

**Replication cont'd****Stepping Down a Member**

The method runs only if the current member is a primary node and produce an error if runs on a non-primary member. `rs.stepDown()` instructs the current Primary node of the replica set to become a secondary which forces an election. After the primary steps down, eligible secondary nodes hold an election for primary.

**Syntax**

**`rs.stepDown(stepDownPeriod, secondaryCatchUpPeriod)`**

where,

**stepDownPeriod-** Specifies the number of seconds to step down the primary, during this time the stepdown member is ineligible for becoming primary. By default it is 60 seconds. It must be greater than the `secondaryCatchUpPeriod`.

**secondaryCatchUpPeriod** – specifies the number of seconds that mongod will wait for an electable secondary to catch up to the primary. This is Optional. The default wait time is 10 seconds.

`rs.stepDown()` method will not immediately step down the primary. If no electable secondary node is up to date with the primary, the primary waits up to `secondaryCatchUpPeriod` for a secondary node to catch up. Once an electable secondary is available, the method steps down the primary. Once stepped down, the original primary becomes a secondary and is ineligible from becoming primary again for the remainder of time specified by `stepDownPeriod`.

```
Command Prompt - mongo --port 27020
indu:PRIMARY>
indu:PRIMARY> rs.stepDown(90,20)
{
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1591082825, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1591082825, 1)
}
indu:SECONDARY> _
```

## Syncing from a member

rs.syncFrom() methods sets the secondary replica set member will sync from the specified member, overriding the default sync target selection logic.

### Syntax

**rs.syncForm("[hostname]:[port]")**

### **Example**

**rs.syncFrom("127.0.0.1:27021")**

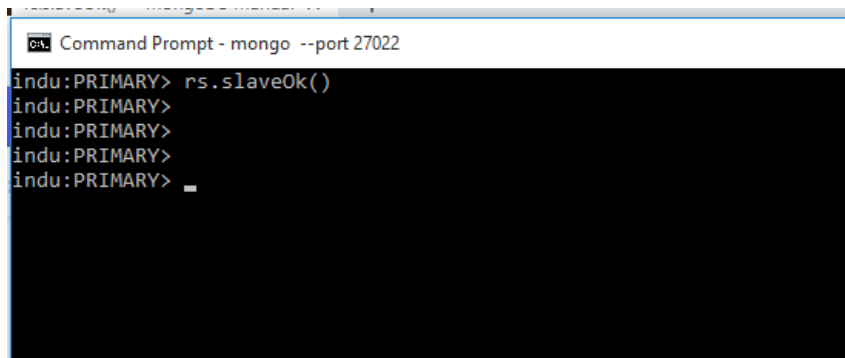
```
Command Prompt - mongo --port 27020
}
indu:SECONDARY>
indu:SECONDARY> rs.syncFrom("127.0.0.1:27021")
{
  "syncFromRequested" : "127.0.0.1:27021",
  "prevSyncTarget" : "127.0.0.1:27021",
  "ok" : 1,
  "$clusterTime" : {
    "clusterTime" : Timestamp(1591082947, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1591082947, 1)
}
indu:SECONDARY>
```

## Read Operation on Secondary Member

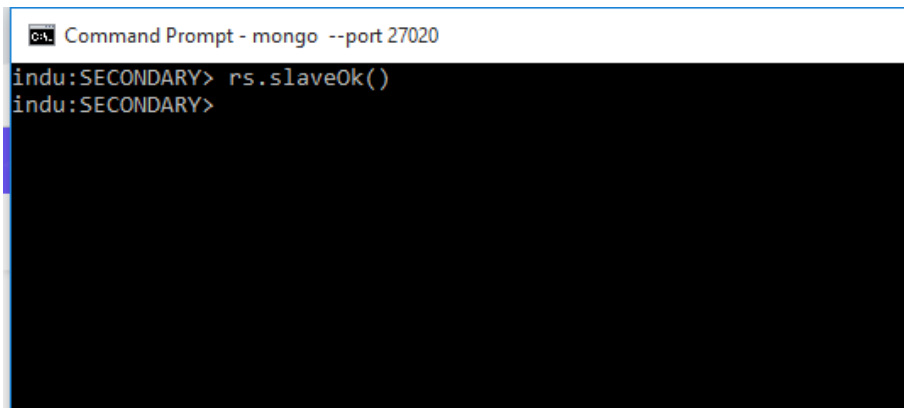
rs.slaveOk() method allows the current connection to allow read operations to run on secondary members. i.e. allow users to run commands in the secondary or arbiter node.

### Syntax

#### rs.slaveOk()



```
Command Prompt - mongo --port 27022
indu:PRIMARY> rs.slaveOk()
indu:PRIMARY>
indu:PRIMARY>
indu:PRIMARY>
indu:PRIMARY> _
```



```
Command Prompt - mongo --port 27020
indu:SECONDARY> rs.slaveOk()
indu:SECONDARY>
```

## Printing the oplog of the replica set

- rs.printReplicationInfo() method prints the report of the replica set member's oplog from the perspective of the Primary Member.

### Syntax

#### rs.printReplicationInfo()

```
C:\> Command Prompt - mongo --port 27022
indu:PRIMARY>
indu:PRIMARY> rs.printReplicationInfo()
configured oplog size: 4715.123436927795MB
log length start to end: 320secs (0.09hrs)
oplog first event time: Tue Jun 02 2020 13:05:11 GMT+0530 (India Standard Time)
oplog last event time: Tue Jun 02 2020 13:10:31 GMT+0530 (India Standard Time)
now: Tue Jun 02 2020 13:10:38 GMT+0530 (India Standard Time)
indu:PRIMARY>
```

```
C:\> Command Prompt - mongo --port 27021
indu:SECONDARY> rs.printReplicationInfo()
configured oplog size: 4730.51523399353MB
log length start to end: 1494secs (0.42hrs)
oplog first event time: Tue Jun 02 2020 12:46:13 GMT+0530 (India Standard Time)
oplog last event time: Tue Jun 02 2020 13:11:07 GMT+0530 (India Standard Time)
now: Tue Jun 02 2020 13:11:08 GMT+0530 (India Standard Time)
indu:PRIMARY> _
```

- rs.printSlaveReplicationInfo() method prints the report of the replica set member's oplog from the perspective of the Secondary Members of the Set.

## Syntax

**rs.printSlaveReplicationInfo()**

## Stopping a Member

db.shutdownServer() method is used to shut down or Stopping a member. It may be a secondary member. This command will run only with admin database.

## Syntax

**db.shutdownServer()**

```
Command Prompt - mongo --port 27020
indu:SECONDARY>
indu:SECONDARY> db.shutdownServer()
shutdown command only works with the admin database; try 'use admin'
indu:SECONDARY> use admin
switched to db admin
indu:SECONDARY> db.shutdownServer()
2020-06-02T13:17:04.383+0530 I NETWORK [js] DBClientConnection failed to receive message from 127.0.0.1:27020 - HostUn
reachable: Connection reset by peer
server should be down...
2020-06-02T13:17:04.441+0530 I NETWORK [js] trying reconnect to 127.0.0.1:27020 failed
2020-06-02T13:17:05.443+0530 I NETWORK [js] reconnect 127.0.0.1:27020 failed failed
> _
```

## **Assignment**

1. How a member can be Step?
2. How to shut down a member?