PRACTICAL5

Aim - Implementation of Replicas in MongoDB.

Replication:-

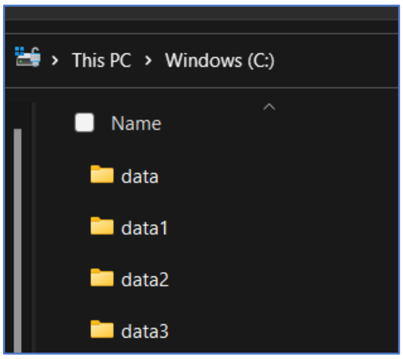
Replication is the process of synchronizing data across multiple servers. Replication provides redundancy and increases data availability with multiple copies of data on different database servers. Replication protects a database from the loss of a single server. Replication also allows you to recover 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.



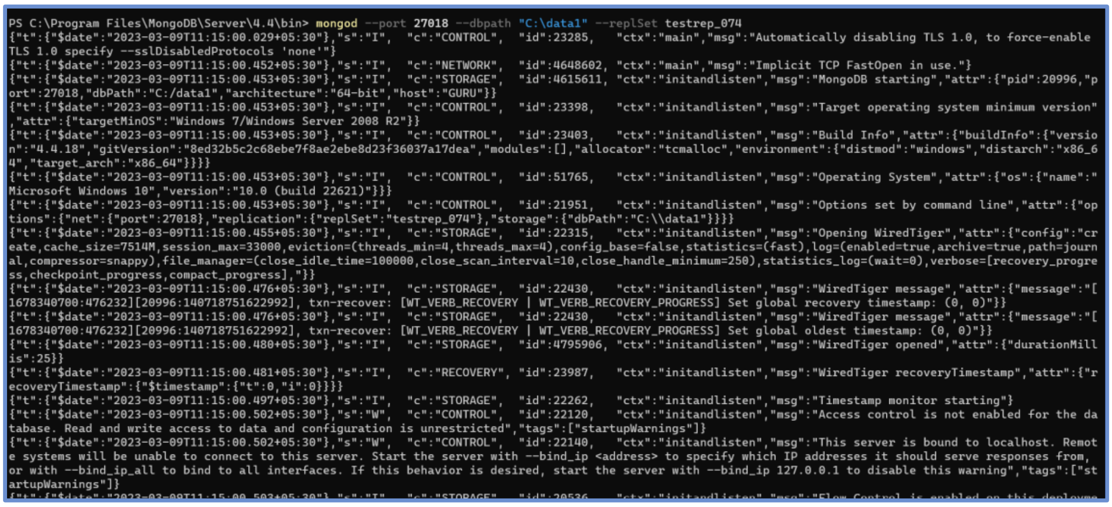
STEP-3(Creating Primary instance)

SYNTAX:

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

COMMAND:

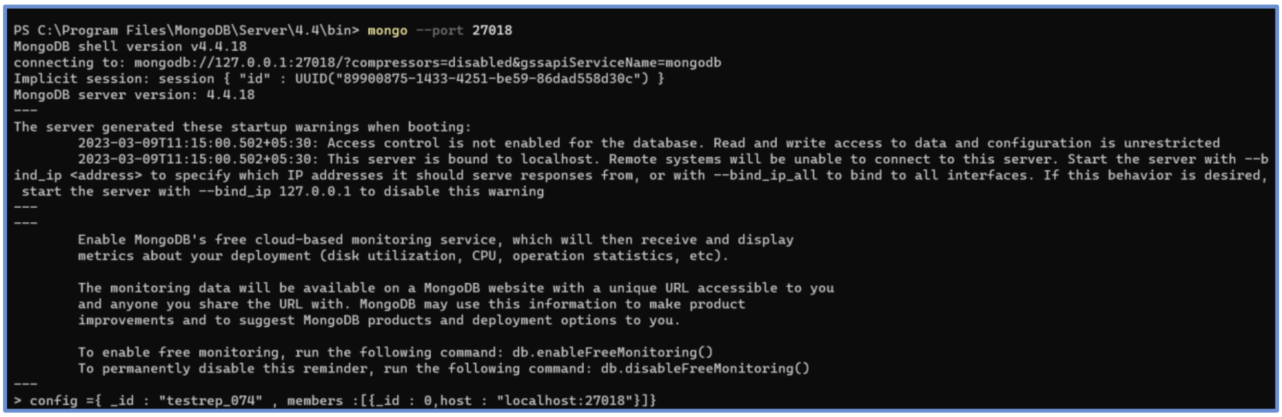
```
mongod --port 27018 --dbpath "C:\data1" --replSet testrep_074
```



```
PS C:\Program Files\MongoDB\Server\4.4\bin> mongod --port 27018 --dbpath "C:\data1" --replSet testrep_074
{"t":{"$date":"2023-03-09T11:15:00.029+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable
TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2023-03-09T11:15:00.452+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2023-03-09T11:15:00.453+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":20996,"p
ort":27018,"dbPath":"C:/data1","architecture":"64-bit","host":"GURU"}}
{"t":{"$date":"2023-03-09T11:15:00.453+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version"
,"attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2023-03-09T11:15:00.453+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"versio
n":"4.4.18","gitVersion":"8ed32b5c2c68ebe7f8ae2ebe8d23f36037a17dea","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_6
4","target_arch":"x86_64"}}}}
{"t":{"$date":"2023-03-09T11:15:00.453+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"
Microsoft Windows 10","version":"10.0 (build 22621)"}}}
{"t":{"$date":"2023-03-09T11:15:00.453+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"op
tions":{"net":{"port":27018},"replication":{"replSet":"testrep_074"},"storage":{"dbPath":"C:\\data1"}}}}
{"t":{"$date":"2023-03-09T11:15:00.455+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"cr
eate,cache_size=7514M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journ
al,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progre
ss,checkpoint_progress,compact_progress],"}}
{"t":{"$date":"2023-03-09T11:15:00.476+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[
1678340700:476232][20996:140718751622992], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}}
{"t":{"$date":"2023-03-09T11:15:00.476+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[
1678340700:476232][20996:140718751622992], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}}
{"t":{"$date":"2023-03-09T11:15:00.480+05:30"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMill
is":25}}
{"t":{"$date":"2023-03-09T11:15:00.481+05:30"},"s":"I",  "c":"RECOVERY", "id":23987,   "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"r
ecoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2023-03-09T11:15:00.497+05:30"},"s":"I",  "c":"STORAGE",  "id":22262,   "ctx":"initandlisten","msg":"Timestamp monitor starting"}
{"t":{"$date":"2023-03-09T11:15:00.502+05:30"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Access control is not enabled for the da
tabase. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2023-03-09T11:15:00.502+05:30"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"initandlisten","msg":"This server is bound to localhost. Remot
e systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from,
or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["st
artupWarnings"]}
{"t":{"$date":"2023-03-09T11:15:00.503+05:30"},"s":"I",  "c":"STORAGE",  "id":20536,   "ctx":"initandlisten","msg":"Flow Control is enabled on this deployme
```

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

```
mongo –port 27018
```

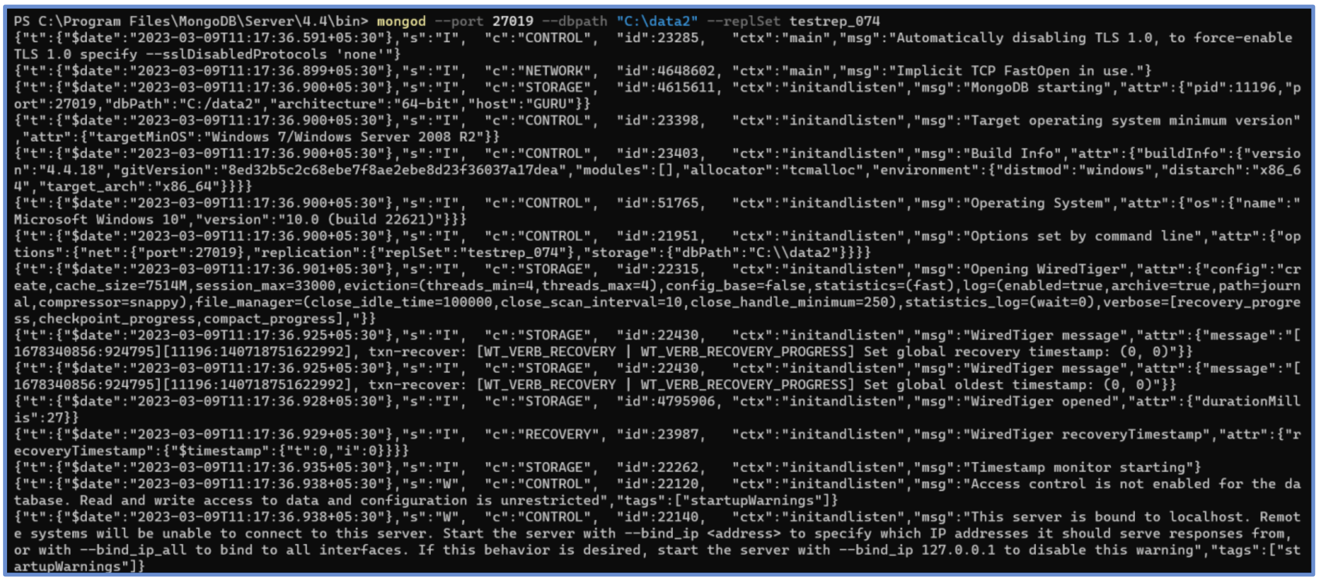


```
PS C:\Program Files\MongoDB\Server\4.4\bin> mongo --port 27018
MongoDB shell version v4.4.18
connecting to: mongodb://127.0.0.1:27018/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("89900875-1433-4251-be59-86dad558d30c") }
MongoDB server version: 4.4.18
---
The server generated these startup warnings when booting:
        2023-03-09T11:15:00.502+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2023-03-09T11:15:00.502+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --b
ind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired,
 start the server with --bind_ip 127.0.0.1 to disable this warning
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> config ={ _id : "testrep_074" , members :[{_id : 0,host : "localhost:27018"}]}
```

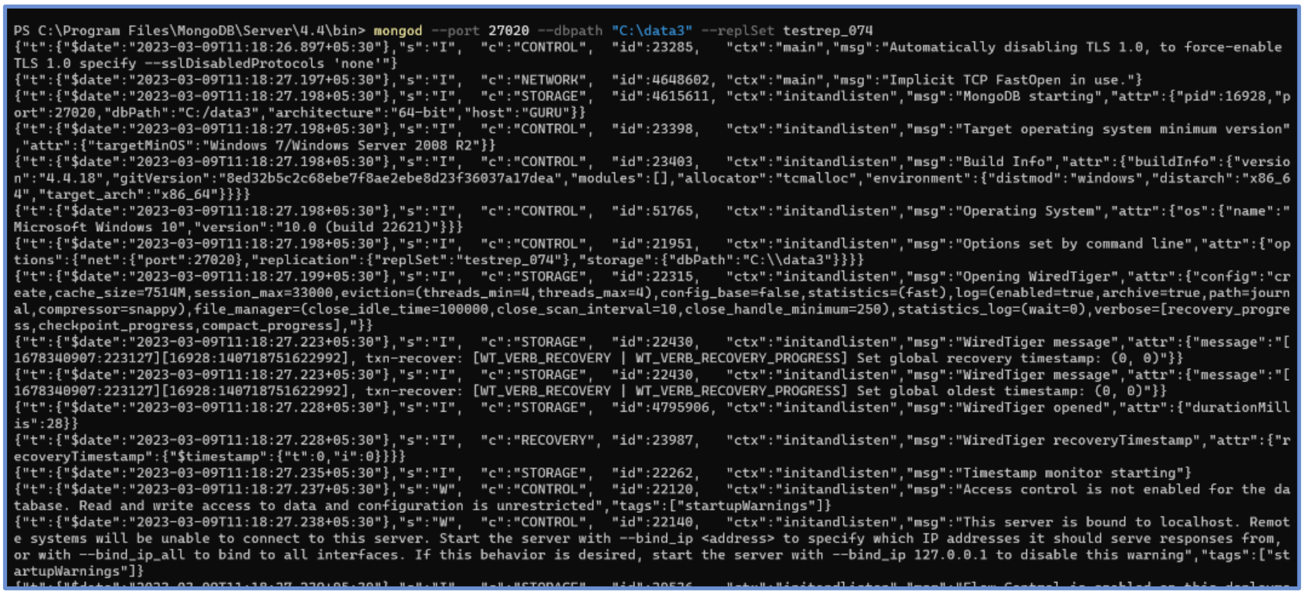
Now primary server is running

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
```



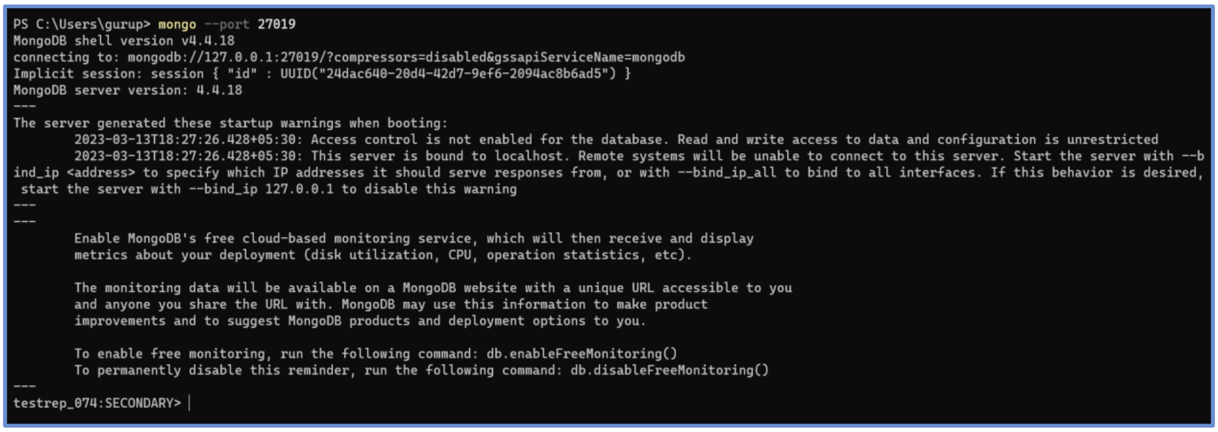
```
PS C:\Program Files\MongoDB\Server\4.4\bin> mongod --port 27019 --dbpath "C:\data2" --replSet testrep_074
{"t":{"$date":"2023-03-09T11:17:36.591+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable
TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2023-03-09T11:17:36.899+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2023-03-09T11:17:36.900+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":11196,"p
ort":27019,"dbPath":"C:/data2","architecture":"64-bit","host":"GURU"}}
{"t":{"$date":"2023-03-09T11:17:36.900+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version"
,"attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2023-03-09T11:17:36.900+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"versio
n":"4.4.18","gitVersion":"8ed32b5c2c68ebe7f8ae2ebe8d23f36037a17dea","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_6
4","target_arch":"x86_64"}}}}
{"t":{"$date":"2023-03-09T11:17:36.900+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"
Microsoft Windows 10","version":"10.0 (build 22621)"}}}
{"t":{"$date":"2023-03-09T11:17:36.900+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"op
tions":{"net":{"port":27019},"replication":{"replSet":"testrep_074"},"storage":{"dbPath":"C:\\data2"}}}}
{"t":{"$date":"2023-03-09T11:17:36.901+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"cr
eate,cache_size=7514M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journ
al,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progre
ss,checkpoint_progress,compact_progress],"}}
{"t":{"$date":"2023-03-09T11:17:36.925+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[
1678340856:924795][11196:140718751622992], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}}
{"t":{"$date":"2023-03-09T11:17:36.925+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[
1678340856:924795][11196:140718751622992], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}}
{"t":{"$date":"2023-03-09T11:17:36.928+05:30"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMill
is":27}}
{"t":{"$date":"2023-03-09T11:17:36.929+05:30"},"s":"I",  "c":"RECOVERY", "id":23987,   "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"r
ecoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2023-03-09T11:17:36.935+05:30"},"s":"I",  "c":"STORAGE",  "id":22262,   "ctx":"initandlisten","msg":"Timestamp monitor starting"}
{"t":{"$date":"2023-03-09T11:17:36.938+05:30"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Access control is not enabled for the da
tabase. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2023-03-09T11:17:36.938+05:30"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"initandlisten","msg":"This server is bound to localhost. Remot
e systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from,
or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["st
artupWarnings"]}
```



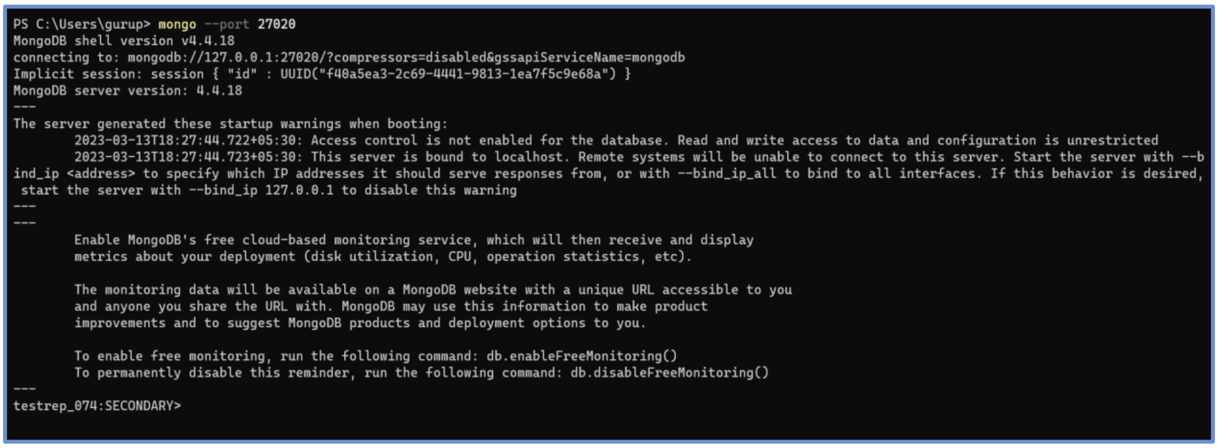
```
PS C:\Program Files\MongoDB\Server\4.4\bin> mongod --port 27020 --dbpath "C:\data3" --replSet testrep_074
{"t":{"$date":"2023-03-09T11:18:26.897+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable
TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2023-03-09T11:18:27.197+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2023-03-09T11:18:27.198+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":16928,"p
ort":27020,"dbPath":"C:/data3","architecture":"64-bit","host":"GURU"}}
{"t":{"$date":"2023-03-09T11:18:27.198+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version"
,"attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2023-03-09T11:18:27.198+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"versio
n":"4.4.18","gitVersion":"8ed32b5c2c68ebe7f8ae2ebe8d23f36037a17dea","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_6
4","target_arch":"x86_64"}}}}
{"t":{"$date":"2023-03-09T11:18:27.198+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"
Microsoft Windows 10","version":"10.0 (build 22621)"}}}
{"t":{"$date":"2023-03-09T11:18:27.198+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"op
tions":{"net":{"port":27020},"replication":{"replSet":"testrep_074"},"storage":{"dbPath":"C:\\data3"}}}}
{"t":{"$date":"2023-03-09T11:18:27.199+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"cr
eate,cache_size=7514M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journ
al,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progre
ss,checkpoint_progress,compact_progress],"}}
{"t":{"$date":"2023-03-09T11:18:27.223+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[
1678340907:223127][16928:140718751622992], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}}
{"t":{"$date":"2023-03-09T11:18:27.223+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[
1678340907:223127][16928:140718751622992], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}}
{"t":{"$date":"2023-03-09T11:18:27.228+05:30"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMill
is":28}}
{"t":{"$date":"2023-03-09T11:18:27.228+05:30"},"s":"I",  "c":"RECOVERY", "id":23987,   "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"r
ecoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2023-03-09T11:18:27.235+05:30"},"s":"I",  "c":"STORAGE",  "id":22262,   "ctx":"initandlisten","msg":"Timestamp monitor starting"}
{"t":{"$date":"2023-03-09T11:18:27.237+05:30"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Access control is not enabled for the da
tabase. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2023-03-09T11:18:27.238+05:30"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"initandlisten","msg":"This server is bound to localhost. Remot
e systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from,
or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["st
artupWarnings"]}
```

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

```
mongo --port 27019

mongo –port 27020
```



```
PS C:\Users\gurup> mongo --port 27019
MongoDB shell version v4.4.18
connecting to: mongodb://127.0.0.1:27019/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("24dac640-20d4-42d7-9ef6-2094ac8b6ad5") }
MongoDB server version: 4.4.18
---
The server generated these startup warnings when booting:
        2023-03-13T18:27:26.428+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2023-03-13T18:27:26.428+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --b
ind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired,
 start the server with --bind_ip 127.0.0.1 to disable this warning
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
testrep_074:SECONDARY>
```



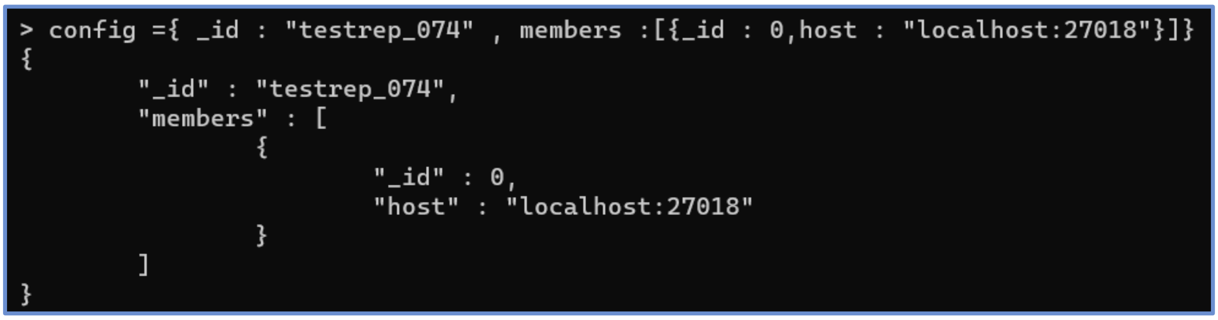
```
PS C:\Users\gurup> mongo --port 27020
MongoDB shell version v4.4.18
connecting to: mongodb://127.0.0.1:27020/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("f40a5ea3-2c69-4441-9813-1ea7f5c9e68a") }
MongoDB server version: 4.4.18
---
The server generated these startup warnings when booting:
        2023-03-13T18:27:44.722+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2023-03-13T18:27:44.723+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --b
ind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired,
 start the server with --bind_ip 127.0.0.1 to disable this warning
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
testrep_074:SECONDARY>
```

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

C:\>mongo --port 27018

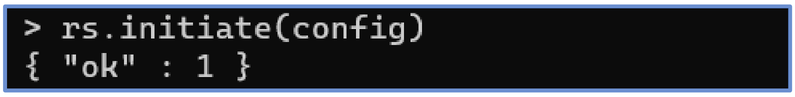
Now type the following code :

```
config ={ _id : "testrep_074" , members :[ { _id : 0, host : "localhost:27018" } ] }
```



```
> config ={ _id : "testrep_074" , members :[{_id : 0,host : "localhost:27018"}]}
{
        "_id" : "testrep_074",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27018"
                }
        ]
}
```

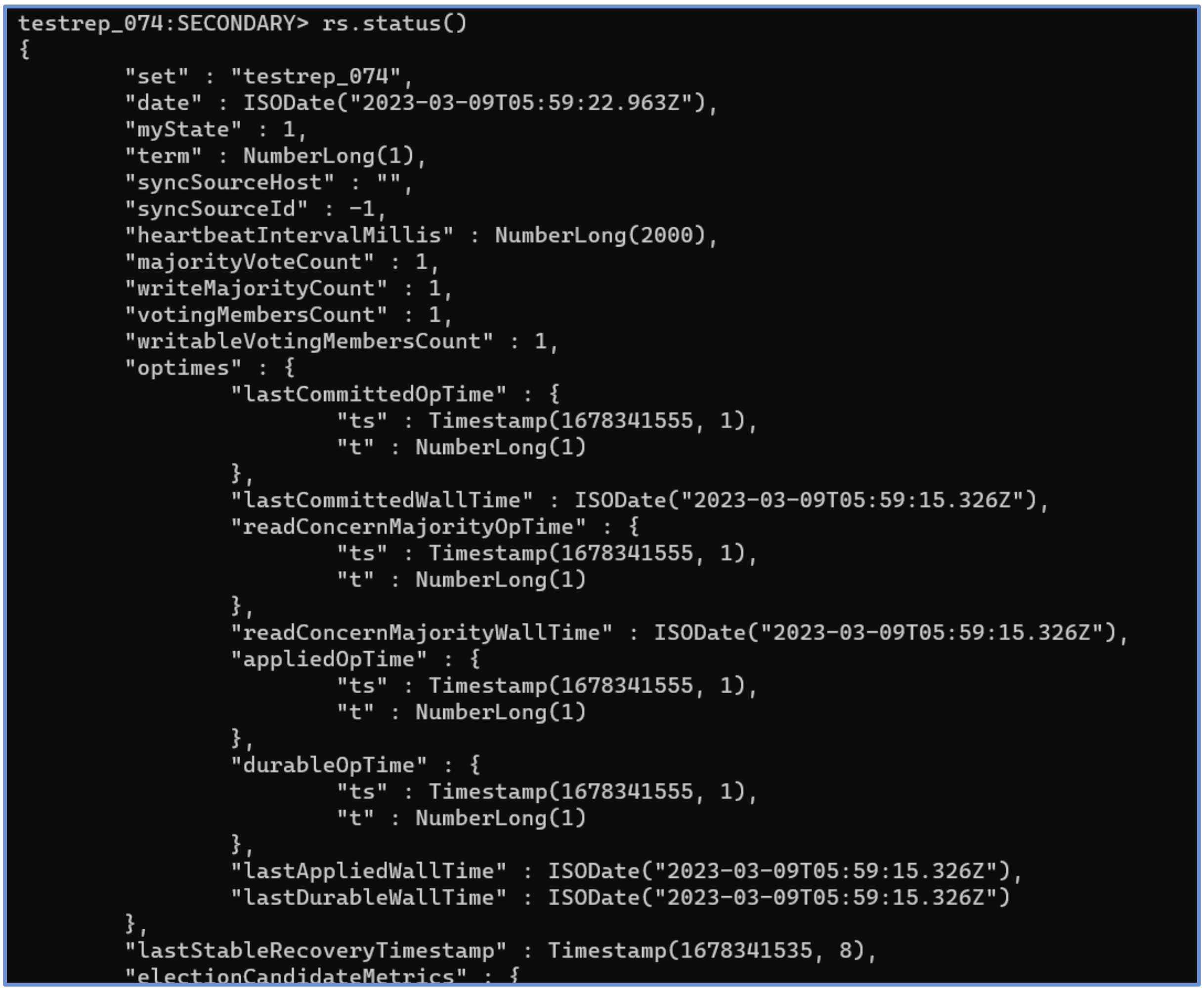
After this write command `rs.initiate(config)` This command initiates a replica set with the current host as its only member. This is confirmed by the output, which should resemble the following:



```
> rs.initiate(config)
{ "ok" : 1 }
```

After this write command :

```
rs.status()
```

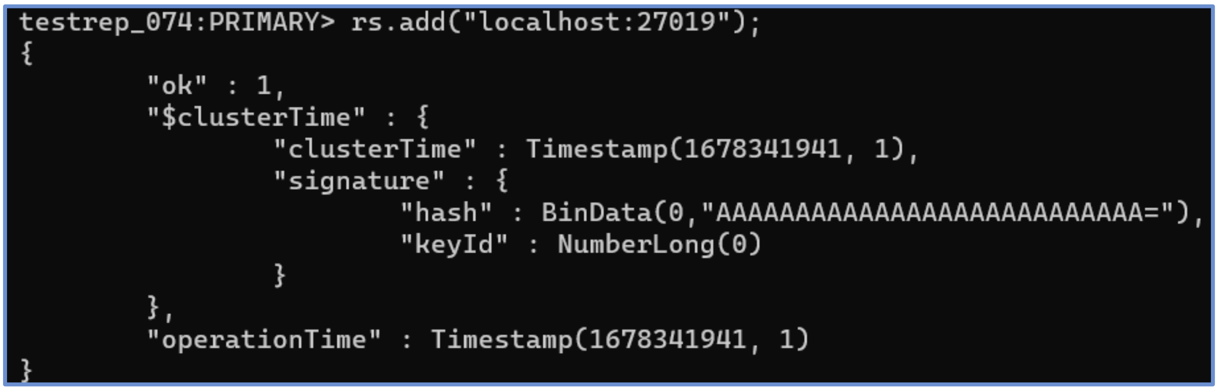


```
testrep_074:SECONDARY> rs.status()
{
        "set" : "testrep_074",
        "date" : ISODate("2023-03-09T05:59:22.963Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 1,
        "writeMajorityCount" : 1,
        "votingMembersCount" : 1,
        "writableVotingMembersCount" : 1,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1678341555, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2023-03-09T05:59:15.326Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1678341555, 1),
                        "t" : NumberLong(1)
                },
                "readConcernMajorityWallTime" : ISODate("2023-03-09T05:59:15.326Z"),
                "appliedOpTime" : {
                        "ts" : Timestamp(1678341555, 1),
                        "t" : NumberLong(1)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1678341555, 1),
                        "t" : NumberLong(1)
                },
                "lastAppliedWallTime" : ISODate("2023-03-09T05:59:15.326Z"),
                "lastDurableWallTime" : ISODate("2023-03-09T05:59:15.326Z")
        },
        "lastStableRecoveryTimestamp" : Timestamp(1678341535, 8),
        "electionCandidateMetrics" : {
```

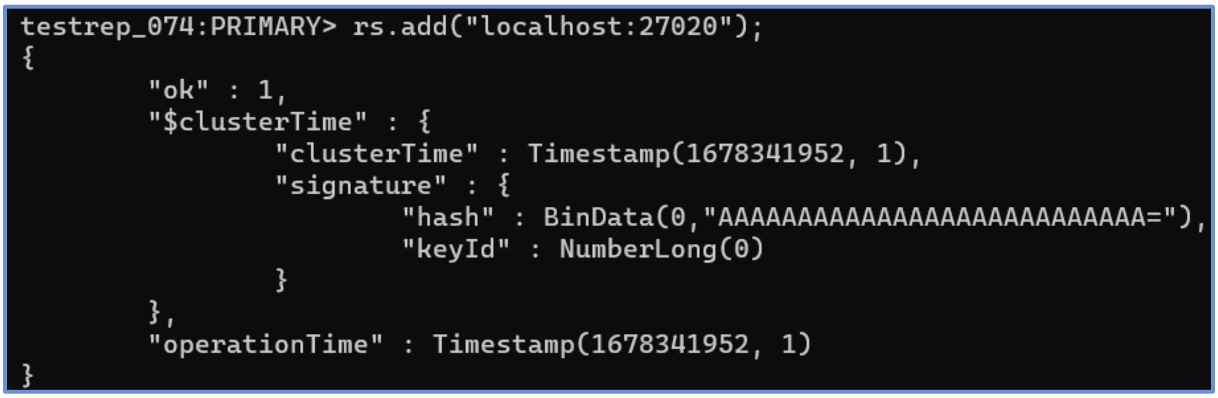
Step : 6 Now go to the cmd client window of primary instance.

Write the following code there :-

```
rs.add("localhost:27019");

rs.add("localhost:27020");
```



```
testrep_074:PRIMARY> rs.add("localhost:27019");
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1678341941, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1678341941, 1)
}
```

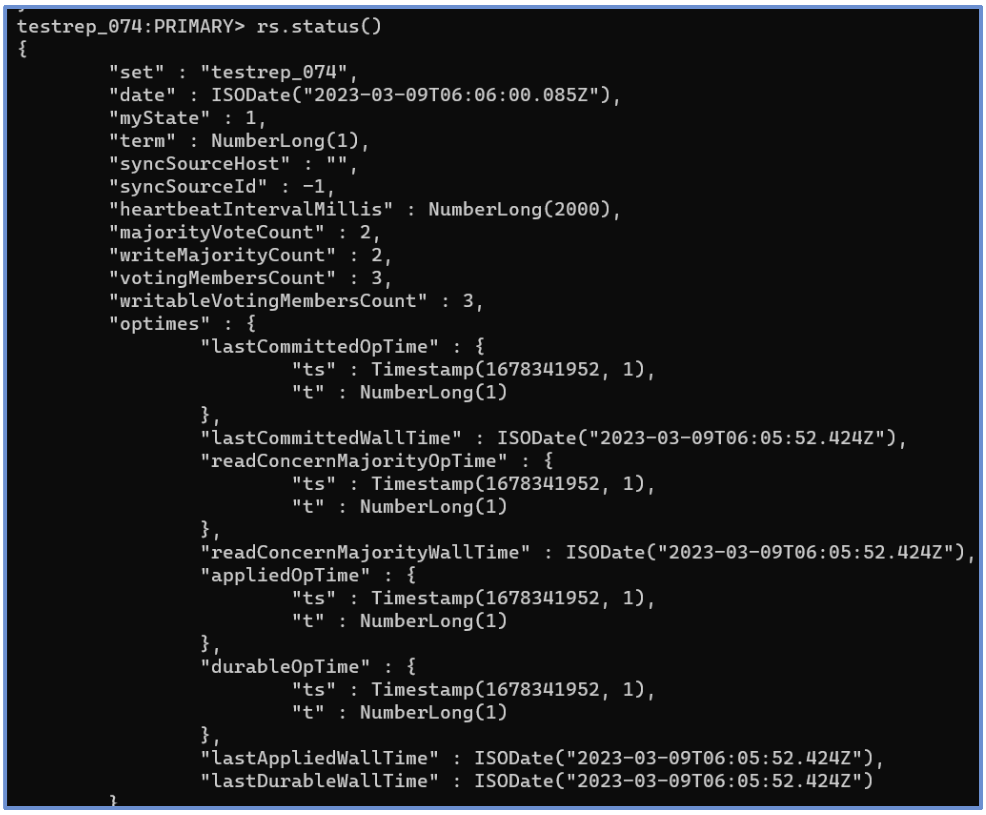


```
testrep_074:PRIMARY> rs.add("localhost:27020");
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1678341952, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1678341952, 1)
}
```

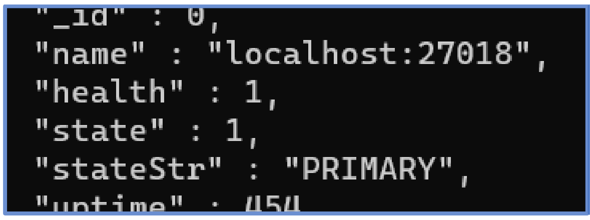
- Now again check the status using following code :-

```
rs.status()
```

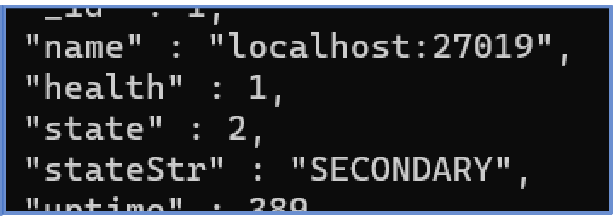
This time it shows Primary instance and secondary instances two replicas of primary instance.



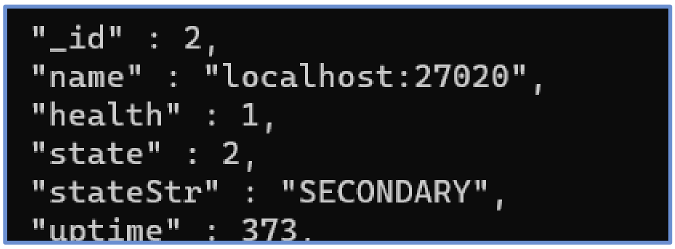
```
testrep_074:PRIMARY> rs.status()
{
        "set" : "testrep_074",
        "date" : ISODate("2023-03-09T06:06:00.085Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 2,
        "writeMajorityCount" : 2,
        "votingMembersCount" : 3,
        "writableVotingMembersCount" : 3,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1678341952, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2023-03-09T06:05:52.424Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1678341952, 1),
                        "t" : NumberLong(1)
                },
                "readConcernMajorityWallTime" : ISODate("2023-03-09T06:05:52.424Z"),
                "appliedOpTime" : {
                        "ts" : Timestamp(1678341952, 1),
                        "t" : NumberLong(1)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1678341952, 1),
                        "t" : NumberLong(1)
                },
                "lastAppliedWallTime" : ISODate("2023-03-09T06:05:52.424Z"),
                "lastDurableWallTime" : ISODate("2023-03-09T06:05:52.424Z")
        }
```



```
"_id" : 0,
"name" : "localhost:27018",
"health" : 1,
"state" : 1,
"stateStr" : "PRIMARY",
"uptime" : 454
```



```
"_id" : 1,
"name" : "localhost:27019",
"health" : 1,
"state" : 2,
"stateStr" : "SECONDARY",
"uptime" : 389
```



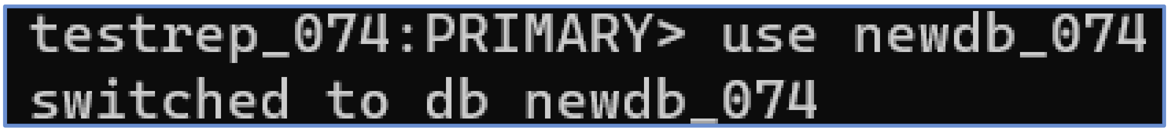
```
"_id" : 2,
"name" : "localhost:27020",
"health" : 1,
"state" : 2,
"stateStr" : "SECONDARY",
"uptime" : 373,
```

<u>Step : 7</u> Now create a new database and new collection in primary server.

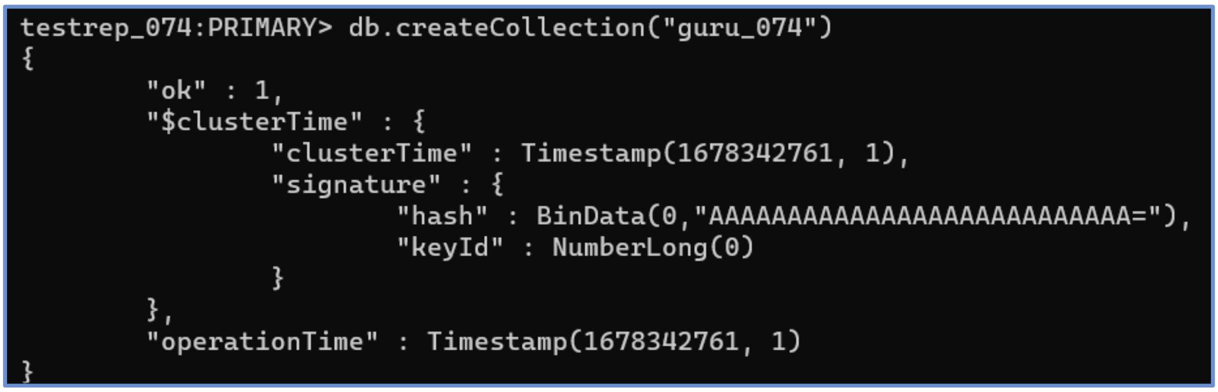
```
use newdb_072
db.createCollection(“divy_072”)
```

Also insert some data in that collection

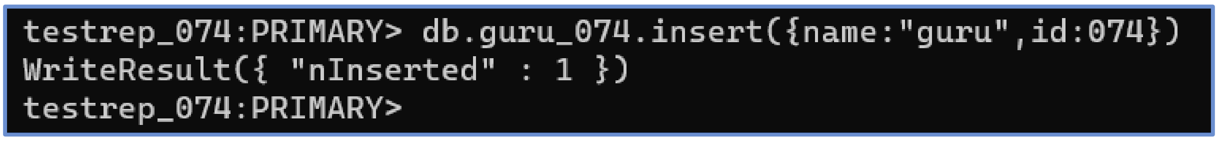
```
db.divy_072.insert({name : “abc”, number : 123})
```



```
testrep_074:PRIMARY> use newdb_074
switched to db newdb_074
```

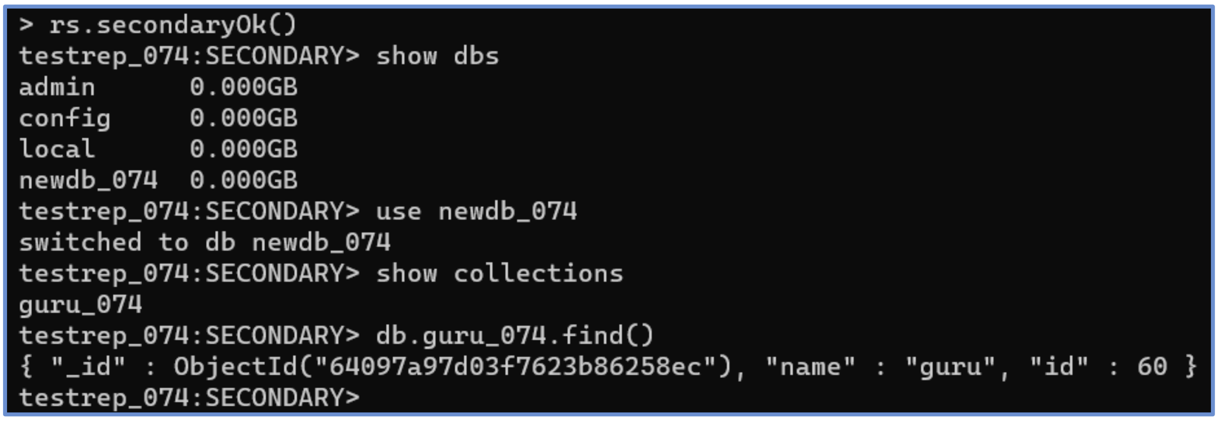


```
testrep_074:PRIMARY> db.createCollection("guru_074")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1678342761, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1678342761, 1)
}
```



```
testrep_074:PRIMARY> db.guru_074.insert({name:"guru",id:074})
WriteResult({ "nInserted" : 1 })
testrep_074:PRIMARY>
```

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



```
> rs.secondaryOk()
testrep_074:SECONDARY> show dbs
admin       0.000GB
config      0.000GB
local       0.000GB
newdb_074   0.000GB
testrep_074:SECONDARY> use newdb_074
switched to db newdb_074
testrep_074:SECONDARY> show collections
guru_074
testrep_074:SECONDARY> db.guru_074.find()
{ "_id" : ObjectId("64097a97d03f7623b86258ec"), "name" : "guru", "id" : 60 }
testrep_074:SECONDARY>
```