

Commands to be used for replication in a single machine with multiple instances of MongoDB running

1. Set permission to the directory in which data directories for replication set is being created.

srv is the folder in which the data directories are created. Change the permission using chmod

Terminal 1

```
sudo chmod -R 777 /srv
```

2. Create directories called 'test0', 'test1', 'test2' that will contain the instances databases files.

Create the directories using mkdir

Terminal 1

```
mkdir -p /srv/mongodb/test0 /srv/mongodb/test1 /srv/mongodb/test2
```

or

```
mkdir -p /srv/mongodb/test0
```

```
mkdir -p /srv/mongodb/test1
```

```
mkdir -p /srv/mongodb/test2
```

3. Open another two terminals to start the mongod server.

Use 3 different port numbers.

27017 is the default port.

Create mongod instances using the mongod command

Terminal 1

```
mongod --port 27040 --dbpath /srv/mongodb/test0 --replSet test --smallfiles  
--oplogSize 128
```

Terminal 2

```
mongod --port 27041 --dbpath /srv/mongodb/test1 --replSet test --smallfiles  
--oplogSize 128
```

Terminal 3

```
mongod --port 27042 --dbpath /srv/mongodb/test2 --replSet test --smallfiles  
--oplogSize 128
```

4. Open another 3 terminals to start the mongo shell.

Use mongo command

Terminal 4

```
mongo --port 27040
```

Terminal 5

```
mongo --port 27041
```

Terminal 6

```
mongo --port 27042
```

5. Make any one terminal as primary and add the remaining terminals as secondary. Use initiate() and add()

First check the hostname of the system by using the command "hostname" in terminal

Terminal 4

```
rs.initiate("test")
```

```
rs.add("Student-mca:27041") //rs.add("hostname:portnumber")
```

```
rs.add("Student-mca:27042")
```

Note: Do not give space between hostname and colon and portnumber. This will give an error.

At this point the command prompt will have

```
test:PRIMARY>
```

6. Make the other two terminals as secondary by adding the function setSlaveOk()

Terminal 5

```
db.getMongo().setSlaveOk()
```

Terminal 6

```
db.getMongo().setSlaveOk()
```

At the end of this command both the terminals will have a command prompt that looks like test:SECONDARY>

Test Replica is ready – primary is at port 27040, secondary at port 27041, 27042

7. Create the database, collection and documents in the primary node and view the same in the secondary nodes.

Note: Remember to use the same database in both the secondary nodes.

Note: Do not close any terminal till the end.

8. To close the nodes shutdown the server.
use admin
db.shutdownServer()

Commands per terminal

Terminal 1

1. sudo chmod -R 777 /srv
2. mkdir -p /srv/mongodb/test0 /srv/mongodb/test1 /srv/mongodb/test2
3. mongod --port 27040 --dbpath /srv/mongodb/test0 --replSet test --smallfiles --oplogSize 128

Terminal 2

```
mongod --port 27041 --dbpath /srv/mongodb/test1 --replSet test --smallfiles --oplogSize 128
```

Terminal 3

```
mongod --port 27042 --dbpath /srv/mongodb/test2 --replSet test --smallfiles --oplogSize 128
```

Terminal 4

```
student@Student-mca:~$ mongo --port 27040
MongoDB shell version: 3.2.18
connecting to: 127.0.0.1:27040/test
> rs.initiate("test")
{
  "info2" : "no configuration specified. Using a default configuration for the set",
  "me" : "Student-mca:27040",
  "ok" : 1
}
test:SECONDARY> rs.add("Student-mca:27041")
{ "ok" : 1 }
test:PRIMARY> rs.add("Student-mca:27042")
{ "ok" : 1 }
test:PRIMARY> use mca
switched to db mca
test:PRIMARY> db.createCollection("User")
{ "ok" : 1 }
test:PRIMARY> db.User.insert({_id:1, name:"Lekha"})
WriteResult({ "nInserted" : 1 })
```

```
test:PRIMARY> db.User.insert({_id:2, name:"Isha"})
WriteResult({ "nInserted" : 1 })
test:PRIMARY> db.User.insert({_id:3, name: "Varshini", age: 25}
... )
WriteResult({ "nInserted" : 1 })

test:PRIMARY> db.User.deleteOne({_id:1} )
{ "acknowledged" : true, "deletedCount" : 1 }
test:PRIMARY>

test:PRIMARY> use admin
switched to db admin
2018-03-20T11:20:22.610+0530 I NETWORK [thread1] trying reconnect to
127.0.0.1:27040 (127.0.0.1) failed
2018-03-20T11:20:22.610+0530 I NETWORK [thread1] reconnect 127.0.0.1:27040
(127.0.0.1) ok
test:SECONDARY> db.shutdownServer()
server should be down...
2018-03-20T11:20:36.897+0530 I NETWORK [thread1] trying reconnect to
127.0.0.1:27040 (127.0.0.1) failed
2018-03-20T11:20:36.897+0530 W NETWORK [thread1] Failed to connect to
127.0.0.1:27040, in(checking socket for error after poll), reason: errno:111 Connection
refused
2018-03-20T11:20:36.897+0530 I NETWORK [thread1] reconnect 127.0.0.1:27040
(127.0.0.1) failed failed
At this time Terminal 1 will switch to ordinary prompt
```

Terminal 5

```
student@Student-mca:~$ mongo --port 27041
MongoDB shell version: 3.2.18
connecting to: 127.0.0.1:27041/test
> db.getMongo().setSlaveOk()
test:SECONDARY>
test:SECONDARY> use mca
switched to db mca
test:SECONDARY> db.User.find()
{ "_id" : 1, "name" : "Lekha" }
{ "_id" : 2, "name" : "Isha" }
test:SECONDARY>
After new insertion in primary
test:SECONDARY> db.User.find()
{ "_id" : 1, "name" : "Lekha" }
{ "_id" : 2, "name" : "Isha" }
{ "_id" : 3, "name" : "Varshini", "age" : 25 }

test:SECONDARY> use admin
switched to db admin
test:SECONDARY> db.shutdownServer()
server should be down...
2018-03-20T11:19:38.365+0530 I NETWORK [thread1] trying reconnect to
127.0.0.1:27041 (127.0.0.1) failed
2018-03-20T11:19:38.366+0530 W NETWORK [thread1] Failed to connect to
127.0.0.1:27041, in(checking socket for error after poll), reason: errno:111 Connection
refused
2018-03-20T11:19:38.366+0530 I NETWORK [thread1] reconnect 127.0.0.1:27041
(127.0.0.1) failed failed
At this time Terminal 2 will switch to ordinary prompt
```

Terminal 6

```
student@Student-mca:~$ mongo --port 27042
```

```
MongoDB shell version: 3.2.18
```

```
connecting to: 127.0.0.1:27042/test
```

```
>db.getMongo().setSlaveOk()
```

```
test:SECONDARY>
```

```
test:SECONDARY> db.User.find()
```

```
test:SECONDARY> use mca
```

```
switched to db mca
```

```
test:SECONDARY> db.User.find()
```

```
{ "_id" : 1, "name" : "Lekha" }
```

```
{ "_id" : 2, "name" : "Isha" }
```

```
test:SECONDARY>
```

After deletion in Terminal 4

```
test:SECONDARY> db.User.find()
```

```
{ "_id" : 2, "name" : "Isha" }
```

```
{ "_id" : 3, "name" : "Varshini", "age" : 25 }
```

```
test:SECONDARY> use admin
```

```
switched to db admin
```

```
test:SECONDARY> db.shutdownServer()
```

```
server should be down...
```

```
2018-03-20T11:19:18.524+0530 I NETWORK [thread1] trying reconnect to  
127.0.0.1:27042 (127.0.0.1) failed
```

```
2018-03-20T11:19:18.524+0530 W NETWORK [thread1] Failed to connect to  
127.0.0.1:27042, in(checking socket for error after poll), reason: errno:111 Connection  
refused
```

```
2018-03-20T11:19:18.524+0530 I NETWORK [thread1] reconnect 127.0.0.1:27042  
(127.0.0.1) failed failed
```

```
At this time Terminal 3 will switch to ordinary prompt
```