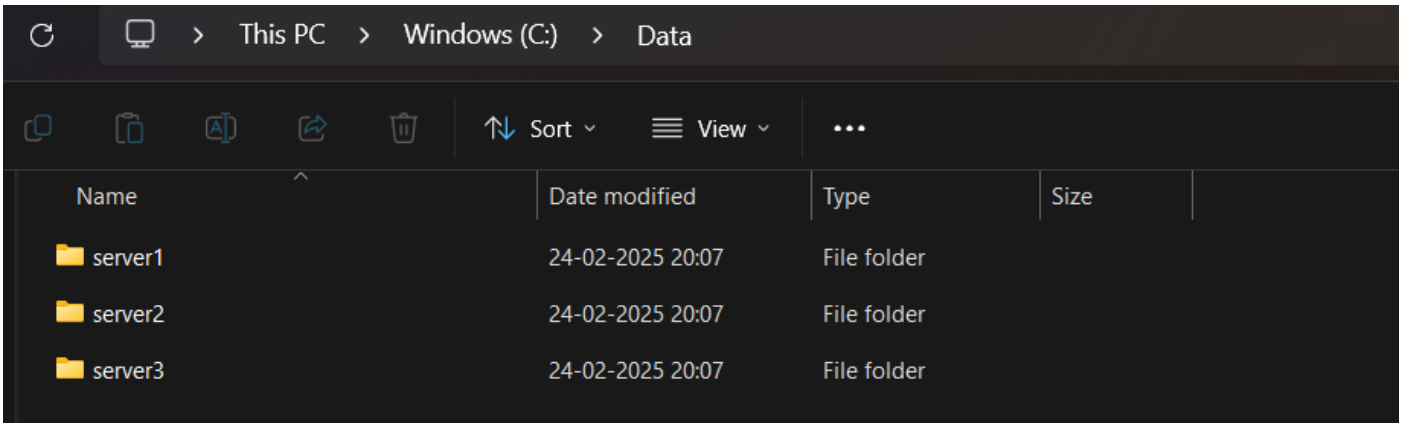
L039 Shruti Bamne

Practical No. 7

Sharding Using MongoDB

Step 1: Create “data” folder inside that create “server1”, “server2”, “server3”.



Step 2: Initialize MongoDB Config Servers with configsvr and replSet options to form a Replica Set of Config Servers.

Server 1

```
PS C:\Users\SHRUTI> mongod --configsvr --port=1030 --replSet="test-replica-set" --dbpath="C:\Data\server1"
{"t":{"$date":"2025-02-24T20:08:46.358+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-","msg":"Initialized wire
 specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClien
t":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true}
}}
{"t":{"$date":"2025-02-24T20:08:46.365+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":{"$date":"2025-02-24T20:08:46.910+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportL
ayer configured during NetworkInterface startup"}
{"t":{"$date":"2025-02-24T20:08:46.911+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP
FastOpen in use."}
```

Server 2

```
PS C:\Users\SHRUTI> mongod --configsvr --port=1040 --replSet="test-replica-set" --dbpath="C:\Data\server2"
{"t":{"$date":"2025-02-24T20:14:07.523+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"main","msg":"Initialized w
ire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalCl
ient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":tr
ue}}}
{"t":{"$date":"2025-02-24T20:14:07.525+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":{"$date":"2025-02-24T20:14:07.525+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportL
ayer configured during NetworkInterface startup"}
{"t":{"$date":"2025-02-24T20:14:07.525+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP
FastOpen in use."}
```

Server 3

```
PS C:\Users\SHRUTI> mongod --configsvr --port=1050 --replSet="test-replica-set" --dbpath="C:\Data\serve3"
{"t":{"$date":"2025-02-24T20:15:09.418+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"-","msg":"Automatically di
sabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-24T20:15:09.420+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-","msg":"Initialized wire
 specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClien
t":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true}
}}
{"t":{"$date":"2025-02-24T20:15:09.851+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportL
ayer configured during NetworkInterface startup"}
{"t":{"$date":"2025-02-24T20:15:09.851+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP
FastOpen in use."}
```

Step 3: Connect to anyone of them using mongosh and Initiate Replica Set.

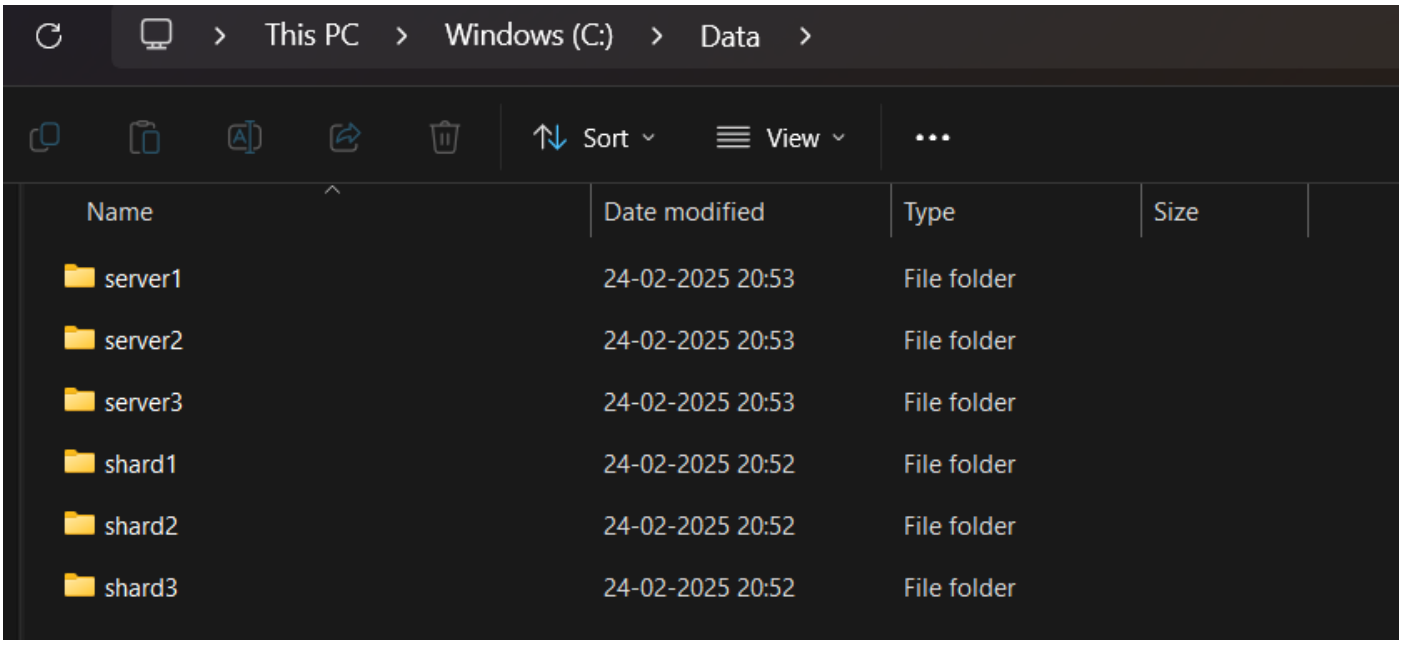
```
PS C:\Users\SHRUTI> mongosh --host="localhost:1030"
Current Mongosh Log ID: 67bc8c6dbec6e97763fa4213
Connecting to:          mongodb://localhost:1030/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.
4.0
Using MongoDB:          5.0.5
Using Mongosh:          2.4.0

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

------
   The server generated these startup warnings when booting
   2025-02-24T20:37:48.749+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
   2025-02-24T20:37:48.749+05:30: This server is bound to localhost. Remote systems will be unable to connect to this se
rver. 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
------
```

```
test> rs.initiate({
... _id:"test-replica-set",
... configsvr:true,
... members:[
... {_id:0, host:"localhost:1030"},
... {_id:1, host:"localhost:1040"},
... {_id:2, host:"localhost:1050"}
... ]
... })
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1740323667, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1740323667, i: 1 })
test-replica-set [direct: secondary] test>
```

Step 4: Create “shards” folder inside that create “shard1”, “shard2”, “shard3”.



Step 5:

Initialize MongoDB Shards

Shard 1

```
PS C:\Users\SHRUTI> mongod --shardsvr --port=1130 --dbpath="C:\Data\shard1" --replSet="shard-replica-set"
{"t":{"$date":"2025-02-24T20:54:43.580+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"main","msg":"Initialized w
ire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalCl
ient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":tr
ue}}}
{"t":{"$date":"2025-02-24T20:54:43.583+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":{"$date":"2025-02-24T20:54:43.584+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportL
ayer configured during NetworkInterface startup"}
{"t":{"$date":"2025-02-24T20:54:43.584+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP
FastOpen in use."}
```

Shard 2

```
PS C:\Users\SHRUTI> mongod --shardsvr --port=1140 --dbpath="C:\Data\shard2" --replSet="shard-replica-set"
{"t":{"$date":"2025-02-24T20:55:34.807+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-","msg":"Initialized wire
 specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClien
t":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true}
}}
{"t":{"$date":"2025-02-24T20:55:35.186+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":{"$date":"2025-02-24T20:55:35.189+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportL
ayer configured during NetworkInterface startup"}
{"t":{"$date":"2025-02-24T20:55:35.189+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP
FastOpen in use."}
```

Shard 3

```
PS C:\Users\SHRUTI> mongod --shardsvr --port=1150 --dbpath="C:\Data\shard3" --replSet="shard-replica-set"
{"t":{"$date":"2025-02-24T20:55:53.456+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"-","msg":"Automatically di
sabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-24T20:55:53.457+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-","msg":"Initialized wire
 specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClien
t":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true}
}}
{"t":{"$date":"2025-02-24T20:55:53.795+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportL
ayer configured during NetworkInterface startup"}
{"t":{"$date":"2025-02-24T20:55:53.796+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP
FastOpen in use."}
```

Connect using mongosh

```
PS C:\Users\SHRUTI> mongosh --host="localhost:1130"
Current Mongosh Log ID: 67bc90011f6d81db9ffa4213
Connecting to:          mongodb://localhost:1130/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.
4.0
Using MongoDB:          5.0.5
Using Mongosh:          2.4.0

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

------
   The server generated these startup warnings when booting
   2025-02-24T20:54:43.667+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
   2025-02-24T20:54:43.668+05:30: This server is bound to localhost. Remote systems will be unable to connect to this se
rver. 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
------
```

Initiate Replica Set

```
test> rs.initiate({
... _id:"shard-replica-set",
... members: [
... {_id: 0, host: "localhost:1130"},
... {_id: 1, host: "localhost:1140"},
... {_id: 2, host: "localhost:1150"}
... ]
... })
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1740323925, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1740323925, i: 1 })
}
shard-replica-set [direct: secondary] test>
```

Step 6:

Initialize a Query Router which is a mongos process.

```
PS C:\Users\SHRUTI> mongos --port=1210 --configdb="test-replica-set/localhost:1030,localhost:1040,localhost:1050"
{"t":{"$date":"2025-02-24T21:00:29.725+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-","msg":"Initialized wire
 specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClien
t":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":13,"maxWireVersion":13},"isInternalClient":true
}}}
{"t":{"$date":"2025-02-24T21:00:29.732+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"-","msg":"Automatically di
sabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-24T21:00:30.047+05:30"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"main","msg":"Access contro
l is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarn
ings"]}
{"t":{"$date":"2025-02-24T21:00:30.047+05:30"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"main","msg":"This server i
s bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address>
```

Now, Connect Shards and Query Router (mongos)

```
PS C:\Users\SHRUTI> mongosh --host="localhost:1210"
Current Mongosh Log ID: 67bc90c20ae4f5ab71fa4213
Connecting to:          mongodb://localhost:1210/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.
4.0
Using MongoDB:          5.0.5
Using Mongosh:          2.4.0

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

------
   The server generated these startup warnings when booting
   2025-02-24T21:00:30.047+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
   2025-02-24T21:00:30.047+05:30: This server is bound to localhost. Remote systems will be unable to connect to this se
rver. 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
------
```

```
[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 7:

Enable Sharding on a Specific Database of Shards Replica Set

```
[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 })
}
```

Shard a Collection on the Sharding Enabled Database

```
[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 })
}
```

Insert Sample Data and Verify

```
[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 8:

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>
```