* **Title:** Comprehensive Guide to Setting Up a Multi-Node Cassandra Cluster and Designing a Database Schema for IoT Temperature Sensor Data

* **Objective:** The aim of this guide is to provide comprehensive instructions for setting up a multi-node Cassandra cluster, installing and configuring DataStax OpsCenter, designing an efficient database schema tailored for storing weather station IoT temperature sensor data, and demonstrating cluster operations utilizing OpsCenter.

* **Theory and Concepts:** Clustering, in the context of Cassandra, is the fundamental mechanism for distributing and organizing data across multiple nodes within a cluster. Cassandra, being a distributed database system, prioritizes high availability and scalability, with clustering playing a pivotal role in achieving these objectives.

* **Data Distribution:** In a Cassandra cluster, data distribution is managed through partitioning, where each node is tasked with storing a specific portion of the dataset. Through intelligent partitioning strategies, Cassandra ensures even distribution of data across the cluster, optimizing performance and fault tolerance.

* **Replication:** To bolster fault tolerance and ensure data availability, Cassandra employs replication techniques, wherein copies of data are stored across multiple nodes. The replication factor, a configurable parameter, dictates the number of copies maintained and their placement across the cluster, enhancing resilience against node failures.

* **Scalability:** Cassandra's architecture facilitates horizontal scalability, allowing the cluster to expand seamlessly by adding more nodes. This scalability feature empowers organizations to accommodate growing data volumes and increasing workloads without encountering downtime or performance bottlenecks.

* **Comprehensive Guide Overview:**
* **Setting Up a Multi-Node Cassandra Cluster:**

Installation of Cassandra on multiple nodes.

Configuration of cluster settings, including seeds and communication protocols.

Validation of cluster connectivity and node synchronization.

* **Installing and Configuring DataStax OpsCenter:**

Deployment of OpsCenter for centralized cluster management and monitoring.

Configuration of monitoring thresholds and alerting policies.

Integration with existing authentication mechanisms for enhanced security.

Designing a Database Schema for Weather Station IoT Temperature Sensor Data:

* **Analysis of data requirements and access patterns.**

Development of an optimized schema leveraging Cassandra's data modeling principles.

Consideration of data partitioning and replication strategies to ensure efficient data storage and retrieval.

* **Demonstrating Cluster Operations Using OpsCenter:**

Utilizing OpsCenter's intuitive interface for cluster administration tasks.

Performing routine maintenance operations, such as node additions or removals.

Monitoring cluster health metrics and diagnosing performance issues.

**Multi node cluster on single machine**

**Steps :**

Install the Cassandra instance on the machine

Create a folder for storing the 3 instances of the Cassandra

Create 3 folders ( 3 nodes ) and copy the Cassandra instances in each of the folder

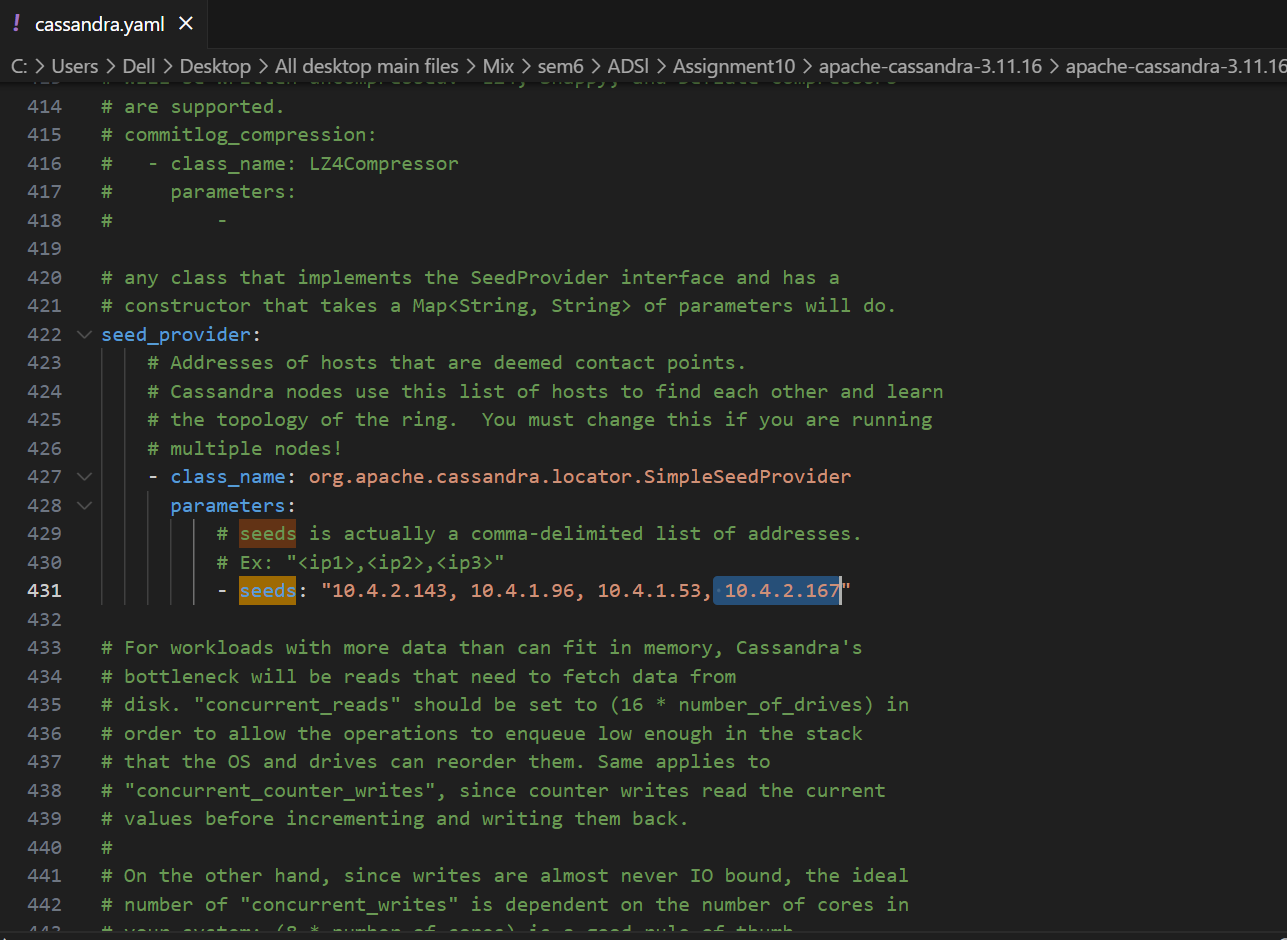
For each of the node in the cluster repeat the following process

* 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
* Then test the cluster using cassandra.bat -f in 1st node Then start the instances using Cassandra -f in each node To see the status of the cluster run command : nodetool status 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





1. **Configuration details of each node**

* **Node 1 :**

**Ip\_address:** 10.4.2.143

* **Node 2 :**

**Ip address:** 10.4.1.96

* **Node 3 :**

**Ip address:** 10.4.1.53

* **Node 4:**

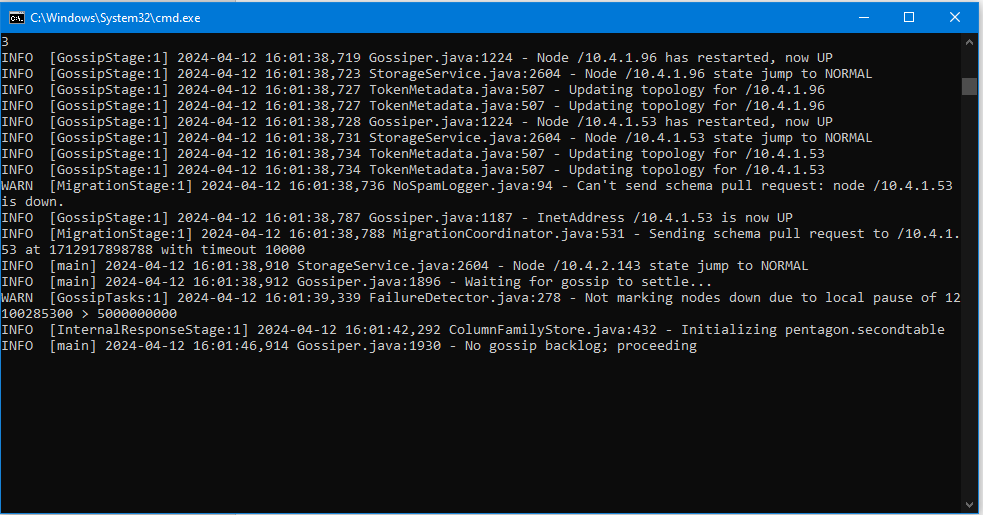
**Ip address:**  10.4.2.167

**Setting up cluster name**

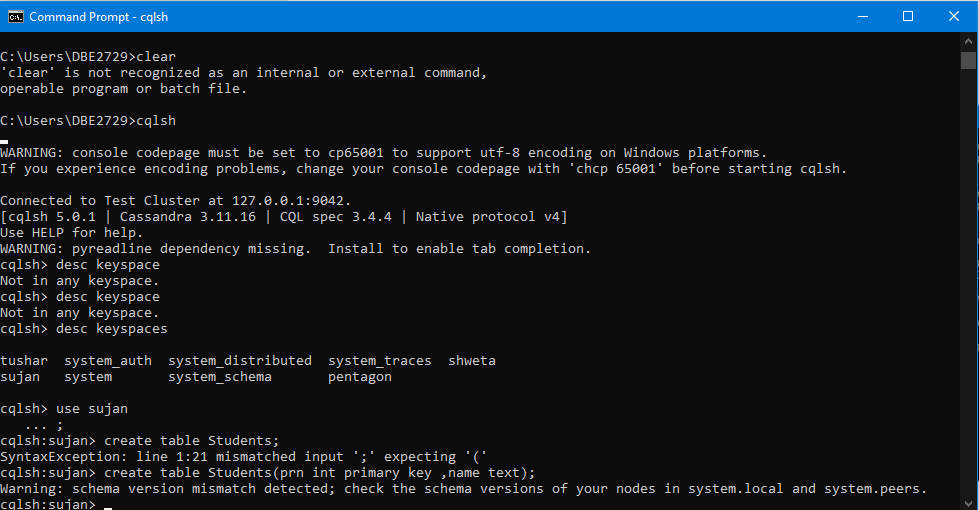
**Starting the node 1**

**Command : Cassandra**

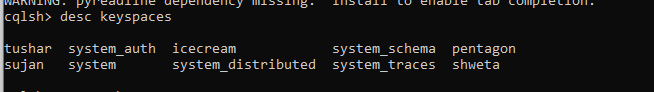
**Starting Gossip protocol**

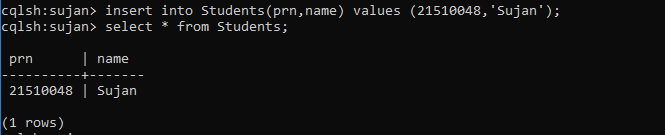


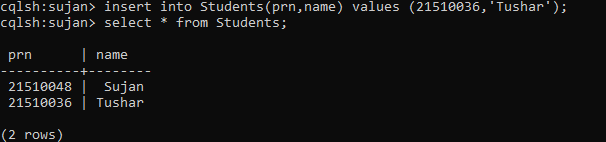
Cassandra provides a prompt Cassandra query language shell (cqlsh) that allows users to communicate with it. Using this shell, you can execute Cassandra Query Language (CQL).



An object called a keyspace is used to **store user-defined types and column families**. A keyspace is similar to an RDBMS database in that it includes indexes, user-defined types, indexes, replication factors, strategy, column families, data center awareness, etc.







**Conclusion :**   
In summary, this assignment encapsulates the comprehensive process of establishing a multi-node Cassandra cluster, deploying and fine-tuning DataStax OpsCenter, while delving into essential concepts such as Cassandra clustering and pivotal network configuration parameters including rpc\_address, listen\_address, and seeds. By engaging in practical tasks and hands-on exercises, participants not only grasp the fundamentals of distributed database systems but also delve into intricate facets of data modelling, proactive monitoring, and proficient cluster management. This holistic approach fosters a deeper understanding of the intricacies involved in orchestrating and optimizing Cassandra clusters for real-world applications, equipping learners with invaluable skills essential for managing robust distributed database infrastructures.