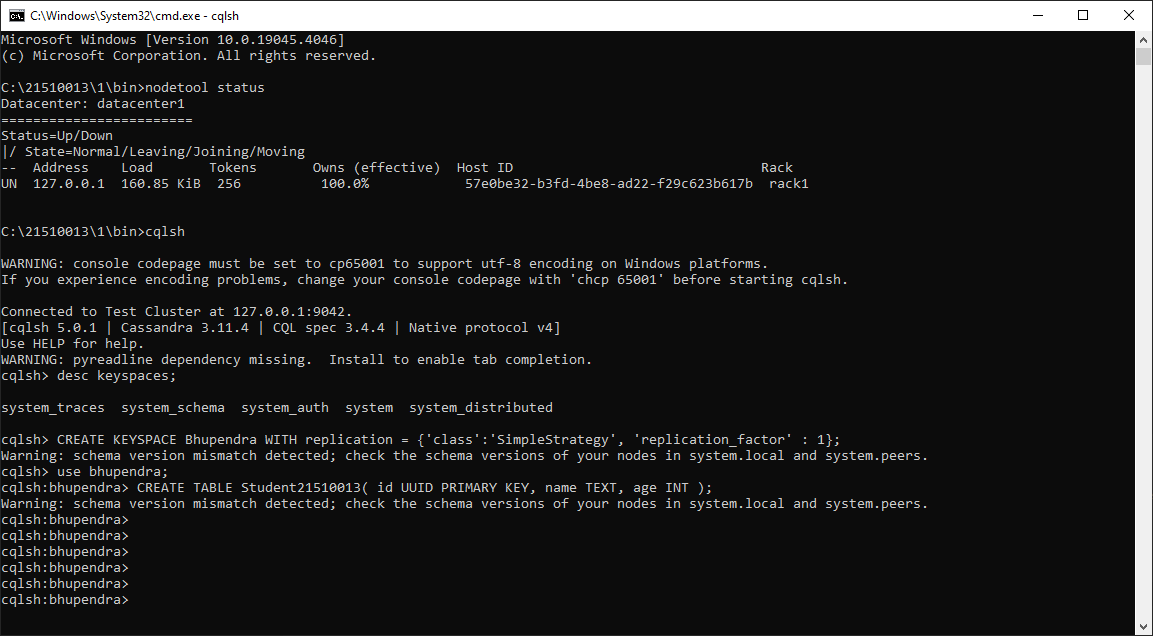
**Creating keyspace on node 1**



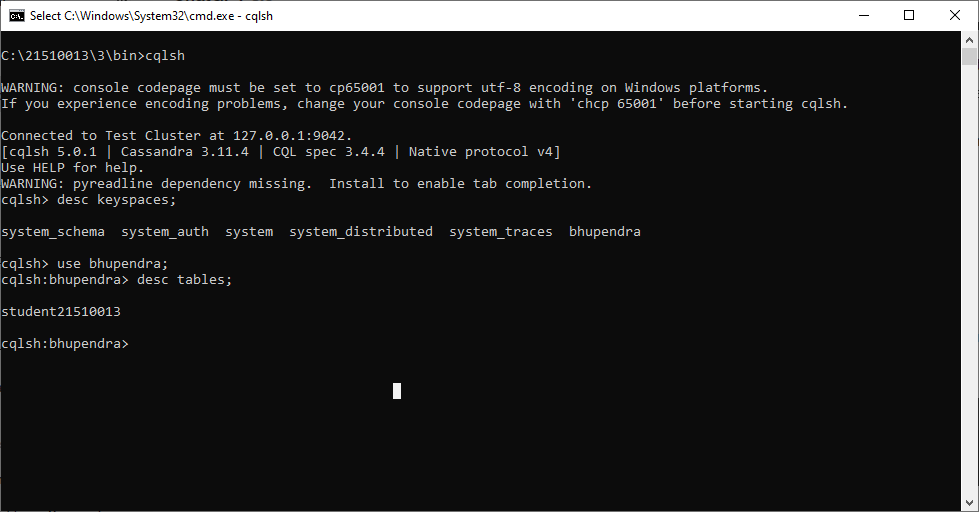
**Keyspace visible at other nodes**



**Table created on node 1**



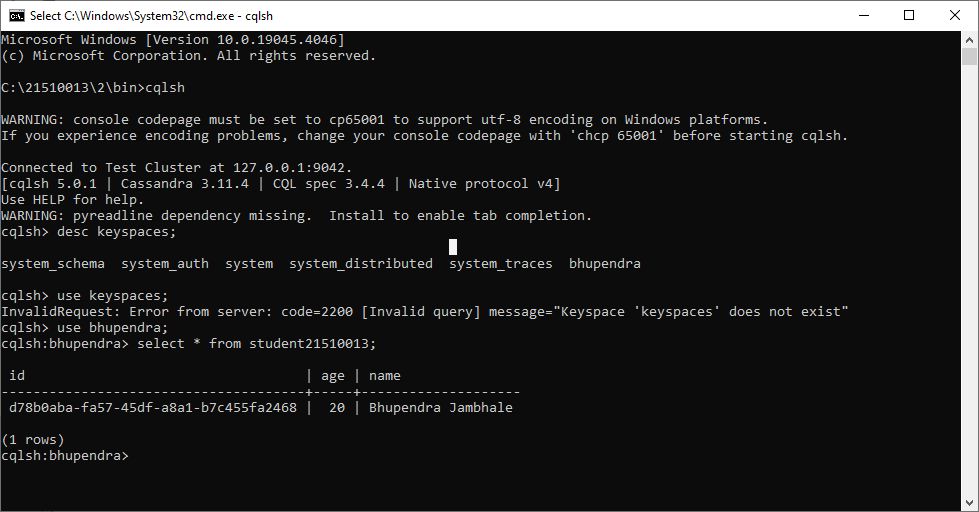
**Tables visible at other nodes**



**Inserting data**



**Data visible at other nodes**



**Configuration details of each node**

1. **Node 1 :**

port : 7199

rpc\_address : 127.0.0.1

listen\_address : 127.0.0.1

seeds : 127.0.0.1, 127.0.0.2, 127.0.0.3

1. **Node 2 :**

port : 7299

rpc\_address : 127.0.0.2

listen\_address : 127.0.0.2

seeds : 127.0.0.1, 127.0.0.2, 127.0.0.3

1. **Node 3 :**

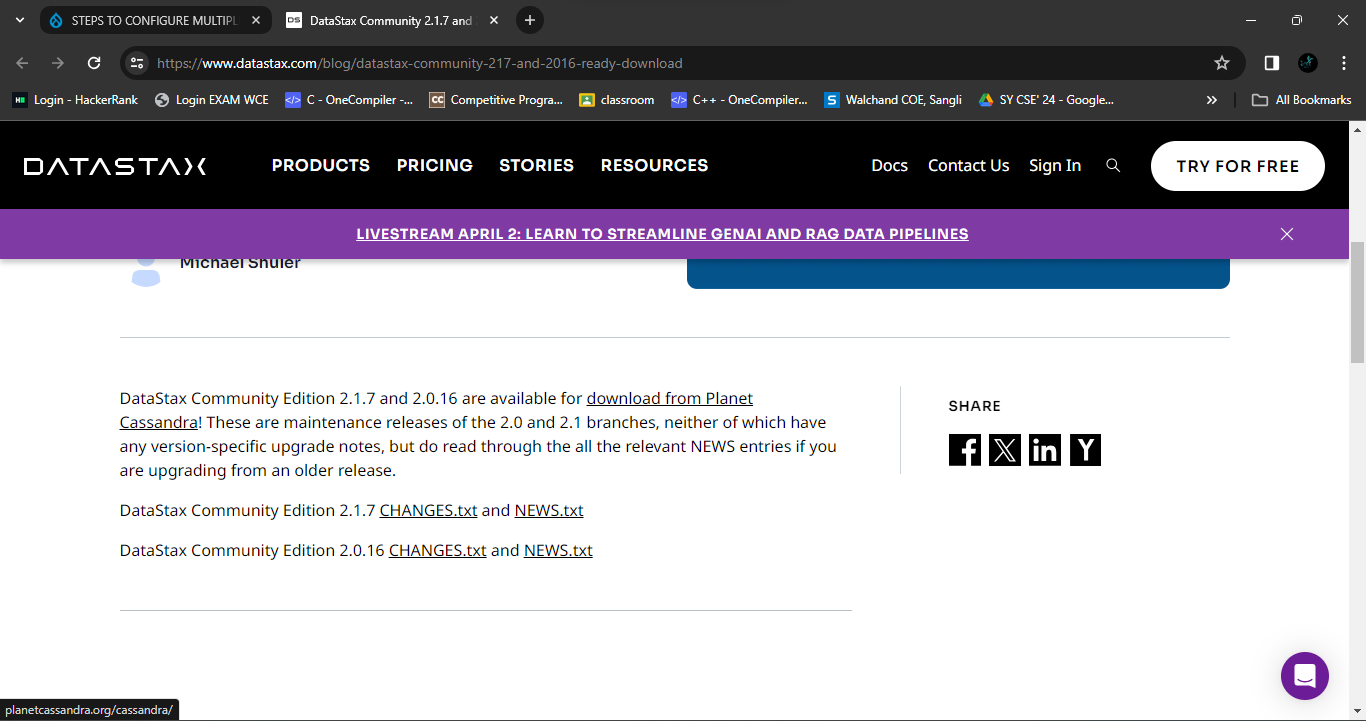
port : 7399

rpc\_address : 127.0.0.3

listen\_address : 127.0.0.3

seeds : 127.0.0.1, 127.0.0.2, 127.0.0.3

**Datastax installation**

****

**Schema**

CREATE KEYSPACE weather\_data WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 3};

USE weather\_data;

CREATE TABLE temperature\_data (

weatherStationID text,

timestamp timestamp,

temperature float,

PRIMARY KEY ((weatherStationID), timestamp)

) WITH CLUSTERING ORDER BY (timestamp DESC);

**References :**

**[1]**  <https://extendit.us/articles/steps-configure-multiple-nodes-cassandra-single-windows-machine>

**[2]** (https://www.datastax.com/blog/datastax-community-217-and-2016- ready-download)

**[3]** <https://cassandra.apache.org/doc/4.0/cassandra/cql/ddl.html>

**[4]** [**https://www.datastax.com/blog/ccm-development-tool-creating-local-cassandra-clusters**](https://www.datastax.com/blog/ccm-development-tool-creating-local-cassandra-clusters)

**Conclusion :**

In conclusion, this assignment encompasses setting up a multi-node Cassandra cluster, installing and configuring DataStax OpsCenter, and understanding key concepts like Cassandra clustering, network configuration parameters (rpc\_address, listen\_address, seeds), and cluster management. Through practical tasks and hands-on experience, one gains insights into distributed database systems, data modeling, monitoring, and operational aspects.