PRACTICAL5

Aim - Implementation of Replicas in MongoDB.

Replication:-

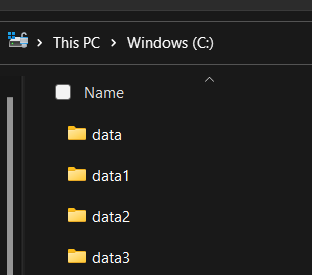
Replication is the process of synchronizing data across multiple servers. Replication provides redundancy and increases data availability with multiple copies of data on different database servers. Replication protects a database from the loss of a single server. Replication also allows you to recover from hardware failure and service interruptions. With additional copies of the data, you can dedicate one to disaster recovery, reporting, or backup.

STEP-1

Shutdown already running MongoDB server.

STEP-2

Create 3 folders named data1,data2,data3 at any location.Over here we have created in C drive.



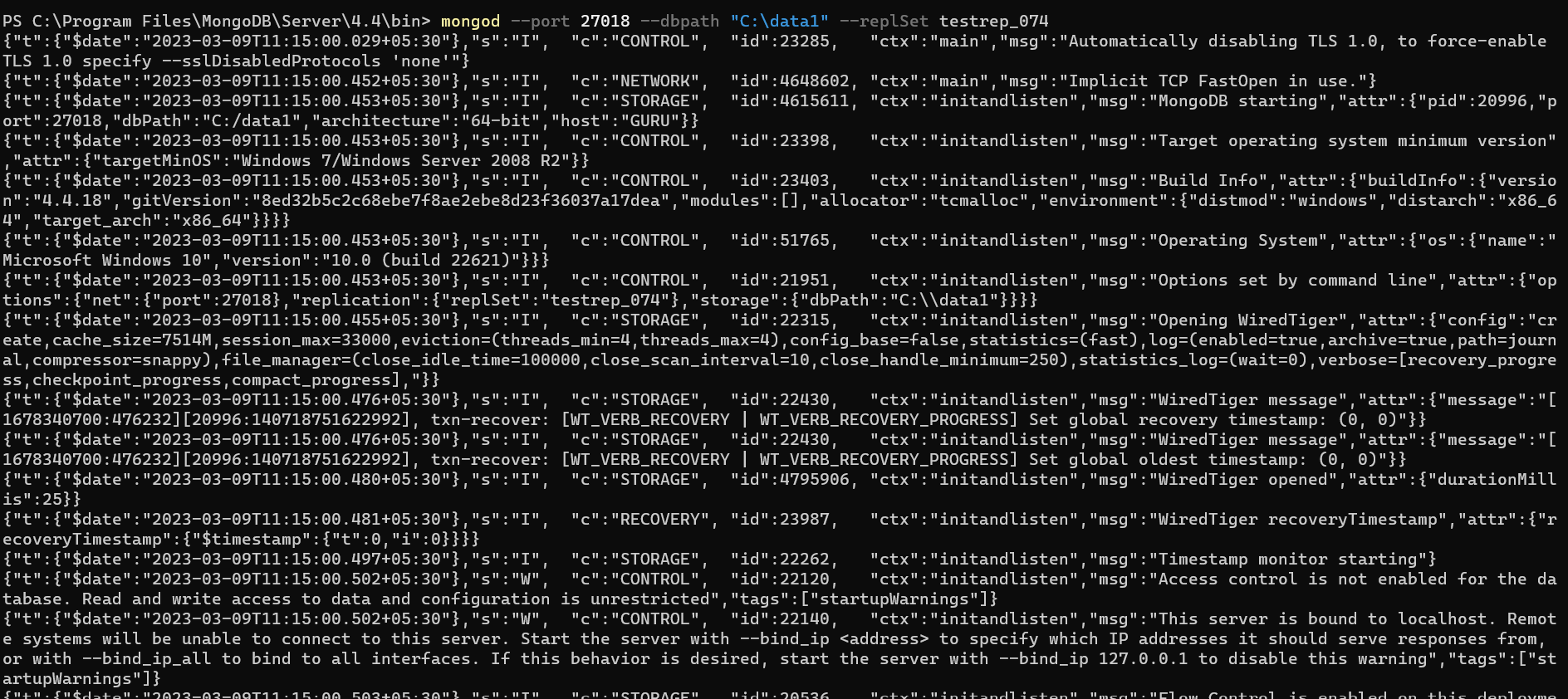
STEP-3(Creating Primary instance)

SYNTAX:

mongod --port "PORT" --dbpath "YOUR\_DB\_DATA\_PATH" --replSet "REPLICA\_SET\_INSTANCE\_NAME"

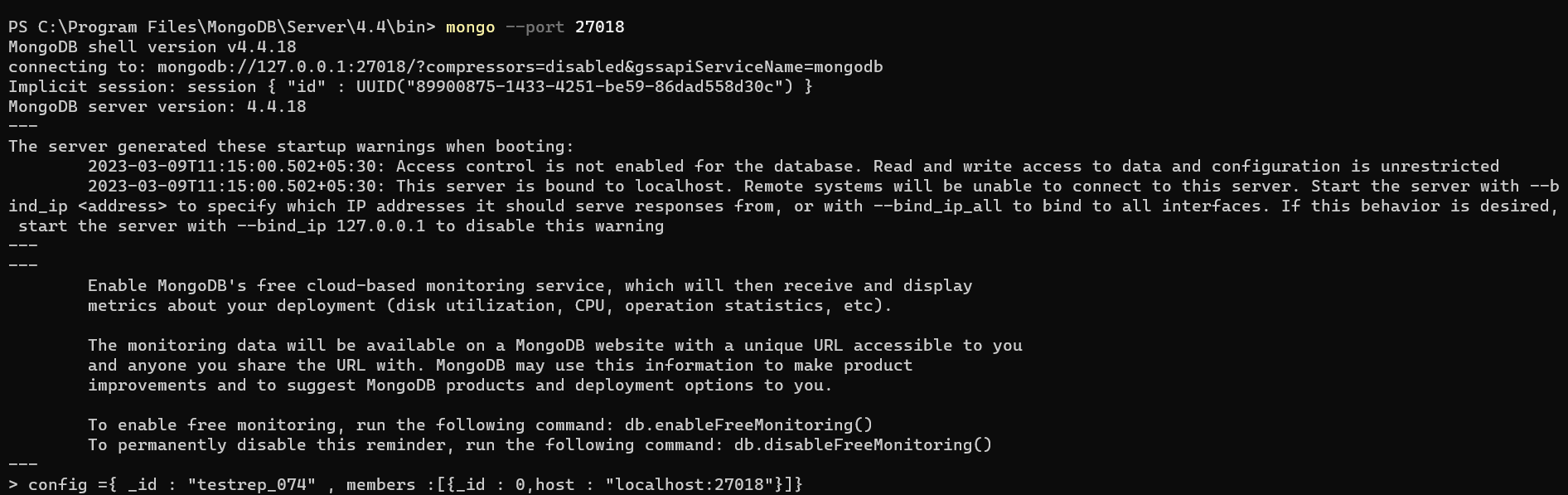
COMMAND:

mongod --port 27018 --dbpath "C:\data1" --replSet testrep\_074



Now Open another command prompt for client. We will use this window to query our first server instance.

mongo –port 27018

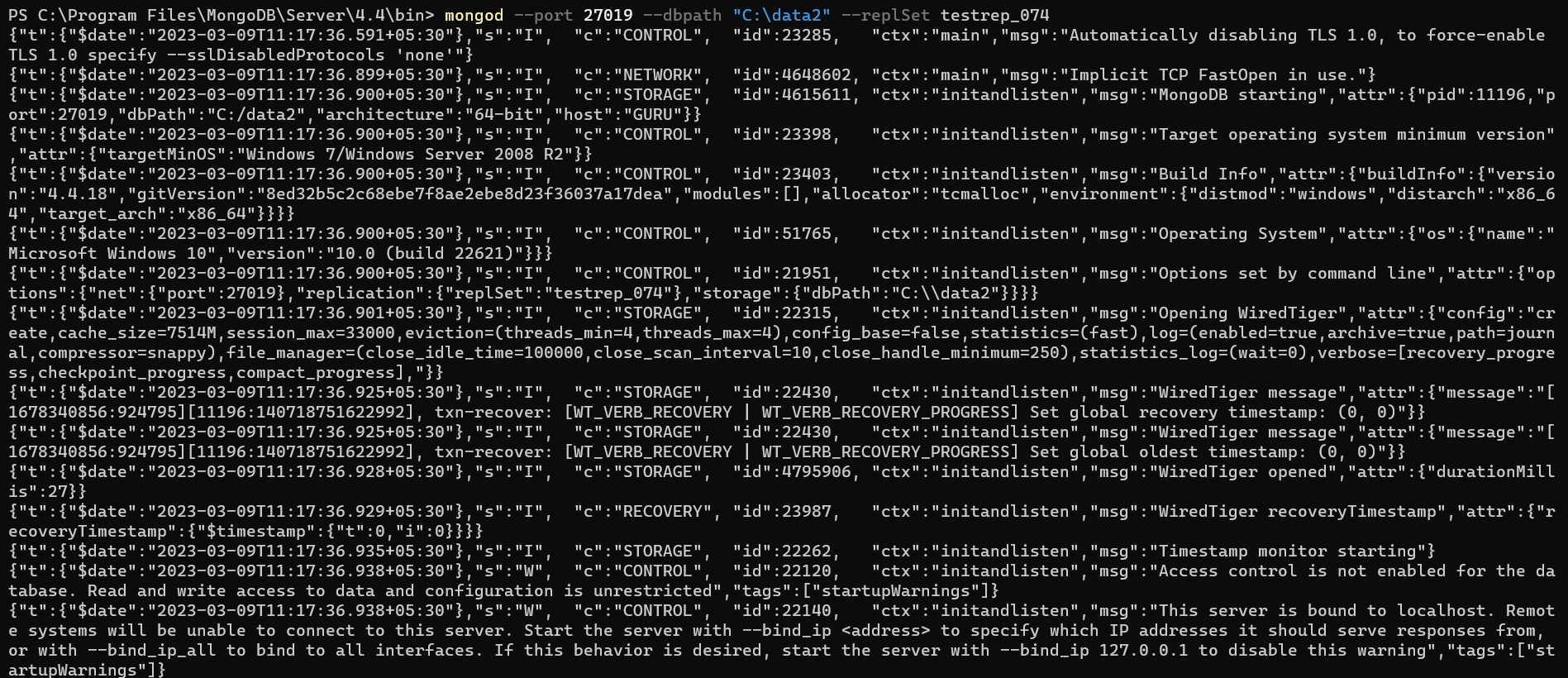


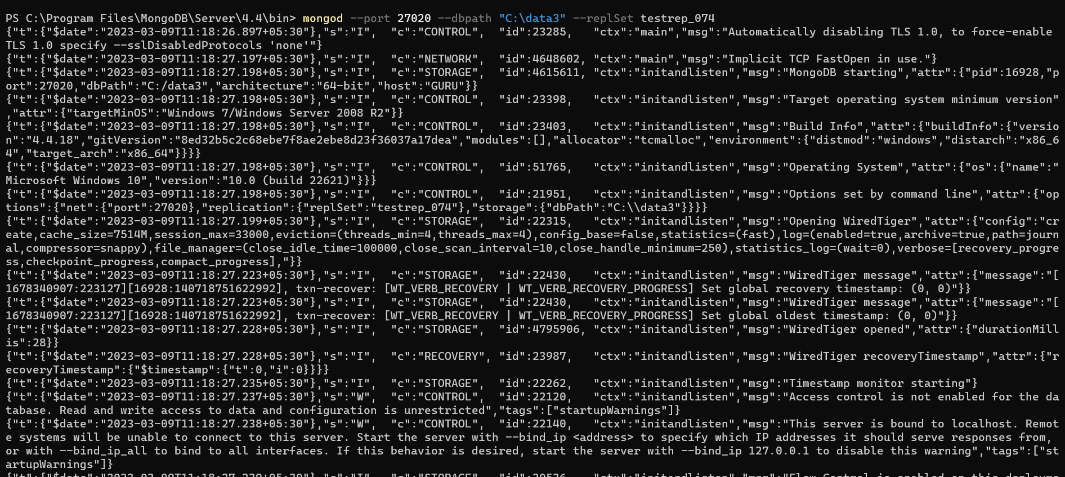
Now primary server is running

STEP4(Creating 2 Replicas of Primary instance)

mongod --port 27019 --dbpath "C:\data2" --replSet testrep\_074

mongod --port 27020 --dbpath "C:\data3" --replSet testrep\_074

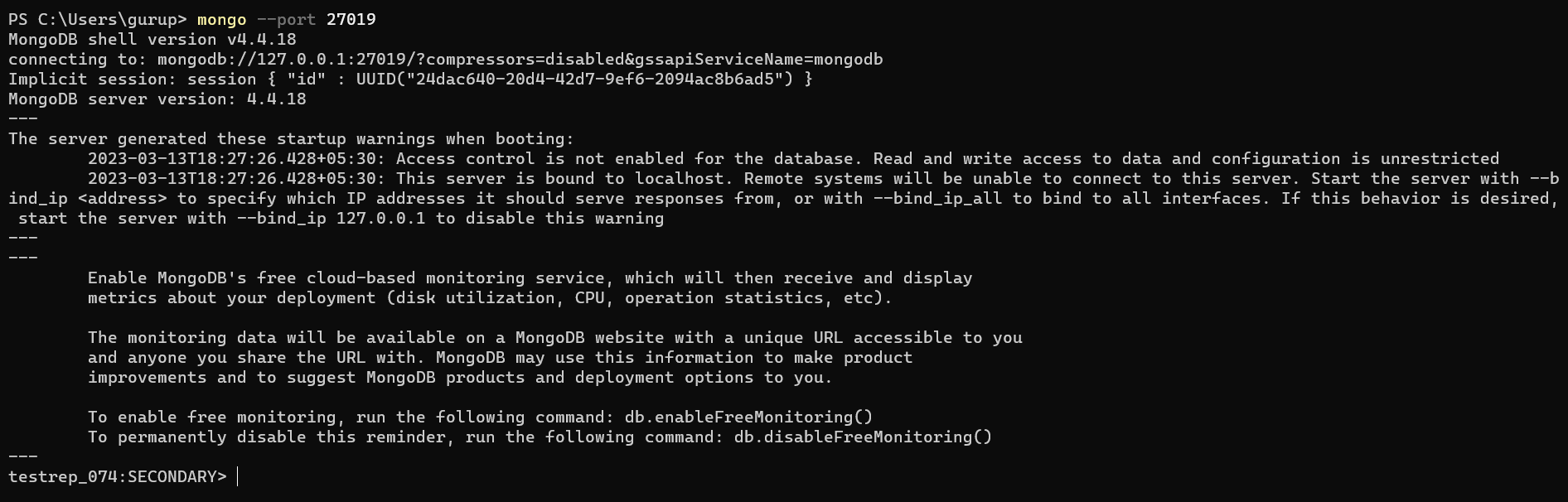


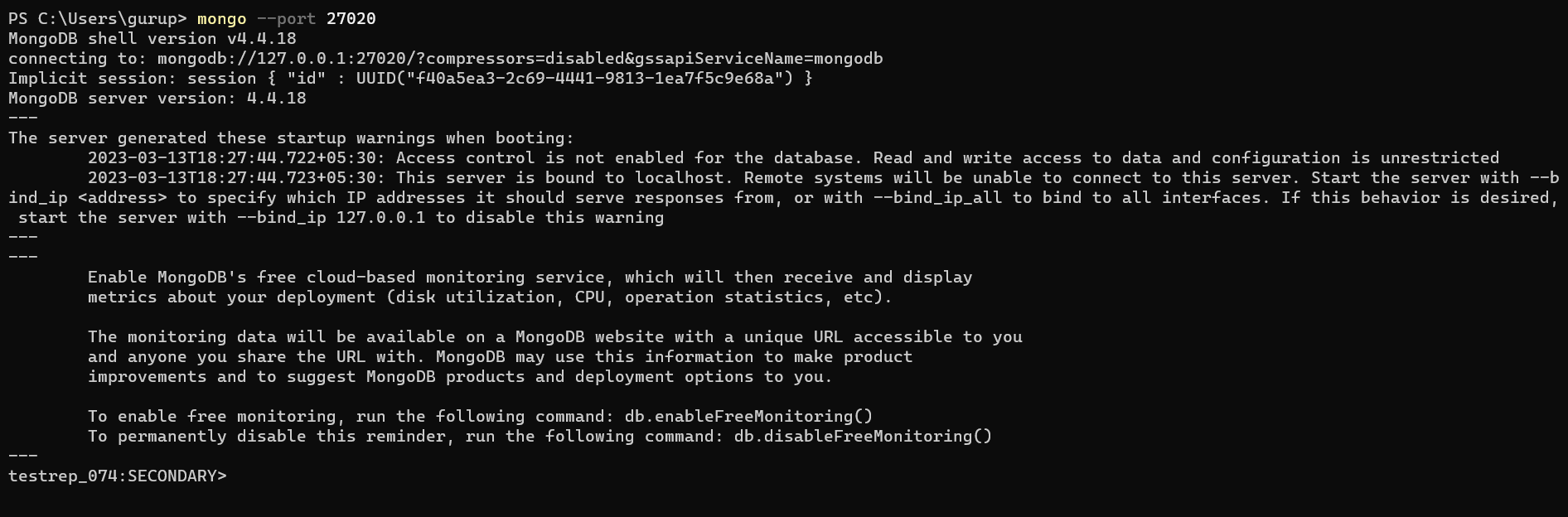


Now Open another command prompt for client. We will use this window to query our second server instance

mongo --port 27019

mongo –port 27020



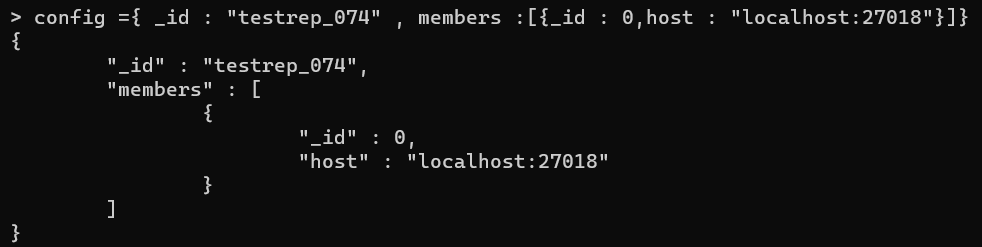


Step 5: Now go to the command prompt of Primary server’s Client instance.

C:\>mongo --port 27018

Now type the following code :

config ={ \_id : "testrep\_074" , members :[ { \_id : 0, host : "localhost:27018" } ] }



After this write command rs.initiate(config) This command initiates a replica set with the current host as its only member. This is confirmed by the output, which should resemble the following:



After this write command :

rs.status()

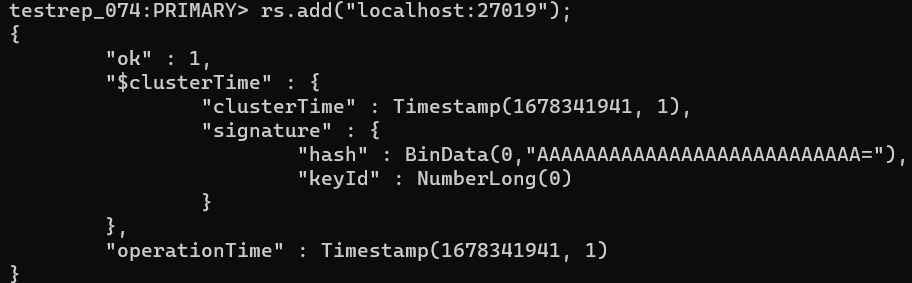


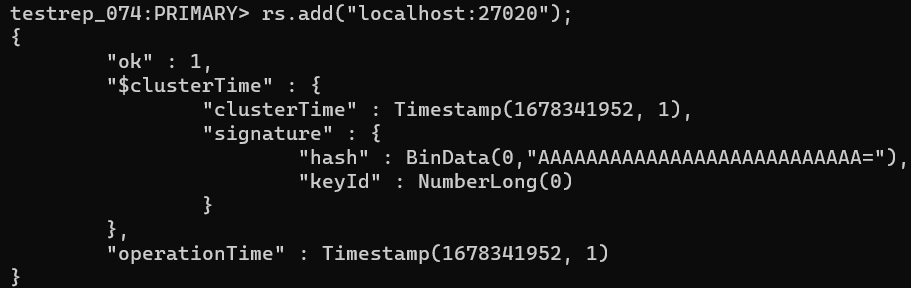
**Step : 6 Now go to the cmd client window of primary instance.**

**Write the following code there :-**

rs.add("localhost:27019");

rs.add("localhost:27020");

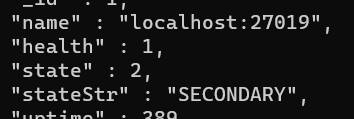
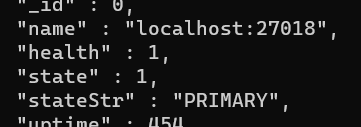


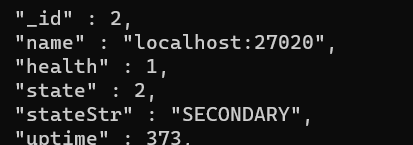


* Now again check the status using following code :-

rs.status()

This time it shows Primary instance and secondary instances two replicas of primary instance.





**Step : 7 Now create a new database and new collection in primary server.**

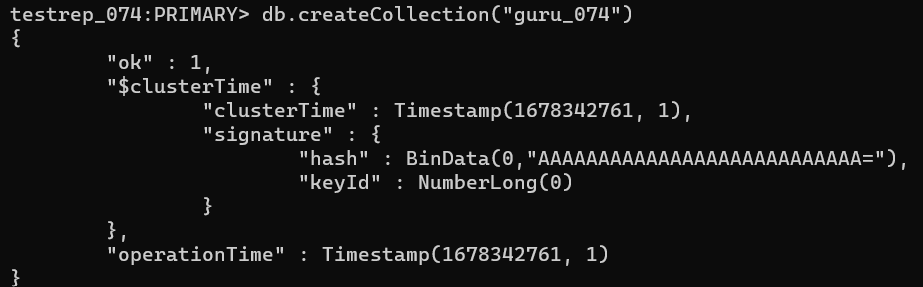
use newdb\_072

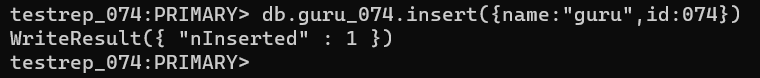
db.createCollection(“divy\_072”)

**Also insert some data in that collection**

db.divy\_072.insert({name : “abc”, number : 123})







**Step : 8 Now go to secondary servers and check the database which you created is present there or not.**

