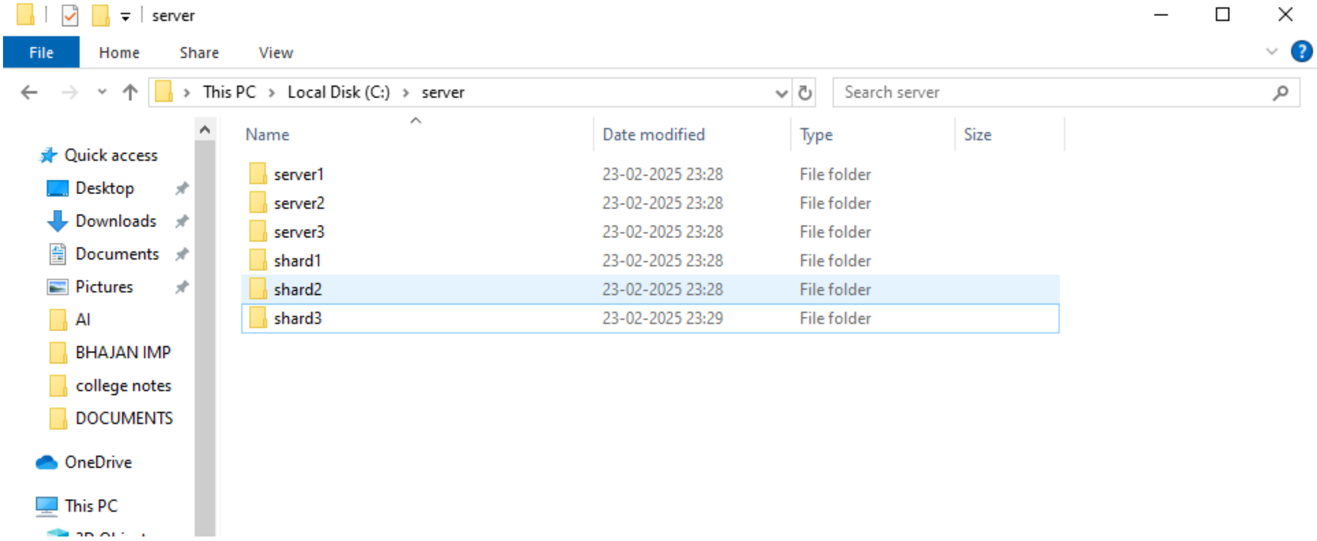
Practical 7 : Sharding in MongoDB

1. Understanding the Components

A MongoDB **sharded cluster** consists of:

- **Config Servers (CSRS)** – Store metadata about the cluster.
- **Shards** – Hold the actual data.
- **Query Routers (mongos)** – Route client queries to the appropriate shard



Name	Date modified	Type	Size
server1	23-02-2025 23:28	File folder	
server2	23-02-2025 23:28	File folder	
server3	23-02-2025 23:28	File folder	
shard1	23-02-2025 23:28	File folder	
shard2	23-02-2025 23:28	File folder	
shard3	23-02-2025 23:29	File folder	

2. Setting Up Config Servers

Config servers form a **Replica Set** that maintains cluster metadata.

Where to Perform:

Run each command in a separate terminal window.

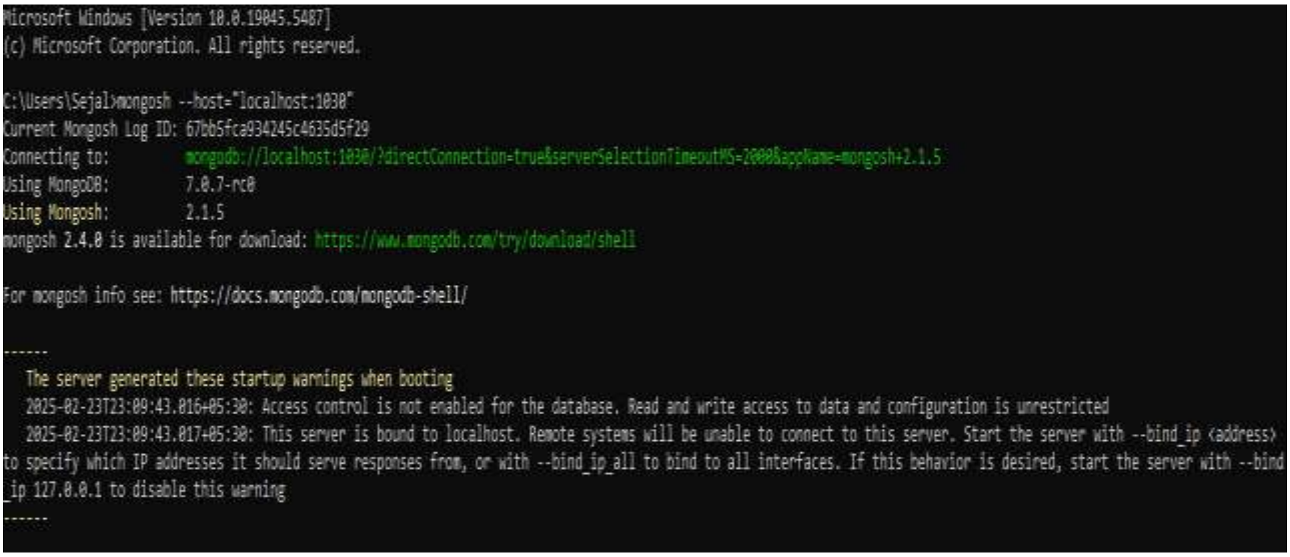
Start Config Servers

```
mongod --configsvr --port=1030 --replSet="configReplSet" --dbpath="/server/server1"

mongod --configsvr --port=1040 --replSet="configReplSet" --dbpath="/server/server2"

mongod --configsvr --port=1050 --replSet="configReplSet" --dbpath="/server/server3"
```

3. Initialize Config Server Replica Set

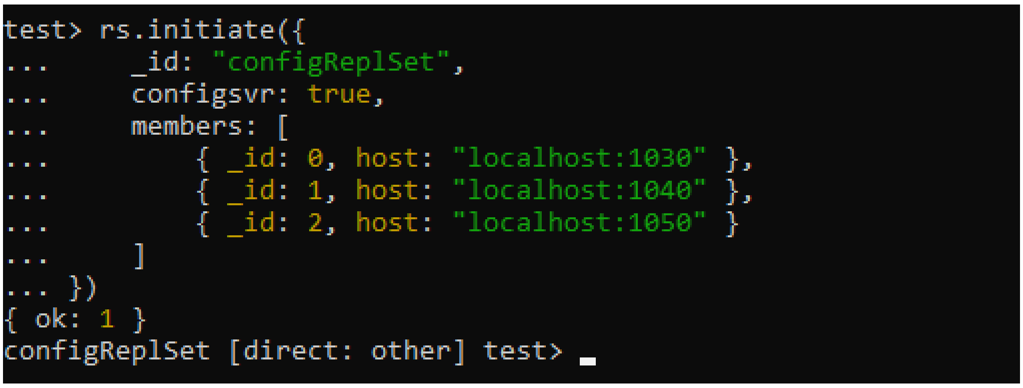
Connect to any one of the config servers using cmd:

Practical 7 : Sharding in MongoDB



```
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sejal>mongosh --host="localhost:1030"
Current Mongosh Log ID: 67bb5fca934245c4635d5f29
Connecting to:          mongodb://localhost:1030/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.5
Using MongoDB:          7.0.7-rc0
Using Mongosh:          2.1.5
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
   2025-02-23T23:09:43.016+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2025-02-23T23:09:43.017+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 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 the replica set



```
test> rs.initiate({
...     _id: "configReplSet",
...     configsvr: true,
...     members: [
...         { _id: 0, host: "localhost:1030" },
...         { _id: 1, host: "localhost:1040" },
...         { _id: 2, host: "localhost:1050" }
...     ]
... })
{ ok: 1 }
configReplSet [direct: other] test> _
```

4. Setting Up Shards

Shards store the actual data and are also configured as **Replica Sets**.

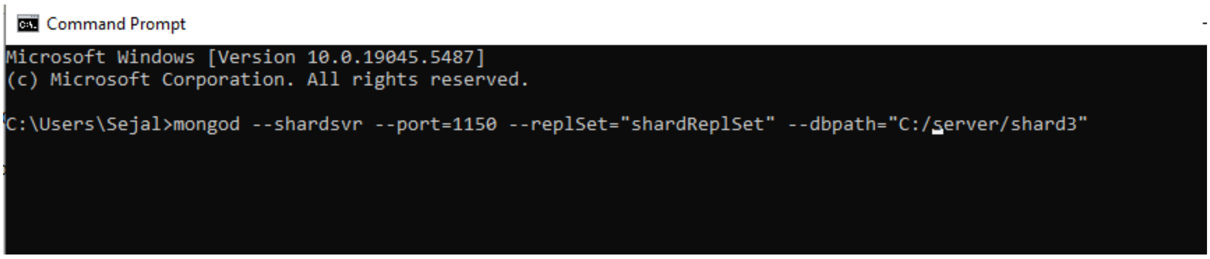
Where to Perform:

Run each command in a separate terminal window.

mongod --shardsvr --port=1130 --replSet="shardReplSet" --dbpath="/server/shard1"

mongod --shardsvr --port=1140 --replSet="shardReplSet" --dbpath="/server/shard2"

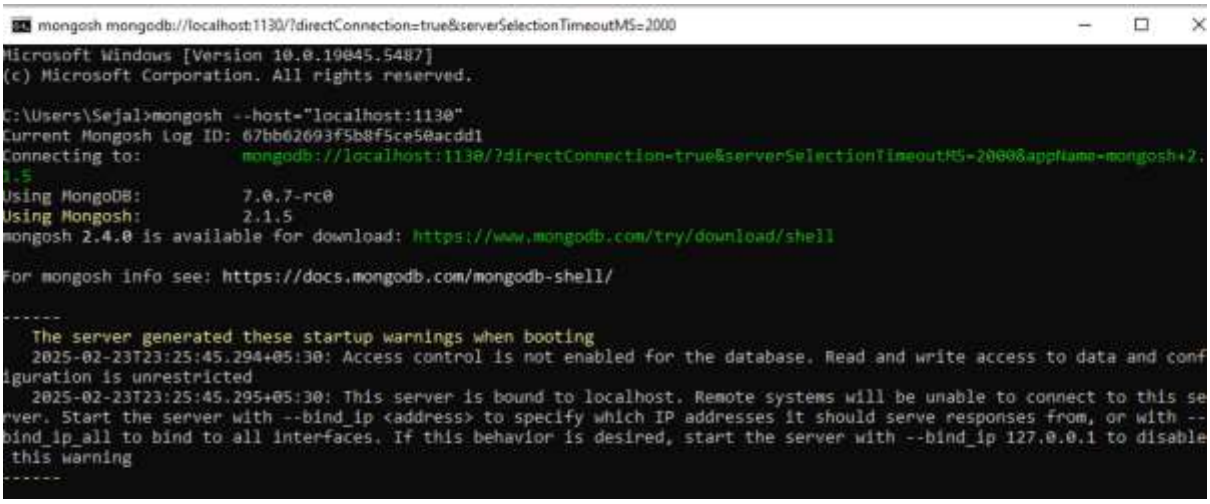
mongod --shardsvr --port=1150 --replSet="shardReplSet" --dbpath="/server/shard3"



```
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sejal>mongod --shardsvr --port=1150 --replSet="shardReplSet" --dbpath="C:/server/shard3"
```

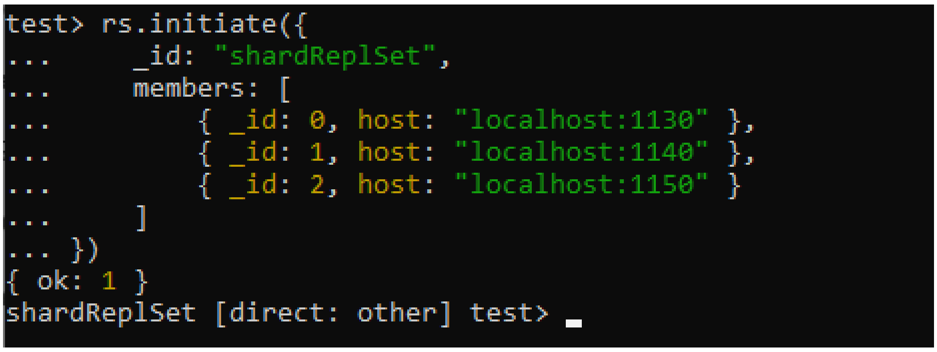
5. Initialize Shard Replica Set

Connect to any one of the shard servers using cmd



```
mongosh mongodb://localhost:1130/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sejal>mongosh --host="localhost:1130"
Current Mongosh Log ID: 67bb62693f5b8f5ce50acdd1
Connecting to:          mongodb://localhost:1130/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.
1.5
Using MongoDB:          7.0.7-rc0
Using Mongosh:          2.1.5
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
   2025-02-23T23:25:45.294+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
   2025-02-23T23:25:45.295+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
------
```

Inside cmd, initiate the replica set:



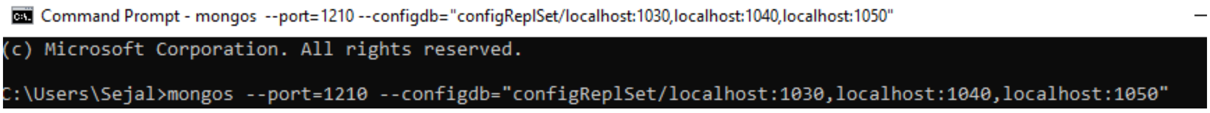
```
test> rs.initiate({
...     _id: "shardReplSet",
...     members: [
...         { _id: 0, host: "localhost:1130" },
...         { _id: 1, host: "localhost:1140" },
...         { _id: 2, host: "localhost:1150" }
...     ]
... })
{ ok: 1 }
shardReplSet [direct: other] test> _
```

6. Setting Up Query Router (mongos)

The **mongos** process acts as a router that distributes queries across shards.

Where to Perform:

Run in a separate terminal.

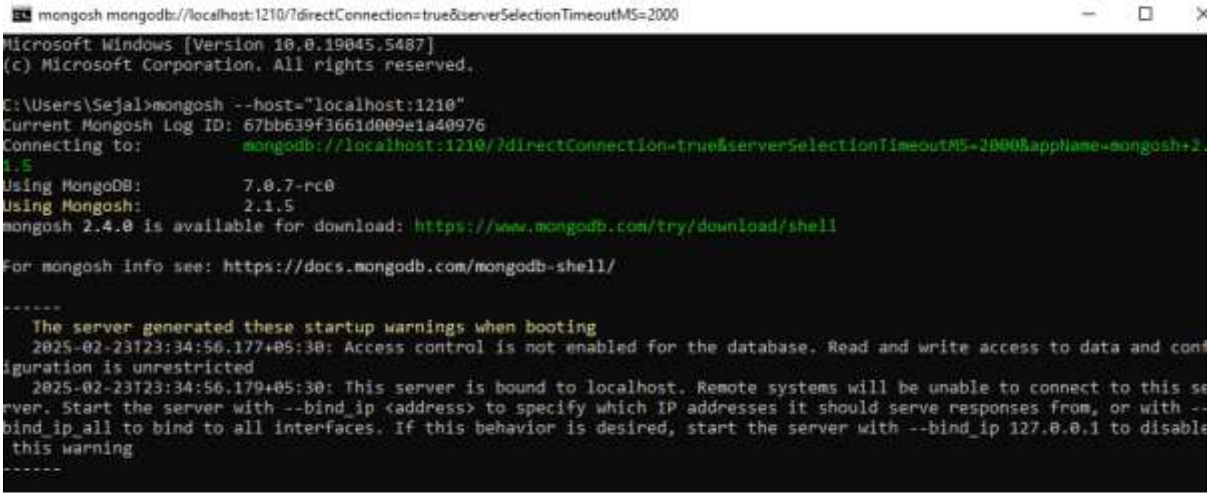


```
Command Prompt - mongos  --port=1210 --configdb="configReplSet/localhost:1030,localhost:1040,localhost:1050"
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sejal>mongos --port=1210 --configdb="configReplSet/localhost:1030,localhost:1040,localhost:1050"
```

Practical 7 : Sharding in MongoDB

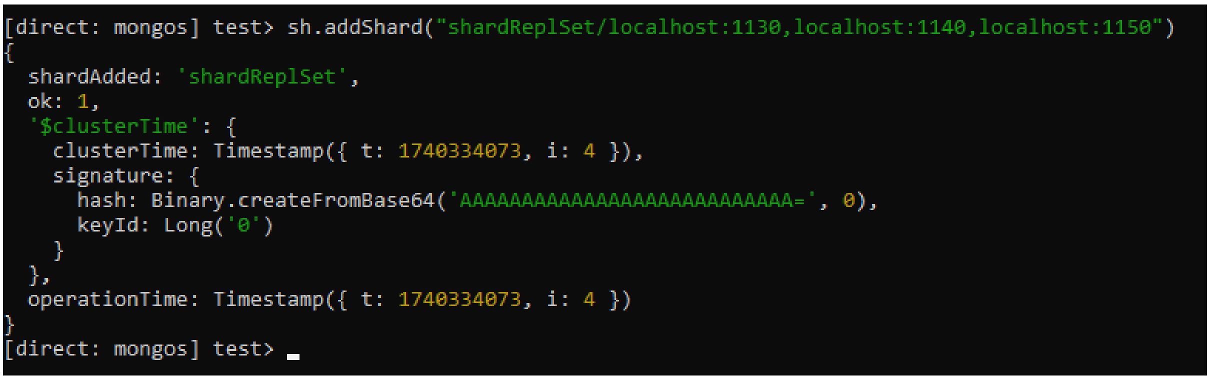
7. Connecting Shards to the Query Router

Connect to the query router using cmd:



```
mongosh mongodb://localhost:1210/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sejal>mongosh --host="localhost:1210"
Current Mongosh Log ID: 67bb639f3661d009e1a40976
Connecting to:          mongodb://localhost:1210/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.
1.5
Using MongoDB:          7.0.7-rc0
Using Mongosh:          2.1.5
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
   2025-02-23T23:34:56.177+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
   2025-02-23T23:34:56.179+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
------
```

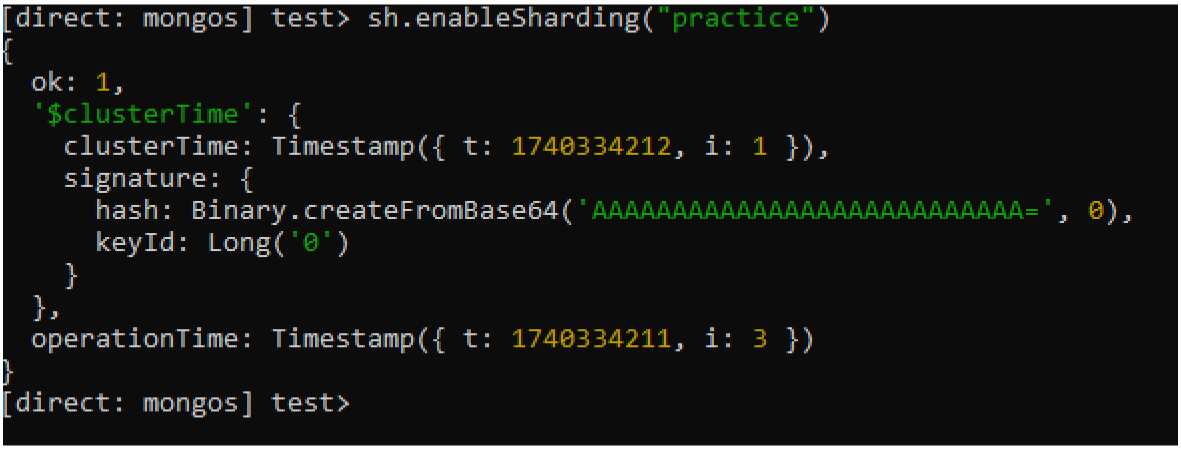
Inside cmd, add the shard replica set:



```
[direct: mongos] test> sh.addShard("shardReplSet/localhost:1130,localhost:1140,localhost:1150")
{
  shardAdded: 'shardReplSet',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1740334073, i: 4 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1740334073, i: 4 })
}
[direct: mongos] test> _
```

8. Enabling Sharding for a Database

Enable sharding for a specific database:

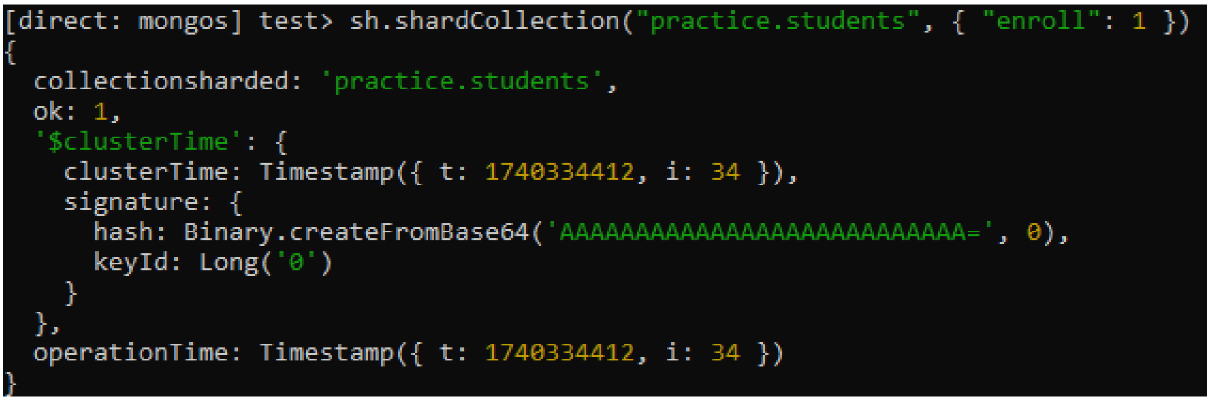


```
[direct: mongos] test> sh.enableSharding("practice")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1740334212, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1740334211, i: 3 })
}
[direct: mongos] test>
```

Practical 7 : Sharding in MongoDB

9. Sharding a Collection

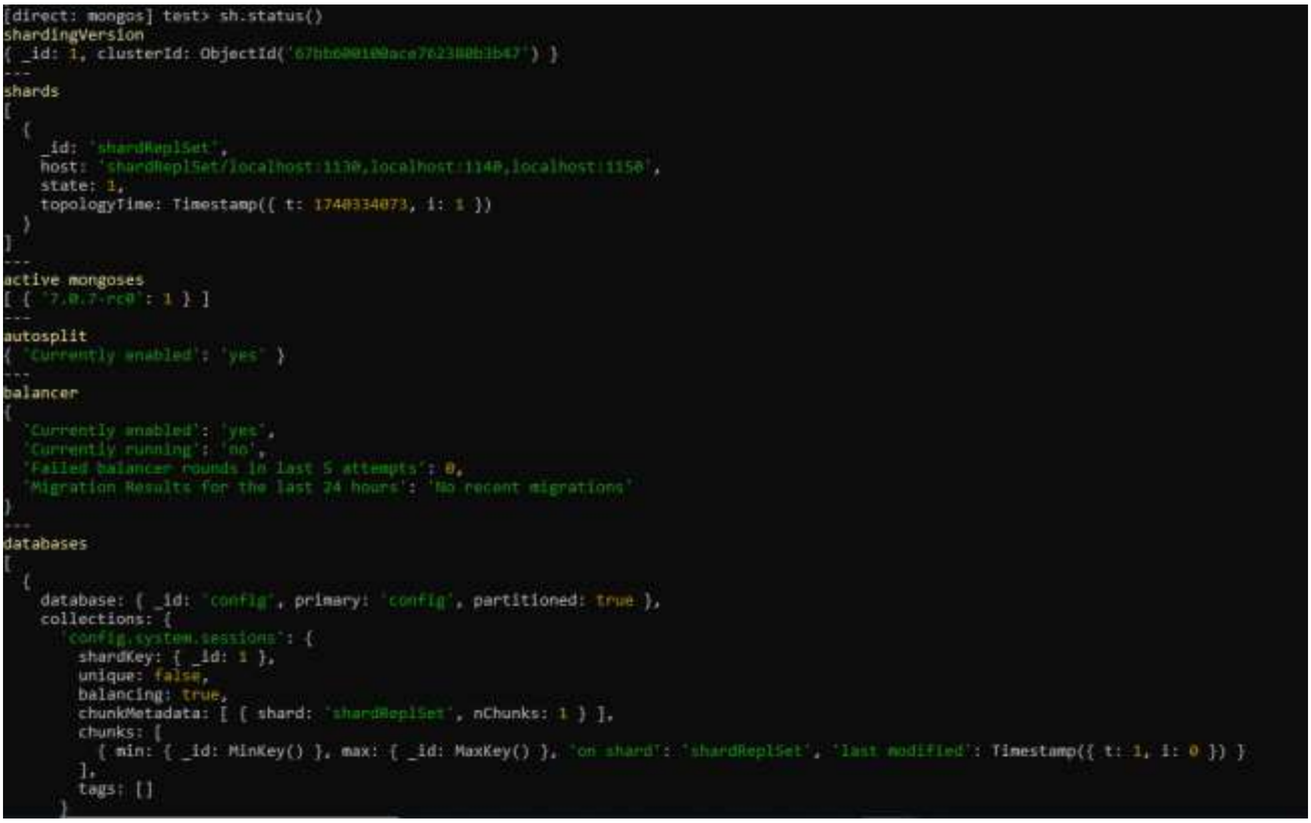
Shard the students collection using enroll as the shard key:



```
[direct: mongos] test> sh.shardCollection("practice.students", { "enroll": 1 })
{
  collectionsharded: 'practice.students',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1740334412, i: 34 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1740334412, i: 34 })
}
```

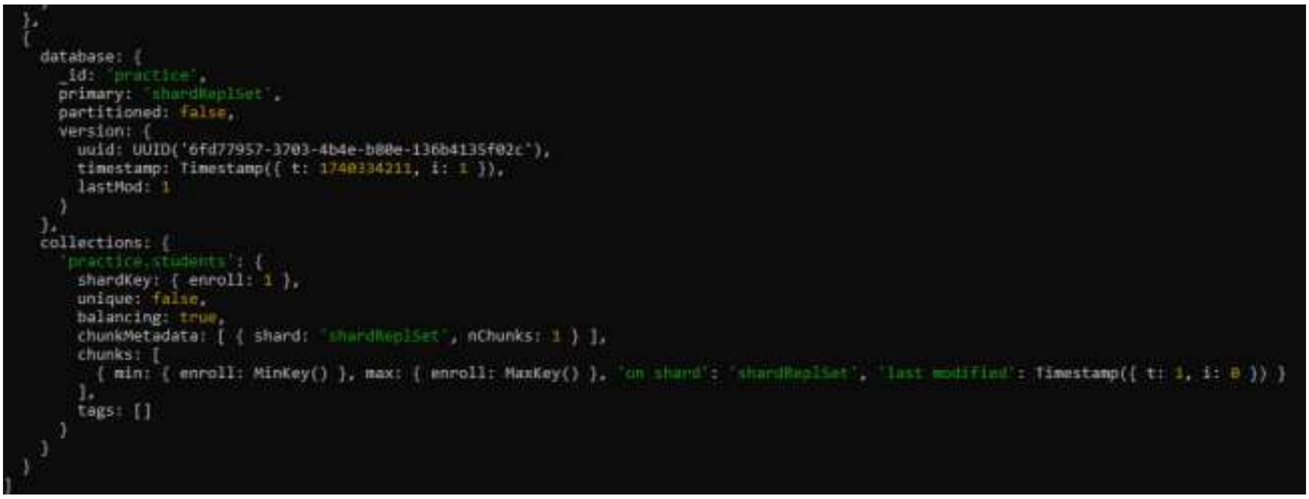
10. Testing the Sharded Cluster

To verify everything is working:



```
[direct: mongos] test> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('67bb600100ace7623b0b3b47') }
---
shards
[
  {
    _id: 'shardReplSet',
    host: 'shardReplSet/localhost:1130,localhost:1140,localhost:1150',
    state: 1,
    topologyTime: Timestamp({ t: 1740334073, i: 1 })
  }
]
---
active mongoses
[ { '7.0.7-rc0': 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'
}
---
databases
[
  {
    database: { _id: 'config', primary: 'config', partitioned: true },
    collections: {
      'config.system.sessions': {
        shardKey: { _id: 1 },
        unique: false,
        balancing: true,
        chunkMetadata: [ { shard: 'shardReplSet', nChunks: 1 } ],
        chunks: [
          { min: { _id: MinKey() }, max: { _id: MaxKey() }, 'on shard': 'shardReplSet', 'last modified': Timestamp({ t: 1, i: 0 }) }
        ],
        tags: []
      }
```

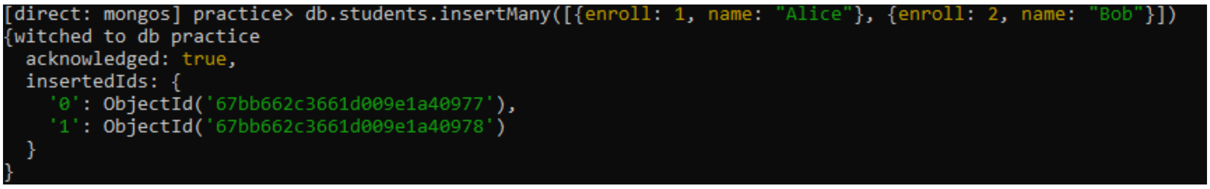
Practical 7 : Sharding in MongoDB



```
  },
  {
    database: {
      _id: 'practice',
      primary: 'shardReplSet',
      partitioned: false,
      version: {
        uuid: UUID('6fd77957-3703-4b4e-b80e-136b4135f02c'),
        timestamp: Timestamp({ t: 1740334211, i: 1 }),
        lastMod: 1
      }
    },
    collections: {
      'practice.students': {
        shardKey: { enroll: 1 },
        unique: false,
        balancing: true,
        chunkMetadata: [ { shard: 'shardReplSet', nChunks: 1 } ],
        chunks: [
          { min: { enroll: MinKey() }, max: { enroll: MaxKey() }, 'on shard': 'shardReplSet', 'last modified': Timestamp({ t: 1, i: 0 }) }
        ],
        tags: []
      }
    }
  }
]
```

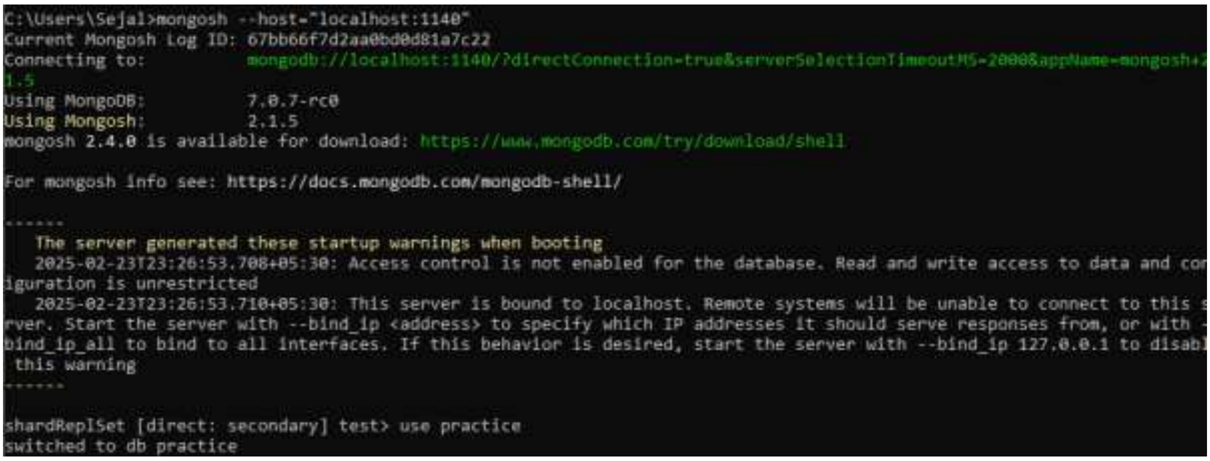
This will show details about shards, config servers, and sharded databases.

Insert data and check distribution:

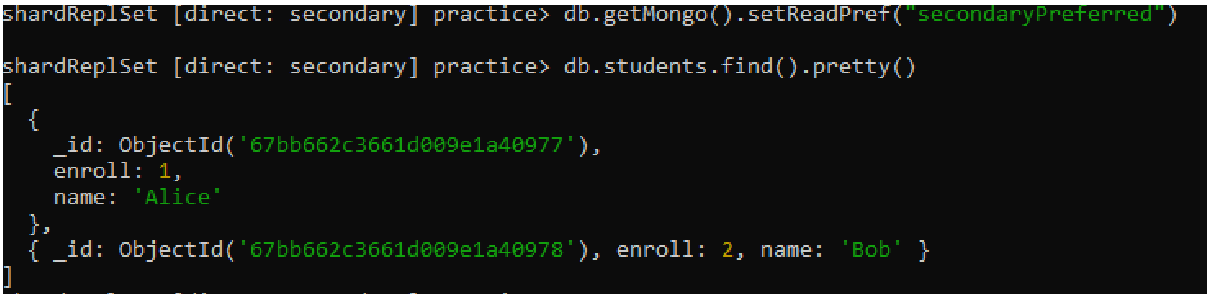


```
[direct: mongos] practice> db.students.insertMany([{enroll: 1, name: "Alice"}, {enroll: 2, name: "Bob"}])
{witched to db practice
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67bb662c3661d009e1a40977'),
    '1': ObjectId('67bb662c3661d009e1a40978')
  }
}
```

11. To confirm data exists on Shard2 (localhost:1140) and Shard3 (localhost:1150):



```
C:\Users\Sejal>mongosh --host="localhost:1140"
Current Mongosh Log ID: 67bb66f7d2aa0bd0d81a7c22
Connecting to:          mongodb://localhost:1140/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
1.5
Using MongoDB:          7.0.7-rc0
Using Mongosh:          2.1.5
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
   2025-02-23T23:26:53.708+05:30: Access control is not enabled for the database. Read and write access to data and con
iguration is unrestricted
   2025-02-23T23:26:53.710+05:30: This server is bound to localhost. Remote systems will be unable to connect to this s
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 disabl
 this warning
------

shardReplSet [direct: secondary] test> use practice
switched to db practice
```



```
shardReplSet [direct: secondary] practice> db.getMongo().setReadPref("secondaryPreferred")

shardReplSet [direct: secondary] practice> db.students.find().pretty()
[
  {
    _id: ObjectId('67bb662c3661d009e1a40977'),
    enroll: 1,
    name: 'Alice'
  },
  { _id: ObjectId('67bb662c3661d009e1a40978'), enroll: 2, name: 'Bob' }
]
```

Practical 7 : Sharding in MongoDB