Correct! SEE DETAILED ANSWER

Chapter 2: Replication

Lab - Initiate a Replica Set Locally

Problem:

In this lab you will launch a replica set with three members from within your Vagrant environment. To secure this replica set, you will create a keyfile for your nodes to use when communicating with each other.

For this lab, you must place this keyfile in the `/var/mongodb/pki` directory and change the permissions so only the owner of the file can read it or write to it:

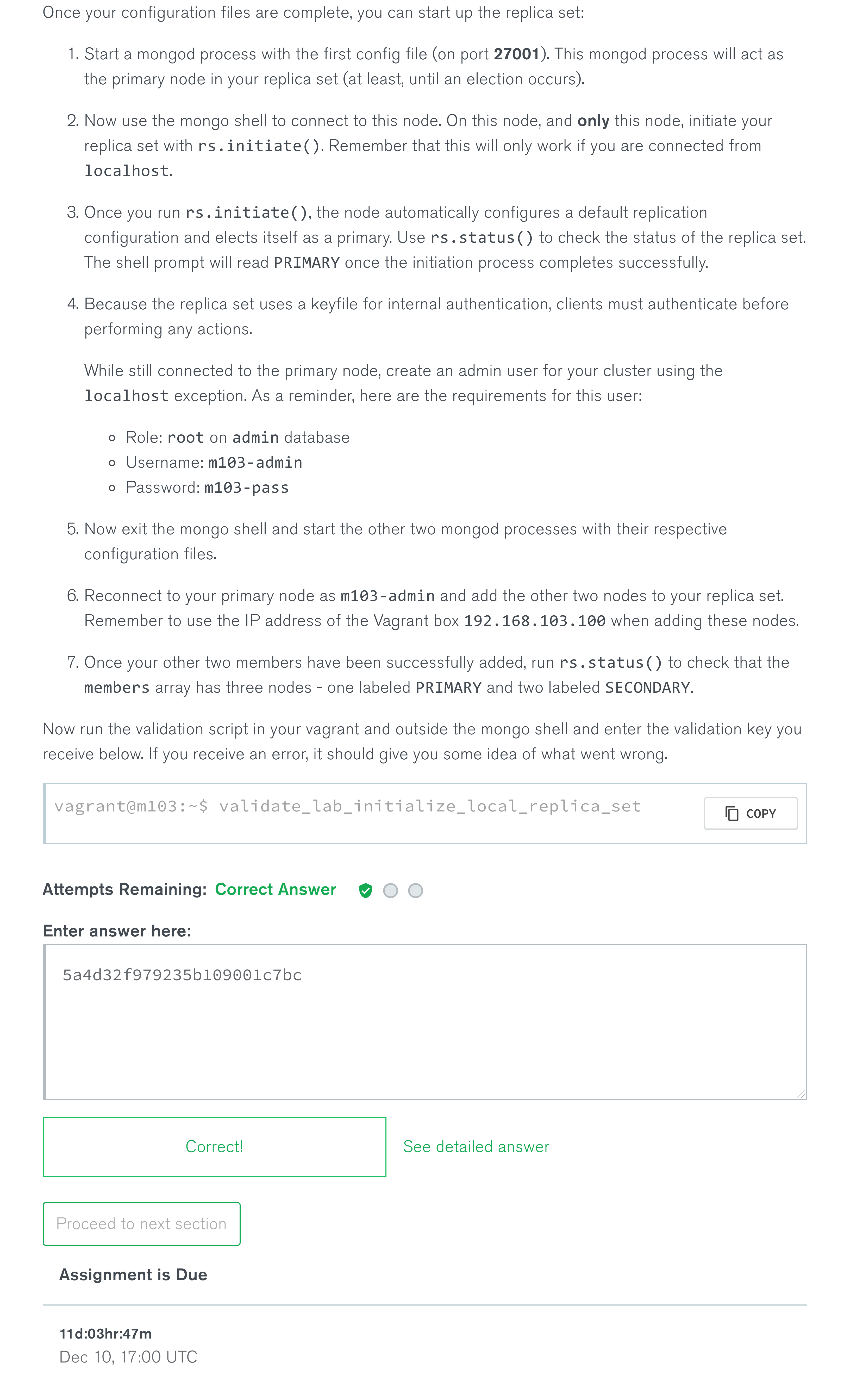
```
sudo mkdir -p /var/mongodb/pki
sudo chown vagrant:vagrant -R /var/mongodb
openssl rand -base64 741 > /var/mongodb/pki/m103-keyfile
chmod 600 /var/mongodb/pki/m103-keyfile
```

Your three mongod processes will each have their own configuration file, and now those config files can reference the keyfile you just made. These config files will be similar to the config file from the previous lab, but with the following adjustments:

type	PRIMARY	SECONDARY	SECONDARY
config filename	mongod-repl-1.conf	mongod-repl-2.conf	mongod-repl-3.conf
port	27001	27002	27003
dbPath	/var/mongodb/db/1	/var/mongodb/db/2	/var/mongodb/db/3
logPath	/var/mongodb/db/mongod1.log	/var/mongodb/db/mongod2.log	/var/mongodb/db/mongod3.log
replSet	m103-repl	m103-repl	m103-repl
keyFile	/var/mongodb/pki/m103-keyfile	/var/mongodb/pki/m103-keyfile	/var/mongodb/pki/m103-keyfile
bindIP	localhost,192.168.103.100	localhost,192.168.103.100	localhost,192.168.103.100

Note that the mongod does **not** automatically create the `dbPath` directory. You will need to create this yourself:

```
mkdir /var/mongodb/db/{1,2,3}
```



Once your configuration files are complete, you can start up the replica set:

1. Start a mongod process with the first config file (on port **27001**). This mongod process will act as the primary node in your replica set (at least, until an election occurs).

2. Now use the mongo shell to connect to this node. On this node, and **only** this node, initiate your replica set with `rs.initiate()`. Remember that this will only work if you are connected from `localhost`.

3. Once you run `rs.initiate()`, the node automatically configures a default replication configuration and elects itself as a primary. Use `rs.status()` to check the status of the replica set. The shell prompt will read `PRIMARY` once the initiation process completes successfully.

4. Because the replica set uses a keyfile for internal authentication, clients must authenticate before performing any actions.

   While still connected to the primary node, create an admin user for your cluster using the `localhost` exception. As a reminder, here are the requirements for this user:

   - Role: `root` on `admin` database
   - Username: `m103-admin`
   - Password: `m103-pass`

5. Now exit the mongo shell and start the other two mongod processes with their respective configuration files.

6. Reconnect to your primary node as `m103-admin` and add the other two nodes to your replica set. Remember to use the IP address of the Vagrant box `192.168.103.100` when adding these nodes.

7. Once your other two members have been successfully added, run `rs.status()` to check that the `members` array has three nodes - one labeled `PRIMARY` and two labeled `SECONDARY`.

Now run the validation script in your vagrant and outside the mongo shell and enter the validation key you receive below. If you receive an error, it should give you some idea of what went wrong.

```
vagrant@m103:~$ validate_lab_initialize_local_replica_set
```

Attempts Remaining: **Correct Answer** ✓ ○ ○

Enter answer here:

```
5a4d32f979235b109001c7bc
```

Assignment is Due

11d:03hr:47m
Dec 10, 17:00 UTC