MONGODB

Sharding Concept

Make sure u have MongoDB shell installed

Add Environment variable -> C:\Program Files\MongoDB\Server\8.0\bin

Create data folder inside C drive

Inside that create

Open PowerShell -> Run as Administrator

Config Server

```
mkdir C:\data\cfg1, C:\data\cfg2, C:\data\cfg3
```

Shard Servers (2 shards, 3 nodes each)

```
mkdir C:\data\shard1_1, C:\data\shard1_2, C:\data\shard1_3
```

```
mkdir C:\data\shard2_1, C:\data\shard2_2, C:\data\shard2_3
```

Mongos router

```
mkdir C:\data\mongos
```

Open 3 different powershells run commands -> configure servers

```
mongod --configsvr --replSet configReplSet --port 27019 --dbpath C:\data\cfg1 --bind_ip localhost
```

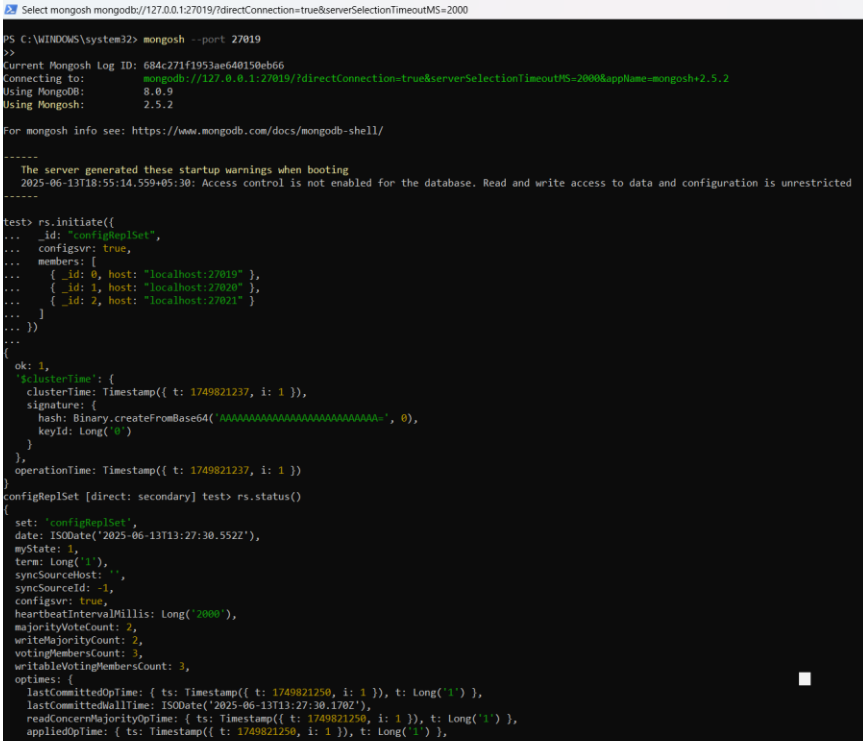
```
mongod --configsvr --replSet configReplSet --port 27020 --dbpath C:\data\cfg2 --bind_ip localhost
```

```
mongod --configsvr --replSet configReplSet --port 27021 --dbpath C:\data\cfg3 --bind_ip localhost
```

```
mongosh --port 27019
```

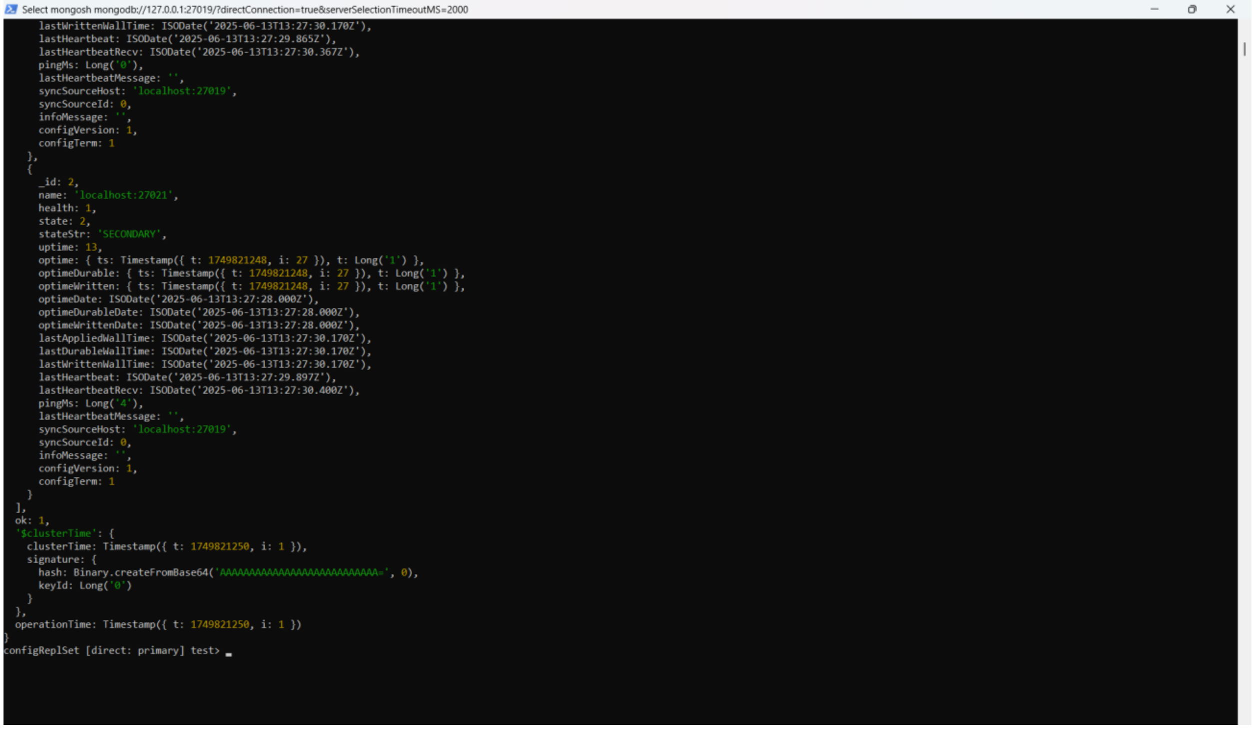
```
rs.initiate({
  _id: "configReplSet",
  configsvr: true,
  members: [
    { _id: 0, host: "localhost:27019" },
    { _id: 1, host: "localhost:27020" },
    { _id: 2, host: "localhost:27021" }
  ]
})
rs.status()
```

Output:



Select mongosh mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000

```
PS C:\WINDOWS\system32> mongosh --port 27019
>>
Current Mongosh Log ID: 684c271f1953ae640150eb66
Connecting to:          mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:          8.0.9
Using Mongosh:          2.5.2

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

------
   The server generated these startup warnings when booting
   2025-06-13T18:55:14.559+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> rs.initiate({
...   _id: "configReplSet",
...   configsvr: true,
...   members: [
...     { _id: 0, host: "localhost:27019" },
...     { _id: 1, host: "localhost:27020" },
...     { _id: 2, host: "localhost:27021" }
...   ]
... })
...
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1749821237, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1749821237, i: 1 })
}
configReplSet [direct: secondary] test> rs.status()
{
  set: 'configReplSet',
  date: ISODate('2025-06-13T13:27:30.552Z'),
  myState: 1,
  term: Long('1'),
  syncSourceHost: '',
  syncSourceId: -1,
  configsvr: true,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1749821250, i: 1 }), t: Long('1') },
    lastCommittedWallTime: ISODate('2025-06-13T13:27:30.170Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1749821250, i: 1 }), t: Long('1') },
    appliedOpTime: { ts: Timestamp({ t: 1749821250, i: 1 }), t: Long('1') },
```


Select mongosh mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000

```
      lastWrittenWallTime: ISODate('2025-06-13T13:27:30.170Z'),
      lastHeartbeat: ISODate('2025-06-13T13:27:29.865Z'),
      lastHeartbeatRecv: ISODate('2025-06-13T13:27:30.367Z'),
      pingMs: Long('0'),
      lastHeartbeatMessage: '',
      syncSourceHost: 'localhost:27019',
      syncSourceId: 0,
      infoMessage: '',
      configVersion: 1,
      configTerm: 1
    },
    {
      _id: 2,
      name: 'localhost:27021',
      health: 1,
      state: 2,
      stateStr: 'SECONDARY',
      uptime: 13,
      optime: { ts: Timestamp({ t: 1749821248, i: 27 }), t: Long('1') },
      optimeDurable: { ts: Timestamp({ t: 1749821248, i: 27 }), t: Long('1') },
      optimeWritten: { ts: Timestamp({ t: 1749821248, i: 27 }), t: Long('1') },
      optimeDate: ISODate('2025-06-13T13:27:28.000Z'),
      optimeDurableDate: ISODate('2025-06-13T13:27:28.000Z'),
      optimeWrittenDate: ISODate('2025-06-13T13:27:28.000Z'),
      lastAppliedWallTime: ISODate('2025-06-13T13:27:30.170Z'),
      lastDurableWallTime: ISODate('2025-06-13T13:27:30.170Z'),
      lastWrittenWallTime: ISODate('2025-06-13T13:27:30.170Z'),
      lastHeartbeat: ISODate('2025-06-13T13:27:29.897Z'),
      lastHeartbeatRecv: ISODate('2025-06-13T13:27:30.400Z'),
      pingMs: Long('4'),
      lastHeartbeatMessage: '',
      syncSourceHost: 'localhost:27019',
      syncSourceId: 0,
      infoMessage: '',
      configVersion: 1,
      configTerm: 1
    }
  ],
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1749821250, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1749821250, i: 1 })
}
configReplSet [direct: primary] test>
```

Start Shard 1 Replica Set (3 nodes)

open 3 new PowerShell windows and run

mongod --shardsvr --replSet shard1ReplSet --port 27022 --dbpath C:\data\shard1_1 --bind_ip localhost

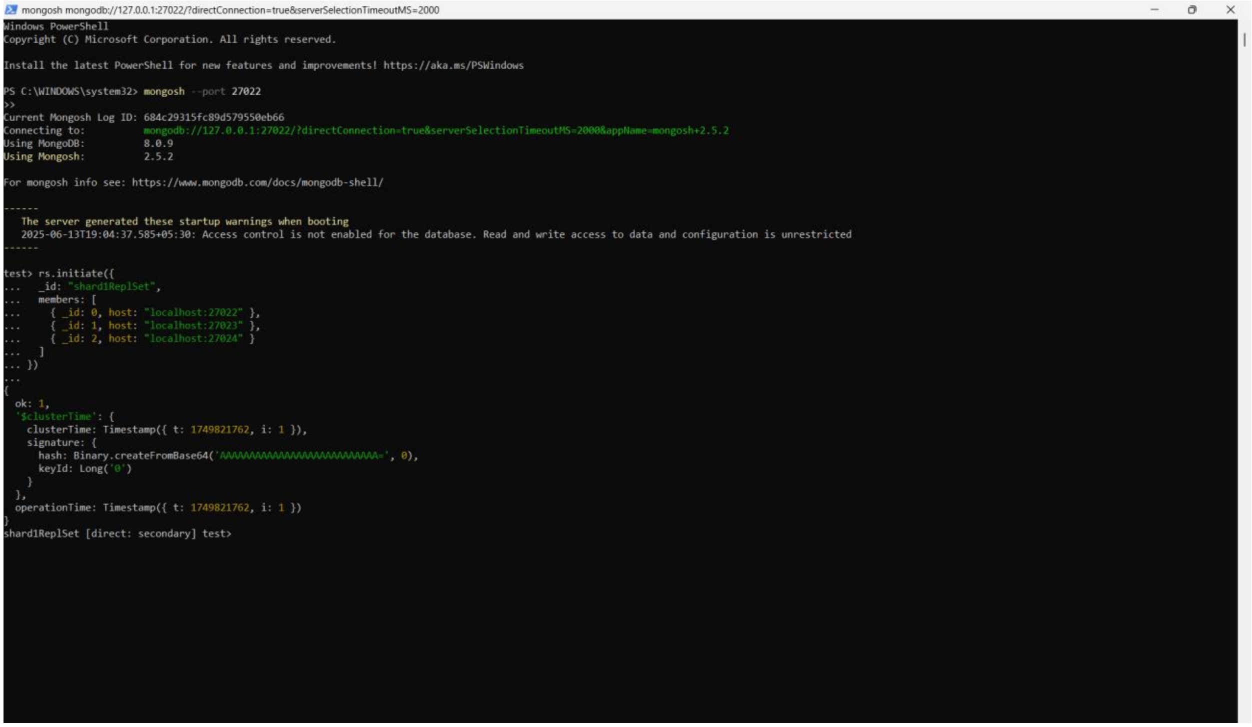
mongod --shardsvr --replSet shard1ReplSet --port 27023 --dbpath C:\data\shard1_2 --bind_ip localhost

mongod --shardsvr --replSet shard1ReplSet --port 27024 --dbpath C:\data\shard1_3 --bind_ip localhost

mongosh --port 27022

```
rs.initiate({
  _id: "shard1ReplSet",
  members: [
    { _id: 0, host: "localhost:27022" },
    { _id: 1, host: "localhost:27023" },
    { _id: 2, host: "localhost:27024" }
  ]
})
```

Output:



mongosh mongodb://127.0.0.1:27022/?directConnection=true&serverSelectionTimeoutMS=2000

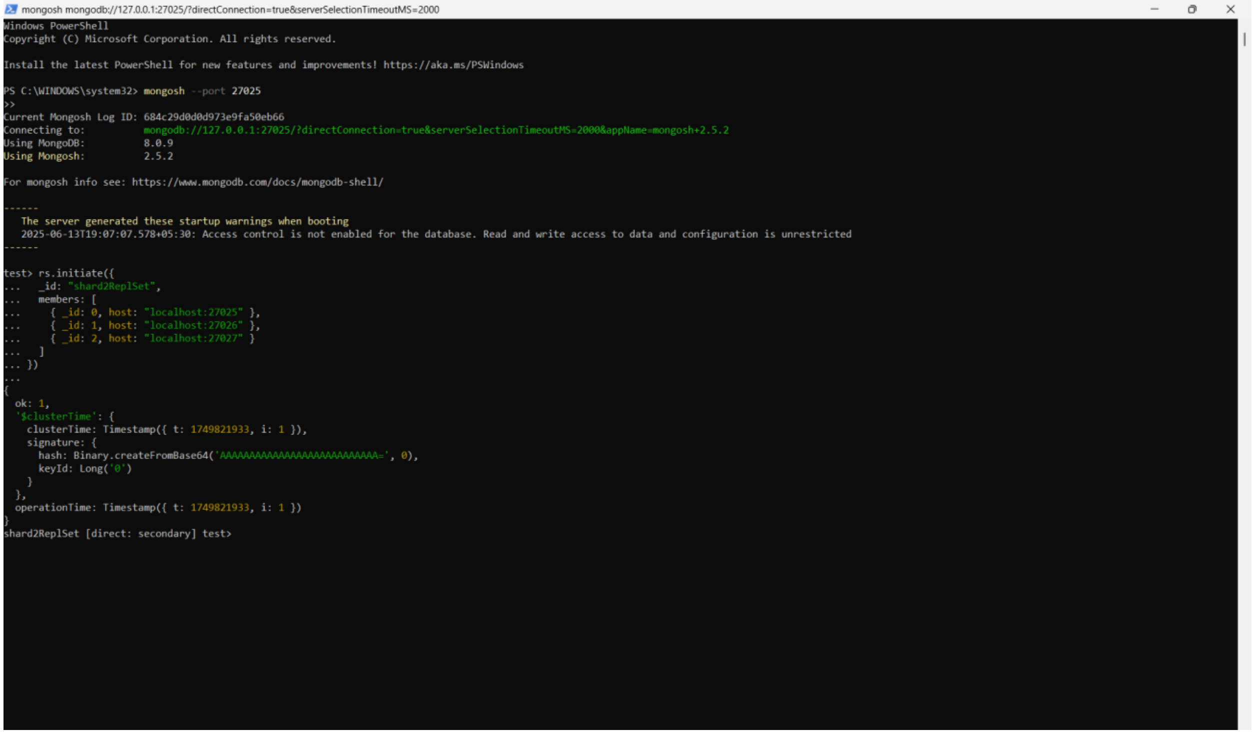
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> mongosh --port 27022
>>
Current Mongosh Log ID: 684c29315fc89d579550eb66
Connecting to:          mongodb://127.0.0.1:27022/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:          8.0.9
Using Mongosh:          2.5.2

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

------
   The server generated these startup warnings when booting
   2025-06-13T19:04:37.585+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> rs.initiate({
...   _id: "shard1ReplSet",
...   members: [
...     { _id: 0, host: "localhost:27022" },
...     { _id: 1, host: "localhost:27023" },
...     { _id: 2, host: "localhost:27024" }
...   ]
... })
...
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1749821762, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1749821762, i: 1 })
}
shard1ReplSet [direct: secondary] test>
```

Start Shard 2 Replica Set (3 nodes)

Open 3 new PowerShell windows and run

```
mongod --shardsvr --replSet shard2ReplSet --port 27025 --dbpath C:\data\shard2_1 --bind_ip localhost

mongod --shardsvr --replSet shard2ReplSet --port 27026 --dbpath C:\data\shard2_2 --bind_ip localhost

mongod --shardsvr --replSet shard2ReplSet --port 27027 --dbpath C:\data\shard2_3 --bind_ip localhost

mongosh --port 27025

rs.initiate({
  _id: "shard2ReplSet",
  members: [
    { _id: 0, host: "localhost:27025" },
    { _id: 1, host: "localhost:27026" },
    { _id: 2, host: "localhost:27027" }
  ]
})
```

Output:



mongosh mongodb://127.0.0.1:27025/?directConnection=true&serverSelectionTimeoutMS=2000

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> mongosh --port 27025
>>
Current Mongosh Log ID: 684c29d0d0d973e9fa50eb66
Connecting to:          mongodb://127.0.0.1:27025/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:          8.0.9
Using Mongosh:          2.5.2

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

------
   The server generated these startup warnings when booting
   2025-06-13T19:07:07.578+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> rs.initiate({
...   _id: "shard2ReplSet",
...   members: [
...     { _id: 0, host: "localhost:27025" },
...     { _id: 1, host: "localhost:27026" },
...     { _id: 2, host: "localhost:27027" }
...   ]
... })
...
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1749821933, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1749821933, i: 1 })
}
shard2ReplSet [direct: secondary] test>
```

Open a new PowerShell window

mongos --configdb configReplSet/localhost:27019,localhost:27020,localhost:27021 --port 27018 --bind_ip localhost

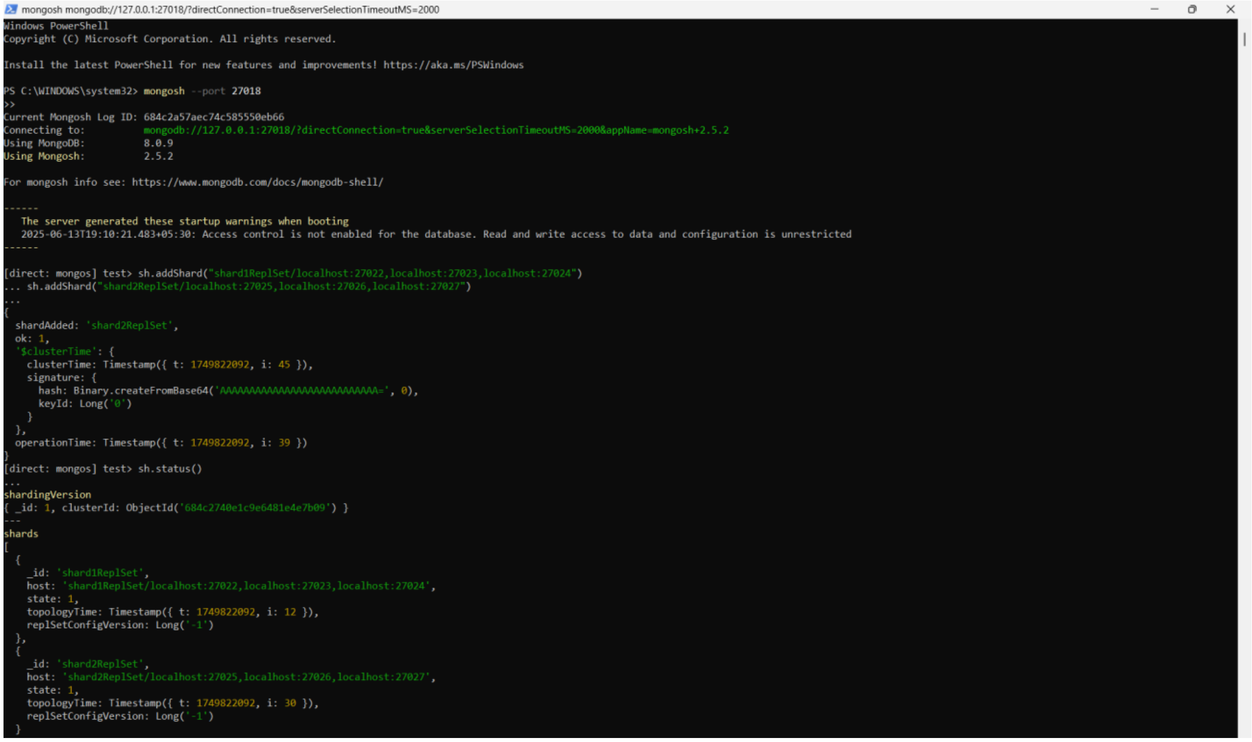
mongosh --port 27018

sh.addShard("shard1ReplSet/localhost:27022,localhost:27023,localhost:27024")

sh.addShard("shard2ReplSet/localhost:27025,localhost:27026,localhost:27027")

sh.status()

Output:

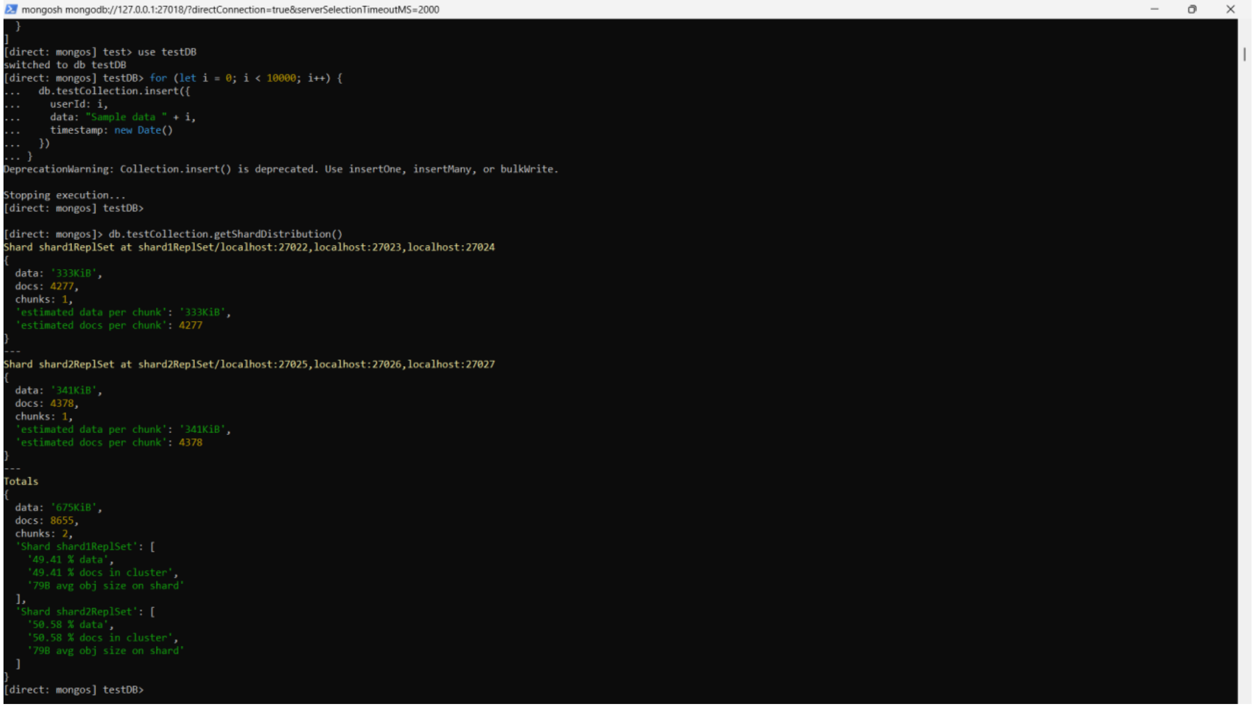


mongosh mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> mongosh --port 27018
>>
Current Mongosh Log ID: 684c2a57aec74c585550eb66
Connecting to:          mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:          8.0.9
Using Mongosh:          2.5.2

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

------
   The server generated these startup warnings when booting
   2025-06-13T19:10:21.483+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

[direct: mongos] test> sh.addShard("shard1ReplSet/localhost:27022,localhost:27023,localhost:27024")
... sh.addShard("shard2ReplSet/localhost:27025,localhost:27026,localhost:27027")
...
{
  shardAdded: 'shard2ReplSet',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1749822092, i: 45 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1749822092, i: 39 })
}
[direct: mongos] test> sh.status()
...
shardingVersion
{ _id: 1, clusterId: ObjectId('684c2740e1c9e6481e4e7b09') }
---
shards
[
  {
    _id: 'shard1ReplSet',
    host: 'shard1ReplSet/localhost:27022,localhost:27023,localhost:27024',
    state: 1,
    topologyTime: Timestamp({ t: 1749822092, i: 12 }),
    replSetConfigVersion: Long('-1')
  },
  {
    _id: 'shard2ReplSet',
    host: 'shard2ReplSet/localhost:27025,localhost:27026,localhost:27027',
    state: 1,
    topologyTime: Timestamp({ t: 1749822092, i: 30 }),
    replSetConfigVersion: Long('-1')
  }
```

```
sh.enableSharding("testDB")
sh.shardCollection("testDB.testCollection", { "_id": "hashed" })
for (let i = 0; i < 1000; i++) {
  db.testCollection.insert({
    userId: i,
    text: "hello " + i,
    time: new Date()
  })
}
db.testCollection.getShardDistribution()
```

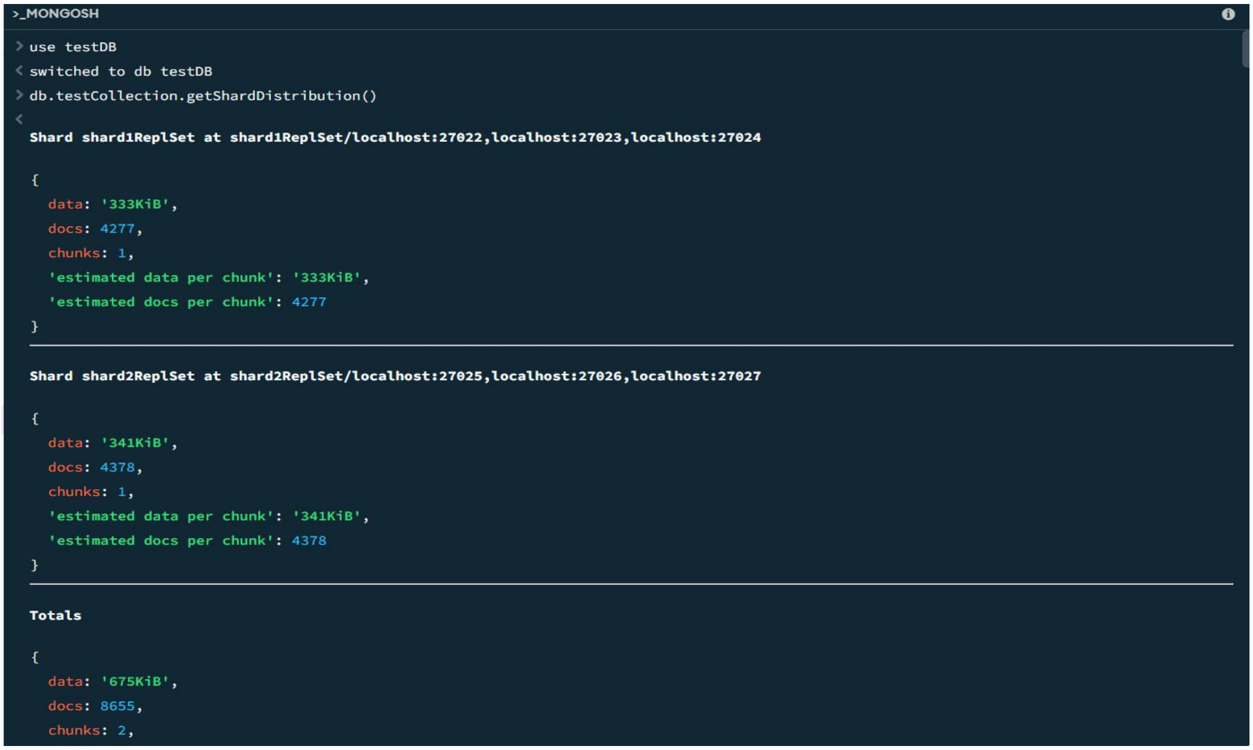
Output:


mongosh mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000

```
  }
]
[direct: mongos] test> use testDB
switched to db testDB
[direct: mongos] testDB> for (let i = 0; i < 10000; i++) {
...   db.testCollection.insert({
...     userId: i,
...     data: "Sample data " + i,
...     timestamp: new Date()
...   })
... }
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

Stopping execution...
[direct: mongos] testDB>

[direct: mongos]> db.testCollection.getShardDistribution()
Shard shard1ReplSet at shard1ReplSet/localhost:27022,localhost:27023,localhost:27024
{
  data: '333KiB',
  docs: 4277,
  chunks: 1,
  'estimated data per chunk': '333KiB',
  'estimated docs per chunk': 4277
}
---
Shard shard2ReplSet at shard2ReplSet/localhost:27025,localhost:27026,localhost:27027
{
  data: '341KiB',
  docs: 4378,
  chunks: 1,
  'estimated data per chunk': '341KiB',
  'estimated docs per chunk': 4378
}
---
Totals
{
  data: '675KiB',
  docs: 8655,
  chunks: 2,
  'Shard shard1ReplSet': [
    '49.41 % data',
    '49.41 % docs in cluster',
    '79B avg obj size on shard'
  ],
  'Shard shard2ReplSet': [
    '50.58 % data',
    '50.58 % docs in cluster',
    '79B avg obj size on shard'
  ]
}
[direct: mongos] testDB>
```

In MongoDB:

Create a Connection using localhost:27018



```
> use testDB
< switched to db testDB
> db.testCollection.getShardDistribution()
<
  Shard shard1ReplSet at shard1ReplSet/localhost:27022,localhost:27023,localhost:27024

  {
    data: '333KiB',
    docs: 4277,
    chunks: 1,
    'estimated data per chunk': '333KiB',
    'estimated docs per chunk': 4277
  }

  Shard shard2ReplSet at shard2ReplSet/localhost:27025,localhost:27026,localhost:27027

  {
    data: '341KiB',
    docs: 4378,
    chunks: 1,
    'estimated data per chunk': '341KiB',
    'estimated docs per chunk': 4378
  }

  Totals

  {
    data: '675KiB',
    docs: 8655,
    chunks: 2,
```