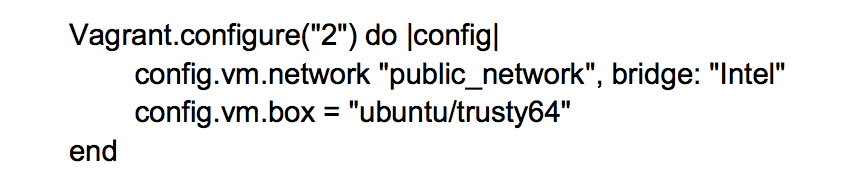
Question2

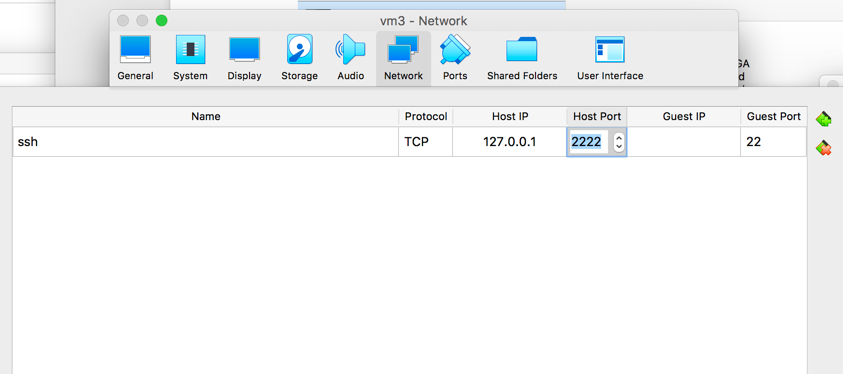
1. Install 3 VM on virtual box with name vm1 vm2 vm3.

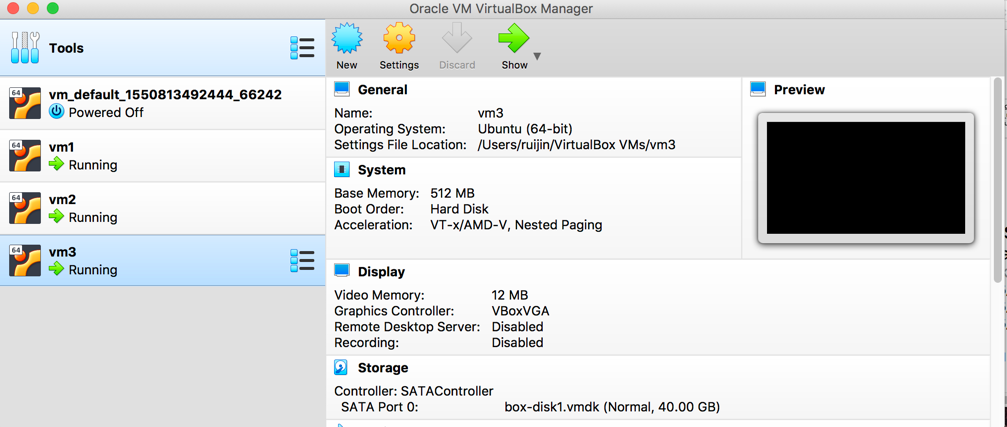
By using vagrant, setup Vagrantfile with configuration.

And vagrant up each VM file install Ubuntu Linux. After installation we get three different VM with three SSH address with port number: 2201, 2200, 2222. (my case)

Also if happens port number collision, we can go to

VirtualBox setting-> NetWork->Advnace->Port Forwarding. And can manually modify the port number to the number we wanted.





2 Open three different terminal window to login each VM.

VM1: ssh -p 2201 vagrant@127.0.0.1

VM2: ssh -p 2200 vagrant@127.0.0.1

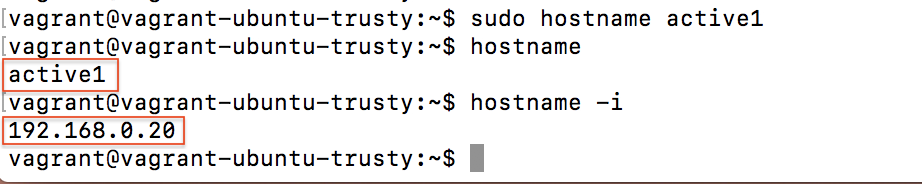
VM3: ssh -p 2222 vagrant@127.0.0.1

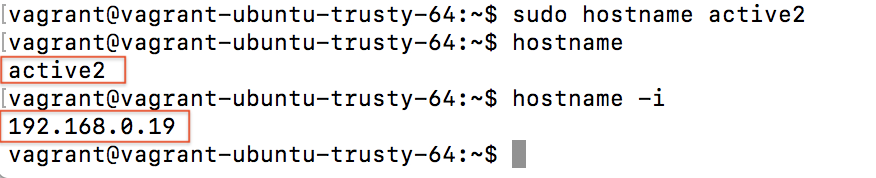
And change hostname in each VM for better identification. Also by command: hostname –i we can get the current network IP address.

VM1: sudo hostname active1 VM1: 192.168.0.20

VM2: sudo hostname active2 VM2: 192.168.0.19

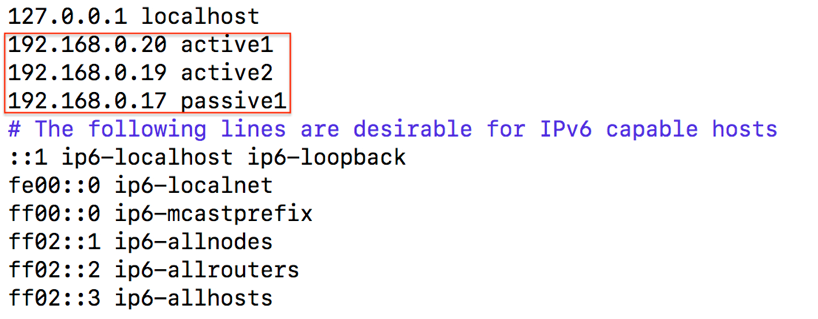
VM3: sudo hostname passive1 VM3: 192.168.0.17





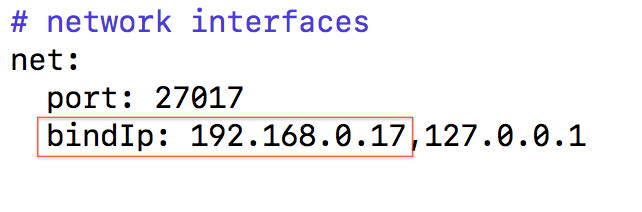


Then check and add each IP address with the name in the ‘hosts’ file at /etc/hosts for all three VM. With command: sudo vim /etc/hosts



Then also modify the /etc/mongo.conf in all three VM with command: sudo vim /etc/mongo.conf

And add the each current ip\_address into the bind\_ip section.

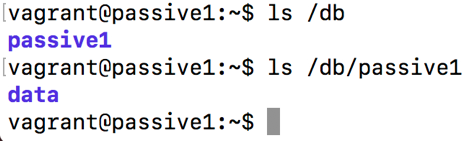
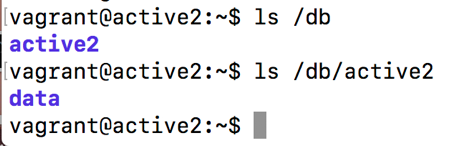
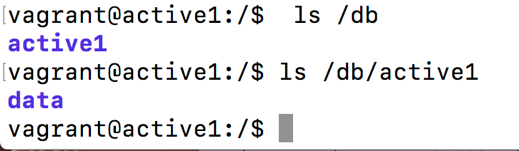


3. in each VM create a directory

VM1: sudo mkdir -p /db/active1/data

VM2: sudo mkdir -p /db/active2/data

VM3: sudo mkdir -p /db/passive1/data



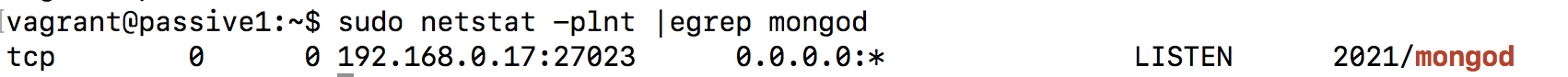
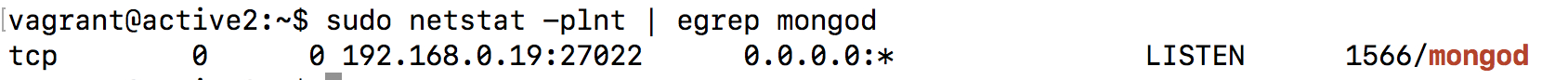
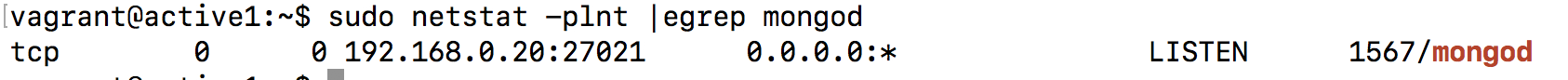
4. Start each mongo instance in active1, active2, passive1 with command:

sudo mongod --port 27021 -dbpath /db/active1/data --replSet testset --smallfiles --oplogSize 128 --bind\_ip 192.168.0.20

sudo mongod --port 27022 -dbpath /db/active2/data --replSet testset --smallfiles --oplogSize 128 --bind\_ip 192.168.0.19

sudo mongod --port 27023 -dbpath /db/passive1/data --replSet testset --smallfiles --oplogSize 128 –bind\_ip 192.168.0.17

Then in each instance, active1 opened with port :27021, active2 with port:27022, passive1 with port:27023.



5 Connect the the active1(primary) instance with command: mongo 192.168.0.20:27021

And initiates a replica set with command: rs.initiate() by check the status rs.status() can see the current condition. That as graphic show below, the active1 has \_id =0 as the primary.



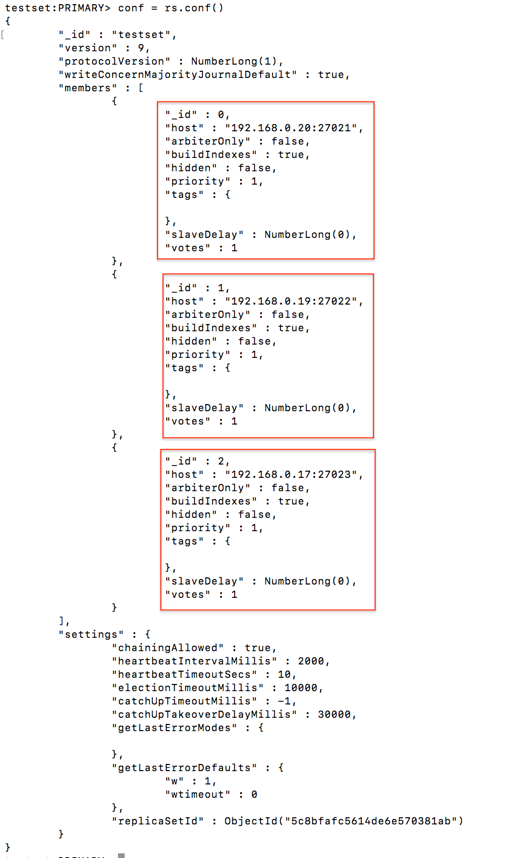
6. Add the active2 and passive1 to the replica set with command:

rs.add(192.168.0.19:27022") rs.add("192.168.0.17:27023")

By checking the status with rs.status(), the graph shows active2 and passive1 has been added into the set.



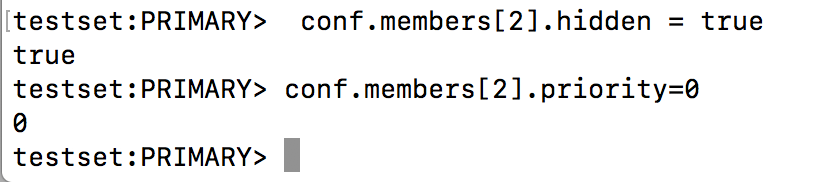
7. With assign conf =rs.conf()



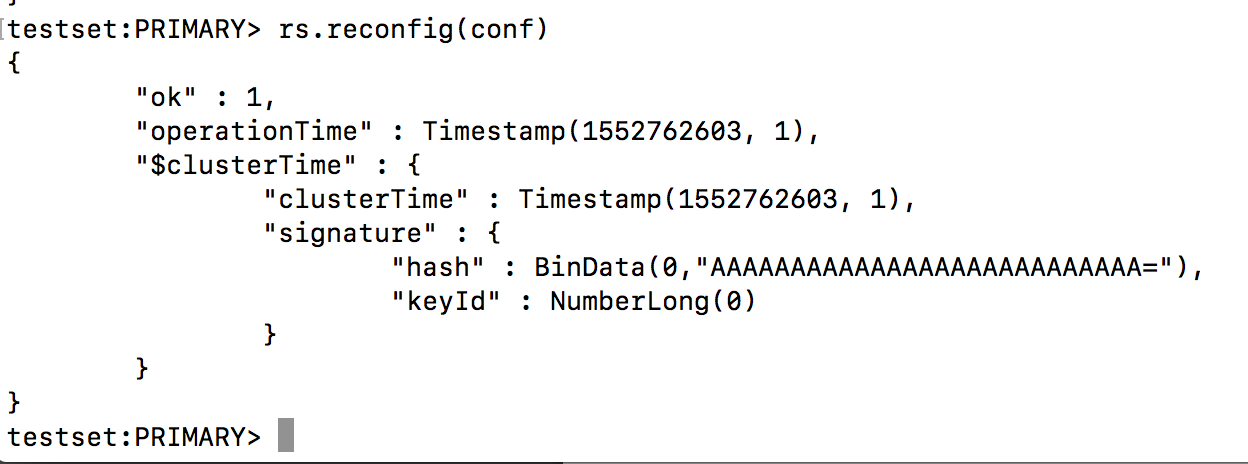
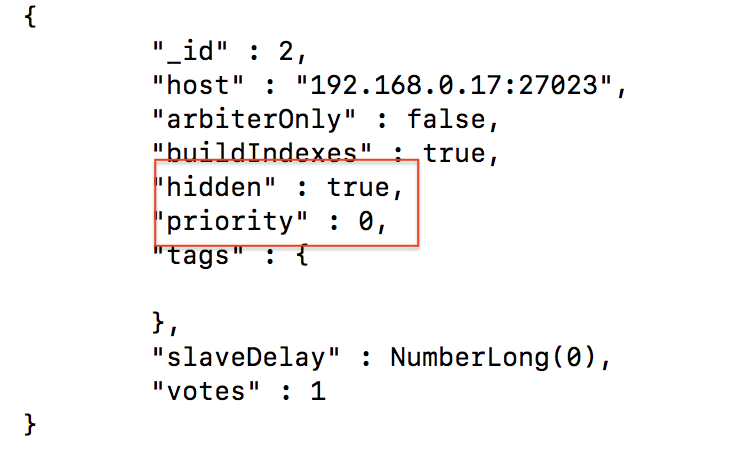
8. Change the passive1’s status with command:

conf.members[2].hiddne = true

conf.members[2].priority=0



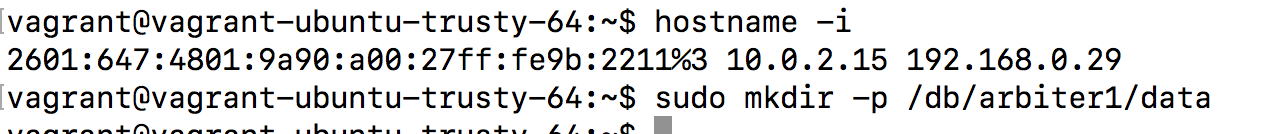
9. then check the changed conf condition by typing conf and then reconfigure the existed status with rs.reconfig(conf)



10. Add arbiter

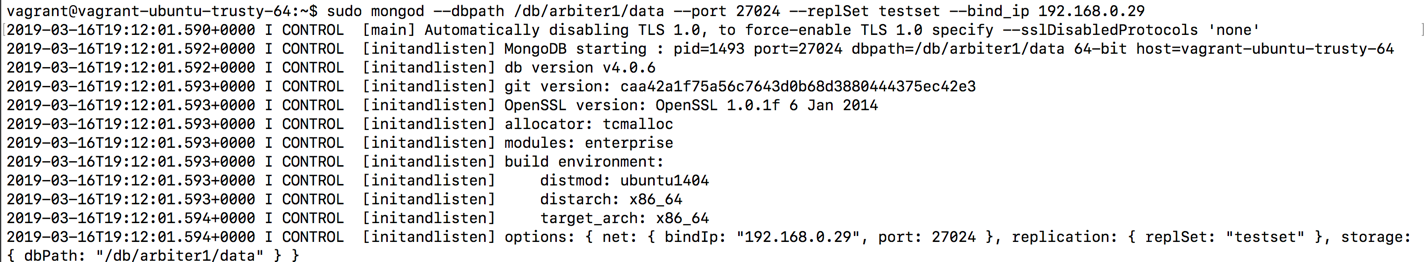
opened a 4th VM with port number 2019. By checking it’s ip hostname –i, The ip address of the

vm4 is 192.168.0.29. And make a directory for arbiter1



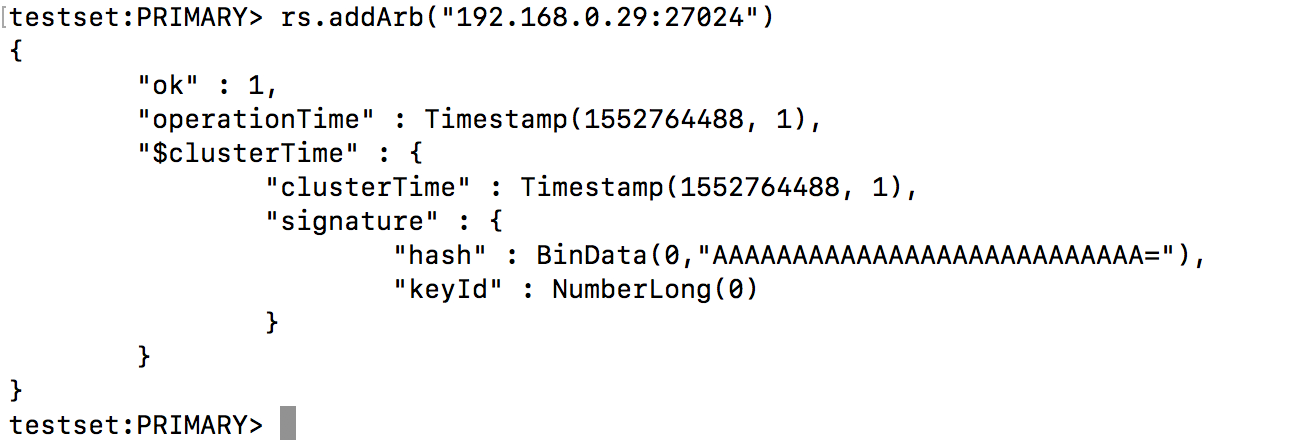
Then connect to the instance(aribiter) with command:

sudo mongod --dbpath /db/arbiter1/data --port 27024 --replSet testset --bind\_ip 192.168.0.29



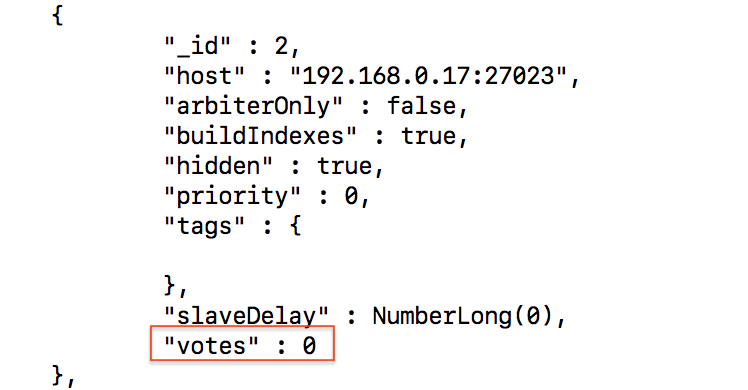
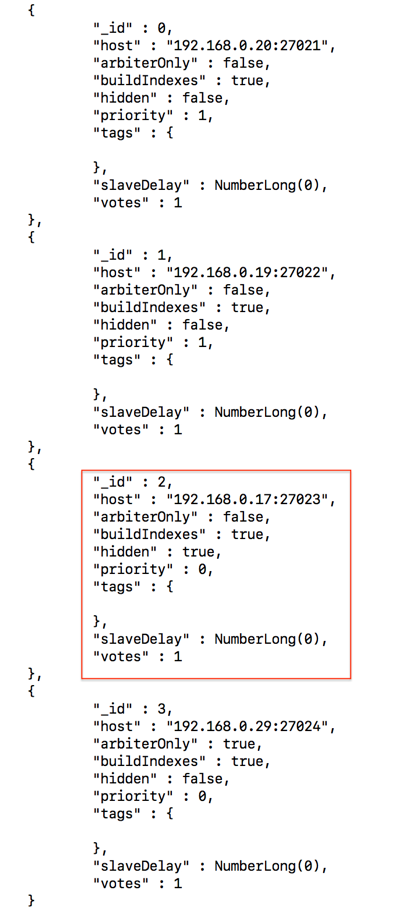
11. Back to testset:PRIMARY (active1) Add arbiter into the set with command:

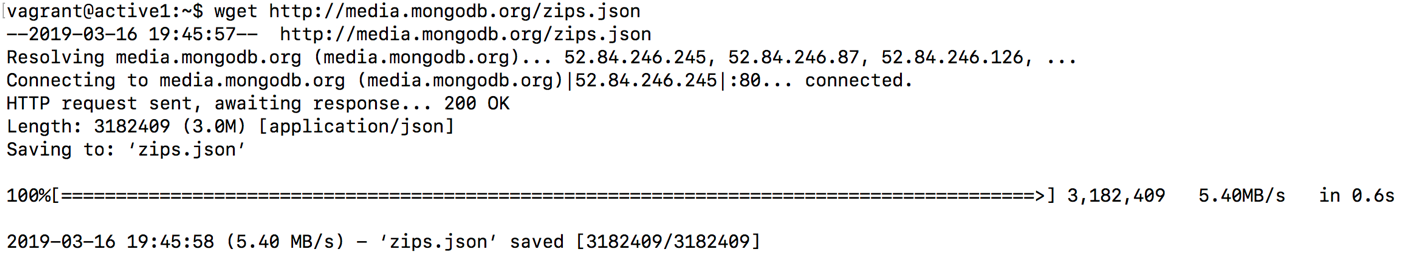
rs.addArb("192.168.0.29:27024") by checking current status with rs.status()



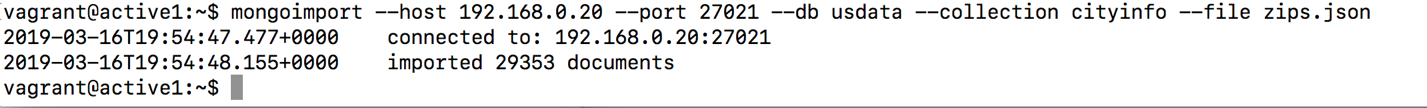
12. After adding arbiter. The three member replica are now even number. So change the status of passive1 to votes =0 will solve the problem. With the command :

conf=rs.conf() => conf.members[2].votes =0 => rs.reconfig(conf)

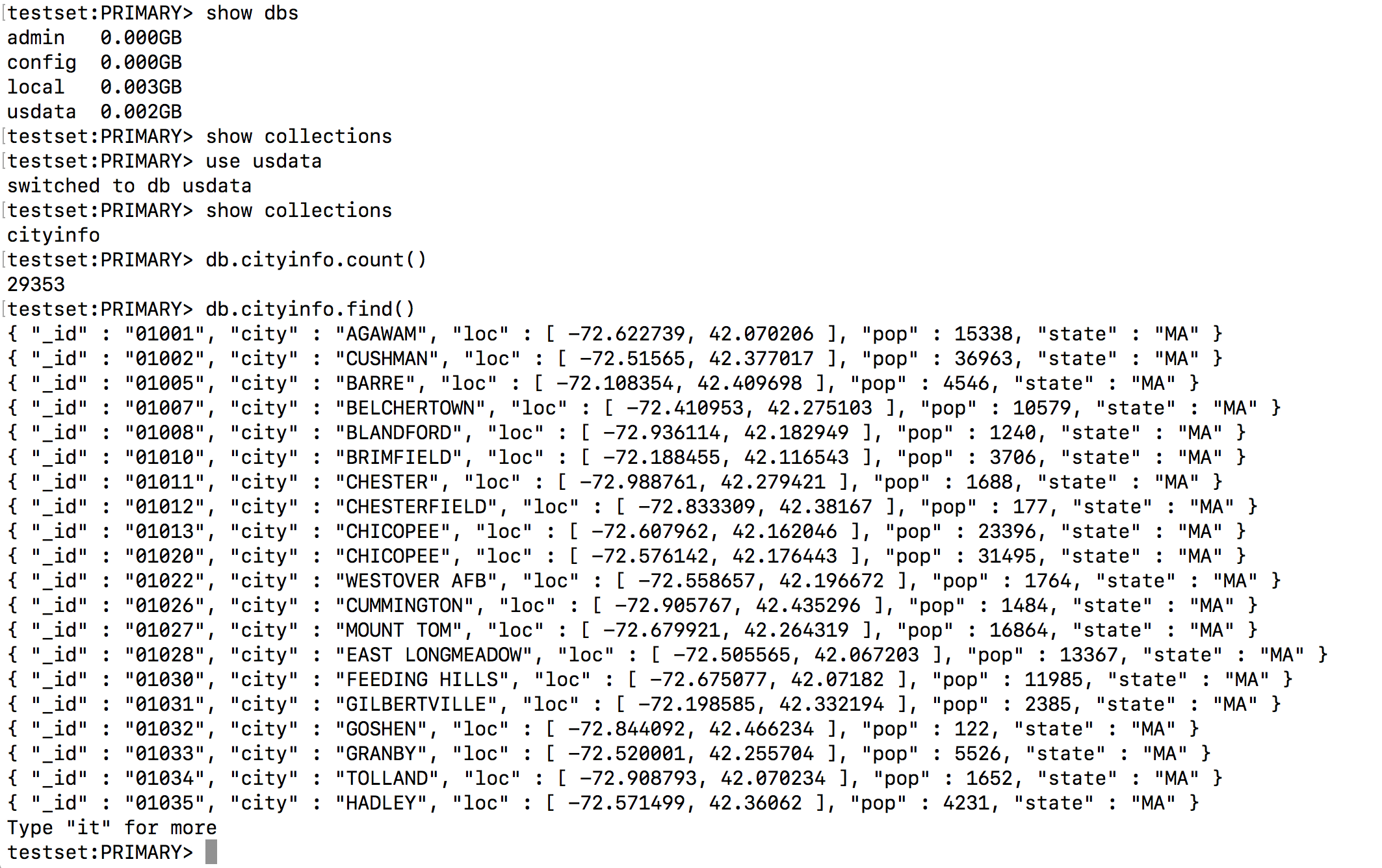


13. populate data into active1 with command : wget 

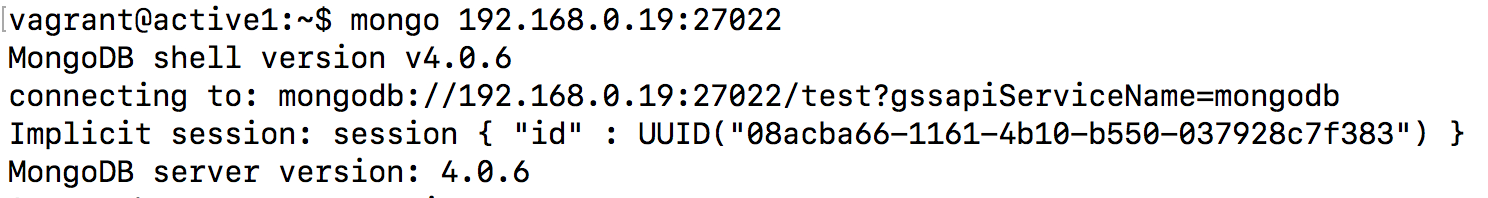
Then import data into database with command:

vagrant@active1:~$ mongoimport --host 192.168.0.20 --port 27021 --db usdata --collection cityinfo --file zips.json

Then enter into active1 (PRIMARY) check data in database: usdata



14. Enter secondary instance (active2) with command: vagrant@active1:~$ mongo 192.168.0.19:27022



15. checking the secondary database and also use rs.slaveOk() to allows the active2 the current connection to asllow read operations to run on the secondary member. By comparing both data, the data has successfully duplicated from active1(PRIMARY) to the active2(secondary).

