Chapter 2: Replication

Lab - Initiate a Replica Set Locally

Back to the Question

Below is an example of a valid config file for the first node in our replica set:

```
storage:
  dbPath: /var/mongodb/db/1
net:
  bindIp: 192.168.103.100,localhost
  port: 27001
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/mongod1.log
  logAppend: true
processManagement:
  fork: true
operationProfiling:
  slowOpThresholdMs: 50
replication:
  replSetName: m103-repl
```

Once we are connected to this first node, we can initiate our replica set with `rs.initiate()`. Again, this command must be run from the same host as the mongod to use the localhost exception.

We can create our `m103-admin` user with the following commands:

```
rs.initiate()
use admin
db.createUser({
  user: "m103-admin",
  pwd: "m103-pass",
  roles: [
    {role: "root", db: "admin"}
  ]
})
```

After exiting the mongo shell, we can start up the other two mongod processes. These will be the secondary nodes in our replica set.

Now that the cluster has a configured user for authentication, we cannot use the localhost exception anymore. Instead, connect using the mongo shell and specify the `m103-admin` user to authenticate and connect to the cluster. As this user, we can add our other two nodes with the following commands:

COPY

```
rs.add("192.168.103.100:27002")
rs.add("192.168.103.100:27003")
```

We should receive a response that says `{"ok" : 1}` from each of these `rs.add()` commands. Now, running `rs.status()` should give us a `members` list with three healthy nodes.

Proceed to next section