TERMINAL -1

PS C:\data\rs1> Start-Process "mongod" -ArgumentList "--replSet m101 --logpath \data\rs1\1.log --dbpath \data\rs1 --port 27020"

PS C:\data\rs1> mongosh --port 27020

Current Mongosh Log ID: 66c1a025198958bd8646b798

Connecting to: mongodb://127.0.0.1:27020/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.5

Using MongoDB: 7.0.9

Using Mongosh: 2.2.5

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

------

The server generated these startup warnings when booting

2024-08-18T12:41:41.838+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2024-08-18T12:41:41.838+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

------

test> config ={id: "m101",members:[{\_id:0,host:"localhost:27020"},{\_id:1,host:"localhost:27021"},{\_id:2,host:"localhost:27022"}]};

{

id: 'm101',

members: [

{ \_id: 0, host: 'localhost:27020' },

{ \_id: 1, host: 'localhost:27021' },

{ \_id: 2, host: 'localhost:27022' }

]

}

test> rs.initiate(config);

MongoServerError[InvalidReplicaSetConfig]: BSON field 'ReplSetConfig.id' is an unknown field.

test>

(To exit, press Ctrl+C again or Ctrl+D or type .exit)

test> config = {

... \_id: "m101", // This is the correct field name for the replica set name

... members: [

... { \_id: 0, host: "localhost:27020" },

... { \_id: 1, host: "localhost:27021" },

... { \_id: 2, host: "localhost:27022" }

... ]

... };

{

\_id: 'm101',

members: [

{ \_id: 0, host: 'localhost:27020' },

{ \_id: 1, host: 'localhost:27021' },

{ \_id: 2, host: 'localhost:27022' }

]

}

test> rs.initiate(config);

{ ok: 1 }

m101 [direct: other] test> rs.status();

{

set: 'm101',

date: ISODate('2024-08-18T07:28:29.384Z'),

myState: 1,

term: Long('1'),

syncSourceHost: '',

syncSourceId: -1,

heartbeatIntervalMillis: Long('2000'),

majorityVoteCount: 2,

writeMajorityCount: 2,

votingMembersCount: 3,

writableVotingMembersCount: 3,

optimes: {

lastCommittedOpTime: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

lastCommittedWallTime: ISODate('2024-08-18T07:28:26.463Z'),

readConcernMajorityOpTime: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

appliedOpTime: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

durableOpTime: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

lastAppliedWallTime: ISODate('2024-08-18T07:28:26.463Z'),

lastDurableWallTime: ISODate('2024-08-18T07:28:26.463Z')

},

lastStableRecoveryTimestamp: Timestamp({ t: 1723966095, i: 1 }),

electionCandidateMetrics: {

lastElectionReason: 'electionTimeout',

lastElectionDate: ISODate('2024-08-18T07:28:25.811Z'),

electionTerm: Long('1'),

lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1723966095, i: 1 }), t: Long('-1') },

lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1723966095, i: 1 }), t: Long('-1') },

numVotesNeeded: 2,

priorityAtElection: 1,

electionTimeoutMillis: Long('10000'),

numCatchUpOps: Long('0'),

newTermStartDate: ISODate('2024-08-18T07:28:25.858Z'),

wMajorityWriteAvailabilityDate: ISODate('2024-08-18T07:28:26.398Z')

},

members: [

{

\_id: 0,

name: 'localhost:27020',

health: 1,

state: 1,

stateStr: 'PRIMARY',

uptime: 1009,

optime: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

optimeDate: ISODate('2024-08-18T07:28:26.000Z'),

lastAppliedWallTime: ISODate('2024-08-18T07:28:26.463Z'),

lastDurableWallTime: ISODate('2024-08-18T07:28:26.463Z'),

syncSourceHost: '',

syncSourceId: -1,

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

electionTime: Timestamp({ t: 1723966105, i: 1 }),

electionDate: ISODate('2024-08-18T07:28:25.000Z'),

configVersion: 1,

configTerm: 1,

self: true,

lastHeartbeatMessage: ''

},

{

\_id: 1,

name: 'localhost:27021',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 13,

optime: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

optimeDurable: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

optimeDate: ISODate('2024-08-18T07:28:26.000Z'),

optimeDurableDate: ISODate('2024-08-18T07:28:26.000Z'),

lastAppliedWallTime: ISODate('2024-08-18T07:28:26.463Z'),

lastDurableWallTime: ISODate('2024-08-18T07:28:26.463Z'),

lastHeartbeat: ISODate('2024-08-18T07:28:27.836Z'),

lastHeartbeatRecv: ISODate('2024-08-18T07:28:28.865Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: 'localhost:27020',

syncSourceId: 0,

infoMessage: '',

configVersion: 1,

configTerm: 1

},

{

\_id: 2,

name: 'localhost:27022',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 13,

optime: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

optimeDurable: { ts: Timestamp({ t: 1723966106, i: 6 }), t: Long('1') },

optimeDate: ISODate('2024-08-18T07:28:26.000Z'),

optimeDurableDate: ISODate('2024-08-18T07:28:26.000Z'),

lastAppliedWallTime: ISODate('2024-08-18T07:28:26.463Z'),

lastDurableWallTime: ISODate('2024-08-18T07:28:26.463Z'),

lastHeartbeat: ISODate('2024-08-18T07:28:27.837Z'),

lastHeartbeatRecv: ISODate('2024-08-18T07:28:28.864Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: 'localhost:27020',

syncSourceId: 0,

infoMessage: '',

configVersion: 1,

configTerm: 1

}

],

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1723966106, i: 6 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1723966106, i: 6 })

}

m101 [direct: primary] test> use CBIT

switched to db CBIT

m101 [direct: primary] CBIT> db.it3.find();

m101 [direct: primary] CBIT> db.shutdownServer();

Error: read ECONNRESET

CBIT> db.shutdownserver();

TypeError: db.shutdownserver is not a function

CBIT> db.shutdownServer();

Uncaught:

MongoServerSelectionError: connect ECONNREFUSED 127.0.0.1:27020

Caused by:

MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27020

Caused by:

Error: connect ECONNREFUSED 127.0.0.1:27020

CBIT>

TERMINAL -2

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\data\rs2> Start-Process "mongod" -ArgumentList "--replSet m101 --logpath \data\rs2\1.log --dbpath \data\rs2 --port 27021"

PS C:\data\rs2> mongosh --port 27021

Current Mongosh Log ID: 66c1a06f35a9fdcede46b798

Connecting to: mongodb://127.0.0.1:27021/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.5

Using MongoDB: 7.0.9

Using Mongosh: 2.2.5

mongosh 2.3.0 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

2024-08-18T12:46:17.638+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2024-08-18T12:46:17.638+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

------

test> rs.secondaryOk();

DeprecationWarning: .setSecondaryOk() is deprecated. Use .setReadPref("primaryPreferred") instead

Setting read preference from "primary" to "primaryPreferred"

m101 [direct: secondary] test> rs.secondaryOk()

Leaving read preference unchanged (is already "primaryPreferred")

m101 [direct: secondary] test> db.products.find();

m101 [direct: secondary] test>

TERMINAL-3

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\data\rs3> Start-Process "mongod" -ArgumentList "--replSet m101 --logpath \data\rs3\1.log --dbpath \data\rs3 --port

27022"

PS C:\data\rs3> mongosh --port 27022

Current Mongosh Log ID: 66c1a07c2d17a4c61e46b798

Connecting to: mongodb://127.0.0.1:27022/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.5

Using MongoDB: 7.0.9

Using Mongosh: 2.2.5

mongosh 2.3.0 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

2024-08-18T12:47:09.627+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

2024-08-18T12:47:09.627+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

------

test> rs secondayOk()

Uncaught:

SyntaxError: Missing semicolon. (1:2)

> 1 | rs secondayOk()

| ^

2 |

m101 [direct: secondary] test> rs secondaryOk()

Uncaught:

SyntaxError: Missing semicolon. (1:2)

> 1 | rs secondaryOk()

| ^

2 |

m101 [direct: secondary] test> rs.secondaryOk()

DeprecationWarning: .setSecondaryOk() is deprecated. Use .setReadPref("primaryPreferred") instead

Setting read preference from "primary" to "primaryPreferred"

m101 [direct: secondary] test> rs.status();

{

set: 'm101',

date: ISODate('2024-08-18T07:33:19.287Z'),

myState: 2,

term: Long('1'),

syncSourceHost: 'localhost:27020',

syncSourceId: 0,

heartbeatIntervalMillis: Long('2000'),

majorityVoteCount: 2,

writeMajorityCount: 2,

votingMembersCount: 3,

writableVotingMembersCount: 3,

optimes: {

lastCommittedOpTime: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

lastCommittedWallTime: ISODate('2024-08-18T07:33:16.106Z'),

readConcernMajorityOpTime: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

appliedOpTime: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

durableOpTime: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

lastAppliedWallTime: ISODate('2024-08-18T07:33:16.106Z'),

lastDurableWallTime: ISODate('2024-08-18T07:33:16.106Z')

},

lastStableRecoveryTimestamp: Timestamp({ t: 1723966346, i: 1 }),

electionParticipantMetrics: {

votedForCandidate: true,

electionTerm: Long('1'),

lastVoteDate: ISODate('2024-08-18T07:28:25.815Z'),

electionCandidateMemberId: 0,

voteReason: '',

lastAppliedOpTimeAtElection: { ts: Timestamp({ t: 1723966095, i: 1 }), t: Long('-1') },

maxAppliedOpTimeInSet: { ts: Timestamp({ t: 1723966095, i: 1 }), t: Long('-1') },

priorityAtElection: 1,

newTermStartDate: ISODate('2024-08-18T07:28:25.858Z'),

newTermAppliedDate: ISODate('2024-08-18T07:28:26.409Z')

},

members: [

{

\_id: 0,

name: 'localhost:27020',

health: 1,

state: 1,

stateStr: 'PRIMARY',

uptime: 303,

optime: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

optimeDurable: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

optimeDate: ISODate('2024-08-18T07:33:16.000Z'),

optimeDurableDate: ISODate('2024-08-18T07:33:16.000Z'),

lastAppliedWallTime: ISODate('2024-08-18T07:33:16.106Z'),

lastDurableWallTime: ISODate('2024-08-18T07:33:16.106Z'),

lastHeartbeat: ISODate('2024-08-18T07:33:18.460Z'),

lastHeartbeatRecv: ISODate('2024-08-18T07:33:17.402Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: '',

syncSourceId: -1,

infoMessage: '',

electionTime: Timestamp({ t: 1723966105, i: 1 }),

electionDate: ISODate('2024-08-18T07:28:25.000Z'),

configVersion: 1,

configTerm: 1

},

{

\_id: 1,

name: 'localhost:27021',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 303,

optime: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

optimeDurable: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

optimeDate: ISODate('2024-08-18T07:33:16.000Z'),

optimeDurableDate: ISODate('2024-08-18T07:33:16.000Z'),

lastAppliedWallTime: ISODate('2024-08-18T07:33:16.106Z'),

lastDurableWallTime: ISODate('2024-08-18T07:33:16.106Z'),

lastHeartbeat: ISODate('2024-08-18T07:33:18.477Z'),

lastHeartbeatRecv: ISODate('2024-08-18T07:33:18.476Z'),

pingMs: Long('0'),

lastHeartbeatMessage: '',

syncSourceHost: 'localhost:27020',

syncSourceId: 0,

infoMessage: '',

configVersion: 1,

configTerm: 1

},

{

\_id: 2,

name: 'localhost:27022',

health: 1,

state: 2,

stateStr: 'SECONDARY',

uptime: 972,

optime: { ts: Timestamp({ t: 1723966396, i: 1 }), t: Long('1') },

optimeDate: ISODate('2024-08-18T07:33:16.000Z'),

lastAppliedWallTime: ISODate('2024-08-18T07:33:16.106Z'),

lastDurableWallTime: ISODate('2024-08-18T07:33:16.106Z'),

syncSourceHost: 'localhost:27020',

syncSourceId: 0,

infoMessage: '',

configVersion: 1,

configTerm: 1,

self: true,

lastHeartbeatMessage: ''

}

],

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1723966396, i: 1 }),

signature: {

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

keyId: Long('0')

}

},

operationTime: Timestamp({ t: 1723966396, i: 1 })

}

m101 [direct: secondary] test>