Setup a 3 Node MongoDB Replica Set on Ubuntu

MongoDB, the database for modern applications. If you are not familiar with MongoDB, its a general purpose, document based, distributed database.

What are we building today?

MongoDB Replica Sets are Groups of mongod processes that maintain the same dataset. Replica Sets provide high availability and redundancy. Replication provides redundancy and increases data availability. Copies of the data are replicated across nodes (mongod processes), replication provides a level of fault tolerance against the loss of a single database server.

Replication can provide increased read capacity, as applications can be configured to read from the slave nodes and write to the primary nodes. With data locality, you can spread your nodes across regions and let the applications read from the nodes closest to them.

In a replica set, a primary receives all the write operations and you can only have one primary capable of confirming writes. The primary records all the changes in its oplog.

Install MongoDB

I have provisioned 3 Ubuntu 20.04 nodes which is in a private subnet, so I will be making use of private networking. MongoDB relies on reverse DNS lookups, so I will be adding the private addresses in my hosts file for each node, as all my communication between the servers will happen via private networking:

```
$ cat /etc/hosts
10.163.68.26 mongodb-1.pvt
10.163.68.12 mongodb-2.pvt
10.163.68.9  mongodb-3.pvt
```

This installation is for ubuntu 20.04, if you are using another distribution, have a look at their documentation.

Go ahead update the repositories, add the mongodb repostory, update and install mongodb on all 3 nodes:

```
$ sudo apt update
$ sudo apt install software-properties-common gnupg apt-transport-https ca-certificates -y
$ wget https://repo.mongodb.org/apt/ubuntu/dists/jammy/mongodb-org/7.0/multiverse/binary-amd64/mongodb-org-server_7.0.14_amd64.deb
$ sudo dpkg -i mongodb-org-server_7.0.14_amd64.deb

## Install mongosh

$ wget -qO- https://www.mongodb.org/static/pgp/server-7.0.asc | sudo tee /etc/apt/trusted.gpg.d/server-7.0.asc
$ sudo apt-get install gnupg
$ wget -qO- https://www.mongodb.org/static/pgp/server-7.0.asc | sudo tee /etc/apt/trusted.gpg.d/server-7.0.asc
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list
$ sudo apt-get update
$ sudo apt-get install -y mongodb-mongosh

## Install mongodbump command

$ wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add -
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu $(lsb_release -cs) mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
$ sudo apt update
$ sudo apt install mongodb-database-tools
$ mongodump –version
```

Verify that mongodb is installed:

```
$ mongod --version
db version v5.0.9
```

Keyfile Authorization

All 3 nodes will use the same pre-shared key to authenticate the members that will contribute to the replica set. Let's create the directory where the file will be stored:

```
$ sudo mkdir /var/lib/mongodb-pki
```

Use `openssl` or anything similar to generate a random key:

```
$ openssl rand -base64 741 > keyfile
```

Copy the file over to the other nodes with scp:

```
$ scp ./keyfile mongodb-2.pvt:/tmp/keyfile
$ scp ./keyfile mongodb-3.pvt:/tmp/keyfile
```

Move the keyfile in place on all 3 nodes and change the permissions of the file:

```
$ sudo mv keyfile /var/lib/mongodb-pki/keyfile
$ sudo chmod 600 /var/lib/mongodb-pki/keyfile
$ sudo chown -R mongodb:mongodb /var/lib/mongodb-pki
```

Configure Mongod

Time to configure our mongod instances, we will configure a configuration for each node. So for the first node:

```
storage:
  dbPath: /var/lib/mongodb

systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

net:
  port: 27017
  bindIp: 0.0.0.0

processManagement:
  timeZoneInfo: /usr/share/zoneinfo

operationProfiling:
  mode: "slowOp"
  slowOpThresholdMs: 50

security:
  authorization: enabled
  keyFile: /var/lib/mongodb-pki/keyfile

replication:
  replSetName: my-demo-replset
```

For the 2nd node:

```
storage:
  dbPath: /var/lib/mongodb

systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

net:
  port: 27017
  bindIp: 0.0.0.0
processManagement:
  timeZoneInfo: /usr/share/zoneinfo

operationProfiling:
  mode: "slowOp"
  slowOpThresholdMs: 50

security:
  authorization: enabled
  keyFile: /var/lib/mongodb-pki/keyfile

replication:
  replSetName: my-demo-replset
```

And for the 3rd node:

```
storage:
  dbPath: /var/lib/mongodb

systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

net:
  port: 27017
  bindIp: 0.0.0.0

processManagement:
  timeZoneInfo: /usr/share/zoneinfo

operationProfiling:
  mode: "slowOp"
  slowOpThresholdMs: 50

security:
  authorization: enabled
  keyFile: /var/lib/mongodb-pki/keyfile

replication:
  replSetName: my-demo-replset
```

Enable and Start Mongod

Enable the service and start the mongod service:

```
$ sudo systemctl enable mongod
$ sudo systemctl restart mongod
```

Verify that the process has started:

```
$ sudo systemctl status mongod
  mongod.service - MongoDB Database Server
    Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
    Active: active (running) since Tue 2022-06-20 21:37:52 BST; 26s ago
      Docs: https://docs.mongodb.org/manual
    Main PID: 2950 (mongod)
    CGroup: /system.slice/mongod.service
            └─2950 /usr/bin/mongod --config /etc/mongod.conf

Jul 02 21:37:52 mongodb-1 systemd[1]: Started MongoDB Database Server.
```

Verify that the port is listening:

```
$ sudo netstat -tulpn | grep -E '(State|2950)'
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Progra

tcp        0      0 0.0.0.0:27017           0.0.0.0:*               LISTEN      2950/mongo
```

Go ahead and do the exact same on the other 2 nodes.

Initialize the MongoDB Replica Set

When mongodb starts for the first time it allows an exception that you can logon without authentication to create the root account, but only on localhost. The exception is only valid until you create the user:

```
$ mongo --host 127.0.0.1 --port 27017
MongoDB shell version v5.0.9
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("14ac25ae-016a-4456-b451-60b5d7e866c7") }
MongoDB server version: 5.0.9
Welcome to the MongoDB shell.
>
```

Switch to the admin database and initialize the mongodb replicaset:

```
> use admin
switched to db admin
```

```
> rs.initiate()
{
    "info2" : "no configuration specified. Using a default configuration for the set",
    "me" : "10.163.68.26:27017",
    "ok" : 1
}
```

Now that we have initialized our replicaset config, create the admin user and apply the root role:

```
my-demo-replset:PRIMARY> db.createUser({user: "mongo-admin", pwd: "mongo-pass", roles: [{r
Successfully added user: {
    "user" : "mongo-admin",
    "roles" : [
        {
            "role" : "root",
            "db" : "admin"
        }
    ]
}
my-demo-replset:PRIMARY> exit
```

Now that we have exited the mongo shell, logon to mongodb with the created credentials also pointing at the replica set in the connection string:

```
$ mongo --host my-demo-replset/mongodb-1.pvt:27017 --username mongo-admin --password mongo

MongoDB shell version v5.0.9
connecting to: mongodb://mongodb-1.pvt:27017/?authSource=admin&gssapiServiceName=mongodb&r
2022-06-20T21:48:05.486+0100 I NETWORK  [js] Successfully connected to 10.163.68.26:27017
MongoDB server version: 4.0.10
my-demo-replset:PRIMARY>
```

Have a look at the replica set status using `` `rs.status()` ``, we will see the members in our replica set. At this moment it will only be one node as we have only initiated the one node. This node will elected itself as a primary. We still need to add the other 2 nodes to the replica set.

Have a look at the status:

```
my-demo-replset:PRIMARY> rs.status()
{
    "set" : "my-demo-replset",
    "date" : ISODate("2022-06-20T20:49:19.596Z"),
    "myState" : 1,
    "term" : NumberLong(1),
    "syncingTo" : "",
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "heartbeatIntervalMillis" : NumberLong(2000),
    "optimes" : {
        "lastCommittedOpTime" : {
            "ts" : Timestamp(1562100552, 1),
            "t" : NumberLong(1)
        },
        "readConcernMajorityOpTime" : {
            "ts" : Timestamp(1562100552, 1),
            "t" : NumberLong(1)
        },
        "appliedOpTime" : {
            "ts" : Timestamp(1562100552, 1),
            "t" : NumberLong(1)
        },
        "durableOpTime" : {
            "ts" : Timestamp(1562100552, 1),
            "t" : NumberLong(1)
        }
    },
    "lastStableCheckpointTimestamp" : Timestamp(1562100512, 1),
    "members" : [
        {
            "_id" : 0,
            "name" : "10.163.68.26:27017",
            "health" : 1,
            "state" : 1,
            "stateStr" : "PRIMARY",
            "uptime" : 687,
            "optime" : {
                "ts" : Timestamp(1562100552, 1),
                "t" : NumberLong(1)
            },
            "optimeDate" : ISODate("2022-06-20T20:49:12Z"),
            "syncingTo" : "",
            "syncSourceHost" : "",
            "syncSourceId" : -1,
            "infoMessage" : "",
            "electionTime" : Timestamp(1562100271, 2),
            "electionDate" : ISODate("2022-06-20T20:44:31Z"),
            "configVersion" : 1,
            "self" : true,
            "lastHeartbeatMessage" : ""
        }
    ],
    "ok" : 1,
    "operationTime" : Timestamp(1562100552, 1),
    "$clusterTime" : {
        "clusterTime" : Timestamp(1562100552, 1),
        "signature" : {
            "hash" : BinData(0,"ftyRJpO6BJ5U/oSfP+LzGUeGJvo="),
            "keyId" : NumberLong("6709169581312704514")
        }
    }
```

```
}
```

We can also as the node if its the primary node, using `` `rs.isMaster()` ``:

```
my-demo-replset:PRIMARY> rs.isMaster()
{
    "hosts" : [
        "10.163.68.26:27017"
    ],
    "setName" : "my-demo-replset",
    "setVersion" : 1,
    "ismaster" : true,
    "secondary" : false,
    "primary" : "10.163.68.26:27017",
    "me" : "10.163.68.26:27017",
    "electionId" : ObjectId("7fffffff0000000000000001"),
    "lastWrite" : {
        "opTime" : {
            "ts" : Timestamp(1562100572, 1),
            "t" : NumberLong(1)
        },
        "lastWriteDate" : ISODate("2022-06-20T20:49:32Z"),
        "majorityOpTime" : {
            "ts" : Timestamp(1562100572, 1),
            "t" : NumberLong(1)
        },
        "majorityWriteDate" : ISODate("2022-06-20T20:49:32Z")
    },
    "maxBsonObjectSize" : 16777216,
    "maxMessageSizeBytes" : 48000000,
    "maxWriteBatchSize" : 100000,
    "localTime" : ISODate("2022-06-20T20:49:41.764Z"),
    "logicalSessionTimeoutMinutes" : 30,
    "minWireVersion" : 0,
    "maxWireVersion" : 7,
    "readOnly" : false,
    "ok" : 1,
    "operationTime" : Timestamp(1562100572, 1),
    "$clusterTime" : {
        "clusterTime" : Timestamp(1562100572, 1),
        "signature" : {
            "hash" : BinData(0,"qn7JFMzIuaiK9sVhNkLAq7NqsMI="),
            "keyId" : NumberLong("6709169581312704514")
        }
    }
}
```

Adding Nodes to the Replica Set

Let's add the second node, `` `mongodb-2.pvt` `` to the replica set:

```
my-demo-replset:PRIMARY> rs.add("mongodb-2.pvt:27017")
{
    "ok" : 1,
    "operationTime" : Timestamp(1562100629, 1),
    "$clusterTime" : {
        "clusterTime" : Timestamp(1562100629, 1),
        "signature" : {
            "hash" : BinData(0,"wTNLWmROCfMKaBT4hjZGIyaiCTc="),
            "keyId" : NumberLong("6709169581312704514")
        }
    }
}
```

And also adding the 3rd node, `mongodb-3.pvt` the the replica set:

```
my-demo-replset:PRIMARY> rs.add("mongodb-3.pvt:27017")
{
    "ok" : 1,
    "operationTime" : Timestamp(1562100664, 1),
    "$clusterTime" : {
        "clusterTime" : Timestamp(1562100664, 1),
        "signature" : {
            "hash" : BinData(0,"WeAHspyluGFOOfyxnVS7pCKHPN8="),
            "keyId" : NumberLong("6709169581312704514")
        }
    }
}
```

Now when we look at the replica set status, we should see that the 3 nodes will be part of the replica set:

```
my-demo-replset:PRIMARY> rs.status()
{
    "set" : "my-demo-replset",
    "date" : ISODate("2022-06-20T20:51:20.003Z"),
    "myState" : 1,
    "term" : NumberLong(1),
    "syncingTo" : "",
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "heartbeatIntervalMillis" : NumberLong(2000),
    "optimes" : {
        "lastCommittedOpTime" : {
            "ts" : Timestamp(1562100664, 1),
            "t" : NumberLong(1)
        },
        "readConcernMajorityOpTime" : {
            "ts" : Timestamp(1562100664, 1),
            "t" : NumberLong(1)
        },
        "appliedOpTime" : {
            "ts" : Timestamp(1562100664, 1),
            "t" : NumberLong(1)
        },
        "durableOpTime" : {
            "ts" : Timestamp(1562100664, 1),
            "t" : NumberLong(1)
        }
    },
    "lastStableCheckpointTimestamp" : Timestamp(1562100629, 1),
    "members" : [
        {
            "_id" : 0,
            "name" : "10.163.68.26:27017",
            "health" : 1,
            "state" : 1,
            "stateStr" : "PRIMARY",
            "uptime" : 808,
            "optime" : {
                "ts" : Timestamp(1562100664, 1),
                "t" : NumberLong(1)
            },
            "optimeDate" : ISODate("2022-06-20T20:51:04Z"),
            "syncingTo" : "",
            "syncSourceHost" : "",
            "syncSourceId" : -1,
            "infoMessage" : "",
            "electionTime" : Timestamp(1562100271, 2),
            "electionDate" : ISODate("2022-06-20T20:44:31Z"),
            "configVersion" : 3,
            "self" : true,
            "lastHeartbeatMessage" : ""
            },
                    {
                        "_id" : 1,
                        "name" : "mongodb-2.pvt:27017",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 50,
                        "optime" : {
                            "ts" : Timestamp(1562100664, 1),
                            "t" : NumberLong(1)
                        },
                        "optimeDurable" : {
                            "ts" : Timestamp(1562100664, 1),
                            "t" : NumberLong(1)
                        },
                        "optimeDate" : ISODate("2022-06-20T20:51:04Z"),
                        "optimeDurableDate" : ISODate("2022-06-20T20:51:04Z"),
                        "lastHeartbeat" : ISODate("2022-06-20T20:51:18.288Z"),
                        "lastHeartbeatRecv" : ISODate("2022-06-20T20:51:19.843Z"),
                        "pingMs" : NumberLong(1),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "configVersion" : 3
                    },
                    {
                        "_id" : 2,
                        "name" : "mongodb-3.pvt:27017",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 15,
                        "optime" : {
                            "ts" : Timestamp(1562100664, 1),
                            "t" : NumberLong(1)
                        },
                        "optimeDurable" : {
                            "ts" : Timestamp(1562100664, 1),
                            "t" : NumberLong(1)
                        },
                        "optimeDate" : ISODate("2022-06-20T20:51:04Z"),
                        "optimeDurableDate" : ISODate("2022-06-20T20:51:04Z"),
                        "lastHeartbeat" : ISODate("2022-06-20T20:51:18.292Z"),
                        "lastHeartbeatRecv" : ISODate("2022-06-20T20:51:19.662Z"),
                        "pingMs" : NumberLong(1),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "configVersion" : 3
                    }
                ],
                "ok" : 1,
                "operationTime" : Timestamp(1562100664, 1),
                "$clusterTime" : {
                    "clusterTime" : Timestamp(1562100664, 1),
                    "signature" : {
                        "hash" : BinData(0,"WeAHspyluGFOOfyxnVS7pCKHPN8="),
                        "keyId" : NumberLong("6709169581312704514")
                    }
                }
            }
```

Connecting to the Replica set

Connecting to a replicaset will always route you to the primary:

```
$ mongo --host my-demo-replset/mongodb-2.pvt:27017 --username mongo-admin --
  password mongo-pass --authenticationDatabase admin
```

```
MongoDB shell version v5.0.9
connecting to:
mongodb://mongodb2.pvt:27017/?authSource=admin&gssapiServiceName=mongodb&replicaSet
=my-demo-replset
2022-06-20T21:53:03.943+0100 I NETWORK  [js] Starting new replica set monitor for
my-demo-replset/mongodb-2.pvt:27017
2022-06-20T21:53:03.945+0100 I NETWORK  [js] Successfully connected to mongodb-
2.pvt:27017 (1 connections now open to mongodb-2.pvt:27017 with a 5 second timeout)
2022-06-20T21:53:03.946+0100 I NETWORK  [js] Successfully connected to
10.163.68.26:27017 (1 connections now open to 10.163.68.26:27017 with a 5 second
timeout)
2022-06-20T21:53:03.946+0100 I NETWORK  [js] changing hosts to my-demo-
replset/10.163.68.26:27017,mongodb-2.pvt:27017,mongodb-3.pvt:27017 from my-demo-
replset/mongodb-2.pvt:27017
my-demo-replset:PRIMARY>
```

We can verify if we are on the primary by doing the following:

```
my-demo-replset:PRIMARY> rs.isMaster()['ismaster']
true

my-demo-replset:PRIMARY> rs.isMaster()['me']
10.163.68.26:27017
```

If you exit and connect without the replica set in the connection string, then it will direct you to that node:

```
$ mongo --host mongodb-2.pvt:27017 --username mongo-admin --password mongo-pass --
  authenticationDatabase admin

MongoDB shell version v5.0.9
connecting to: mongodb://mongodb-
2.pvt:27017/?authSource=admin&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3c343871-8814-4136-859d-350c78c2a2a7") }
MongoDB server version: 4.0.10
my-demo-replset:SECONDARY>
```

Reading from Slave Nodes

When we have a replica set we can write to our Primary (master) node and Read from our Secondary (slave) nodes. Before we can read from the secondary nodes, we need to tell mongodb that we want to read.

By default when you connect to a secondary node and want to read you will get this exception:

```
my-demo-replset:SECONDARY> show databases;
2022-06-20T21:59:13.169+0100 E QUERY    [js] Error: listDatabases failed:{
    "operationTime" : Timestamp(1562101152, 1),
    "ok" : 0,
    "errmsg" : "not master and slaveOk=false",
    "code" : 13435,
    "codeName" : "NotMasterNoSlaveOk”,
...
```

We first need to instruct mongodb that we want to read:

```
my-demo-replset:SECONDARY> rs.slaveOk()
```

Now that we have done that, we can read from the secondary:

```
my-demo-replset:SECONDARY> show databases;
admin   0.000GB
config  0.000GB
local   0.000GB
```

Upgrading a MongoDB Replica Set

When you want to upgrade a mongodb node in a replica set, you can accomplish that with a rolling upgrade.

You will first shutdown a secondary, do the upgrade, start the node, then upgrade the other secondary. When all the secondaries are upgraded, then upgrade the primary. This involves initiating a stepdown on the primary which will trigger an election, where one of the secondary nodes will be promoted as a primary, once the primary is selected on another node, the old primary can be upgraded.

The correct way of shutting down a mongodb node is with using db.shutdownServer():

```
my-demo-replset:PRIMARY> use admin
switched to db admin
my-demo-replset:PRIMARY> db.shutdownServer()
```

And to stepdown a node from primary to secondary:

```
my-demo-replset:PRIMARY> rs.stepDown()
my-demo-replset:SECONDARY>
```