**Advanced Databases**

**MongoDB CA**

**Student Number:** C19366191

**Student Name:** Lina Mir

**Programme Code:** TU856

1. Setting up the cluster and replication

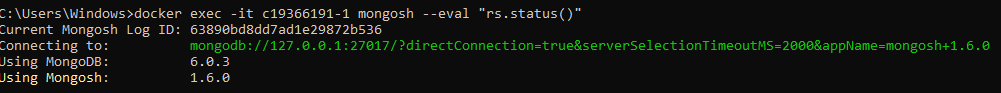
<<describe your cluster, now many nodes etc.>>

This cluster has three nodes connected to it.

The three nodes are c19366191-1, c19366191-2, c19366191-3

<<use rs.status to verify that the cluster and replication is working, include a screenshot, comment on the information provided>>

Rs.status() was used on the cluster



Here it shows that it is connecting to MongoDB and which version it is currently using. It also shows the version of Mongosh and the log ID.

Text

Description automatically generated

This shows the different node members in the cluster. This node is called c19366191-1. It is a secondary node.

Text

Description automatically generated

In this node with an ID of 1, the name of it is c19366191-2 and this is a primary node.

Below is the third node with an ID of 2 and is a secondary node called c19366191-3

Text

Description automatically generated

A database called c19366191 was created:

A black background with white text

Description automatically generated with low confidence

1. Porting the data to Mongo

<<verify that the data exists>>

<<use db.collection.find >>

The python file called c19366191.py converts the data from Cassandra to mongodb.

1. Working with the Golf collection in MongoDB:
   1. Basic query on golf data involving a text field.

<<use explain, capture the output and include it here, comment on what is happening>>

* 1. Adding a secondary index to golf data on a text field.

<<use explain, capture the output and include it here, comment on what is happening>>

1. Working with aggregation in MongoDB:
   1. Create an aggregation pipeline

<<use explain, capture the output and include it here, comment on what is happening>>

* 1. Add relevant indexes and reorder your stages.

<<use explain, capture the output and include it here, comment on what is happening>>

1. Replication working

<<pause/stop your primary node>>

<<use rs.status to show that another node has been elected >>