Practical-5 Replication

Aim - Implementation of Replicas in MongoDB.

Summary:

Replication is the process of synchronizing data across multiple servers.

Replication is referred to the process of ensuring that the same data is available on more than one Mongo DB Server. This is sometimes required for the purpose of increasing data availability. 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. Because if your main MongoDB Server goes down for any reason, there will be no access to the data. But if you had the data replicated to another server at regular intervals, you will be able to access the data from another server even if the primary server fails. MongoDB achieves replication by the use of replica set. A replica set is a group of mongod instances that host 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).

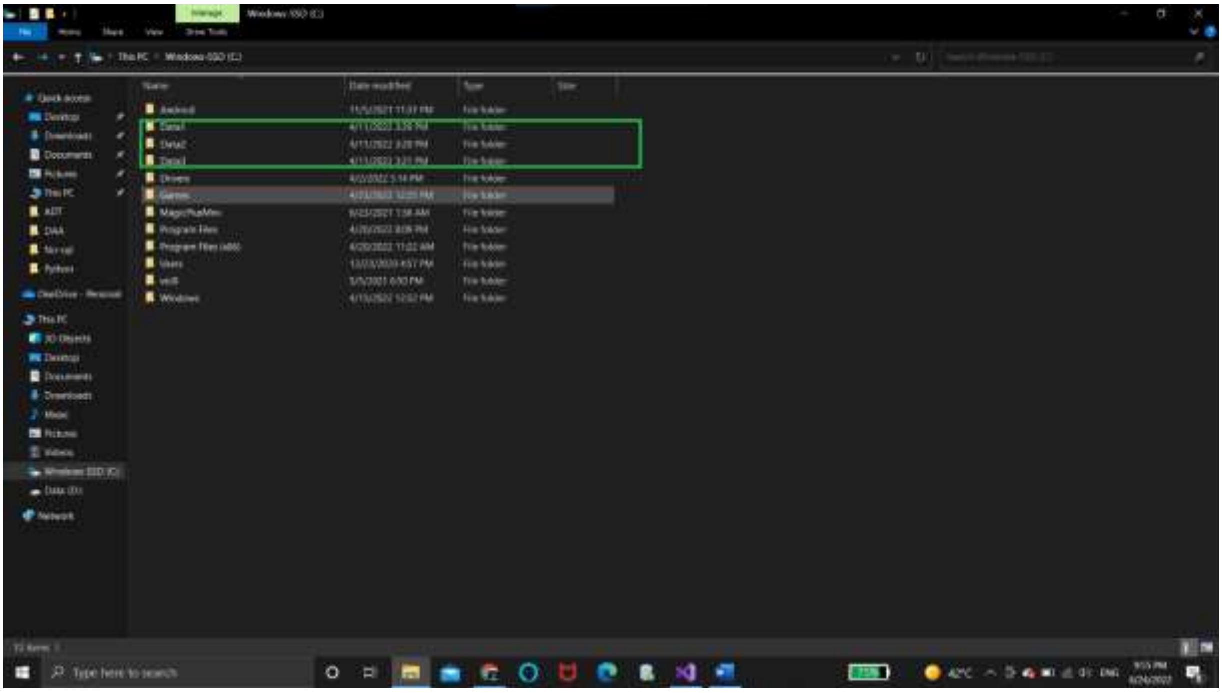
Example how to Set Up a Replica Set

Step 1:

Shutdown already running MongoDB server.

Step 2:

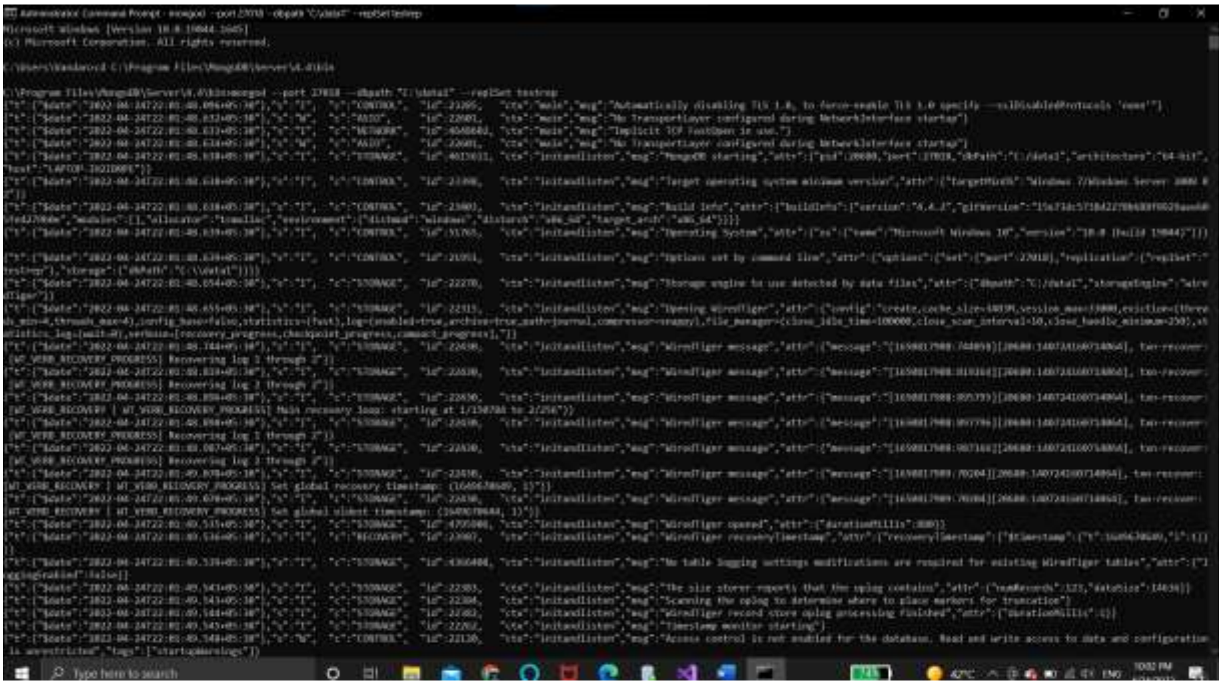
Create 3 folders with name data1,data2,data3 at any location.


This PC > Windows-SSD (C:)

Name	Date modified	Type	Size
Android	11/5/2021 11:37 PM	File folder	
Data1	4/11/2022 3:20 PM	File folder	
Data2	4/11/2022 3:20 PM	File folder	
Data3	4/11/2022 3:21 PM	File folder	
Drivers	4/22/2022 5:14 PM	File folder	
Games	4/23/2022 12:23 PM	File folder	
MagicPutMini	6/22/2021 1:58 AM	File folder	
Program Files	4/20/2022 8:09 PM	File folder	
Program Files (x86)	4/20/2022 11:02 AM	File folder	
Users	12/23/2020 4:57 PM	File folder	
vcl8	5/5/2021 6:00 PM	File folder	
Windows	4/15/2022 12:02 PM	File folder	

Step 3: (Creating Primary instance)

First change the cmd path.


Administrator: Command Prompt - mongod --port 27018 --dbpath "C:\data1" --replSet testrep

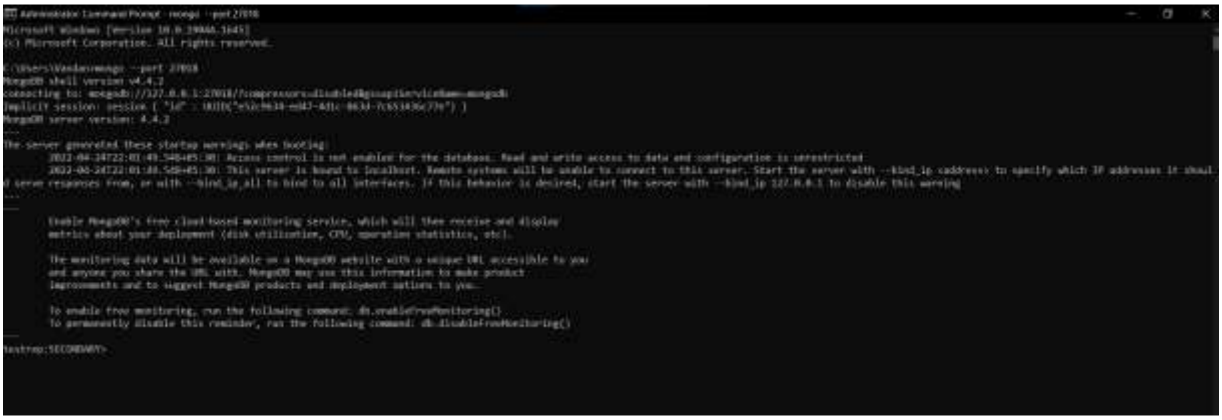
```
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vandan>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongod --port 27018 --dbpath "C:\data1" --replSet testrep
{"t":{"$date":"2022-04-24T22:01:48.096+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-04-24T22:01:48.632+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-24T22:01:48.633+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2022-04-24T22:01:48.634+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-24T22:01:48.638+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":20680,"port":27018,"dbPath":"C:/data1","architecture":"64-bit","host":"LAPTOP-[illegible]"}}
{"t":{"$date":"2022-04-24T22:01:48.638+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":{"$date":"2022-04-24T22:01:48.638+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"4.4.2","gitVersion":"[illegible]","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2022-04-24T22:01:48.639+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19044)"}}}
{"t":{"$date":"2022-04-24T22:01:48.639+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{"net":{"port":27018},"replication":{"replSet":"testrep"},"storage":{"dbPath":"C:\\data1"}}}}
{"t":{"$date":"2022-04-24T22:01:48.654+05:30"},"s":"I",  "c":"STORAGE",  "id":22270,   "ctx":"initandlisten","msg":"Storage engine to use detected by data files","attr":{"dbpath":"C:/data1","storageEngine":"wiredTiger"}}
{"t":{"$date":"2022-04-24T22:01:48.655+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"create,cache_size=[illegible],session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress,compact_progress],"}}
{"t":{"$date":"2022-04-24T22:01:48.[illegible]"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[illegible] [WT_VERB_RECOVERY_PROGRESS] Recovering log 1 through 2"}}
{"t":{"$date":"2022-04-24T22:01:48.[illegible]"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[illegible] [WT_VERB_RECOVERY_PROGRESS] Recovering log 2 through 2"}}
{"t":{"$date":"2022-04-24T22:01:48.[illegible]"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[illegible] [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Main recovery loop: starting at 1/[illegible] to 2/256"}}
{"t":{"$date":"2022-04-24T22:01:48.[illegible]"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[illegible] [WT_VERB_RECOVERY_PROGRESS] Recovering log 1 through 2"}}
{"t":{"$date":"2022-04-24T22:01:48.[illegible]"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[illegible] [WT_VERB_RECOVERY_PROGRESS] Recovering log 2 through 2"}}
{"t":{"$date":"2022-04-24T22:01:48.[illegible]"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[illegible] [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (1649678649, 1)"}}
{"t":{"$date":"2022-04-24T22:01:48.[illegible]"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[illegible] [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (1649678644, 1)"}}
{"t":{"$date":"2022-04-24T22:01:49.535+05:30"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMillis":[illegible]}}
{"t":{"$date":"2022-04-24T22:01:49.536+05:30"},"s":"I",  "c":"RECOVERY", "id":23987,   "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"recoveryTimestamp":{"$timestamp":{"t":1649678649,"i":1}}}}
{"t":{"$date":"2022-04-24T22:01:49.539+05:30"},"s":"I",  "c":"STORAGE",  "id":4366408, "ctx":"initandlisten","msg":"No table logging settings modifications are required for existing WiredTiger tables","attr":{"loggingEnabled":false}}
{"t":{"$date":"2022-04-24T22:01:49.543+05:30"},"s":"I",  "c":"STORAGE",  "id":22383,   "ctx":"initandlisten","msg":"The size storer reports that the oplog contains","attr":{"numRecords":123,"dataSize":[illegible]}}
{"t":{"$date":"2022-04-24T22:01:49.543+05:30"},"s":"I",  "c":"STORAGE",  "id":22384,   "ctx":"initandlisten","msg":"Scanning the oplog to determine where to place markers for truncation"}
{"t":{"$date":"2022-04-24T22:01:49.[illegible]"},"s":"I",  "c":"STORAGE",  "id":22382,   "ctx":"initandlisten","msg":"WiredTiger record store oplog processing finished","attr":{"durationMillis":[illegible]}}
{"t":{"$date":"2022-04-24T22:01:49.545+05:30"},"s":"I",  "c":"STORAGE",  "id":22262,   "ctx":"initandlisten","msg":"Timestamp monitor starting"}
{"t":{"$date":"2022-04-24T22:01:49.548+05:30"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted.","tags":["startupWarnings"]}
```

Start MongoDB server and write the following query with proper file path.

mongod --port "PORT" --dbpath "YOUR_DB_DATA_PATH" --replSet "REPLICA_SET_INSTANCE_NAME"6

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

C:\>mongo --port 27018

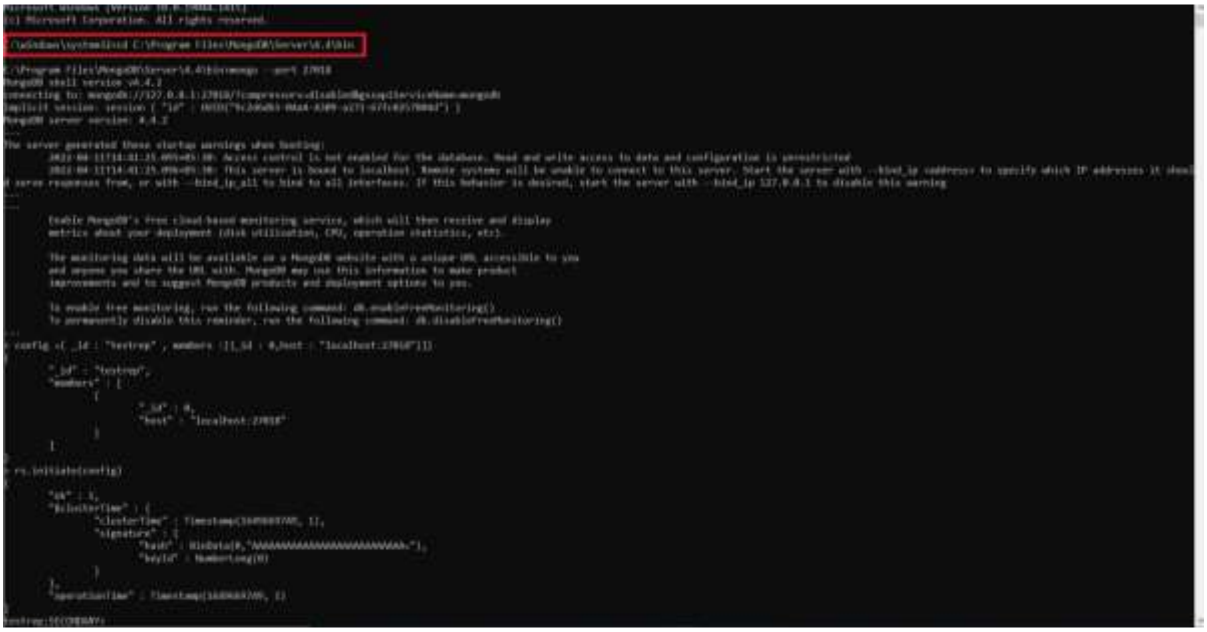


```
Administrator: Command Prompt - mongo --port 27018
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vandan>mongo --port 27018
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27018/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("[illegible]") }
MongoDB server version: 4.4.2
---
The server generated these startup warnings when booting:
        2022-04-24T22:01:49.[illegible]+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2022-04-24T22:01:48.[illegible]+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it shoul
d 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:SECONDARY>
```

Step-4

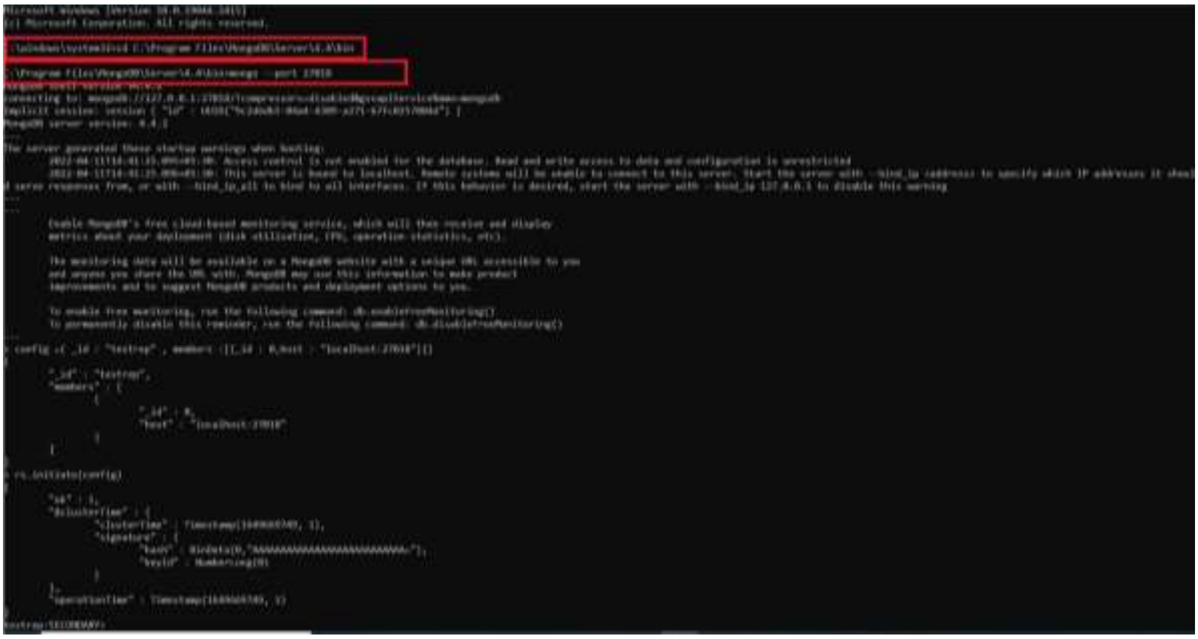
Open another cmd and in that write the following query.

First change the cmd path.



```
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongo --port 27018
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27018/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("[illegible]") }
MongoDB server version: 4.4.2
---
The server generated these startup warnings when booting:
        2022-04-11T14:41:25.[illegible]+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2022-04-11T14:41:25.[illegible]+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it shoul
d 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" , members :[{_id : 0,host : "localhost:27018"}]}
{
        "_id" : "testrep",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27018"
                }
        ]
}
> rs.initiate(config)
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp([illegible], 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp([illegible], 1)
}
testrep:SECONDARY>
```

mongo --port 27018



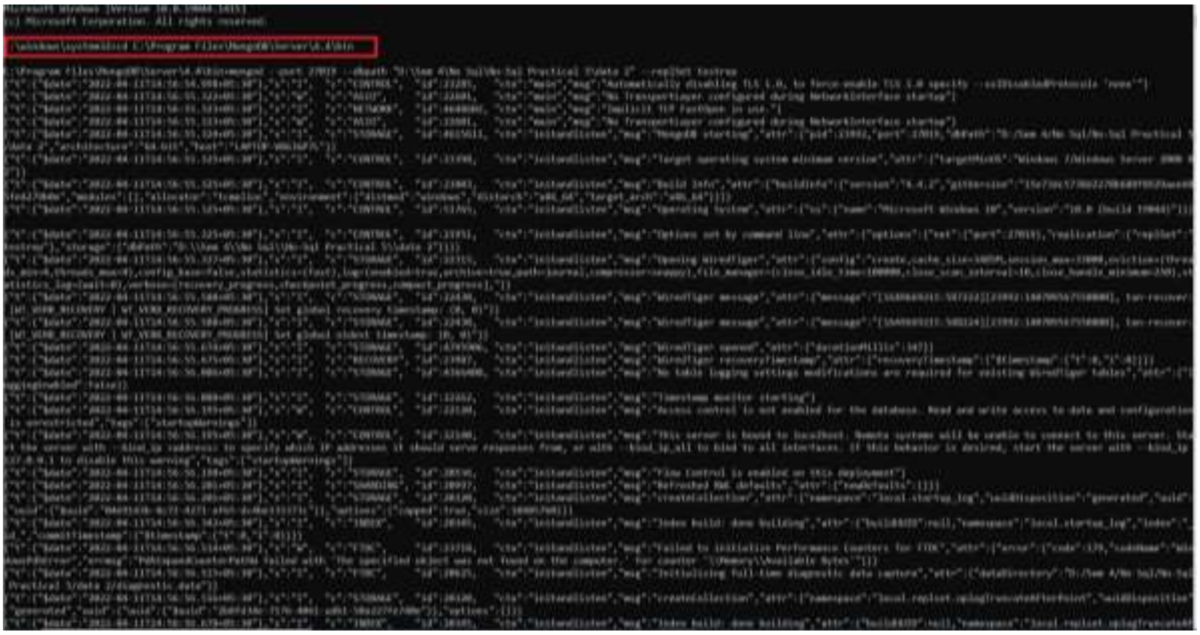
```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongo --port 27018
MongoDB shell version v4.4.1
connecting to: mongodb://127.0.0.1:27018/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("[illegible]") }
MongoDB server version: 4.4.1
---
The server generated these startup warnings when booting:
        [illegible]: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        [illegible]: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it shoul
d 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" , members :[{_id : 0,host : "localhost:27018"}]}
{
        "_id" : "testrep",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27018"
                }
        ]
}
> rs.initiate(config)
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp([illegible], 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp([illegible], 1)
}
testrep:SECONDARY>
```

Now our primary server is running on port 27018.

- Now repeat step on 3 for starting secondary server but this time we have to give file path of folder data2/data3 we made earlier

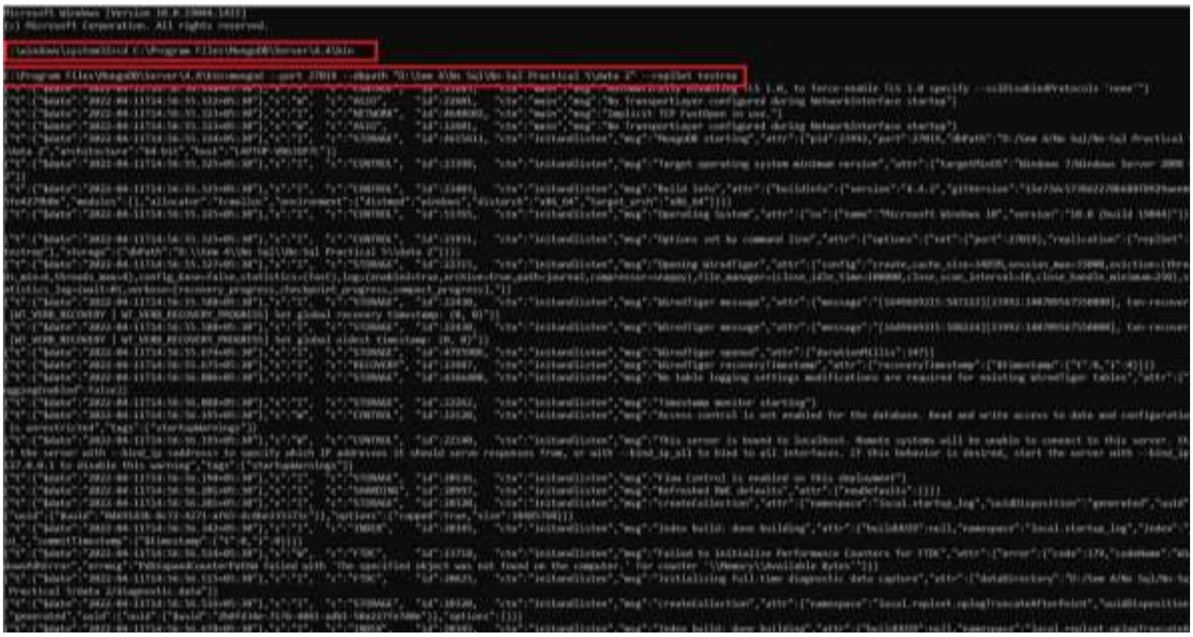
And in other terminal write following query mongo -port 27019.

First change the cmd path.



```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongod --port 27019 --dbpath "D:\Sem 4\No Sql\No-Sql Practical 5\data 2" --replSet testrep
[illegible]
```

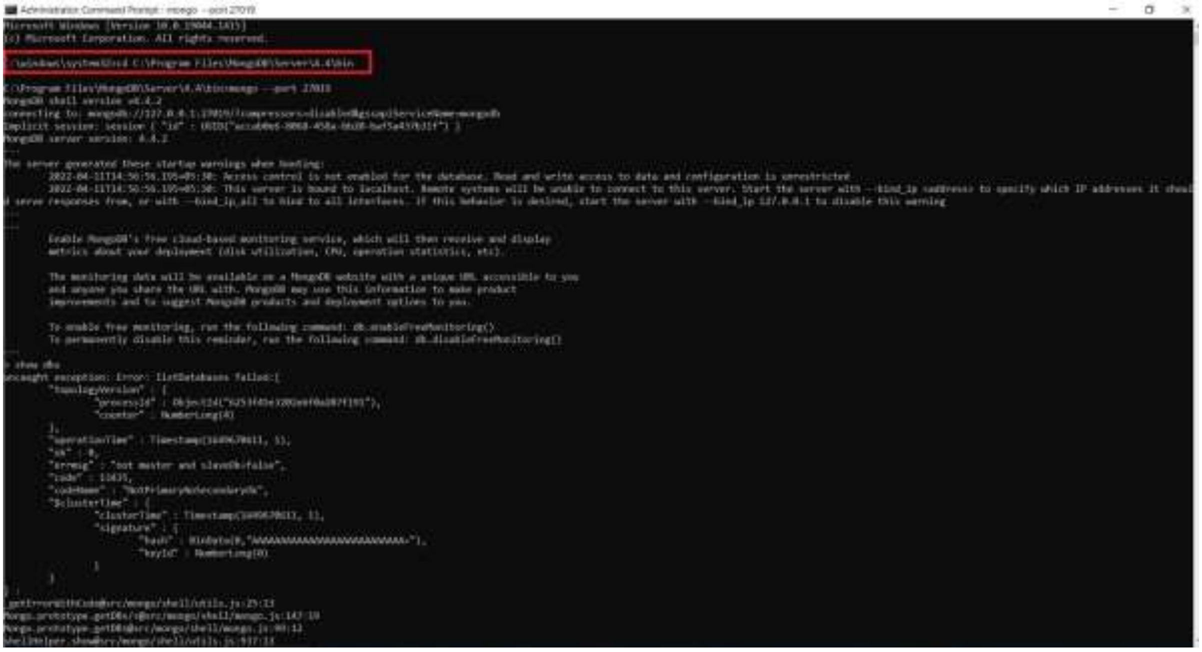
mongod --port 27019 --dbpath "D:\Sem 4\No Sql\No-Sql Practical 5\data 2" --replSet testrep



```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongod --port 27019 --dbpath "D:\Sem 4\No Sql\No Sql Practical 5\data 2" --replSet testrep
{"t":{"$date":"2022-04-11T14:56:55.122+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-11T14:56:55.123+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2022-04-11T14:56:55.123+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-11T14:56:55.124+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting", [illegible]
\data 2","architecture":"64-bit","host":"LAPTOP-[illegible]"}}
[illegible]
```

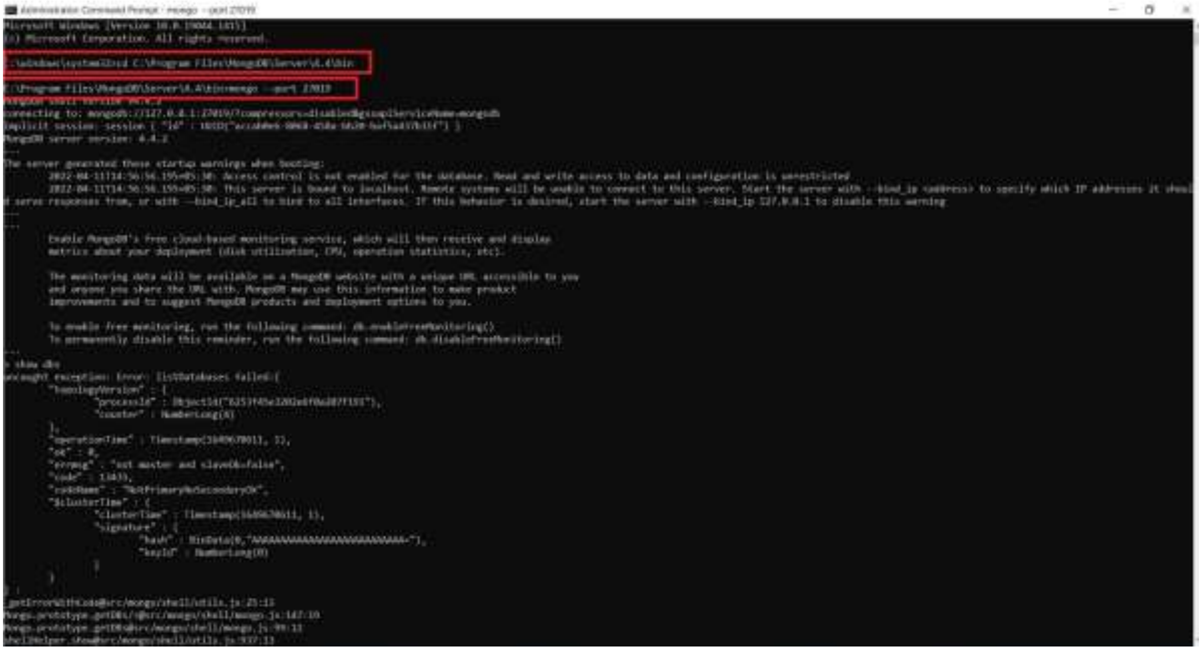
Now Open another command prompt for client. We will use this window to query our second server instance

First change the cmd path.



```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongo --port 27019
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27019/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("[illegible]") }
MongoDB server version: 4.4.2
---
The server generated these startup warnings when booting:
        2022-04-11T14:56:56.195+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2022-04-11T14:56:56.195+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it shoul
d 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()
---
> show dbs
uncaught exception: Error: listDatabases failed:{
        "topologyVersion" : {
                "processId" : ObjectId("[illegible]"),
                "counter" : NumberLong(4)
        },
        "operationTime" : Timestamp([illegible], 1),
        "ok" : 0,
        "errmsg" : "not master and slaveOk=false",
        "code" : 13435,
        "codeName" : "NotPrimaryNoSecondaryOk",
        "$clusterTime" : {
                "clusterTime" : Timestamp([illegible], 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:147:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:99:12
shellHelper.show@src/mongo/shell/utils.js:937:13
```

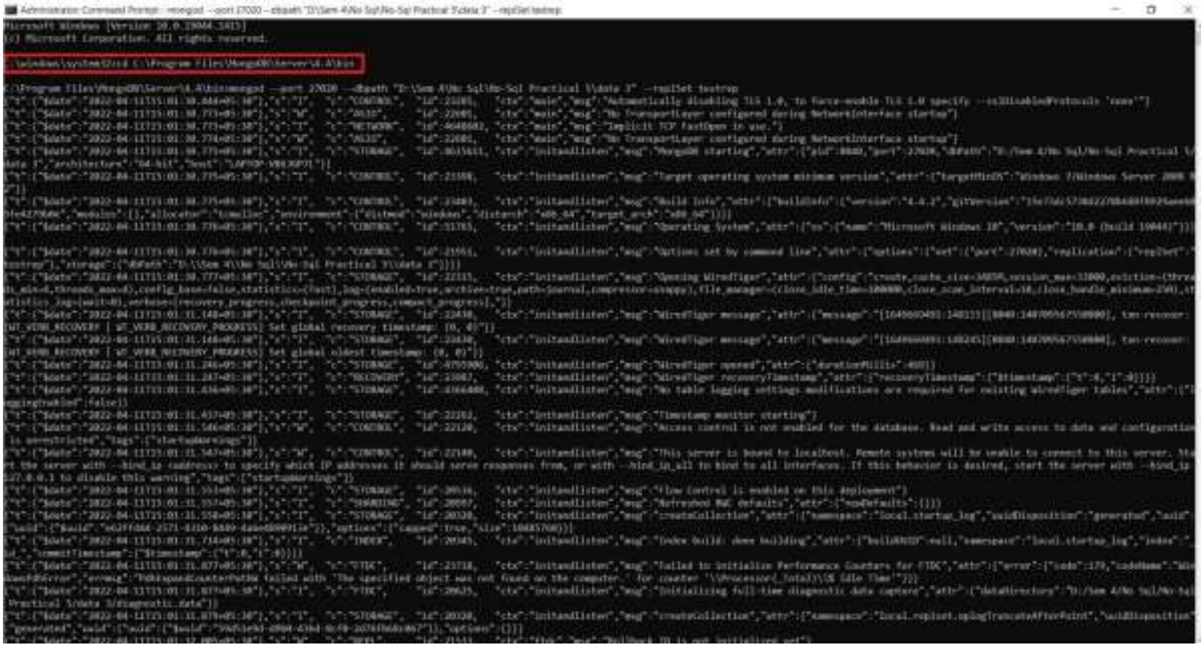
mongo --port 27019



```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongo --port 27019
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27019/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("[illegible]") }
MongoDB server version: 4.4.2
---
The server generated these startup warnings when booting:
        2022-04-11T14:56:56.195+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2022-04-11T14:56:56.195+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it shoul
d 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()
---
> show dbs
uncaught exception: Error: listDatabases failed:{
        "topologyVersion" : {
                "processId" : ObjectId("[illegible]"),
                "counter" : NumberLong(0)
        },
        "operationTime" : Timestamp(1649670011, 1),
        "ok" : 0,
        "errmsg" : "not master and slaveOk=false",
        "code" : 13435,
        "codeName" : "NotPrimaryNoSecondaryOk",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649670011, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:147:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:99:12
shellHelper.show@src/mongo/shell/utils.js:937:13
```

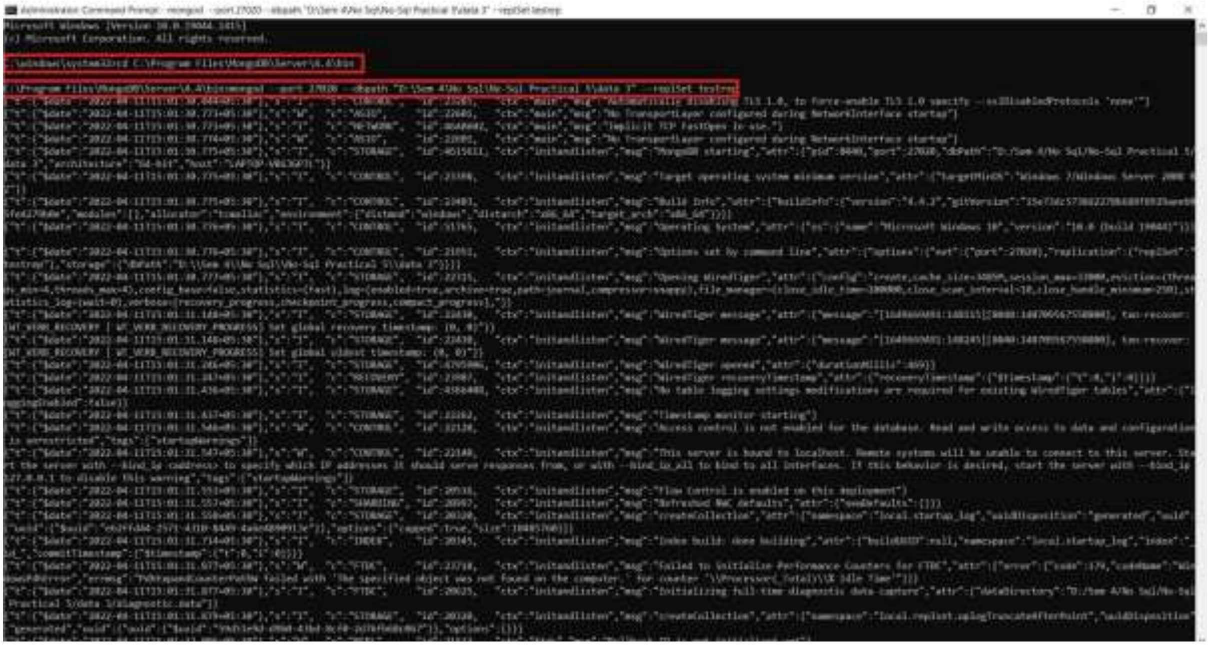
Now Open another command prompt for client. We will use this window to query our second server instance

First change the cmd path.



```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongod --port 27020 --dbpath "D:\Sem 4\No Sql\No-Sql Practical 5\data 3" --replSet testrep
{"t":{"$date":"2022-04-11T15:01:30.444+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-04-11T15:01:30.773+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-11T15:01:30.773+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2022-04-11T15:01:30.774+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-11T15:01:30.775+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":[illegible],"port":27020,"dbPath":"D:/Sem 4/No Sql/No-Sql Practical 5/
data 3","architecture":"64-bit","host":"[illegible]"}}
{"t":{"$date":"2022-04-11T15:01:30.775+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R
2"}}
{"t":{"$date":"2022-04-11T15:01:30.775+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"4.4.2","gitVersion":"[illegible]
[illegible]","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2022-04-11T15:01:30.776+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19044)"}}}

{"t":{"$date":"2022-04-11T15:01:30.776+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{"net":{"port":27020},"replication":{"replSet":"
testrep"},"storage":{"dbPath":"D:\\Sem 4\\No Sql\\No-Sql Practical 5\\data 3"}}}}
{"t":{"$date":"2022-04-11T15:01:30.777+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"create,cache_size=[illegible],session_max=33000,eviction=(threa
ds_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),st
atistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress,compact_progress],"}}
{"t":{"$date":"2022-04-11T15:01:31.148+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[illegible]
[illegible]
```

mongod --port 27020 --dbpath "D:\Sem 4\No Sql\No-Sql Practical 5\data 3" --replSet testrep
C:\>mongo --port 27020

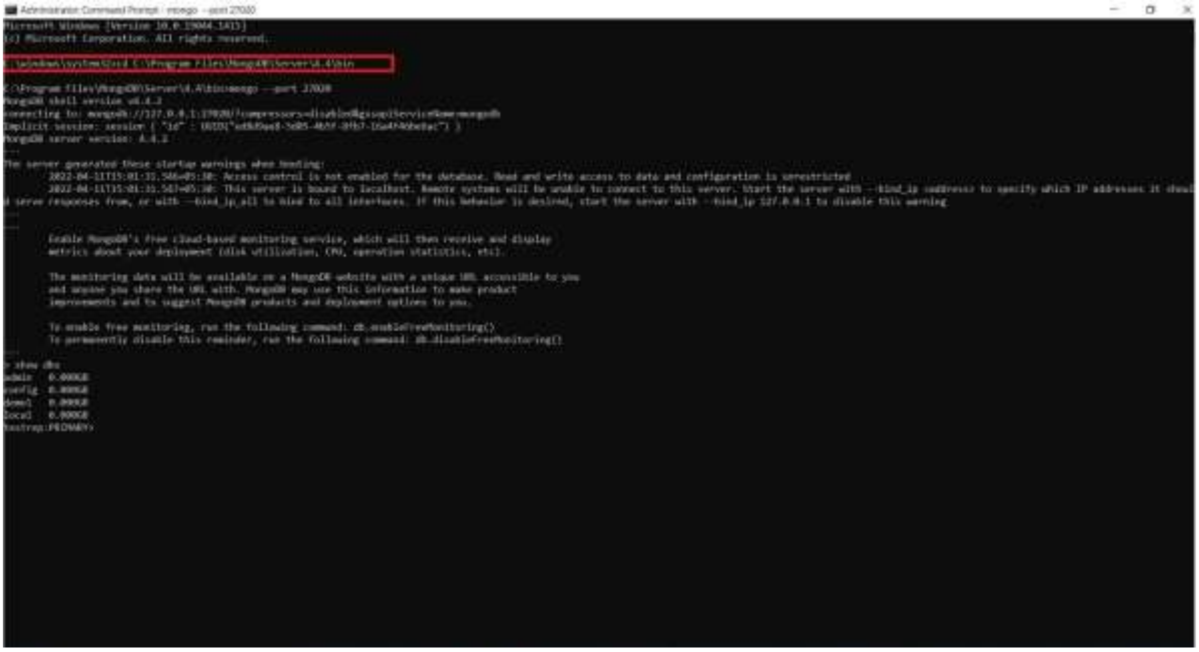


Administrator: Command Prompt - mongod --port 27020 --dbpath "D:\Sem 4\No Sql\No-Sql Practical 5\data 3" --replSet testrep

```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongod --port 27020 --dbpath "D:\Sem 4\No Sql\No-Sql Practical 5\data 3" --replSet testrep
{"t":{"$date":"2022-04-11T15:01:30.644+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-04-11T15:01:30.773+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-11T15:01:30.773+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2022-04-11T15:01:30.774+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-11T15:01:30.775+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":[illegible],"port":27020,"dbPath":"D:/Sem 4/No Sql/No-Sql Practical 5/data 3","architecture":"64-bit","host":"LAPTOP-[illegible]"}}
{"t":{"$date":"2022-04-11T15:01:30.775+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":{"$date":"2022-04-11T15:01:30.775+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"4.4.2","gitVersion":"[illegible]","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2022-04-11T15:01:30.776+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19044)"}}}

{"t":{"$date":"2022-04-11T15:01:30.776+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{"net":{"port":27020},"replication":{"replSet":"testrep"},"storage":{"dbPath":"D:\\Sem 4\\No Sql\\No-Sql Practical 5\\data 3"}}}}
{"t":{"$date":"2022-04-11T15:01:30.777+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"create,cache_size=[illegible],session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress,compact_progress],"}}
{"t":{"$date":"2022-04-11T15:01:31.148+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[[illegible]][[illegible]], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}}
{"t":{"$date":"2022-04-11T15:01:31.148+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[[illegible]][[illegible]], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMillis":[illegible]}}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"RECOVERY", "id":23987,   "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"recoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"STORAGE",  "id":4366408, "ctx":"initandlisten","msg":"No table logging settings modifications are required for existing WiredTiger tables","attr":{"loggingEnabled":false}}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"STORAGE",  "id":22262,   "ctx":"initandlisten","msg":"Timestamp monitor starting"}
{"t":{"$date":"2022-04-11T15:01:31.546+05:30"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2022-04-11T15:01:31.547+05:30"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"initandlisten","msg":"This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarnings"]}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"STORAGE",  "id":20536,   "ctx":"initandlisten","msg":"Flow Control is enabled on this deployment"}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"SHARDING", "id":20997,   "ctx":"initandlisten","msg":"Refreshed RWC defaults","attr":{"newDefaults":{}}}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"STORAGE",  "id":20320,   "ctx":"initandlisten","msg":"createCollection","attr":{"namespace":"local.startup_log","uuidDisposition":"generated","uuid":{"uuid":{"$uuid":"[illegible]"}},"options":{"capped":true,"size":10485760}}}
{"t":{"$date":"2022-04-11T15:01:31.714+05:30"},"s":"I",  "c":"INDEX",    "id":20345,   "ctx":"initandlisten","msg":"Index build: done building","attr":{"buildUUID":null,"namespace":"local.startup_log","index":"_id_","commitTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"W",  "c":"FTDC",     "id":23718,   "ctx":"initandlisten","msg":"Failed to initialize Performance Counters for FTDC","attr":{"error":{"code":179,"codeName":"WindowsPdhError","errmsg":"PdhExpandCounterPathW failed with 'The specified object was not found on the computer.' for counter '\\Processor(_Total)\\% Idle Time'"}}}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"FTDC",     "id":20625,   "ctx":"initandlisten","msg":"Initializing full-time diagnostic data capture","attr":{"dataDirectory":"D:/Sem 4/No Sql/No-Sql Practical 5/data 3/diagnostic.data"}}
{"t":{"$date":"2022-04-11T15:01:31.[illegible]+05:30"},"s":"I",  "c":"STORAGE",  "id":20320,   "ctx":"initandlisten","msg":"createCollection","attr":{"namespace":"local.replset.oplogTruncateAfterPoint","uuidDisposition":"generated","uuid":{"uuid":{"$uuid":"[illegible]"}},"options":{}}}
[illegible]
```

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

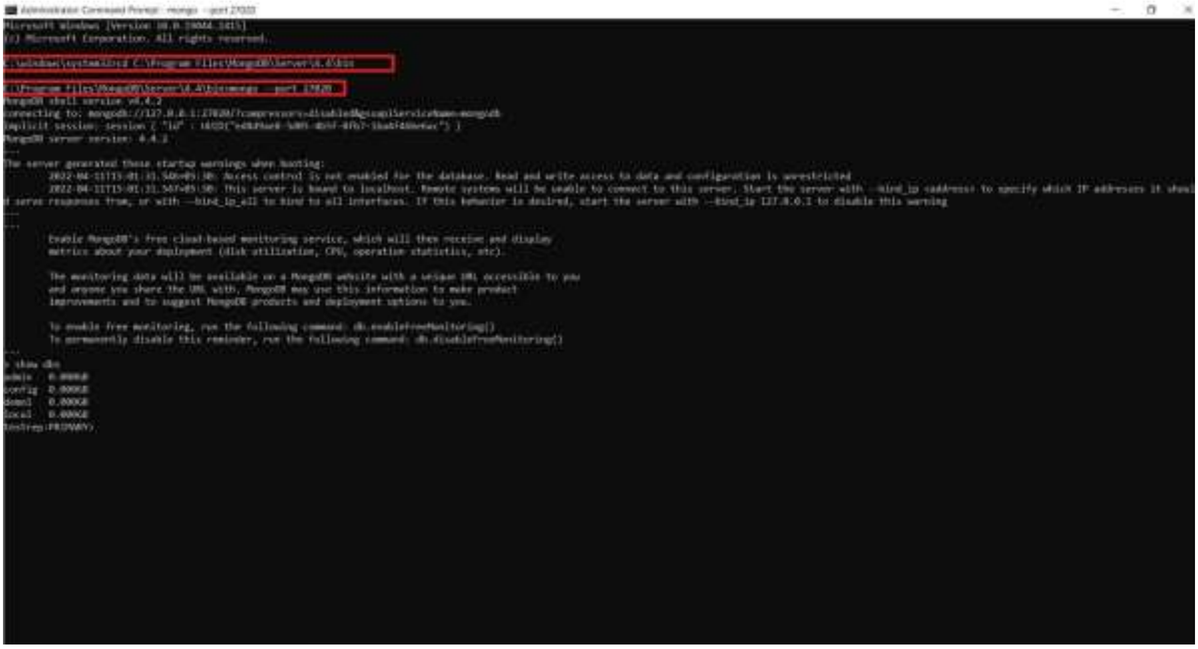
First change the cmd path.



Administrator: Command Prompt - mongo --port 27020

```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongo --port 27020
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27020/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("[illegible]") }
MongoDB server version: 4.4.2
---
The server generated these startup warnings when booting:
        2022-04-11T15:01:31.546+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2022-04-11T15:01:31.547+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it shoul
d 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()
---
> show dbs
admin   0.000GB
config  0.000GB
demo1   0.000GB
local   0.000GB
testrep:PRIMARY>
```

mongo --port 27020



Administrator: Command Prompt - mongo --port 27020

```
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\windows\system32>cd C:\Program Files\MongoDB\Server\4.4\bin

C:\Program Files\MongoDB\Server\4.4\bin>mongo --port 27020
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27020/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("[illegible]") }
MongoDB server version: 4.4.2
---
The server generated these startup warnings when booting:
        2022-04-11T15:01:31.546+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2022-04-11T15:01:31.547+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it shoul
d 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()
---
> show dbs
admin   0.000GB
config  0.000GB
demo1   0.000GB
local   0.000GB
testrep:PRIMARY>
```

Now we have in total 6 CMDs among them 2 for primary and other pairs for secondary.

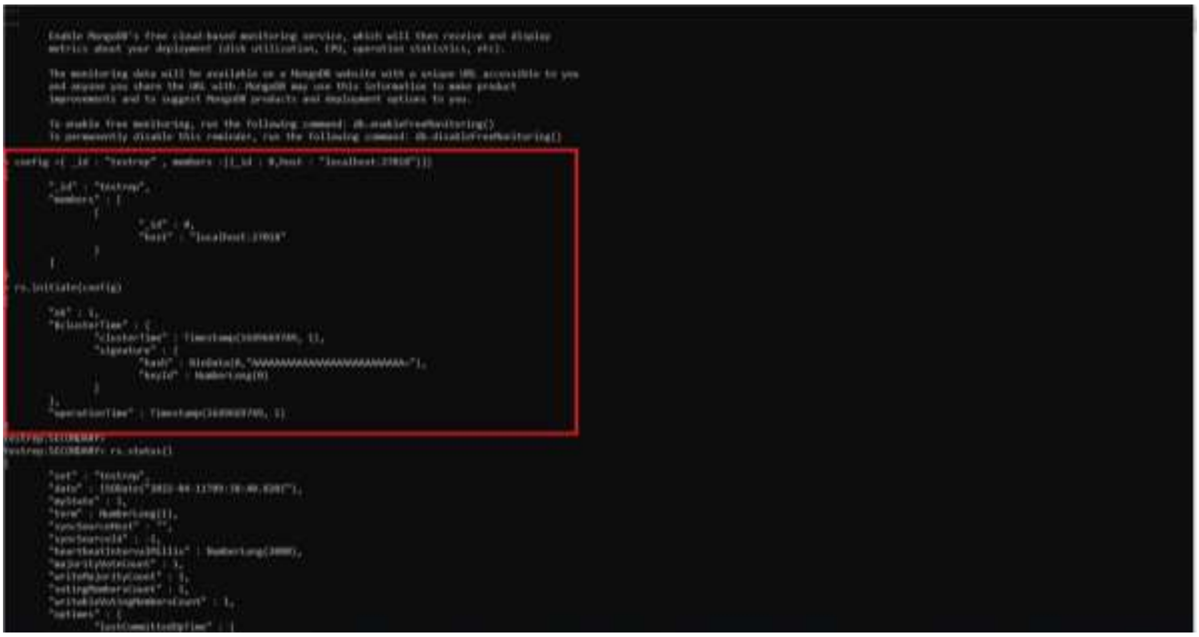
Step 5:

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

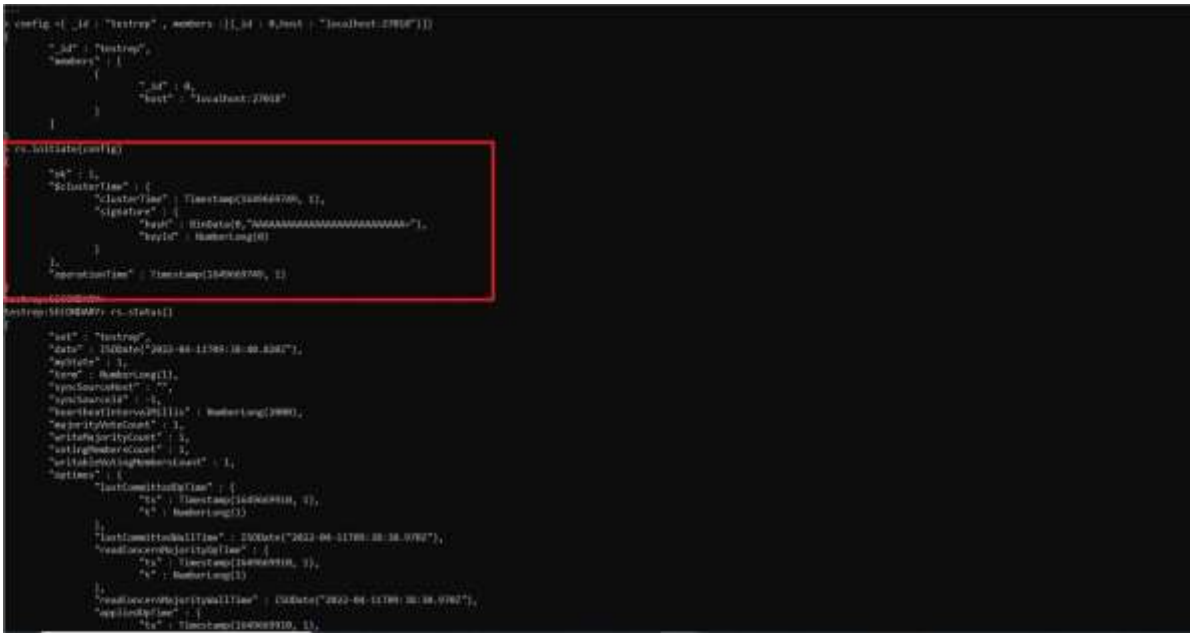


```
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        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" , members :[{_id : 0,host : "localhost:27018"}]}
{
        "_id" : "testrep",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27018"
                }
        ]
}
> rs.initiate(config)
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp([illegible], 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp([illegible], 1)
}
testrep:SECONDARY>
testrep:SECONDARY> rs.status()
{
        "set" : "testrep",
        "date" : ISODate("2022-04-11T[illegible]Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 1,
        "writeMajorityCount" : 1,
        "votingMembersCount" : 1,
        "writableVotingMembersCount" : 1,
        "optimes" : {
                "lastCommittedOpTime" : {
```

After this write command rs.initiate(config)

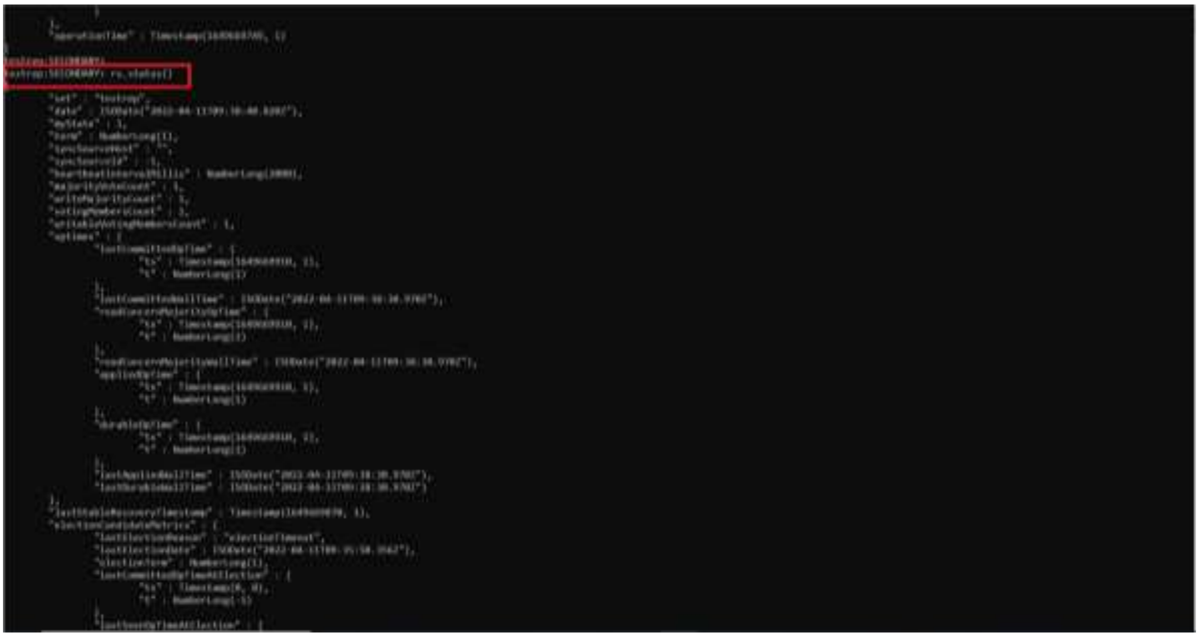
This command initiates a replica set with the current host as its only member.

This is confirmed by the output, which should resemble the following:

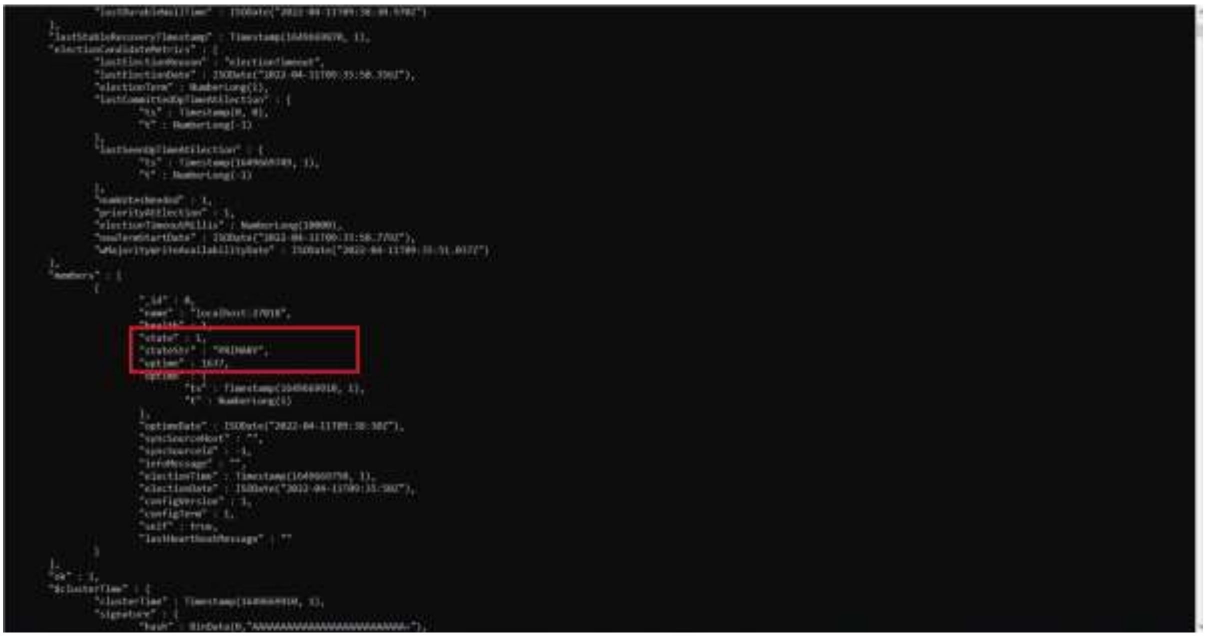


```
> config ={ _id : "testrep" , members :[{_id : 0,host : "localhost:27018"}]}
{
        "_id" : "testrep",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27018"
                }
        ]
}
> rs.initiate(config)
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649669740, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649669740, 1)
}
testrep:SECONDARY>
testrep:SECONDARY> rs.status()
{
        "set" : "testrep",
        "date" : ISODate("2022-04-11T09:38:40.828Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 1,
        "writeMajorityCount" : 1,
        "votingMembersCount" : 1,
        "writableVotingMembersCount" : 1,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1649669910, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2022-04-11T09:38:38.978Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1649669910, 1),
                        "t" : NumberLong(1)
                },
                "readConcernMajorityWallTime" : ISODate("2022-04-11T09:38:38.978Z"),
                "appliedOpTime" : {
                        "ts" : Timestamp(1649669910, 1),
```

After In primary server’s cmd we have to execute rs.status()query. rs.status()



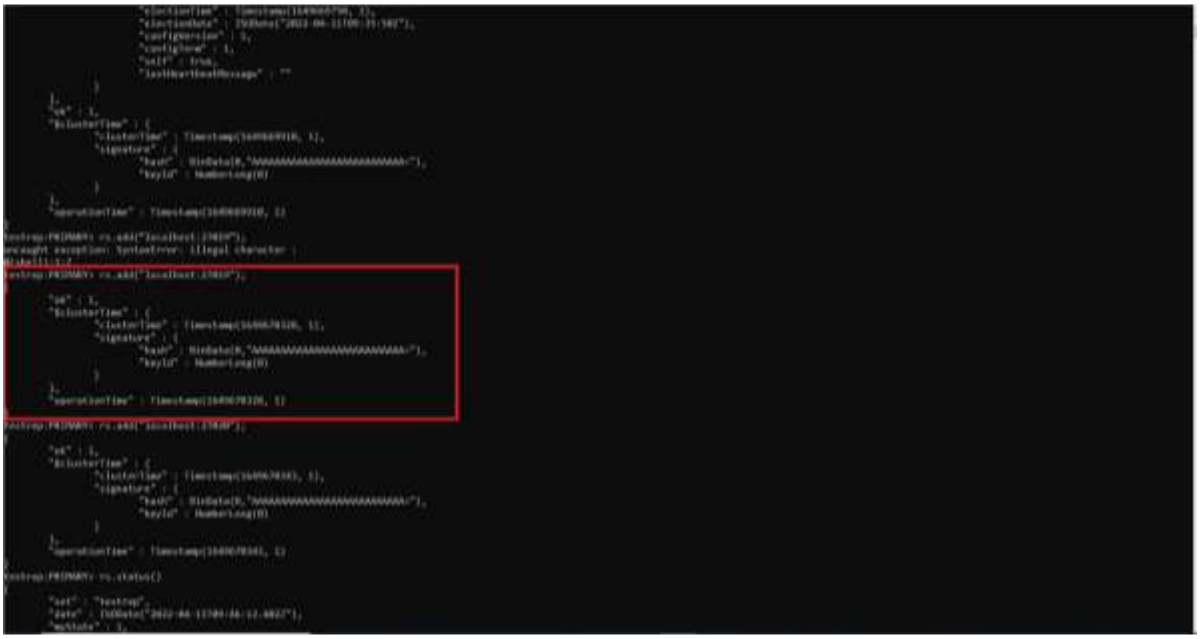
```
                }
        },
        "operationTime" : Timestamp(1649669740, 1)
}
testrep:SECONDARY>
testrep:SECONDARY> rs.status()
{
        "set" : "testrep",
        "date" : ISODate("2022-04-11T09:38:40.828Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 1,
        "writeMajorityCount" : 1,
        "votingMembersCount" : 1,
        "writableVotingMembersCount" : 1,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1649669910, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2022-04-11T09:38:38.978Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1649669910, 1),
                        "t" : NumberLong(1)
                },
                "readConcernMajorityWallTime" : ISODate("2022-04-11T09:38:38.978Z"),
                "appliedOpTime" : {
                        "ts" : Timestamp(1649669910, 1),
                        "t" : NumberLong(1)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1649669910, 1),
                        "t" : NumberLong(1)
                },
                "lastAppliedWallTime" : ISODate("2022-04-11T09:38:38.978Z"),
                "lastDurableWallTime" : ISODate("2022-04-11T09:38:38.978Z")
        },
        "lastStableRecoveryTimestamp" : Timestamp(1649669870, 1),
        "electionCandidateMetrics" : {
                "lastElectionReason" : "electionTimeout",
                "lastElectionDate" : ISODate("2022-04-11T09:35:58.354Z"),
                "electionTerm" : NumberLong(1),
                "lastCommittedOpTimeAtElection" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
                },
                "lastSeenOpTimeAtElection" : {
```



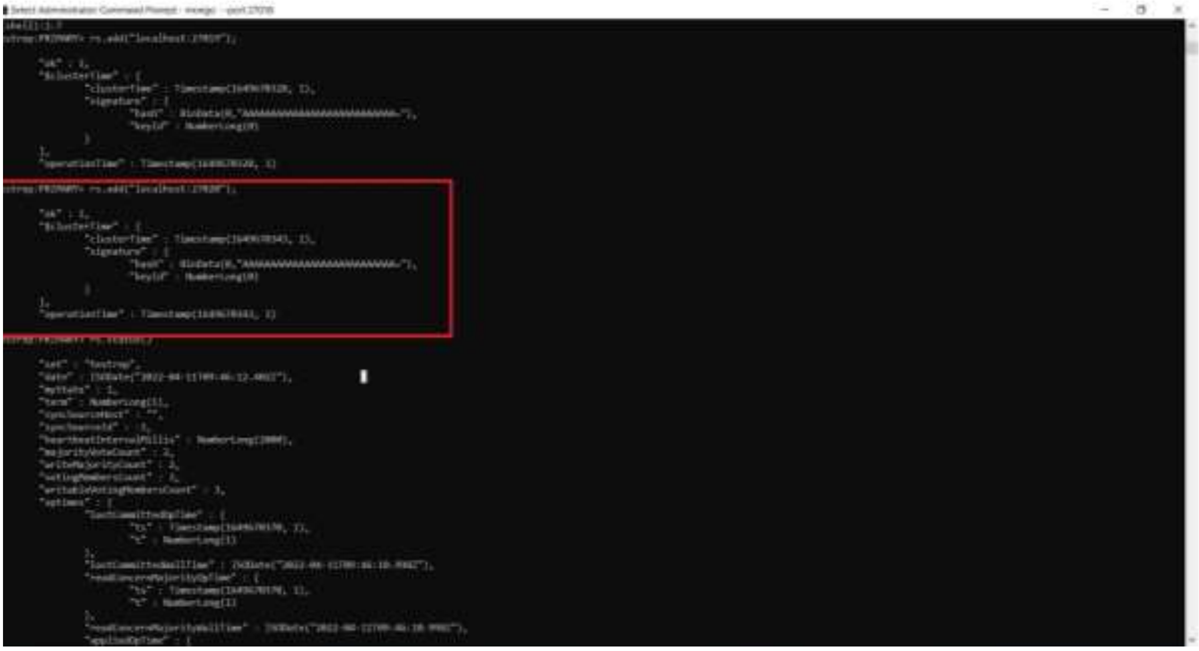
```
                "lastDurableWallTime" : ISODate("2022-04-11T09:38:49.570Z")
        },
        "lastStableRecoveryTimestamp" : Timestamp(1649669878, 1),
        "electionCandidateMetrics" : {
                "lastElectionReason" : "electionTimeout",
                "lastElectionDate" : ISODate("2022-04-11T09:35:58.356Z"),
                "electionTerm" : NumberLong(1),
                "lastCommittedOpTimeAtElection" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
                },
                "lastSeenOpTimeAtElection" : {
                        "ts" : Timestamp(1649669749, 1),
                        "t" : NumberLong(-1)
                },
                "numVotesNeeded" : 1,
                "priorityAtElection" : 1,
                "electionTimeoutMillis" : NumberLong(10000),
                "newTermStartDate" : ISODate("2022-04-11T09:35:58.770Z"),
                "wMajorityWriteAvailabilityDate" : ISODate("2022-04-11T09:35:51.037Z")
        },
        "members" : [
                {
                        "_id" : 0,
                        "name" : "localhost:27018",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 1677,
                        "optime" : {
                                "ts" : Timestamp(1649669918, 1),
                                "t" : NumberLong(1)
                        },
                        "optimeDate" : ISODate("2022-04-11T09:38:38Z"),
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1649669758, 1),
                        "electionDate" : ISODate("2022-04-11T09:35:58Z"),
                        "configVersion" : 1,
                        "configTerm" : 1,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                }
        ],
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649669918, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
```

Step 6:

In primary server’s cmd we have to execute rs.add() query.



```
                        "electionTime" : Timestamp(1649669758, 1),
                        "electionDate" : ISODate("2022-04-11T09:35:58Z"),
                        "configVersion" : 1,
                        "configTerm" : 1,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                }
        ],
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649669918, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649669918, 1)
}
testrep:PRIMARY> rs.add("localhost:27019");
uncaught exception: SyntaxError: illegal character :
@(shell):1:2
testrep:PRIMARY> rs.add("localhost:27019");
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649670126, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649670126, 1)
}
testrep:PRIMARY> rs.add("localhost:27020");
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649670141, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649670141, 1)
}
testrep:PRIMARY> rs.status()
{
        "set" : "testrep",
        "date" : ISODate("2022-04-11T09:46:12.482Z"),
        "myState" : 1,
```

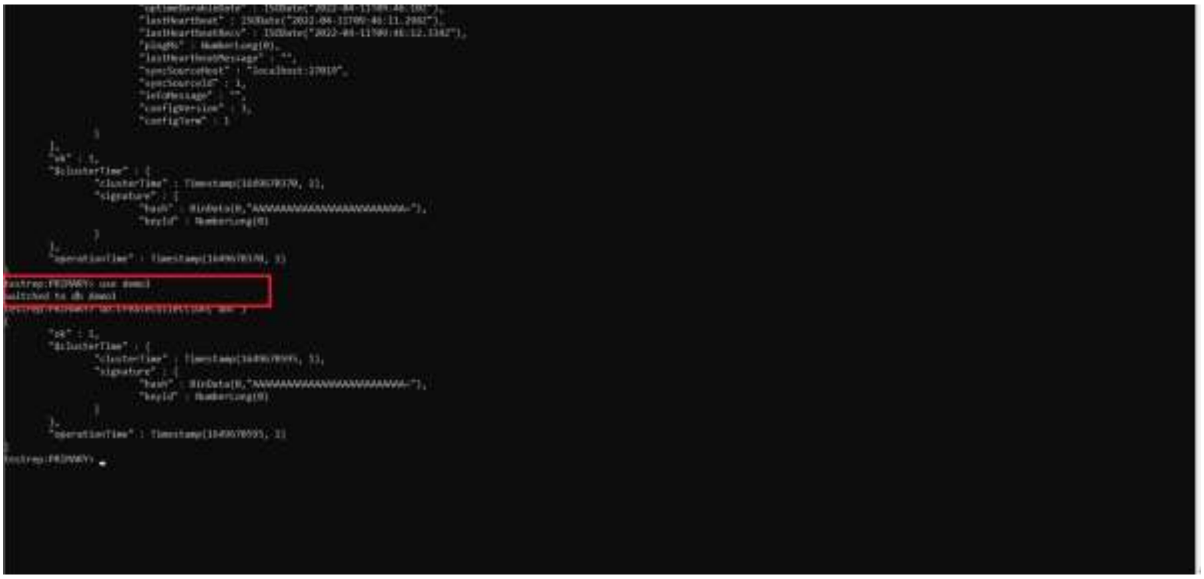


```
shell]:1:7
strep:PRIMARY> rs.add("localhost:27019");
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649678128, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649678128, 1)
}
strep:PRIMARY> rs.add("localhost:27020");
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649678143, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649678143, 1)
}
strep:PRIMARY> rs.status()
{
        "set" : "testrep",
        "date" : ISODate("2022-04-11T09:46:12.401Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 2,
        "writeMajorityCount" : 2,
        "votingMembersCount" : 3,
        "writableVotingMembersCount" : 3,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1649678170, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2022-04-11T09:46:10.998Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1649678170, 1),
                        "t" : NumberLong(1)
                },
                "readConcernMajorityWallTime" : ISODate("2022-04-11T09:46:10.998Z"),
                "appliedOpTime" : {
```

Now to check whether everything is working fine or not we’ll create one collection and check for it in secondary servers.

Step 7:

Create a database named demo1 and collection inside named demo1.On CMD client window of Primary client (port 27018).

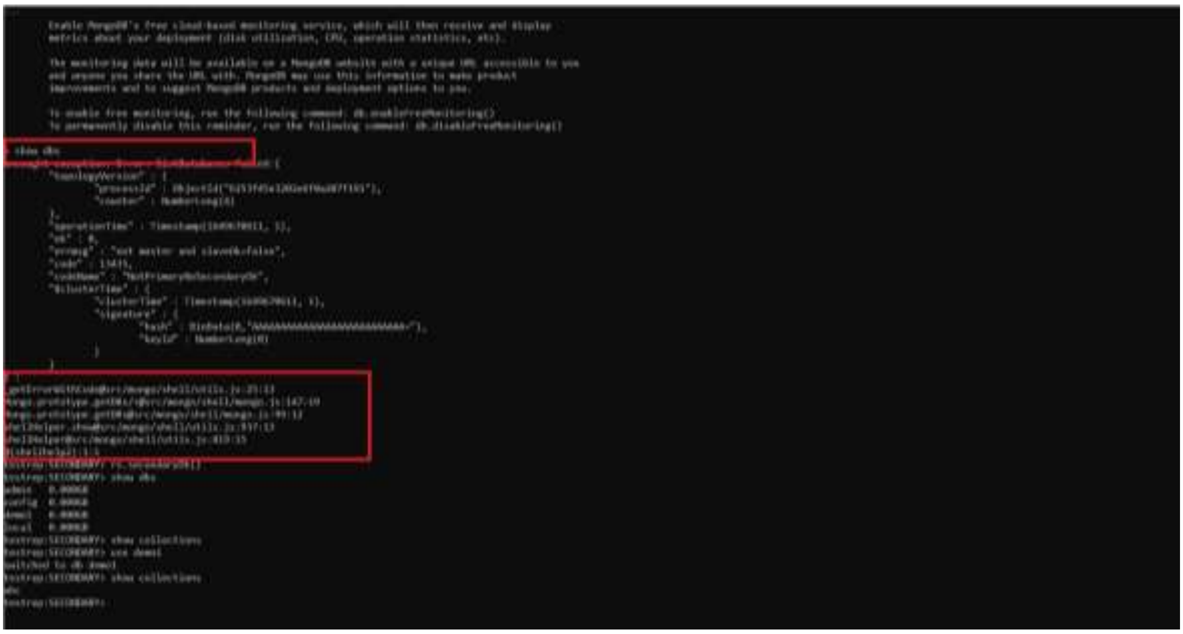


```
                        "optimeDurableDate" : ISODate("2022-04-11T09:46:10Z"),
                        "lastHeartbeat" : ISODate("2022-04-11T09:46:11.290Z"),
                        "lastHeartbeatRecv" : ISODate("2022-04-11T09:46:12.334Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncSourceHost" : "localhost:27018",
                        "syncSourceId" : 1,
                        "infoMessage" : "",
                        "configVersion" : 3,
                        "configTerm" : 1
                }
        ],
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649678170, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649678170, 1)
}
testrep:PRIMARY> use demo1
switched to db demo1
testrep:PRIMARY> db.createCollection("demo1")
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649678595, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649678595, 1)
}
testrep:PRIMARY>
```

Step 8:

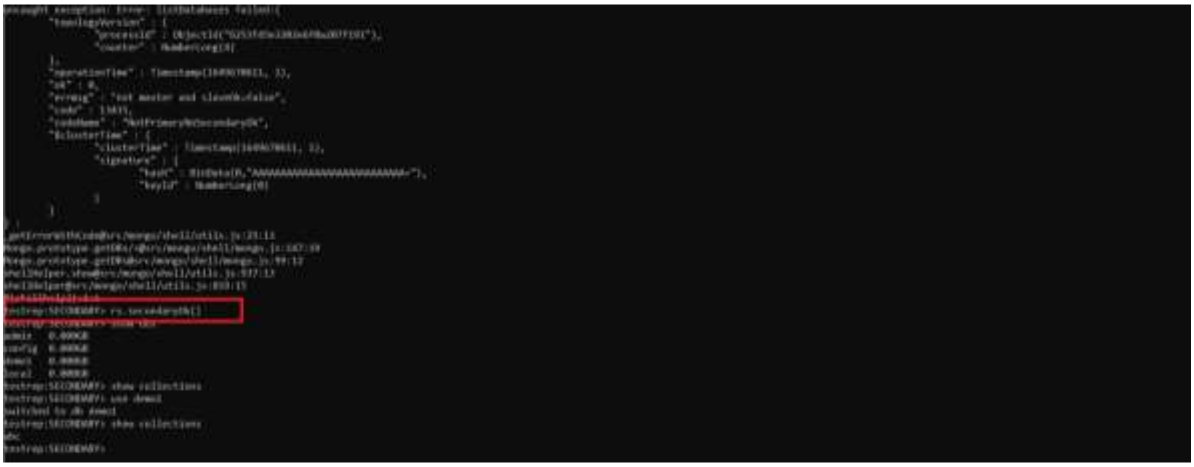
Now goto client window of port 27019(To check whether we can see the database replica over there or not).

First try to execute show dbs directly, it will show error or operation failed!



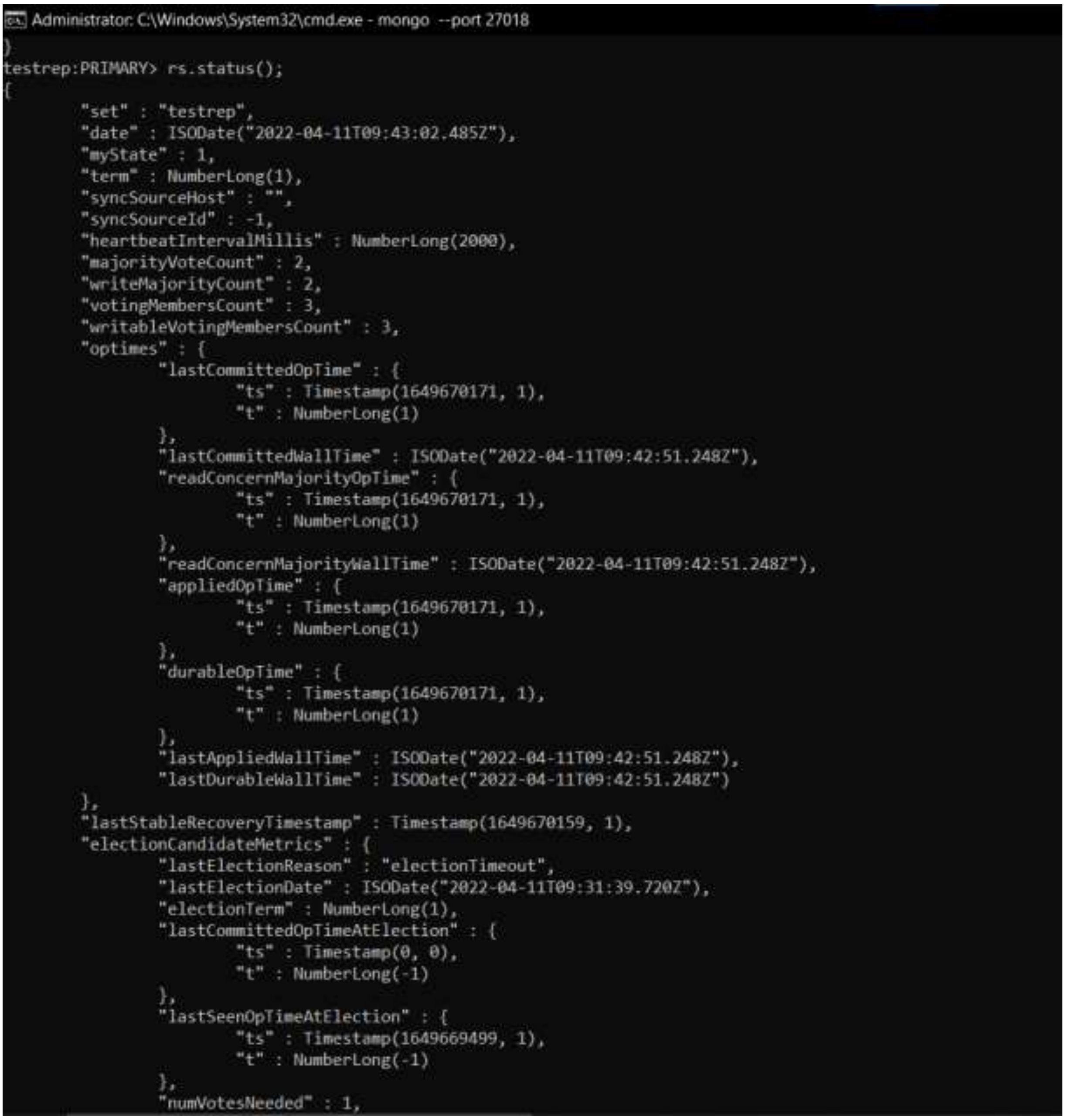
```
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        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()

> show dbs
uncaught exception: Error: listDatabases failed:{
        "topologyVersion" : {
                "processId" : ObjectId("[illegible]"),
                "counter" : NumberLong(4)
        },
        "operationTime" : Timestamp(1649678611, 1),
        "ok" : 0,
        "errmsg" : "not master and slaveOk=false",
        "code" : 13435,
        "codeName" : "NotPrimaryNoSecondaryOk",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649678611, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:147:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:99:12
shellHelper.show@src/mongo/shell/utils.js:937:13
shellHelper@src/mongo/shell/utils.js:819:15
@(shellhelp2):1:1
testrep:SECONDARY> rs.secondaryOk()
testrep:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
demo1   0.000GB
local   0.000GB
testrep:SECONDARY> show collections
testrep:SECONDARY> use demo1
switched to db demo1
testrep:SECONDARY> show collections
abc
testrep:SECONDARY>
```

Then to get perfect ouput we need to execute rs.secondaryOk() method.



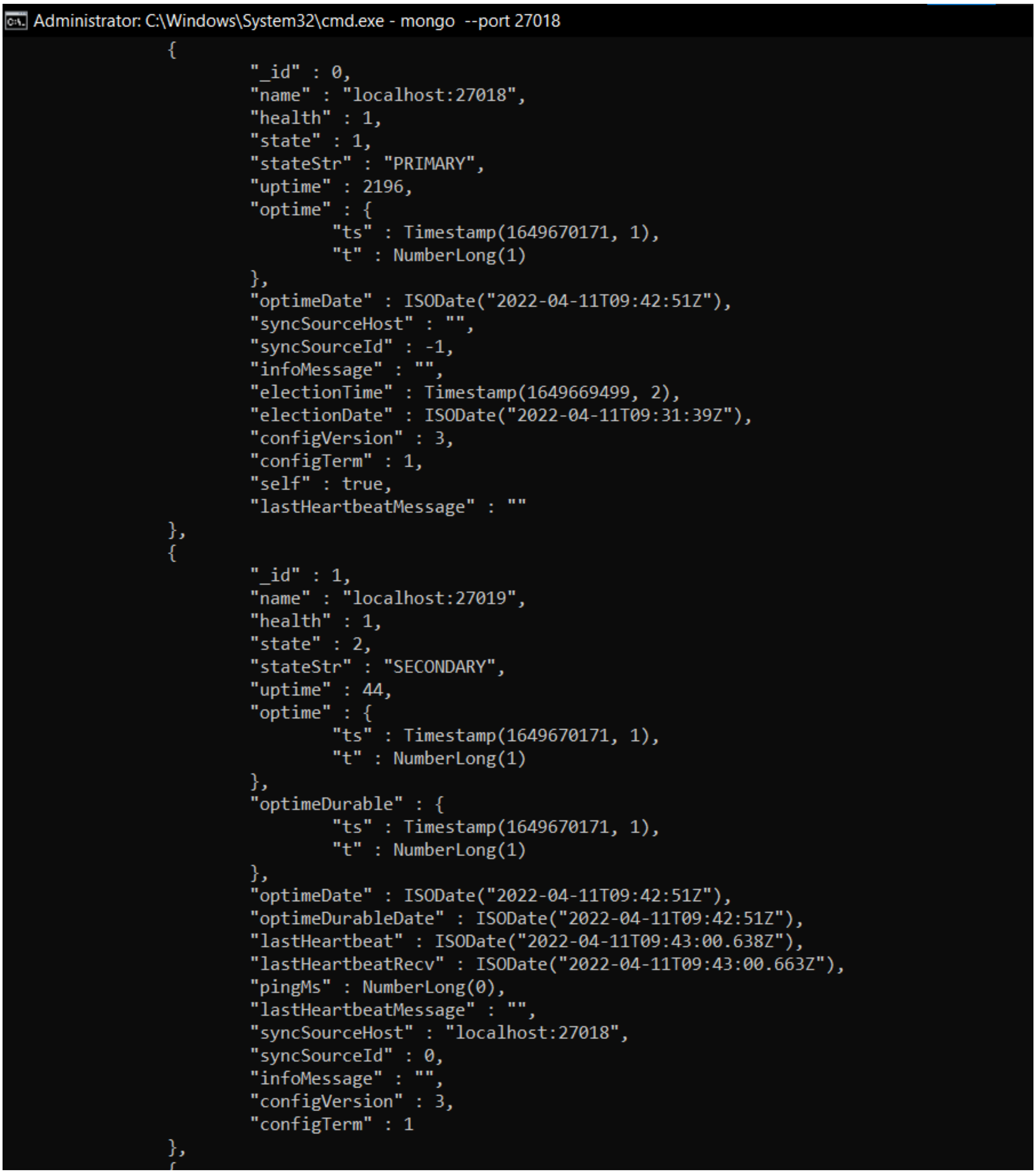
```
uncaught exception: Error: listDatabases failed:{
        "topologyVersion" : {
                "processId" : ObjectId("[illegible]"),
                "counter" : NumberLong(4)
        },
        "operationTime" : Timestamp(1649678611, 1),
        "ok" : 0,
        "errmsg" : "not master and slaveOk=false",
        "code" : 13435,
        "codeName" : "NotPrimaryNoSecondaryOk",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649678611, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs/<@src/mongo/shell/mongo.js:147:19
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:99:12
shellHelper.show@src/mongo/shell/utils.js:937:13
shellHelper@src/mongo/shell/utils.js:819:15
@(shellhelp2):1:1
testrep:SECONDARY> rs.secondaryOk()
testrep:SECONDARY> show dbs
admin   0.000GB
config  0.000GB
demo1   0.000GB
local   0.000GB
testrep:SECONDARY> show collections
testrep:SECONDARY> use demo1
switched to db demo1
testrep:SECONDARY> show collections
abc
testrep:SECONDARY>
```

We can also use rs.status() to check status of our servers whether its primary or secondary.



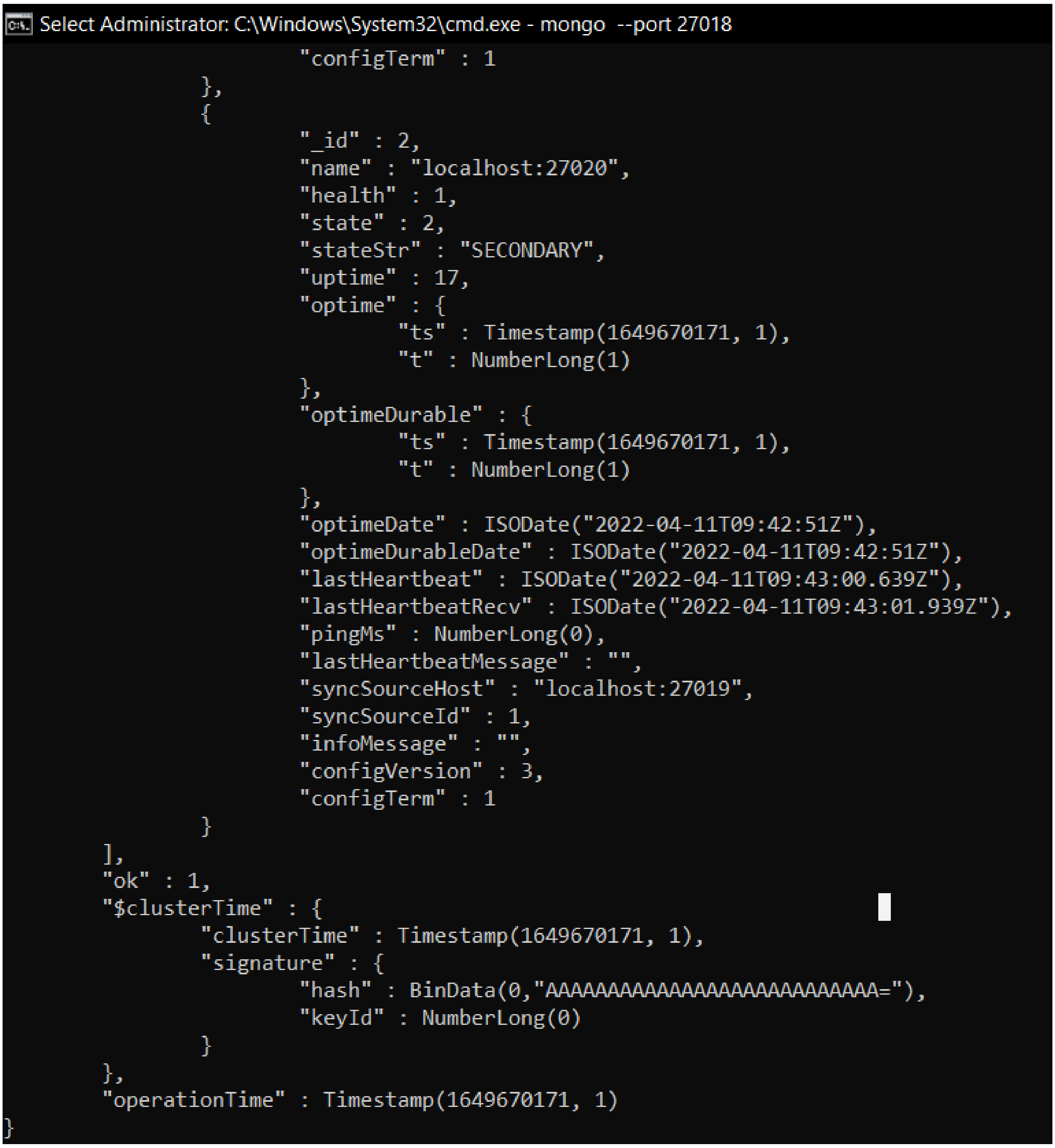
Administrator: C:\Windows\System32\cmd.exe - mongo --port 27018

```
}
testrep:PRIMARY> rs.status();
{
        "set" : "testrep",
        "date" : ISODate("2022-04-11T09:43:02.485Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "majorityVoteCount" : 2,
        "writeMajorityCount" : 2,
        "votingMembersCount" : 3,
        "writableVotingMembersCount" : 3,
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1649670171, 1),
                        "t" : NumberLong(1)
                },
                "lastCommittedWallTime" : ISODate("2022-04-11T09:42:51.248Z"),
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1649670171, 1),
                        "t" : NumberLong(1)
                },
                "readConcernMajorityWallTime" : ISODate("2022-04-11T09:42:51.248Z"),
                "appliedOpTime" : {
                        "ts" : Timestamp(1649670171, 1),
                        "t" : NumberLong(1)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1649670171, 1),
                        "t" : NumberLong(1)
                },
                "lastAppliedWallTime" : ISODate("2022-04-11T09:42:51.248Z"),
                "lastDurableWallTime" : ISODate("2022-04-11T09:42:51.248Z")
        },
        "lastStableRecoveryTimestamp" : Timestamp(1649670159, 1),
        "electionCandidateMetrics" : {
                "lastElectionReason" : "electionTimeout",
                "lastElectionDate" : ISODate("2022-04-11T09:31:39.720Z"),
                "electionTerm" : NumberLong(1),
                "lastCommittedOpTimeAtElection" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
                },
                "lastSeenOpTimeAtElection" : {
                        "ts" : Timestamp(1649669499, 1),
                        "t" : NumberLong(-1)
                },
                "numVotesNeeded" : 1,
```



Administrator: C:\Windows\System32\cmd.exe - mongo --port 27018

```
                {
                        "_id" : 0,
                        "name" : "localhost:27018",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 2196,
                        "optime" : {
                                "ts" : Timestamp(1649670171, 1),
                                "t" : NumberLong(1)
                        },
                        "optimeDate" : ISODate("2022-04-11T09:42:51Z"),
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1649669499, 2),
                        "electionDate" : ISODate("2022-04-11T09:31:39Z"),
                        "configVersion" : 3,
                        "configTerm" : 1,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                },
                {
                        "_id" : 1,
                        "name" : "localhost:27019",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 44,
                        "optime" : {
                                "ts" : Timestamp(1649670171, 1),
                                "t" : NumberLong(1)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1649670171, 1),
                                "t" : NumberLong(1)
                        },
                        "optimeDate" : ISODate("2022-04-11T09:42:51Z"),
                        "optimeDurableDate" : ISODate("2022-04-11T09:42:51Z"),
                        "lastHeartbeat" : ISODate("2022-04-11T09:43:00.638Z"),
                        "lastHeartbeatRecv" : ISODate("2022-04-11T09:43:00.663Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncSourceHost" : "localhost:27018",
                        "syncSourceId" : 0,
                        "infoMessage" : "",
                        "configVersion" : 3,
                        "configTerm" : 1
                },
```



Select Administrator: C:\Windows\System32\cmd.exe - mongo --port 27018

```
                        "configTerm" : 1
                },
                {
                        "_id" : 2,
                        "name" : "localhost:27020",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 17,
                        "optime" : {
                                "ts" : Timestamp(1649670171, 1),
                                "t" : NumberLong(1)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1649670171, 1),
                                "t" : NumberLong(1)
                        },
                        "optimeDate" : ISODate("2022-04-11T09:42:51Z"),
                        "optimeDurableDate" : ISODate("2022-04-11T09:42:51Z"),
                        "lastHeartbeat" : ISODate("2022-04-11T09:43:00.639Z"),
                        "lastHeartbeatRecv" : ISODate("2022-04-11T09:43:01.939Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncSourceHost" : "localhost:27019",
                        "syncSourceId" : 1,
                        "infoMessage" : "",
                        "configVersion" : 3,
                        "configTerm" : 1
                }
        ],
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1649670171, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1649670171, 1)
}
```