**TY B.Tech. (CSE) – II [ 2022-23 ]**

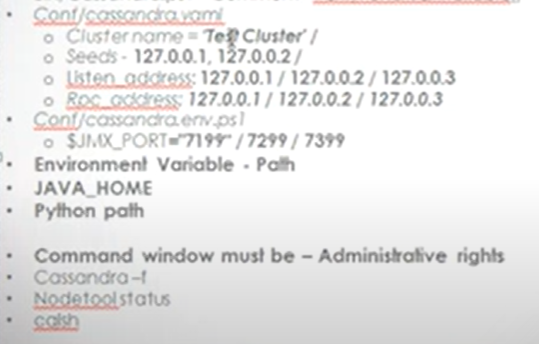
**5CS372 : Advanced Database System Lab.**

**Assignment No. 10**

**Cassandra Clustering**

1. Setup a multi-node Cassandra Cluster on single windows machine. Give your group name (2020GRP\*\*) to cluster. Follow the steps given in below link

Repeat above experimentation with physical more than 3 machines in lab.

1. Install Cassandra from official website, install older version
2. Extract it using 7zip
3. Rename it as 1
4. Keep it in a folder called cassandra in C drive
5. Run “cassandra.bat -f” in cmd in the above path
6. If it shows error , install java and set its env variable at JAVA\_HOME, use jre link or set execution policy
7. Make copies of 1 named 2 and 3 and save them in Cassandra folder too
8. Change data\_file\_directory of each folder in Cassandra.yaml
9. edit Cassandra.bat to change local.port number to 3 different port numbers in each folder
10. 
11. In bin/Cassandra.ps1 comment verifyPortsAreAvailable for each node
12. Nodetool status

Create a Docker network that will be used by the Cassandra containers to communicate with each other. Open a command prompt and enter the following command:-

**docker network create cassandra-net**

Start the first Cassandra node. Open a command prompt and enter the following command:-

**docker run --name cassandra-node1 --network cassandra-net -d cassandra:latest**

Start the second Cassandra node. Open a command prompt and enter the following command: -

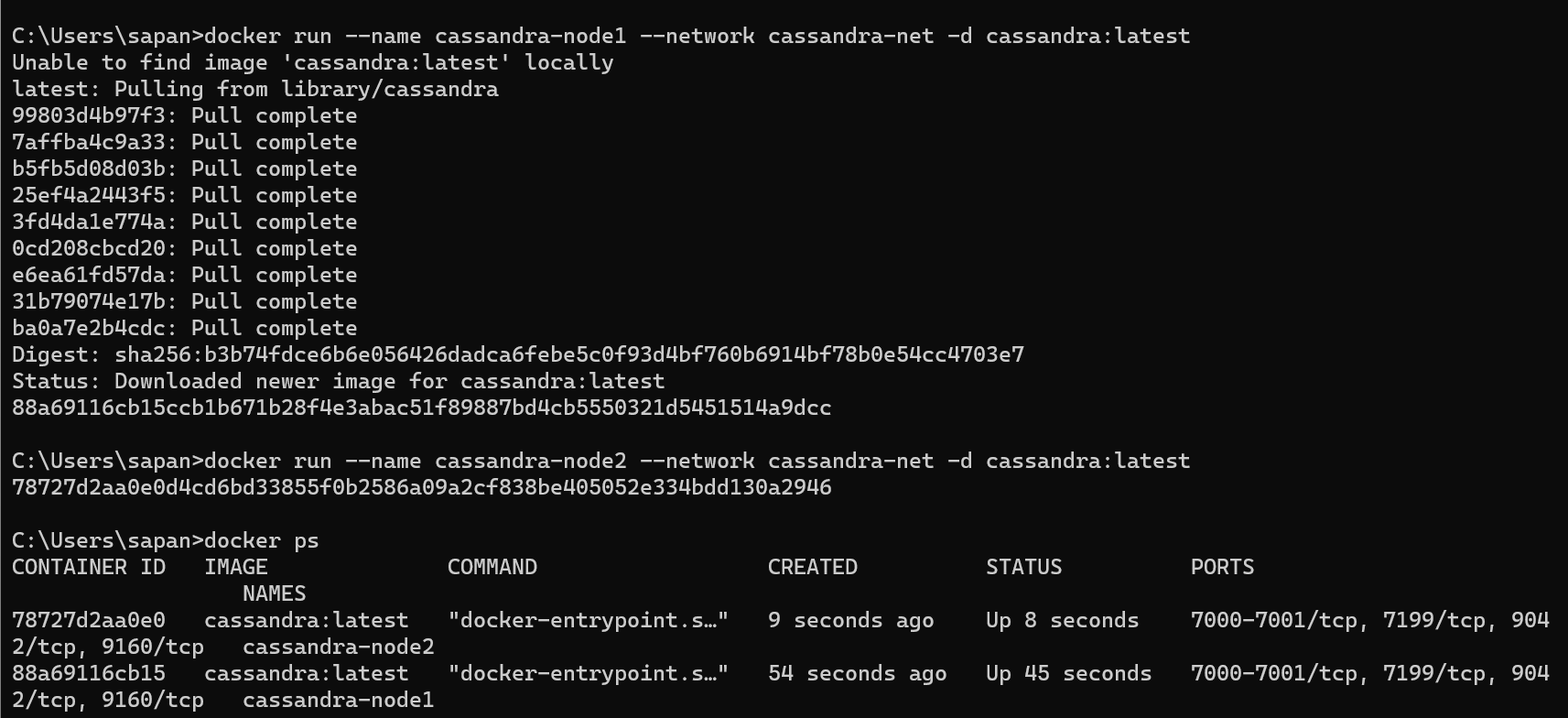
**docker run --name cassandra-node2 --network cassandra-net -d cassandra:latest**

This will start a new Cassandra container named cassandra-node2 on the cassandra-net network.

Verify that the Cassandra nodes are running. Open a command prompt and enter the following command:

**docker ps**

This should show a list of running Docker containers, including the cassandra-node1 and cassandra-node2 containers.



Connect to the Cassandra cluster. Open a new command prompt and enter the following command:

**docker exec -it cassandra-node1 cqlsh**

This will connect you to the first Cassandra node using the CQL shell.

Create a new keyspace and table in the cluster. In the CQL shell, enter the following commands:

**CREATE KEYSPACE 2020GRPCS15**

**WITH REPLICATION = {'class' : 'SimpleStrategy', 'replication\_factor' : 2};**

**USE 2020GRPCS15;**

**CREATE TABLE employees (**

**id int PRIMARY KEY,**

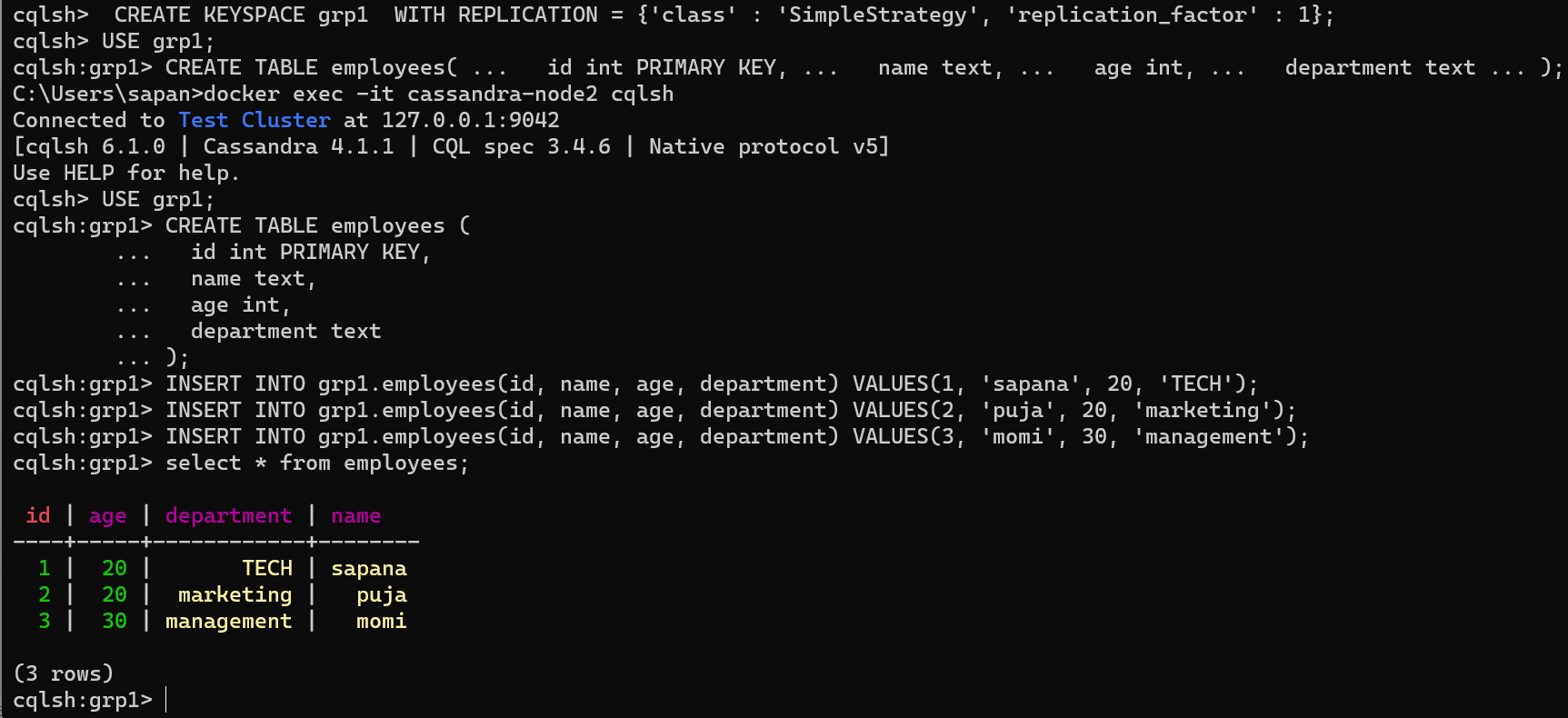
**name text,**

**age int,**

**department text**

**);**

This will create a new keyspace named 2020GRPCS15 with a replication factor of 2, and a new table named employees.



1. Install the DataStax OpsCenter community edition

<(>[https://www.datastax.com/blog/datastax-community-217-and-2016ready-](https://www.datastax.com/blog/datastax-community-217-and-2016-ready-download) [download)](https://www.datastax.com/blog/datastax-community-217-and-2016-ready-download) and configure it for above cluster formed.

**Configuration** –

Once the installation is complete, start the OpsCenter service by running the following command:

**sudo service opscenterd start**

Open a web browser and navigate to the OpsCenter web console by entering the following URL:

http://<opscenter-host>:8888/

* Replace <opscenter-host> with the hostname or IP address of the machine running OpsCenter.
* When prompted, enter the username and password for the OpsCenter web console.
* Once you are logged in to the web console, you can add a new cluster by clicking the "Add Cluster" button on the top right corner of the page.
* In the "Add Cluster" dialog, enter a name for the cluster and the contact points for the nodes in the cluster. The contact points should be the IP addresses or hostnames of the nodes that you want to monitor.
* Click the "Add" button to save the cluster configuration.

OpsCenter will now connect to the nodes in the cluster and start monitoring their status and performance. You can view various metrics and statistics for the nodes and the cluster in the OpsCenter web console.

1. Use Case - **Weather Station IoT Temperature Sensor Data** : There are set of weather stations at different remote location with “weatherStationID”. Each station record the temperature after every 5 minutes and push the data to nearest node in above cluster. Design the cluster database to hold these weather data. User should be able to retrieve the data in any dimensions.

1. Demonstrate the cluster operations of above use case using OpsCenter.