MongoDB sharding

Config servers - 172.31.85.222 & 172.31.84.174
Mongos - 172.31.91.138
Sharding servers - 172.31.83.139 & 172.31.88.13

Install Mongo in all servers:

```
 $ 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
```

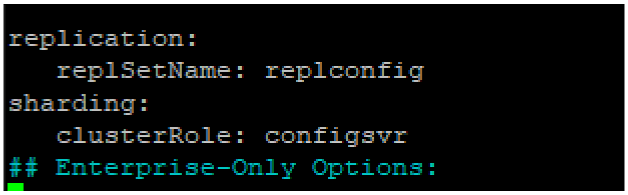
```
$ sudo apt install mongodb-org -y
```

```
$ sudo systemctl enable mongod
```

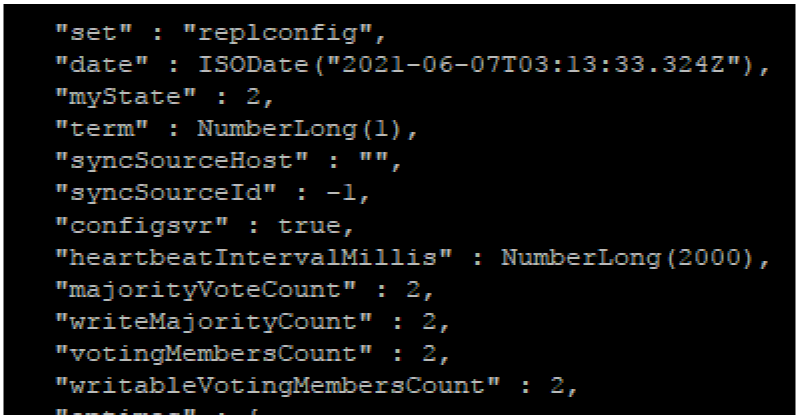
```
$ sudo systemctl start mongod
```

Login in to the config server.

```
$ nano /etc/mongod.conf
```



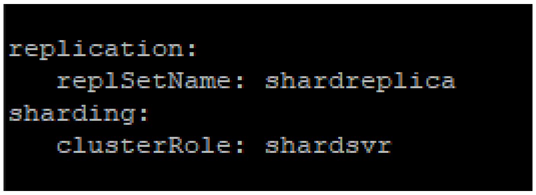
```
replication:
   replSetName: replconfig
sharding:
   clusterRole: configsvr
## Enterprise-Only Options:
```



```
   "set" : "replconfig",
   "date" : ISODate("2021-06-07T03:13:33.324Z"),
   "myState" : 2,
   "term" : NumberLong(1),
   "syncSourceHost" : "",
   "syncSourceId" : -1,
   "configsvr" : true,
   "heartbeatIntervalMillis" : NumberLong(2000),
   "majorityVoteCount" : 2,
   "writeMajorityCount" : 2,
   "votingMembersCount" : 2,
   "writableVotingMembersCount" : 2,
   "optimes" : {
```

Login to the shard server:

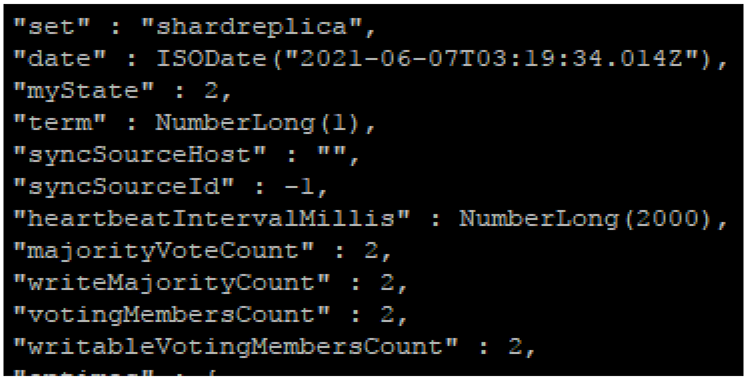
$ nano /etc/mongod.conf



```
replication:
   replSetName: shardreplica
sharding:
   clusterRole: shardsvr
```

$ rs.initiate()

$ rs.status()

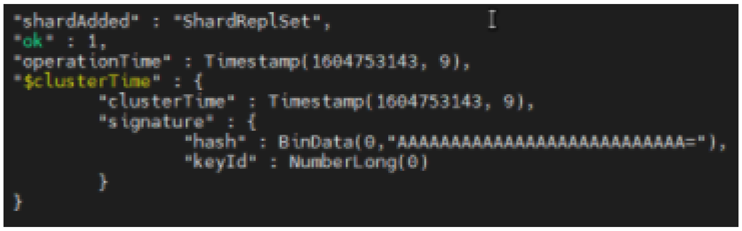


```
"set" : "shardreplica",
"date" : ISODate("2021-06-07T03:19:34.014Z"),
"myState" : 2,
"term" : NumberLong(1),
"syncSourceHost" : "",
"syncSourceId" : -1,
"heartbeatIntervalMillis" : NumberLong(2000),
"majorityVoteCount" : 2,
"writeMajorityCount" : 2,
"votingMembersCount" : 2,
"writableVotingMembersCount" : 2,
```

Login in to the Mongos server.

Enable sharding.

```
sh.addShard(
"Shardreplica/54.89.248.226:27017,Shardreplica/184.72.107.153:27017")
```

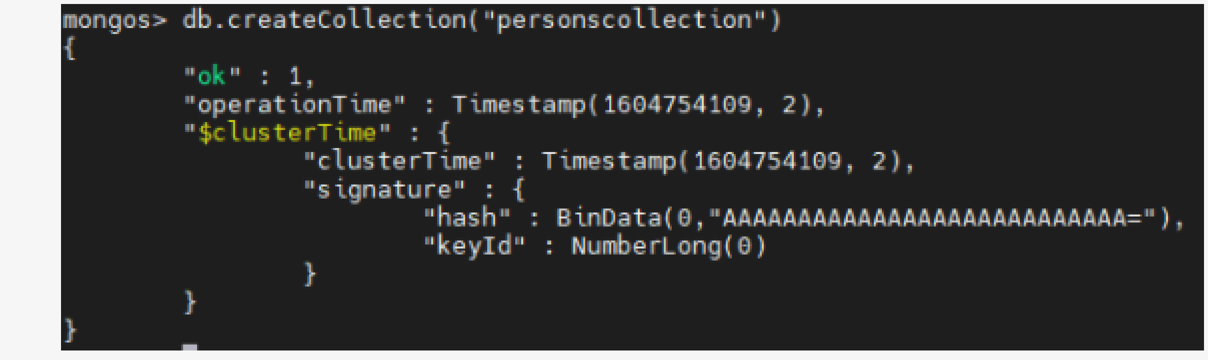


```
"shardAdded" : "ShardReplSet",
"ok" : 1,
"operationTime" : Timestamp(1604753143, 9),
"$clusterTime" : {
        "clusterTime" : Timestamp(1604753143, 9),
        "signature" : {
                "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                "keyId" : NumberLong(0)
        }
}
```

use persons
sh.enableSharding("persons")

Creating the sharding dataset:

```
db.createCollection("personscollection")
```



```
mongos> db.createCollection("personscollection")
{
        "ok" : 1,
        "operationTime" : Timestamp(1604754109, 2),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1604754109, 2),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

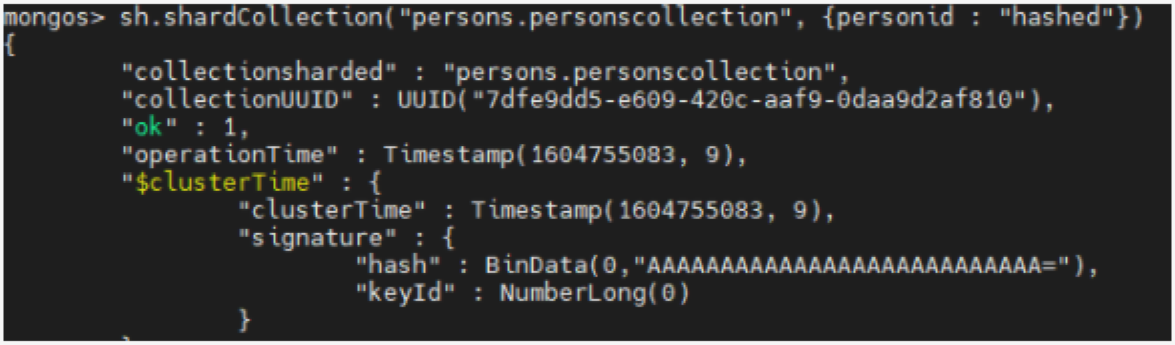
```
 db.personscollection.createIndex({personid: -1})
```

```
db.personscollection.createIndex({personid: -1})
```

```
db.personscollection.insertOne({personid: 10001})
```

Enabling sharding for the personscollection:

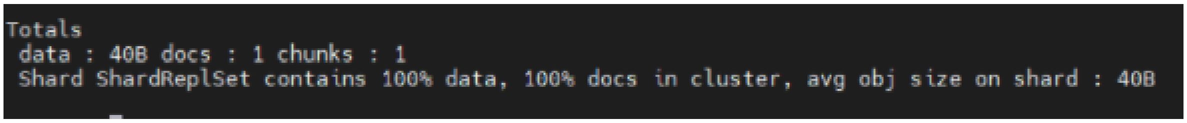
```
db.personscollection.ensureIndex({personid : "hashed"})

sh.shardCollection("persons.personscollection", {personid : "hashed"})
```



```
mongos> sh.shardCollection("persons.personscollection", {personid : "hashed"})
{
        "collectionsharded" : "persons.personscollection",
        "collectionUUID" : UUID("7dfe9dd5-e609-420c-aaf9-0daa9d2af810"),
        "ok" : 1,
        "operationTime" : Timestamp(1604755083, 9),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1604755083, 9),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
```

Verify if sharding is working as intended

db.personscollection.getShardDistribution()



```
Totals
 data : 40B docs : 1 chunks : 1
 Shard ShardReplSet contains 100% data, 100% docs in cluster, avg obj size on shard : 40B
```