Pradnya Tagad

L037

PRACTICAL-7

Step 1: Create Database Folders (Primary,Secondary1,Secondary2).

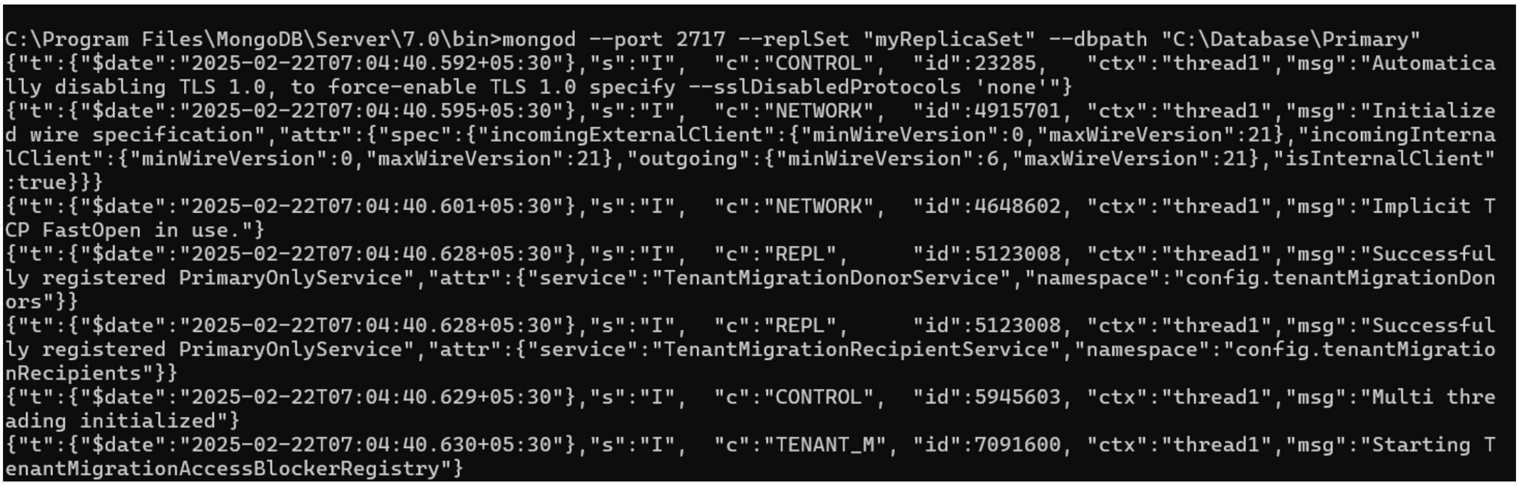
-Open MongoDB terminal (cmd) from bin, and follow the below commands:

*For primary: mongod --port 2717 --replSet "myReplicaSet" --dbpath "/data/primary"

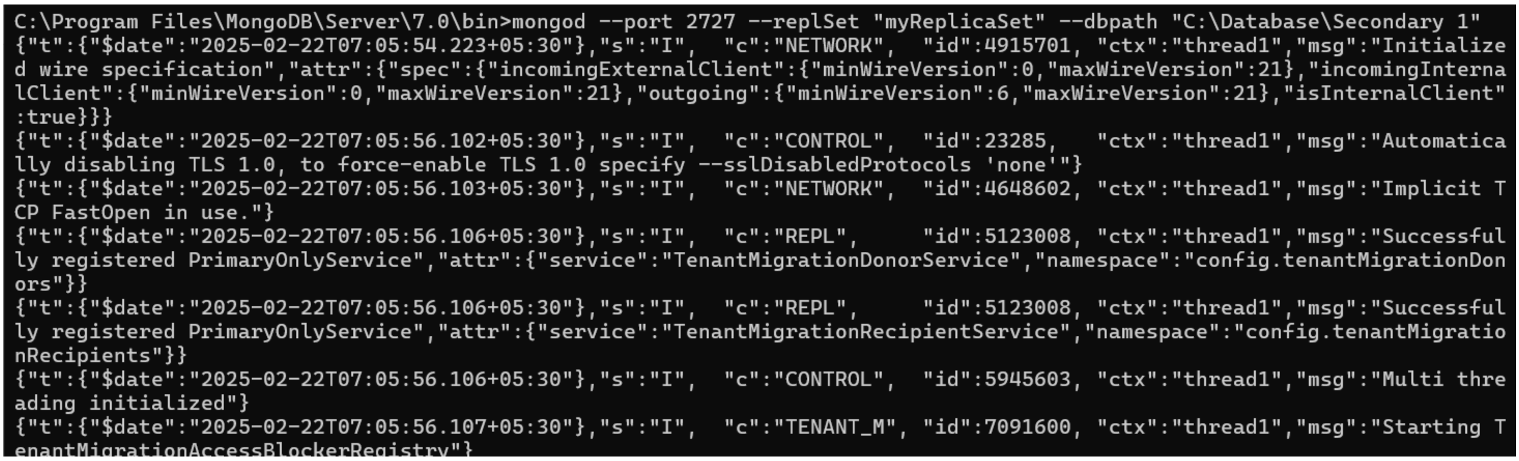
*For Secondary1: mongod --port 2727 --replSet "myReplicaSet" --dbpath "/data/secondary1"

*For Secondary2: mongod --port 2737 --replSet "myReplicaSet" --dbpath "/data/secondary2"

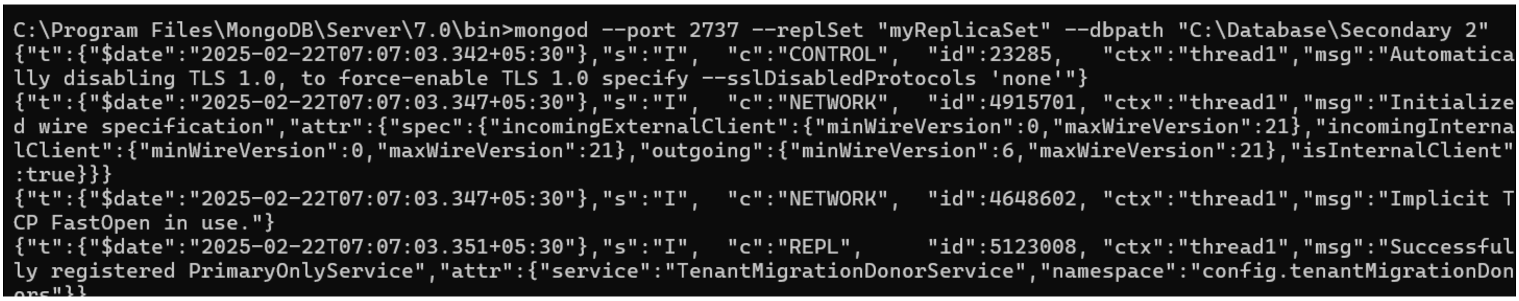
#Note: Open three separate cmds for above three servers.



```
C:\Program Files\MongoDB\Server\7.0\bin>mongod --port 2717 --replSet "myReplicaSet" --dbpath "C:\Database\Primary"
{"t":{"$date":"2025-02-22T07:04:40.592+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatica
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-22T07:04:40.595+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialize
d wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInterna
lClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient"
:true}}}
{"t":{"$date":"2025-02-22T07:04:40.601+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit T
CP FastOpen in use."}
{"t":{"$date":"2025-02-22T07:04:40.628+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDon
ors"}}
{"t":{"$date":"2025-02-22T07:04:40.628+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigratio
nRecipients"}}
{"t":{"$date":"2025-02-22T07:04:40.629+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi thre
ading initialized"}
{"t":{"$date":"2025-02-22T07:04:40.630+05:30"},"s":"I",  "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting T
enantMigrationAccessBlockerRegistry"}
```



```
C:\Program Files\MongoDB\Server\7.0\bin>mongod --port 2727 --replSet "myReplicaSet" --dbpath "C:\Database\Secondary 1"
{"t":{"$date":"2025-02-22T07:05:54.223+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialize
d wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInterna
lClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient"
:true}}}
{"t":{"$date":"2025-02-22T07:05:56.102+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatica
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-22T07:05:56.103+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit T
CP FastOpen in use."}
{"t":{"$date":"2025-02-22T07:05:56.106+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDon
ors"}}
{"t":{"$date":"2025-02-22T07:05:56.106+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigratio
nRecipients"}}
{"t":{"$date":"2025-02-22T07:05:56.106+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi thre
ading initialized"}
{"t":{"$date":"2025-02-22T07:05:56.107+05:30"},"s":"I",  "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting T
enantMigrationAccessBlockerRegistry"}
```



```
C:\Program Files\MongoDB\Server\7.0\bin>mongod --port 2737 --replSet "myReplicaSet" --dbpath "C:\Database\Secondary 2"
{"t":{"$date":"2025-02-22T07:07:03.342+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatica
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-22T07:07:03.347+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialize
d wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInterna
lClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVersion":6,"maxWireVersion":21},"isInternalClient"
:true}}}
{"t":{"$date":"2025-02-22T07:07:03.347+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit T
CP FastOpen in use."}
{"t":{"$date":"2025-02-22T07:07:03.351+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successful
ly registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDon
ors"}}
```

Step 2: Open cmd from bin and type the below command to connect the primary server:

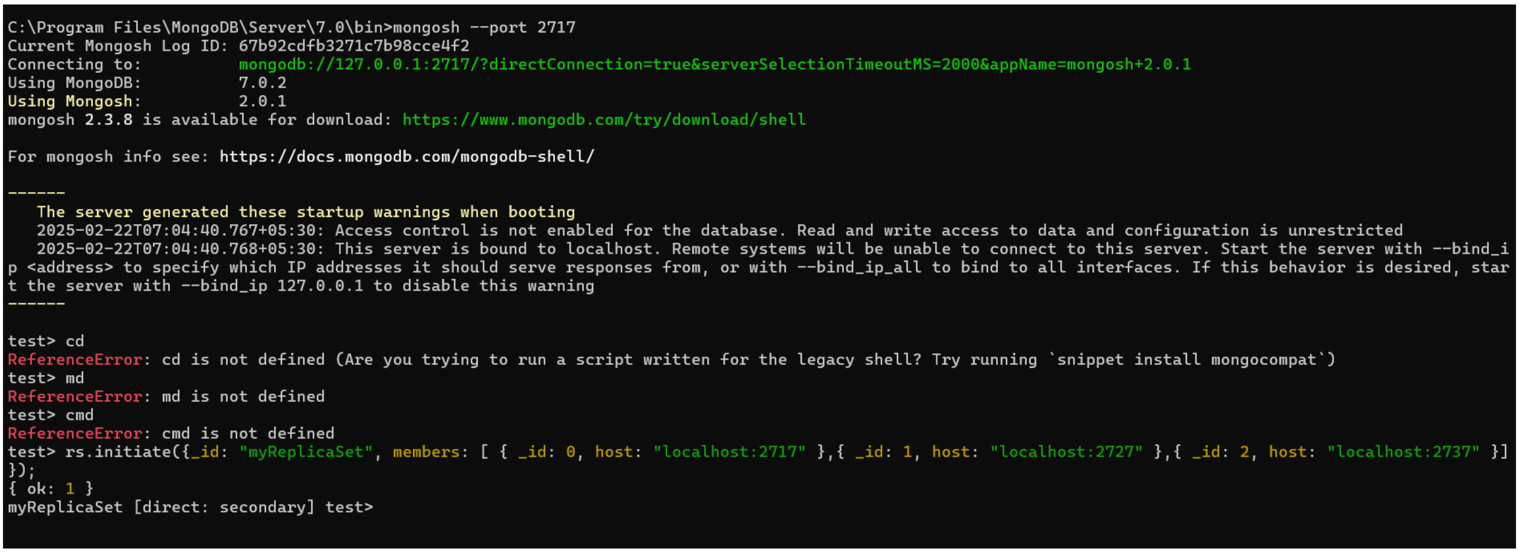
>mongosh --port 2717

Step 3: Initiate the Replica Set.

-Open same cmd where you have connected the primary server ie from step 2

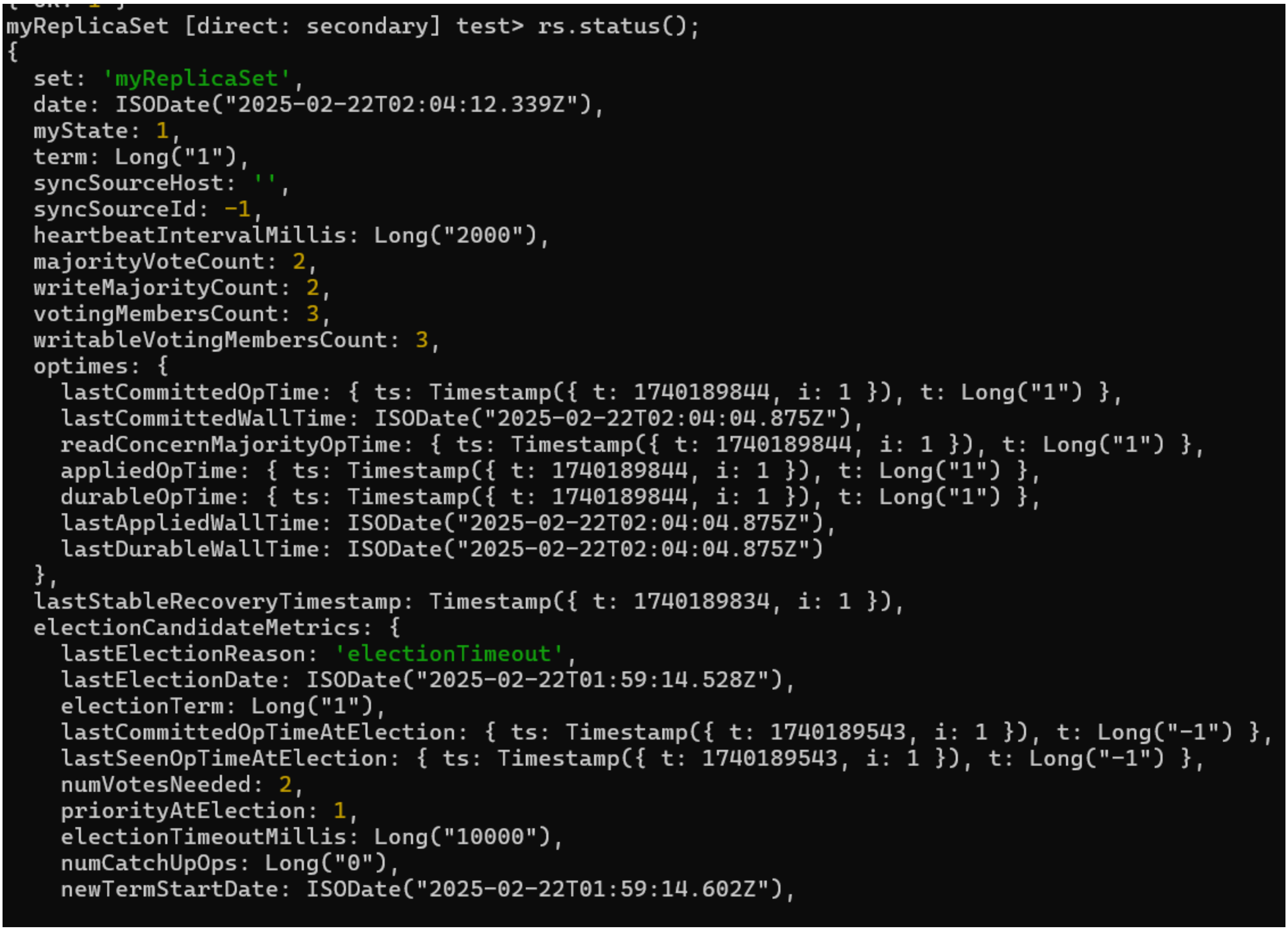
-Write the below command:

rs.initiate({ _id: "myReplicaSet", members: [{ _id: 0, host: "localhost:2717" }, { _id: 1, host: "localhost:2727" }, { _id: 2, host: "localhost:2737" }] });



```
C:\Program Files\MongoDB\Server\7.0\bin>mongosh --port 2717
Current Mongosh Log ID: 67b92cdfb3271c7b98cce4f2
Connecting to:          mongodb://127.0.0.1:2717/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.1
Using MongoDB:          7.0.2
Using Mongosh:          2.0.1
mongosh 2.3.8 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-02-22T07:04:40.767+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2025-02-22T07:04:40.768+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_i
p <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, star
t the server with --bind_ip 127.0.0.1 to disable this warning
------

test> cd
ReferenceError: cd is not defined (Are you trying to run a script written for the legacy shell? Try running `snippet install mongocompat`)
test> md
ReferenceError: md is not defined
test> cmd
ReferenceError: cmd is not defined
test> rs.initiate({_id: "myReplicaSet", members: [ { _id: 0, host: "localhost:2717" },{ _id: 1, host: "localhost:2727" },{ _id: 2, host: "localhost:2737" }]
});
{ ok: 1 }
myReplicaSet [direct: secondary] test>
```

Now,check the status:



```
myReplicaSet [direct: secondary] test> rs.status();
{
  set: 'myReplicaSet',
  date: ISODate("2025-02-22T02:04:12.339Z"),
  myState: 1,
  term: Long("1"),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long("2000"),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1740189844, i: 1 }), t: Long("1") },
    lastCommittedWallTime: ISODate("2025-02-22T02:04:04.875Z"),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1740189844, i: 1 }), t: Long("1") },
    appliedOpTime: { ts: Timestamp({ t: 1740189844, i: 1 }), t: Long("1") },
    durableOpTime: { ts: Timestamp({ t: 1740189844, i: 1 }), t: Long("1") },
    lastAppliedWallTime: ISODate("2025-02-22T02:04:04.875Z"),
    lastDurableWallTime: ISODate("2025-02-22T02:04:04.875Z")
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1740189834, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate("2025-02-22T01:59:14.528Z"),
    electionTerm: Long("1"),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1740189543, i: 1 }), t: Long("-1") },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1740189543, i: 1 }), t: Long("-1") },
    numVotesNeeded: 2,
    priorityAtElection: 1,
    electionTimeoutMillis: Long("10000"),
    numCatchUpOps: Long("0"),
    newTermStartDate: ISODate("2025-02-22T01:59:14.602Z"),
```

SHARDING

Step 1: Create Database Folders (Shard1, Shard2, Shard3).

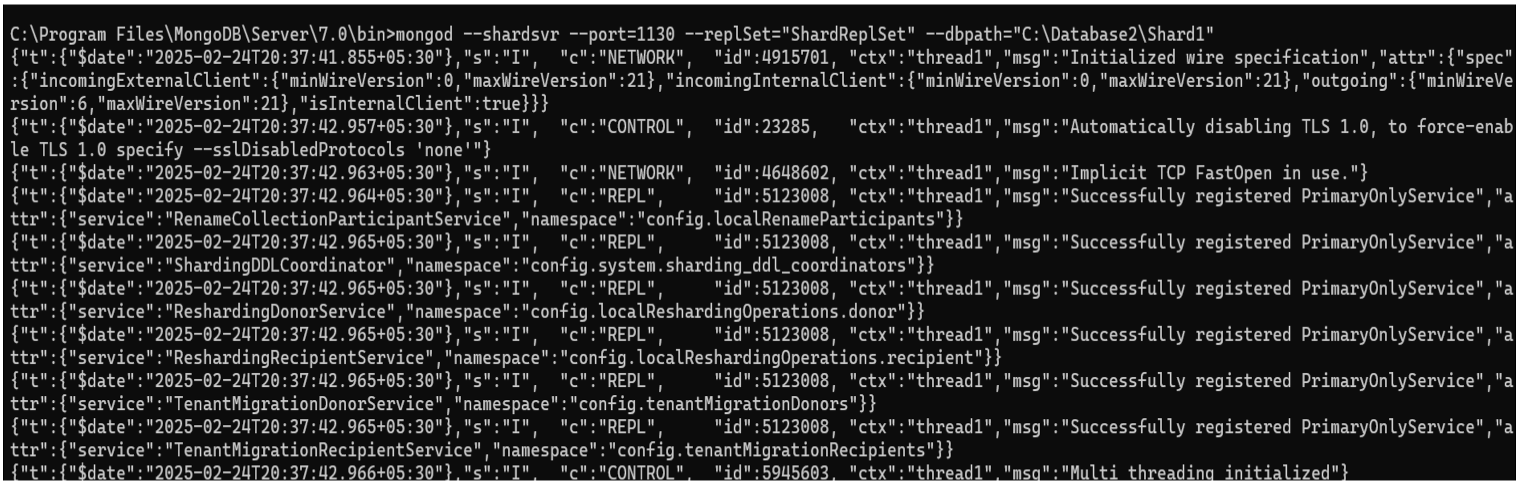
-Open MongoDB terminal (cmd) from bin, and follow the below commands:

*For Shard1: mongod --shardsvr --port=1130 --replSet="ShardReplSet" --dbpath="C:/data/shard1"

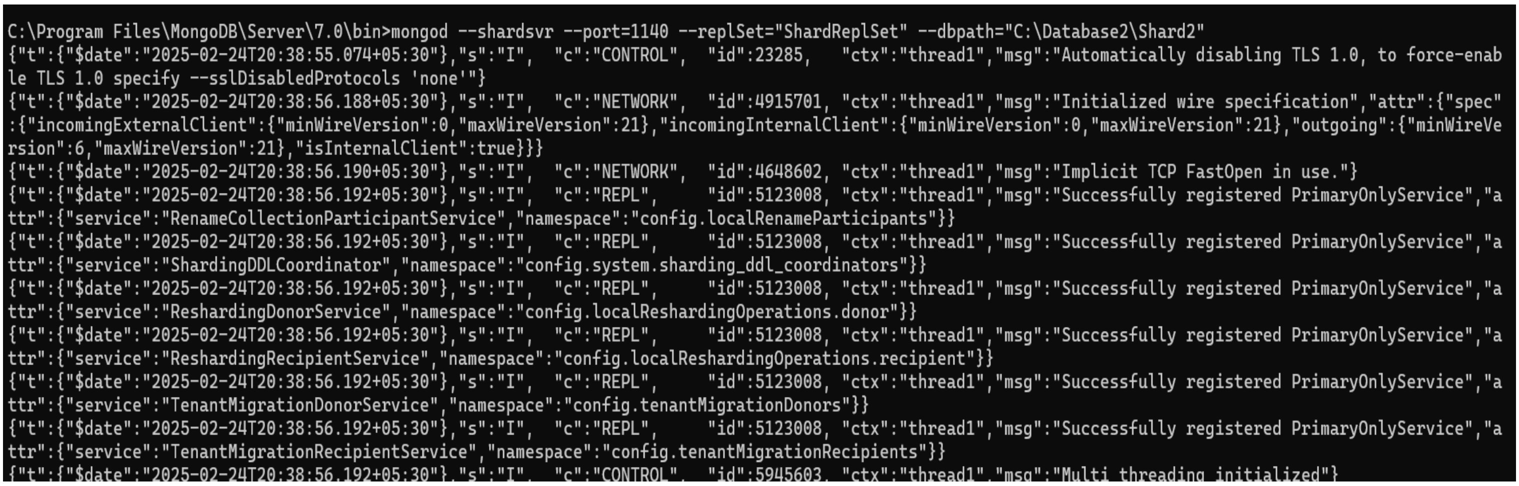
*For Shard2: mongod --shardsvr --port=1140 --replSet="ShardReplSet" --dbpath="C:/data/shard2"

*For Shard3 : mongod --shardsvr --port=1150 --replSet="ShardReplSet" --dbpath="C:/data/shard3"

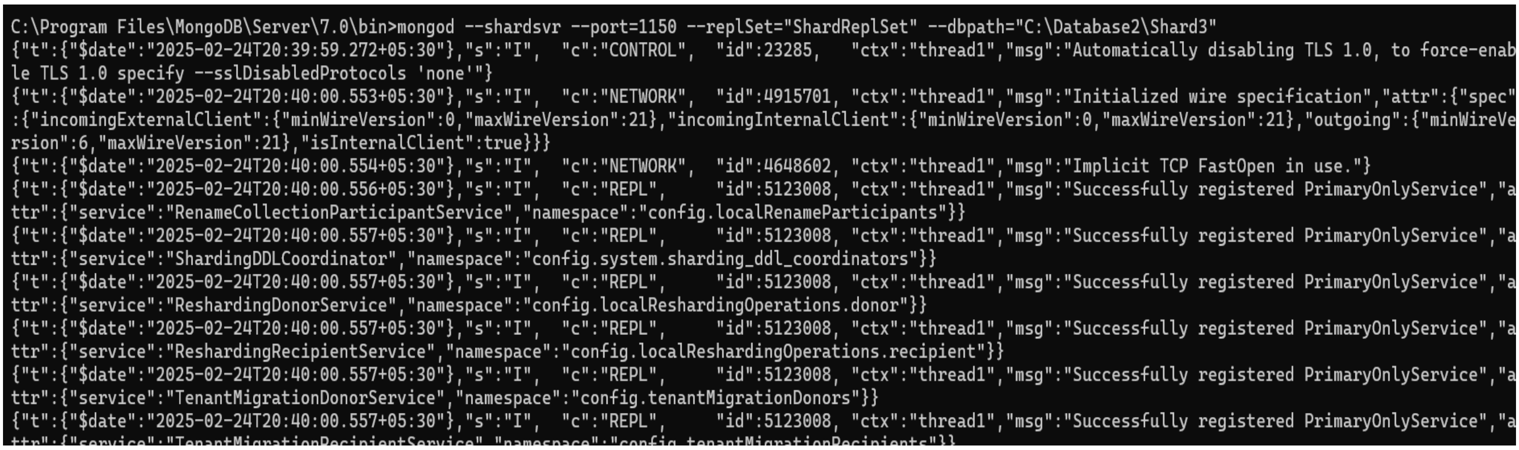
#Note: Open three separate cmds for above three servers.



```
C:\Program Files\MongoDB\Server\7.0\bin>mongod --shardsvr --port=1130 --replSet="ShardReplSet" --dbpath="C:\Database2\Shard1"
{"t":{"$date":"2025-02-24T20:37:41.855+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec"
:{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVe
rsion":6,"maxWireVersion":21},"isInternalClient":true}}}
{"t":{"$date":"2025-02-24T20:37:42.957+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enab
le TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-24T20:37:42.963+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2025-02-24T20:37:42.964+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"RenameCollectionParticipantService","namespace":"config.localRenameParticipants"}}
{"t":{"$date":"2025-02-24T20:37:42.965+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ShardingDDLCoordinator","namespace":"config.system.sharding_ddl_coordinators"}}
{"t":{"$date":"2025-02-24T20:37:42.965+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ReshardingDonorService","namespace":"config.localReshardingOperations.donor"}}
{"t":{"$date":"2025-02-24T20:37:42.965+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ReshardingRecipientService","namespace":"config.localReshardingOperations.recipient"}}
{"t":{"$date":"2025-02-24T20:37:42.965+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
{"t":{"$date":"2025-02-24T20:37:42.965+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2025-02-24T20:37:42.966+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
```



```
C:\Program Files\MongoDB\Server\7.0\bin>mongod --shardsvr --port=1140 --replSet="ShardReplSet" --dbpath="C:\Database2\Shard2"
{"t":{"$date":"2025-02-24T20:38:55.074+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enab
le TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-24T20:38:56.188+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec"
:{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVe
rsion":6,"maxWireVersion":21},"isInternalClient":true}}}
{"t":{"$date":"2025-02-24T20:38:56.190+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2025-02-24T20:38:56.192+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"RenameCollectionParticipantService","namespace":"config.localRenameParticipants"}}
{"t":{"$date":"2025-02-24T20:38:56.192+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ShardingDDLCoordinator","namespace":"config.system.sharding_ddl_coordinators"}}
{"t":{"$date":"2025-02-24T20:38:56.192+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ReshardingDonorService","namespace":"config.localReshardingOperations.donor"}}
{"t":{"$date":"2025-02-24T20:38:56.192+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ReshardingRecipientService","namespace":"config.localReshardingOperations.recipient"}}
{"t":{"$date":"2025-02-24T20:38:56.192+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
{"t":{"$date":"2025-02-24T20:38:56.192+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2025-02-24T20:38:56.192+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
```

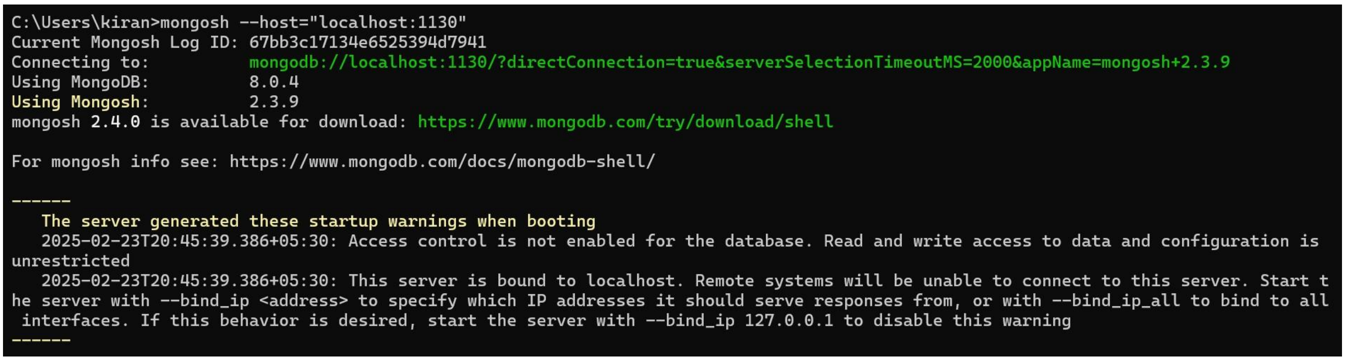


```
C:\Program Files\MongoDB\Server\7.0\bin>mongod --shardsvr --port=1150 --replSet="ShardReplSet" --dbpath="C:\Database2\Shard3"
{"t":{"$date":"2025-02-24T20:39:59.272+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enab
le TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-24T20:40:00.553+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec"
:{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVe
rsion":6,"maxWireVersion":21},"isInternalClient":true}}}
{"t":{"$date":"2025-02-24T20:40:00.554+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2025-02-24T20:40:00.556+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"RenameCollectionParticipantService","namespace":"config.localRenameParticipants"}}
{"t":{"$date":"2025-02-24T20:40:00.557+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ShardingDDLCoordinator","namespace":"config.system.sharding_ddl_coordinators"}}
{"t":{"$date":"2025-02-24T20:40:00.557+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ReshardingDonorService","namespace":"config.localReshardingOperations.donor"}}
{"t":{"$date":"2025-02-24T20:40:00.557+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"ReshardingRecipientService","namespace":"config.localReshardingOperations.recipient"}}
{"t":{"$date":"2025-02-24T20:40:00.557+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
{"t":{"$date":"2025-02-24T20:40:00.557+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","a
ttr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}
```

Step 2: Connect to One Shard & Initiate Replica Set.

Open MongoDB cmd :

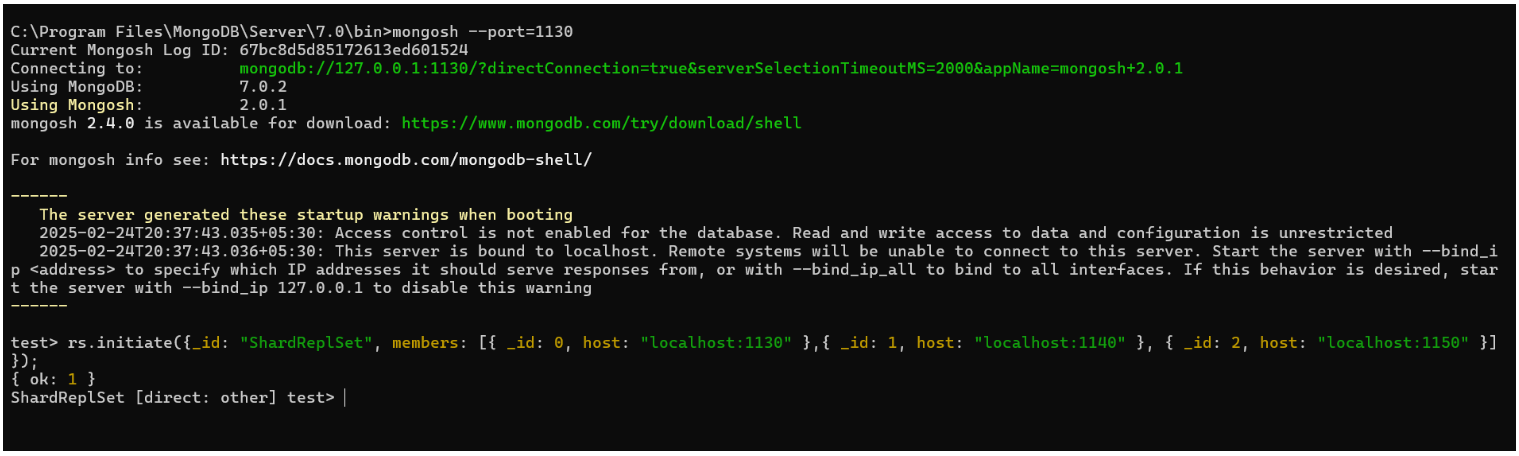
>mongosh --port=1130



```
C:\Users\kiran>mongosh --host="localhost:1130"
Current Mongosh Log ID: 67bb3c17134e6525394d7941
Connecting to:          mongodb://localhost:1130/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.9
Using MongoDB:          8.0.4
Using Mongosh:          2.3.9
mongosh 2.4.0 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-02-23T20:45:39.386+05:30: Access control is not enabled for the database. Read and write access to data and configuration is
unrestricted
   2025-02-23T20:45:39.386+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start t
he 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
------
```

Step 3: In same cmd Write the js code:

```
rs.initiate({
   _id: "ShardReplSet",
   members: [
      { _id: 0, host: "localhost:1130" },
      { _id: 1, host: "localhost:1140" },
      { _id: 2, host: "localhost:1150" }
   ]
});
```

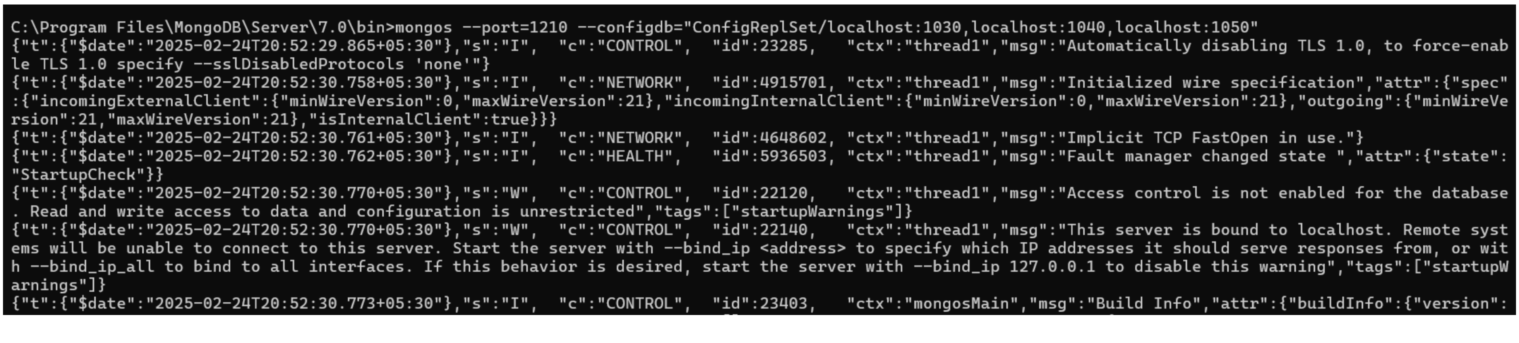


```
C:\Program Files\MongoDB\Server\7.0\bin>mongosh --port=1130
Current Mongosh Log ID: 67bc8d5d85172613ed601524
Connecting to:          mongodb://127.0.0.1:1130/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.1
Using MongoDB:          7.0.2
Using Mongosh:          2.0.1
mongosh 2.4.0 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-02-24T20:37:43.035+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2025-02-24T20:37:43.036+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_i
p <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, star
t the server with --bind_ip 127.0.0.1 to disable this warning
------

test> rs.initiate({_id: "ShardReplSet", members: [{ _id: 0, host: "localhost:1130" },{ _id: 1, host: "localhost:1140" }, { _id: 2, host: "localhost:1150" }]
});
{ ok: 1 }
ShardReplSet [direct: other] test> |
```

****Shard Replica Set is ready…**

Step 4 : Start the Query Router (mongos):

>mongos --port=1210 --configdb="ConfigReplSet/localhost:1030,localhost:1040,localhost:1050"



```
C:\Program Files\MongoDB\Server\7.0\bin>mongos --port=1210 --configdb="ConfigReplSet/localhost:1030,localhost:1040,localhost:1050"
{"t":{"$date":"2025-02-24T20:52:29.865+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enab
le TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-24T20:52:30.758+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec"
:{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"outgoing":{"minWireVe
rsion":21,"maxWireVersion":21},"isInternalClient":true}}}
{"t":{"$date":"2025-02-24T20:52:30.761+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2025-02-24T20:52:30.762+05:30"},"s":"I",  "c":"HEALTH",   "id":5936503, "ctx":"thread1","msg":"Fault manager changed state ","attr":{"state":
"StartupCheck"}}
{"t":{"$date":"2025-02-24T20:52:30.770+05:30"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"thread1","msg":"Access control is not enabled for the database
. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2025-02-24T20:52:30.770+05:30"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"thread1","msg":"This server is bound to localhost. Remote syst
ems 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 wit
h --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":["startupW
arnings"]}
{"t":{"$date":"2025-02-24T20:52:30.773+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"mongosMain","msg":"Build Info","attr":{"buildInfo":{"version":
```

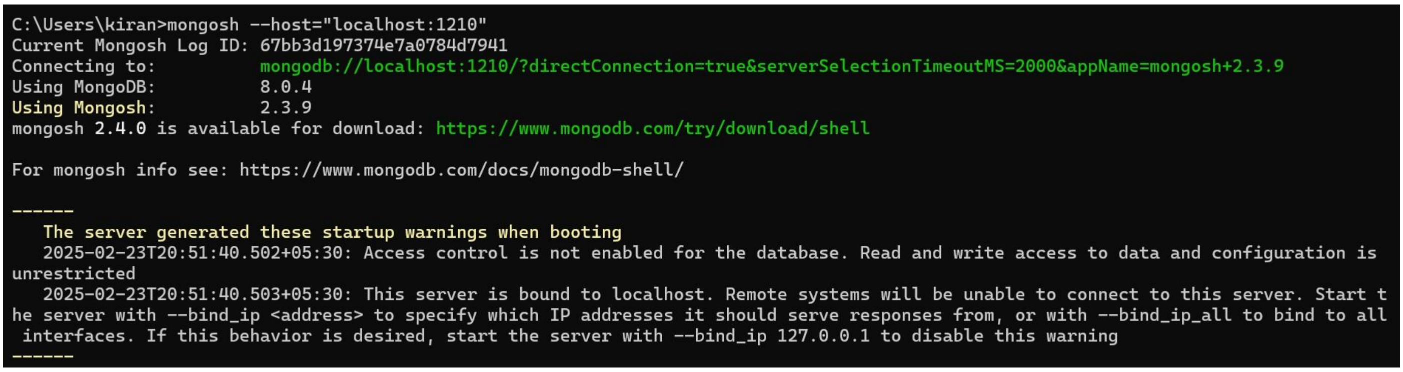
-Query router is ready.

Step 5: Connect Query Router & Add Shards.

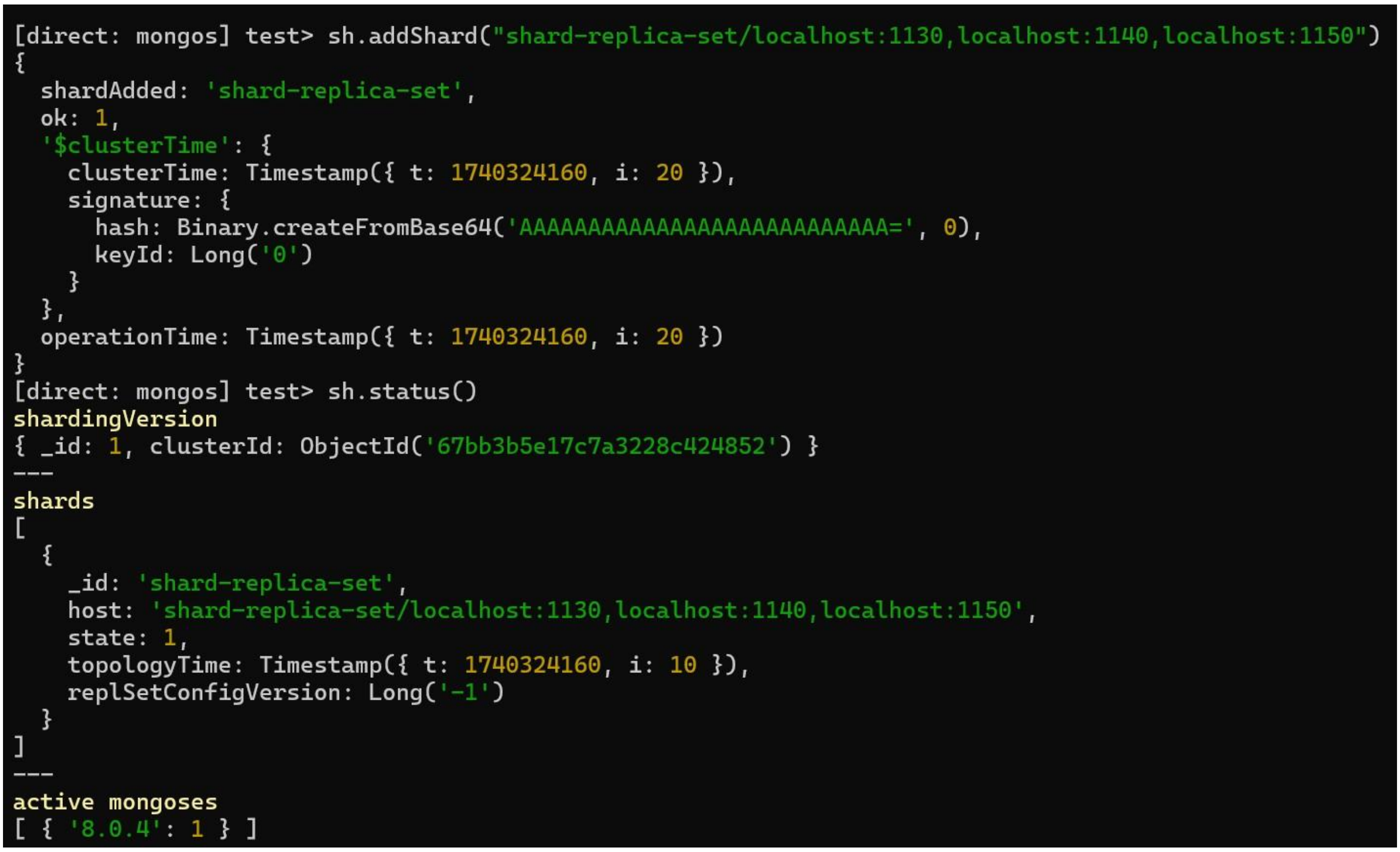
>mongosh --port=1210

On the same cmd write the code,

>sh.addShard("ShardReplSet/localhost:1130,localhost:1140,localhost:1150");



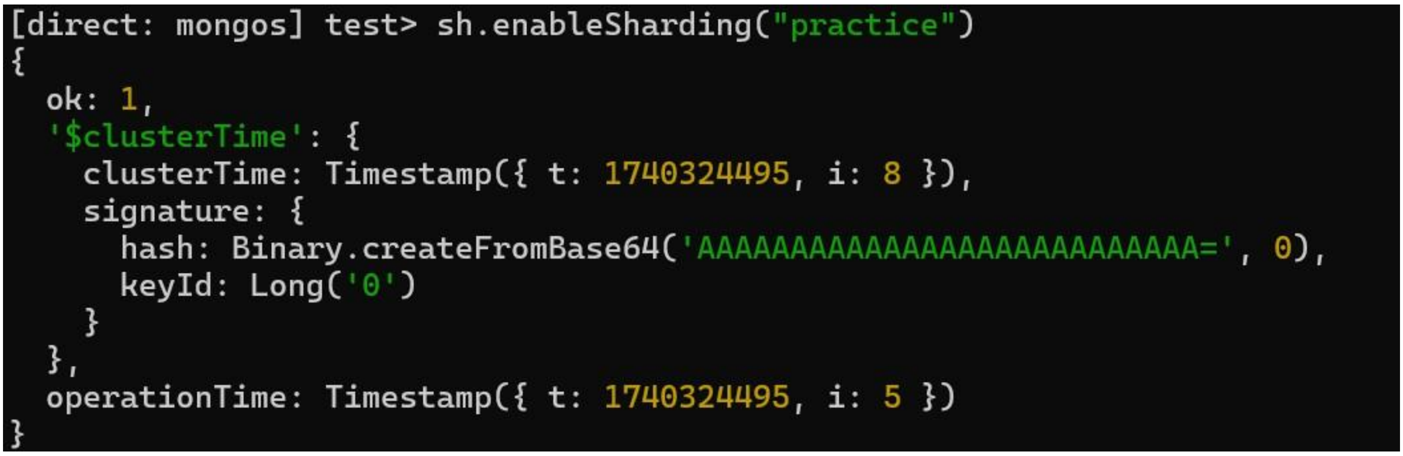
```
C:\Users\kiran>mongosh --host="localhost:1210"
Current Mongosh Log ID: 67bb3d197374e7a0784d7941
Connecting to:          mongodb://localhost:1210/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.9
Using MongoDB:          8.0.4
Using Mongosh:          2.3.9
mongosh 2.4.0 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-02-23T20:51:40.502+05:30: Access control is not enabled for the database. Read and write access to data and configuration is
unrestricted
   2025-02-23T20:51:40.503+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start t
he 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
------
```



```
[direct: mongos] test> sh.addShard("shard-replica-set/localhost:1130,localhost:1140,localhost:1150")
{
  shardAdded: 'shard-replica-set',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1740324160, i: 20 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1740324160, i: 20 })
}
[direct: mongos] test> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('67bb3b5e17c7a3228c424852') }
---
shards
[
  {
    _id: 'shard-replica-set',
    host: 'shard-replica-set/localhost:1130,localhost:1140,localhost:1150',
    state: 1,
    topologyTime: Timestamp({ t: 1740324160, i: 10 }),
    replSetConfigVersion: Long('-1')
  }
]
---
active mongoses
[ { '8.0.4': 1 } ]
```

Step 6: Enable Sharding on a Database

>sh.enableSharding("practice");



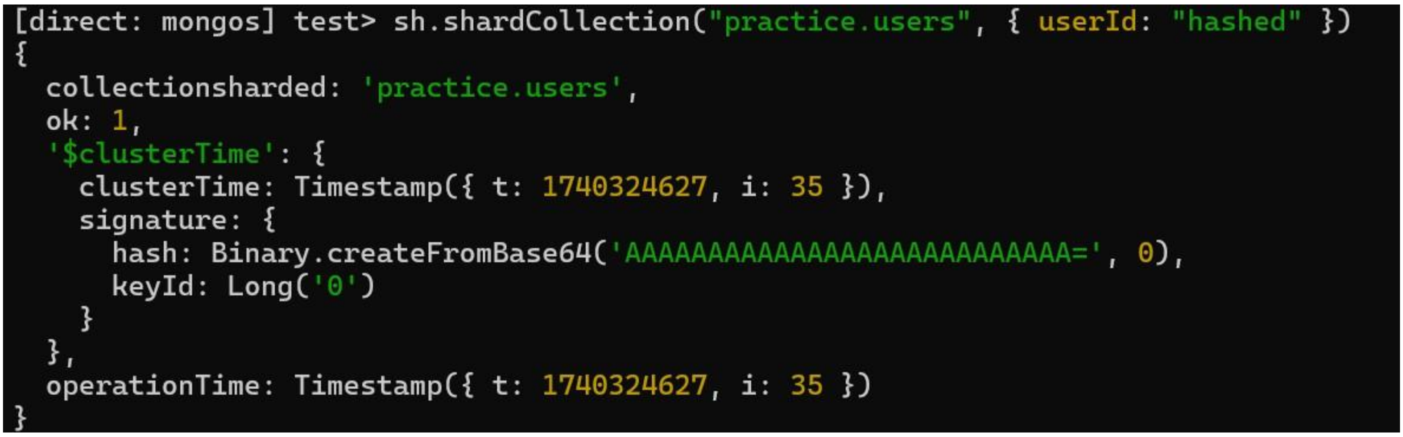
```
[direct: mongos] test> sh.enableSharding("practice")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1740324495, i: 8 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1740324495, i: 5 })
}
```

-Sharding enabled for the database practice!

Step 7: Shard a Collection

Hashed-based sharding (Random Distribution)

>sh.shardCollection("practice.users", { "userId": "hashed" });

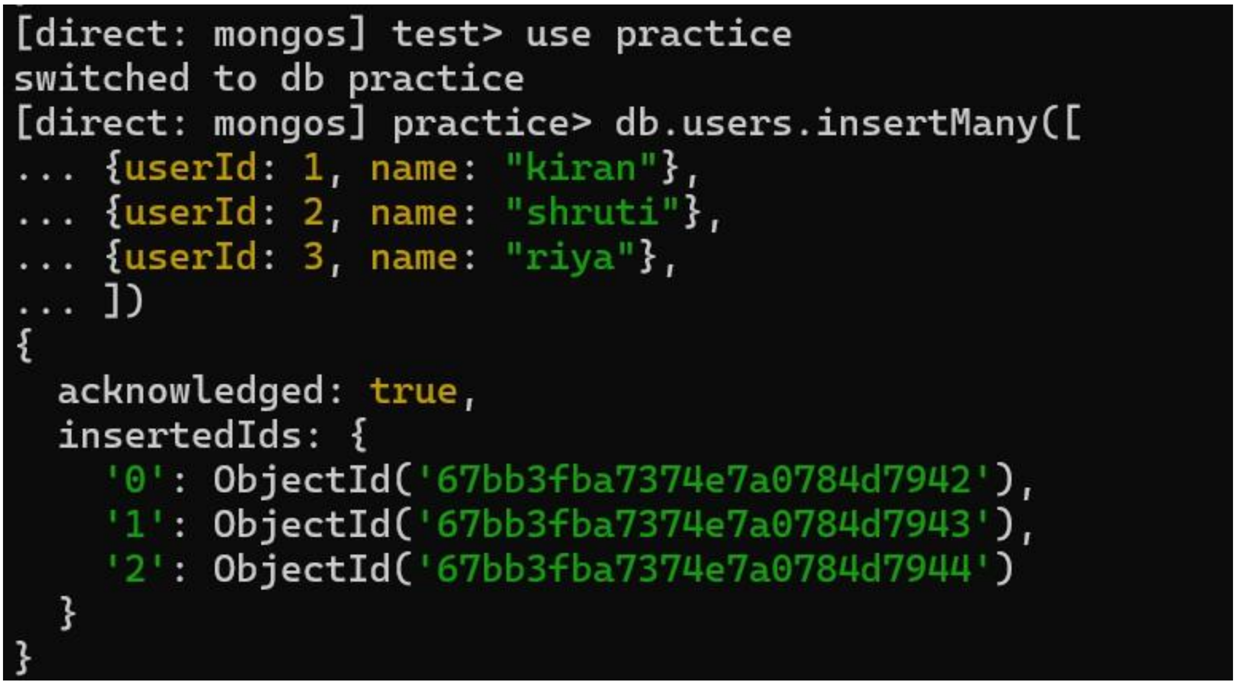


```
[direct: mongos] test> sh.shardCollection("practice.users", { userId: "hashed" })
{
  collectionsharded: 'practice.users',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1740324627, i: 35 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1740324627, i: 35 })
}
```

-**Collection students is now sharded!**

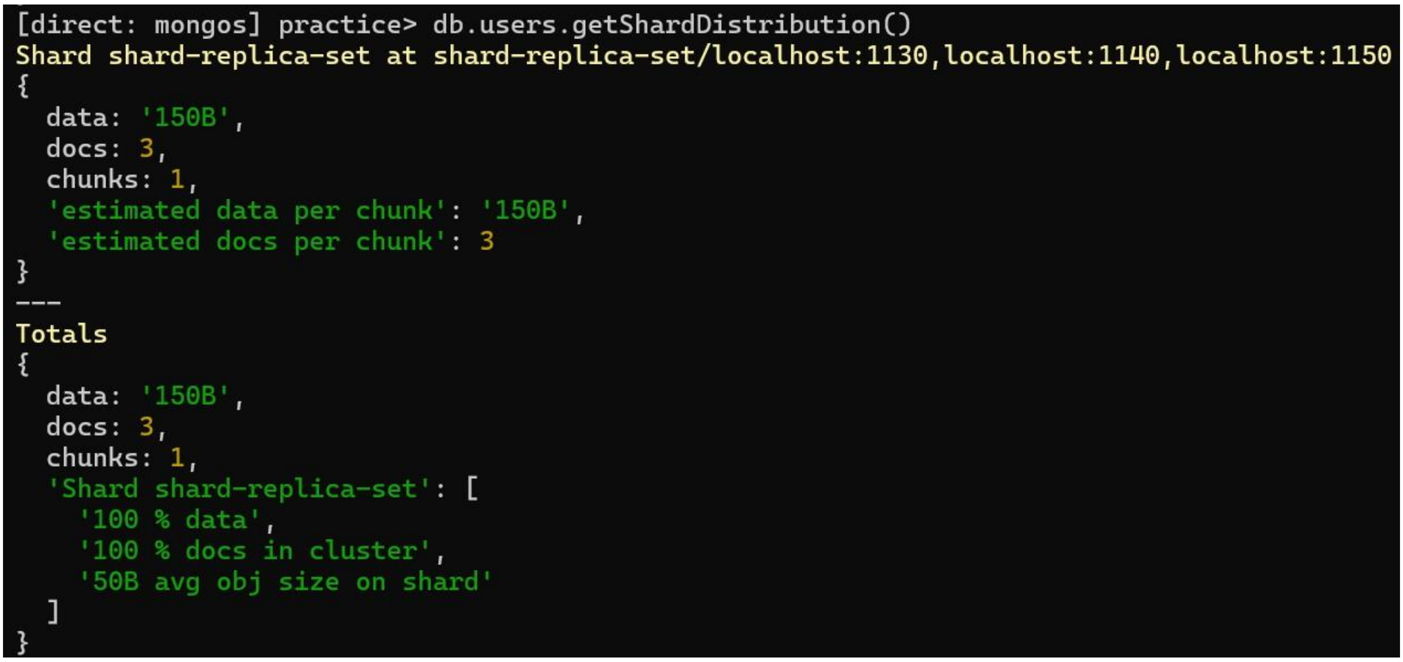
Step 8: Insert Sample Data and Verify

>use practice



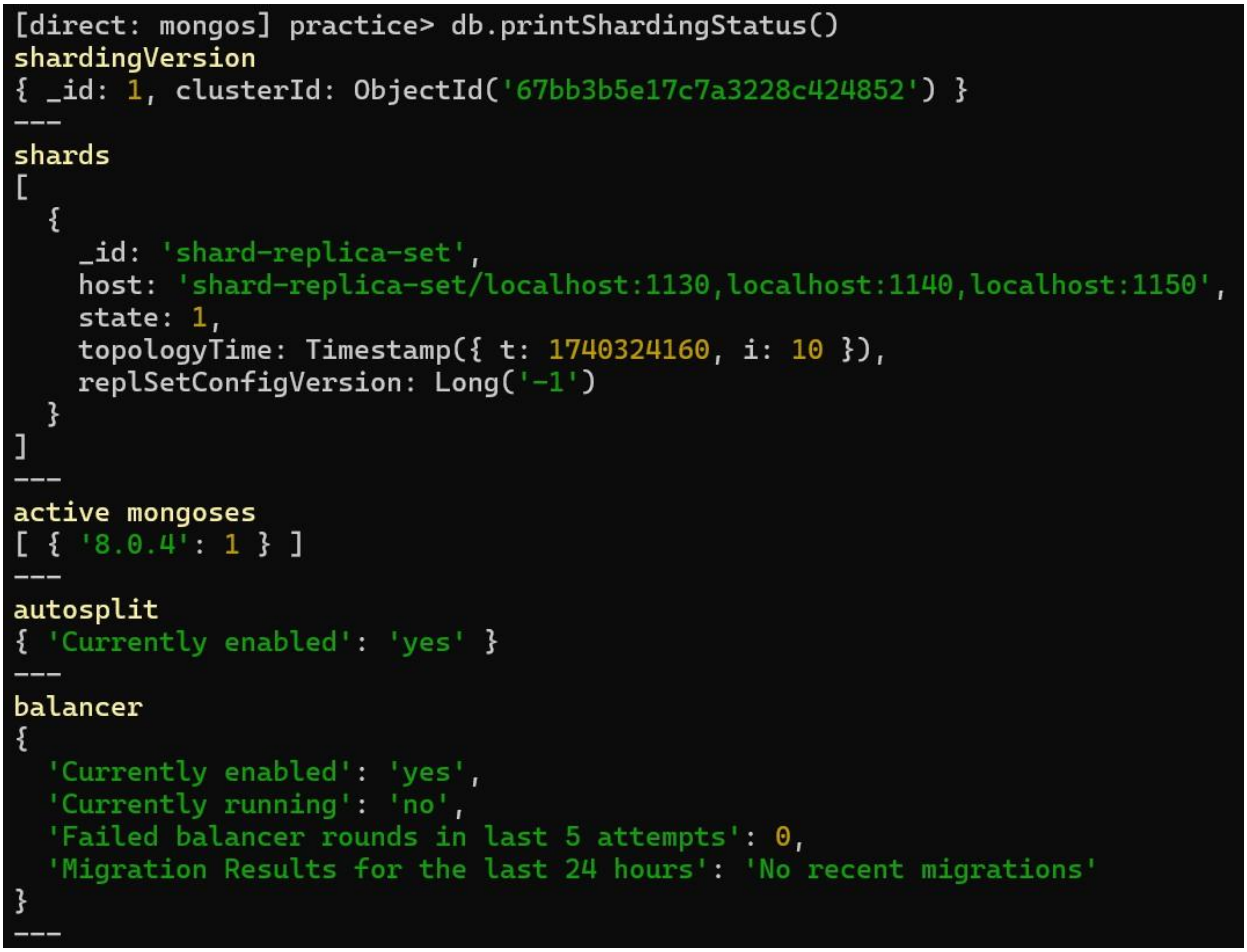
```
[direct: mongos] test> use practice
switched to db practice
[direct: mongos] practice> db.users.insertMany([
... {userId: 1, name: "kiran"},
... {userId: 2, name: "shruti"},
... {userId: 3, name: "riya"},
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67bb3fba7374e7a0784d7942'),
    '1': ObjectId('67bb3fba7374e7a0784d7943'),
    '2': ObjectId('67bb3fba7374e7a0784d7944')
  }
}
```

Step 9: To check where documents are stored

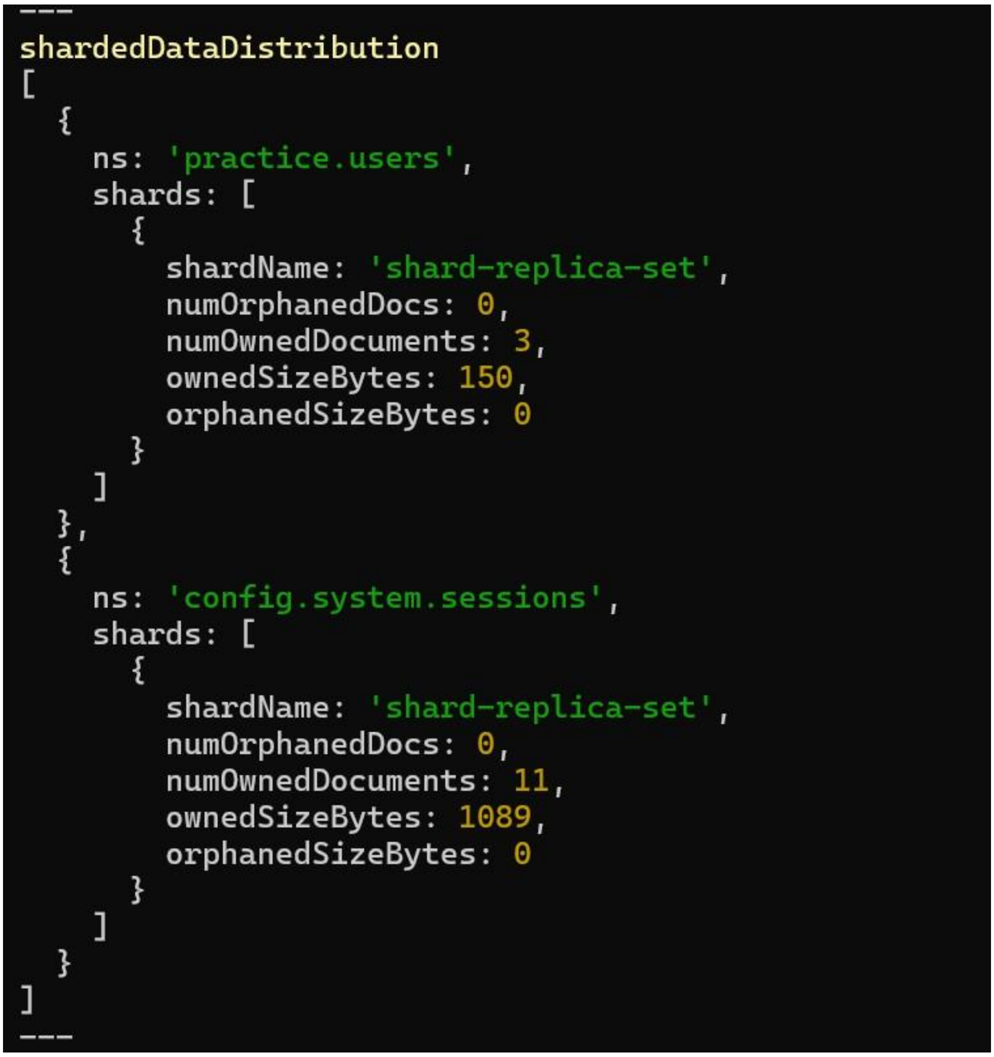


```
[direct: mongos] practice> db.users.getShardDistribution()
Shard shard-replica-set at shard-replica-set/localhost:1130,localhost:1140,localhost:1150
{
  data: '150B',
  docs: 3,
  chunks: 1,
  'estimated data per chunk': '150B',
  'estimated docs per chunk': 3
}
---
Totals
{
  data: '150B',
  docs: 3,
  chunks: 1,
  'Shard shard-replica-set': [
    '100 % data',
    '100 % docs in cluster',
    '50B avg obj size on shard'
  ]
}
```

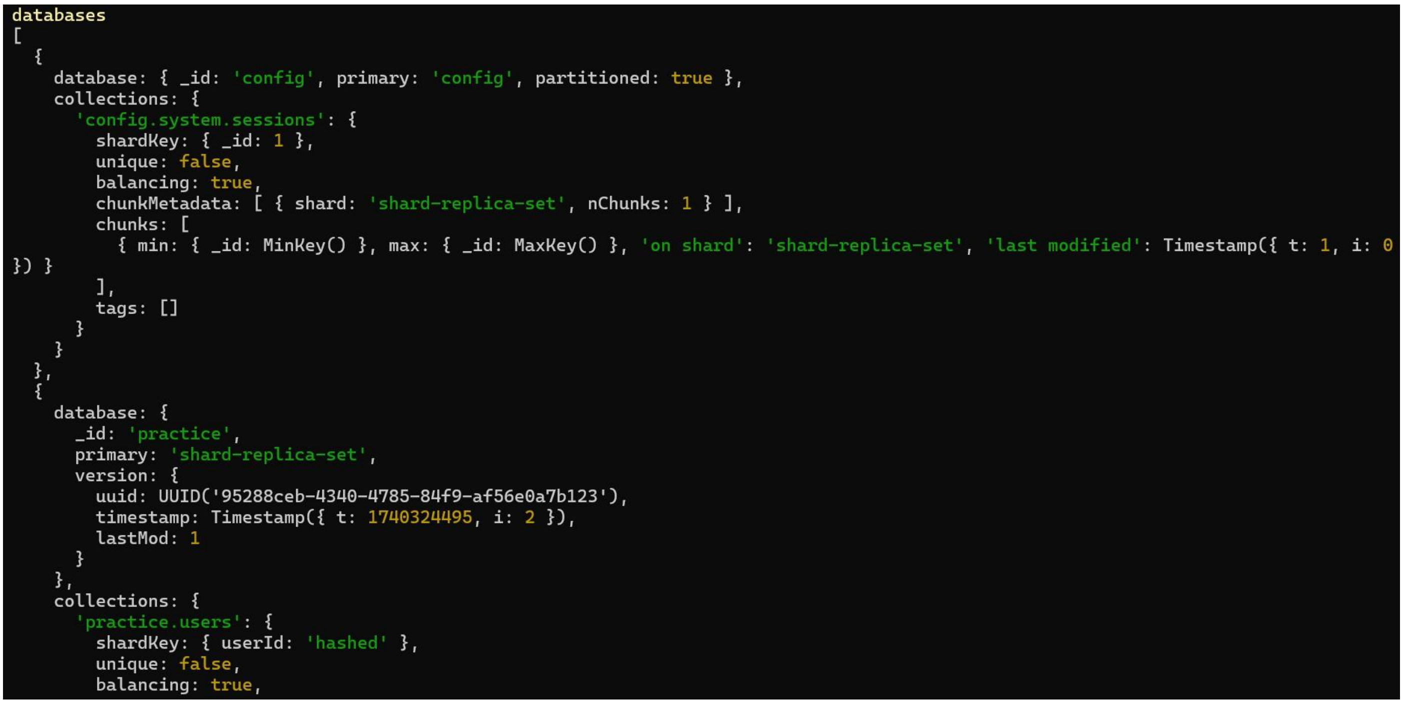
-To check collection-level sharding



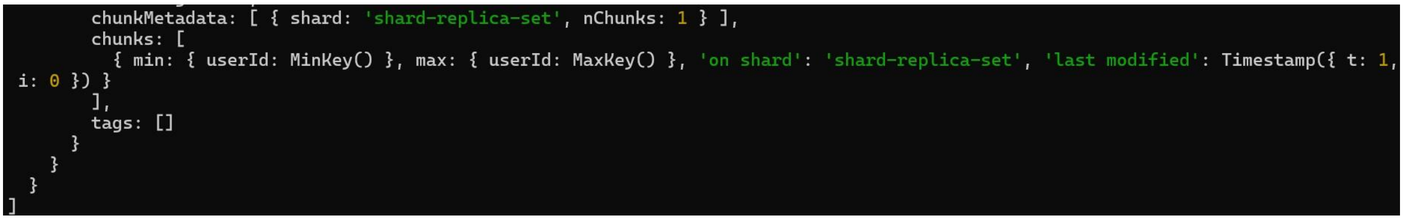
```
[direct: mongos] practice> db.printShardingStatus()
shardingVersion
{ _id: 1, clusterId: ObjectId('67bb3b5e17c7a3228c424852') }
---
shards
[
  {
    _id: 'shard-replica-set',
    host: 'shard-replica-set/localhost:1130,localhost:1140,localhost:1150',
    state: 1,
    topologyTime: Timestamp({ t: 1740324160, i: 10 }),
    replSetConfigVersion: Long('-1')
  }
]
---
active mongoses
[ { '8.0.4': 1 } ]
---
autosplit
{ 'Currently enabled': 'yes' }
---
balancer
{
  'Currently enabled': 'yes',
  'Currently running': 'no',
  'Failed balancer rounds in last 5 attempts': 0,
  'Migration Results for the last 24 hours': 'No recent migrations'
}
---
```



```
---
shardedDataDistribution
[
  {
    ns: 'practice.users',
    shards: [
      {
        shardName: 'shard-replica-set',
        numOrphanedDocs: 0,
        numOwnedDocuments: 3,
        ownedSizeBytes: 150,
        orphanedSizeBytes: 0
      }
    ]
  },
  {
    ns: 'config.system.sessions',
    shards: [
      {
        shardName: 'shard-replica-set',
        numOrphanedDocs: 0,
        numOwnedDocuments: 11,
        ownedSizeBytes: 1089,
        orphanedSizeBytes: 0
      }
    ]
  }
]
---
```

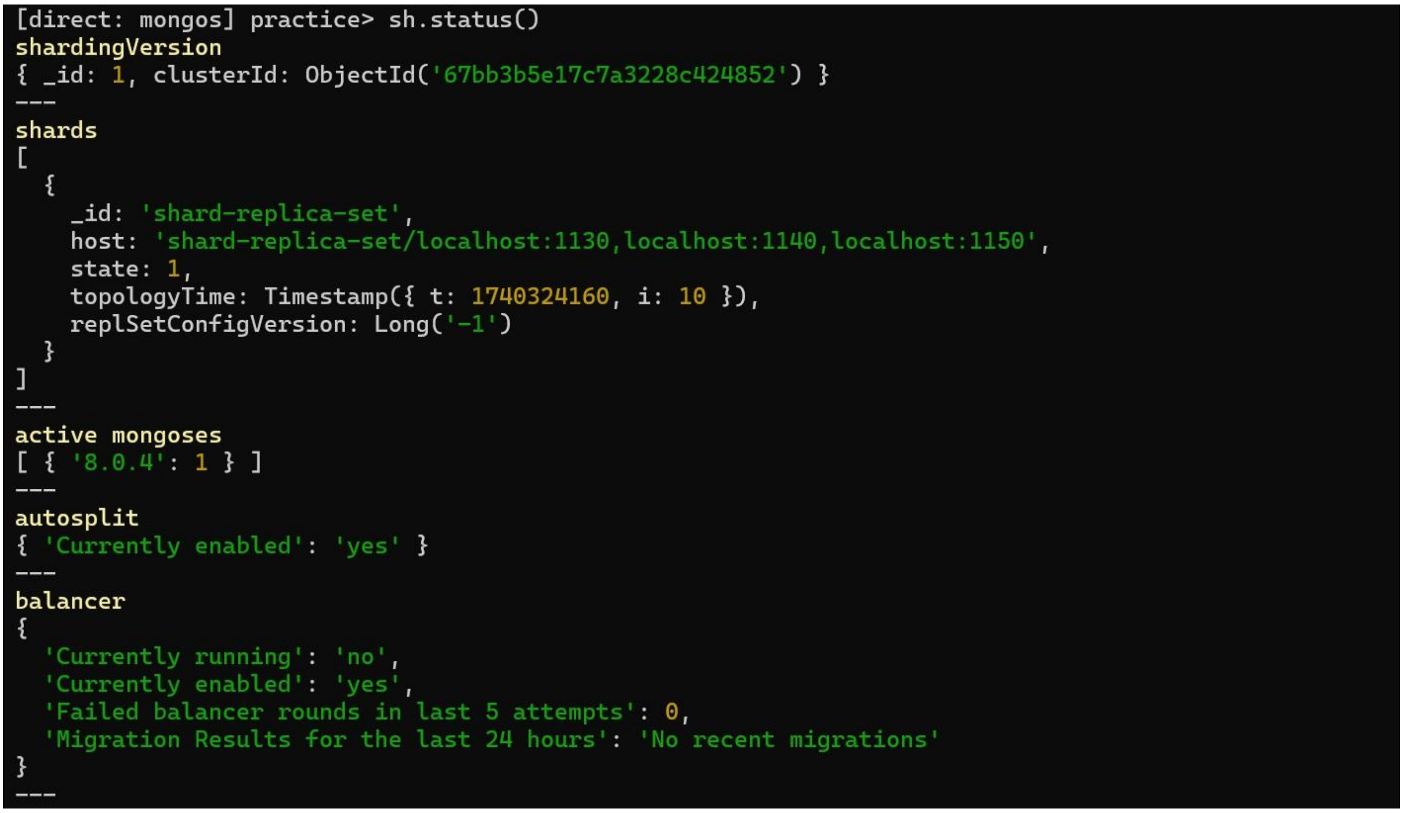


```
databases
[
  {
    database: { _id: 'config', primary: 'config', partitioned: true },
    collections: {
      'config.system.sessions': {
        shardKey: { _id: 1 },
        unique: false,
        balancing: true,
        chunkMetadata: [ { shard: 'shard-replica-set', nChunks: 1 } ],
        chunks: [
          { min: { _id: MinKey() }, max: { _id: MaxKey() }, 'on shard': 'shard-replica-set', 'last modified': Timestamp({ t: 1, i: 0
}) }
        ],
        tags: []
      }
    }
  },
  {
    database: {
      _id: 'practice',
      primary: 'shard-replica-set',
      version: {
        uuid: UUID('95288ceb-4340-4785-84f9-af56e0a7b123'),
        timestamp: Timestamp({ t: 1740324495, i: 2 }),
        lastMod: 1
      }
    },
    collections: {
      'practice.users': {
        shardKey: { userId: 'hashed' },
        unique: false,
        balancing: true,
```

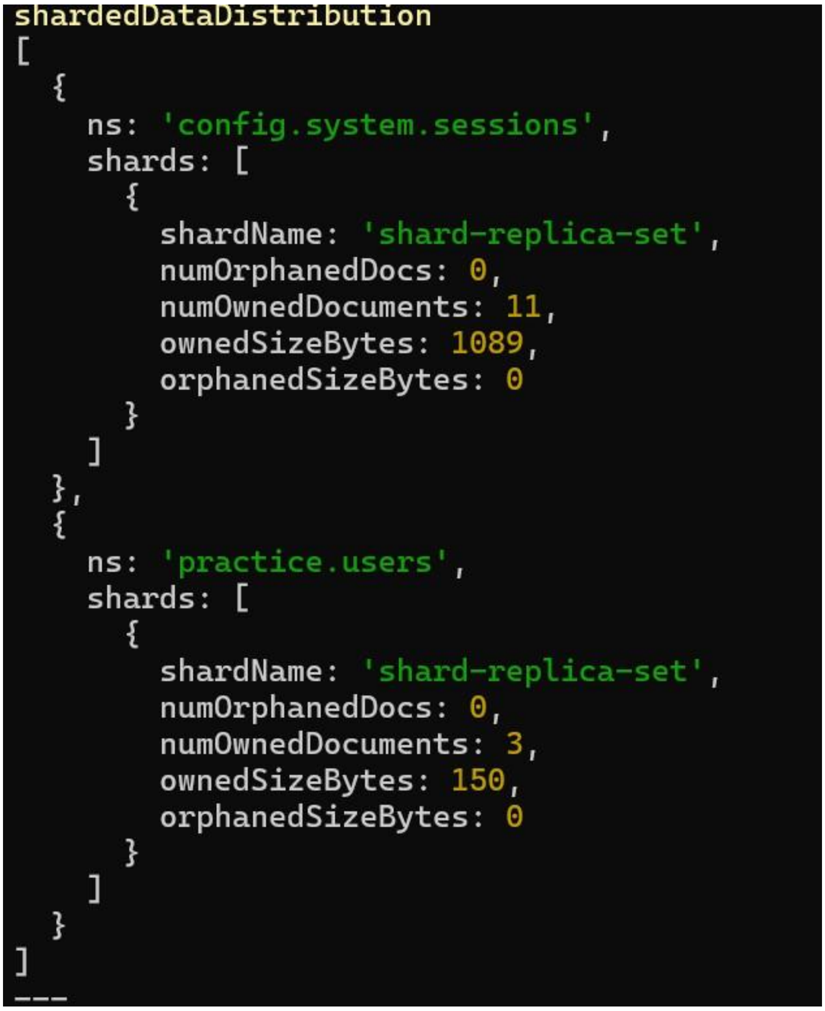


```
        chunkMetadata: [ { shard: 'shard-replica-set', nChunks: 1 } ],
        chunks: [
          { min: { userId: MinKey() }, max: { userId: MaxKey() }, 'on shard': 'shard-replica-set', 'last modified': Timestamp({ t: 1,
 i: 0 }) }
        ],
        tags: []
      }
    }
  }
]
```

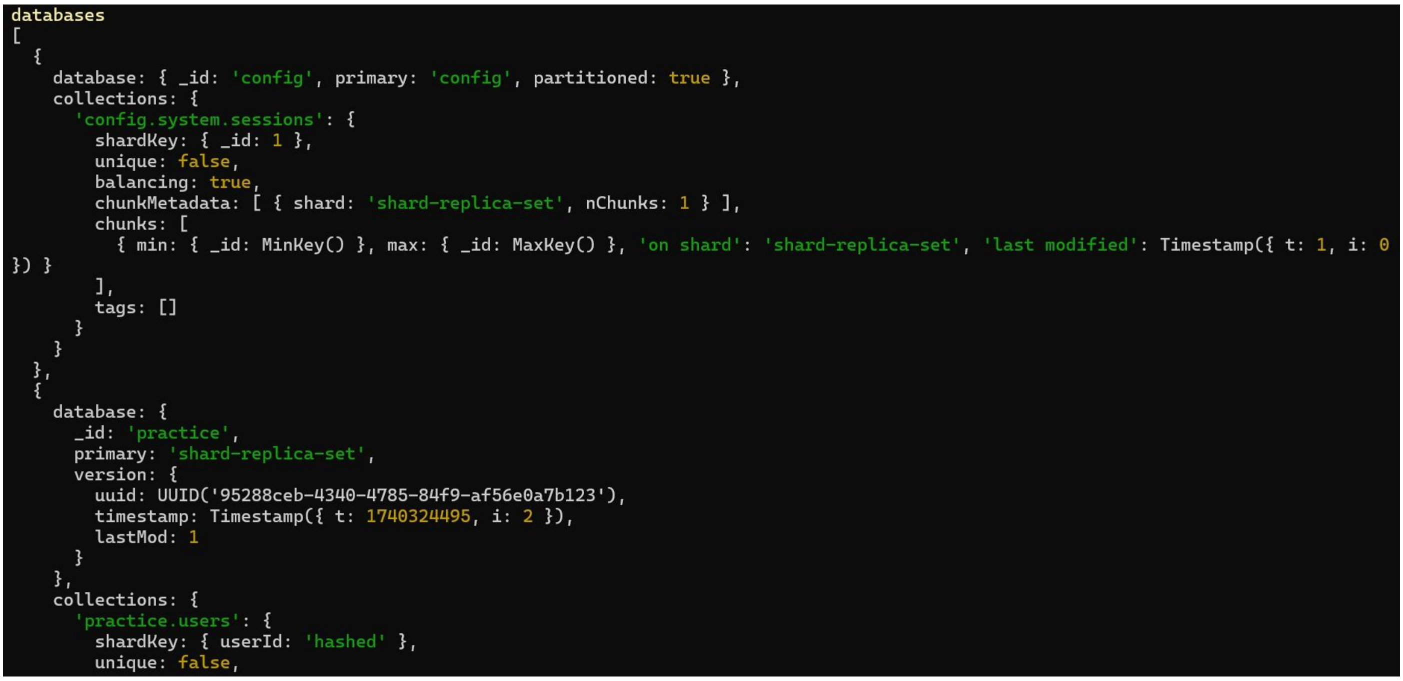
-To check overall cluster health



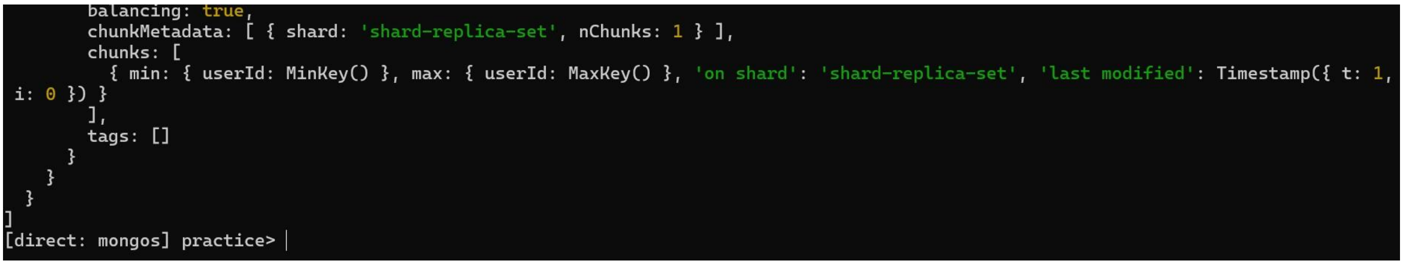
```
[direct: mongos] practice> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('67bb3b5e17c7a3228c424852') }
---
shards
[
  {
    _id: 'shard-replica-set',
    host: 'shard-replica-set/localhost:1130,localhost:1140,localhost:1150',
    state: 1,
    topologyTime: Timestamp({ t: 1740324160, i: 10 }),
    replSetConfigVersion: Long('-1')
  }
]
---
active mongoses
[ { '8.0.4': 1 } ]
---
autosplit
{ 'Currently enabled': 'yes' }
---
balancer
{
  'Currently running': 'no',
  'Currently enabled': 'yes',
  'Failed balancer rounds in last 5 attempts': 0,
  'Migration Results for the last 24 hours': 'No recent migrations'
}
---
```



```
shardedDataDistribution
[
  {
    ns: 'config.system.sessions',
    shards: [
      {
        shardName: 'shard-replica-set',
        numOrphanedDocs: 0,
        numOwnedDocuments: 11,
        ownedSizeBytes: 1089,
        orphanedSizeBytes: 0
      }
    ]
  },
  {
    ns: 'practice.users',
    shards: [
      {
        shardName: 'shard-replica-set',
        numOrphanedDocs: 0,
        numOwnedDocuments: 3,
        ownedSizeBytes: 150,
        orphanedSizeBytes: 0
      }
    ]
  }
]
---
```



```
databases
[
  {
    database: { _id: 'config', primary: 'config', partitioned: true },
    collections: {
      'config.system.sessions': {
        shardKey: { _id: 1 },
        unique: false,
        balancing: true,
        chunkMetadata: [ { shard: 'shard-replica-set', nChunks: 1 } ],
        chunks: [
          { min: { _id: MinKey() }, max: { _id: MaxKey() }, 'on shard': 'shard-replica-set', 'last modified': Timestamp({ t: 1, i: 0
}) }
        ],
        tags: []
      }
    }
  },
  {
    database: {
      _id: 'practice',
      primary: 'shard-replica-set',
      version: {
        uuid: UUID('95288ceb-4340-4785-84f9-af56e0a7b123'),
        timestamp: Timestamp({ t: 1740324495, i: 2 }),
        lastMod: 1
      }
    },
    collections: {
      'practice.users': {
        shardKey: { userId: 'hashed' },
        unique: false,
```



```
        balancing: true,
        chunkMetadata: [ { shard: 'shard-replica-set', nChunks: 1 } ],
        chunks: [
          { min: { userId: MinKey() }, max: { userId: MaxKey() }, 'on shard': 'shard-replica-set', 'last modified': Timestamp({ t: 1,
 i: 0 }) }
        ],
        tags: []
      }
    }
  }
]
[direct: mongos] practice>
```