**Lecture Instructions**

Configuration file for first config server `csrs_1.conf`:

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.log
  logAppend: true
processManagement:
  fork: true
storage:
  dbPath: /var/mongodb/db/csrs1
```

`csrs_2.conf`:

COPY

```
sharding:
  clusterRole: configsvr
replication:
  replSetName: m103-csrs
security:
  keyFile: /var/mongodb/pki/m103-keyfile
net:
  bindIp: localhost,192.168.103.100
  port: 26002
systemLog:
  destination: file
  path: /var/mongodb/db/csrs2.log
  logAppend: true
processManagement:
  fork: true
storage:
  dbPath: /var/mongodb/db/csrs2
```

`csrs_3.conf`:

COPY

```
sharding:
  clusterRole: configsvr
replication:
  replSetName: m103-csrs
security:
  keyFile: /var/mongodb/pki/m103-keyfile
net:
  bindIp: localhost,192.168.103.100
  port: 26003
systemLog:
  destination: file
  path: /var/mongodb/db/csrs3.log
  logAppend: true
processManagement:
  fork: true
storage:
  dbPath: /var/mongodb/db/csrs3
```

Starting the three config servers:

COPY

```
mongod -f csrs_1.conf
mongod -f csrs_2.conf
mongod -f csrs_3.conf
```

Connect to one of the config servers:

```
mongo --port 26001
```

COPY

Initiating the CSRS:

```
rs.initiate()
```

COPY

Creating super user on CSRS:

COPY

```
use admin
db.createUser({
  user: "m103-admin"
```

```
  user: "m103-admin",
  pwd: "m103-pass",
  roles: [
    {role: "root", db: "admin"}
  ]
})
```

Authenticating as the super user:

```
db.auth("m103-admin", "m103-pass")
```

COPY

Add the second and third node to the CSRS:

COPY

```
rs.add("192.168.103.100:26002")
rs.add("192.168.103.100:26003")
```

Mongos config (`mongos.conf`):

COPY

```
sharding:
  configDB: m103-
csrs/192.168.103.100:26001,192.168.103.100:26002,192.168.103.100:26003
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
```

Connect to mongos:

```
vagrant@m103:~$ mongo --port 26000 --username m103-admin --
```

COPY

Check sharding status:

```
MongoDB Enterprise mongos> sh.status()
```

COPY

Updated configuration for `node1.conf`:

COPY

```
sharding:
  clusterRole: shardsvr
storage:
  dbPath: /var/mongodb/db/node1
  wiredTiger:
    engineConfig:
      cacheSizeGB: .1
net:
  bindIp: 192.168.103.100,localhost
  port: 27011
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/node1/mongod.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: m103-repl
```

Updated configuration for `node2.conf`:

COPY

```
sharding:
  clusterRole: shardsvr
storage:
  dbPath: /var/mongodb/db/node2
  wiredTiger:
    engineConfig:
      cacheSizeGB: .1
net:
  bindIp: 192.168.103.100,localhost
  port: 27012
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
```

```
  path: /var/mongodb/db/node2/mongod.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: m103-repl
```

Updated configuration for `node3.conf`:

COPY

```
sharding:
  clusterRole: shardsvr
storage:
  dbPath: /var/mongodb/db/node3
  wiredTiger:
    engineConfig:
      cacheSizeGB: .1
net:
  bindIp: 192.168.103.100,localhost
  port: 27013
security:
  keyFile: /var/mongodb/pki/m103-keyfile
systemLog:
  destination: file
  path: /var/mongodb/db/node3/mongod.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: m103-repl
```

Connecting directly to secondary node (note that if an election has taken place in your replica set, the specified node may have become primary):

```
mongo --port 27012 -u "m103-admin" -p "m103-pass" --authent
```

COPY

Shutting down node:

COPY

```
use admin
db.shutdownServer()
```

Restarting node with new configuration:

```
mongod -f node2.conf
```

COPY

Stepping down current primary:

```
rs.stepDown()
```

COPY

Adding new shard to cluster from `mongos`:

```
sh.addShard("m103-repl/192.168.103.100:27012")
```