***Advanced Database System Lab***

**Assignment no. 10**

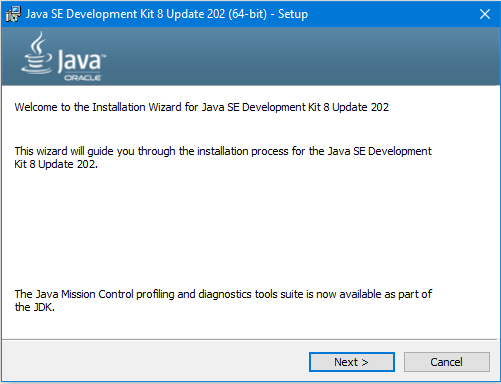
# **PRN:** 2020BTECS00005

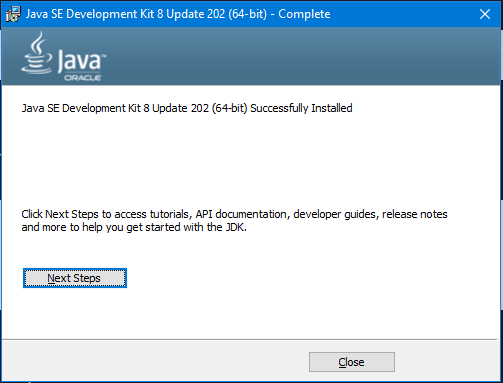
# **Name:** Sanket Shivaji Jadhav

* **Title:** Casandra Clustering.

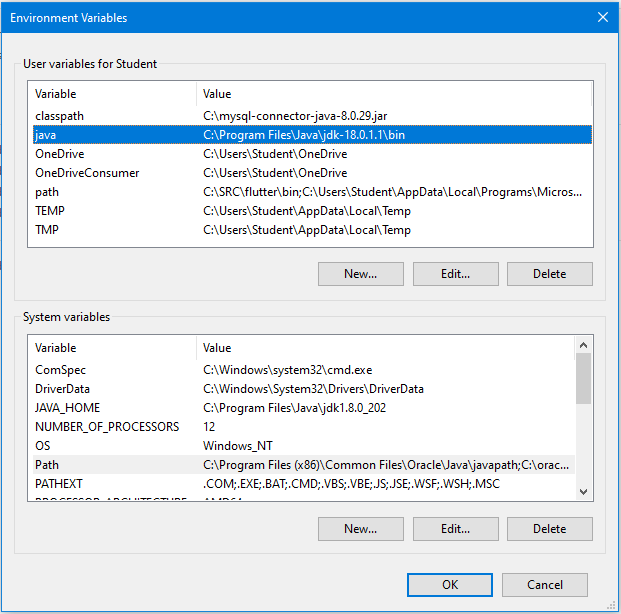
* **Aim:** Configure CassandraDB for multi-node cluster & configure DataStax OpsCenter for the same.

**Install Java 8**

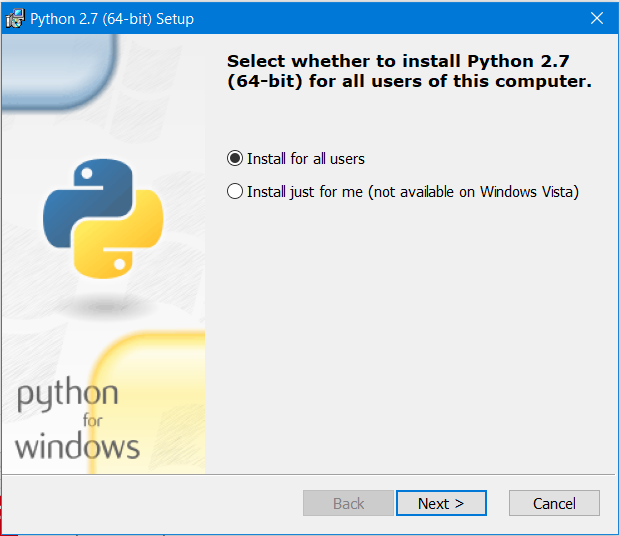
****

****

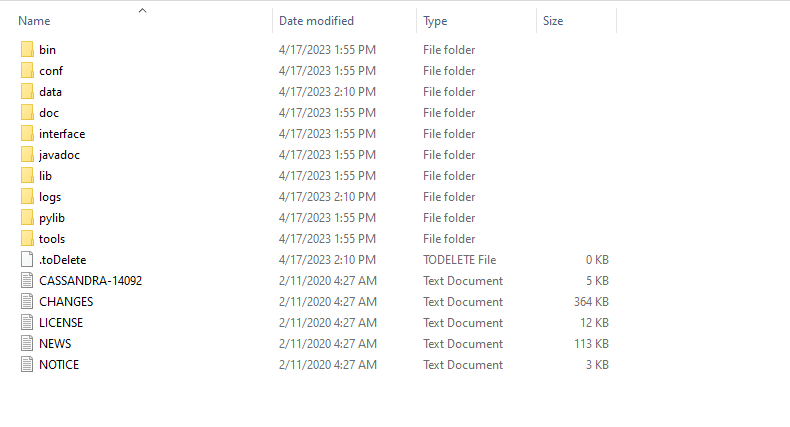
**Add path to environment variables**

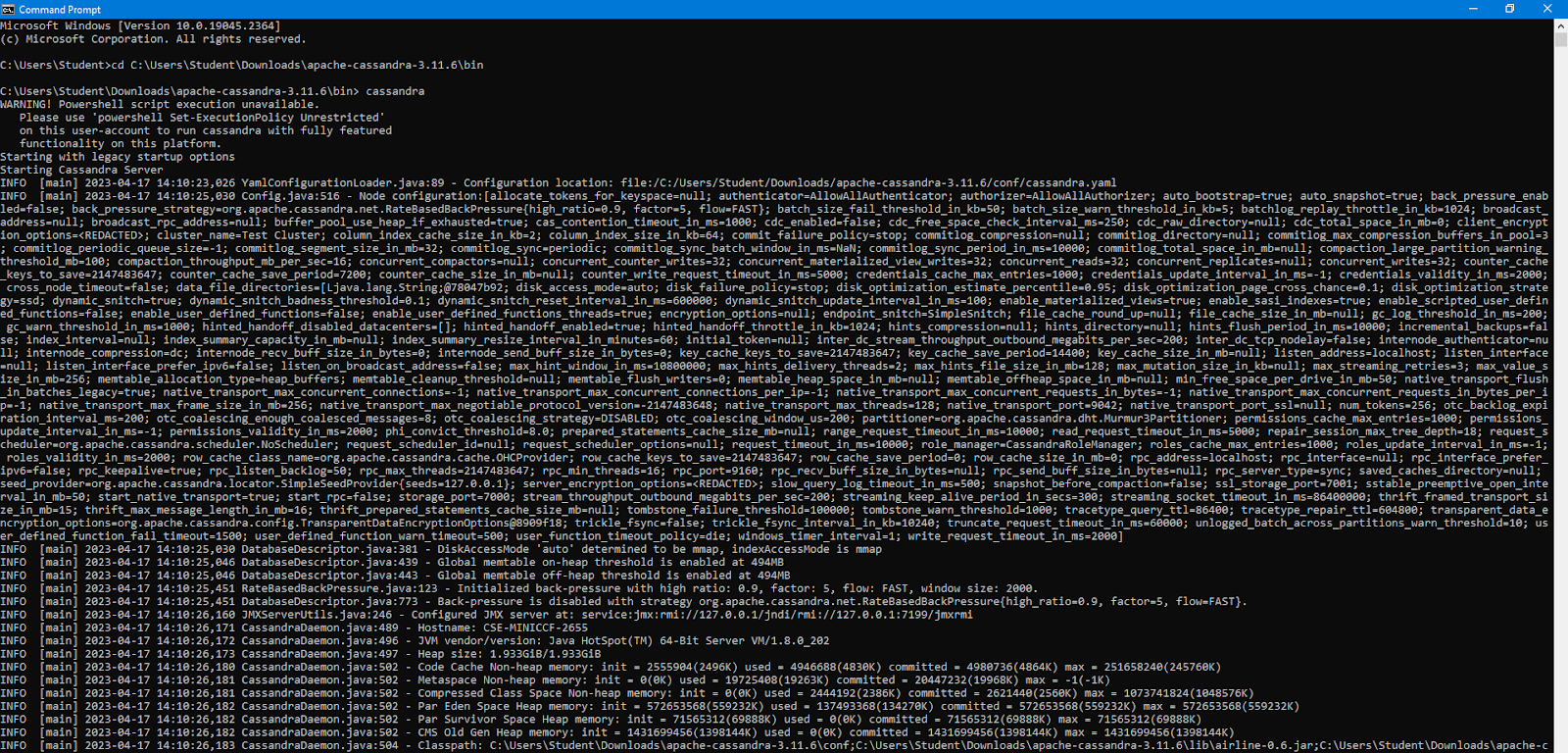
****

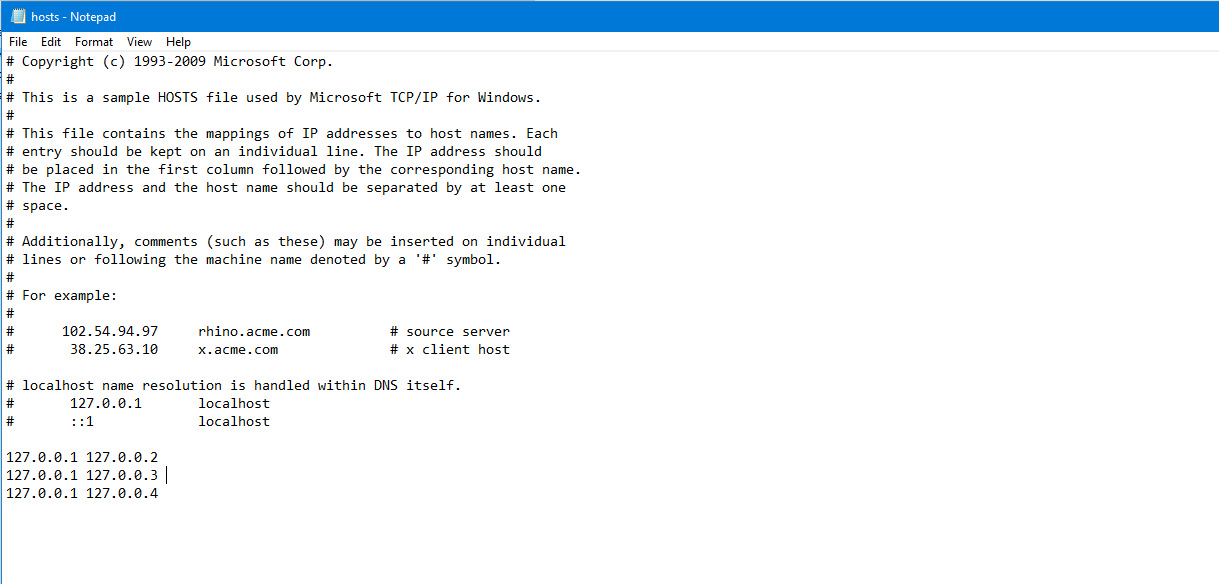
**Install Python 2.7**

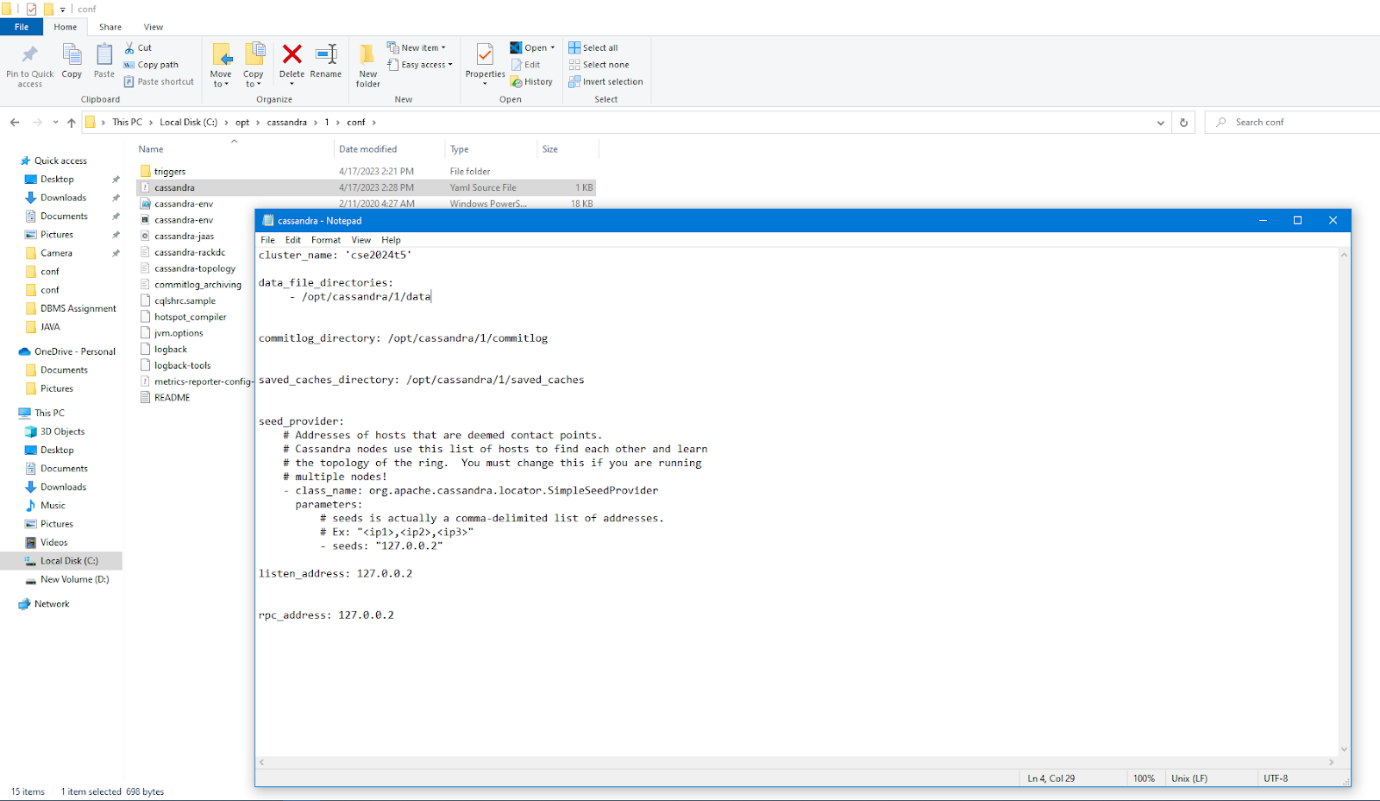
****

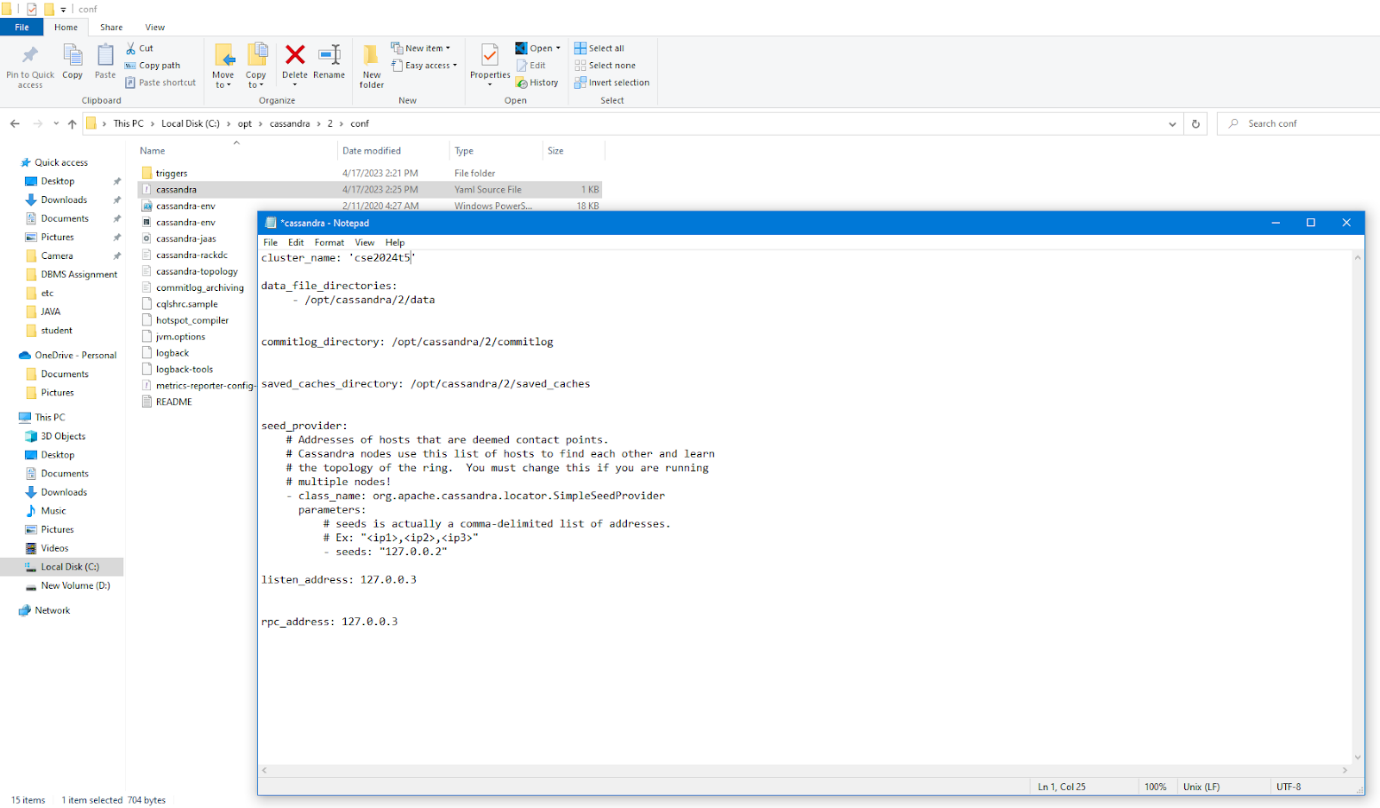
**Install Cassandra**

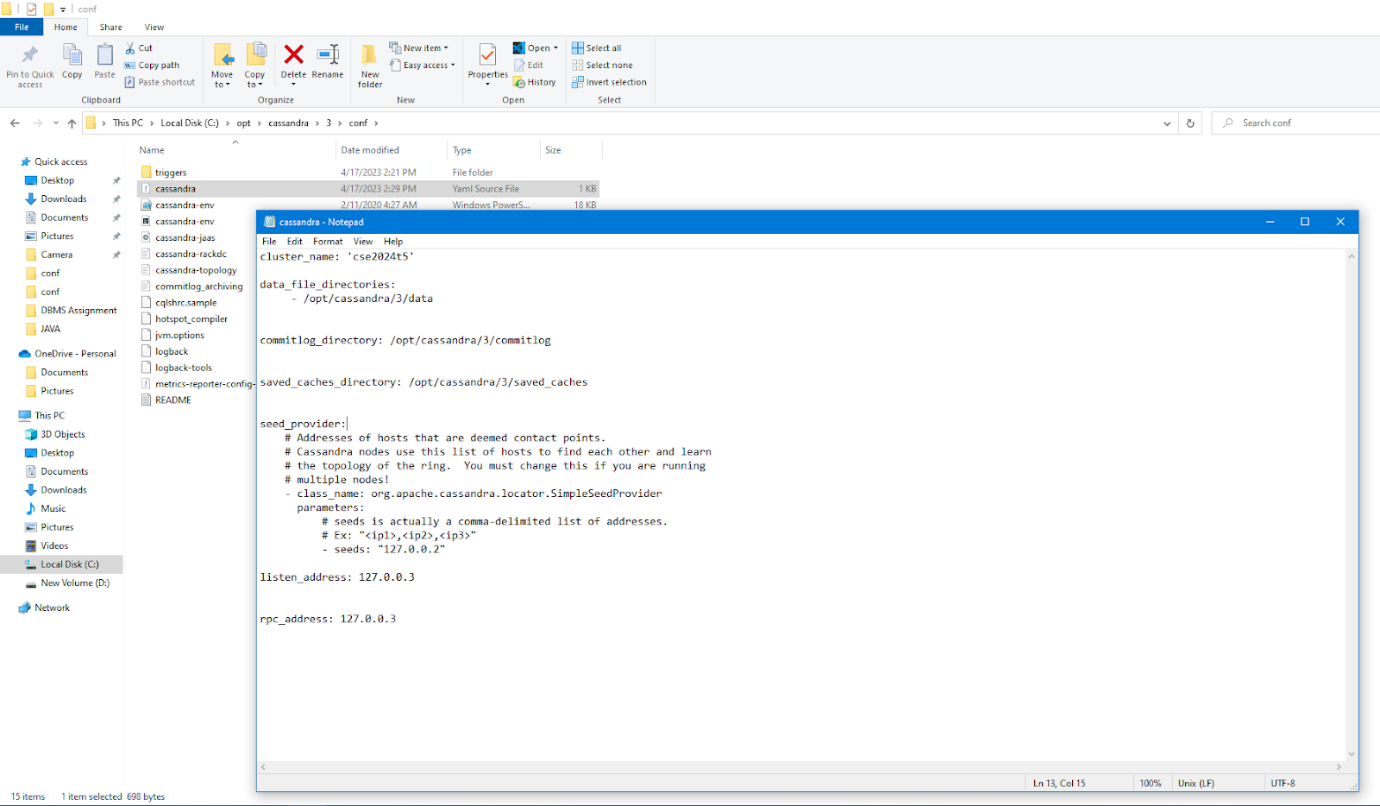
****

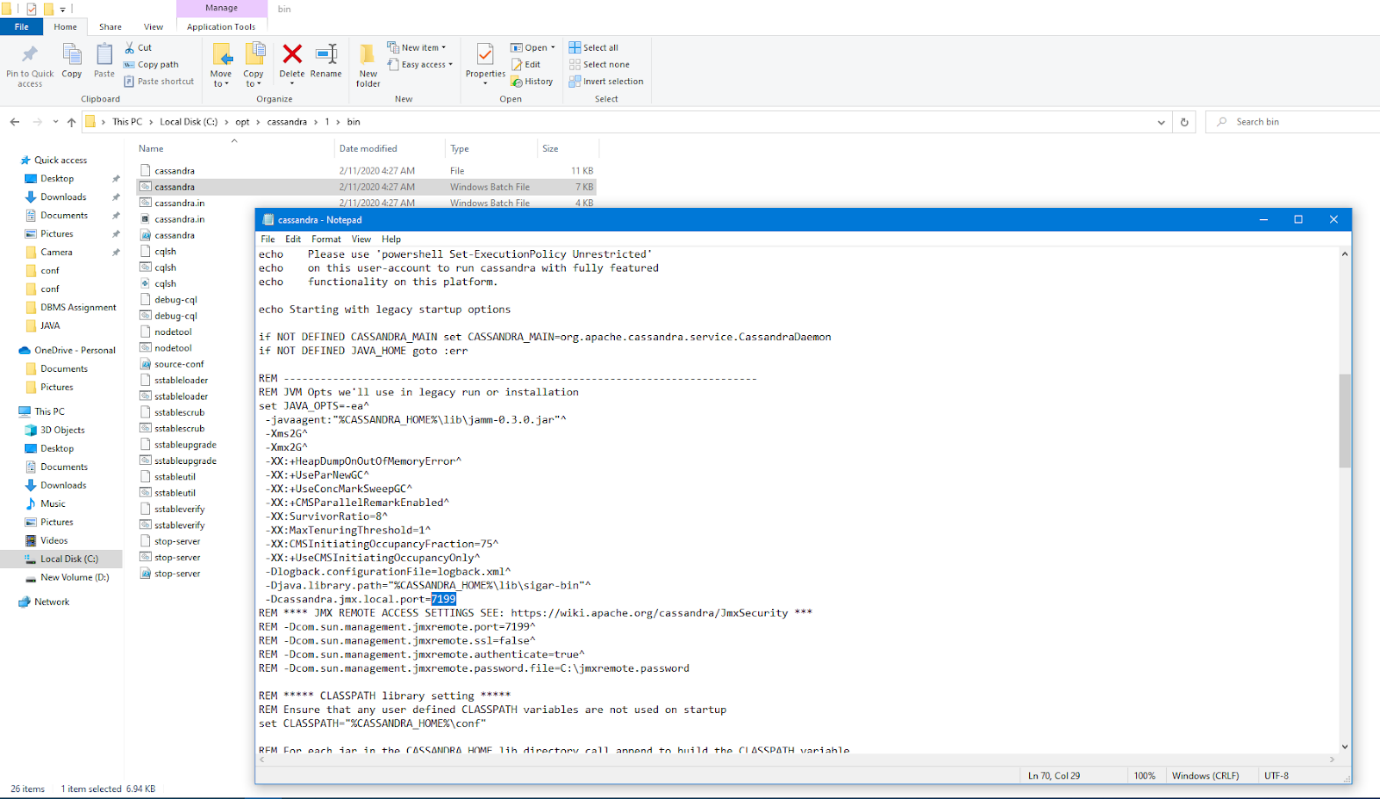
****

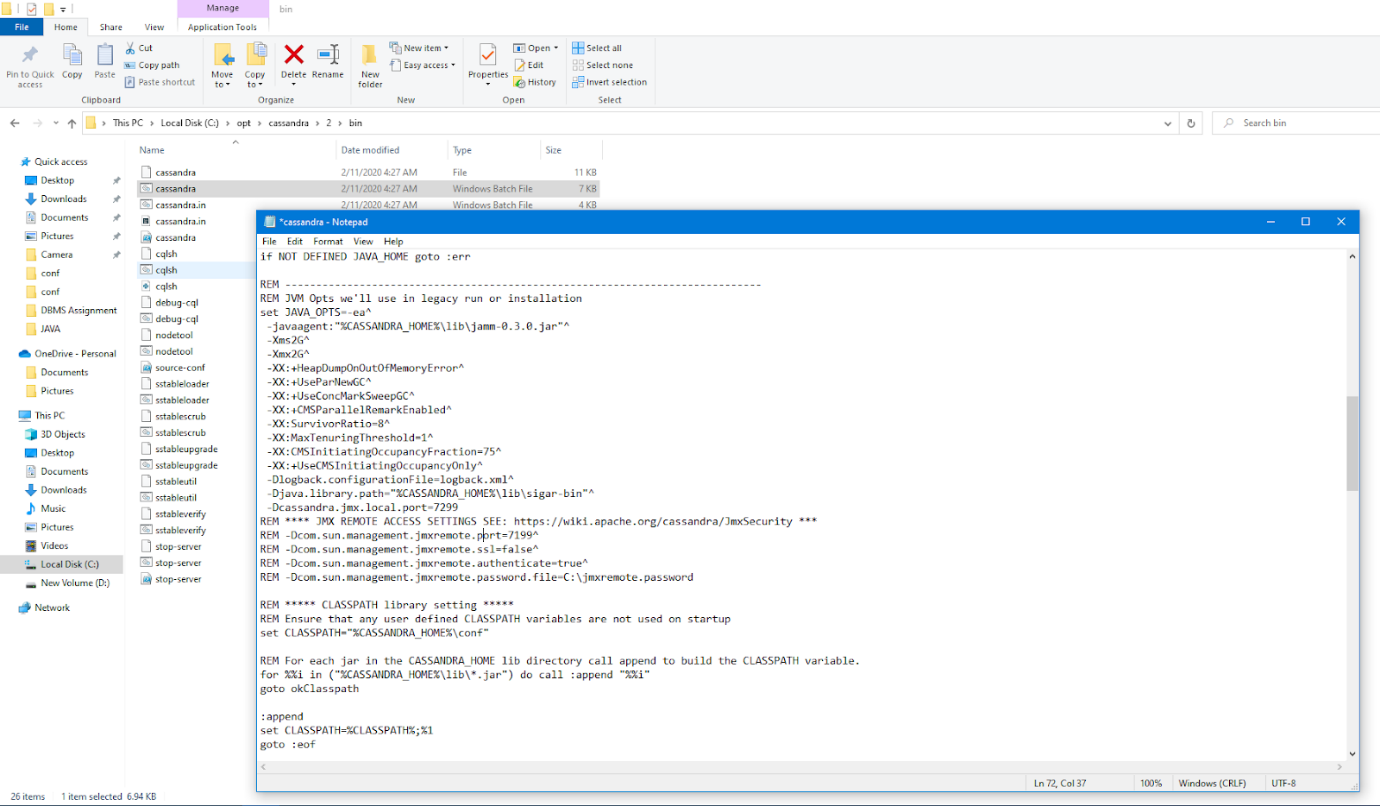
****

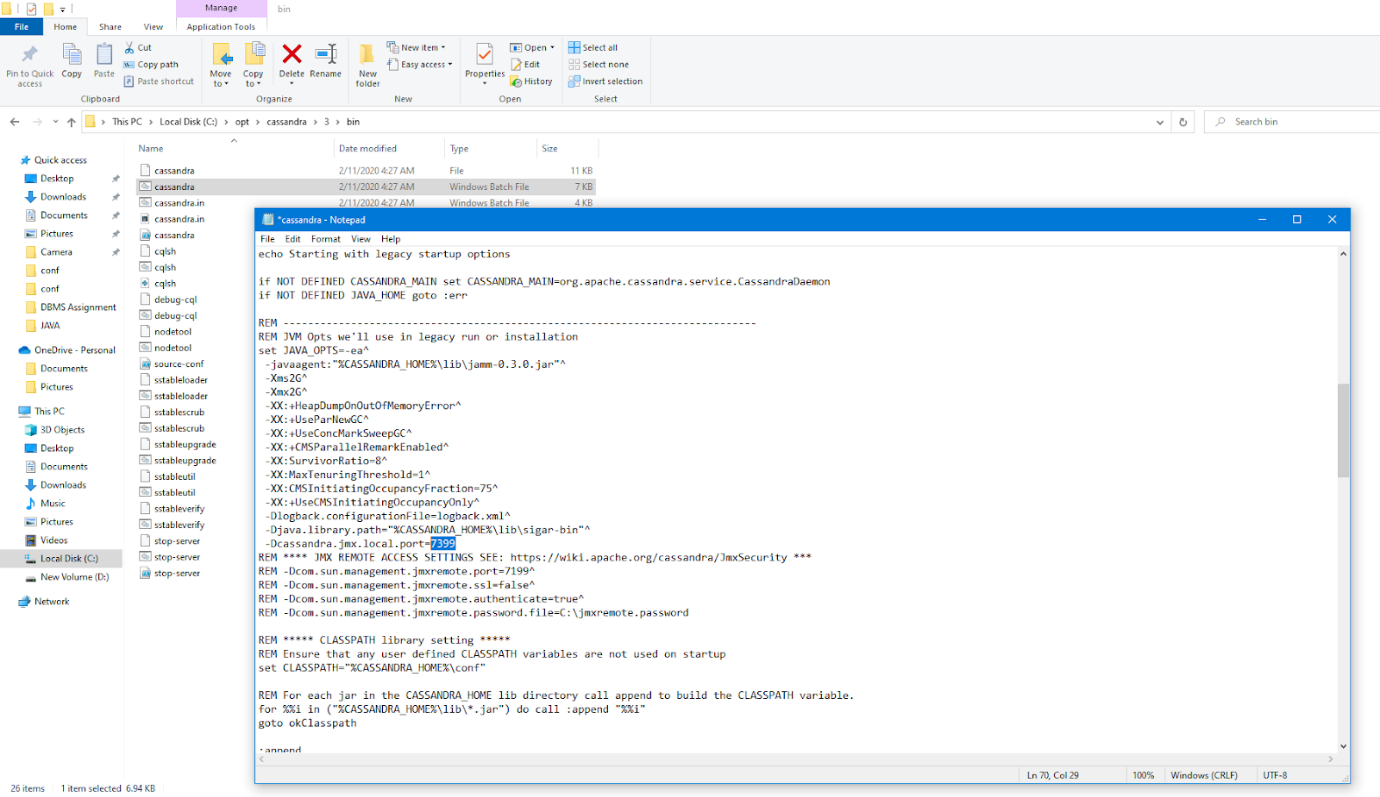
****

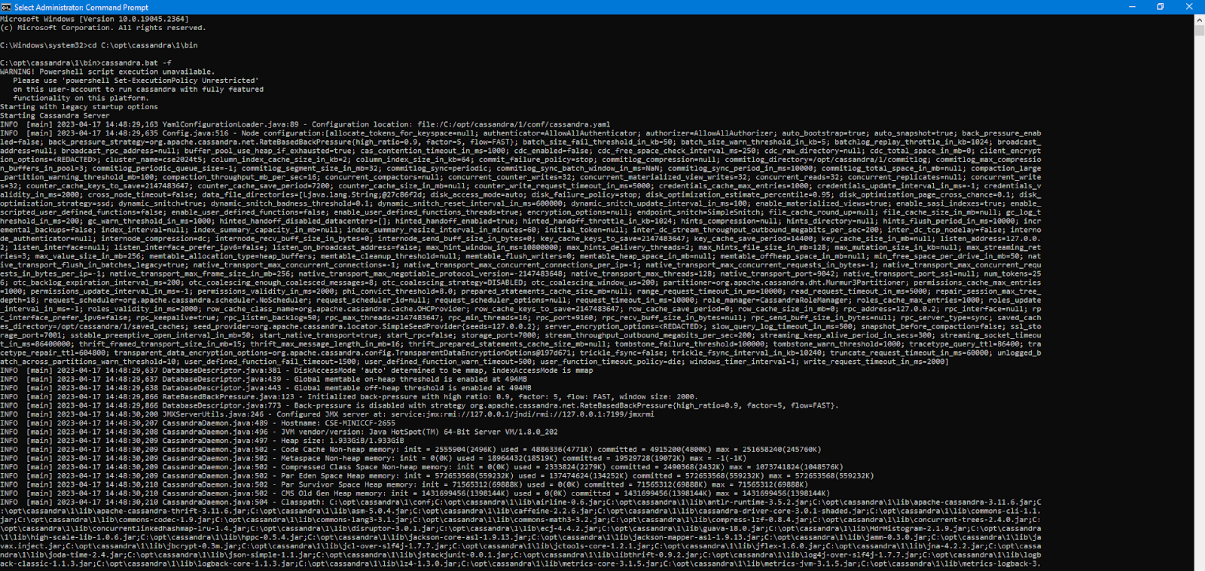
****

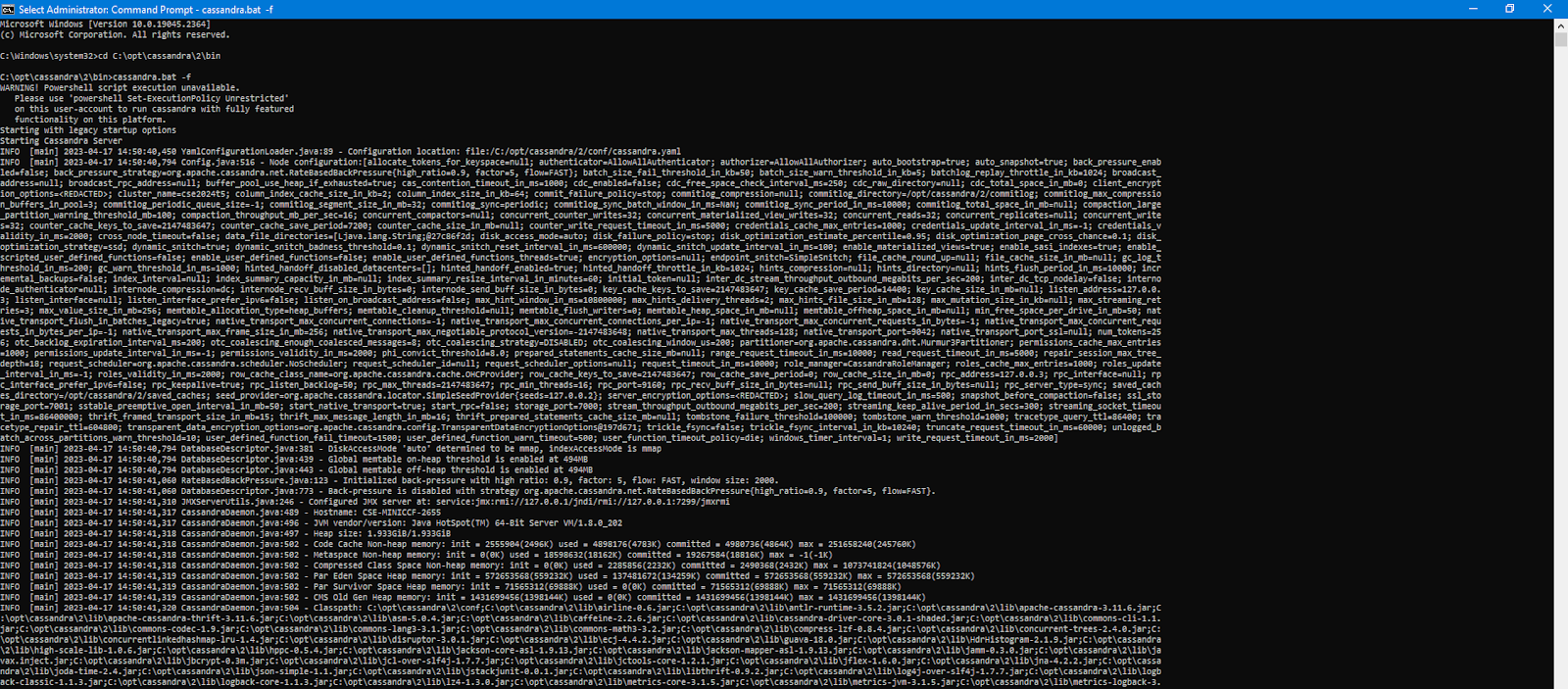
****

****

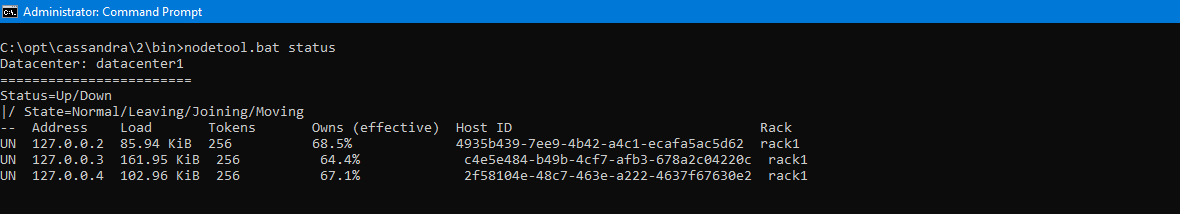
****

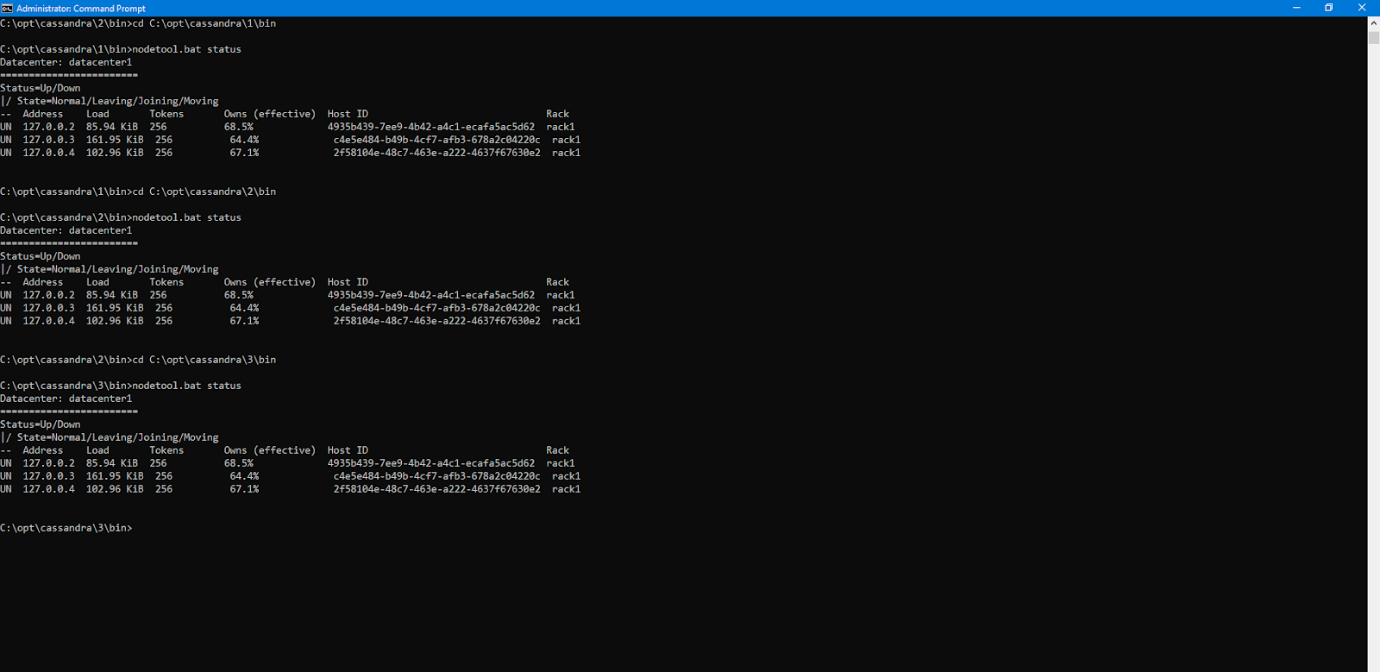
****

****

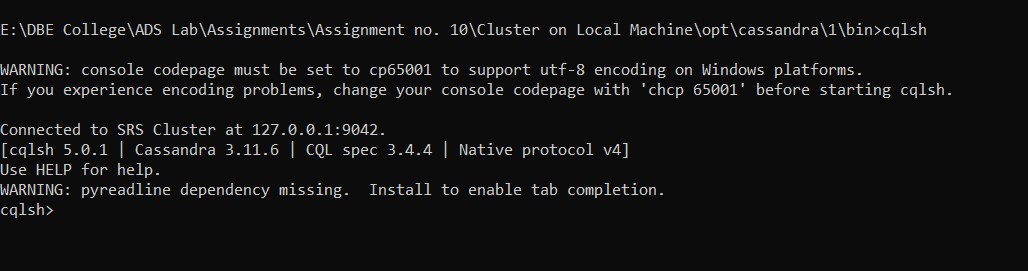
****

****

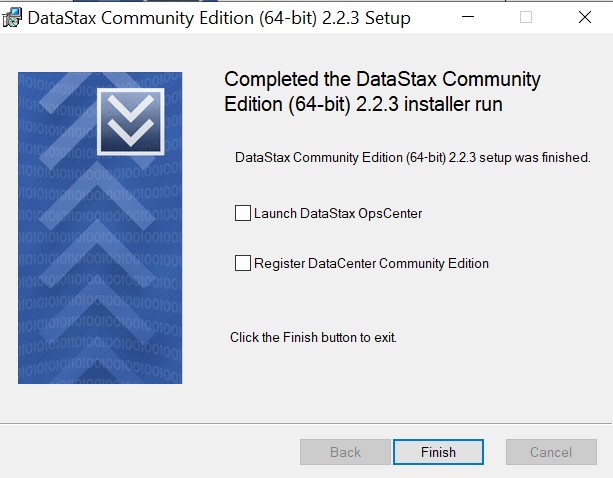
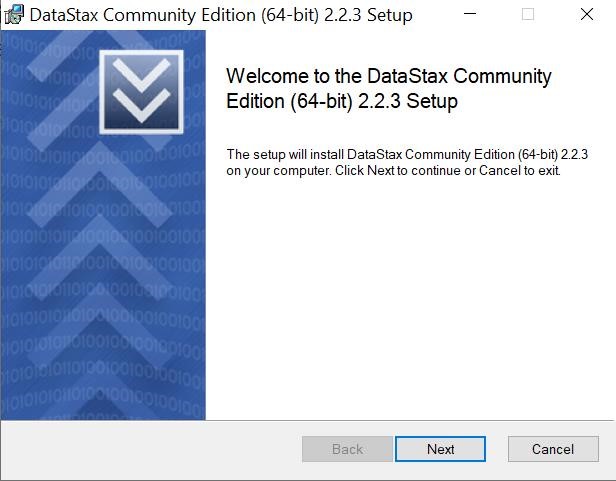
****

Ý****

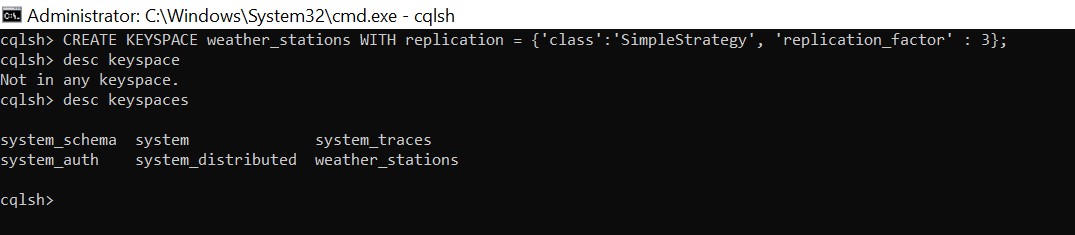
**Cluster Configuration**



**DataStax Community Edition OpsCenter:**

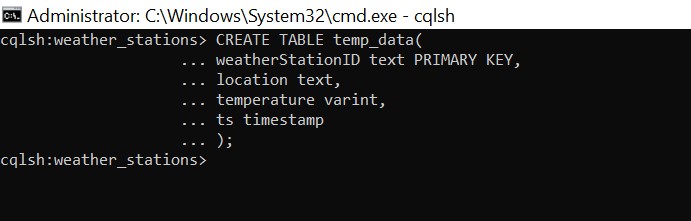


1. Creating Keyspace:



2.

Creating table:



1. Inserting data after every 5 mins:

|  |
| --- |
| import requests  from cassandra.cluster import Cluster import time    # Connecting to Cassandra DB # Configuring nodes & cluster nodes = ['127.0.0.1', '127.0.0.2', '127.0.0.3'] |
| cluster = Cluster(nodes)    # Connecting to the keyspace conn = cluster.connect('weather\_stations')  print("Connected to CassandraDB!")  # Weather API  CITIES = ['Sangli', 'Kolhapur', 'Pune']  API\_KEY = "58d0b46000077168350c534486d5ed29"  # Getting temperature def getTemp(city):  url =  "https://api.openweathermap.org/data/2.5/weather?q={}&appid={}".format(city  , API\_KEY)  resp = requests.get(url)  if resp.status\_code == 200: api\_data = resp.json() temp\_K = int(api\_data['main']['temp']) temp\_C = temp\_K - 273 return temp\_C else:  return 0    # inserting temperature data def insertTemp(node\_no, city):  query = "INSERT INTO temp\_data(weatherStationID,location,temperature,ts) VALUES(%(ws\_id)s,  %(loc)s, %(temp)s,toTimestamp(now()))" conn.execute(query, {"ws\_id": "WS"+str(node\_no), "loc": city, "temp":  getTemp(city)})  print("Inserted temperature data successfully by  WS{}!".format(node\_no))  while True: for i, c in enumerate(CITIES):  insertTemp(i+1, c) time.sleep(300) |

1. Viewing the inserted data:

