C:\Windows\System32>mongosh --port 27034

# Connecting to Shard1 primary node

Current Mongosh Log ID: 684c2c3ce6a869a7b750eb66

Connecting to: mongodb://127.0.0.1:27022/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2

Using MongoDB: 8.0.4

Using Mongosh: 2.5.2

------

The server generated these startup warnings when booting

2025-06-13T19:16:28.278+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"}]})

# Initializing Shard1 replica set

{

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749822772, i: 1 }),

signature: {

hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749822772, i: 1 })

}

C:\Windows\System32>mongosh --port 27025

# Connecting to Shard2 primary node

Current Mongosh Log ID: 684c2fc649d321d7bf50eb66

Connecting to: mongodb://127.0.0.1:27025/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2

Using MongoDB: 8.0.4

Using Mongosh: 2.5.2

------

The server generated these startup warnings when booting

2025-06-13T19:32:08.286+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"}]})

# Initializing Shard2 replica set

{

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749823576, i: 1 }),

signature: {

hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749823576, i: 1 })

}

C:\Windows\System32>mongosh --port 27018

# Connecting to mongos router

Current Mongosh Log ID: 684c30a68af88ca4b350eb66

Connecting to: mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2

Using MongoDB: 8.0.4

Using Mongosh: 2.5.2

------

The server generated these startup warnings when booting

2025-06-13T19:37:07.027+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")

# Adding Shard1 to cluster

{

shardAdded: 'shard1ReplSet',

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749823728, i: 20 }),

signature: {

hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749823728, i: 20 })

}

[direct: mongos] test> sh.addShard("shard2ReplSet/localhost:27025,localhost:27026,localhost:27027")

# Adding Shard2 to cluster

{

shardAdded: 'shard2ReplSet',

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749823794, i: 25 }),

signature: {

hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749823794, i: 19 })

}

[direct: mongos] test> sh.enableSharding("testDB")

# Enabling sharding on testDB

{

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749823859, i: 9 }),

signature: {

hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749823859, i: 6 })

}

[direct: mongos] test> sh.shardCollection("testDB.testCollection",{"\_id":"hashed"})

# Sharding collection testCollection on \_id using hashed sharding

{

collectionsharded: 'testDB.testCollection',

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749823930, i: 45 }),

signature: {

hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749823930, i: 45 })

}

[direct: mongos] test> use testDB

# Switching to testDB

switched to db testDB

[direct: mongos] testDB> db.testCollection.getShardDistribution()

# Checking shard distribution (before inserting data)

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

{

data: '0B',

docs: 0,

chunks: 1,

'estimated data per chunk': '0B',

'estimated docs per chunk': 0

}

---

Shard shard1ReplSet at shard1ReplSet/localhost:27022,localhost:27023,localhost:27024

{

data: '0B',

docs: 0,

chunks: 1,

'estimated data per chunk': '0B',

'estimated docs per chunk': 0

}

---

Totals

{

data: '0B',

docs: 0,

chunks: 2,

'Shard shard2ReplSet': [ '0 % data', '0 % docs in cluster', '0B avg obj size on shard' ],

'Shard shard1ReplSet': [ '0 % data', '0 % docs in cluster', '0B avg obj size on shard' ]

}

[direct: mongos] testDB> for(let i=0;i<10000;i++){ db.testCollection.insert({ userId:i, data:"Sample data"+ i, timestamp:new Date() }) }

# Inserting 10,000 documents into testCollection

{

acknowledged: true,

insertedIds: { '0': ObjectId('684c363e8af88ca4b3511276') }

}

[direct: mongos] testDB> db.testCollection.getShardDistribution()

# Checking shard distribution after data insertion

Shard shard1ReplSet at shard1ReplSet/localhost:27022,localhost:27023,localhost:27024

{

data: '389KiB',

docs: 5052,

chunks: 1,

'estimated data per chunk': '389KiB',

'estimated docs per chunk': 5052

}

---

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

{

data: '381KiB',

docs: 4948,

chunks: 1,

'estimated data per chunk': '381KiB',

'estimated docs per chunk': 4948

}

---

Totals

{

data: '770KiB',

docs: 10000,

chunks: 2,

'Shard shard1ReplSet': [

'50.51 % data',

'50.52 % docs in cluster',

'78B avg obj size on shard'

],

'Shard shard2ReplSet': [

'49.48 % data',

'49.48 % docs in cluster',

'78B avg obj size on shard'

]

}