# Mongodb-cluster(Sharding) without replication

# Step1: We have 4 machines to test sharding:

1) monogs router: 192.168.10.1

2) Mongo-config 192.168.10.2

3) Shard-node-1 192.168.10.3

4) Shard-node-2 192.168.10.4

# Step2: Installation of mongodb on each-server:

$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4

$ echo "deb [ arch=amd64 ] <https://repo.mongodb.org/apt/ubuntu> bionic/mongodb-org/4.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org.list

$ sudo apt update

$ sudo apt install mongodb-org

# Step3 : Create an Admin-user for config and shard nodes:

* Mongo
* Use admin
* db.createUser({user: "**sanyam**", pwd: "**sanyam**", roles:[{role: "root", db: "admin"}]})

# Step4: Setup mongodb-auth on each server:

openssl rand -base64 756 > ~/mongodb.key

mkdir /opt/mongo

mv ~/mongo.key /opt/mongo

sudo chown -R mongodb:mongodb /opt/mongo/mongo.key

sudo chmod -R 400 /opt/mongo/mongo.key

Copy this key on each server at given location.

## Step5: Set-up config server for sharding:

Login to config server and open file:

Vim /etc/mongod.conf

port: 27019

bindIp: 192.168.10.2

security:

keyFile: /opt/mongo/mongo.key

replication:

replSetName: config-rs

sharding:

clusterRole: configsvr

Save and close

**Restart mongodb-service**

$ Systemctl restart mongod.service

**Initiate the replica-set on config-servers**

Mongo - - host 192.168.10.2 --port 27021 -u sanyam --authenticationDatabase admin

Mongo> rs.initiate()

# Step6: creation of Shard-server-1

Open the mongod conf file and change the configuration:

net:

port: 27018

bindIp: 192.168.10.3

security:

keyFile: /opt/mongo/mongo.key

replication:

replSetName: rs0

sharding:

clusterRole: shardsvr

Replication is Optional in this part here

**Restart the mongodb-service:**

$ systemctl restart mongod.service

# Step7: creation of Shard-server-2

Open the mongod conf file and change the configuration:

net:

port: 27018

bindIp: 192.168.10.4

security:

keyFile: /opt/mongo/mongo.key

replication:

replSetName: rs1

sharding:

clusterRole: shardsvr

Replication is Optional in this part here

**Restart the mongodb-service:**

$ systemctl restart mongod.service

# Step8: Configure Mongos (query-router)

Login into the instance and stop the mongodb service:

* systemctl stop mongod
* systemctl disable mongod

configure mongos:

* vim /etc/mongos.conf

systemLog:

destination: file

logAppend: true

path: /var/log/mongodb/mongos.log

net:

port: 27017

bindIp: 192.168.10.1

security:

keyFile: /opt/mongo/mongo/key

sharding:

configDB: config-rs/192.168.10.2:27019

Next, create a systemd service unit file for mongos

sudo nano /lib/systemd/system/mongos.service

Add the following parameters:

[Unit]

Description=Mongo Cluster Router

After=network.target

[Service]

User=mongodb

Group=mongodb

ExecStart=/usr/bin/mongos --config /etc/mongos.conf

LimitFSIZE=infinity

LimitCPU=infinity

LimitAS=infinity

LimitNOFILE=64000

LimitNPROC=64000

TasksMax=infinity

TasksAccounting=false

[Install]

WantedBy=multi-user.target

Now Save and close.

Start the mongos service:

sudo systemctl start mongos

sudo systemctl enable mongos

# Step9: Add shard to this cluster:

Login to the query server

mongo --host 192.168.10.1 --port 27017 -u sanyam --authenticationDatabase admin

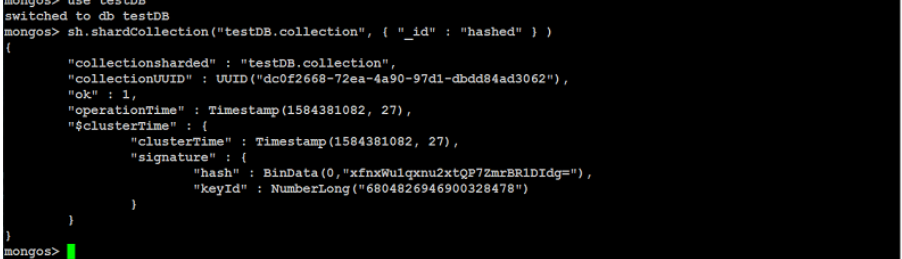
mongos> sh.addShard( "192.168.10.3:27018")

mongos> sh.addShard( "192.168.10.4:27018")

Testing the cluster :

Mongos> use testDB

Mongos> db.testCollection.ensureIndex( { \_id : "hashed" } )

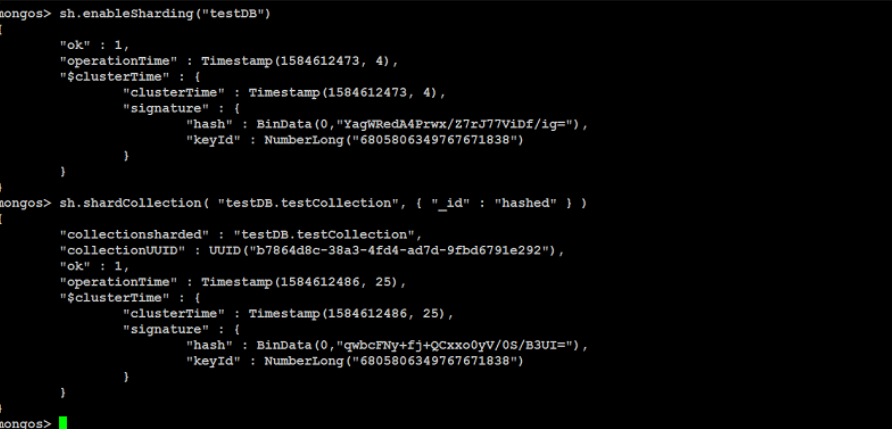


Enabling sharding for new or existing Database:

sh.enableSharding("testDB")

sh.shardCollection( "testDB.testCollection", { "\_id" : "hashed" } )

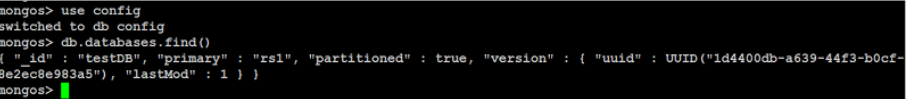
Output will be similar:



To verify sharding is Successfully created:

* use config
* db.databases.find()

If sharding enabled properly it will be looks like this:



# Step10: Test MongoDB sharding cluster:

Mongos> use testDb

Mongos> for (var i = 1; i <= 10000; i++) db.testCollection.insert( { x : i } )

**Run the command to check the data distribution :**

Mongos> db.testCollection.getShardDistribution()

------------------------------------------------------------------xx-----------------------------------------------------------------