Setup MongoDB Replica Set – Example

Setup MongoDB Replica Set

Setup MongoDB Replica Set– In this MongoDB Tutorial, we shall learn to set up a replica set on a single machine or across multiple machines connected in a network.

Index

Setup Replica Set on a single machine with multiple instances

In this scenario, there is only a single machine, but we run multiple (for this example, two) mongod instances. One instance acts as PRIMARY and other instances act as SECONDARY.

To set-up Replica Set on a single machine with multiple mongod instances, following is a step by step guide :

1. Start a mongod instance.

Standalone mongod instance by default uses 27017 port and /var/lib/mongodb/ (in ubuntu) data path. We shall use the same for this mongod instance. Run the following command to start a mongod instance.

```
$ mongod –port 27017
```

```
$ mongod –port 27017 –dbpath /var/lib/mongodb –replSet rs0
```

–replSet rs0 : meaning we are setting up a set of nodes to replicate, and the name given to this set is rs0.

2. Start another mongod instance.

We should use unique set of port and data path for each instance, otherwise we may get port binding error and data path issues. For this instance we shall use 27018 port and /var/lib/mongodb1/ data path. You may create the directory, /var/lib/mongodb1/ using following command.

```
$ mkdir
```

```
$ mkdir /var/lib/mongodb1/
```

Run the following command to start mongod instance.

```
$ mongod –port 27017
```

```
$ mongod –port 27017 –dbpath /var/lib/mongodb –replSet rs0
```

Observe that this instance is also started with same Replica Set.

3. Start Replication.

You may create as many number of instances as required and feasible, by following the second step. To start replication, open mongo shell and run the following command :

```
> rs.initiate()
```

Initiate replication

```
root@tutorialkart:/home
```

```
root@tutorialkart:/home/arjun# mongo
MongoDB shell version v3.4.10
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.10
> rs.initiate()
{
    "info2" : "no configuration specified. Using a default configuration for the set",
    "me" : "tutorialkart:27017",
    "ok" : 1
}
```

Now the mongo shell prompt would be changed to rs0:PRIMARY>

You may also check the status using rs.status() method.

4. Add a MongoDB instance to the Replica Set.

To add the MongoDB instances that we already started in step 2, run the following command in the mongo shell that we already opened in the previous step 2.

```
> rs.add(<hostname:port>);
```

To check the hostname, Open a new Terminal and run the following command

```
$ hostname
```

```
root@tutorialkart:~#
```

```
root@tutorialkart:~# hostname
tutorialkart
```

We shall add the second MongoDB instance running at **tutorialkart:27018**.

```
rs0:PRIMARY>
rs.add("tutorialkart:270
```

```
rs0:PRIMARY> rs.add("tutorialkart:27018");
{ "ok" : 1 }
```

Response { “ok” : 1 } meaning addition of mongod instance to the replica set is successful.

5. Check the Status.

You may check the status of the Replica Set by running the following command

```
> rs.status();
```

Replica Set Status

```
rs0:PRIMARY>
rs.status()
```

```
rs0:PRIMARY> rs.status()
{
    "set" : "rs0",
    "date" : ISODate("2017-11-07T09:22:13.627Z"),
    "myState" : 1,
    "term" : NumberLong(1),
    "heartbeatIntervalMillis" : NumberLong(2000),
    "optimes" : {
        "lastCommittedOpTime" : {
            "ts" : Timestamp(1510046524, 1),
            "t" : NumberLong(1)
        },
        "appliedOpTime" : {
            "ts" : Timestamp(1510046524, 1),
            "t" : NumberLong(1)
        },
        "durableOpTime" : {
            "ts" : Timestamp(1510046524, 1),
            "t" : NumberLong(1)
        }
    },
    "members" : [
        {
```

```
            "_id" : 0,
            "name" : "tutorialkart:27017",
            "health" : 1,
            "state" : 1,
            "stateStr" : "PRIMARY",
            "uptime" : 448,
            "optime" : {
                "ts" : Timestamp(1510046524, 1),
                "t" : NumberLong(1)
            },
            "optimeDate" : ISODate("2017-11-07T09:22:04Z"),
            "electionTime" : Timestamp(1510046162, 2),
            "electionDate" : ISODate("2017-11-07T09:16:02Z"),
            "configVersion" : 2,
            "self" : true
        },
        {
            "_id" : 1,
            "name" : "tutorialkart:27018",
            "health" : 1,
            "state" : 2,
            "stateStr" : "SECONDARY",
            "uptime" : 29,
            "optime" : {
                "ts" : Timestamp(1510046524, 1),
                "t" : NumberLong(1)
            },
            "optimeDurable" : {
                "ts" : Timestamp(1510046524, 1),
                "t" : NumberLong(1)
            },
            "optimeDate" : ISODate("2017-11-07T09:22:04Z"),
            "optimeDurableDate" : ISODate("2017-11-07T09:22:04Z"),
            "lastHeartbeat" : ISODate("2017-11-07T09:22:11.678Z"),
            "lastHeartbeatRecv" : ISODate("2017-11-07T09:22:11.698Z"),
            "pingMs" : NumberLong(0),
            "syncingTo" : "tutorialkart:27017",
            "configVersion" : 2
        }
    ],
    "ok" : 1
}
```


Now there are two members in the Replica Set, with tutorialkart:27017 being PRIMARY and tutorialkart:27018 being SECONDARY.

6. ## Check Replication.

Now we shall check if the replication is happening correctly. In the PRIMARY instance,insert a document to MongoDB Database.

On PRIMARY node

```
rs0:PRIMARY> use
```

```
rs0:PRIMARY> use fruits
switched to db fruits
rs0:PRIMARY> db.seasonal.insertOne({ name: "Mango", season: "Summer"})
{
    "acknowledged" : true,
    "insertedId" : ObjectId("5a018ea7c89da78ba2076f25")
}
rs0:PRIMARY>
```

It is time to check in the SECONDARY instance, if this has replicated. To connect to MongoDB instance running at 27018, run the following command

```
$ mongo --port 27018
```

```
$ mongo --port 27018
```

On SECONDARY node

```
rs0:SECONDARY>
```

```
rs0:SECONDARY> use fruits
switched to db fruits
rs0:SECONDARY> db.seasonal.find();
{ "_id" : ObjectId("5a018ea7c89da78ba2076f25"), "name" : "Mango", "season" : "Summer" }
rs0:SECONDARY>
```

Yay!! The replication is happening just as fine.

Set-up Replica Set with multiple machine

In this scenario, there are multiple machines connected over network and there is a MongoDB instance on each of the nodes.

The steps to **Setup Replica Set with multiple machines** is same as that of Setup Replica Set with multiple machine, with only difference being, instead of starting a second instance one the same machine, start the mongoDB instance on the other machines on the network.

Initially, the one instance from which you start replication, acts as PRIMARY and other instances acts as

SECONDARY.

Conclusion :

In this MongoDB Tutorial –**Setup MongoDB Replica Set**, we have learnt to Setup a Replica Set on a single machine (with multiple mongod instances on a single machine) or across multiple machines connected in a network (with mongod instances running on different machines connected in network).