Practical -5

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.

Why Replication?

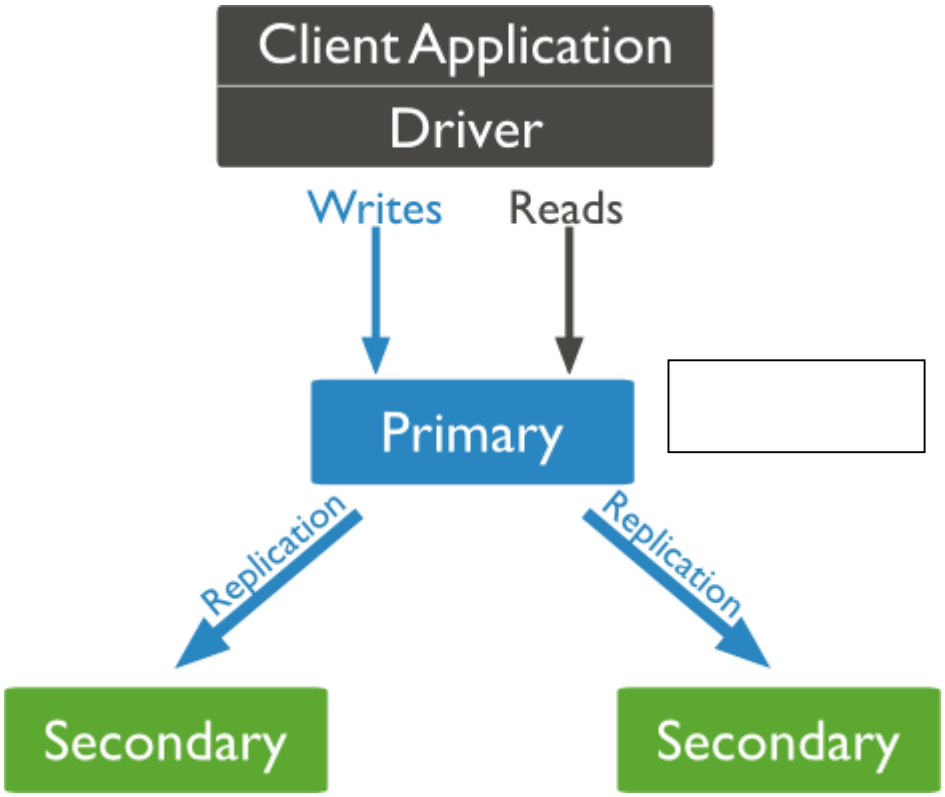
- To keep your data safe
- High (24*7) availability of data
- Disaster recovery
- No downtime for maintenance (like backups, index rebuilds, compaction)
- Read scaling (extra copies to read from)
- Replica set is transparent to the application

How Replication Works in MongoDB

MongoDB achieves replication by the use of replica set. A replica set is a group of **mongod** instances that host the same data set. In a replica, one node is primary node that receives all write operations. All other instances, such as secondaries, apply operations from the primary so that they have the same data set. Replica set can have only one primary node.

- Replica set is a group of two or more nodes (generally minimum 3 nodes are required).
- In a replica set, one node is primary node and remaining nodes are secondary.
- All data replicates from primary to secondary node.
- At the time of automatic failover or maintenance, election establishes for primary and a new primary node is elected.
- After the recovery of failed node, it again join the replica set and works as a secondary node.

A typical diagram of MongoDB replication is shown in which client application always interact with the primary node and the primary node then replicates the data to the secondary nodes.



Replica Set Features

- A cluster of N nodes
- Any one node can be primary
- All write operations go to primary
- Automatic failover
- Automatic recovery
- Consensus election of primary

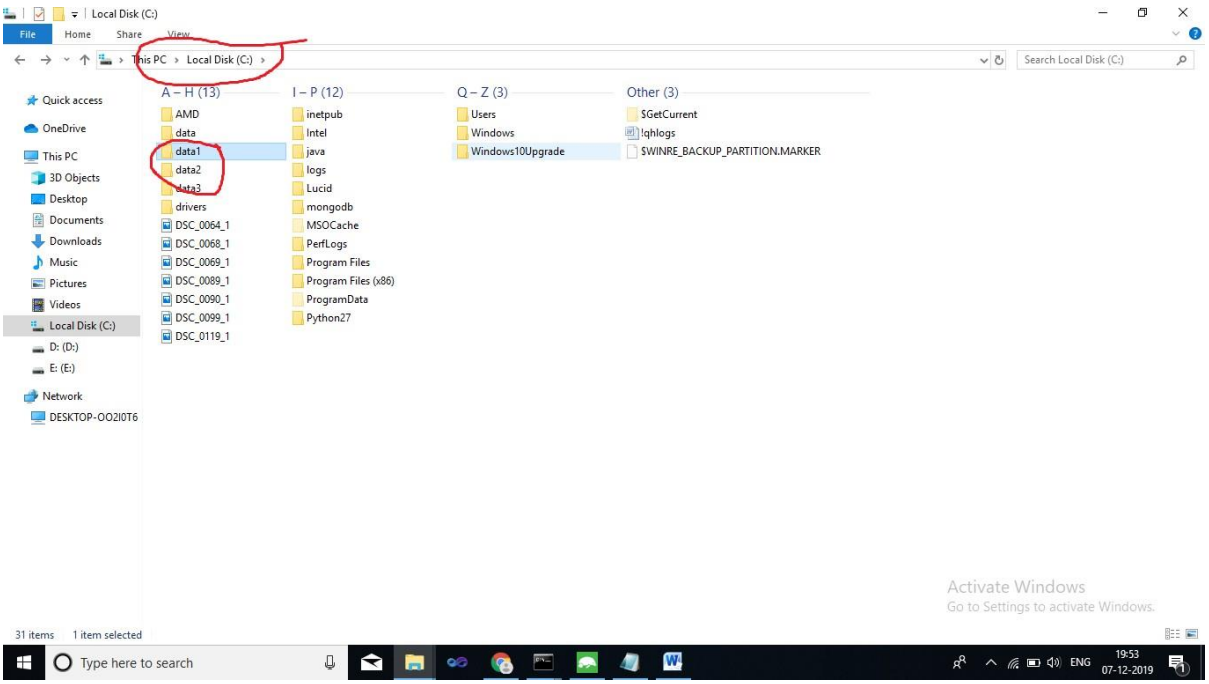
Example how to Set Up a Replica Set

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.



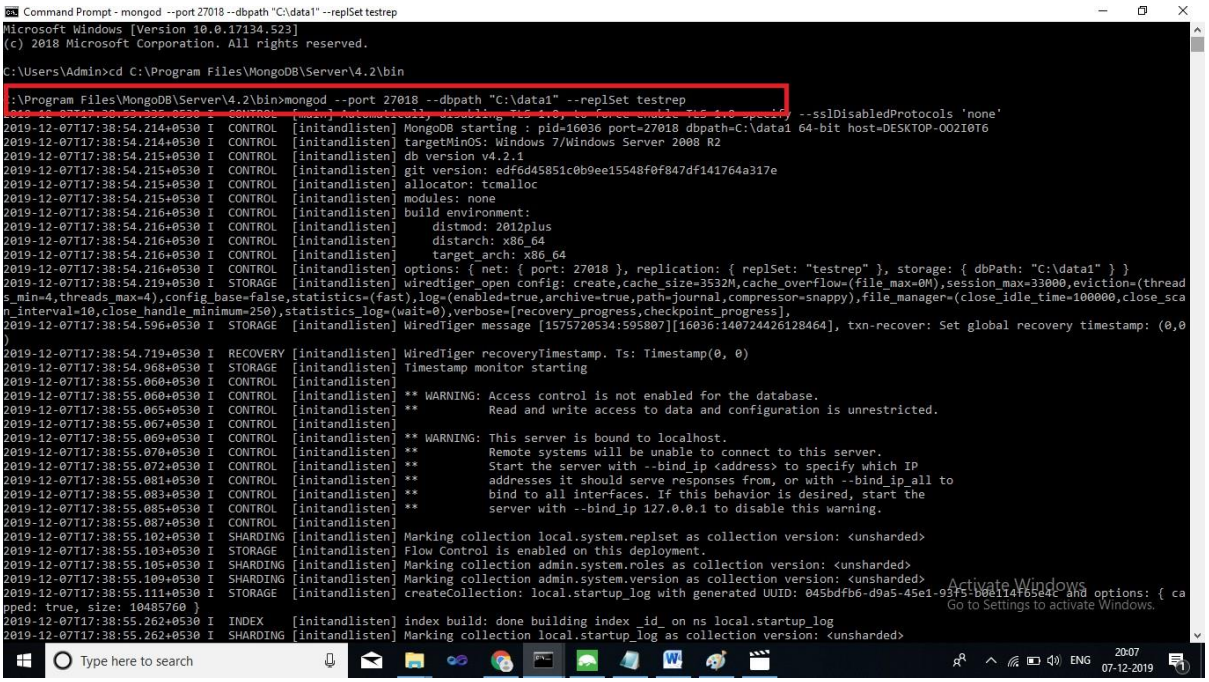
- Over here we consider data1 is our primary server and data2 and data3 is replica.

Step 3(Creating Primary instance)

Start the MongoDB server by specifying -- replSet option. Following is the basic syntax.

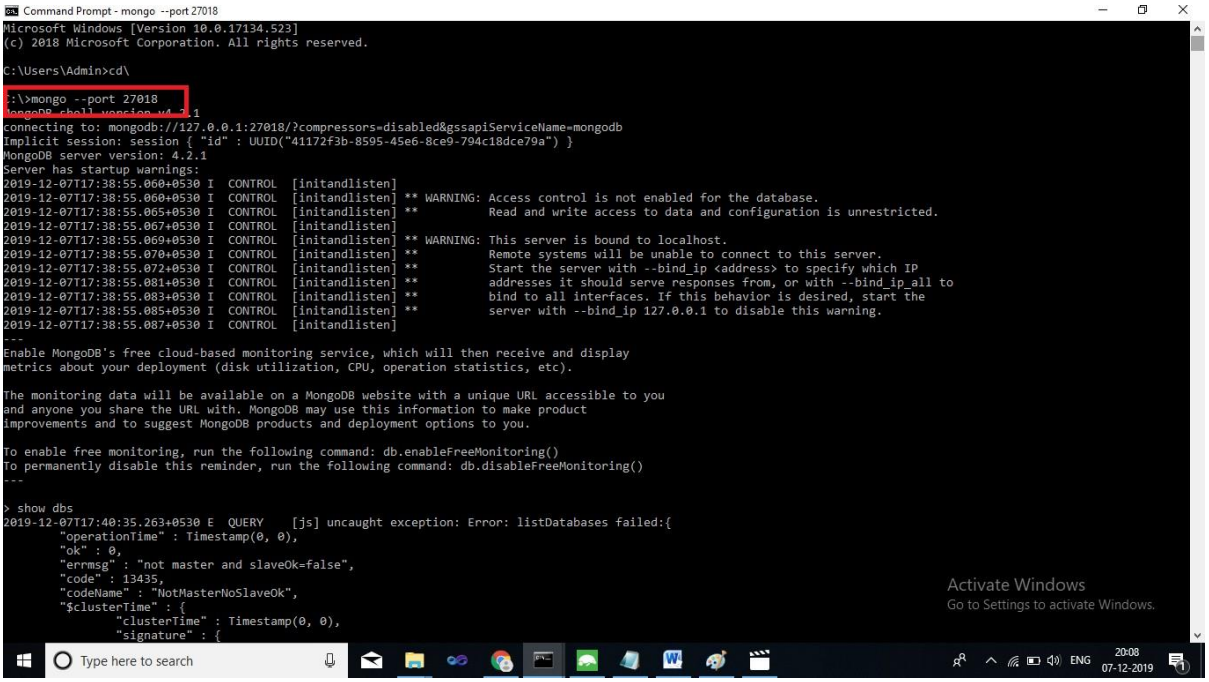
```
mongod --port "PORT" --dbpath "YOUR_DB_DATA_PATH" --replSet
"REPLICA_SET_INSTANCE_NAME"

C:\Program Files\MongoDB\Server\4.2\bin>mongod --port 27018 --
dbpath "C:\data1" --replSet testrep
```



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

C:\>mongo --port 27018

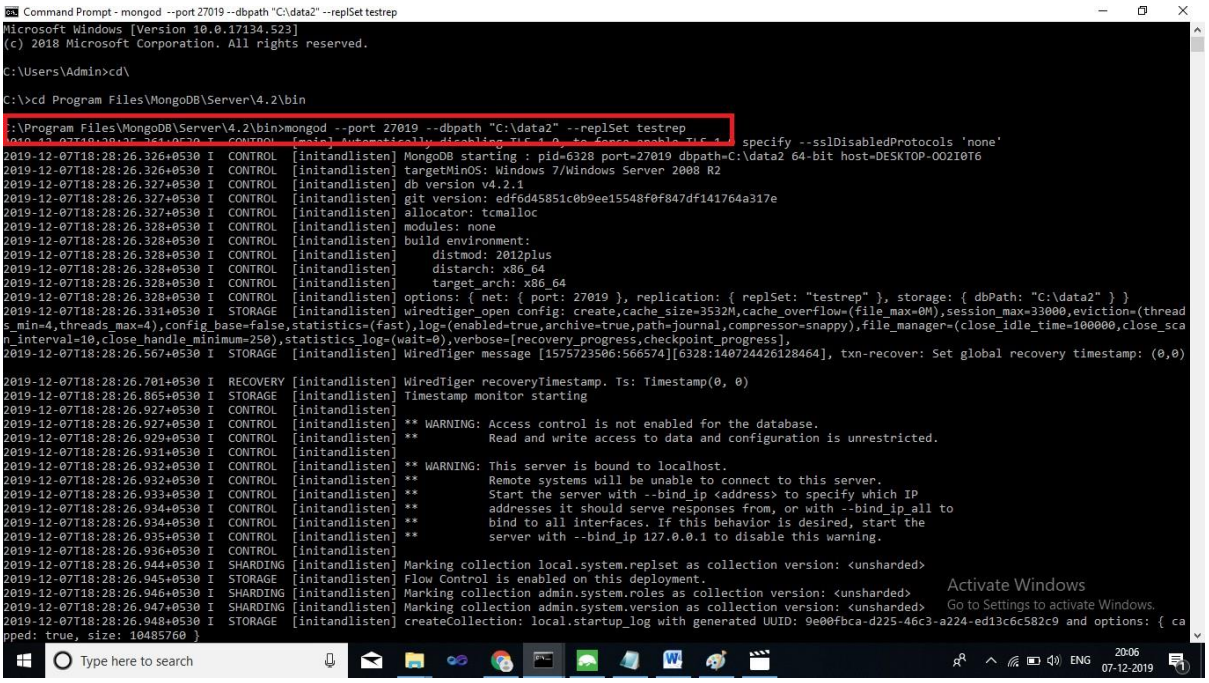


Now Primary server is working.

Step 4(Creating 1(first) Replicas of Primary instance)

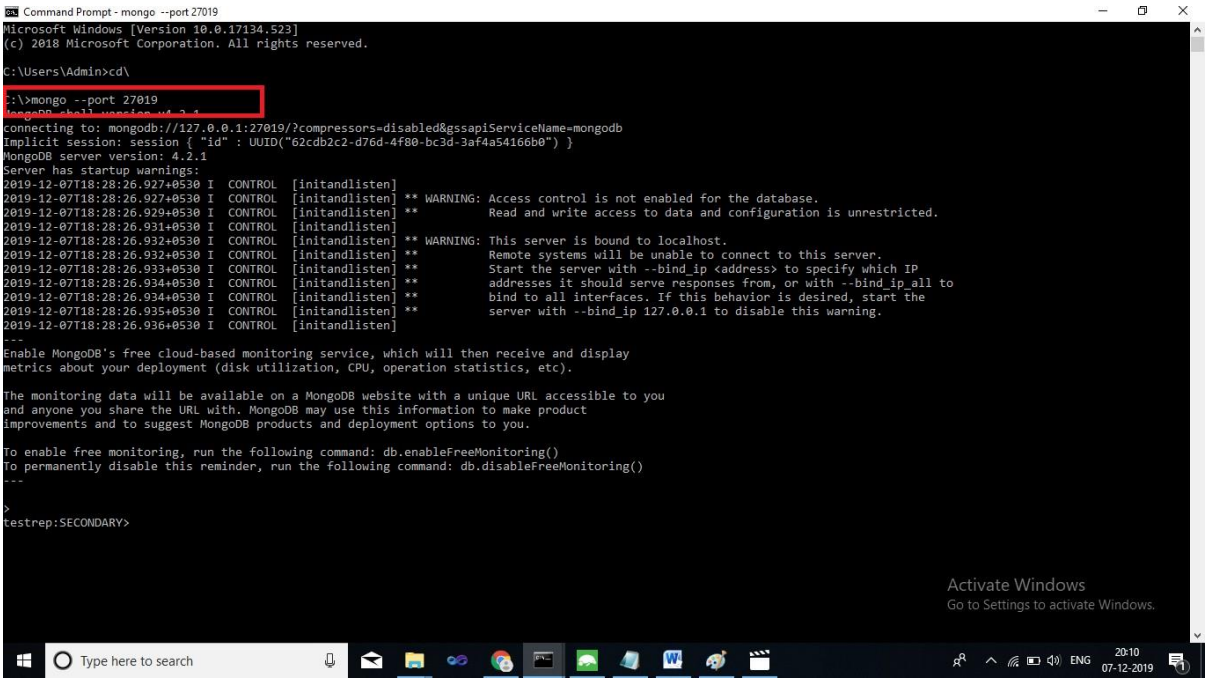
Syntax

C:\Program Files\MongoDB\Server\4.2\bin>mongod --port 27019 --dbpath "C:\data2" --replSet testrep



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

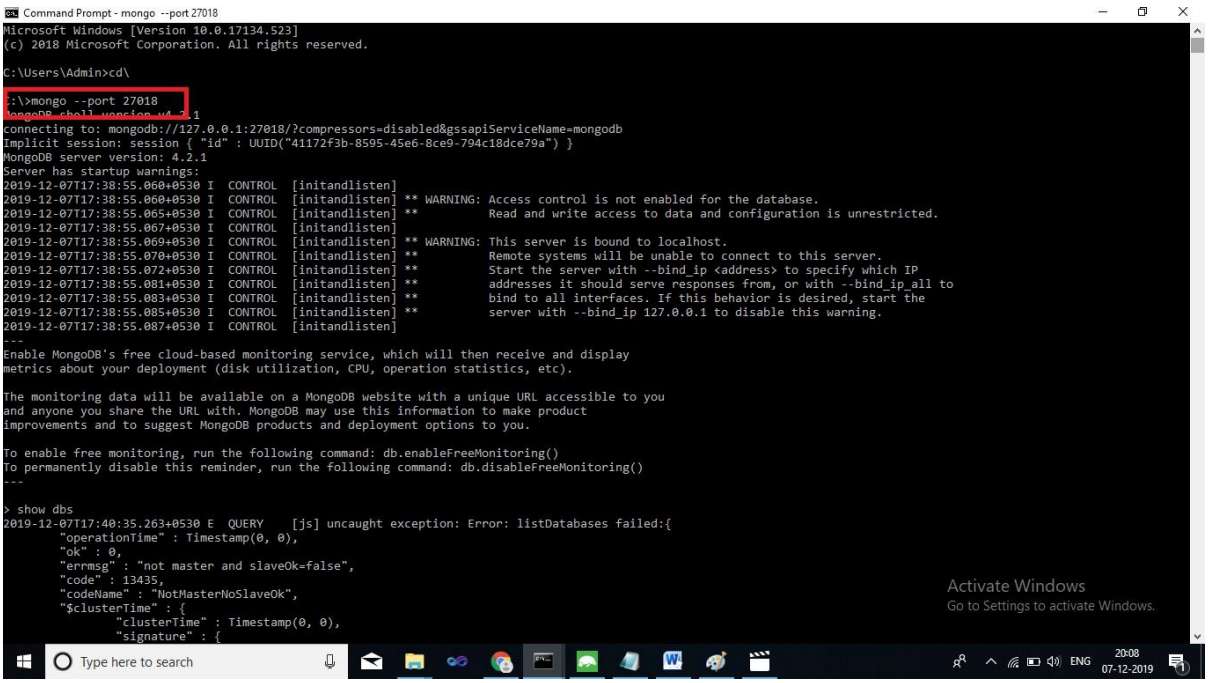
C:\>mongo --port 27019



Step 5

Now go to the command prompt of Primary server's Client instance.

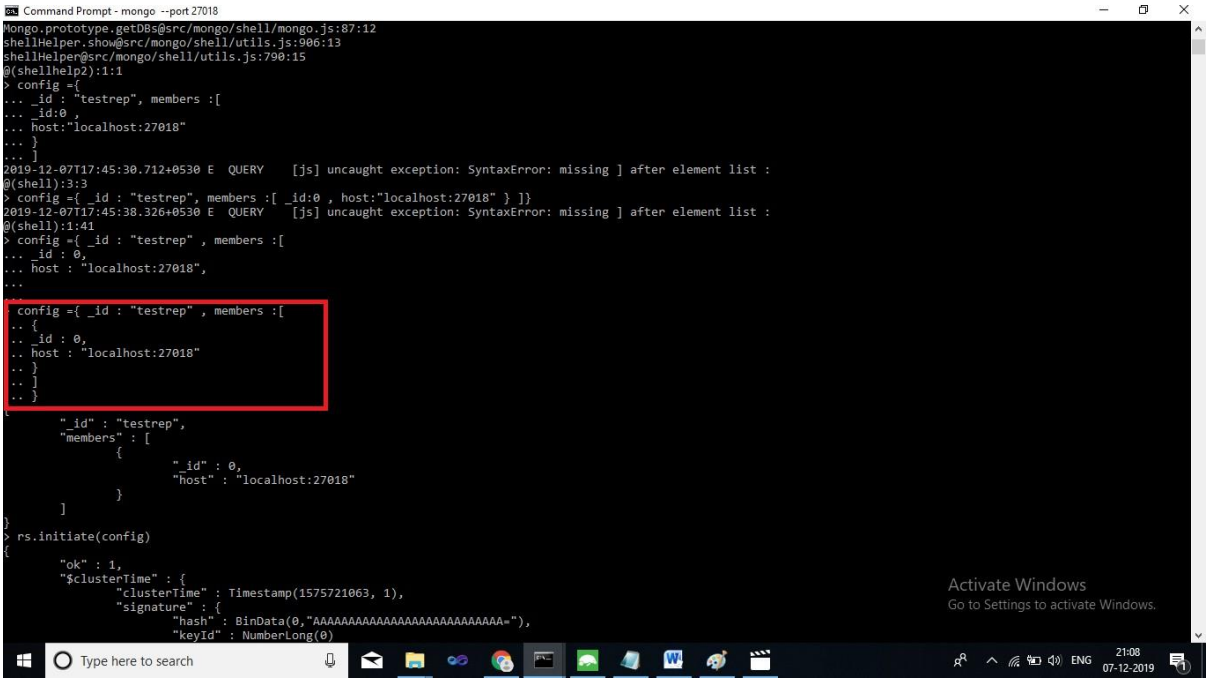
C:\>mongo --port 27018



Now type the following code

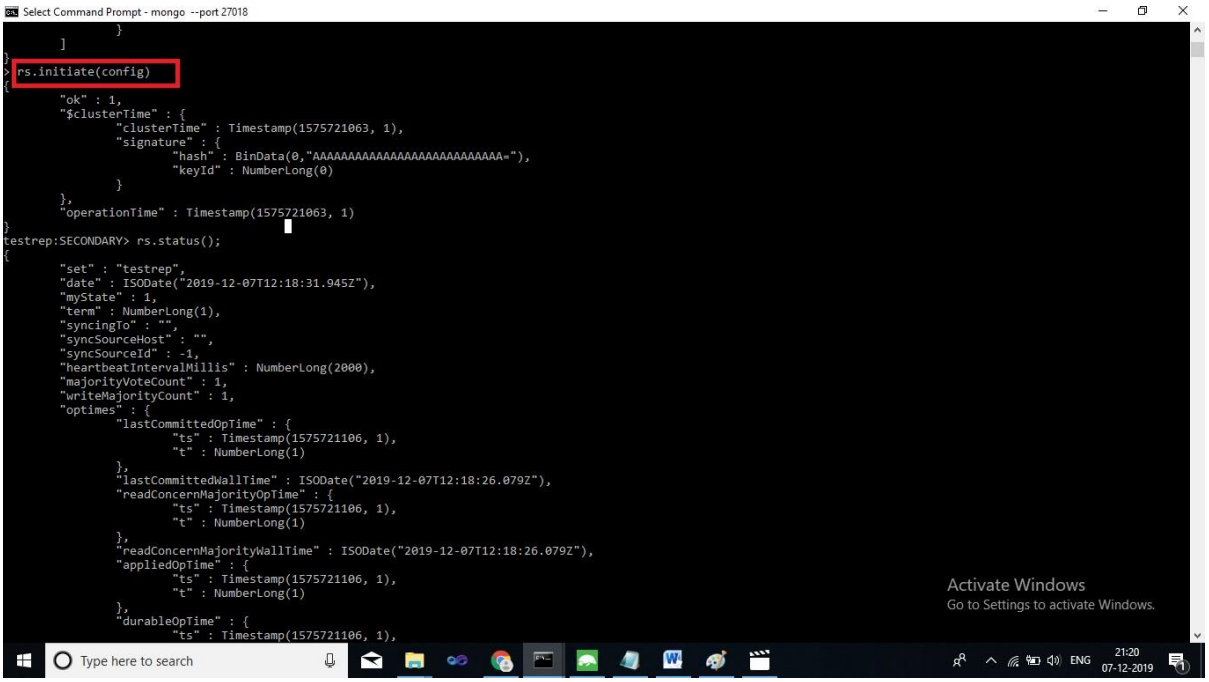
```
config ={ _id : "testrep" , members :[
{
_id : 0,
 host : "localhost:27018"
}
 ]
}
```

Above code set id=0 to the first replica instance which is on port 27018.(PRIMARY INSTANCE)



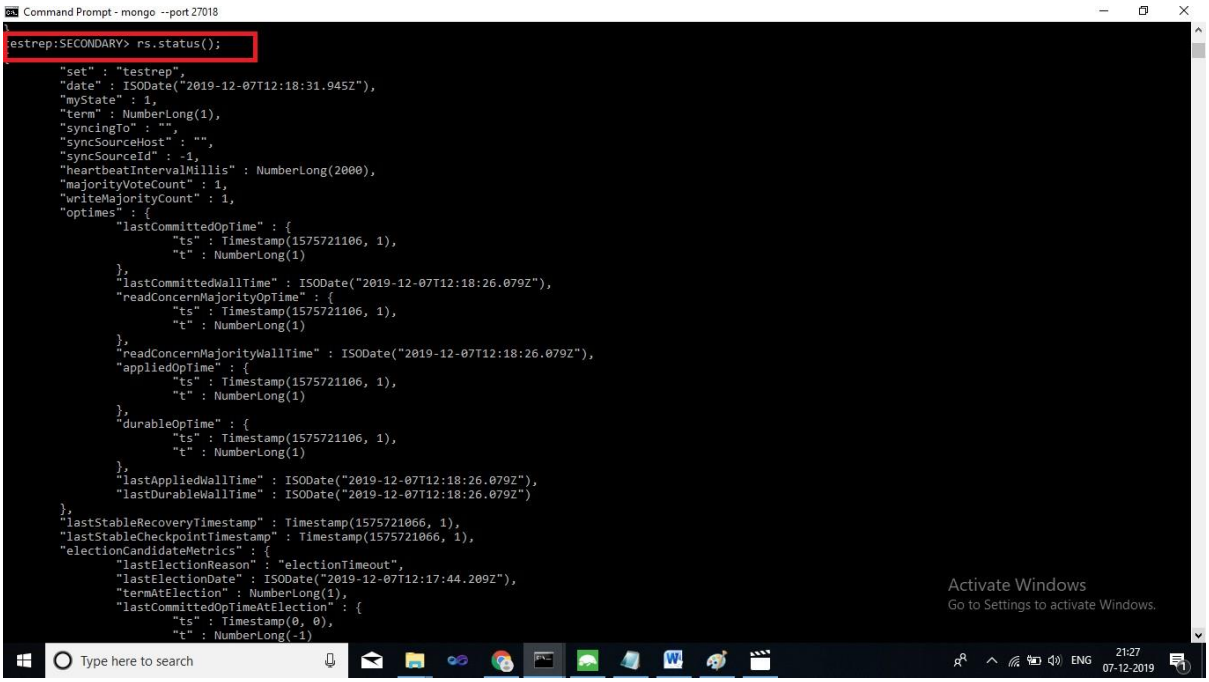
After this write command
rs.initiate(config)

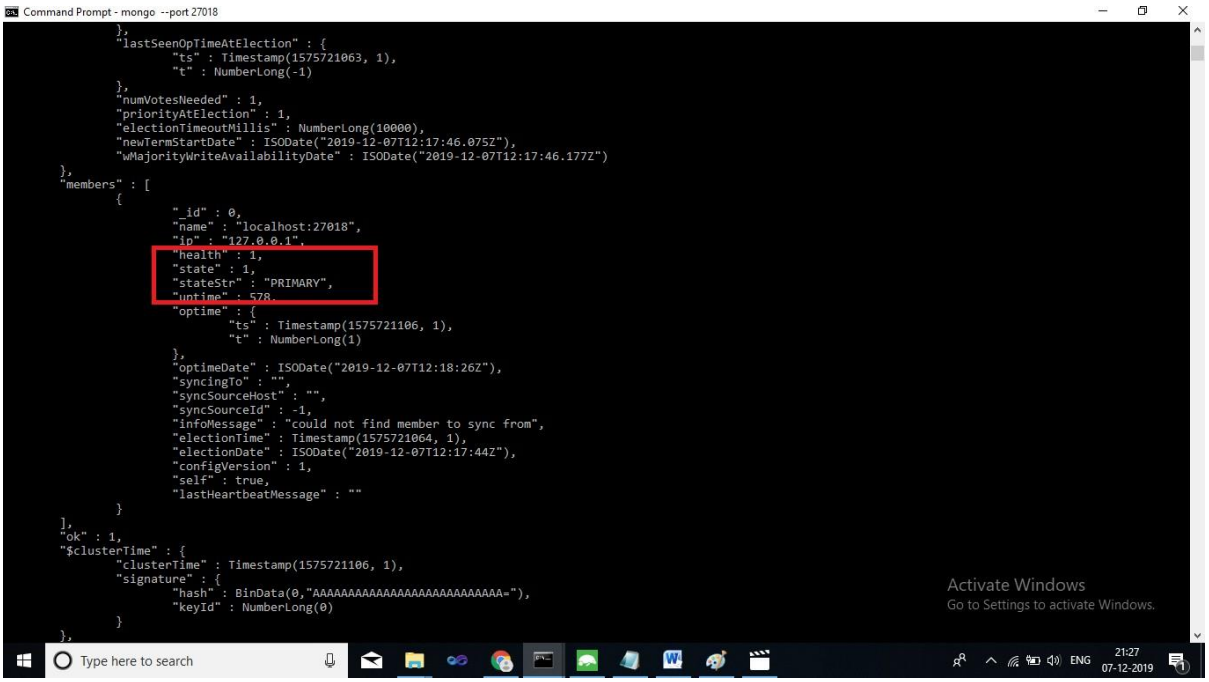
This command initiates a replica set with the current host as its only member. This is confirmed by the output, which should resemble the following:



After this writte write command

rs.status()

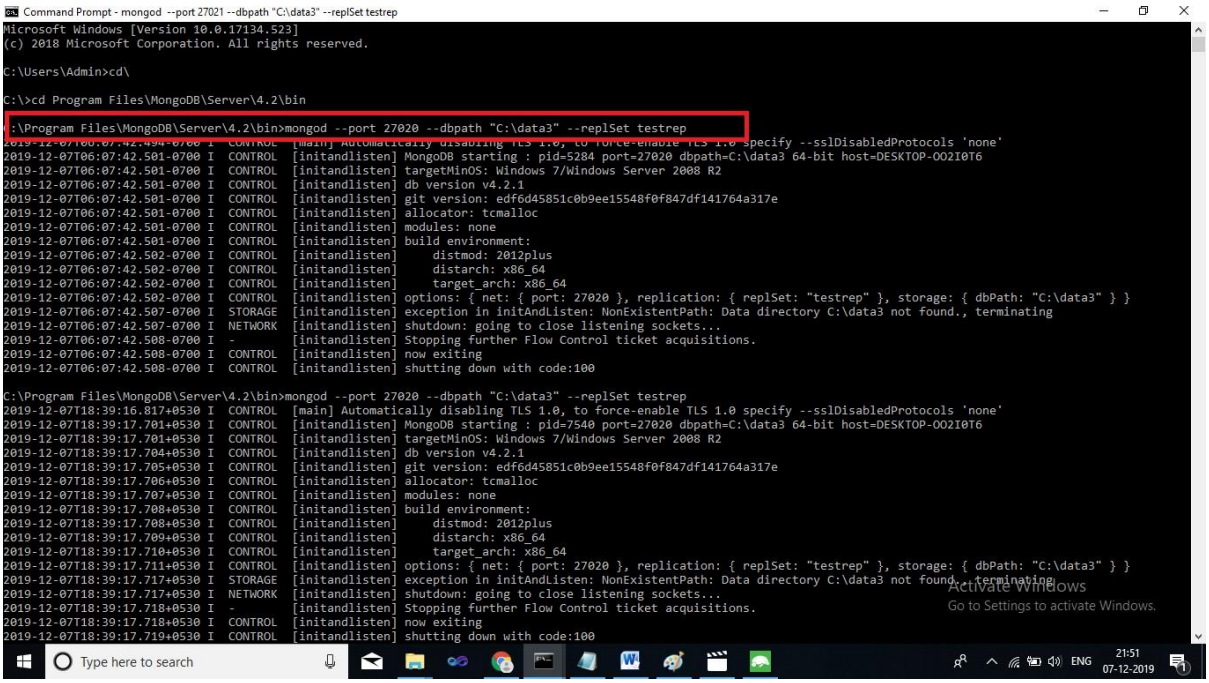




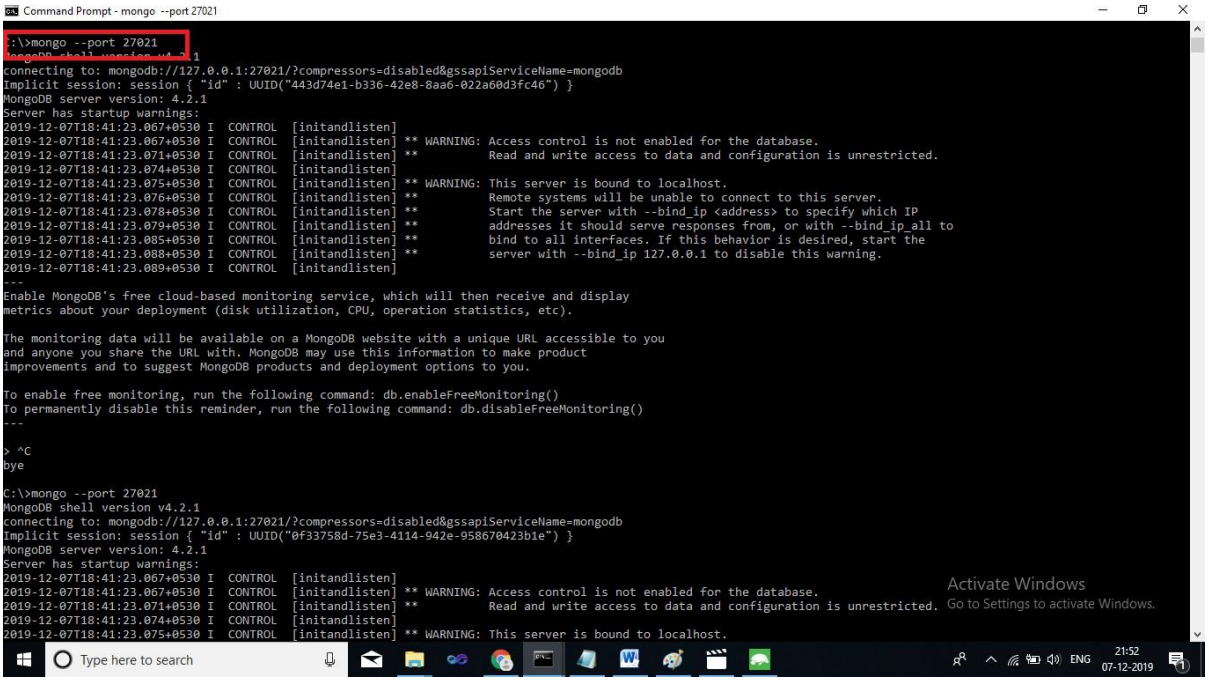
Step 6(Creating 2(Second—port 27021) Replicas of Primary instance)

Syntax

C:\Program Files\MongoDB\Server\4.2\bin>mongod --port 27021 --dbpath "C:\data3" --replSet testrep



After this start mongo client of 2 seconday instance.



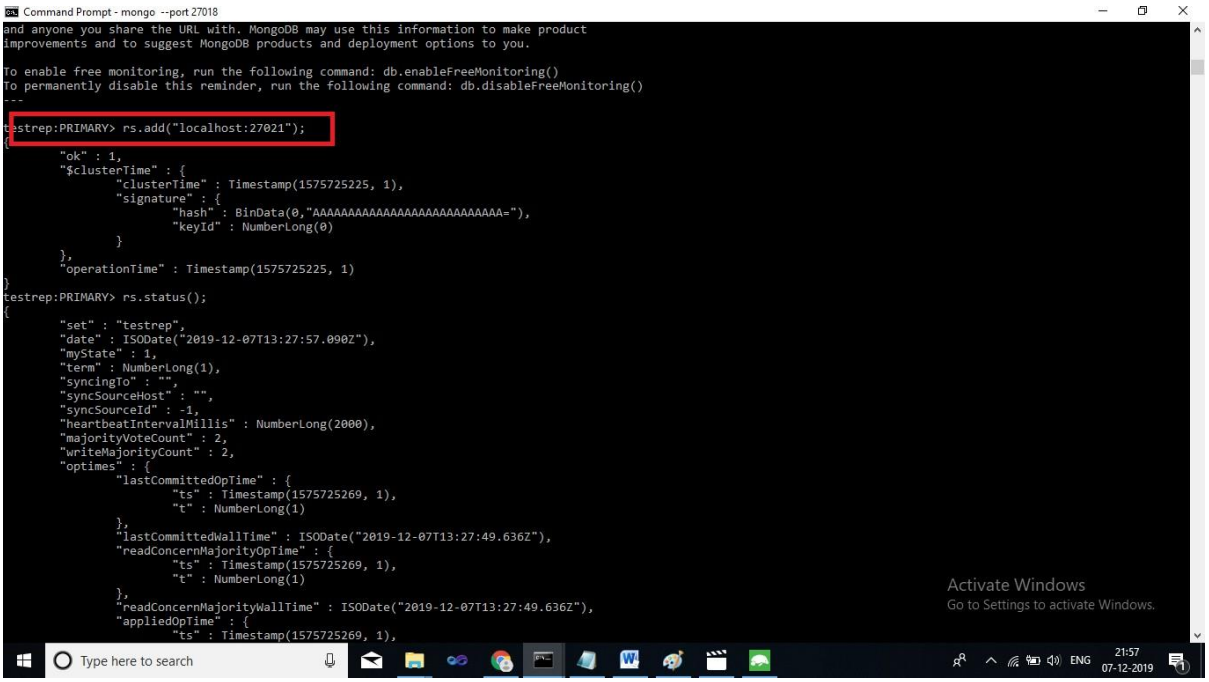
Step 7

Now goto the CMD Client window of Primary Client(port 27018)

Execute rs.add() method.

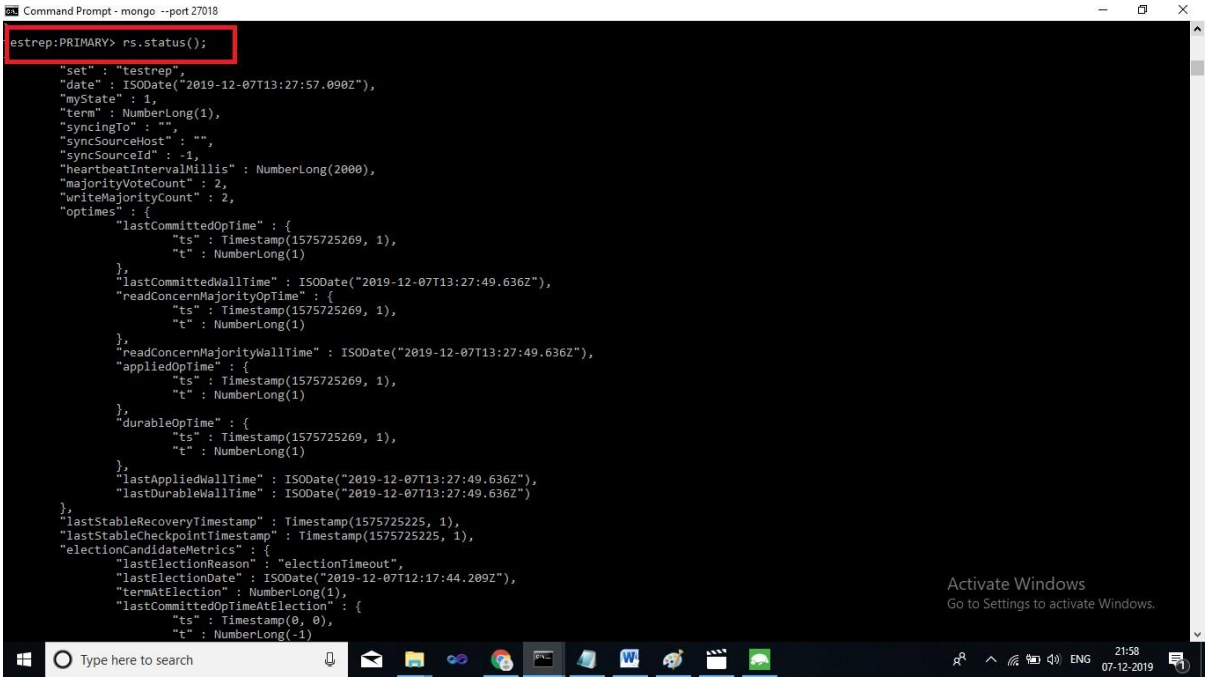
Syntax

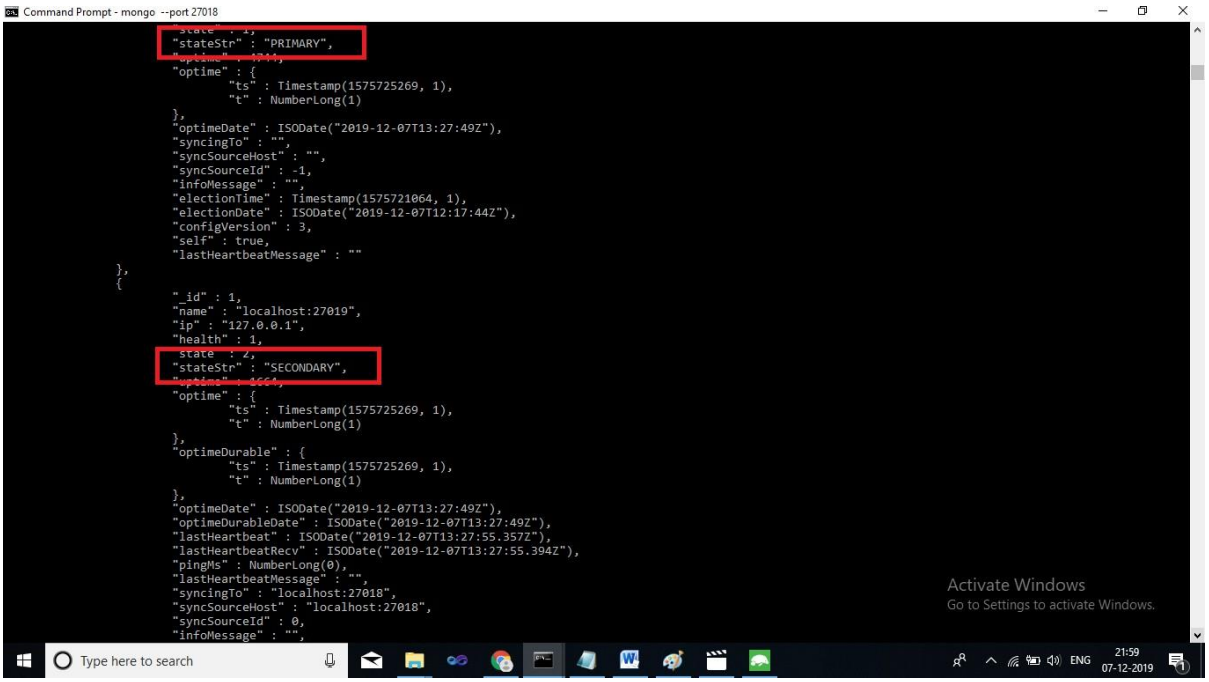
rs.add(“localhost:27021”);

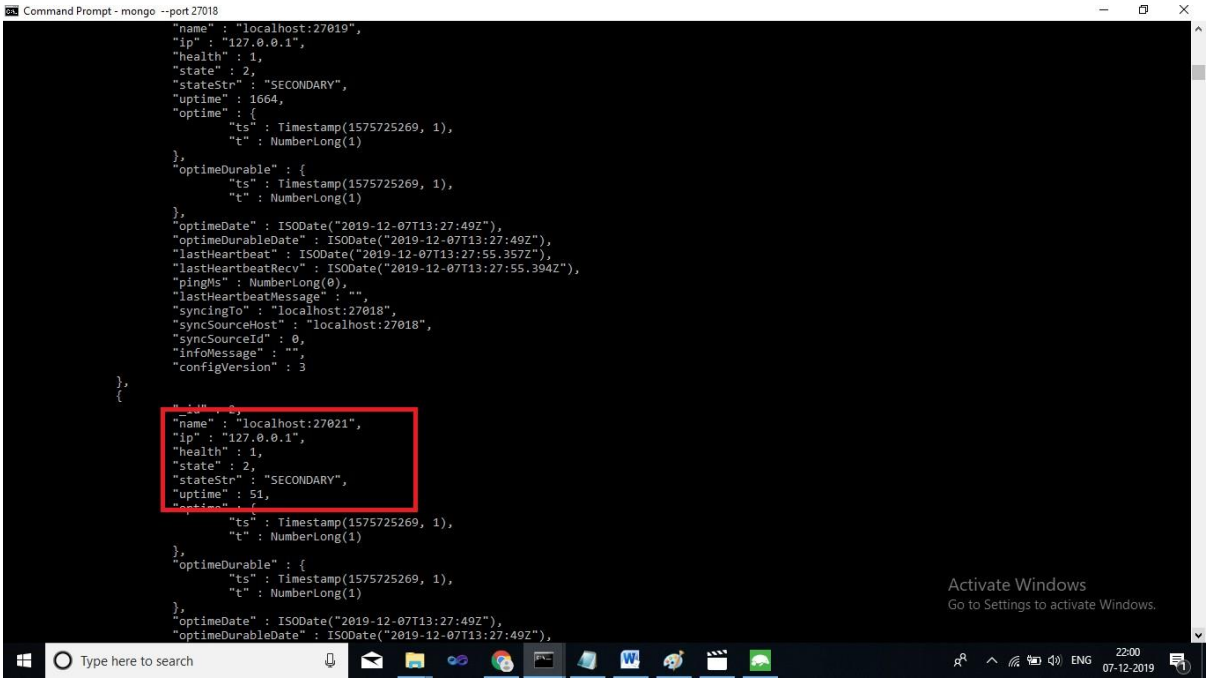


After this execute rs.status command.

This will show status of our Cluster. In our cluster three instance first Primary and remaining two are replicas of primary instance.



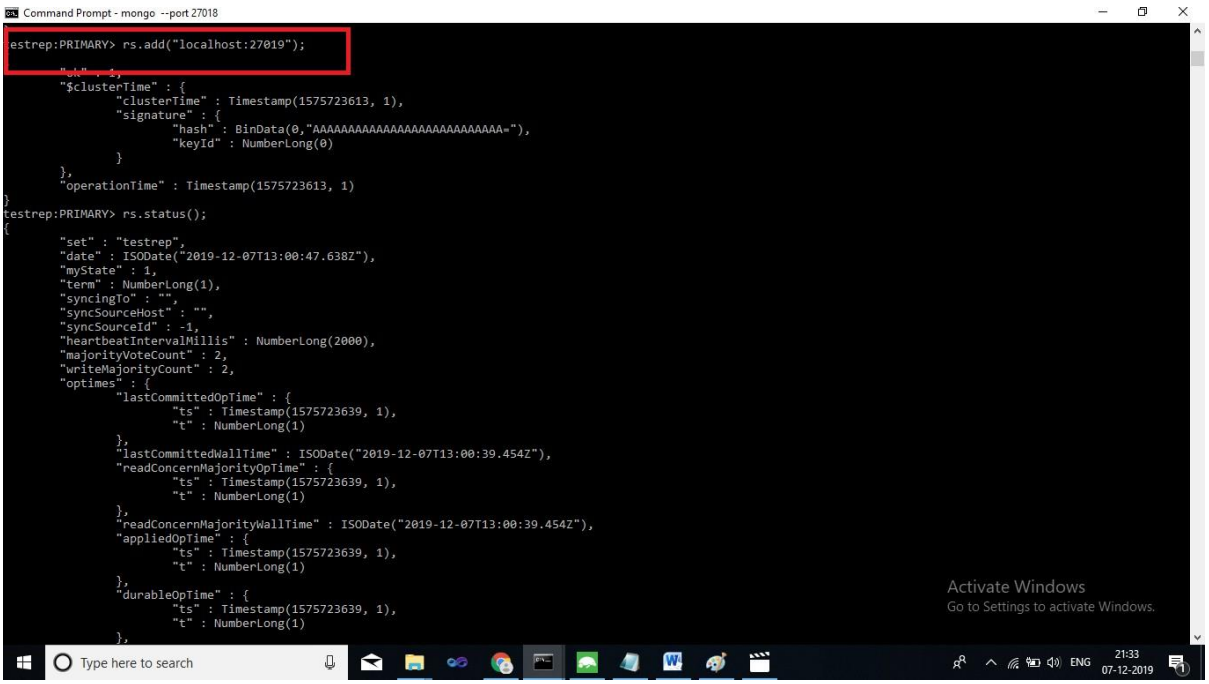




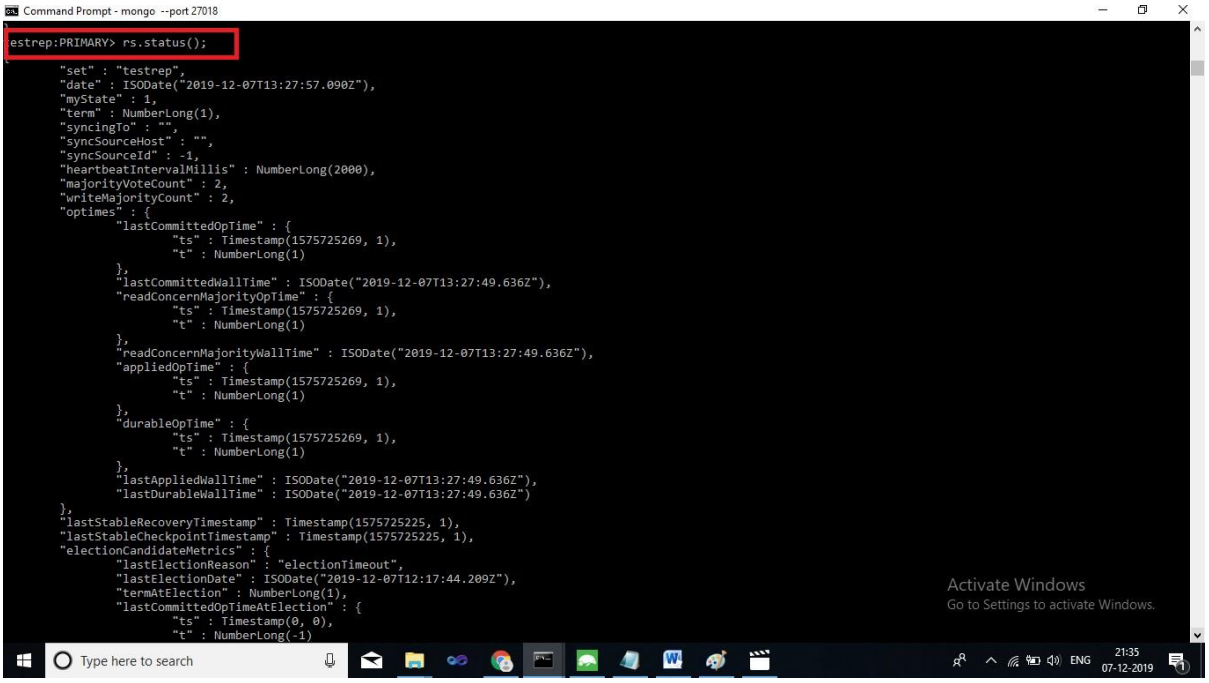
Step 8

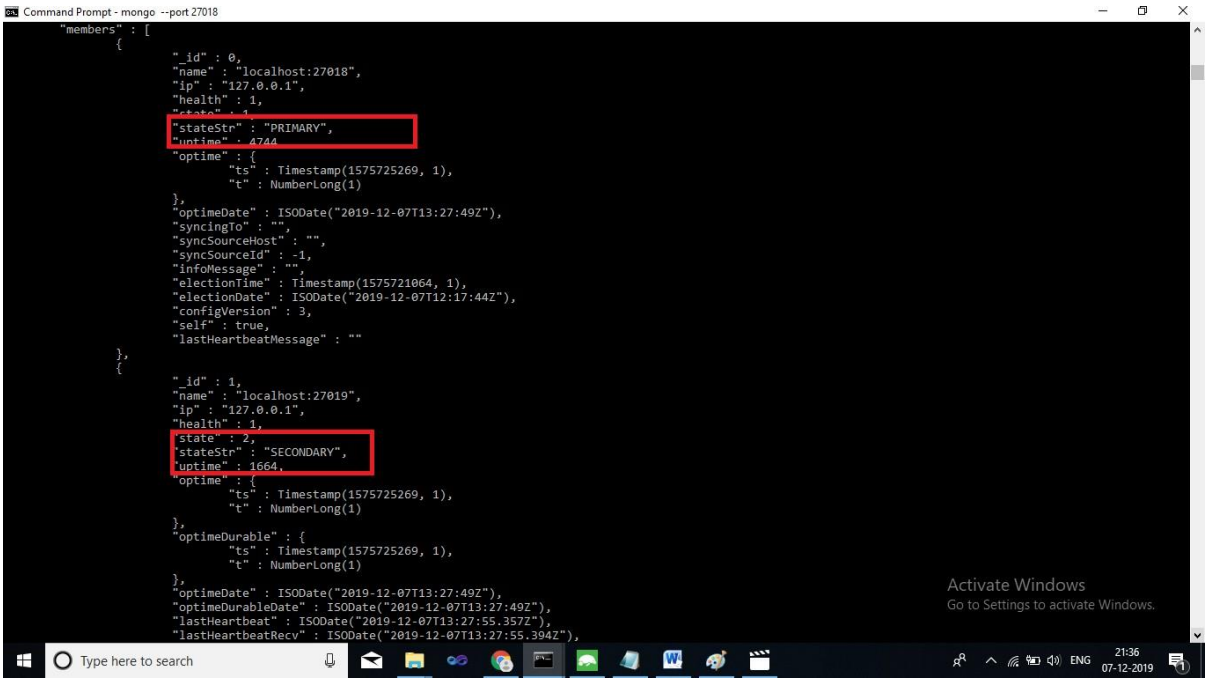
On cmd window of(client of 27018 main) Add the replica of main instance which is created on local host port 27019 in Cluster using rs.add() method.

rs.add("localhost:27019");



Now execute rs.status() command so we can see now in our cluster there are two instance one is primary and other is secondary.



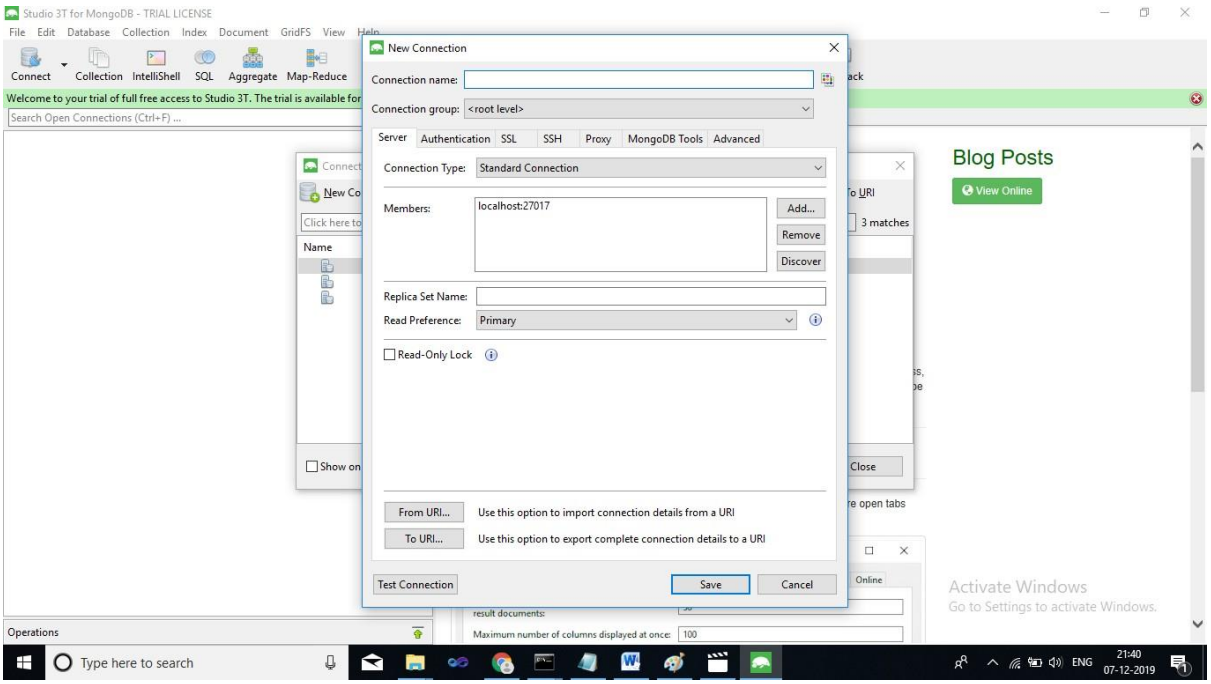


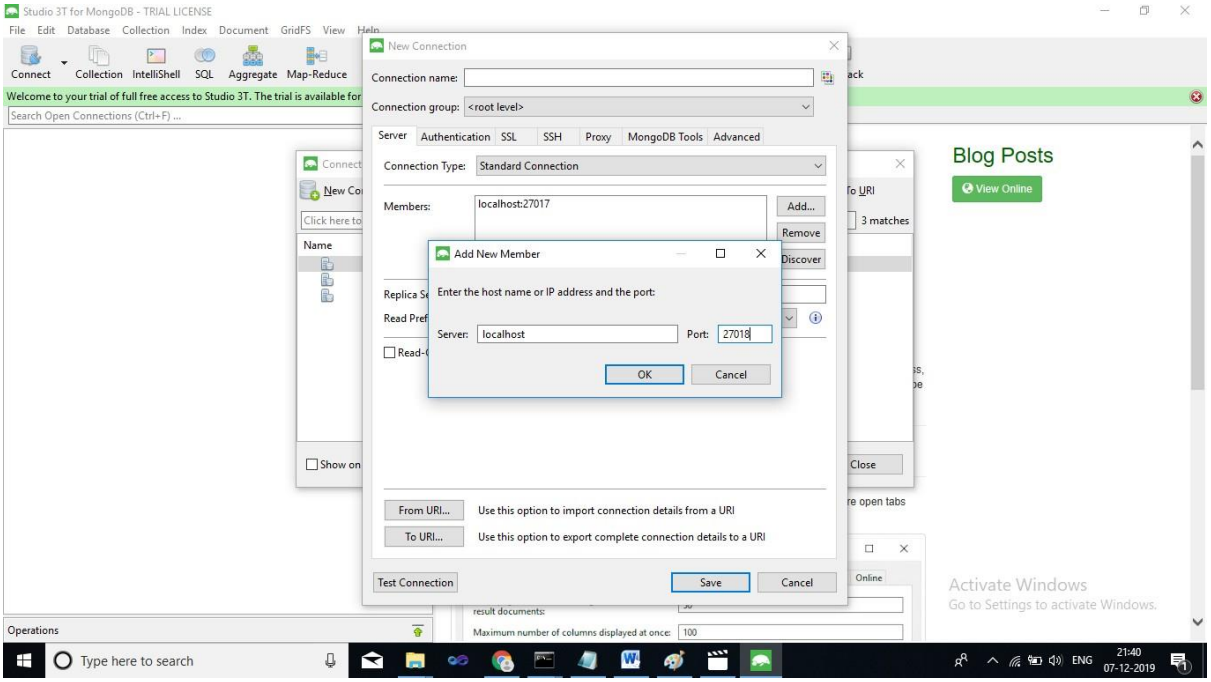
Step 9

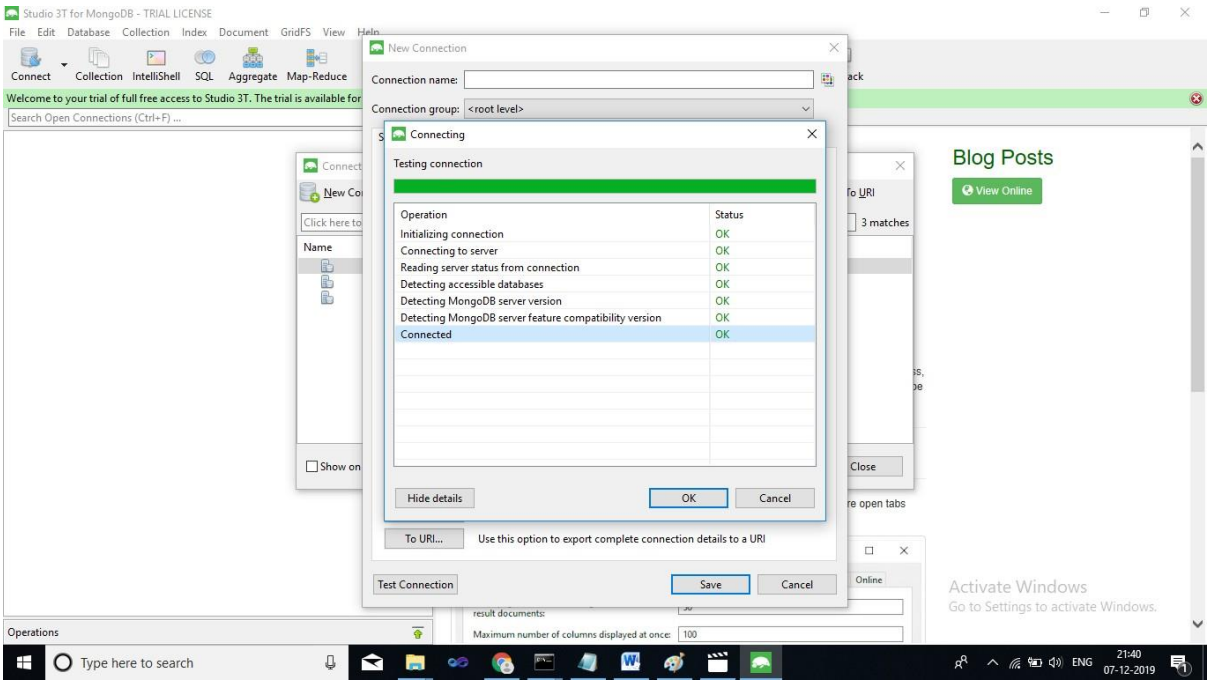
Now OPEN studio 3T

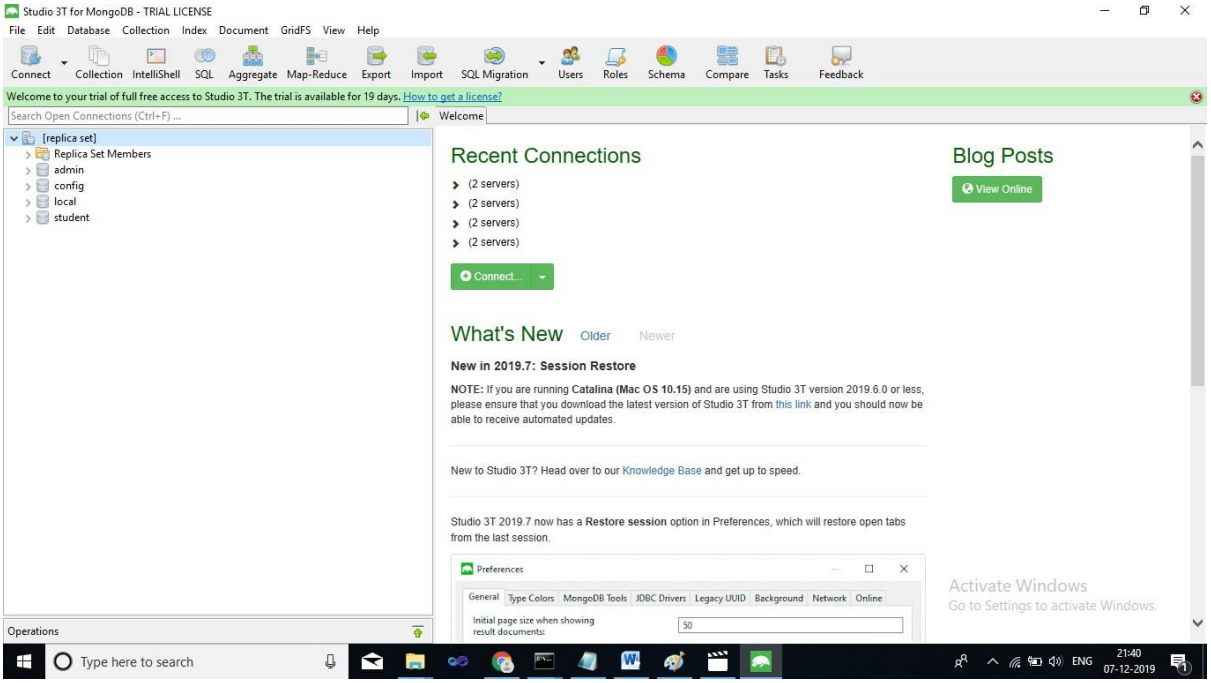
Click on connect and add both 27018 Primary and 27019,27021 seconday servers. As given below.

Connecting with primary instance 27018


Studio 3T for MongoDB - TRIAL LICENSE
File Edit Database Collection Index Document GridFS View Help
Connect Collection IntelliShell SQL Aggregate Map-Reduce
Welcome to your trial of full free access to Studio 3T. The trial is available for
Search Open Connections (Ctrl+F) ...
New Connection
Connection name:
Connection group: <root level>
Server Authentication SSL SSH Proxy MongoDB Tools Advanced
Connection Type: Standard Connection
Members: localhost:27017
Add...
Remove
Discover
Replica Set Name:
Read Preference: Primary
Read-Only Lock
From URI... Use this option to import connection details from a URI
To URI... Use this option to export complete connection details to a URI
Test Connection
Save
Cancel
Blog Posts
View Online
Activate Windows
Go to Settings to activate Windows.
Operations
Maximum number of columns displayed at once: 100
Type here to search
ENG
21:40
07-12-2019

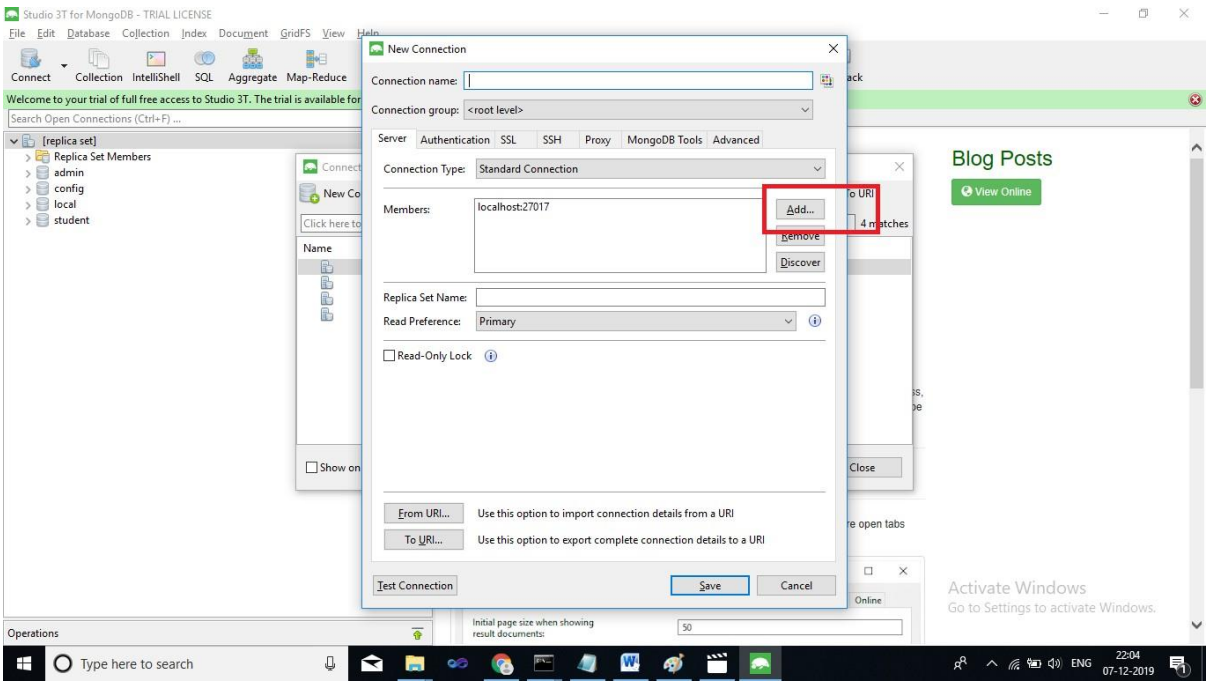

Studio 3T for MongoDB - TRIAL LICENSE
File Edit Database Collection Index Document GridFS View Help
Connect Collection IntelliShell SQL Aggregate Map-Reduce
Welcome to your trial of full free access to Studio 3T. The trial is available for
Search Open Connections (Ctrl+F) ...
New Connection
Connection name:
Connection group: <root level>
Server Authentication SSL SSH Proxy MongoDB Tools Advanced
Connection Type: Standard Connection
Members: localhost:27017
Add...
Remove
Discover
Add New Member
Enter the host name or IP address and the port:
Server: localhost
Port: 27018
OK
Cancel
From URI... Use this option to import connection details from a URI
To URI... Use this option to export complete connection details to a URI
Test Connection
Save
Cancel
Blog Posts
View Online
Activate Windows
Go to Settings to activate Windows.
Operations
Maximum number of columns displayed at once: 100
Type here to search
ENG
21:40
07-12-2019

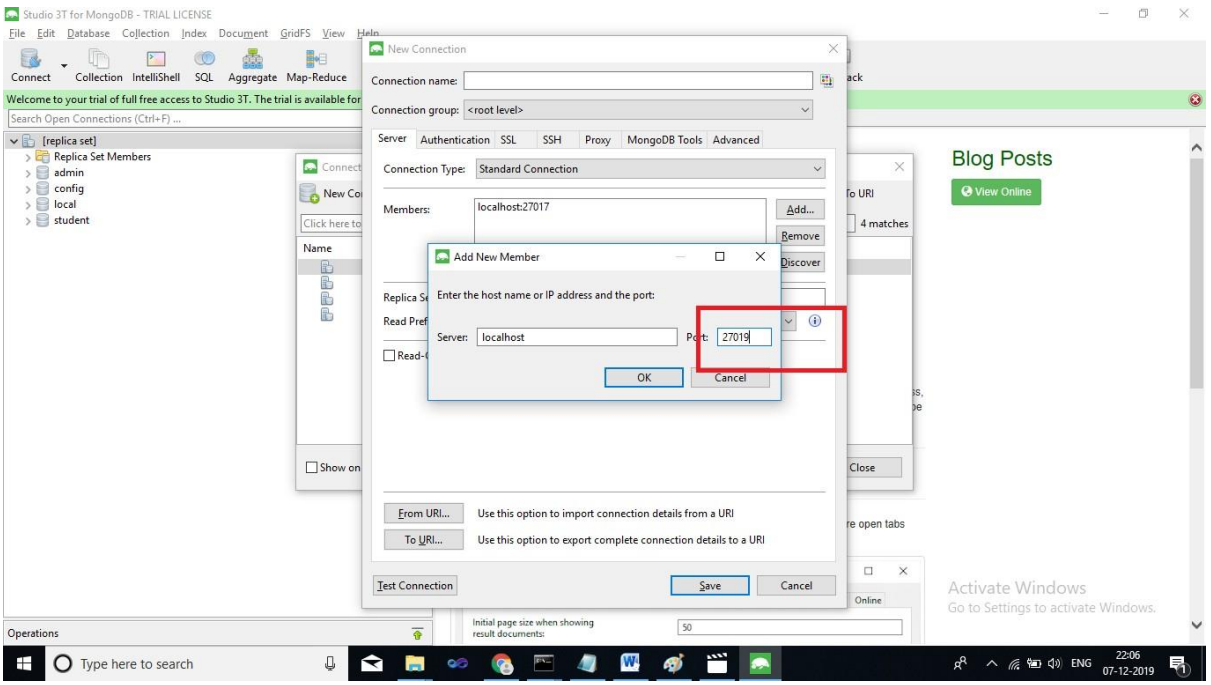

Studio 3T for MongoDB - TRIAL LICENSE
File Edit Database Collection Index Document GridFS View Help
Connect Collection IntelliShell SQL Aggregate Map-Reduce
New Connection
Connection name:
Connection group: <root level>
Connecting
Testing connection
Operation Status
Initializing connection OK
Connecting to server OK
Reading server status from connection OK
Detecting accessible databases OK
Detecting MongoDB server version OK
Detecting MongoDB server feature compatibility version OK
Connected OK
Hide details OK Cancel
To URI... Use this option to export complete connection details to a URI
Test Connection Save Cancel
Blog Posts
View Online
Activate Windows
Go to Settings to activate Windows.
Operations
Type here to search
ENG 21:40 07-12-2019

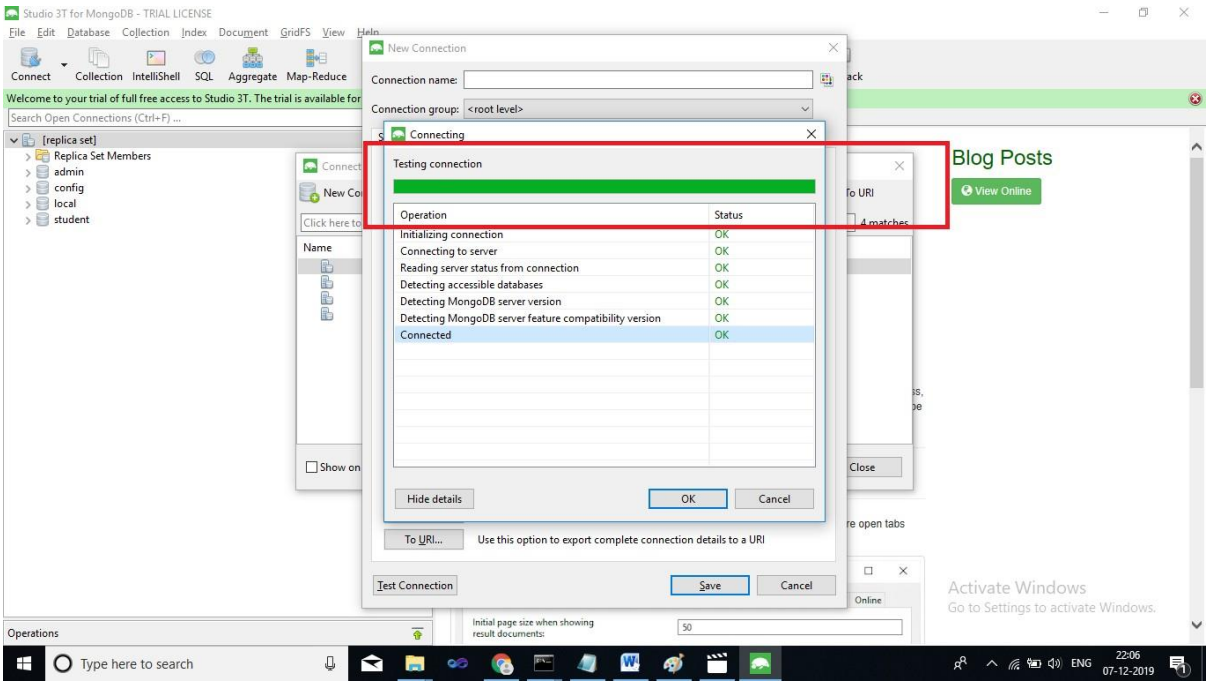

Studio 3T for MongoDB - TRIAL LICENSE
File Edit Database Collection Index Document GridFS View Help
Connect Collection IntelliShell SQL Aggregate Map-Reduce Export Import SQL Migration Users Roles Schema Compare Tasks Feedback
Welcome to your trial of full free access to Studio 3T. The trial is available for 19 days. How to get a license?
Search Open Connections (Ctrl+F) ...
Welcome
[replica set]
Replica Set Members
admin
config
local
student
Recent Connections
(2 servers)
(2 servers)
(2 servers)
(2 servers)
Connect...
What's New Older Newer
New in 2019.7: Session Restore
NOTE: If you are running Catalina (Mac OS 10.15) and are using Studio 3T version 2019.6.0 or less, please ensure that you download the latest version of Studio 3T from this link and you should now be able to receive automated updates.
New to Studio 3T? Head over to our Knowledge Base and get up to speed.
Studio 3T 2019.7 now has a Restore session option in Preferences, which will restore open tabs from the last session.
Preferences
General Type Colors MongoDB Tools JDBC Drivers Legacy UUID Background Network Online
Initial page size when showing result documents: 50
Blog Posts
View Online
Activate Windows
Go to Settings to activate Windows.
Operations
Type here to search
ENG 21:40 07-12-2019

STEP 10

Add the 1(first) Secondary instance in Studio 3t. Localhost port—27019



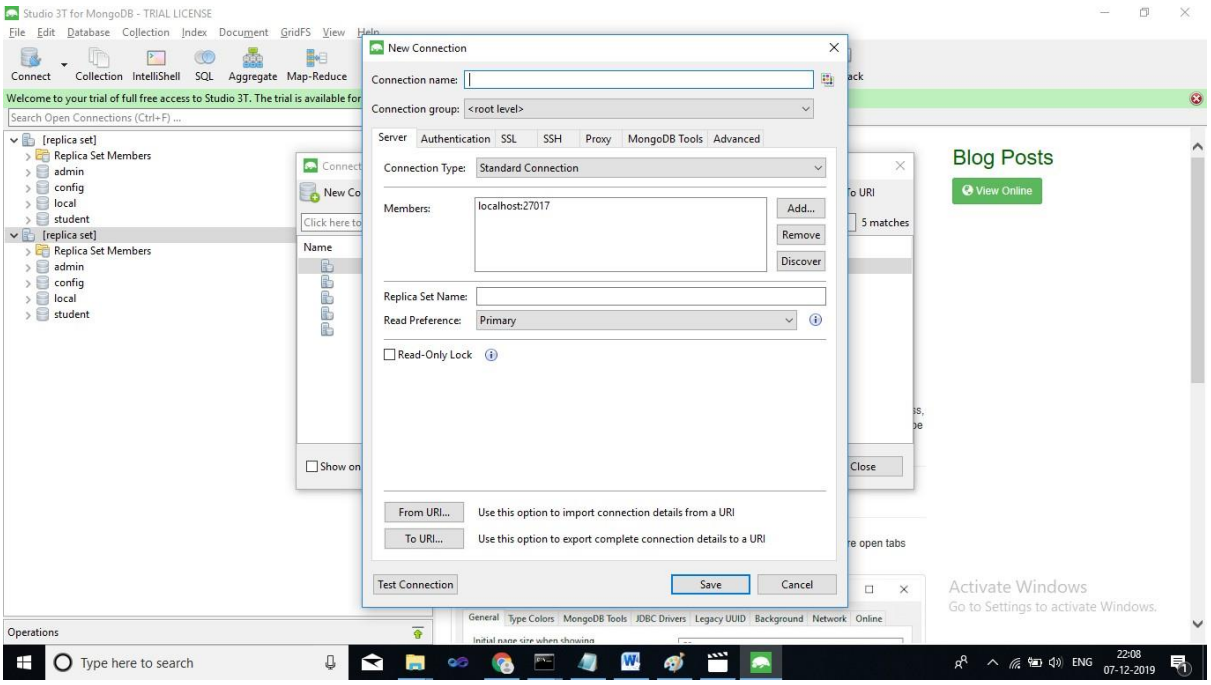


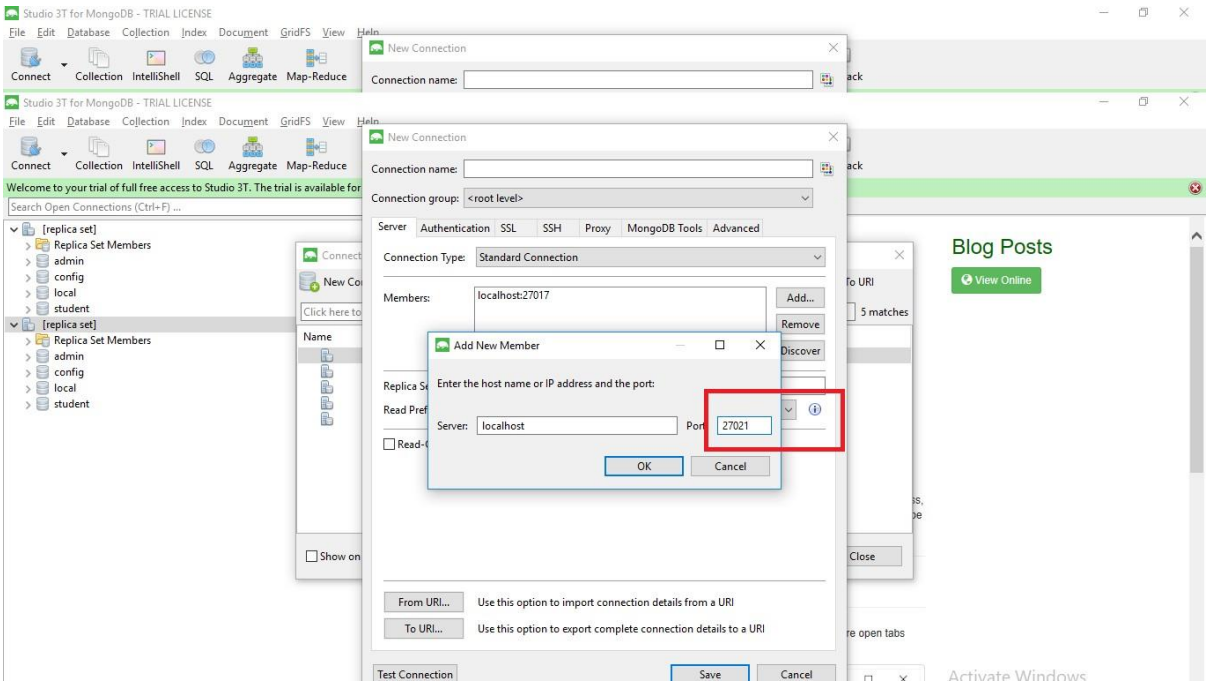


STEP 11

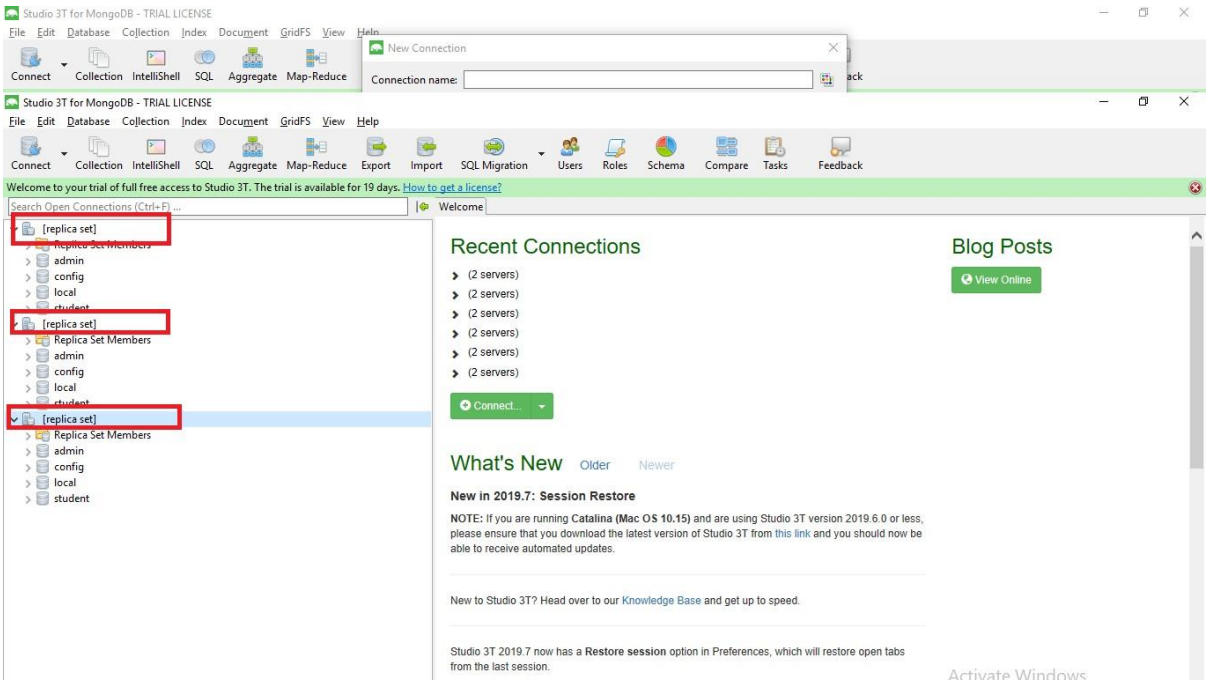
Add the 2(second) Secondary instance in Studio 3t.

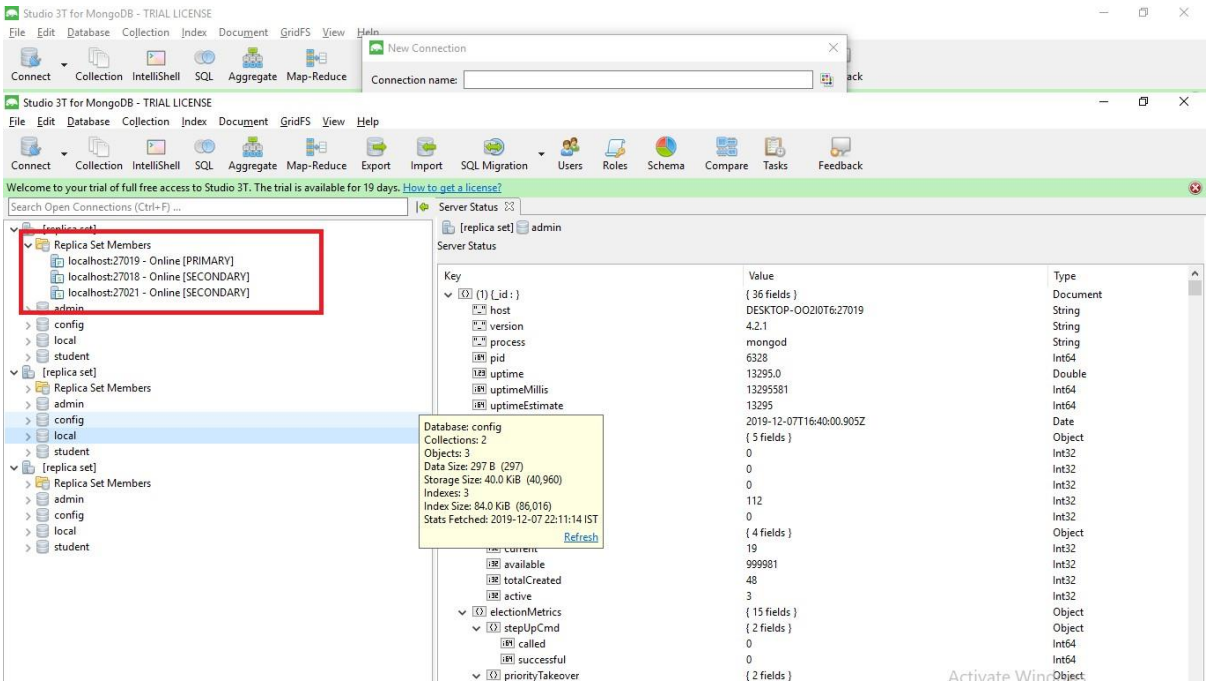
Localhost port—27021



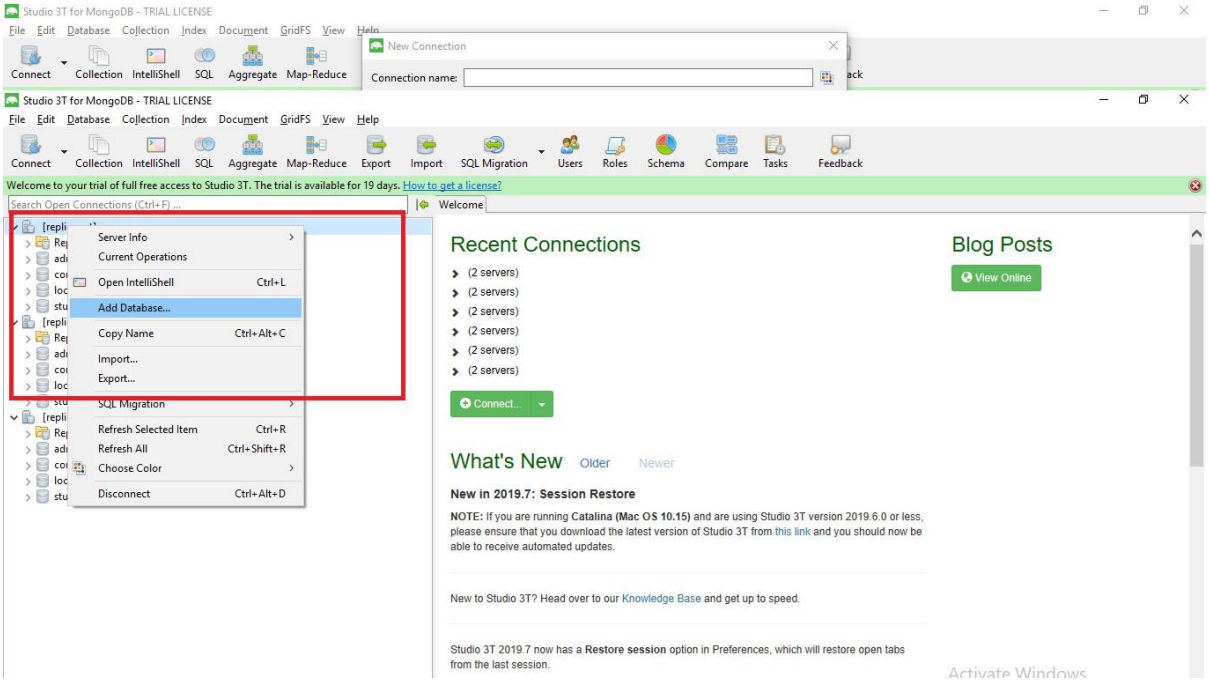


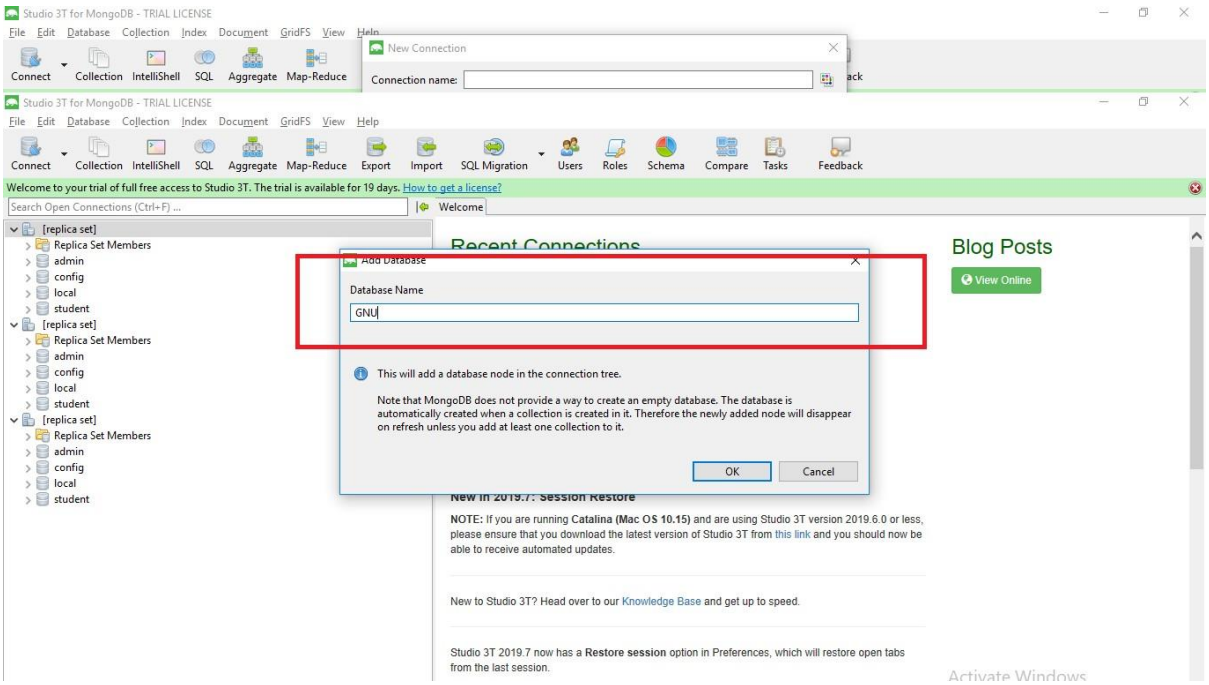
Now we can see we have 3 instance. One is Primary and other two are secondary.



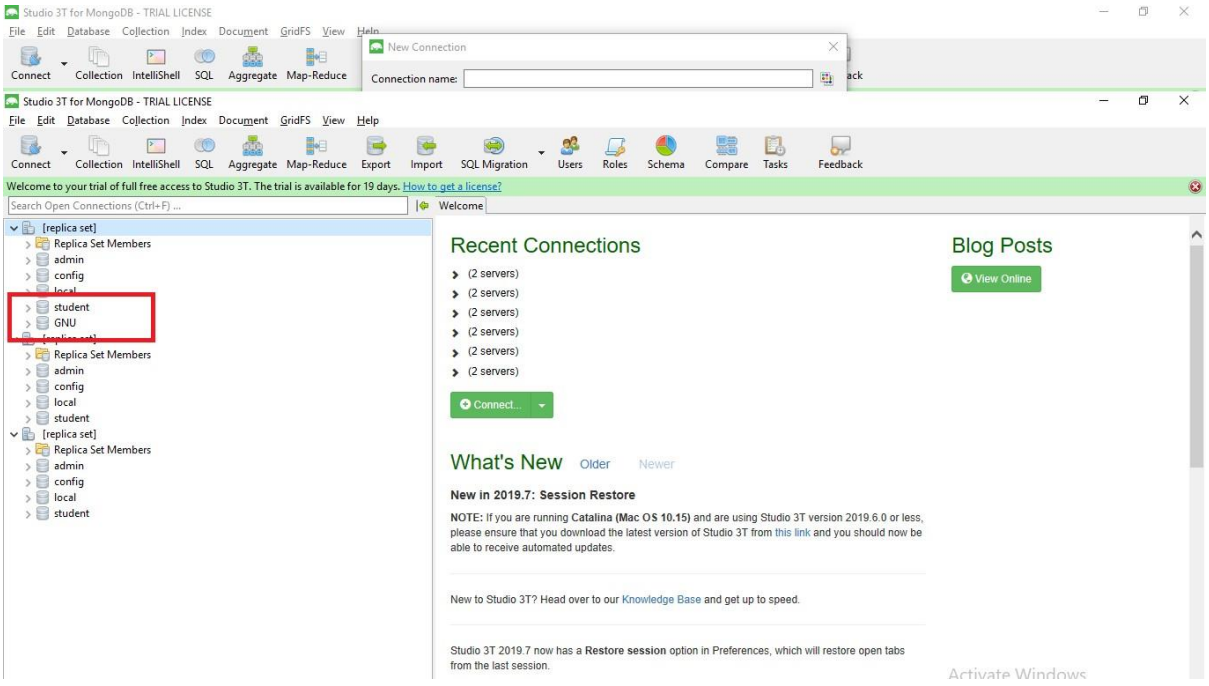


STEP 12. Create Database on Primary server.

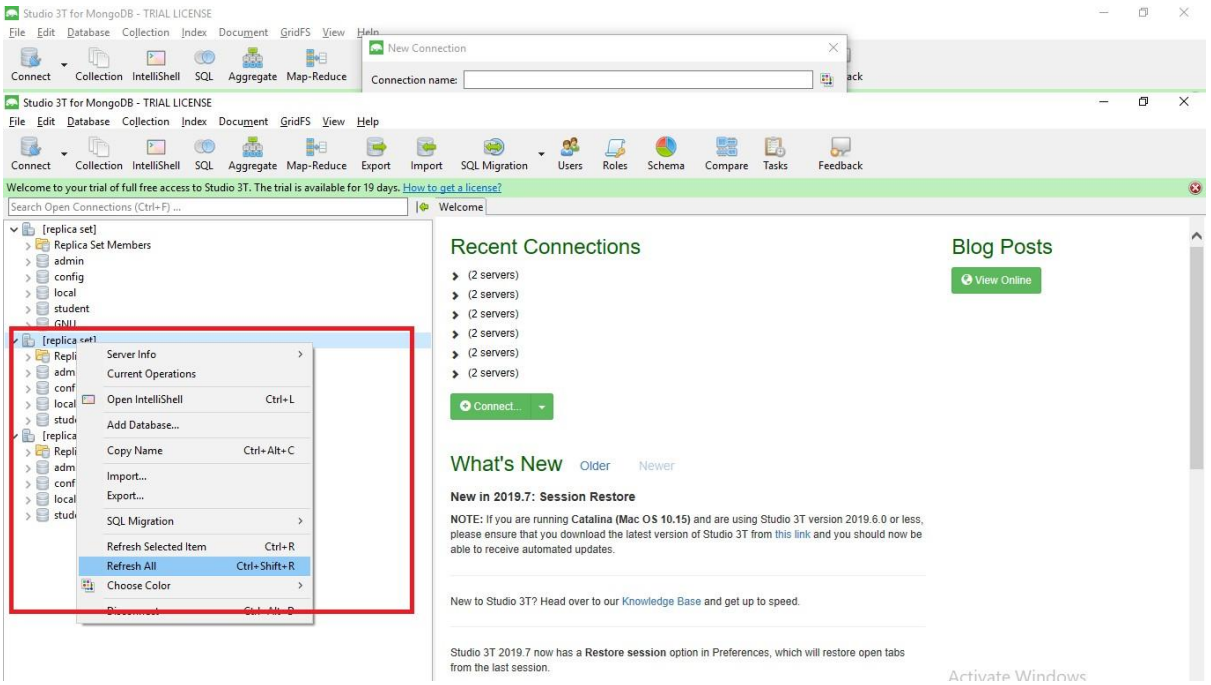


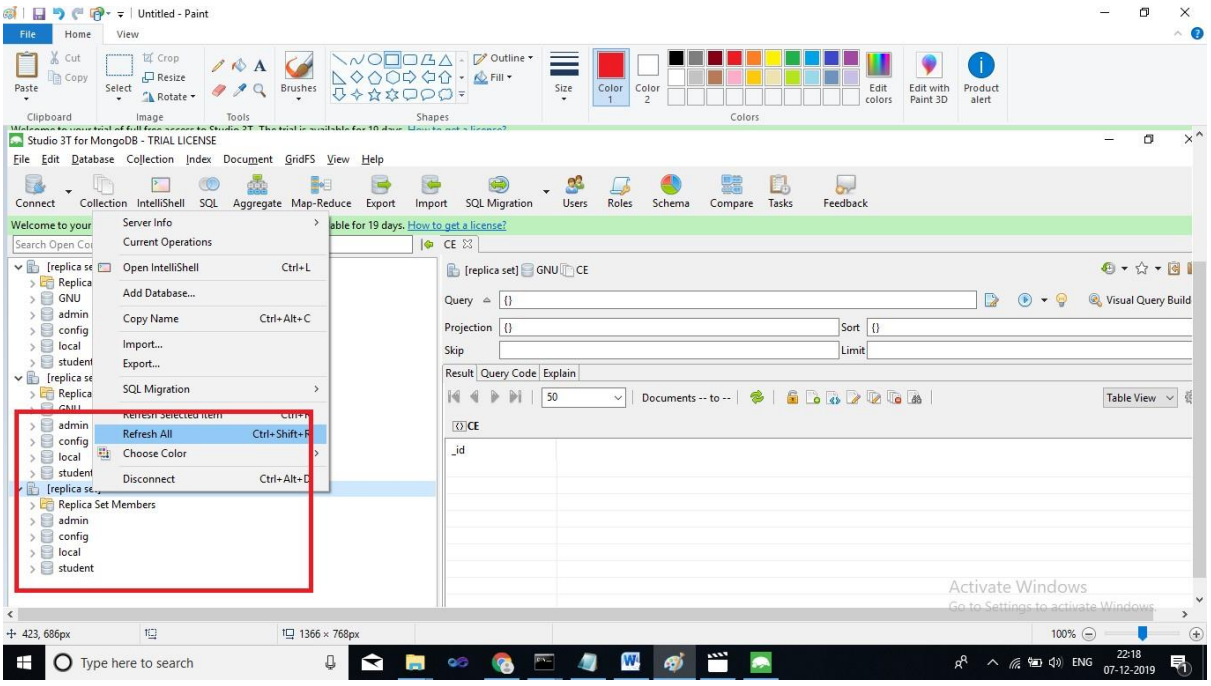


Database is created.—GNU



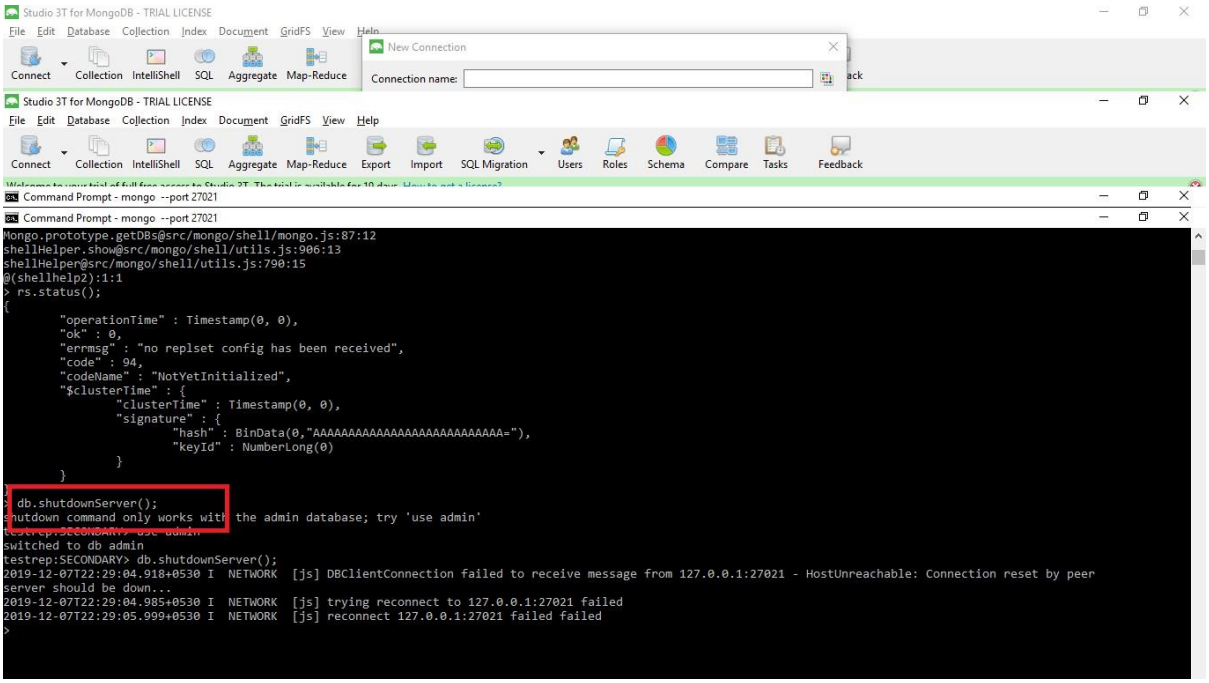
STEP 13 To Update Secondary Instances JUST PRESS REFRESH ALL OPTION.



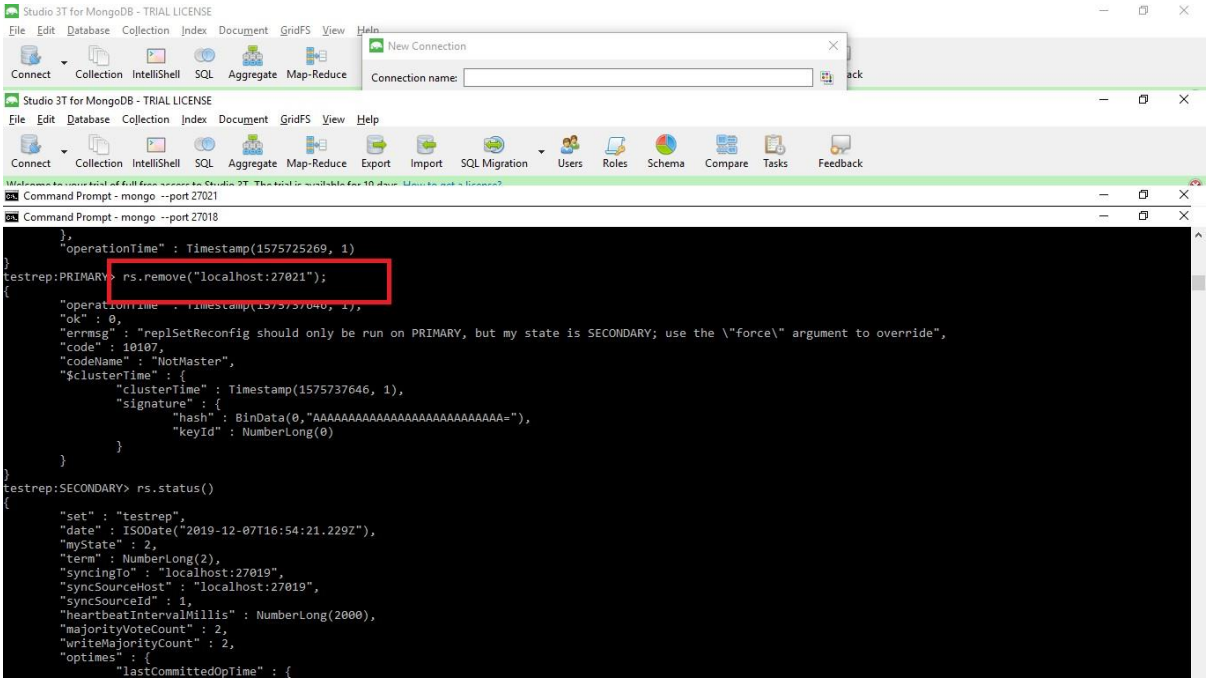


STEP 14 To REMOVE ANY Secondary Instances.

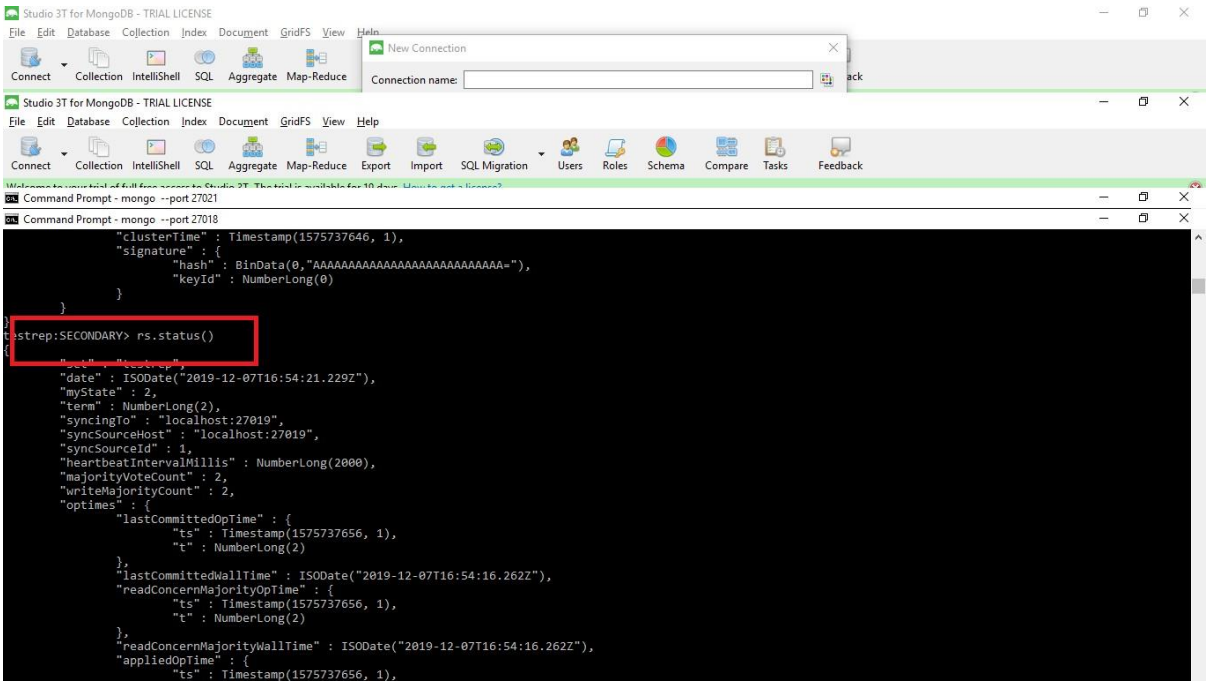
Shutdown the server which you want to remove.
Over here we remove second instance (port 27021)
From mongo client of port 27021
Syntax
db.shutdownServer();



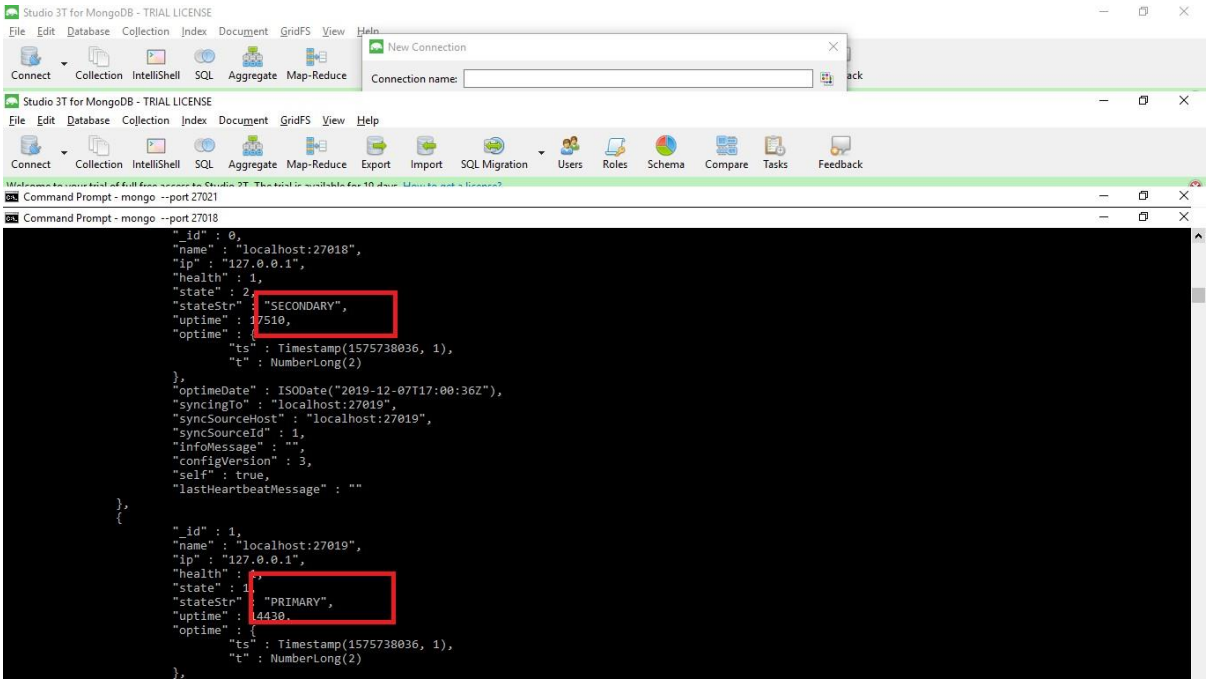
Now go to the CMD of primary clent.(Port 27018)
And write rs.remove() method to remove secondary instance from cluster.
Syntax
rs.remove("localhost:27021");



Now check status of Cluster two know member of group.
Use command rs.status();



Now in cluster there are only two instances.One is Primary and other is secondary.



To know which is Primary instance from any running client use function rs.isMaster();