Replication Cont'd

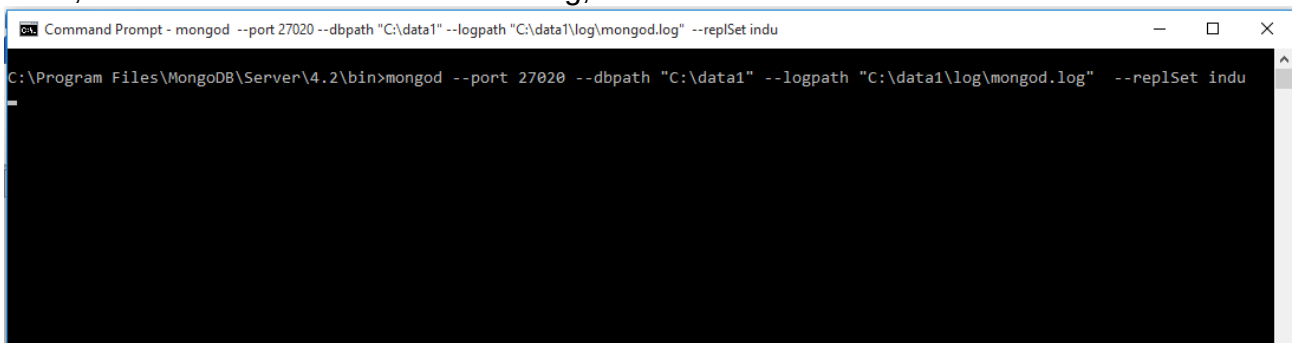
Replication Configuration

During creation and configuring the Replica Set, ensure that all the instances of MongoDB Server must be running. These instances were started(& converted to Replica Set) using **mongod** Command on Windows prompt:

Example

```
mongod --port 27020 --dbpath "C:\data1" --logpath "C:\data1\log\mongod.log"  --replSet indu
mongod --port 27021 --dbpath "C:\data2" --logpath "C:\data2\log\mongod.log"  --replSet indu
mongod --port 27022 --dbpath "C:\data3" --logpath "C:\data3\log\mongod.log"  --replSet indu
```

Here, three instances have been running, like



This command will start a **mongod** instance with the name indu, on port **27020,** all the instances must be running like this.

- To connect to mongod instance for example on port 27020, and start the Mongo command prompt from the windows command prompt (shall be from bin folder where the application file exists), the command will be

  mongo - -port 27020

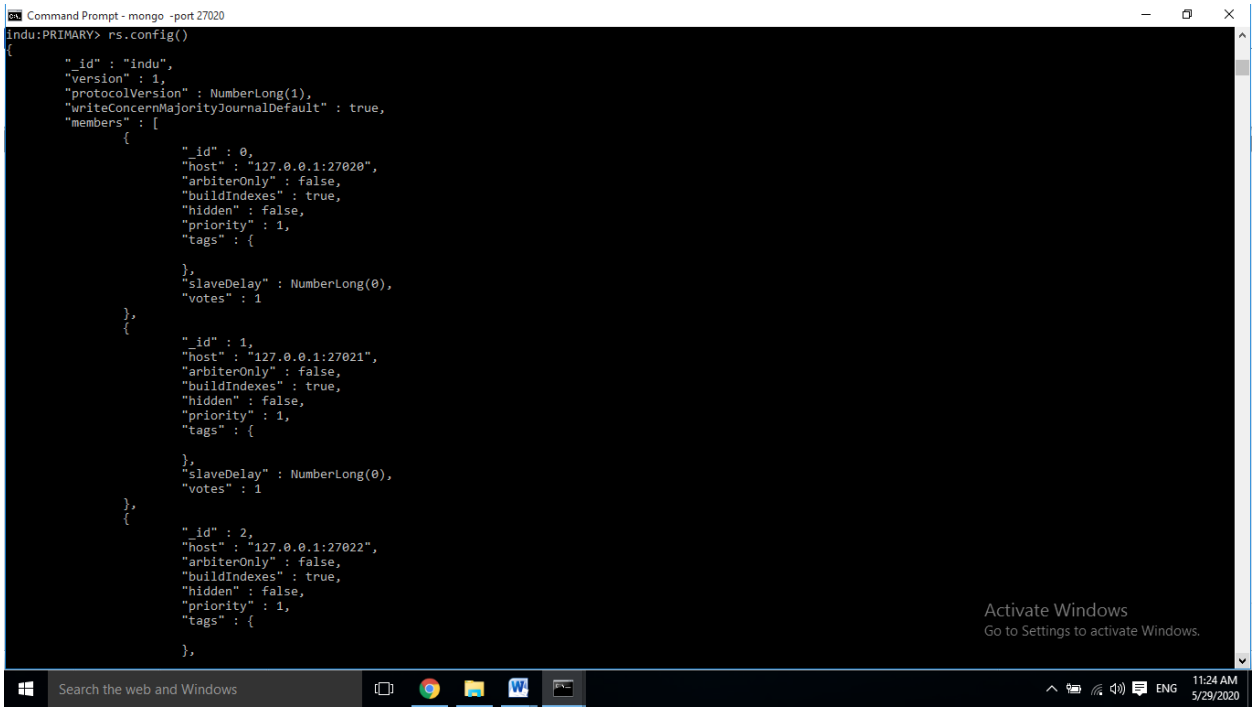
  This will open, Mongo shell. We may open different shells connected to other instances also.

After this, you can initiate various settings and configurations as mentioned in previous session and here also.

Checking the Configuration

rs.conf() method (**rs.config()** may also be used) without any parameter is used to check the present configuration of the application.

> rs.config()



This shows that the application has three members in replica set.

Change Replication Configuration

We can modify the replica set at any time as per requirement and need later. We may add new members to the Replica Set or remove any of the Existing members.

To add a new member into the replica set, we need to run the rs.add() command like

>rs.add("127.0.0.1:27017")

Note: instance of this member (port 27017) has not been started with Replica Set .

Lets add another member,

>rs.add("127.0.0.1:27023")

Note: instance of this member (port 27023) is neither created nor started with Replica Set .

Member can be added to replica set simply by specifying the hostname and port number. But it is recmonded to add members replica set whose instance has been started with a Replica Set. Otherwise it will show error message in rs.status() under lastHeartbeatMessage.

Now if we check the configuration again, after adding the member

```
> rs.config()
{
	"_id" : "indu",
	"version" : 7,
	"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" : {

                },
                "slaveDelay" : NumberLong(0),
                "votes" : 1
        },
        {
                "_id" : 3,
                "host" : "127.0.0.1:27017",
                "arbiterOnly" : false,
                "buildIndexes" : true,
                "hidden" : false,
                "priority" : 1,
                "tags" : {

                },
                "slaveDelay" : NumberLong(0),
                "votes" : 1
        },
        {
                "_id" : 4,
                "host" : "127.0.0.1:27023",
                "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("5ecffc87e7d9c40b106bbd3a")
    }
}
```

We can see that two hosts has been added into the replica set.

Status of the Replica Set using rs.status()

rs.status() command is used to see the status of the replica set.

Syntax

rs.status()

Lets check it now as we have added two hosts.

This will show the message like this:

```
{
        "set" : "indu",
        "date" : ISODate("2020-05-29T06:17:49.792Z"),
        "myState" : 1,
        "term" : NumberLong(6),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 3,
        "writeMajorityCount" : 3,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1590733061, 1),
                        "t" : NumberLong(6)
                },
                "lastCommittedWallTime" : ISODate("2020-05-29T06:17:41.788Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1590733061, 1),
                        "t" : NumberLong(6)
                },
                "readConcernMajorityWallTime" : ISODate("2020-05-29T06:17:41.788Z"),
                "appliedOpTime" : {
                        "ts" : Timestamp(1590733061, 1),
                        "t" : NumberLong(6)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1590733061, 1),
                        "t" : NumberLong(6)
                },
```

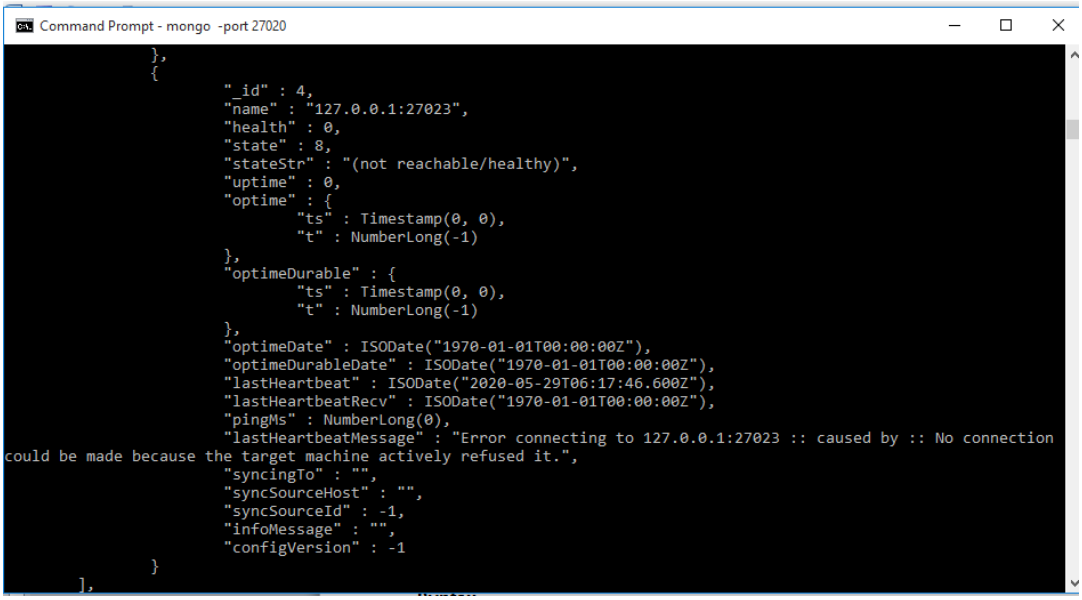
```
        "lastAppliedWallTime" : ISODate("2020-05-29T06:17:41.788Z"),
        "lastDurableWallTime" : ISODate("2020-05-29T06:17:41.788Z")
    },
    "lastStableRecoveryTimestamp" : Timestamp(1590733061, 1),
    "lastStableCheckpointTimestamp" : Timestamp(1590733061, 1),
    "electionCandidateMetrics" : {
        "lastElectionReason" : "electionTimeout",
        "lastElectionDate" : ISODate("2020-05-29T05:48:49.859Z"),
        "electionTerm" : NumberLong(6),
        "lastCommittedOpTimeAtElection" : {
            "ts" : Timestamp(0, 0),
            "t" : NumberLong(-1)
        },
        "lastSeenOpTimeAtElection" : {
            "ts" : Timestamp(1590692225, 1),
            "t" : NumberLong(4)
        },
        "numVotesNeeded" : 2,
        "priorityAtElection" : 1,
        "electionTimeoutMillis" : NumberLong(10000),
        "numCatchUpOps" : NumberLong(0),
        "newTermStartDate" : ISODate("2020-05-29T05:48:51.633Z"),
        "wMajorityWriteAvailabilityDate" : ISODate("2020-05-29T05:48:52.045Z")
    },
    "members" : [
        {
            "_id" : 0,
            "name" : "127.0.0.1:27020",
            "health" : 1,
            "state" : 1,
            "stateStr" : "PRIMARY",
            "uptime" : 1755,
            "optime" : {
                "ts" : Timestamp(1590733061, 1),
                "t" : NumberLong(6)
            },
            "optimeDate" : ISODate("2020-05-29T06:17:41Z"),
            "syncingTo" : "",
            "syncSourceHost" : "",
            "syncSourceId" : -1,
            "infoMessage" : "",
            "electionTime" : Timestamp(1590731330, 1),
            "electionDate" : ISODate("2020-05-29T05:48:50Z"),
            "configVersion" : 3,
            "self" : true,
            "lastHeartbeatMessage" : ""
        },
        {
            "_id" : 1,
            "name" : "127.0.0.1:27021",
            "health" : 1,
            "state" : 2,
            "stateStr" : "SECONDARY",
            "uptime" : 1742,
            "optime" : {
                "ts" : Timestamp(1590733061, 1),
                "t" : NumberLong(6)
            },
            "optimeDurable" : {
```

```
                "ts" : Timestamp(1590733061, 1),
                "t" : NumberLong(6)
        },
        "optimeDate" : ISODate("2020-05-29T06:17:41Z"),
        "optimeDurableDate" : ISODate("2020-05-29T06:17:41Z"),
        "lastHeartbeat" : ISODate("2020-05-29T06:17:49.537Z"),
        "lastHeartbeatRecv" : ISODate("2020-05-29T06:17:49.077Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "127.0.0.1:27020",
        "syncSourceHost" : "127.0.0.1:27020",
        "syncSourceId" : 0,
        "infoMessage" : "",
        "configVersion" : 3
},
{
        "_id" : 2,
        "name" : "127.0.0.1:27022",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 1728,
        "optime" : {
                "ts" : Timestamp(1590733061, 1),
                "t" : NumberLong(6)
        },
        "optimeDurable" : {
                "ts" : Timestamp(1590733061, 1),
                "t" : NumberLong(6)
        },
        "optimeDate" : ISODate("2020-05-29T06:17:41Z"),
        "optimeDurableDate" : ISODate("2020-05-29T06:17:41Z"),
        "lastHeartbeat" : ISODate("2020-05-29T06:17:49.541Z"),
        "lastHeartbeatRecv" : ISODate("2020-05-29T06:17:49.077Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncingTo" : "127.0.0.1:27020",
        "syncSourceHost" : "127.0.0.1:27020",
        "syncSourceId" : 0,
        "infoMessage" : "",
        "configVersion" : 3
},
{
        "_id" : 3,
        "name" : "127.0.0.1:27017",
        "health" : 0,
        "state" : 8,
        "stateStr" : "(not reachable/healthy)",
        "uptime" : 0,
        "optime" : {
                "ts" : Timestamp(0, 0),
                "t" : NumberLong(-1)
        },
        "optimeDurable" : {
                "ts" : Timestamp(0, 0),
                "t" : NumberLong(-1)
        },
        "optimeDate" : ISODate("1970-01-01T00:00:00Z"),
        "optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
```

```
                "lastHeartbeat" : ISODate("2020-05-29T06:17:49.629Z"),
                "lastHeartbeatRecv" : ISODate("1970-01-01T00:00:00Z"),
                "pingMs" : NumberLong(0),
                "lastHeartbeatMessage" : "not running with --replSet",
                "syncingTo" : "",
                "syncSourceHost" : "",
                "syncSourceId" : -1,
                "infoMessage" : "",
                "configVersion" : -1
        },
        {
                "_id" : 4,
                "name" : "127.0.0.1:27023",
                "health" : 0,
                "state" : 8,
                "stateStr" : "(not reachable/healthy)",
                "uptime" : 0,
                "optime" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
                },
                "optimeDurable" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
                },
                "optimeDate" : ISODate("1970-01-01T00:00:00Z"),
                "optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
                "lastHeartbeat" : ISODate("2020-05-29T06:17:46.600Z"),
                "lastHeartbeatRecv" : ISODate("1970-01-01T00:00:00Z"),
                "pingMs" : NumberLong(0),
                "lastHeartbeatMessage" : "Error connecting to 127.0.0.1:27023 :: caused by :: No connection could be made because the target machine actively refused it.",
                "syncingTo" : "",
                "syncSourceHost" : "",
                "syncSourceId" : -1,
                "infoMessage" : "",
                "configVersion" : -1
        }
    ],
    "ok" : 1,
    "$clusterTime" : {
        "clusterTime" : Timestamp(1590733061, 1),
        "signature" : {
                "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                "keyId" : NumberLong(0)
        }
    },
    "operationTime" : Timestamp(1590733061, 1)
}
```


- **See the message for Port 27017 which was not started with replica Set –**
  - **"stateStr" : "(not reachable/healthy)",**
  - **"lastHeartbeatMessage" : "not running with --replSet",**

- **If instance has not started even not created with replica Set, the messages will be like**
    - **stateStr" : "(not reachable/healthy)"**
    - **"lastHeartbeatMessage" : "Error connecting to 127.0.0.1:27023 :: caused by :: No connection could be made because the target machine actively refused it.",**



Remove Members

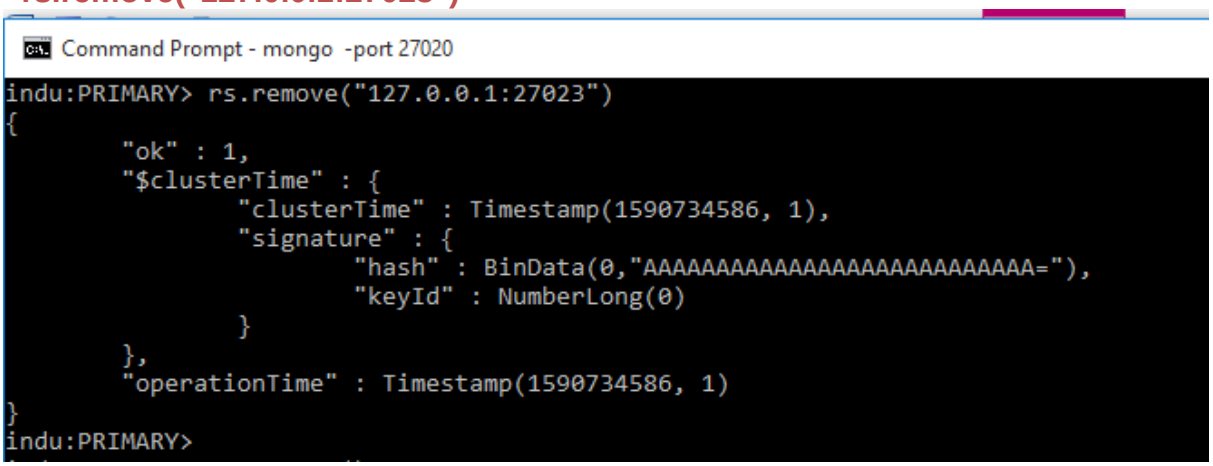
We can remove any member(s) from the existing replica set using the rs.remove() command.

Syntax

>rs.remove("HOST_NAME:PORT")

Example, lets remove the member

>rs.remove("127.0.0.1:27023")



If we run **rs.config()** command again, we will see that the particular port has been removed from the replica set.

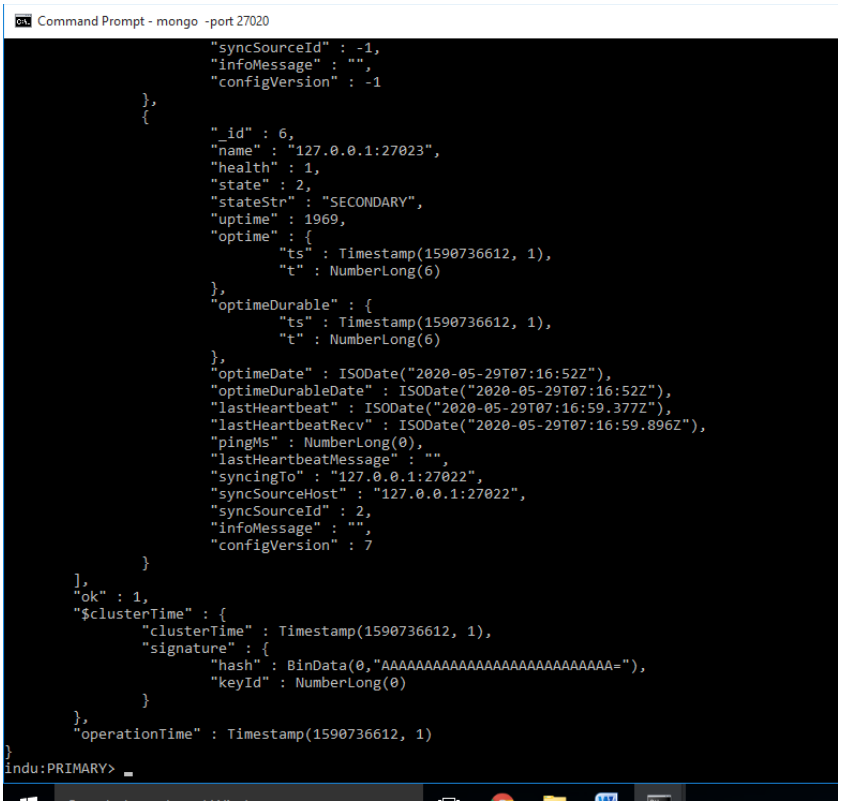
Adding a new Port / instance in replica set: Steps and Cautions

1. First create the folders for the database and log files
2. Copy and Rename the configuration file
3. Edit the configuration file with the db & log Path, Port number and bind IP , Replica name and size
4. Start the MongoDB instance with Replica Set and Port no like

5. Now Add the Host to the replica Set using rs.add () command

   >rs.add("127.0.0.1:27023")

6. If desired, check the replica set configuration using rs.conf()
7. Check the status of Replica Set with rs.status() again after adding the member properly. Now it shows that
   - **"stateStr" : "SECONDARY" -- Means that it is Secondary**
   - **"lastHeartbeatMessage" : "" -- means no error and is active**



Assignment

1. How to add and remove members in Replica Set?
2. Can we add a member without creating the instance? If so, how we can verify it using rs.status() command ?
3. What are the steps to create and add a member to replica set?