



## Lecture Instructions

node4.conf:

 COPY


```
storage:
  dbPath: /var/mongodb/db/node4
net:
  bindIp: 192.168.103.100,localhost
  port: 27014
systemLog:
  destination: file
  path: /var/mongodb/db/node4/mongod.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: m103-example
```

arbiter.conf:

 COPY


```
storage:
  dbPath: /var/mongodb/db/arbiter
net:
  bindIp: 192.168.103.100,localhost
  port: 28000
systemLog:
  destination: file
  path: /var/mongodb/db/arbiter/mongod.log
  logAppend: true
processManagement:
  fork: true
replication:
  replSetName: m103-example
```

Starting up mongod processes for our fourth node and arbiter:

 COPY

```
mongod -f node4.conf
mongod -f arbiter.conf
```

From the Mongo shell of the replica set, adding the new secondary and the new arbiter:

 COPY

```
rs.add("m103.mongodb.university:27014")
rs.addArb("m103.mongodb.university:28000")
```

Checking replica set makeup after adding two new nodes:

 COPY

```
rs.isMaster()
```

Removing the arbiter from our replica set:

```
rs.remove("m103.mongodb.university:28000")
```

 COPY

Assigning the current configuration to a shell variable we can edit, in order to reconfigure the replica set:

```
cfg = rs.conf()
```

 COPY

Editing our new variable `cfg` to change topology - specifically, by modifying `cfg.members`:

```
cfg.members[3].votes = 0
cfg.members[3].hidden = true
cfg.members[3].priority = 0
```

 COPY

Updating our replica set to use the new configuration `cfg`:

```
rs.reconfig(cfg)
```