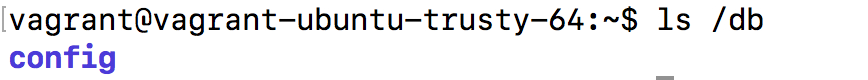
Question 3:

**I Set up Shards**

ssh -p 1999 [vagrant@127.0.0.1](mailto:vagrant@127.0.0.1) enter the Ubuntu-trusty064 system

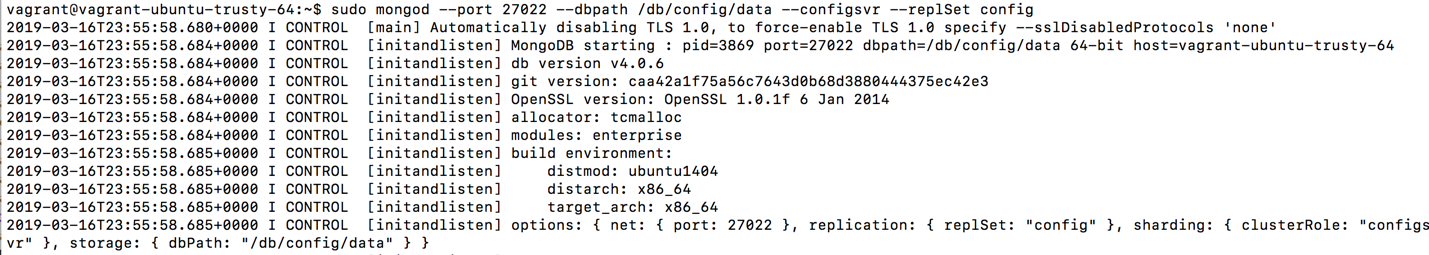
1 Set up a config replica set

(a)create a directory: vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/config/data

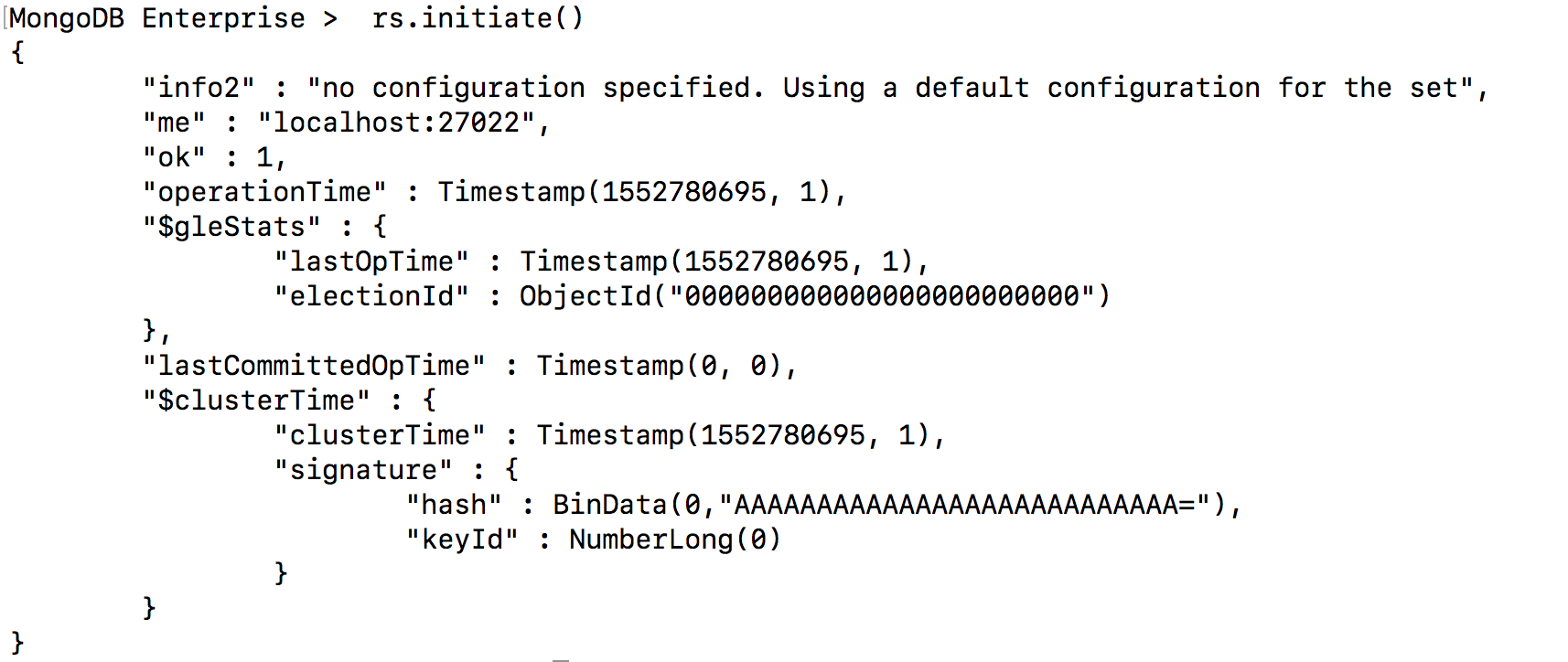


(b) Start a mongod serving as a config server:

vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --port 27022 --dbpath /db/config/data --configsvr --replSet config

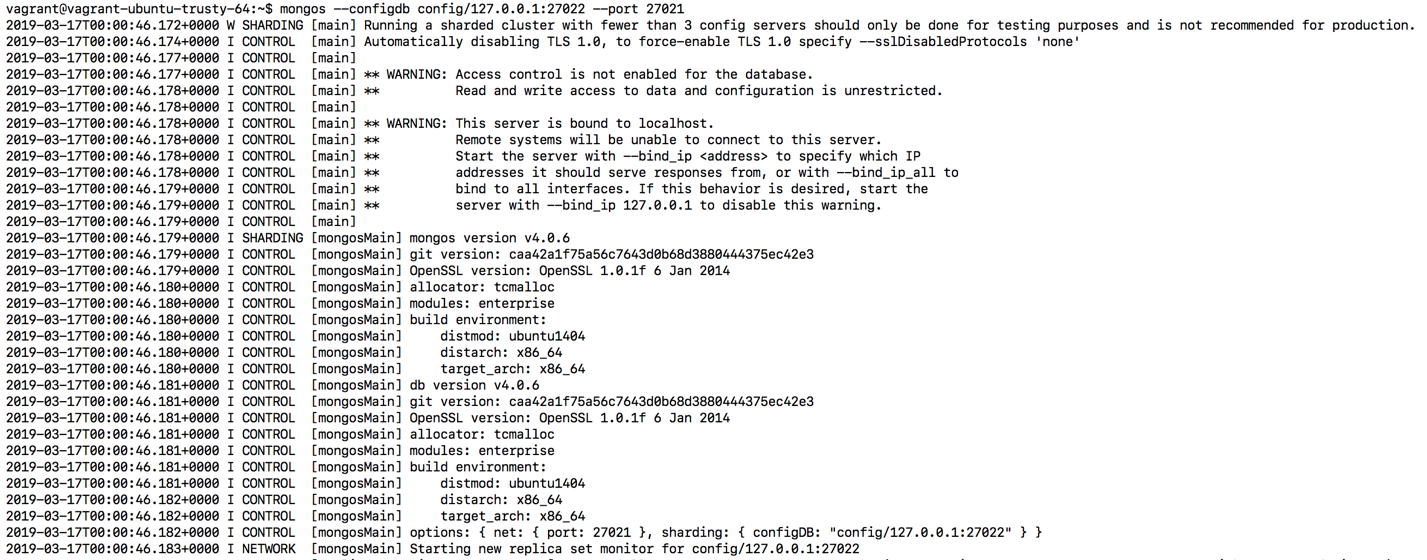


(c) In a new window, initialize the replica set: vagrant@vagrant-ubuntu-trusty-64:~$ mongo --port 27022



2. Set up the shard controller (mongos) in a new window

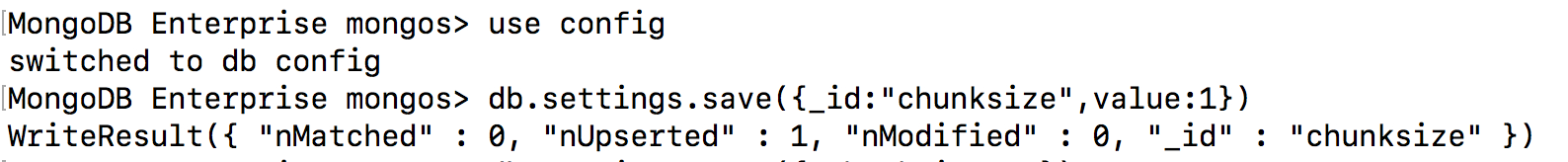
vagrant@vagrant-ubuntu-trusty-64:~$ mongos --configdb config/127.0.0.1:27022 --port 27021



Then get into the Shard Controller (27021) with command:

vagrant@vagrant-ubuntu-trusty-64:~$ mongo 127.0.0.1:27021 to setup chunk size

with command: use config => db.settings.save({\_id:"chunksize",value:1})

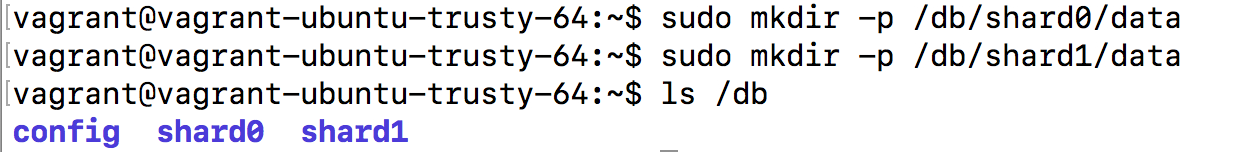


3 Bring up the two shard servers

1. Open two new terminals, one for each server and make directories.

vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/shard0/data

vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/shard1/data

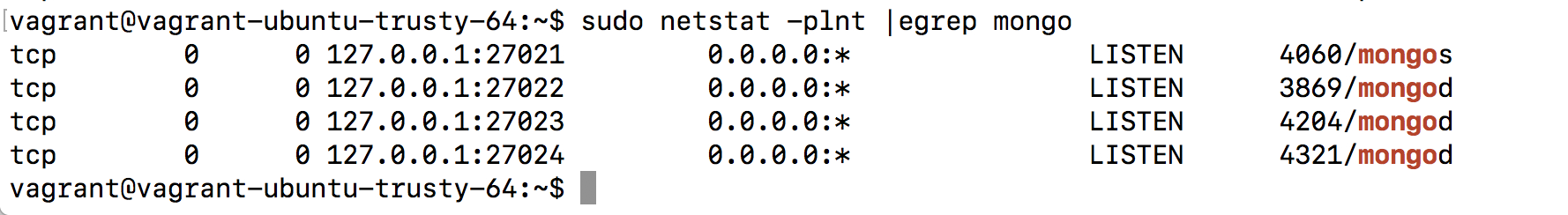


1. Start each shard server with port 27023 and 27024 with command:

sudo mongod --shardsvr --port 27023 --dbpath /db/shard0/data

sudo mongod --shardsvr --port 27024 --dbpath /db/shard1/data

1. Checking the port status with sudo netstat –plnt | egrep mongo. It shows all current opened port for listen.



4. Tell the sharding system (mongos) where the shard servers are located.

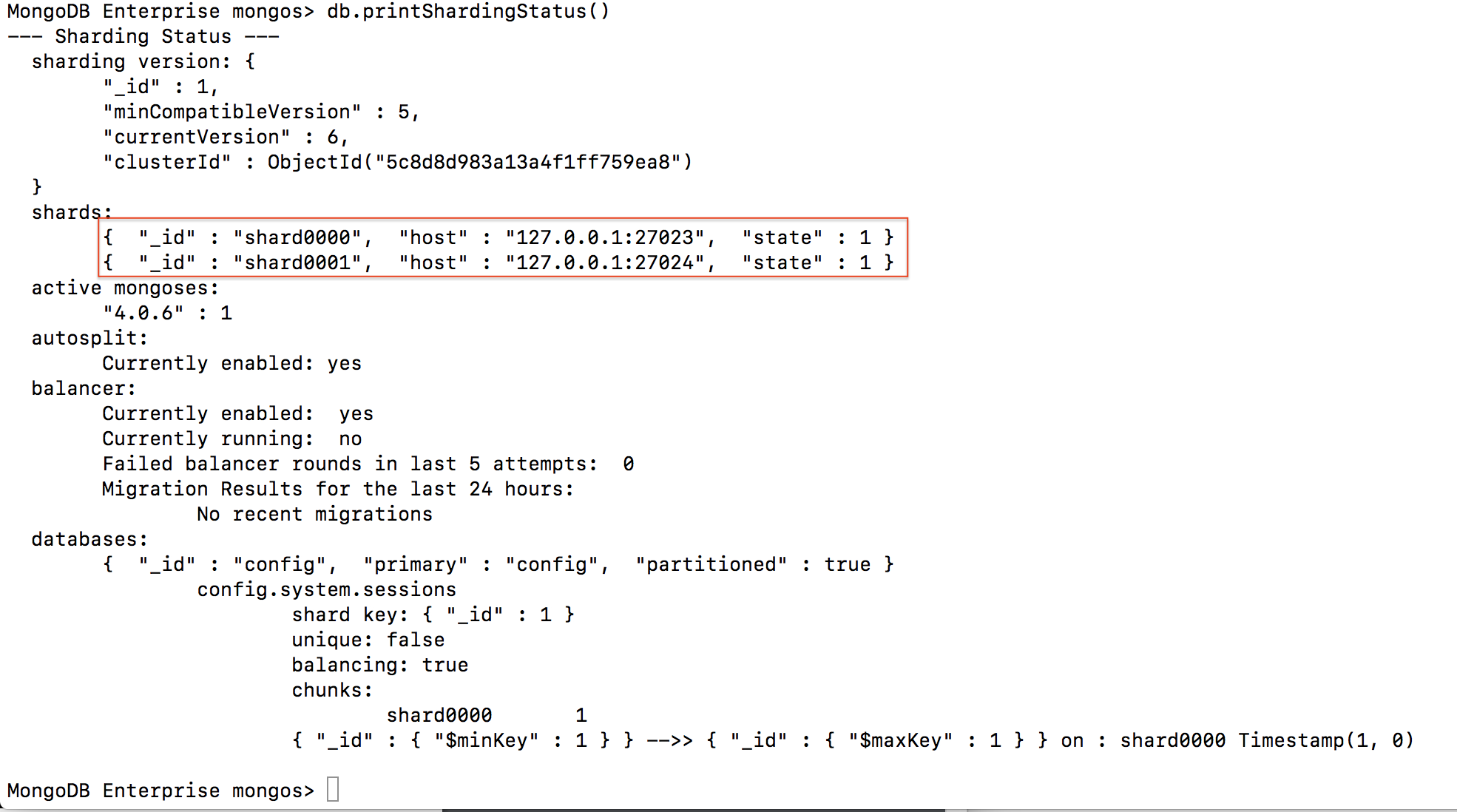
Add each shard server to the system with command:

sh.addShard("127.0.0.1:27023") sh.addShard("127.0.0.1:27024")

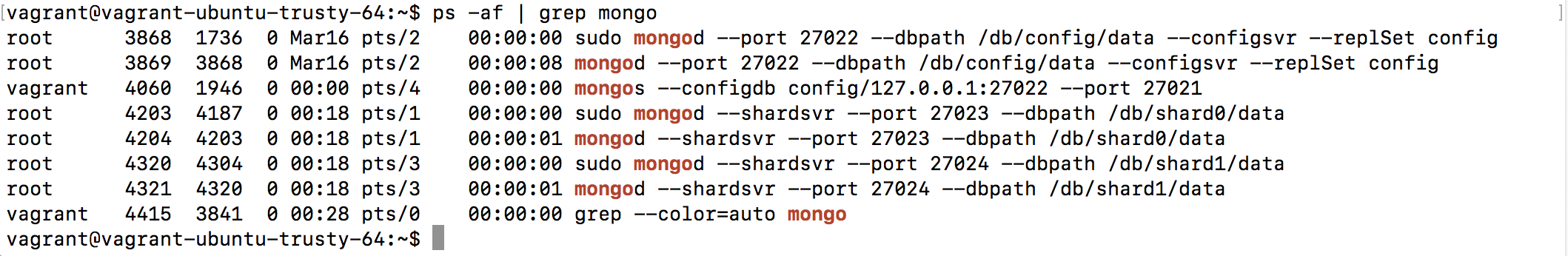


5. Check the shards (before populating data)

Use the Command: db.printShardingStatus()



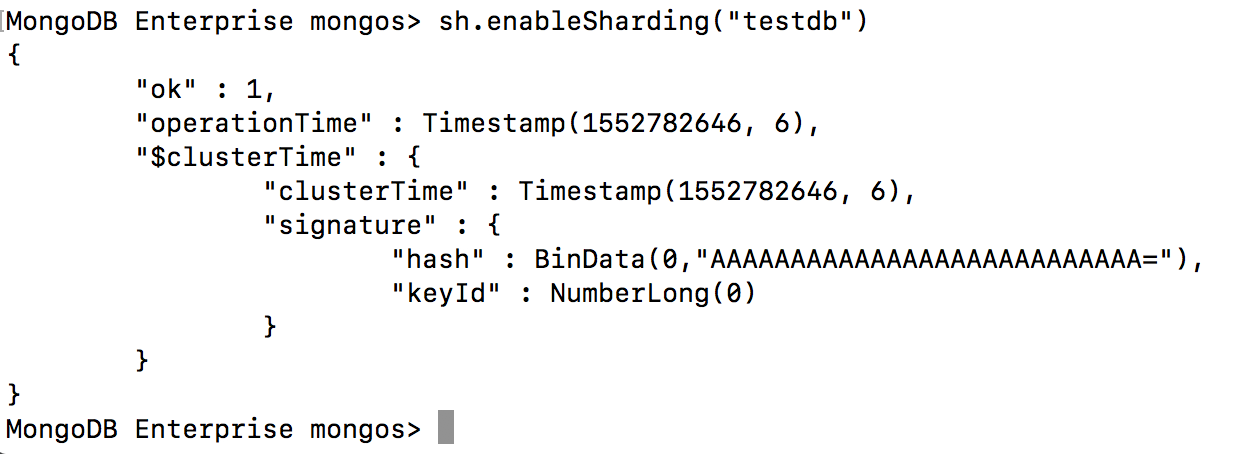
also with command: ps –af | grep mongo to check port match details



6. Explicitly tell both database and collection that wanted to be sharded.

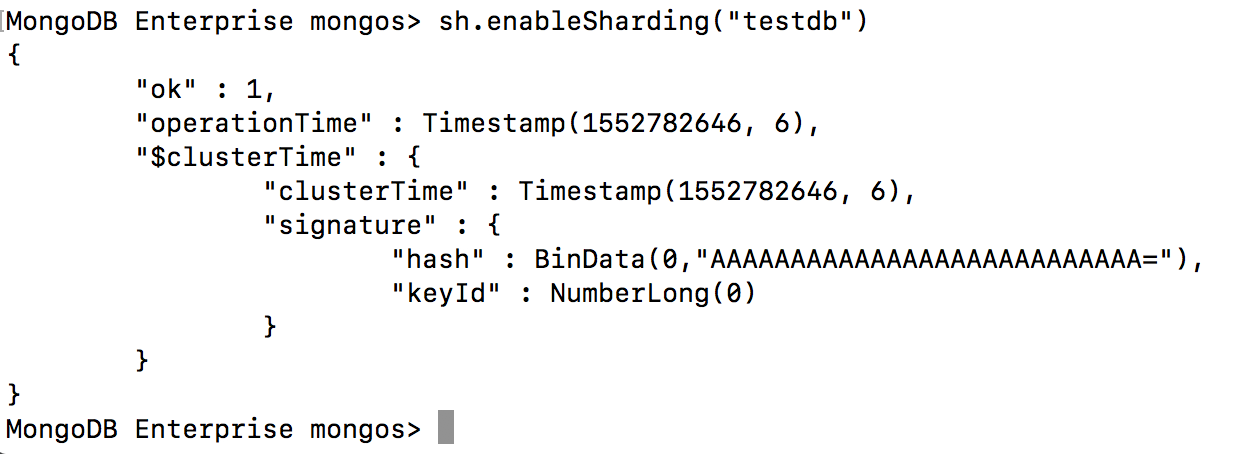
(a) Enabling a database for sharding is a prerequisite to sharding on of its collection

with command: mongos> sh.enableSharding("testdb")

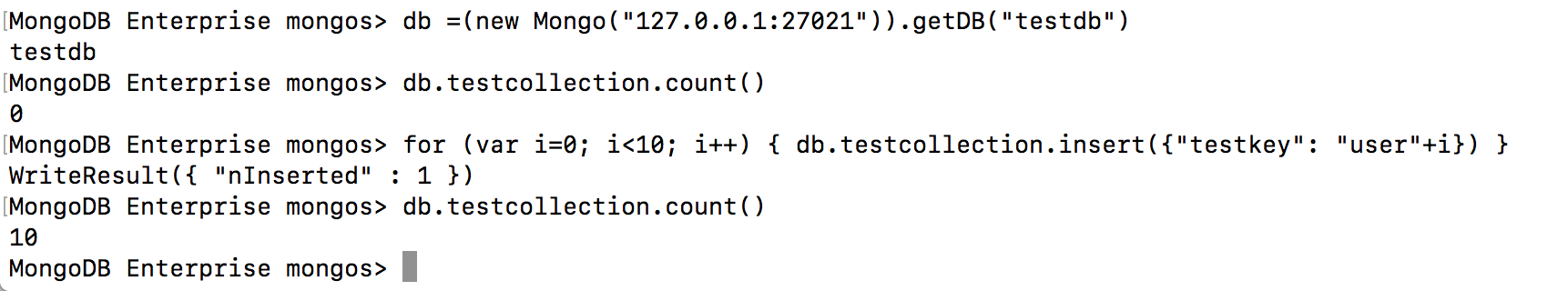


(b) Then given collection is going to be split into chunks with command:

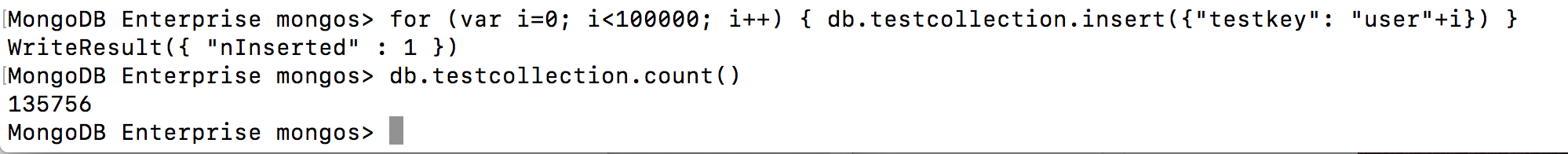
sh.shardCollection("testdb.testcollection", {testkey:1})



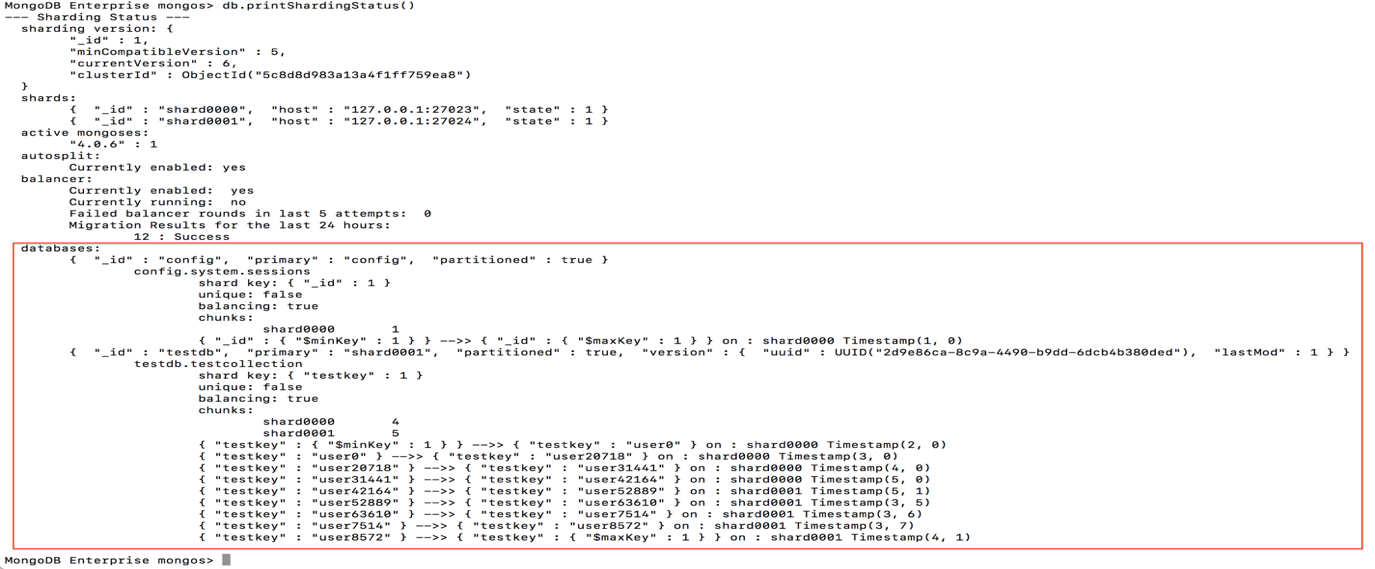
7. Populate data in testdb.testCollection



Now insert 100.000 more documents



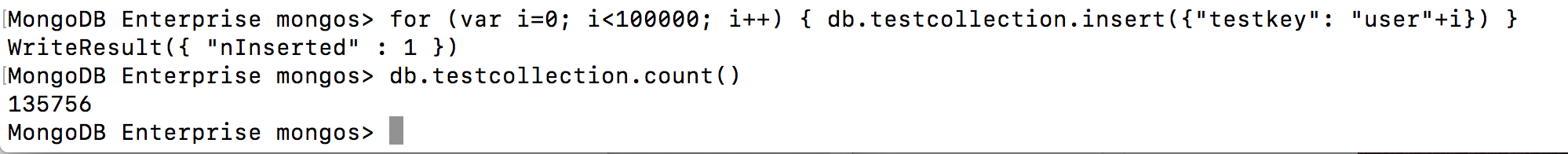
By checking it’s status with command: db.printShardingStatus()



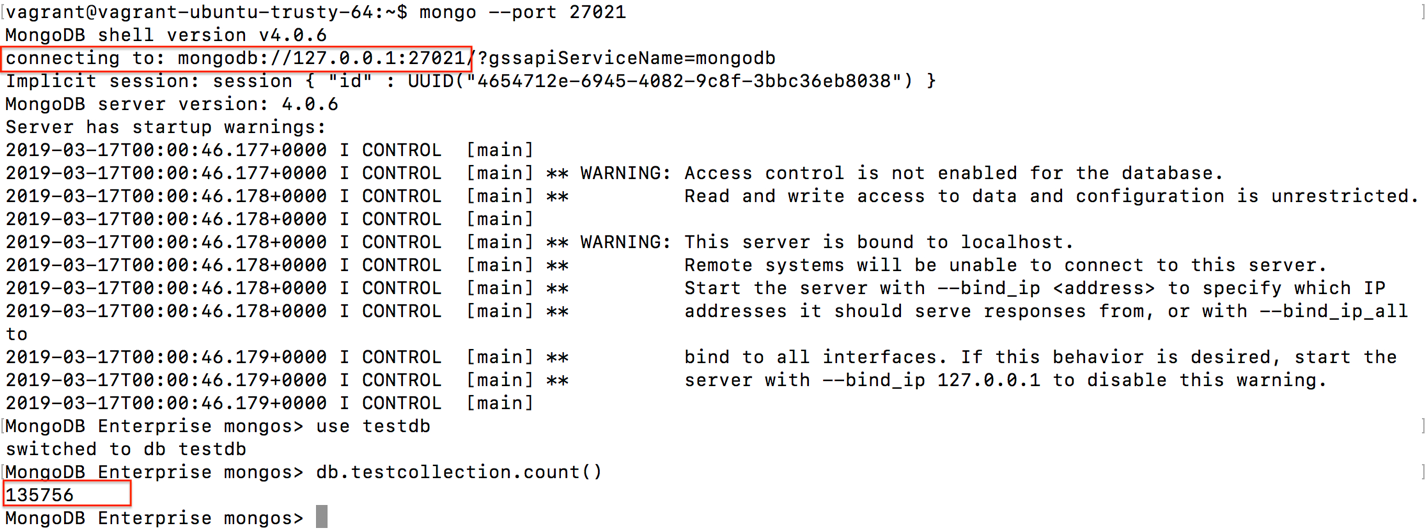
9. Connect to the shards and see how many documents are stored in each shard.

During population, there are partially created 35746 documents so the document number is

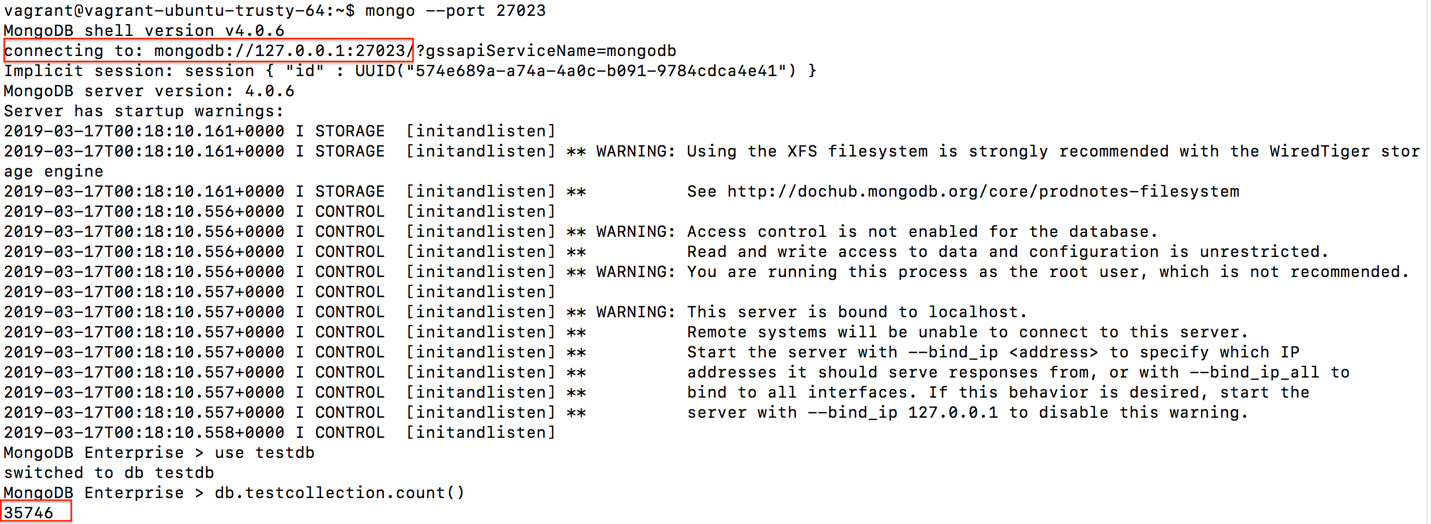
10 +35746+ 100000



By checking number of documents in 27021, we get total 135756 documents as shown below.

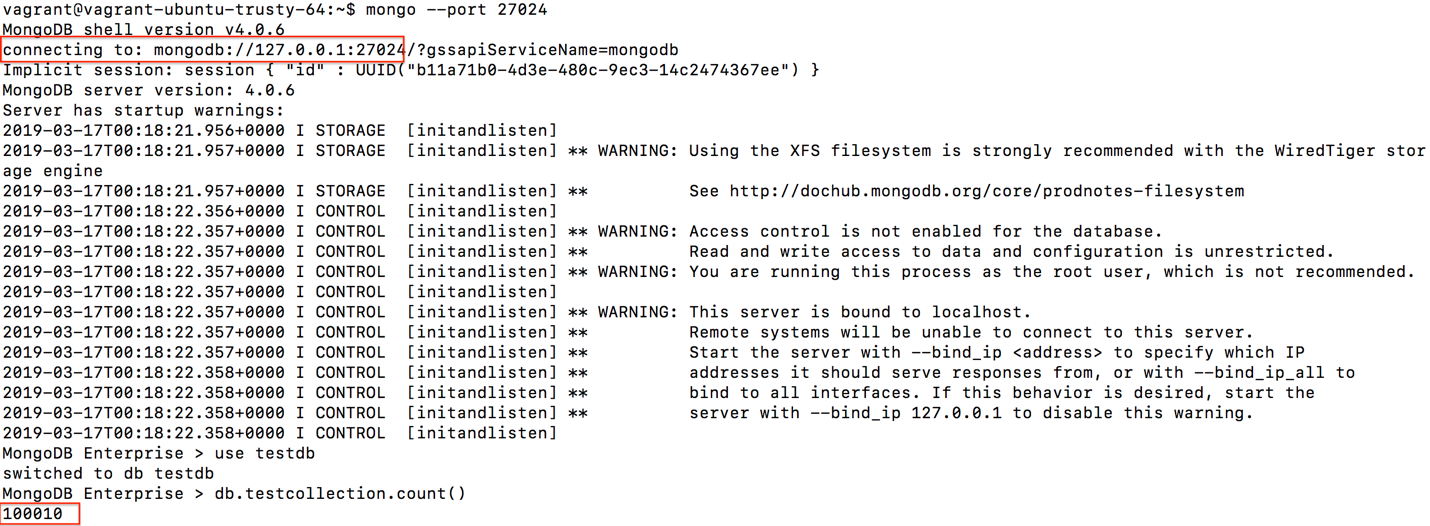


By checking number of documents in 27023, we get partitioned number 35746 shown below



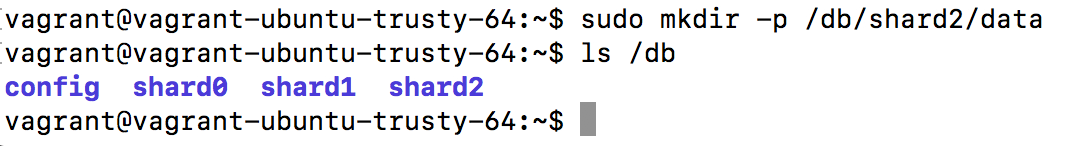
And by checking number of documents in 27024, we get partitioned number 100010 below

And which is 35746+100010=135756



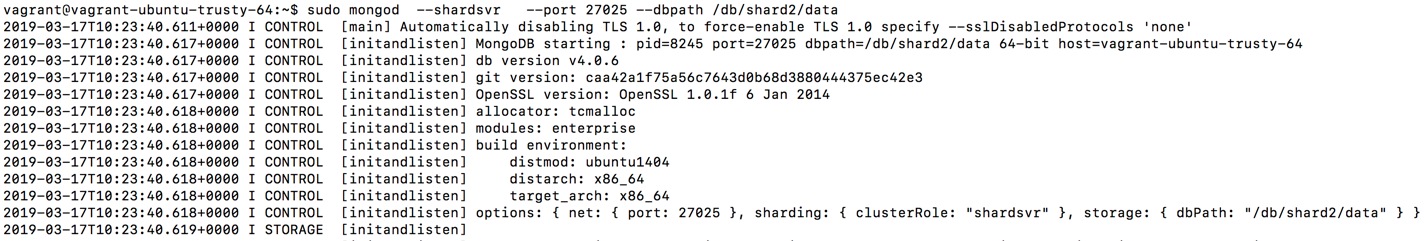
**II. Adding a new shard to the cluster**

1. Make a directory and launch mongod for the shard 2

****

connect the instance of shard2 with open port number 27025 with the command:

sudo mongod --shardsvr --port 27025 –dbpath /db/shard2/data



2. Add the new shard to the cluster. Connect to mongos and addShard.



3. Check the shard status after adding shard 2.



Also After added Shard2, number of document the total document has change to 123841



By checking collection number in each shard. We get

27023: 35746



27024: 64264



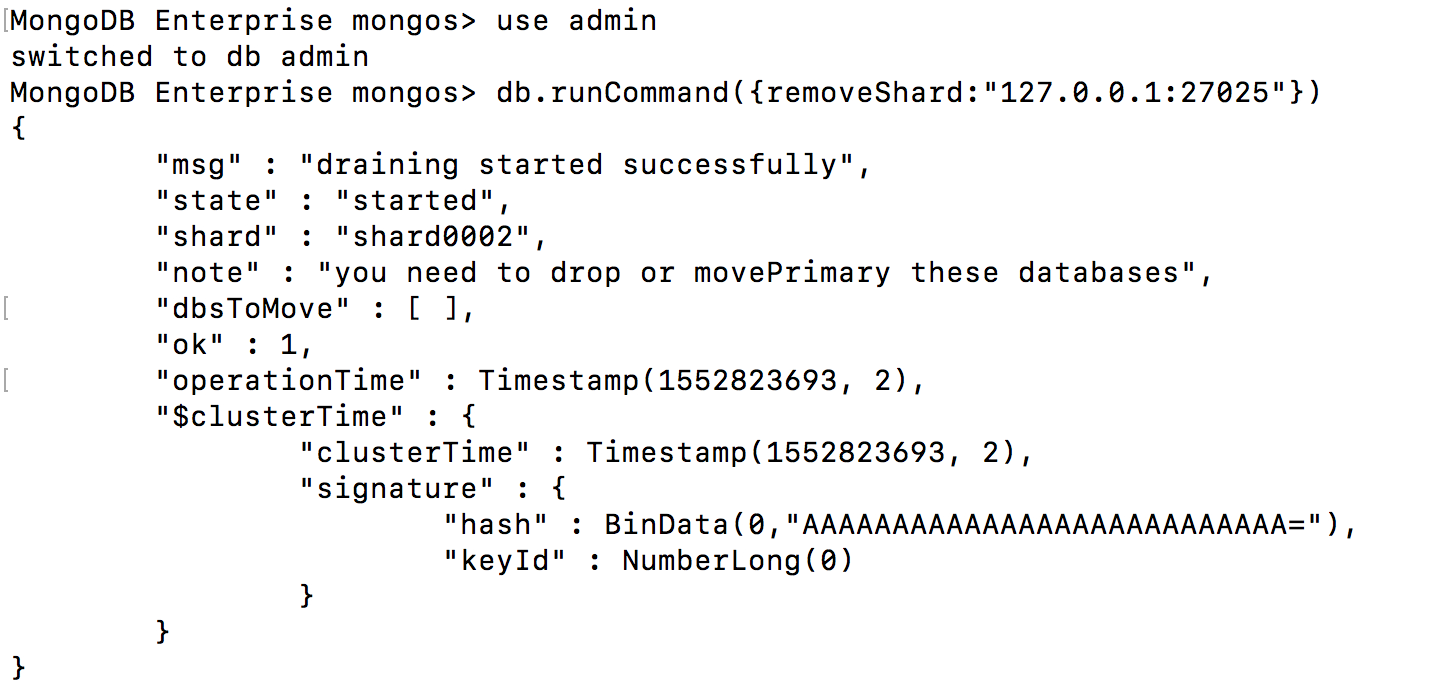
27025: 23831



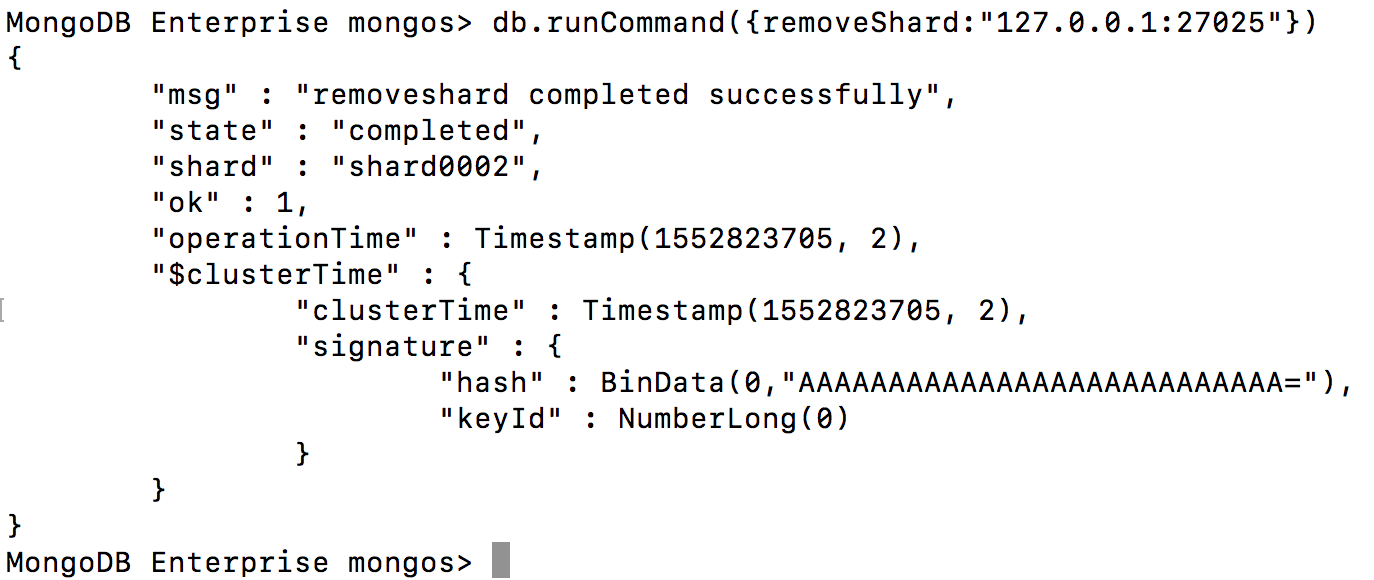
which is 35746+64264+23831=123841

**III. Removing a shard from the cluster**

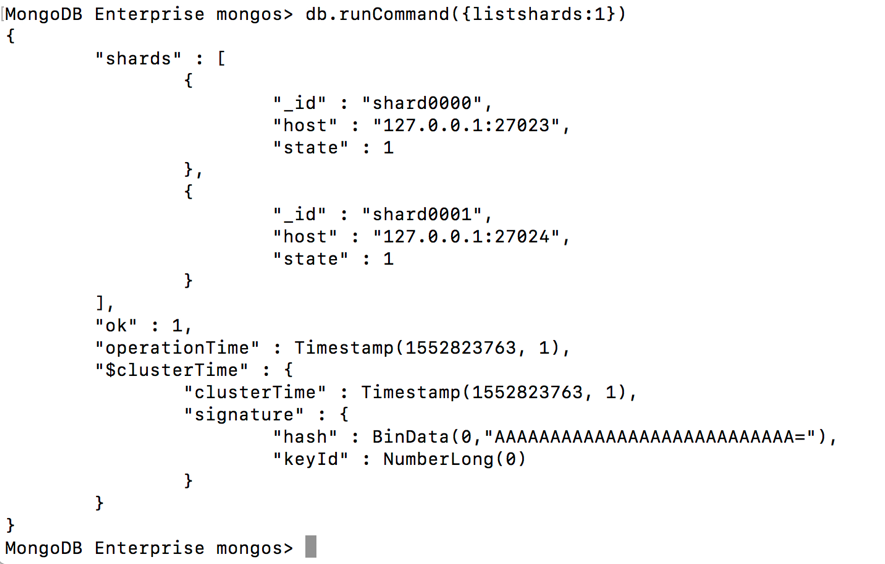
1. **Switch to admin by the command: mongos> use admin**



1. **Then remove the shard with command:** db.runCommand({removeShard:"127.0.0.1:27025"})



1. **Run the remove again**



**To verify that a process is a mongos mongos> db.runCommand({isdbgrid:1})**

