Guide: Prof. Sasikala R
Name: Asif Sayyad
Reg No: 18MCS0020

NoSQL – Assignment 3

Create table Customers (Id int PRIMARY KEY, Name varchar, ReferredBy int, Mobile int);

Create table Invoices (Id int, BillingDate date, CustomerId int, PRIMARY KEY (Id, BillingDate));

Create index IndexMob on simplekeyspace1.Customers(Mobile);

2. The list should be ordered by billing date.

select * from Invoices where id=1008 order by BillingDate desc ;

Output:

Results | Query Trace

id	billingdate	customerid
1008	2015-01-11	7
1008	2013-01-12	5
1008	2013-01-11	5
1008	2007-01-11	5

3. Create Customers list based on id (unique key)

create table Customers (Id uuid PRIMARY KEY, Name varchar, ReferredBy int, Mobile int);

Output:

```
cqlsh:demo> insert into Customers (Id, Name) values(now(), 'Asif');
cqlsh:demo> select * from Customers;

 id                                   | mobile | name | referredby
--------------------------------------+--------+------+------------
 e38f3830-2703-11e9-bc54-25607de025b7 |   null | Asif |       null

(1 rows)
cqlsh:demo> insert into Customers (Id, Name) values(now(), 'Abhishek');
cqlsh:demo> insert into Customers (Id, Name) values(now(), 'Asif');
cqlsh:demo> select * from Customers;

 id                                   | mobile | name     | referredby
--------------------------------------+--------+----------+------------
 e38f3830-2703-11e9-bc54-25607de025b7 |   null |     Asif |       null
 f5402e40-2703-11e9-bc54-25607de025b7 |   null | Abhishek |       null
 f986b190-2703-11e9-bc54-25607de025b7 |   null |     Asif |       null

(3 rows)
```

4. Retrieve the customer details based on Mobile Number(index)

Create table Customers (Id int PRIMARY KEY, Name varchar, ReferredBy int, Mobile int);

Create index IndexMob on simplekeyspace1.Customers(Mobile);

SELECT * FROM customers WHERE mobile=40368390;

Output:

Results | Query Trace

id	mobile	name	referredby
5	40368390	Ram	3
9	40368390	Balaji	7

5. Create a simple Cassandra cluster with 3 nodes

Step 1: Installation of Cassandra on all 3 nodes.

Step 2: Each node has open communication between the other nodes. Set up the firewall rules between nodes.

System 1 - 192.168.43.152 - Asif

System 2 - 192.168.43.52 – Abhishek

System 3 - 192.168.43.115 – Vinay

sudo ufw allow proto tcp from 192.168.43.115 to any port 7000,9042 comment "Cassandra TCP"

sudo ufw allow proto tcp from 192.168.43.152 to any port 7000,9042 comment "Cassandra TCP"

sudo ufw allow proto tcp from 192.168.43.52 to any port 7000,9042 comment "Cassandra TCP"

Step 3: Edit your **cassandra.yaml** file for each node.

a) Cluster name should be same for all three nodes.

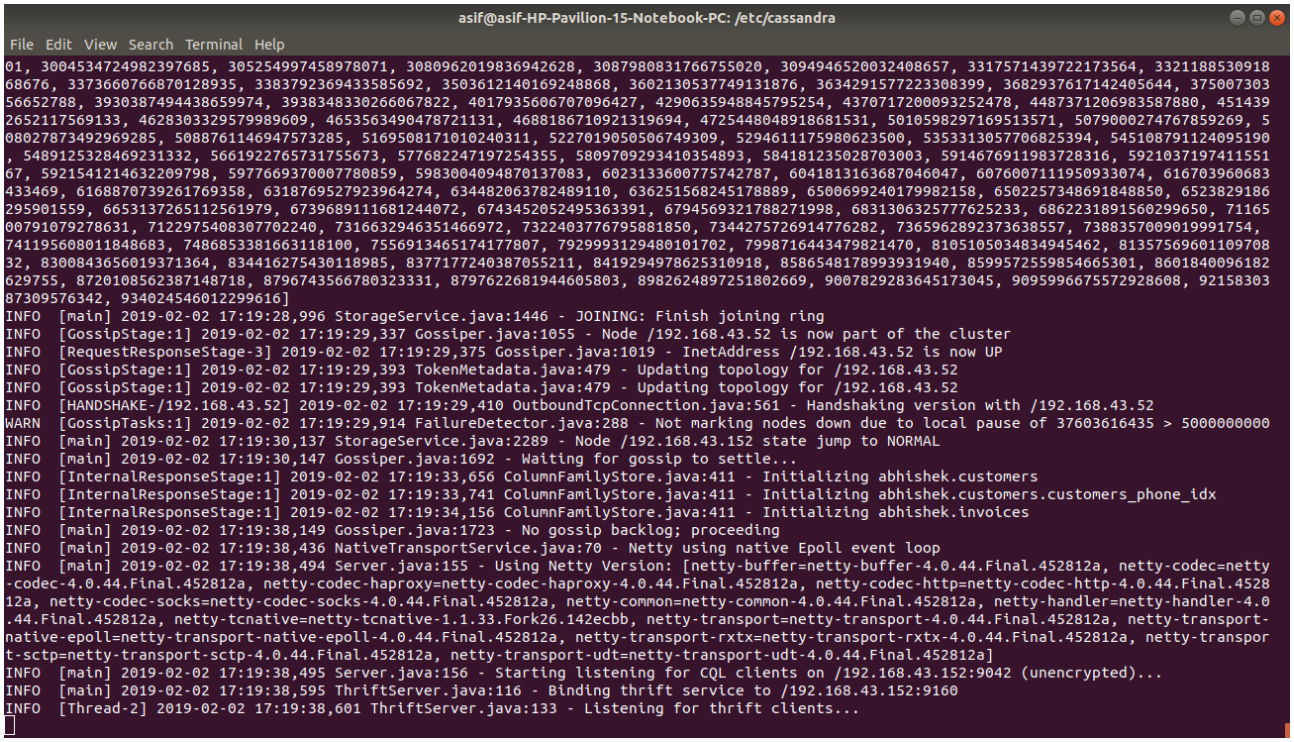
b) seeds: "192.168.43.115, 192.168.43.152, 192.168.43.52"

c) auto_default: True

d) Each node will specify listen_address and rpc_address as its own IP address.

Step 4: Run node 1 first,once this is up and running we can start the remaining nodes.

-- sudo cassandra -f -R (Node 1)

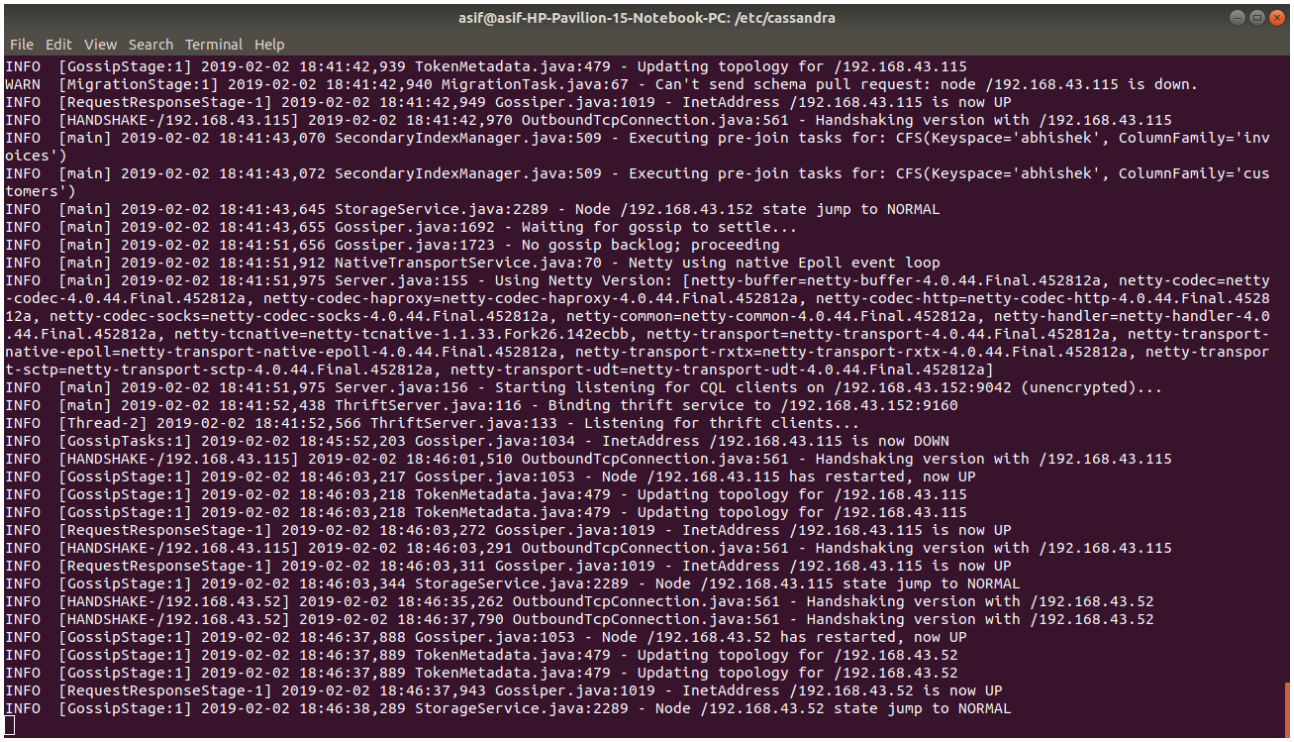


Step 5: Then, we will start other two nodes

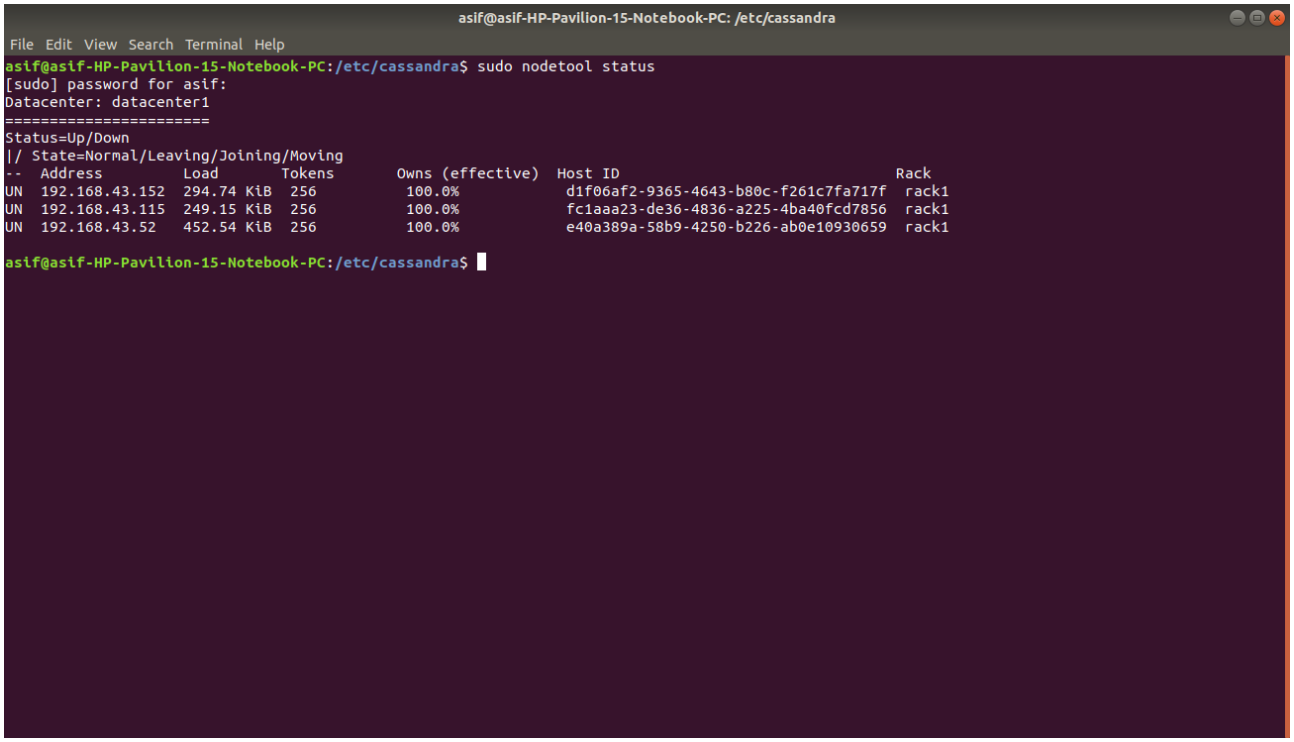
-- sudo cassandra -f -R (Node 2)

-- sudo cassandra -f -R (Node 3)

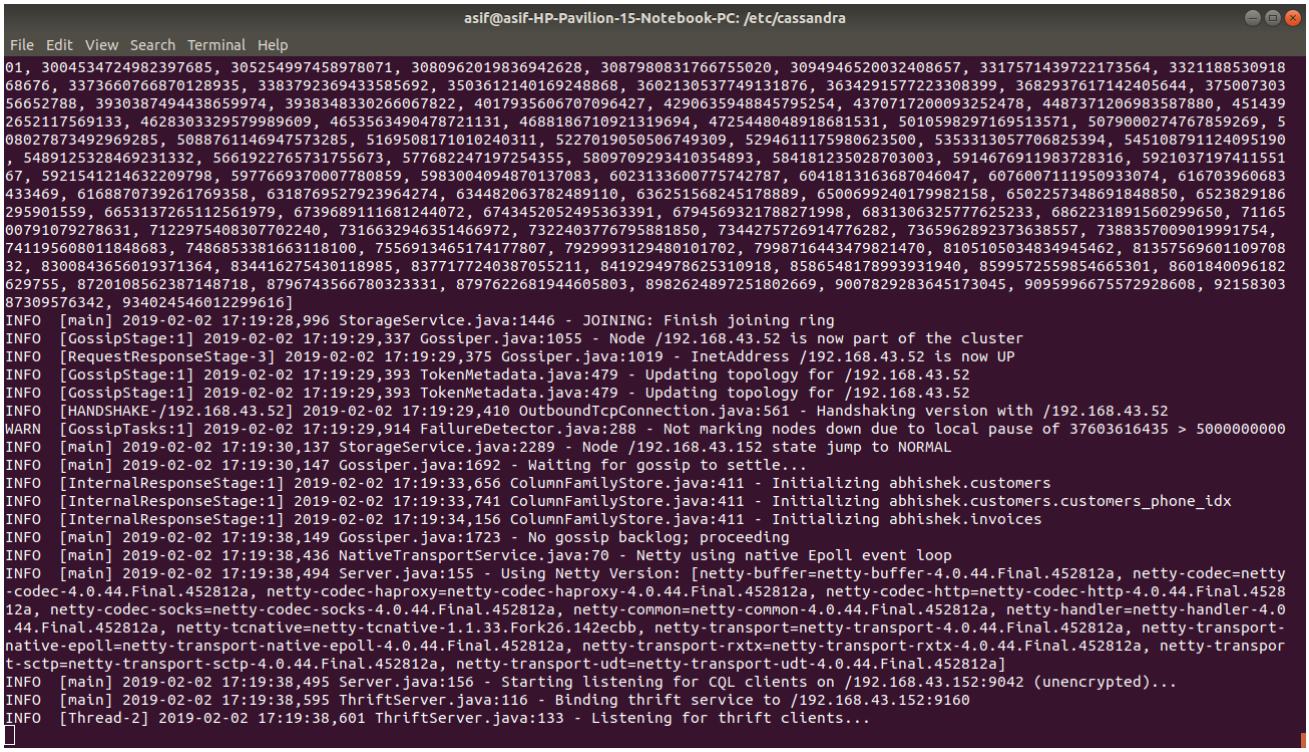
Step 6: Other two nodes connected to first node.



Step 7: Once all of services are started, we can use the **nodetool status** command to check the status of our nodes. We can run this from any Cassandra server. As you can see with the below output, all three servers are available in the **un** data centre on **rack1.**



Step 8: Nodes are able to query other nodes and fetch data located at other nodes.



```
asif@asif-HP-Pavilion-15-Notebook-PC:/etc/cassandra$ cqlsh
Connection error: ('Unable to connect to any servers', {'127.0.0.1': error(111, "Tried connecting to [('127.0.0.1', 9042)]. Last error: Connec
tion refused")})
asif@asif-HP-Pavilion-15-Notebook-PC:/etc/cassandra$ cqlsh 192.168.43.152
Connected to Test Cluster at 192.168.43.152:9042.
[cqlsh 5.0.1 | Cassandra 3.11.3 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> describe keyspaces

system_schema  system_auth  system  system_distributed  abhishek  system_traces
```