MongoDB Replicaset

In AWS we created 3 ubuntu 18.04 Instances for creating mongodb replica set. Instance type - t2.micro

Initially add hostnames for all 3 nodes in the replica set.

$ vi /etc/hosts

```
root@ip-172-31-93-200:~# cat /etc/hosts
172.31.93.200 db-node1
172.31.92.245 db-node2
172.31.84.52  db-node3
```

Install Mongodb in all 3 nodes.

$ sudo apt-get update

$ wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -

$ echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu (lsb_release -cs)/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list

$ sudo apt-get update

$ sudo apt install mongodb-org -y

$ sudo systemctl enable mongod

$ sudo systemctl start mongod

Verify Installation:

$ sudo systemctl status mongod

```
root@ip-172-31-93-200:~# mongo --version
MongoDB shell version v4.4.6
Build Info: {
    "version": "4.4.6",
    "gitVersion": "72e66213c2c3eab37d9358d5e78ad7f5c1d0d0d7",
    "openSSLVersion": "OpenSSL 1.1.1  11 Sep 2018",
    "modules": [],
    "allocator": "tcmalloc",
    "environment": {
        "distmod": "ubuntu1804",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
```

Once we installed mongodb in Server, add replica set name unser replication in config file in all 3 nodes.

$ nano /etc/mongod.conf

```
net:
  port: 27017
  bindIp: 127.0.0.1


# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo

#security:

#operationProfiling:

replication:
  replSetName: rs-demo
#sharding:
```

Save and exit

$ sudo systemctl enable mongod

$ sudo systemctl stop mongod

$ sudo systemctl start mongod

$ mongo

Initiate replica-set in master node using following command.

$ rs.initiate()

```
> rs.initiate()
{
        "info2" : "no configuration specified. Using a default configuration for the set",
        "me" : "ip-172-31-93-200:27017",
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622641756, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1622641756, 1)
}
```

$ rs.isMaster()

```
"hosts" : [
        "ip-172-31-93-200:27017"
],
"setName" : "rs-demo",
"setVersion" : 1,
"ismaster" : true,
"secondary" : false,
"primary" : "ip-172-31-93-200:27017",
"me" : "ip-172-31-93-200:27017",
"electionId" : ObjectId("7fffffff0000000000000001"),
```

$ rs.add(“172.31.92.245:27017”)

```
rs-demo:PRIMARY> rs.add("172.31.92.245:27017")
{
        "ok" : 1,
```

$ rs.add(“172.31.84.52:27017”)

```
rs-demo:PRIMARY> rs.add("172.31.84.52:27017")
{
        "ok" : 1,
```

$ rs.status()

```
"members" : [
        {
                "_id" : 0,
                "name" : "ip-172-31-93-200:27017",
                "health" : 1,
                "state" : 1,
                "stateStr" : "PRIMARY",
```

```
        "_id" : 1,
        "name" : "172.31.92.245:27017",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
```

```
        "_id" : 2,
        "name" : "172.31.84.52:27017",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
```

```
rs-demo:PRIMARY> use test-replicaset
switched to db test-replicaset
```

```
rs-demo:PRIMARY> db.eclhur.insert({"name":"Elchuru"})
WriteResult({ "nInserted" : 1 })
```

```
rs-demo:PRIMARY> show dbs
admin            0.000GB
config           0.000GB
local            0.000GB
test-replicaset  0.000GB
rs-demo:PRIMARY> show collections
eclhur
```

```
rs-demo:SECONDARY> show dbs
admin            0.000GB
config           0.000GB
local            0.000GB
test-replicaset  0.000GB
rs-demo:SECONDARY> use test-replicaset
switched to db test-replicaset
rs-demo:SECONDARY> show collections
eclhur
```