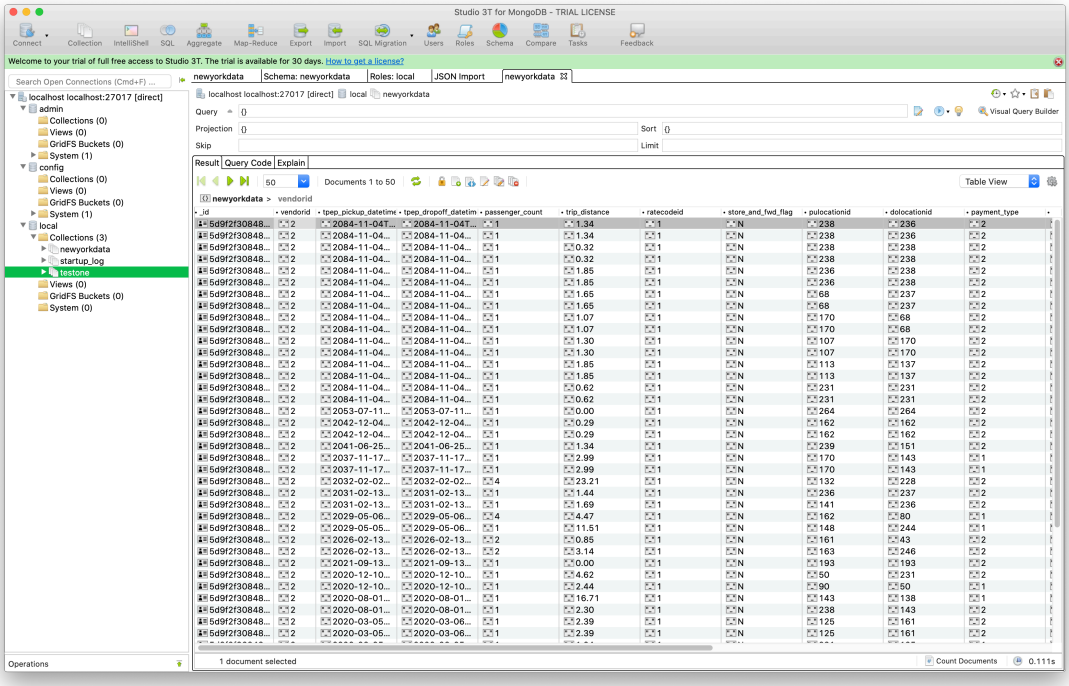
Part 1 - Design (weighted factor for grades = 2)

1. Design and explain interactions between main components in your architecture of mysimbdp (1 point)

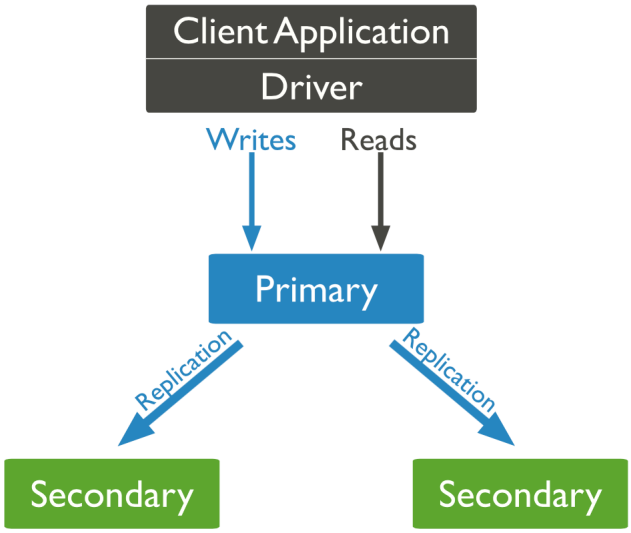
The mysimpbdp-coredms is MongoDB. For mysimbdp-daas, the producers availability of the data I used mongo engine and pymongo. In mysimbdp-dataingest I used Studio3T.



2. Explain how many nodes are needed in the deployment of my simbdp-coredms so that this component can work property (theoretically based on the selected technology) (1 point)

A replica set is a group of mongod instances that maintain the same data set. A replica set contains several data bearing nodes and optionally one arbiter node. Of the data bearing nodes, one and only one member is deemed the primary node, while the other nodes are deemed secondary nodes.

The primary node receives all write operations. A replica set can have only one primary capable of confirming writes with write concern; although in some circumstances, another mongod instance may transiently believe itself to also be primary. The primary records all changes to its data sets in its operation log, i.e. oplog. For more information on primary node operation, see Replica Set Primary.



3. Will you use VMs or containers for mysimbdp and explain the reasons for each component (1 point)

I will use containers wit the requisites.

4. Explain how would you scale mysimbdp to allow a lot of users using mysimbdp-dataingest to push data into mysimbdp (1 point)

I used Pymongo for the acces to mongoDB using python. Creating users, ingesting data, reading the data.

5. Explain your choice of industrial cloud infrastructure and/or mysimbdp-coredms provider, when you do not have enough infrastructural resources for provisioning mysimbdp (1 point)

I will choose amazon documentDB because they claim they are fast, scalable, and highly available document database that is designed to be compatible with your existing MongoDB applications and tools. Amazon DocumentDB uses a purpose-built SSD-based storage layer, with 6x replication across 3 separate Availability Zones. The storage layer is distributed, fault-tolerant, and self-healing, giving you the the performance, scalability, and availability needed to run production-scale MongoDB workloads.
https://aws.amazon.com/es/blogs/aws/new-amazon-documentdb-with-mongodb-compatibility-fast-scalable-and-highly-available/

Part 2 - Development and deployment (weighted factor for grades = 2)

1. Design and explain the data schema/structure for mysimbdp-coredms (1point)

We have the DB storage in localhost. With one collection named newyorkdata, where we ingest the json files from the new York taxi data using STUDIO 3T.

2. Explain how would you partition the data in mysimbdp-coredms into different shards/partitions (1 point)

I will make the partition depending on the amount of data needed to storage. After, I will make depending on the date of the trip.

3. Write a mysimbdp-dataingest that takes data from your selected sources and stores the data into mysimbdp-coredms (1 point)

Using STUDIO 3T, it is easy to import and export data from/to the mongoDB.

4. Given your deployment environment, show the uploading performance(response time and failure) of the tests for 1,5, 10, .., n of concurrent mysimbdp-dataingest pushing data into mysimbdp-coredms (1 point)

5. Observing the performance and failure problems when you push a lot of data into mysimbdp-coredms (you do not need to worry about duplicated data in mysimbdp), propose the change of your deployment to avoid such problems (or explain why you do not have any problem with your deployment) (1 point)