mongosh --port 26050

rs.initiate({

\_id: "configReplSet",

configsvr: true,

members: [

{ \_id: 0, host: "localhost:26050" },

{ \_id: 1, host: "localhost:26051" },

{ \_id: 2, host: "localhost:26052" }

]

})

Current Mongosh Log ID: 684c24f272e93fcef7c59f34

Connecting to: mongodb://127.0.0.1:26050/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.1

Using MongoDB: 8.0.9

Using Mongosh: 2.5.1

mongosh 2.5.2 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

2025-06-13T18:45:55.780+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem

2025-06-13T18:45:56.117+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2025-06-13T18:45:56.117+05:30: Soft rlimits for open file descriptors too low

2025-06-13T18:45:56.117+05:30: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile

2025-06-13T18:45:56.117+05:30: We suggest setting the contents of sysfsFile to 0.

2025-06-13T18:45:56.117+05:30: Your system has glibc support for rseq built in, which is not yet supported by tcmalloc-google and has critical performance implications. Please set the environment variable GLIBC\_TUNABLES=glibc.pthread.rseq=0

2025-06-13T18:45:56.117+05:30: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.

------

test> rs.initiate({\_id: "configReplSet",configsvr : true, members : [{\_id : 0, host : "localhost:26050"},{\_id : 1, host: "localhost:26051"},{\_id:2, host:"localhost:26052"}]})

{

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749820816, i: 1 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749820816, i: 1 })

}

configReplSet [direct: secondary] test> rs.status

[Function: status] AsyncFunction {

apiVersions: [ 0, 0 ],

returnsPromise: true,

serverVersions: [ '0.0.0', '999.999.999' ],

topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],

returnType: { type: 'unknown', attributes: {} },

deprecated: false,

platforms: [ 'Compass', 'Browser', 'CLI' ],

isDirectShellCommand: false,

acceptsRawInput: false,

shellCommandCompleter: undefined,

help: [Function (anonymous)] Help

}

configReplSet [direct: primary] test> rs.status()

{

set: 'configReplSet',

date: ISODate('2025-06-13T13:21:12.795Z'),

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: 1749820871, i: 1 }), t: Long('1') },

lastCommittedWallTime: ISODate('2025-06-13T13:21:11.976Z'),

readConcernMajorityOpTime: { ts: Timestamp({ t: 1749820871, i: 1 }), t: Long('1') },

appliedOpTime: { ts: Timestamp({ t: 1749820871, i: 1 }), t: Long('1') },

durableOpTime: { ts: Timestamp({ t: 1749820871, i: 1 }), t: Long('1') },

writtenOpTime: { ts: Timestamp({ t: 1749820871, i: 1 }), t: Long('1') },

lastAppliedWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastDurableWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastWrittenWallTime: ISODate('2025-06-13T13:21:11.976Z')

},

lastStableRecoveryTimestamp: Timestamp({ t: 1749820816, i: 1 }),

electionCandidateMetrics: {

lastElectionReason: 'electionTimeout',

lastElectionDate: ISODate('2025-06-13T13:20:27.594Z'),

electionTerm: Long('1'),

lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1749820816, i: 1 }), t: Long('-1') },

lastSeenWrittenOpTimeAtElection: { ts: Timestamp({ t: 1749820816, i: 1 }), t: Long('-1') },

lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1749820816, i: 1 }), t: Long('-1') },

numVotesNeeded: 2,

priorityAtElection: 1,

electionTimeoutMillis: Long('10000'),

numCatchUpOps: Long('0'),

newTermStartDate: ISODate('2025-06-13T13:20:27.654Z'),

wMajorityWriteAvailabilityDate: ISODate('2025-06-13T13:20:28.120Z')

},

members: [

{

\_id: 0,

name: 'localhost:26050',

health: 1,

state: 1,

stateStr: 'PRIMARY',

uptime: 317,

optime: { ts: Timestamp({ t: 1749820871, i: 1 }), t: Long('1') },

optimeDate: ISODate('2025-06-13T13:21:11.000Z'),

optimeWritten: { ts: Timestamp({ t: 1749820871, i: 1 }), t: Long('1') },

optimeWrittenDate: ISODate('2025-06-13T13:21:11.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastDurableWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastWrittenWallTime: ISODate('2025-06-13T13:21:11.976Z'),

syncSourceHost: '',

syncSourceId: -1,

infoMessage: 'Could not find member to sync from',

electionTime: Timestamp({ t: 1749820827, i: 1 }),

electionDate: ISODate('2025-06-13T13:20:27.000Z'),

configVersion: 1,

configTerm: 1,

self: true,

lastHeartbeatMessage: ''

},

{

\_id: 1,

name: 'localhost:26051',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 55,

optime: { ts: Timestamp({ t: 1749820870, i: 1 }), t: Long('1') },

optimeDurable: { ts: Timestamp({ t: 1749820870, i: 1 }), t: Long('1') },

optimeWritten: { ts: Timestamp({ t: 1749820870, i: 1 }), t: Long('1') },

optimeDate: ISODate('2025-06-13T13:21:10.000Z'),

optimeDurableDate: ISODate('2025-06-13T13:21:10.000Z'),

optimeWrittenDate: ISODate('2025-06-13T13:21:10.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastDurableWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastWrittenWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastHeartbeat: ISODate('2025-06-13T13:21:11.629Z'),

lastHeartbeatRecv: ISODate('2025-06-13T13:21:12.632Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: 'localhost:26050',

syncSourceId: 0,

infoMessage: '',

configVersion: 1,

configTerm: 1

},

{

\_id: 2,

name: 'localhost:26052',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 55,

optime: { ts: Timestamp({ t: 1749820870, i: 1 }), t: Long('1') },

optimeDurable: { ts: Timestamp({ t: 1749820870, i: 1 }), t: Long('1') },

optimeWritten: { ts: Timestamp({ t: 1749820870, i: 1 }), t: Long('1') },

optimeDate: ISODate('2025-06-13T13:21:10.000Z'),

optimeDurableDate: ISODate('2025-06-13T13:21:10.000Z'),

optimeWrittenDate: ISODate('2025-06-13T13:21:10.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastDurableWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastWrittenWallTime: ISODate('2025-06-13T13:21:11.976Z'),

lastHeartbeat: ISODate('2025-06-13T13:21:11.628Z'),

lastHeartbeatRecv: ISODate('2025-06-13T13:21:12.632Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: 'localhost:26050',

syncSourceId: 0,

infoMessage: '',

configVersion: 1,

configTerm: 1

}

],

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749820871, i: 1 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749820871, i: 1 })

}

configReplSet [direct: primary] test>

mongosh --port 27021

Current Mongosh Log ID: 684c2c6a7aebd4a1f9c59f34

Connecting to: mongodb://127.0.0.1:27021/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.1

Using MongoDB: 8.0.10

Using Mongosh: 2.5.1

mongosh 2.5.2 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

2025-06-13T19:11:42.189+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem

2025-06-13T19:11:42.320+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2025-06-13T19:11:42.320+05:30: Soft rlimits for open file descriptors too low

2025-06-13T19:11:42.320+05:30: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile

2025-06-13T19:11:42.320+05:30: We suggest setting the contents of sysfsFile to 0.

2025-06-13T19:11:42.320+05:30: Your system has glibc support for rseq built in, which is not yet supported by tcmalloc-google and has critical performance implications. Please set the environment variable GLIBC\_TUNABLES=glibc.pthread.rseq=0

2025-06-13T19:11:42.320+05:30: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.

------

test> rs.initiate({\_id : "shardReplSet1", members : [{\_id :0, host : "localhost:27021"},{\_id: 1, host : "localhost:27022"},{\_id : 2, host : "localhost:27023"}]})

{

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749822766, i: 1 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749822766, i: 1 })

}

shardReplSet1 [direct: secondary] test> rs.status()

{

set: 'shardReplSet1',

date: ISODate('2025-06-13T13:53:09.101Z'),

myState: 1,

term: Long('1'),

syncSourceHost: '',

syncSourceId: -1,

heartbeatIntervalMillis: Long('2000'),

majorityVoteCount: 2,

writeMajorityCount: 2,

votingMembersCount: 3,

writableVotingMembersCount: 3,

optimes: {

lastCommittedOpTime: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

lastCommittedWallTime: ISODate('2025-06-13T13:52:58.547Z'),

readConcernMajorityOpTime: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

appliedOpTime: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

durableOpTime: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

writtenOpTime: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

lastAppliedWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastDurableWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastWrittenWallTime: ISODate('2025-06-13T13:52:58.547Z')

},

lastStableRecoveryTimestamp: Timestamp({ t: 1749822766, i: 1 }),

electionCandidateMetrics: {

lastElectionReason: 'electionTimeout',

lastElectionDate: ISODate('2025-06-13T13:52:58.320Z'),

electionTerm: Long('1'),

lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1749822766, i: 1 }), t: Long('-1') },

lastSeenWrittenOpTimeAtElection: { ts: Timestamp({ t: 1749822766, i: 1 }), t: Long('-1') },

lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1749822766, i: 1 }), t: Long('-1') },

numVotesNeeded: 2,

priorityAtElection: 1,

electionTimeoutMillis: Long('10000'),

numCatchUpOps: Long('0'),

newTermStartDate: ISODate('2025-06-13T13:52:58.411Z'),

wMajorityWriteAvailabilityDate: ISODate('2025-06-13T13:52:58.844Z')

},

members: [

{

\_id: 0,

name: 'localhost:27021',

health: 1,

state: 1,

stateStr: 'PRIMARY',

uptime: 687,

optime: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

optimeDate: ISODate('2025-06-13T13:52:58.000Z'),

optimeWritten: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

optimeWrittenDate: ISODate('2025-06-13T13:52:58.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastDurableWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastWrittenWallTime: ISODate('2025-06-13T13:52:58.547Z'),

syncSourceHost: '',

syncSourceId: -1,

infoMessage: 'Could not find member to sync from',

electionTime: Timestamp({ t: 1749822778, i: 1 }),

electionDate: ISODate('2025-06-13T13:52:58.000Z'),

configVersion: 1,

configTerm: 1,

self: true,

lastHeartbeatMessage: ''

},

{

\_id: 1,

name: 'localhost:27022',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 22,

optime: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

optimeDurable: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

optimeWritten: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

optimeDate: ISODate('2025-06-13T13:52:58.000Z'),

optimeDurableDate: ISODate('2025-06-13T13:52:58.000Z'),

optimeWrittenDate: ISODate('2025-06-13T13:52:58.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastDurableWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastWrittenWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastHeartbeat: ISODate('2025-06-13T13:53:08.343Z'),

lastHeartbeatRecv: ISODate('2025-06-13T13:53:07.344Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: 'localhost:27021',

syncSourceId: 0,

infoMessage: '',

configVersion: 1,

configTerm: 1

},

{

\_id: 2,

name: 'localhost:27023',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 22,

optime: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

optimeDurable: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

optimeWritten: { ts: Timestamp({ t: 1749822778, i: 15 }), t: Long('1') },

optimeDate: ISODate('2025-06-13T13:52:58.000Z'),

optimeDurableDate: ISODate('2025-06-13T13:52:58.000Z'),

optimeWrittenDate: ISODate('2025-06-13T13:52:58.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastDurableWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastWrittenWallTime: ISODate('2025-06-13T13:52:58.547Z'),

lastHeartbeat: ISODate('2025-06-13T13:53:08.344Z'),

lastHeartbeatRecv: ISODate('2025-06-13T13:53:08.846Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: 'localhost:27021',

syncSourceId: 0,

infoMessage: '',

configVersion: 1,

configTerm: 1

}

],

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749822778, i: 15 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749822778, i: 15 })

}

shardReplSet1 [direct: primary] test>

mongosh --port 27024

Current Mongosh Log ID: 684c363a3e41b5abb9c59f34

Connecting to: mongodb://127.0.0.1:27024/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.1

Using MongoDB: 8.0.10

Using Mongosh: 2.5.1

mongosh 2.5.2 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

2025-06-13T19:25:17.445+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------

[direct: mongos] test> use testdb

switched to db testdb

[direct: mongos] testdb> for(let i=1; i <= 2000 ; i++){db.newUsers.insertOne({name : "Users" + i, age : Math.floor(Math.random()\*50 + 20)})}

{

acknowledged: true,

insertedId: ObjectId('684c36da3e41b5abb9c5a704')

}

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

{

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749825262, i: 1 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749825262, i: 1 })

}

[direct: mongos] testdb> sh.shardCollection("testdb.newUsers", {age : 1})

MongoServerError[InvalidOptions]: Please create an index that starts with the proposed shard key before sharding the collection.

[direct: mongos] testdb> sh.shardCollection("testdb.newUsers", {age : 1})

[direct: mongos] testdb> db.newUsers.createIndex({age : 1})

age\_1

[direct: mongos] testdb> sh.shardCollection("testdb.newUsers", {age : 1})

{

collectionsharded: 'testdb.newUsers',

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749825409, i: 38 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749825409, i: 37 })

}

[direct: mongos] testdb> db.newUsers.getShardDistribution()

Shard shardReplSet1 at shardReplSet1/localhost:27021,localhost:27022,localhost:27023

{

data: '98KiB',

docs: 2000,

chunks: 1,

'estimated data per chunk': '98KiB',

'estimated docs per chunk': 2000

}

---

Totals

{

data: '98KiB',

docs: 2000,

chunks: 1,

'Shard shardReplSet1': [

'100 % data',

'100 % docs in cluster',

'50B avg obj size on shard'

]

}

[direct: mongos] testdb> sh.startBalancer()

{

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749825471, i: 4 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749825471, i: 4 })

}

[direct: mongos] testdb>

mongosh --port 27031

Current Mongosh Log ID: 684c3580f3bb6e0c7ac59f34

Connecting to: mongodb://127.0.0.1:27031/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.1

Using MongoDB: 8.0.10

Using Mongosh: 2.5.1

mongosh 2.5.2 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

2025-06-13T19:57:38.191+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem

2025-06-13T19:57:38.338+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2025-06-13T19:57:38.338+05:30: Soft rlimits for open file descriptors too low

2025-06-13T19:57:38.338+05:30: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile

2025-06-13T19:57:38.338+05:30: We suggest setting the contents of sysfsFile to 0.

2025-06-13T19:57:38.338+05:30: Your system has glibc support for rseq built in, which is not yet supported by tcmalloc-google and has critical performance implications. Please set the environment variable GLIBC\_TUNABLES=glibc.pthread.rseq=0

2025-06-13T19:57:38.338+05:30: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.

------

test> rs.initiate({\_id : "shardReplSet2", members :[{\_id : 0, host : "localhost:27031"},{\_id : 1, host : "localhost:27032"},{\_id : 2, host : "localhost:27033"}]})

{

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749825023, i: 1 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749825023, i: 1 })

}

shardReplSet2 [direct: secondary] test> rs.status()

{

set: 'shardReplSet2',

date: ISODate('2025-06-13T14:30:35.156Z'),

myState: 1,

term: Long('1'),

syncSourceHost: '',

syncSourceId: -1,

heartbeatIntervalMillis: Long('2000'),

majorityVoteCount: 2,

writeMajorityCount: 2,

votingMembersCount: 3,

writableVotingMembersCount: 3,

optimes: {

lastCommittedOpTime: { ts: Timestamp({ t: 1749825034, i: 15 }), t: Long('1') },

lastCommittedWallTime: ISODate('2025-06-13T14:30:34.764Z'),

readConcernMajorityOpTime: { ts: Timestamp({ t: 1749825034, i: 15 }), t: Long('1') },

appliedOpTime: { ts: Timestamp({ t: 1749825034, i: 15 }), t: Long('1') },

durableOpTime: { ts: Timestamp({ t: 1749825034, i: 15 }), t: Long('1') },

writtenOpTime: { ts: Timestamp({ t: 1749825034, i: 15 }), t: Long('1') },

lastAppliedWallTime: ISODate('2025-06-13T14:30:34.764Z'),

lastDurableWallTime: ISODate('2025-06-13T14:30:34.764Z'),

lastWrittenWallTime: ISODate('2025-06-13T14:30:34.764Z')

},

lastStableRecoveryTimestamp: Timestamp({ t: 1749825023, i: 1 }),

electionCandidateMetrics: {

lastElectionReason: 'electionTimeout',

lastElectionDate: ISODate('2025-06-13T14:30:34.545Z'),

electionTerm: Long('1'),

lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

lastSeenWrittenOpTimeAtElection: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

numVotesNeeded: 2,

priorityAtElection: 1,

electionTimeoutMillis: Long('10000'),

numCatchUpOps: Long('0'),

newTermStartDate: ISODate('2025-06-13T14:30:34.625Z'),

wMajorityWriteAvailabilityDate: ISODate('2025-06-13T14:30:35.074Z')

},

members: [

{

\_id: 0,

name: 'localhost:27031',

health: 1,

state: 1,

stateStr: 'PRIMARY',

uptime: 177,

optime: { ts: Timestamp({ t: 1749825034, i: 15 }), t: Long('1') },

optimeDate: ISODate('2025-06-13T14:30:34.000Z'),

optimeWritten: { ts: Timestamp({ t: 1749825034, i: 15 }), t: Long('1') },

optimeWrittenDate: ISODate('2025-06-13T14:30:34.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T14:30:34.764Z'),

lastDurableWallTime: ISODate('2025-06-13T14:30:34.764Z'),

lastWrittenWallTime: ISODate('2025-06-13T14:30:34.764Z'),

syncSourceHost: '',

syncSourceId: -1,

infoMessage: 'Could not find member to sync from',

electionTime: Timestamp({ t: 1749825034, i: 1 }),

electionDate: ISODate('2025-06-13T14:30:34.000Z'),

configVersion: 1,

configTerm: 1,

self: true,

lastHeartbeatMessage: ''

},

{

\_id: 1,

name: 'localhost:27032',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 11,

optime: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

optimeDurable: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

optimeWritten: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

optimeDate: ISODate('2025-06-13T14:30:23.000Z'),

optimeDurableDate: ISODate('2025-06-13T14:30:23.000Z'),

optimeWrittenDate: ISODate('2025-06-13T14:30:23.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T14:30:34.575Z'),

lastDurableWallTime: ISODate('2025-06-13T14:30:34.764Z'),

lastWrittenWallTime: ISODate('2025-06-13T14:30:34.764Z'),

lastHeartbeat: ISODate('2025-06-13T14:30:34.565Z'),

lastHeartbeatRecv: ISODate('2025-06-13T14:30:35.066Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: '',

syncSourceId: -1,

infoMessage: '',

configVersion: 1,

configTerm: 1

},

{

\_id: 2,

name: 'localhost:27033',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 11,

optime: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

optimeDurable: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

optimeWritten: { ts: Timestamp({ t: 1749825023, i: 1 }), t: Long('-1') },

optimeDate: ISODate('2025-06-13T14:30:23.000Z'),

optimeDurableDate: ISODate('2025-06-13T14:30:23.000Z'),

optimeWrittenDate: ISODate('2025-06-13T14:30:23.000Z'),

lastAppliedWallTime: ISODate('2025-06-13T14:30:34.575Z'),

lastDurableWallTime: ISODate('2025-06-13T14:30:34.764Z'),

lastWrittenWallTime: ISODate('2025-06-13T14:30:34.764Z'),

lastHeartbeat: ISODate('2025-06-13T14:30:34.563Z'),

lastHeartbeatRecv: ISODate('2025-06-13T14:30:35.064Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: '',

syncSourceId: -1,

infoMessage: '',

configVersion: 1,

configTerm: 1

}

],

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1749825034, i: 15 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1749825034, i: 15 })

}

shardReplSet2 [direct: primary] test>