Estes códigos foram utilizados tanto no Docker Desktop, quanto no MongoDB Compass

docker network create mongoCluster11

docker run -d --rm -p 27017:27017 --name mongo1 --network mongoCluster11 mongodb/mongodb-communi

ty-server:latest --replSet myReplicaSet --bind\_ip localhost,mongo1

docker run -d --rm -p 27018:27017 --name mongo2 --network mongoCluster11 mongodb/mongodb-communi

ty-server:latest --replSet myReplicaSet --bind\_ip localhost,mongo2

docker run -d --rm -p 27019:27017 --name mongo3 --network mongoCluster11 mongodb/mongodb-communi

ty-server:latest --replSet myReplicaSet --bind\_ip localhost,mongo3

docker run -d --rm -p 27020:27017 --name mongo4 --network mongoCluster11 mongodb/mongodb-communi

ty-server:latest --replSet myReplicaSet --bind\_ip localhost,mongo4

docker exec -it mongo1 mongosh

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.4.2

db.runCommand ({hello:1})

rs.initiate({

... members: [

... { \_id: 0, host: "mongo1" },

... { \_id: 1, host: "mongo2" },

... { \_id: 2, host: "mongo3" },

... { \_id: 3, host: "mongo4" }

... ]

... });

exit

docker exec -it mongo1 mongosh --eval "rs.status()"

rs.isMaster().primary

db.cliente.insertOne({codigo:1, nome: "Giovanna"});

db.cliente.insertOne({codigo:2, nome: "Arionalda"});

db.cliente.insertOne({codigo:3, nome: "Gabriel"});

db.cliente.insertOne({codigo:4, nome: "Fábio"});

db.cliente.find()

docker stop mongo1

docker exec -it mongo2 mongosh --eval "rs.status()"