**REPLICATION**

Commands to be run…..

db0 - mongod --port 27017 --dbpath ~/Desktop/mongoDB/replicaPractice/db0 --replSet "rs0" --bind\_ip localhost --fork --logpath ~/db0.log

db1- mongod --port 27018 --dbpath ~/Desktop/mongoDB/replicaPractice/db1 --replSet "rs0" --bind\_ip localhost --fork --logpath ~/db1.log

db2- mongod --port 27019 --dbpath ~/Desktop/mongoDB/replicaPractice/db2 --replSet "rs0" --bind\_ip localhost --fork --logpath ~/db2.log

mongosh --port 27017

rs.initiate({

\_id: "rs0",

members: [

{ \_id: 0, host: "localhost:27017" },

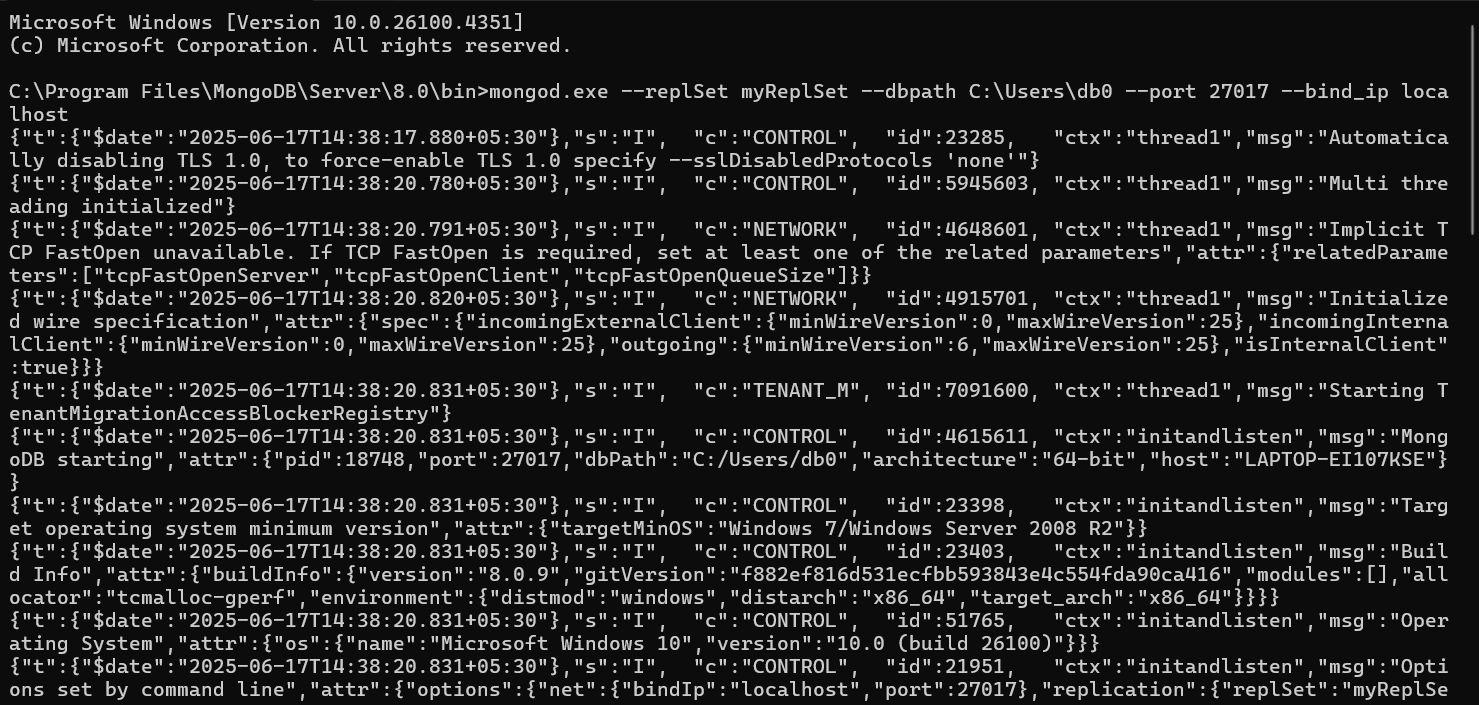
{ \_id: 1, host: "localhost:27018" },

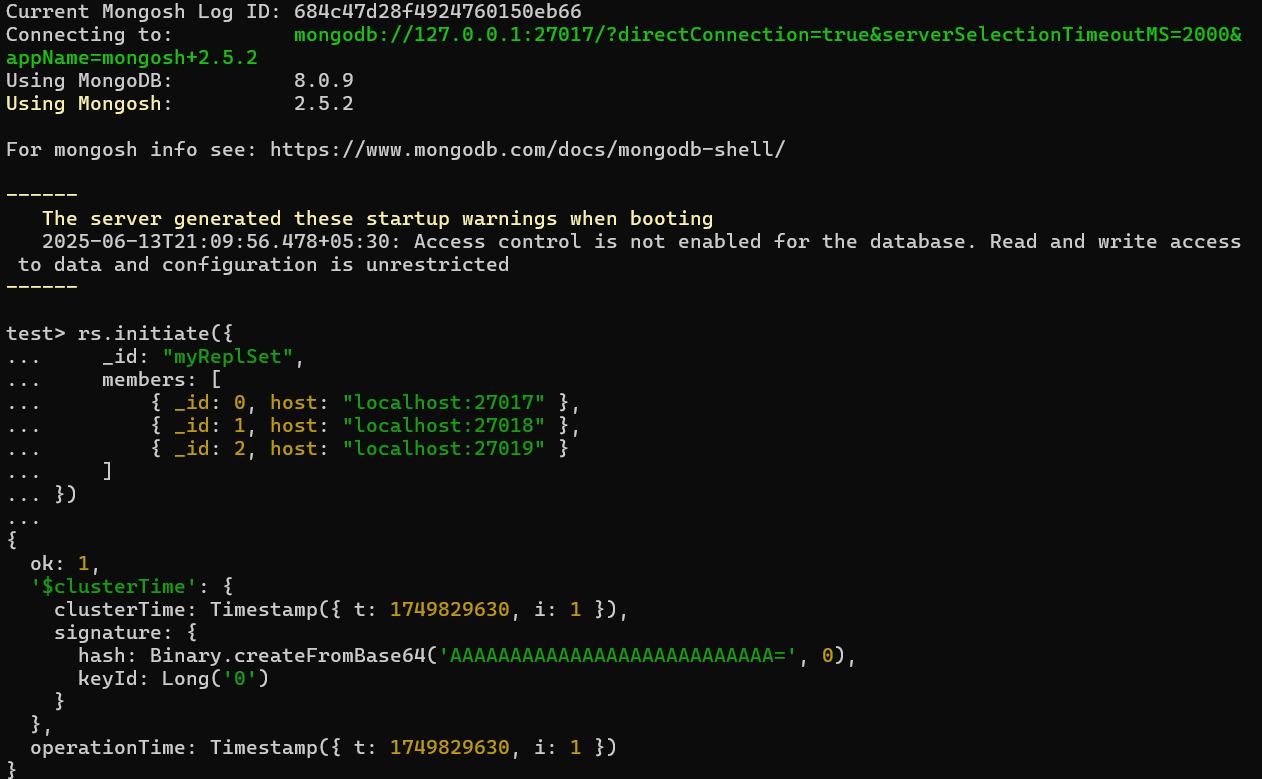
{ \_id: 2, host: "localhost:27019" }

]

})

rs.status()





**For rs. initiate**   
**test> rs.initiate({\_id: "rs0", members: [{\_id : 0, host: "localhost:27018"},{\_id : 1, host: "localhost:27019"}, {\_id : 2, host : "localhost:27020"}]})**

**{**

**ok: 1,**

**'$clusterTime': {**

**clusterTime: Timestamp({ t: 1749736156, i: 1 }),**

**signature: {**

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

**keyId: Long('0')**

**}},**

**operationTime: Timestamp({ t: 1749736156, i: 1 })**

**FOR rs.status**

**rs0 [direct: secondary] test> rs.status()**

**{**

**set: 'rs0',**

**date: ISODate('2025-06-12T13:49:36.129Z'),**

**myState: 1,**

**term: Long('1'),**

**syncSourceHost: '',**

**syncSourceId: -1,**

**heartbeatIntervalMillis: Long('2000'),**

**majorityVoteCount: 2,**

**writeMajorityCount: 2,**

**votingMembersCount: 3,**

**writableVotingMembersCount: 3,**

**optimes: {**

**lastCommittedOpTime: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**lastCommittedWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**readConcernMajorityOpTime: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**appliedOpTime: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**durableOpTime: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**writtenOpTime: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**lastAppliedWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastDurableWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastWrittenWallTime: ISODate('2025-06-12T13:49:27.252Z')**

**},**

**lastStableRecoveryTimestamp: Timestamp({ t: 1749736156, i: 1 }),**

**electionCandidateMetrics: {**

**lastElectionReason: 'electionTimeout',**

**lastElectionDate: ISODate('2025-06-12T13:49:27.079Z'),**

**electionTerm: Long('1'),**

**lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1749736156, i: 1 }), t: Long('-1') },**

**lastSeenWrittenOpTimeAtElection: { ts: Timestamp({ t: 1749736156, i: 1 }), t: Long('-1') },**

**lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1749736156, i: 1 }), t: Long('-1') },**

**numVotesNeeded: 2,**

**priorityAtElection: 1,**

**electionTimeoutMillis: Long('10000'),**

**numCatchUpOps: Long('0'),**

**newTermStartDate: ISODate('2025-06-12T13:49:27.139Z'),**

**wMajorityWriteAvailabilityDate: ISODate('2025-06-12T13:49:27.605Z')**

**},**

**members: [**

**{**

**\_id: 0,**

**name: 'localhost:27018',**

**health: 1,**

**state: 1,**

**stateStr: 'PRIMARY',**

**uptime: 615,**

**optime: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**optimeDate: ISODate('2025-06-12T13:49:27.000Z'),**

**optimeWritten: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**optimeWrittenDate: ISODate('2025-06-12T13:49:27.000Z'),**

**lastAppliedWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastDurableWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastWrittenWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**syncSourceHost: '',**

**syncSourceId: -1,**

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

**electionTime: Timestamp({ t: 1749736167, i: 1 }),**

**electionDate: ISODate('2025-06-12T13:49:27.000Z'),**

**configVersion: 1,**

**configTerm: 1,**

**self: true,**

**lastHeartbeatMessage: ''**

**},**

**{**

**\_id: 1,**

**name: 'localhost:27019',**

**health: 1,**

**state: 2,**

**stateStr: 'SECONDARY',**

**uptime: 19,**

**optime: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**optimeDurable: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**optimeWritten: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**optimeDate: ISODate('2025-06-12T13:49:27.000Z'),**

**optimeDurableDate: ISODate('2025-06-12T13:49:27.000Z'),**

**optimeWrittenDate: ISODate('2025-06-12T13:49:27.000Z'),**

**lastAppliedWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastDurableWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastWrittenWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastHeartbeat: ISODate('2025-06-12T13:49:35.103Z'),**

**lastHeartbeatRecv: ISODate('2025-06-12T13:49:36.105Z'),**

**pingMs: Long('0'),**

**lastHeartbeatMessage: '',**

**syncSourceHost: 'localhost:27018',**

**syncSourceId: 0,**

**infoMessage: '',**

**configVersion: 1,**

**configTerm: 1**

**},**

**{**

**\_id: 2,**

**name: 'localhost:27020',**

**health: 1,**

**state: 2,**

**stateStr: 'SECONDARY',**

**uptime: 19,**

**optime: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**optimeDurable: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**optimeWritten: { ts: Timestamp({ t: 1749736167, i: 16 }), t: Long('1') },**

**optimeDate: ISODate('2025-06-12T13:49:27.000Z'),**

**optimeDurableDate: ISODate('2025-06-12T13:49:27.000Z'),**

**optimeWrittenDate: ISODate('2025-06-12T13:49:27.000Z'),**

**lastAppliedWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastDurableWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastWrittenWallTime: ISODate('2025-06-12T13:49:27.252Z'),**

**lastHeartbeat: ISODate('2025-06-12T13:49:35.101Z'),**

**lastHeartbeatRecv: ISODate('2025-06-12T13:49:36.103Z'),**

**pingMs: Long('0'),**

**lastHeartbeatMessage: '',**

**syncSourceHost: 'localhost:27018',**

**syncSourceId: 0,**

**infoMessage: '',**

**configVersion: 1,**

**configTerm: 1**

**}**

**],**

**ok: 1,**

**'$clusterTime': {**

**clusterTime: Timestamp({ t: 1749736167, i: 16 }),**

**signature: {**

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

**keyId: Long('0')**

**}**

**},**

**operationTime: Timestamp({ t: 1749736167, i: 16 })**

**}**

**PRIMARY NODE**

**rs0 [direct: primary] test> use school**

**switched to db school**

**rs0 [direct: primary] school> db.students.insertOne({name : "Yathesht" , marks : 99})**

**{**

**acknowledged: true,**

**insertedId: ObjectId('684adc99977a3e88a4c59f35')**

**}**

**rs0 [direct: primary] school> db.students.insertOne({name : "DeleteMe", marks : 100})**

**{**

**acknowledged: true,**

**insertedId: ObjectId('684ae192977a3e88a4c59f36')**

**}**

**rs0 [direct: primary] school> db.students.deleteOne({name : "DeleteMe"})**

**{ acknowledged: true, deletedCount: 1 }**

**rs0 [direct: primary] school>**

**mongosh --port 27019**

**Current Mongosh Log ID: 684adcb5332658918bc59f34**

**Connecting to: mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.1**

**Using MongoDB: 8.0.9**

**Using Mongosh: 2.5.**

**SECONDARY**

**rs0 [direct: secondary] test> rs.slaveOk()**

**TypeError: rs.slaveOk is not a function**

**rs0 [direct: secondary] test> rs.slaveOk()**

**TypeError: rs.slaveOk is not a function**

**rs0 [direct: secondary] test>**

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

**rs0 [direct: secondary] test> db.getMongo().setReadPref("secondary")**

**rs0 [direct: secondary] test> use school**

**switched to db school**

**rs0 [direct: secondary] school> db.students.find()[{**

**\_id: ObjectId('684adc99977a3e88a4c59f35'),**

**name: 'Yathesht',**

**marks: 99}]**

**rs0 [direct: secondary] school> db.students.find({})[{**

**\_id: ObjectId('684adc99977a3e88a4c59f35'),**

**name: 'Yathesht',**

**marks: 99},{**

**\_id: ObjectId('684ae192977a3e88a4c59f36'),**

**name: 'DeleteMe',**

**marks: 100}]**

**rs0 [direct: secondary] school> db.students.find({})[{**

**\_id: ObjectId('684adc99977a3e88a4c59f35'),**

**name: 'Yathesht',**

**marks: 99} ]**