Cloud Computing – Assignment 2

(Docker MongoDB Installation)

Group Members: Saurav Nanda and Mengjie Wang

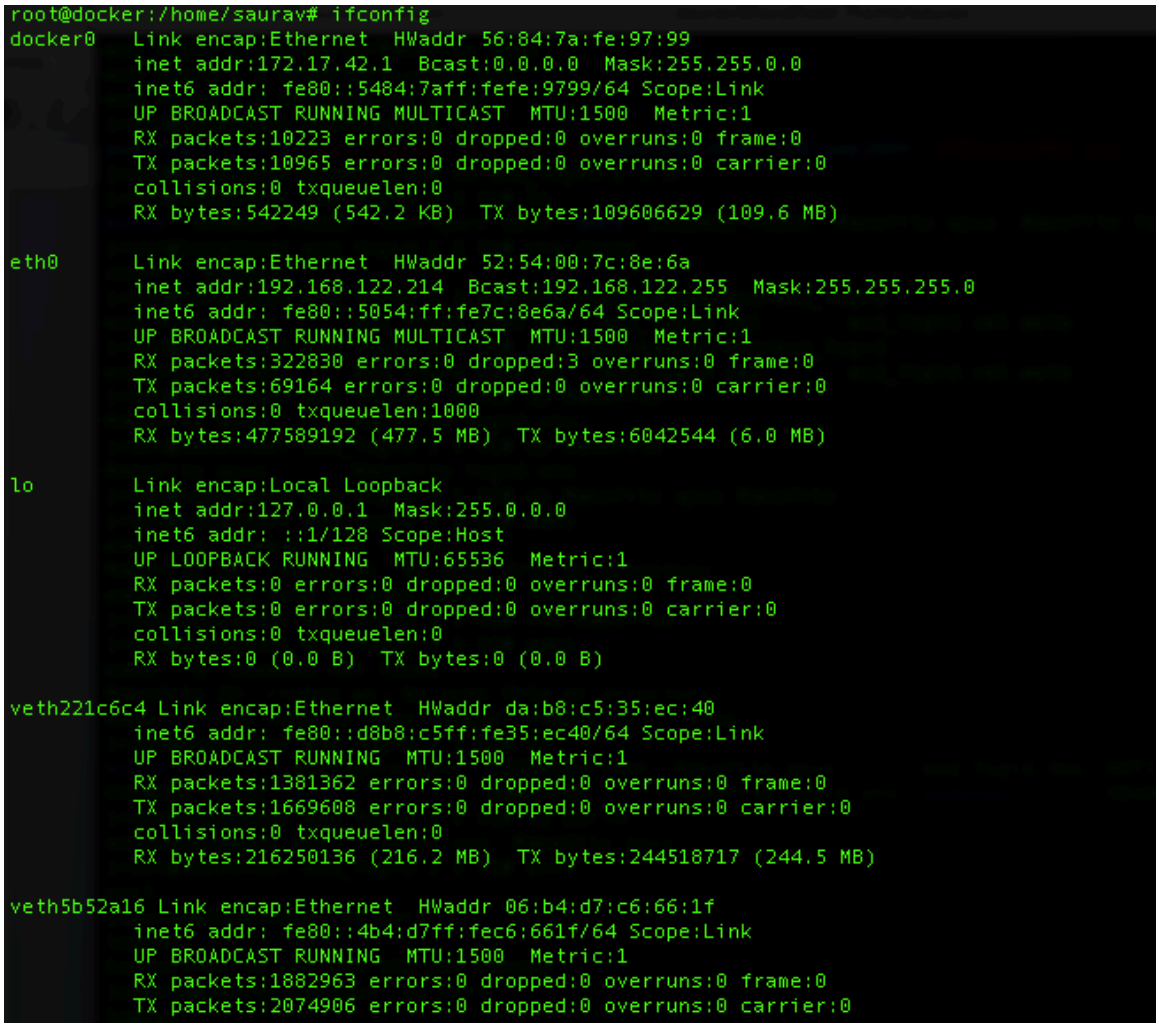
Report

VM Configurations

---

1. **Docker VM**: 2556 MB RAM, 1 CPU, 60GB Storage and Ubuntu OS

Network configuration:



2. **Install Docker**

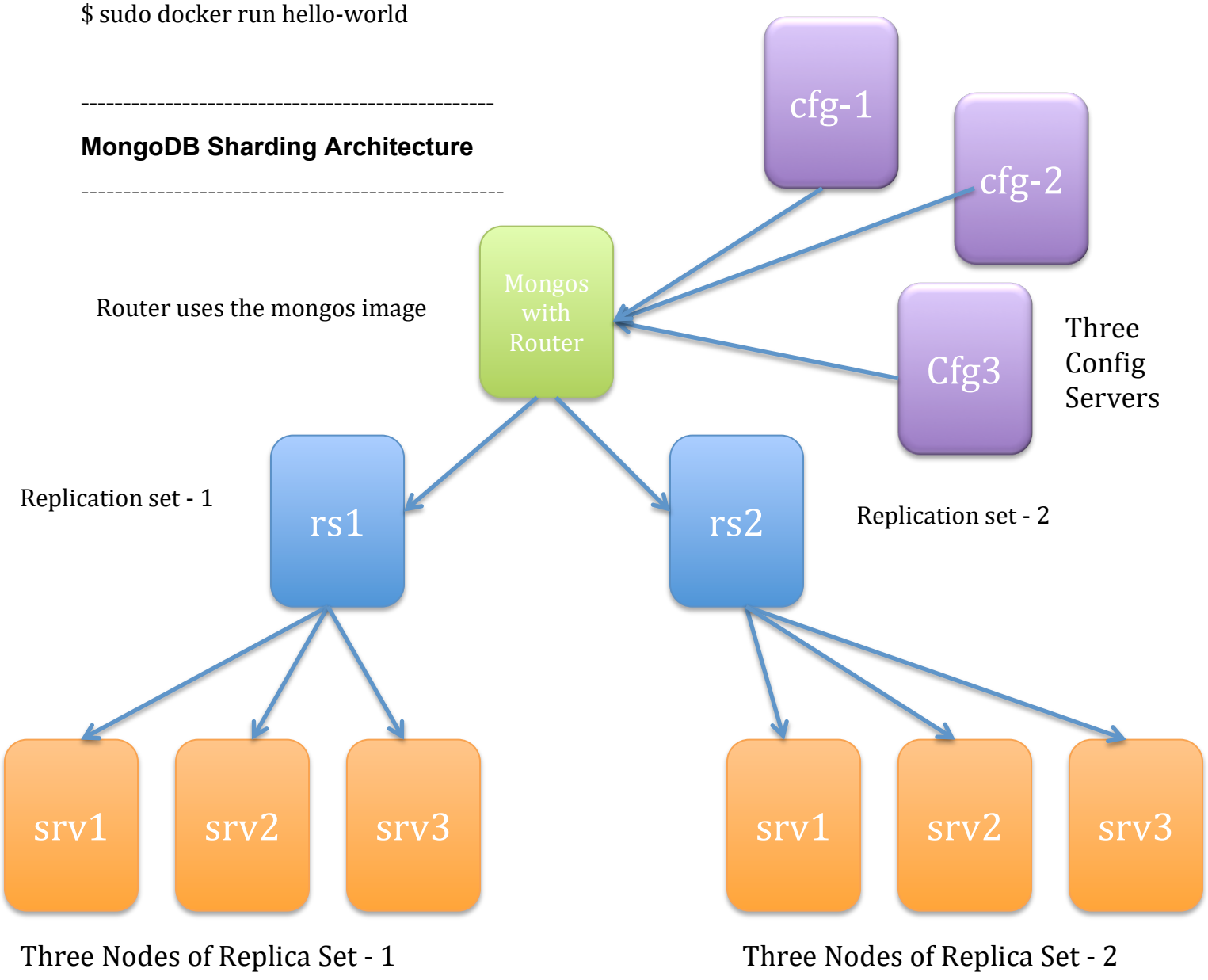
Get the latest Docker package.

$ wget -qO- https://get.docker.com/ | sh

Verify docker is installed correctly.

$ sudo docker run hello-world

MongoDB Sharding Architecture



Docker Configurations

---

1. **Create the Docker Files**

We created two Dockerfiles: one for mongod and another one for mongos.

- Root Folder: /home/saurav/assgn2
- Create 2 more Folders: “mongod" & “mongos”
  $mkdir /home/saurav/assgn2/mongod
  $mkdir /home/saurav/assgn2/mongos

```
root@docker:/home/saurav# ls
assgn2
root@docker:/home/saurav# pwd
/home/saurav
root@docker:/home/saurav# ls assgn2/
mongod  mongos
```

- Create separate Dockerfiles for mongos and mogod respectively.
- Docker file for mongod:

```
FROM ubuntu:latest

# Add 10gen official apt source to the sources list
RUN apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
RUN echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | tee /etc/apt/sources.list.d/10gen.list

# Install MongoDB
RUN apt-get update
RUN apt-get install mongodb-10gen

# Create the MongoDB data directory
RUN mkdir -p /data/db

EXPOSE 27017
ENTRYPOINT ["usr/bin/mongod"]
~
~
```

- Docker file for mongos:

```
FROM saurav/mongodb:latest

EXPOSE 27017
ENTRYPOINT ["usr/bin/mongos"]
~
```

2. **Build Docker Files**

- Go to mongod folder and execute this command:

$sudo docker build \ -t saurav/mongodb mongod

- Go to mongos folder and execute this command:

  $sudo docker build \ -t saurav/mongos mongos

3. **Create the Replica Sets**

- First Replica Set – rs1

  $sudo docker run \ -P -name rs1_srv1 \ -d saurav/mongodb \ --replSet rs1 \ --noprealloc --smallfiles

  $sudo docker run \ -P -name rs1_srv2 \ -d saurav/mongodb \ --replSet rs1 \ --noprealloc --smallfiles

  $sudo docker run \ -P -name rs1_srv3 \ -d saurav/mongodb \ --replSet rs1 \ --noprealloc --smallfiles

- Second Replica Set – rs2

  $sudo docker run \ -P -name rs2_srv1 \ -d saurav/mongodb \ --replSet rs2 \ --noprealloc --smallfiles

  $sudo docker run \ -P -name rs2_srv2 \ -d saurav/mongodb \ --replSet rs2 \ --noprealloc --smallfiles

  $sudo docker run \ -P -name rs2_srv3 \ -d saurav/mongodb \ --replSet rs2 \ --noprealloc –smallfiles

4. **Intialize the Replica Sets**

- Note the IP addresses of the containers:

  $sudo docker inspect rs1_srv1

  $sudo docker inspect rs1_srv2

  $sudo docker inspect rs1_srv3

  $sudo docker inspect rs2_srv1

  $sudo docker inspect rs2_srv2

  $sudo docker inspect rs2_srv3

  Sample output:

IP Addresses of our containers:

rs1_srv1 - 172.17.0.8

rs1_srv2 - 172.17.0.9

rs1_srv3 - 172.17.0.10

rs2_srv1 - 172.17.0.11

rs2_srv2 - 172.17.0.12

rs2_srv3 - 172.17.0.13

- Initialize Replication Set1

Connect to MongoDB running in container rs1_srv1

a) First enter the bash of the contain

$sudo docker exec -i -t <container-id> bash

b) Enter the MongoDB

$mongo

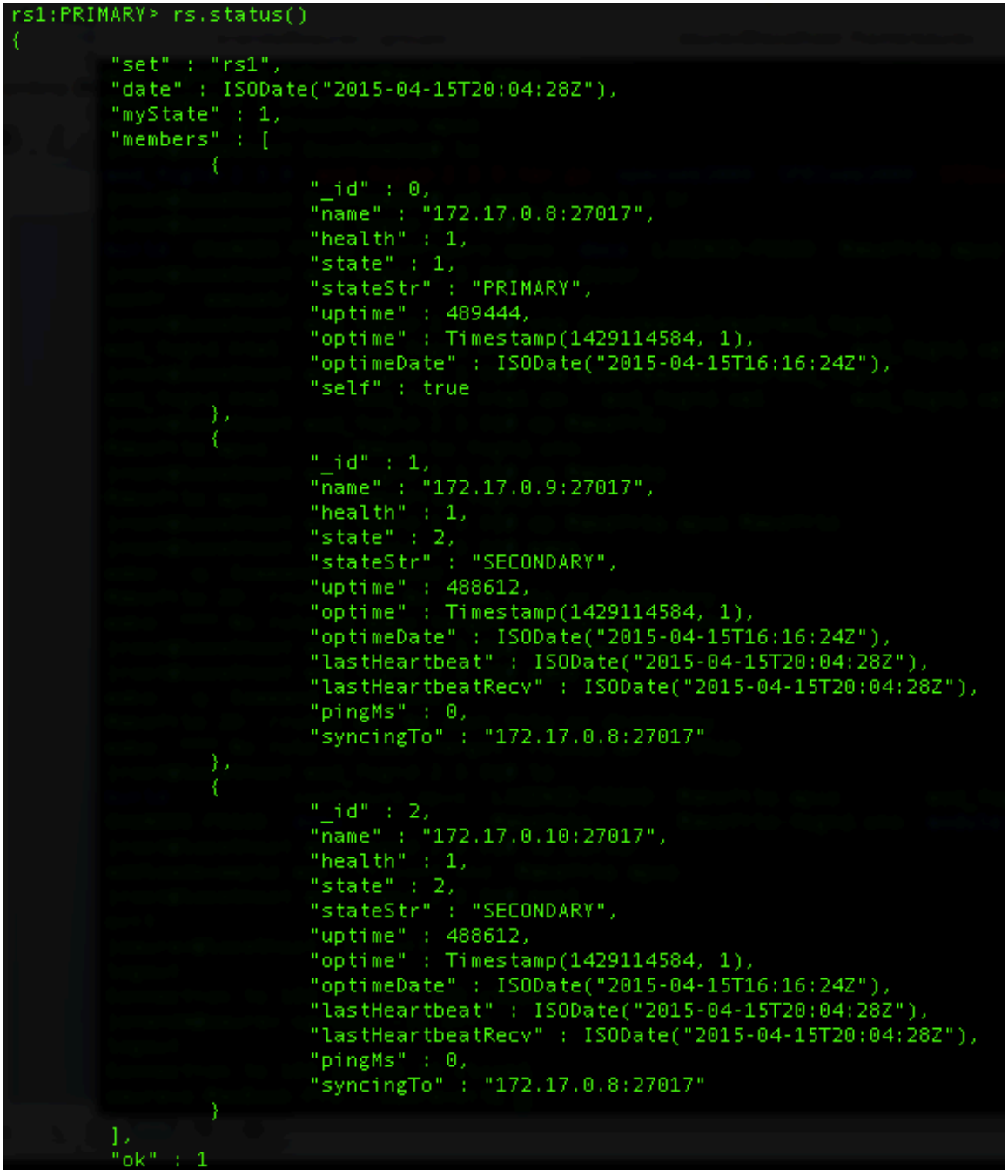
c) Once you enter MongoDB shell , execute these commands:

rs.initiate()

rs.add("<IP_of_rs1_srv2>:27017")

rs.add("<IP_of_rs1_srv3>:27017")

rs.status()



- Initialize Replication Set2

  Connect to MongoDB running in container rs2_srv1

  a) First enter the bash of the contain

  $sudo docker exec -i -t <container-id> bash

  b) Enter the MongoDB

  $mongo

  c) Once you enter MongoDB shell , execute these commands:

  rs.initiate()

  rs.add("<IP_of_rs2_srv2>:27017")

  rs.add("<IP_of_rs2_srv3>:27017")

  rs.status()

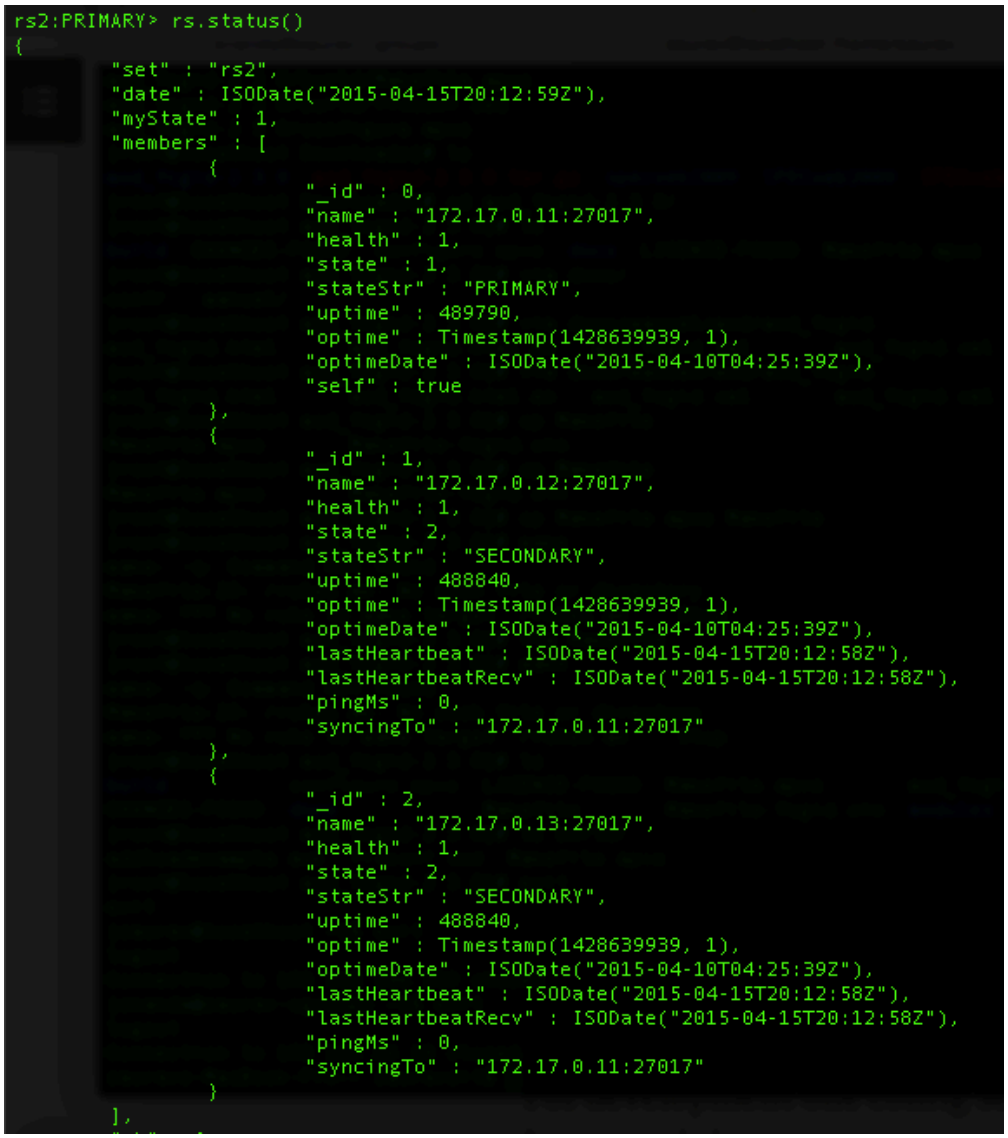
  d) Change the hostname to the IP address.

  cfg = rs.conf()

  cfg.members[0].host = "<IP_of_rs2_srv1>:27017"

  rs.reconfig(cfg)

  rs.status()



5. **Create Config Servers**

   $sudo docker run \ -P -name cfg1 \ -d saurav/mongodb \ --noprealloc --smallfiles \ --configsvr \ --dbpath /data/db \ --port 27017

   $sudo docker run \ -P -name cfg2 \ -d saurav/mongodb \ --noprealloc --smallfiles \ --configsvr \ --dbpath /data/db \ --port 27017

   $sudo docker run \ -P -name cfg3 \ -d saurav/mongodb \ --noprealloc --smallfiles \ --configsvr \ --dbpath /data/db \ --port 27017

IP Addresses of the Config Servers:

cfg1 - 172.17.0.14

cfg2 - 172.17.0.15

cfg3 - 172.17.0.16

6. **Create Router**
   - Here we are using the mongos image created earlier:

$sudo docker run \ -P -name mongos1 \ -d saurav/mongos \ --port 27017 \ --configdb \ <IP_of_container_cfg1>:27017, \ <IP_of_container_cfg2>:27017, \ <IP_of_container_cfg3>:27017

   - IP Address of Router: "mongos3" - 172.17.0.21
7. **Initialize the Shard**
   - Login to MongoDB router and execute the below commands:

     $sudo docker exec -i -t <container_id> bash

   - Enter MongoDB and execute below commands:

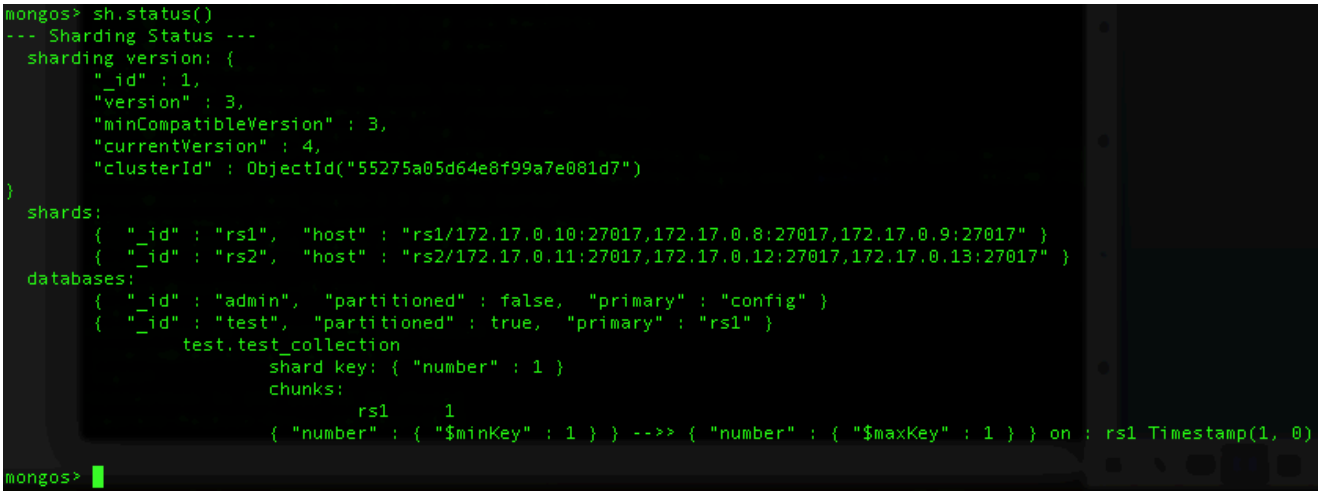
     $mongo

   - Execute below MongoDB queries:

     sh.addShard("rs1/<IP_of_rs1_srv1>:27017")

     sh.addShard("rs2/<IP_of_rs2_srv1>:27017")

     sh.status()



APPENDIX A – Dockerfile (mongod)

```
FROM ubuntu:latest

# Add 10gen official apt source to the sources list
RUN apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
RUN echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' |
tee /etc/apt/sources.list.d/10gen.list

# Install MongoDB
RUN apt-get update
RUN apt-get install mongodb-10gen

# Create the MongoDB data directory
RUN mkdir -p /data/db

EXPOSE 27017
ENTRYPOINT ["usr/bin/mongod"]
```

APPENDIX B – Dockerfile (mongos)

FROM saurav/mongodb:latest

EXPOSE 27017

ENTRYPOINT ["usr/bin/mongos"]

APPENDIX C – LIST OF RUNNING CONTAINERS

```
root@docker:/home/saurav# docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS                     NAMES
473b2de03e6d        saurav/mongos:latest   "usr/bin/mongos --po   6 days ago          Up 6 days           0.0.0.0:49166->27017/tcp  mongos3
93f9df413205        saurav/mongodb:latest  "usr/bin/mongod --no   6 days ago          Up 6 days           0.0.0.0:49161->27017/tcp  cfg3
1139f54d55ab        saurav/mongodb:latest  "usr/bin/mongod --no   6 days ago          Up 6 days           0.0.0.0:49160->27017/tcp  cfg2
655d56f1fa0e        saurav/mongodb:latest  "usr/bin/mongod --no   6 days ago          Up 6 days           0.0.0.0:49159->27017/tcp  cfg1
1b1ed97d78e6        saurav/mongodb:latest  "usr/bin/mongod --re   6 days ago          Up 6 days           0.0.0.0:49158->27017/tcp  rs2_srv3
a19e7f22f885        saurav/mongodb:latest  "usr/bin/mongod --re   6 days ago          Up 6 days           0.0.0.0:49157->27017/tcp  rs2_srv2
2853205d3b87        saurav/mongodb:latest  "usr/bin/mongod --re   6 days ago          Up 6 days           0.0.0.0:49156->27017/tcp  rs2_srv1
8d1e98e35cb4        saurav/mongodb:latest  "usr/bin/mongod --re   6 days ago          Up 6 days           0.0.0.0:49155->27017/tcp  rs1_srv3
33ebbefc1645        saurav/mongodb:latest  "usr/bin/mongod --re   6 days ago          Up 6 days           0.0.0.0:49154->27017/tcp  rs1_srv2
01c16f37ae62        saurav/mongodb:latest  "usr/bin/mongod --re   6 days ago          Up 6 days           0.0.0.0:49153->27017/tcp  rs1_srv1
root@docker:/home/saurav#
```