NIELIT Gorakhpur

Course name: A level **SUBJECT: DATABASE TECHNOLOGIES**

Topic: MongoDB **DATE: 28/5/2020**

Replication cont'd

Managing Configuration File in MongoDB

Before we start creating the Replica Set in MongoDB, it is desired to create separate configuration file for each replica set i.e. for each new port. These replicas will be treated as separate instance of MongoDB Server.

These server instances can be on the different machines for regular use and it is suggested that the replication servers must be kept apart also for safety and to meet disaster recovery. However, for demo purposes we may create multiple instances of the MongoDB server on the same system also.

For each case, we have to at least specify the ReplicaSet Name, the server Port number and Bind IP (as applicable), database and logfile in each of the configuration file.

The configuration file of MongoDB resides in the Bin folder of MongoDB server. If we have installed the MongoDB on a Windows Machine, the configuration file "mongod.cfg" will be available in c:\program files\mongodb\server\4.2\bin directory. This file can be edited in Notepad++ easily.

Suppose, we want to create Three (3) Replica's, then we have to first create directories for database and log files for each of the Secondary(replicated) servers, in this case three.

1. Create directories c:\data1, c:\data2, c:\data3 for database and c:\data1\log, c:\data3\log, c:\data3\log for log files.

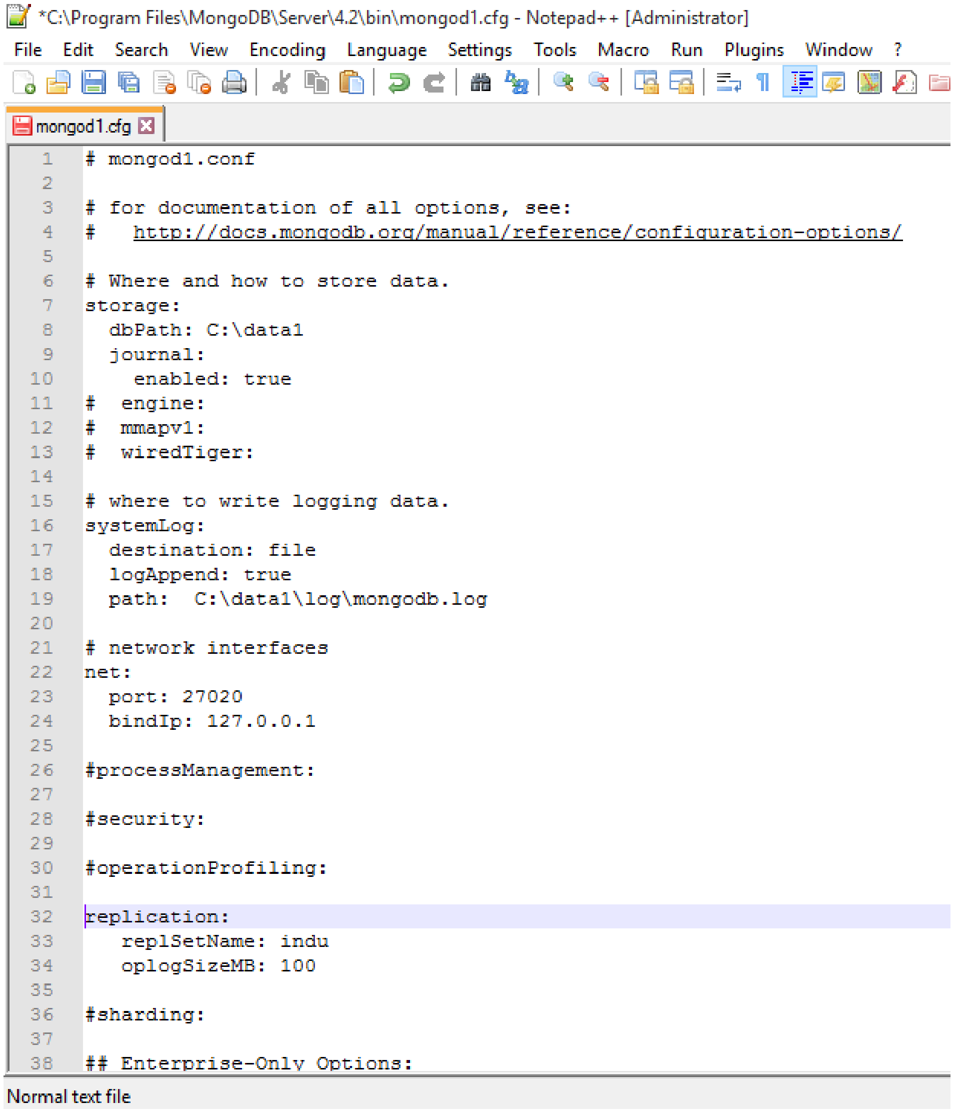
2. Copy configurations file **mongod.cfg** to mongod1.cfg, mongod2.cfg, mongod2.cfg

3. Edit all Configuration files including mongod.cfg and specify the ReplicaSet Name, the server Port number and Bind IP. Lets edit the configuration files: (here mongod1.conf)

```
# mongod1.conf
storage:
  dbPath: C:\data1
systemLog:
  destination: file
  logAppend: true
  path:  C:\data1\log\mongodb.log
# network interfaces
```

```
net:
  port: 27020
  bindIp: 127.0.0.1
 replication:
   replSetName: indu
   oplogSizeMB: 100
```

- *in case of different server machines, Bind IP has to the IP of that server machine.*
- *oplogSizeMB is the Maximum size of the log file in MB*
- *dbPath: C:\data1, is the path of the database*
- *path: C:\data1\log\mongodb.log, is the path of the logfile.*


*C:\Program Files\MongoDB\Server\4.2\bin\mongod1.cfg - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
mongod1.cfg
Normal text file

```
# mongod1.conf

# for documentation of all options, see:
#   http://docs.mongodb.org/manual/reference/configuration-options/

# Where and how to store data.
storage:
  dbPath: C:\data1
  journal:
    enabled: true
#  engine:
#  mmapv1:
#  wiredTiger:

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path:  C:\data1\log\mongodb.log

# network interfaces
net:
  port: 27020
  bindIp: 127.0.0.1

#processManagement:

#security:

#operationProfiling:

replication:
   replSetName: indu
   oplogSizeMB: 100

#sharding:

## Enterprise-Only Options:
```

We have to edit all the configuration files. In our case, we have 3 new configuration files, one for each replica.

- mongod1.cfg
- mongod2.cfg
- mongod3.cfg

Setting up a Replica Set in MongoDB

We may convert standalone MongoDB instance to a replica set. To convert to replica set, following are the steps −

- Shutdown already running MongoDB server.
- Start the MongoDB server by specifying -- replSet option.

Syntax

mongod --port "PORT" --dbpath "YOUR_DB_DATA_PATH"   --logpath "YOUR_log_File_PATH" --replSet "NAME_REPLICA_SET"

Where,
Port – is the Port Number
YOUR_DB_DATA_PATH – Is the complete path of the Database
Name_ REPLICA_SET – is the Name of the Replica Set instance
YOUR_log_File_PATH – is the file name and path of Log file.

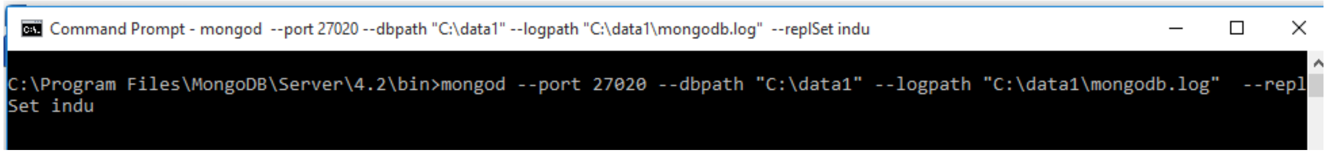
Example

mongod --port 27020 --dbpath "C:\data1" --logpath "C:\data1\log\mongodb.log" --replSet indu

mongod --port 27021 --dbpath "C:\data2" --logpath "C:\data2\log\mongodb.log" --replSet indu

mongod --port 27022 --dbpath "C:\data3" --logpath "C:\data3\log\mongodb.log" --replSet indu

These command will be run on the windows prompt, at the bin directory of the MongoDB.


Command Prompt - mongod --port 27020 --dbpath "C:\data1" --logpath "C:\data1\mongodb.log" --replSet indu

```
C:\Program Files\MongoDB\Server\4.2\bin>mongod --port 27020 --dbpath "C:\data1" --logpath "C:\data1\mongodb.log"  --repl
Set indu
```

This will convert MongoDB instance to replica set with name “indu” and on port 27020, 27021, 27022.

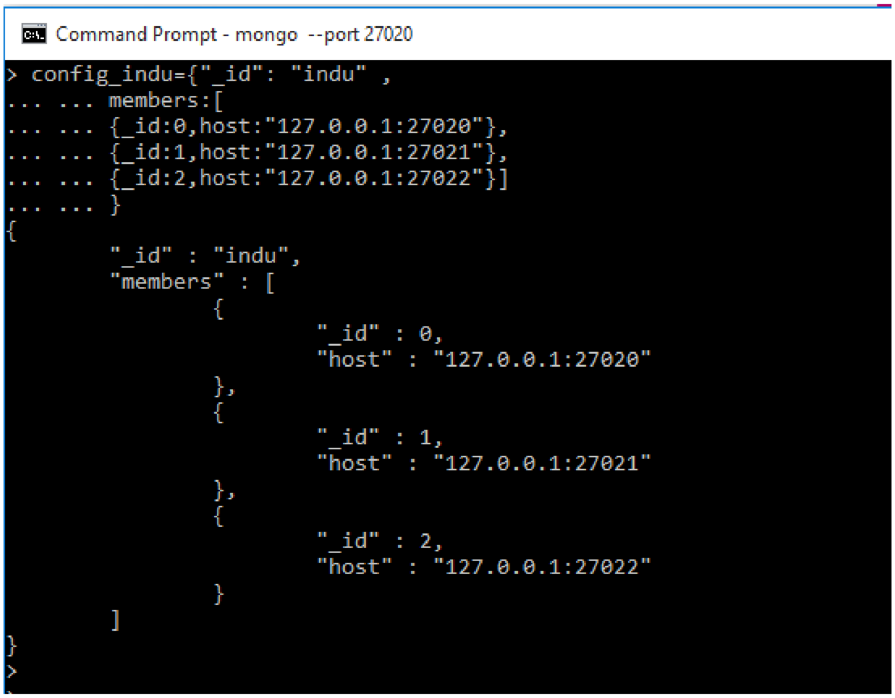
- Now start the command prompt and connect to this mongod instance, from the windows command prompt.

  mongo - -port 27020

- **Setting replication :** Open a mongo shell and initiate a replication by running below configuration

  config_indu={"_id": "indu" , members:[{_id:0,host:"127.0.0.1:27020"},

  {_id:1,host:"127.0.0.1:27021"}, {_id:2,host:"127.0.0.1:27022"}] }

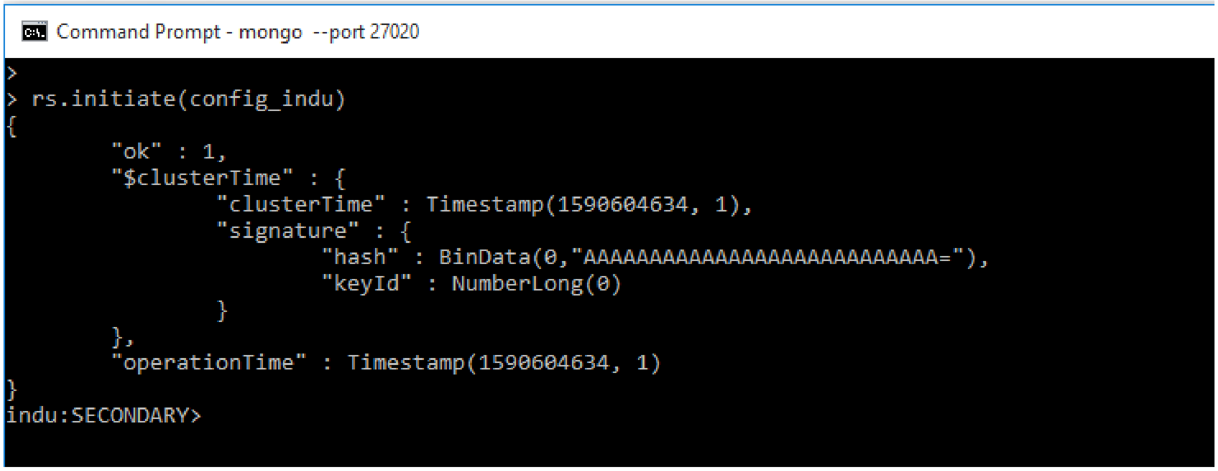


```
> config_indu={"_id": "indu" ,
... ... members:[
... ... {_id:0,host:"127.0.0.1:27020"},
... ... {_id:1,host:"127.0.0.1:27021"},
... ... {_id:2,host:"127.0.0.1:27022"}]
... ... }
{
        "_id" : "indu",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "127.0.0.1:27020"
                },
                {
                        "_id" : 1,
                        "host" : "127.0.0.1:27021"
                },
                {
                        "_id" : 2,
                        "host" : "127.0.0.1:27022"
                }
        ]
}
>
```

- Now, run the command,

  rs.initiate(config_indu)

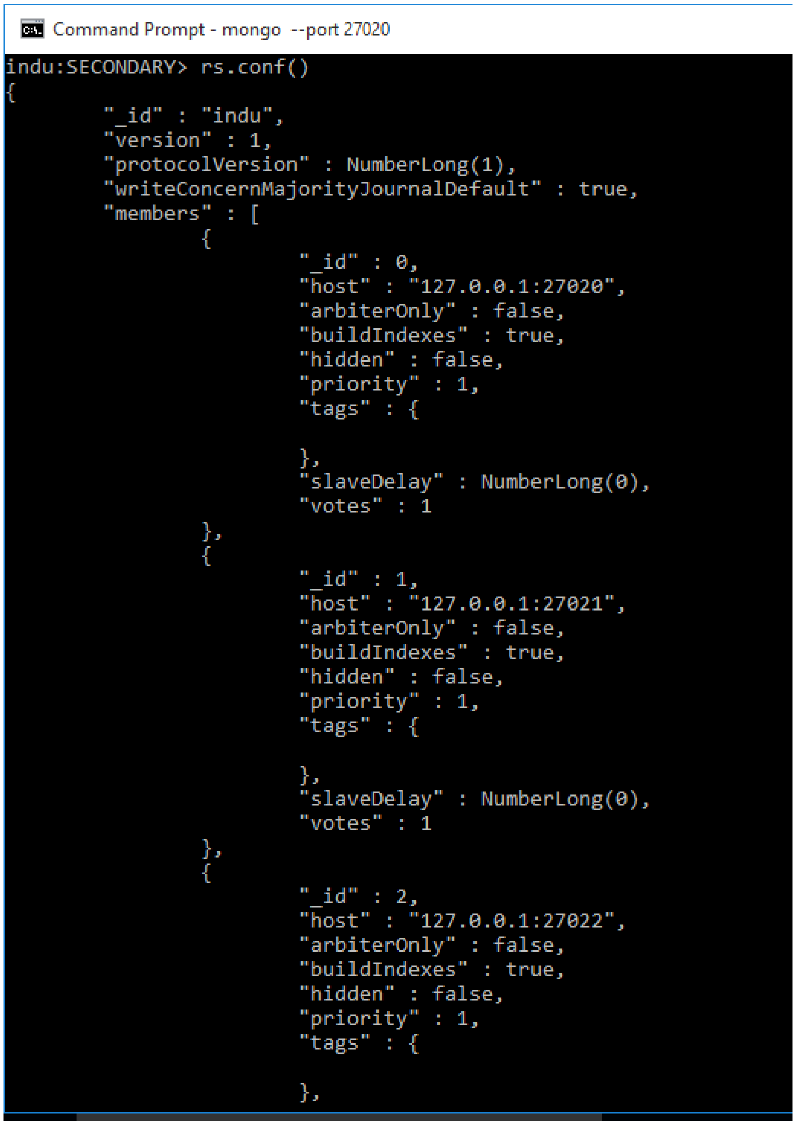
  This will initiate a new replica set.



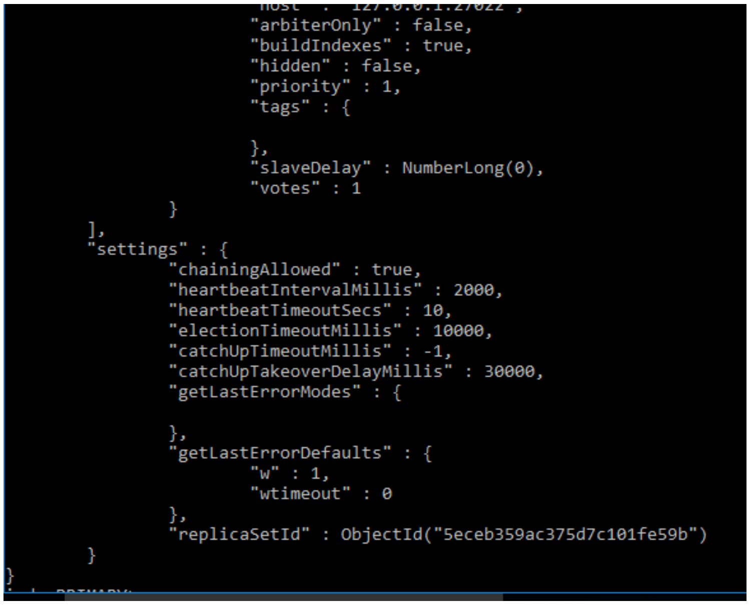
```
>
> rs.initiate(config_indu)
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590604634, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1590604634, 1)
}
indu:SECONDARY>
```

- As we run this command, we can see that Prompt shows as **indu.SECONDARY>**
- To check the replica set configuration, the command is

  rs.conf()



```
indu:SECONDARY> rs.conf()
{
        "_id" : "indu",
        "version" : 1,
        "protocolVersion" : NumberLong(1),
        "writeConcernMajorityJournalDefault" : true,
        "members" : [
                {
                        "_id" : 0,
                        "host" : "127.0.0.1:27020",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 1,
                        "tags" : {

                        },
                        "slaveDelay" : NumberLong(0),
                        "votes" : 1
                },
                {
                        "_id" : 1,
                        "host" : "127.0.0.1:27021",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 1,
                        "tags" : {

                        },
                        "slaveDelay" : NumberLong(0),
                        "votes" : 1
                },
                {
                        "_id" : 2,
                        "host" : "127.0.0.1:27022",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 1,
                        "tags" : {

                        },
```

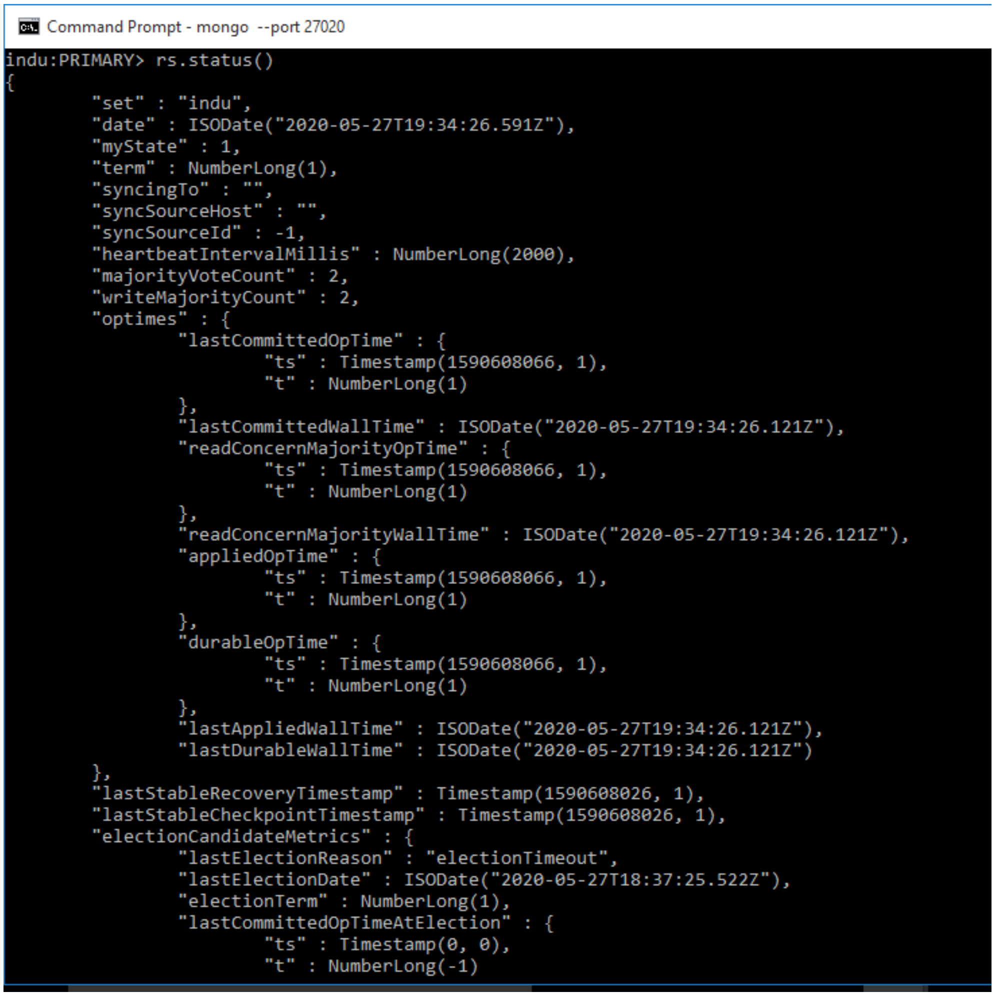


```
                        "host" : "127.0.0.1:27022",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 1,
                        "tags" : {

                        },
                        "slaveDelay" : NumberLong(0),
                        "votes" : 1
                }
        ],
        "settings" : {
                "chainingAllowed" : true,
                "heartbeatIntervalMillis" : 2000,
                "heartbeatTimeoutSecs" : 10,
                "electionTimeoutMillis" : 10000,
                "catchUpTimeoutMillis" : -1,
                "catchUpTakeoverDelayMillis" : 30000,
                "getLastErrorModes" : {

                },
                "getLastErrorDefaults" : {
                        "w" : 1,
                        "wtimeout" : 0
                },
                "replicaSetId" : ObjectId("5eceb359ac375d7c101fe59b")
        }
}
indu:PRIMARY>
```

- Now, we can see that Prompt shows as **indu.PRIMARY>**

- To check the status of replica set, command is

  rs.status()



```
indu:PRIMARY> rs.status()
{
        "set" : "indu",
        "date" : ISODate("2020-05-27T19:34:26.591Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 2,
        "writeMajorityCount" : 2,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1590608066, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2020-05-27T19:34:26.121Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1590608066, 1),
                        "t" : NumberLong(1)
                },
                "readConcernMajorityWallTime" : ISODate("2020-05-27T19:34:26.121Z"),
                "appliedOpTime" : {
                        "ts" : Timestamp(1590608066, 1),
                        "t" : NumberLong(1)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1590608066, 1),
                        "t" : NumberLong(1)
                },
                "lastAppliedWallTime" : ISODate("2020-05-27T19:34:26.121Z"),
                "lastDurableWallTime" : ISODate("2020-05-27T19:34:26.121Z")
        },
        "lastStableRecoveryTimestamp" : Timestamp(1590608026, 1),
        "lastStableCheckpointTimestamp" : Timestamp(1590608026, 1),
        "electionCandidateMetrics" : {
                "lastElectionReason" : "electionTimeout",
                "lastElectionDate" : ISODate("2020-05-27T18:37:25.522Z"),
                "electionTerm" : NumberLong(1),
                "lastCommittedOpTimeAtElection" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
```

Where,

mystate: 1->primary，2->slave for backup

health：1->normal，0->exception(means down)

It shows a lot of other details also.

Add Members to Replica Set

rs.add() commands adds members to the replica set. You can add mongod instance to replica set only if are connected to primary node.

db.isMaster() command is to check whether you are connected to primary or not.

Syntax

>rs.add(HOST_NAME:PORT)

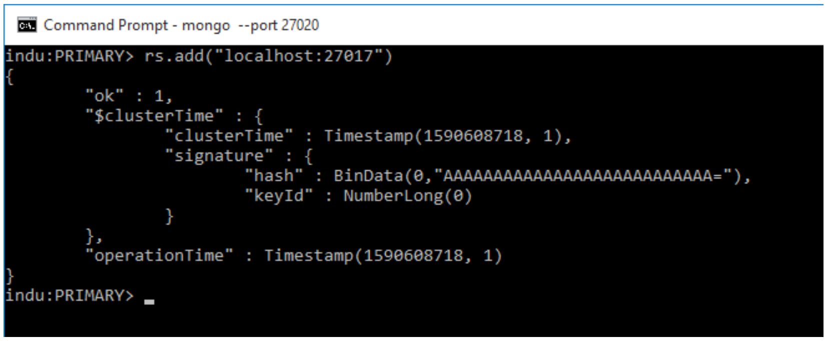
Example

Suppose your mongod instance name (HOST Name) is localhost (or 127.0.0.1) and it is running on port **27017**. To add this instance to replica set, command will be

>rs.add("localhost:27017")

Or

> >rs.add("127.0.0.1:27017")



```
indu:PRIMARY> rs.add("localhost:27017")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590608718, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1590608718, 1)
}
indu:PRIMARY> _
```

Assignment

1. How the configuration file in MongoDB can be created for other instances?

2. Write steps to configure MongoDB replica set.

3. How a new member is added to the Replica Set?