**PRACTICAL NO 4**

**REPLICATION, BACKUP AND RESTORE**

1. **Write a MongoDB query to create Replica of existing database.**

C:\Users\ugave>mongod --port 27017 --dbpath "C:\data\db"

{"t":{"$date":"2025-09-24T00:57:50.288+05:30"},"s":"I", "c":"CONTROL",

"id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}

{"t":{"$date":"2025-09-24T00:57:50.294+05:30"},"s":"I", "c":"CONTROL",

"id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}

C:\Users\ugave>mongod --port 27017

{"t":{"$date":"2025-09-24T00:58:59.762+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}

{"t":{"$date":"2025-09-24T00:58:59.765+05:30"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}

C:\Users\ugave>mongosh --port 27017

Current Mongosh Log ID: 68d2f519c30fba9f99eec4a8

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&

serverSelectionTimeoutMS=2000&appName=mongosh+2.5.6

Using MongoDB:8.0.9 Using Mongosh:2.5.6 mongosh 2.5.8

is available for download: <https://www.mongodb.com/try/download/shell>

test> rs.initiate()

MongoServerError[NoReplicationEnabled]: This node was not started with replication enabled.

**Primary Server (Port 27017) :**

C:\Users\ugave>mongod --port 27017 --dbpath "C:\data\db" --replSet primaryrs0 --bind\_ip localhost

{"t":{"$date":"2025-09-24T01:07:35.434+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}

**Secondary server1 (Port 27018) :**

C:\Users\ugave>mongod --port 27018 --dbpath "C:\data\db1" --replSet primaryrs0 --bind\_ip localhost

{"t":{"$date":"2025-09-24T01:10:29.901+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}

**Secondary server2 (Port 27019) command :**

C:\Users\ugave>mongod --port 27019 --dbpath "C:\data\db2" --replSet primaryrs0 --bind\_ip localhost

{"t":{"$date":"2025-09-24T01:11:48.390+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}

test> rs.initiate()

{ info2: 'no configuration specified. Using a default configuration for the set', me: '127.0.0.1:27017', ok: 1,

'$clusterTime': { clusterTime: Timestamp({ t: 1758656584, i: 1 }),

signature: { hash: Binary.createFromBase64('A=', 0),keyId:

Long('0') } }, operationTime: Timestamp({ t: 1758656584, i: 1 })}

primaryrs0 [direct: secondary] test> rs.reconfig({\_id: "primaryrs0", members:[{\_id : 0, host : "localhost:27017"},{\_id : 1, host : "localhost:27018"},{\_id : 2, host : "localhost:27019"}]},{force : true}) { ok: 1,'$clusterTime': {

clusterTime: Timestamp({ t: 1758656935, i: 1 }),signature: {

hash: Binary.createFromBase64('A=', 0), keyId: Long('0') }

}, operationTime: Timestamp({ t: 1758656935, i: 1 })}

primaryrs0 [direct: primary] test> rs.conf()

{\_id: 'primaryrs0', version: 13881, members: [

{\_id: 0, host: 'localhost:27017', arbiterOnly: false,

buildIndexes: true, hidden: false, priority: 1, tags: {},

secondaryDelaySecs: Long('0'), votes: 1 },{

\_id: 1, host: 'localhost:27018', arbiterOnly: false,

buildIndexes: true, hidden: false, priority: 1, tags: {},

secondaryDelaySecs: Long('0'), votes: 1},{

\_id: 2, host: 'localhost:27019', arbiterOnly: false,

buildIndexes: true, hidden: false, priority: 1, tags: {},

secondaryDelaySecs: Long('0'), votes: 1

}],

protocolVersion: Long('1'),

writeConcernMajorityJournalDefault: true,

settings: {chainingAllowed: true, heartbeatIntervalMillis: 2000,

heartbeatTimeoutSecs: 10, electionTimeoutMillis: 10000,

catchUpTimeoutMillis: -1, catchUpTakeoverDelayMillis: 30000,

getLastErrorModes: {},getLastErrorDefaults: { w: 1, wtimeout: 0 },

replicaSetId: ObjectId('68d2f8489a2c79d27e9df623')

}

}

primaryrs0 [direct: primary] test> rs.status()

{

set: 'primaryrs0',

date: ISODate('2025-09-23T19:52:19.319Z'),

myState: 1,

term: Long('1'),

syncSourceHost: '',

syncSourceId: -1,

heartbeatIntervalMillis: Long('2000'),

majorityVoteCount: 2,

writeMajorityCount: 2,

votingMembersCount: 3,

writableVotingMembersCount: 3,

optimes: {

lastCommittedOpTime: { ts: Timestamp({ t: 1758657135, i: 1 }),

t: Long('1') },

lastCommittedWallTime: ISODate('2025-09-23T19:52:15.239Z'),

readConcernMajorityOpTime: { ts: Timestamp({ t: 1758657135,

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

appliedOpTime: { ts: Timestamp({ t: 1758657135, i: 1 }),

t: Long('1') },

durableOpTime: { ts: Timestamp({ t: 1758657135, i: 1 }),

t: Long('1') },

writtenOpTime: { ts: Timestamp({ t: 1758657135, i: 1 }),

t: Long('1') },

lastAppliedWallTime: ISODate('2025-09-23T19:52:15.239Z'),

lastDurableWallTime: ISODate('2025-09-23T19:52:15.239Z'),

lastWrittenWallTime: ISODate('2025-09-23T19:52:15.239Z')

},

lastStableRecoveryTimestamp: Timestamp({ t: 1758657125, i: 1 }),

electionCandidateMetrics: {

lastElectionReason: 'electionTimeout',

lastElectionDate: ISODate('2025-09-23T19:43:05.047Z'),

electionTerm: Long('1'),

lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1758656584,

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

lastSeenWrittenOpTimeAtElection: { ts: Timestamp({ t: 1758656584,

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

lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1758656584, i: 1 }),

t: Long('-1') },

numVotesNeeded: 1,

priorityAtElection: 1,

electionTimeoutMillis: Long('10000'),

newTermStartDate: ISODate('2025-09-23T19:43:05.115Z'),

wMajorityWriteAvailabilityDate: ISODate('2025-09-23T19:43:05.

159Z')

},

members: [

{\_id: 0,name: 'localhost:27017', health: 1, state: 1,

stateStr: 'PRIMARY', uptime: 1091, },{

\_id: 1, name: 'localhost:27018', health: 1, state: 2,

stateStr: 'SECONDARY', uptime: 201,},{

\_id: 2, name: 'localhost:27019', health: 1, state: 2,

stateStr: 'SECONDARY', uptime: 201,

],

ok: 1,

'$clusterTime': {

clusterTime: Timestamp({ t: 1758657135, i: 1 }),

signature: { hash: Binary.createFromBase64('A=', 0),

keyId: Long('0') }

}, operationTime: Timestamp({ t: 1758657135, i: 1 }) }

primaryrs0 [direct: primary] test> show dbs

admin 112.00 KiB

config 176.00 KiB

local 436.00 KiB

primaryrs0 [direct: primary] test> use primary

switched to db primary

primaryrs0 [direct: primary] primary> db.createCollection("demo")

{ ok: 1 }

primaryrs0 [direct: primary] primary> db.demo.insertOne({\_id:1,

name:"yashodip"})

{ acknowledged: true, insertedId: 1 }

primaryrs0 [direct: primary] primary> db.demo.insertOne({\_id:2,

name:"lajari"})

{ acknowledged: true, insertedId: 2 }

primaryrs0 [direct: primary] primary> exit

switch to port 27018

primaryrs0 [direct: secondary] test> show dbs

admin 80.00 KiB

config 232.00 KiB

local 372.00 KiB

primary 40.00 KiB

primaryrs0 [direct: secondary] test> use primary

switched to db primary

primaryrs0 [direct: secondary] primary> show collections

demo

primaryrs0 [direct: secondary] primary> db.demo.find()

[ { \_id: 1, name: 'yashodip' }, { \_id: 2, name: 'lajari' } ]

primaryrs0 [direct: secondary] primary> db.demo.insertOne({\_id:3, name:"vaibhavi"})

MongoServerError[NotWritablePrimary]: not primary

primaryrs0 [direct: secondary] primary>

db.getMongo().setReadPref('secondary')

C:\Users\ugave>mongosh "mongodb://localhost:27017/?replicaSet=primaryrs0"

primaryrs0 [primary] test>

1. **Write a MongoDB query to create a backup of existing database.**
2. Backup Entire mongod instance :

C:\Users\ugave>mongodump

2025-09-24T20:55:27.081+0530 writing admin.system.version to dump\admin\system.version.bson

2025-09-24T20:55:27.087+0530 done dumping admin.system.version (1 document)

2025-09-24T20:55:27.087+0530 writing primary.demo to dump\primary\demo.bson

2025-09-24T20:55:27.097+0530 done dumping primary.demo (2 documents)

C:\Users\ugave>mongodump --out "C:\mongobackup"

2025-09-24T20:56:25.930+0530 writing admin.system.version to C:\mongobackup\admin\system.version.bson

2025-09-24T20:56:25.934+0530 done dumping admin.system.version (1 document)

2025-09-24T20:56:25.934+0530 writing primary.demo to C:\mongobackup\primary\demo.bson

2025-09-24T20:56:25.937+0530 done dumping primary.demo (2 documents)

1. For specific database backup :

C:\Users\ugave>mongodump --db "primary" --out "C:\mongobackup"

2025-09-24T21:03:33.232+0530 writing primary.demo to C:\mongobackup\primary\demo.bson

2025-09-24T21:03:33.236+0530 done dumping primary.demo (2 documents)

1. For specific collection backup within the database :

C:\Users\ugave>mongodump --db "primary" --collection "demo" --out "C:\mongobackup"

2025-09-24T21:06:53.638+0530 writing primary.demo to C:\mongobackup\primary\demo.bson

2025-09-24T21:06:53.643+0530 done dumping primary.demo (2 documents)

1. **Write a MongoDB query to restore database from the backup.**
2. Restore entire mongod instance :

C:\Users\ugave>mongorestore C:\mongobackup

2025-09-24T21:55:56.768+0530 preparing collections to restore from

2025-09-24T21:55:56.771+0530 don't know what to do with file "C:\mongobackup\prelude.json", skipping...

2025-09-24T21:55:56.771+0530 don't know what to do with file "C:\mongobackup\primary\prelude.json", skipping...

2025-09-24T21:55:56.771+0530 reading metadata for primary.demo from C:\mongobackup\primary\demo.metadata.json

2025-09-24T21:55:56.786+0530 restoring primary.demo from C:\mongobackup\primary\demo.bson

2025-09-24T21:55:56.797+0530 finished restoring primary.demo (2 documents, 0 failures)

2025-09-24T21:55:56.797+0530 no indexes to restore for collection primary.demo

2025-09-24T21:55:56.797+0530 2 document(s) restored successfully. 0 document(s).

1. Restore specific database of mongod instance :

C:\Users\ugave>mongorestore --db "primary" --dir C:\mongobackup\primary

2025-09-24T21:39:55.574+0530 The --db and --collection flags are deprecated for this use-case; please use --nsInclude instead, i.e. with --nsInclude=${DATABASE}.${COLLECTION}

2025-09-24T21:39:55.576+0530 building a list of collections to restore from C:\mongobackup\primary dir

2025-09-24T21:39:55.577+0530 don't know what to do with file "C:\mongobackup\primary\prelude.json", skipping...

2025-09-24T21:39:55.578+0530 reading metadata for primary.demo from C:\mongobackup\primary\demo.metadata.json

2025-09-24T21:39:55.595+0530 restoring primary.demo from C:\mongobackup\primary\demo.bson

2025-09-24T21:39:55.607+0530 finished restoring primary.demo (2 documents, 0 failures)

2025-09-24T21:39:55.607+0530 no indexes to restore for collection primary.demo

2025-09-24T21:39:55.607+0530 2 document(s) restored successfully. 0 document(s).

1. Restore a specific collation from database :

C:\Users\ugave>mongorestore -d primary -c demo C:\mongobackup\primary\demo.bson

2025-09-24T21:54:23.222+0530 checking for collection data in C:\mongobackup\primary\demo.bson

2025-09-24T21:54:23.224+0530 reading metadata for primary.demo from C:\mongobackup\primary\demo.metadata.json

2025-09-24T21:54:23.240+0530 restoring primary.demo from C:\mongobackup\primary\demo.bson

2025-09-24T21:54:23.283+0530 finished restoring primary.demo (2 documents, 0 failures)

2025-09-24T21:54:23.283+0530 no indexes to restore for collection primary.demo

2025-09-24T21:54:23.283+0530 2 document(s) restored successfully. 0 document(s).