*** Database Replication ***

1. Create a Docker network
docker network create mongo-cluster

```
PS C:\Users\razva> docker network create mongo-cluster
4cd4fa43b8c83d4bf54baf045ab214ec634e5f23834a61db829128a189848a70
PS C:\Users\razva>
```

2. Create three independent MongoDB nodes
docker run -d --name mongo-node1 -p 27800:27017 --net mongo-cluster mongo --replSet replicaset_test
docker run -d --name mongo-node2 -p 27801:27017 --net mongo-cluster mongo --replSet replicaset_test
docker run -d --name mongo-node3 -p 27802:27017 --net mongo-cluster mongo --replSet replicaset_test

```
PS C:\Users\razva> docker run -d --name mongo-node1 -p 27800:27017 --net mongo-cluster mongo --replSet replicaset_test
f575196f3be169f2c1210b8e91bd2a8f5e06cbeea27f599920b7aafa0b40faa5
PS C:\Users\razva> docker run -d --name mongo-node2 -p 27801:27017 --net mongo-cluster mongo --replSet replicaset_test
4de7ff7cd5dd561ac3d7cf88166247e4196d5a547a8db2d8f770257c51fb398a
PS C:\Users\razva> docker run -d --name mongo-node3 -p 27802:27017 --net mongo-cluster mongo --replSet replicaset_test
b6fb863ff03c2afd3e0159b0c3605c1454151d91156ab27472e6fbd8803254a6
PS C:\Users\razva>
```

3. Connect to any node, e.g. node #1
mongosh mongodb://localhost:27800/

```
PS C:\Users\razva> mongosh mongodb://localhost:27800/
Current Mongosh Log ID: 66492edeee9658a02769194b
Connecting to:          mongodb://localhost:27800/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.0
Using MongoDB:          7.0.9
Using Mongosh:          2.0.0
mongosh 2.2.6 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2024-05-18T22:41:47.736+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2024-05-18T22:41:48.294+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2024-05-18T22:41:48.294+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never' in this binary version
   2024-05-18T22:41:48.294+00:00: vm.max_map_count is too low
------

test>
```

4. Initiate the replica then check the replica status
At first, we create a JSON object that contains all members of the replica set

```
> config = {
        "_id" : "replicaset_test",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "mongo-node1:27017"
                },
                {
                        "_id" : 1,
                        "host" : "mongo-node2:27017"
                },
                {
                        "_id" : 2,
                        "host" : "mongo-node3:27017"
                }
        ]
```

}

```
test>  config = {
...      "_id" : "replicaset_test",
...      "members" : [
...              {
...                      "_id" : 0,
...                      "host" : "mongo-node1:27017"
...              },
...              {
...                      "_id" : 1,
...                      "host" : "mongo-node2:27017"
...              },
...              {
...                      "_id" : 2,
...                      "host" : "mongo-node3:27017"
...              }
...      ]
...    }
{
  _id: 'replicaset_test',
  members: [
    { _id: 0, host: 'mongo-node1:27017' },
    { _id: 1, host: 'mongo-node2:27017' },
    { _id: 2, host: 'mongo-node3:27017' }
  ]
}
test>
```

Then we initiate the replica set using the configuration we've just created
> rs.initiate(config)

```
}
test> rs.initiate(config)
{ ok: 1 }
replicaset_test [direct: secondary] test>
```

At the end, check the replica set status and see that one node is PRIMARY and the rest are secondary
> rs.status()

5. Use the following connection string to connect to the MongoDB replica-set. Notice that we specify three hosts instead on one.

mongosh
mongodb://localhost:27800,localhost:27801,localhost:27802/repldb?directConnection=true

6. Let's do an insert into the replica set:

```
> db.actors.insertMany([
                { first_name: "John", last_name: "Travolta" },
                { first_name: "Joe", last_name: "Pesci" },
                { first_name: "Will", last_name: "Smith" },
        ]
)
```

DO IT YOURSELF:
A. Insert some actor names using the "majority" write-concern and a timeout of 2 seconds.

```
db.actors.insertMany(
    [
        { first_name: "Robert", last_name: "Downey Jr." },
        { first_name: "Scarlett", last_name: "Johansson" },
        { first_name: "Chris", last_name: "Hemsworth" }
    ],
    {
        writeConcern: { w: "majority", wtimeout: 2000 }
    }
)
```

NEXT:
We will explore time series collections. Familiarize yourself with the topic at
https://docs.mongodb.com/manual/core/timeseries-collections/