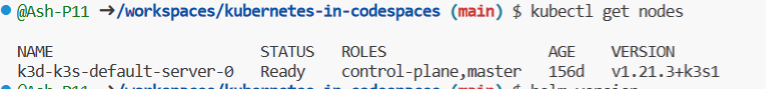
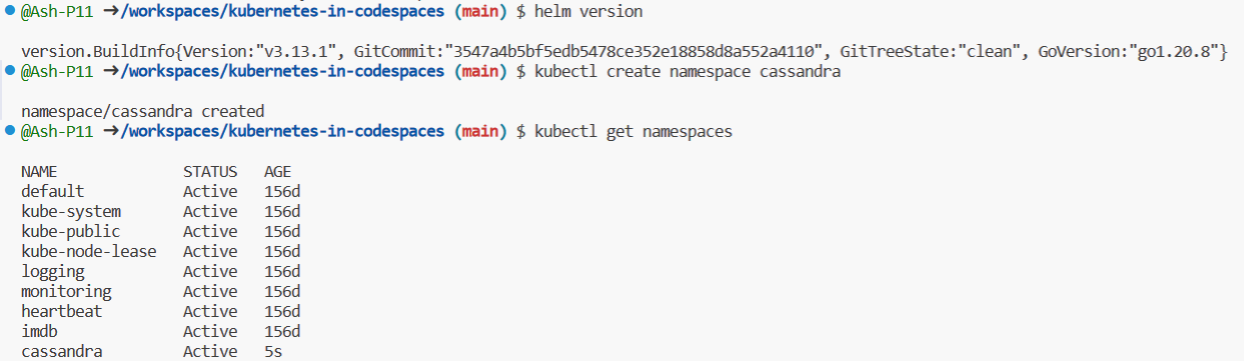
**Advanced Open Source Configuration**

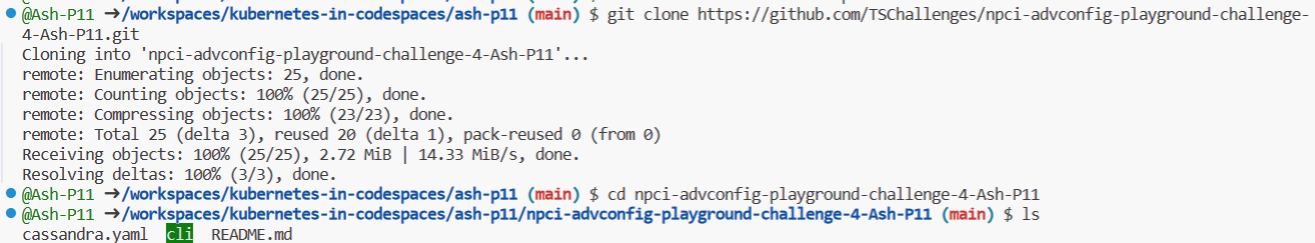
**Playground Challenge – 4 Submission**

**Cassandra Kubernetes Deployment & Optimization**

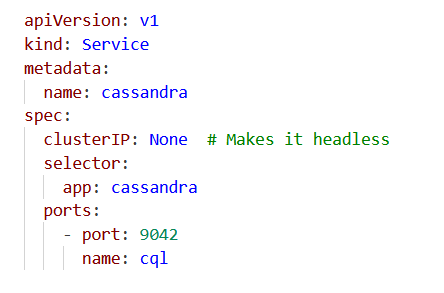
**Part 1: Cassandra Setup on Kubernetes**

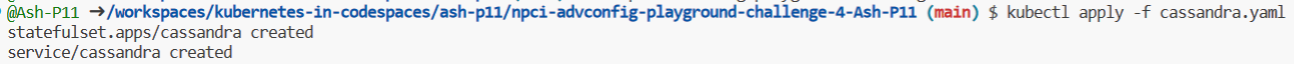




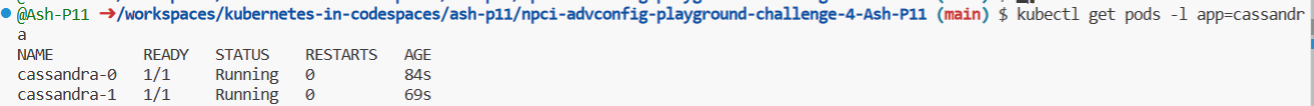


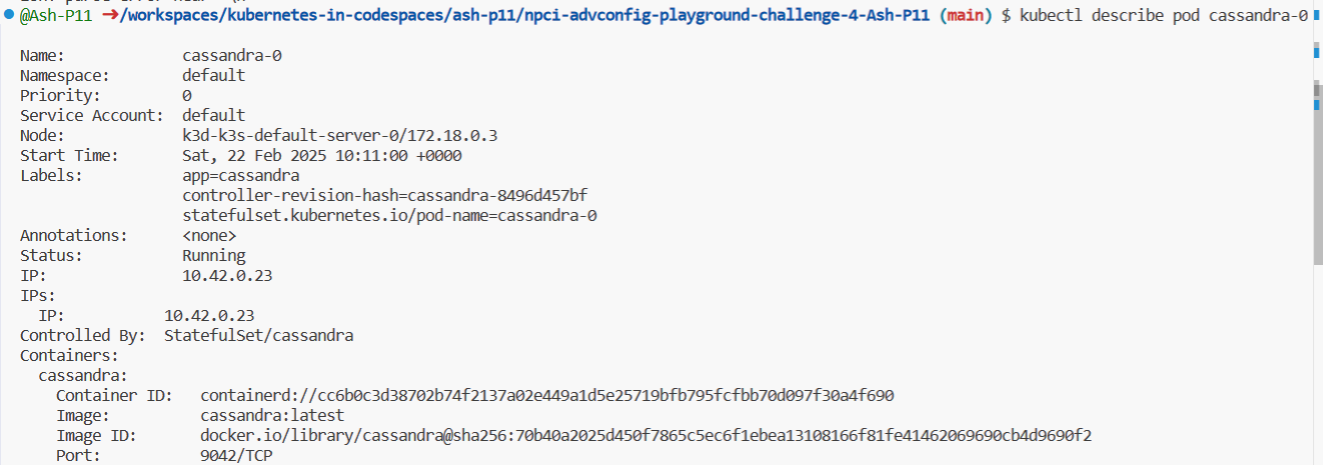
**Deploy a two-node Cassandra cluster on Kubernetes:**



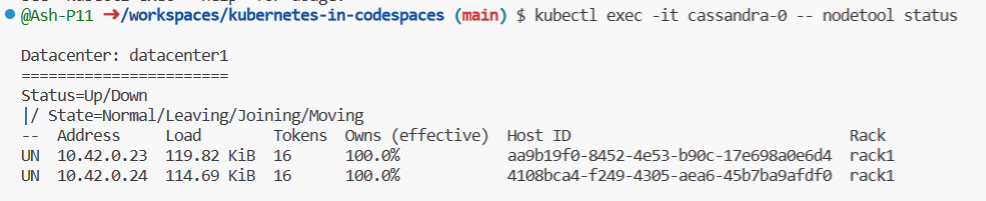
Verify the cluster is running and healthy





#### ****Part 2: Cassandra Cluster Monitoring****

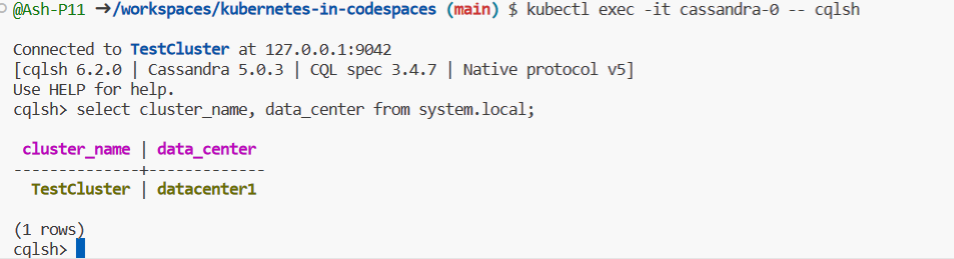
1. Use nodetool to check the status of the cluster and ensure all nodes are in an UN (Up Normal) state.



1. Use nodetool info to gather resource usage metrics (disk space, memory, etc.).

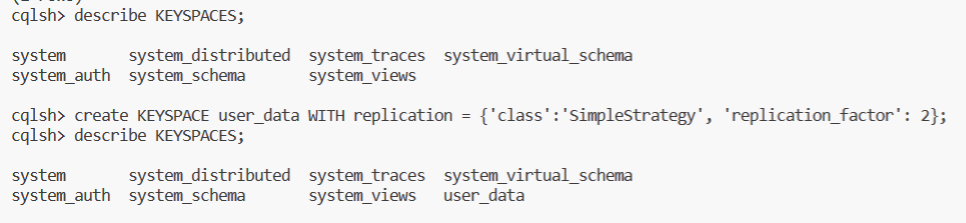


1. Connect to one of the Cassandra nodes using cqlsh and ensure that you can execute CQL commands.



#### ****Part 3: Running CQL Basic Commands****

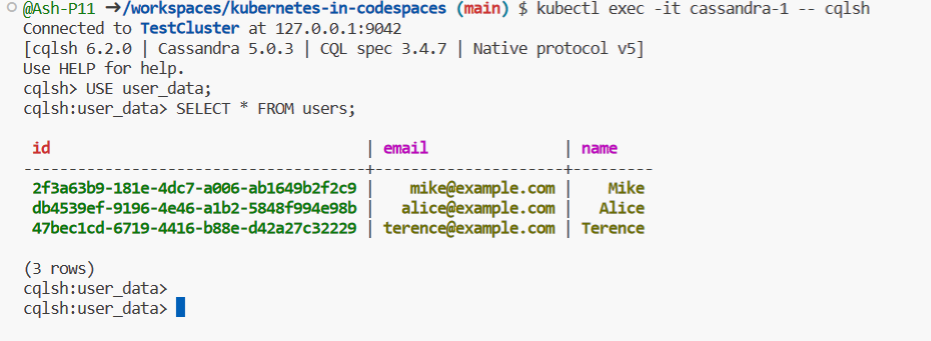
1. Create a keyspace named **user\_data** with replication factor 2.



1. Create a table **users** with columns id, name, and email.
2. Insert three records for the below users and query them to verify insertion:
   * Alice, [alice@example.com](mailto:alice@example.com)
   * Terence, [terence@example.com](mailto:terence@example.com)
   * Mike, [mike@example.com](mailto:mike@example.com)



1. Login to the 2nd node and test if all the table data is accessible



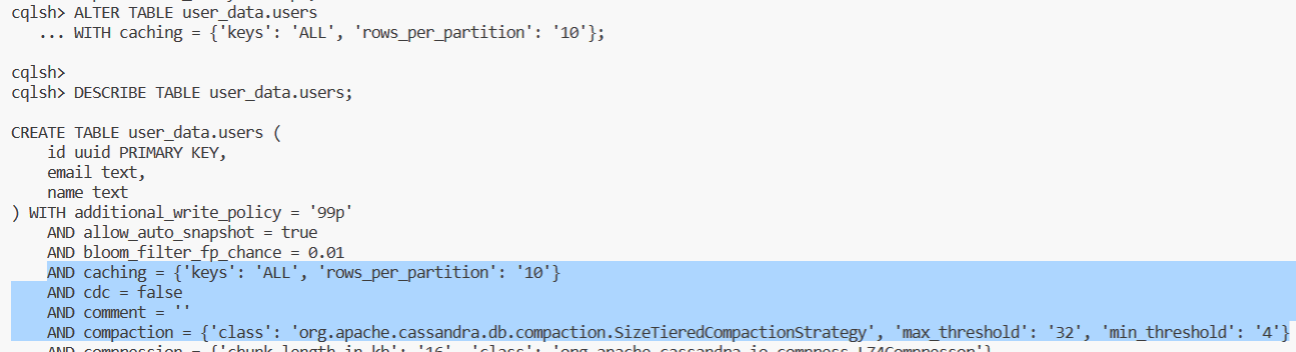
**Part 4: Performance Optimization**

The development team complains that queries on the users table are slow. Complete the following tasks to resolve the issues.

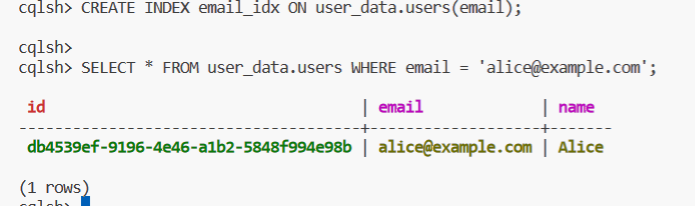
1. Check the existing caching and the compaction strategies for the users table.



1. Implement a caching strategy to set the rows for each partition to 10.



1. Searching by email is frequently required. Optimize read performance of such queries by adding an appropriate indexing strategy.



1. Use batch write queries to insert 2 new user records into the table:
   * Priya, [priya@example.com](mailto:priya@example.com)
   * Shahid, [shahid@example.com](mailto:shahid@example.com)

