



Correct! SEE DETAILED ANSWER

×

Chapter 3: Sharding

Lab - Configure a Sharded Cluster

Problem:

In this lab, you will turn your replica set into one shard in a sharded cluster. To begin your sharded cluster configuration in MongoDB, you will need to do the following:

- 1. Bring up config servers
- 2. Bring up mongos
- 3. Enable sharding on `m103-repl`
- 4. Add `m103-repl` as the primary shard in the cluster

1. Bring up the config server replica set (CSRS)

Config servers store all of the metadata for a sharded cluster, making them a necessary part of any sharded cluster. In this lab you will bring up a replica set of **three** config servers (`m103-csrs`) to store this metadata.

Here are the requirements for all three config servers:

Type	Primary	Secondary	Secondary
Port	26001	26002	26003
DBPath	/var/mongodb/db/csrs1	/var/mongodb/db/csrs2	/var/mongodb/db/csrs3
LogPath	/var/mongodb/db/csrs1/mongod.log	/var/mongodb/db/csrs2/mongod.log	/var/mongodb/db/csrs3/mongod.log
replSetName	m103-csrs	m103-csrs	m103-csrs
clusterRole	configsvr	configsvr	configsvr
keyFile	/var/mongodb/pki/m103-keyfile	/var/mongodb/pki/m103-keyfile	/var/mongodb/pki/m103-keyfile

Here is the config file for the **primary** config server:

 COPY

```
sharding:
  clusterRole: configsvr
replication:
  replSetName: m103-csrs
security:
  keyFile: /var/mongodb/pki/m103-keyfile
net:
  bindIp: localhost,192.168.103.100
  port: 26001
systemLog:
  destination: file
  path: /var/mongodb/db/csrs1/mongod.log
  logAppend: true
processManagement:
  fork: true
storage:
  dbPath: /var/mongodb/db/csrs1
```

You may use this config file for your primary config server, but you will need to make two more to complete the replica set.

Notice that the CSRS has the same **keyFile** as the data-bearing replica set **m103-repl1**. Because **m103-repl1** uses internal keyfile authentication, all other mongod and mongos processes in your cluster must use internal keyfile authentication with the same keyfile.

When initializing **m103-csrs**, remember that keyfile authentication implies client authentication. This means that while no users are configured, the CSRS will only allow connections through the **localhost**.

As a reminder, here are the login credentials of the admin user from previous labs:

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

2. Bring up the mongos

Once the CSRS is running, you can start up the mongos process. Here is the config file for the mongos:

 COPY

```
sharding:
  configDB: m103-csrs/192.168.103.100:26001
security:
```

```
keyFile: /var/mongodb/pki/m103-keyfile
net:
  bindIp: localhost,192.168.103.100
  port: 26000
systemLog:
  destination: file
  path: /var/mongodb/db/mongos.log
  logAppend: true
processManagement:
  fork: true
```

If your CSRS already has the **m103-admin** user when mongos is started, mongos will inherit that user. You will be able to authenticate to mongos immediately as **m103-admin**.

3. Reconfigure m103-repl

To enable **m103-repl** to be a shard, you must reconfigure the nodes in your replica set with the following lines added to each of their config files:

```
sharding:
  clusterRole: shardsvr
storage:
  wiredTiger:
    engineConfig:
      cacheSizeGB: .1
```

 COPY

The **clusterRole: shardsvr** section tells mongod that the node can be used in a sharded cluster.

The **cacheSizeGB: .1** section restricts the memory usage of each running mongod. Note that this is **not good practice**. However, in order to run a sharded cluster inside a virtual machine with only 2GB of memory, certain adjustments must be made.

All three nodes of the **m103-repl** replica set will need to be restarted with sharding enabled, but given that this is a replica set, you can do this operation without any downtime. This replica set will become the **primary shard** in your sharded cluster.

4. Add m103-repl as the first shard

Once **m103-repl** has sharding enabled, you can add it as the primary shard with:

```
sh.addShard("m103-repl/192.168.103.100:27001")
```

 COPY

Check the output of **sh.status()** to make sure it's included as a shard.

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_first_sharded_cluster
```

 COPY

Attempts Remaining: **Correct Answer**   

Enter answer here:

5a57de1cb1575291ce6e560a

Correct!

[See detailed answer](#)

[Proceed to next section](#)

Assignment is Due

18d:02hr:39m
Dec 17, 17:00 UTC

Your Grade

PASS/FAIL
Submitted