

Practical 6

07-02-2025

Replication using Mongoddb

Write-up: -

- Replication
- Replication Architecture
- Replication benefits and limitation

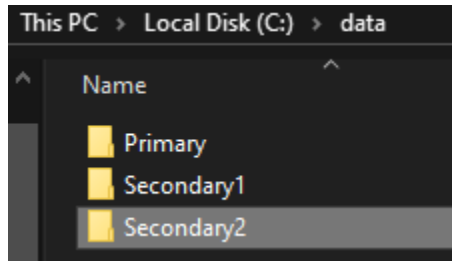
Implement Replication

You are a database administrator for a company, and you need to set up a MongoDB replica set to ensure high availability and data redundancy. Perform the following tasks:

- Initialize a replica set with three nodes on different ports (27017, 27018, 27019).
- Check the status of the replica set.
- Add a new secondary node to the existing replica set.
- Simulate a failover scenario by stepping down the primary and observing the election of a new primary.
- Check replication status and read from a secondary node using `readPreference`

Steps

1. In C Drive create a folder “data”
2. In the data folder create 3 folders
 - a. Primary folder
 - b. Secondary1 - First Secondary folder
 - c. Secondary2 - Second Secondary folder



3. Open Windows cmd / Powershell

4. Write the foll cmd

```
mongod --port=2717 --dbpath="C:\data\primary"
--replSet="test-replica-set"
```

```
Select Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Rehmah Ahmed> mongod --port=2717 --dbpath="C:\data\primary" --replSet="test-replica-set"
{"t":{"$date":"2025-02-12T23:41:11.236+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1",
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'}
{"t":{"$date":"2025-02-12T23:41:15.891+05:30"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1",
ading initialized"}
{"t":{"$date":"2025-02-12T23:41:15.892+05:30"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"thread1",
```

5. Open a new instance & write the foll cmd

```
mongod --port=2727 --dbpath="C:\data\secondary1"
--replSet="test-replica-set"
```

```
Select Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Rehmah Ahmed> mongod --port=2727 --dbpath="C:\data\secondary1" --replSet="test-replica-set"
{"t":{"$date":"2025-02-12T23:41:29.156+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1",
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'}
{"t":{"$date":"2025-02-12T23:41:34.526+05:30"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1",
ading initialized"}
{"t":{"$date":"2025-02-12T23:41:34.527+05:30"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"thread1",
```

6. Then create another instance and run the foll cmd
`mongod --port=2737 --dbpath="C:\data\secondary2"
--replSet="test-replica-set"`

```
Select Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Rehmah Ahmed> mongod --port=2737 --dbpath="C:\data\secondary2" --replSet="test-replica-set"
{"t":{"$date":"2025-02-12T23:41:46.371+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","m
nly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-02-12T23:41:51.998+05:30"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","m
ading initialized"}
{"t":{"$date":"2025-02-12T23:41:51.999+05:30"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"thread1","m
CP FastOpen unavailable. If TCP FastOpen is required, set at least one of the related parameters."attr":
```

7. Create a new instance and write the foll cmd
`mongosh --host="localhost:2717"`

```
mongosh mongodb://localhost:2717/?directConnection=true&serverSelectionTimeoutM
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Rehmah Ahmed> mongosh --host="localhost:2717"
Current Mongosh Log ID: 67ace7f00f9d5426634d7941
Connecting to:      mongodb://localhost:2717/?directConnection=true
3.9
Using MongoDB:      8.0.4
Using Mongosh:      2.3.9

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

-----
The server generated these startup warnings when booting
2025-02-12T23:55:55.478+05:30: Access control is not enabled for the
figuration is unrestricted
2025-02-12T23:55:55.480+05:30: This server is bound to localhost. Re
server. Start the server with --bind_ip <address> to specify which IP add
bind_ip_all to bind to all interfaces. If this behavior is desired, sta
this warning
-----
```

8. `rs.initiate()`

9. rs.add({host:"localhost:2727"});

```
test-replica-set [direct: primary] test> rs.add({host:"localhost:2727"});
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1739384911, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1739384911, i: 1 })
}
```

10. rs.add({host:"localhost:2737"});

```
test-replica-set [direct: primary] test> rs.add({host:"localhost:2737"});
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1739384920, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1739384920, i: 1 })
}
```

11. rs.status()

```
test-replica-set [direct: primary] test> rs.status()
{
  set: 'test-replica-set',
  date: ISODate('2025-02-12T18:28:49.684Z'),
  myState: 1,
  term: Long('2'),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1739384922, i: 1 }),
    lastCommittedWallTime: ISODate('2025-02-12T18:28:49.684Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1739384922, i: 1 }),
    appliedOpTime: { ts: Timestamp({ t: 1739384922, i: 1 }),
    durableOpTime: { ts: Timestamp({ t: 1739384922, i: 1 }),
    writtenOpTime: { ts: Timestamp({ t: 1739384922, i: 1 }),
    lastAppliedWallTime: ISODate('2025-02-12T18:28:49.684Z'),
    lastAppliedOpTime: { ts: Timestamp({ t: 1739384922, i: 1 })
  }
}
```

```
test-replica-set [direct: primary] test>
admin      80.00 KiB
config     128.00 KiB
local      452.00 KiB
```

The same databases should be reflecting in our secondary servers

Secondary1:

```
test-replica-set [direct: secondary] test> show dbs
admin      80.00 KiB
config     168.00 KiB
local      436.00 KiB
test-replica-set [direct: secondary] test>
```

Secondary2:

```
testtest-replica-set [direct: secondary] test> show dbs
admin      80.00 KiB
config     168.00 KiB
local      420.00 KiB
testtest-replica-set [direct: secondary] test>
```

15. Open the primary server instance then create database use practical6

```
test-replica-set [direct: primary] practical6>
switched to db practical6
```

16. Insert some records

db.users.insertMany([{ name: "Rehmah", roll_no: 10, age: 21 }, { name: "abc", roll_no: 18, age: 22 }, { name: "xyz", roll_no: 21, age: 23, address: "Mumbai" }, { name: "efg", roll_no: 12, address: "Delhi" }]);

```
test-replica-set [direct: primary] practical6> db.users.insertMany([ { name: "abc", roll_no: 18, age: 22 }, { name: "xyz", roll_no: 21, age: 23, address: "Delhi" } ]);
{local  452.00 KiB
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67acea1b0f9d5426634d7942'),
    '1': ObjectId('67acea1b0f9d5426634d7943'),
    '2': ObjectId('67acea1b0f9d5426634d7944'),
    '3': ObjectId('67acea1b0f9d5426634d7945')
  }
}
```

17. Reading the entries

db.users.find()

```
test-replica-set [direct: primary] practical6> db.users.find()
[
  {
    _id: ObjectId('67acea1b0f9d5426634d7942'),
    name: 'Rehman',
    roll_no: 10,
    age: 21
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7943'),
    name: 'abc',
    roll_no: 18,
    age: 22
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7944'),
    name: 'xyz',
    roll_no: 21,
    age: 23,
    address: 'Mumbai'
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7945'),
    name: 'efg',
    roll_no: 12,
    address: 'Delhi'
  }
]
```

18. Now switch to your secondary server

use practical6

db.users.find()

All the changes we made are reflected in the secondary server

First secondary server

Second Secondary server:

```
mongosh mongodb://localhost:2727/?directConnection=true&serverSelectionTimeoutMS=2000
test-replica-set [direct: secondary] practical> use practical6
switched to db practical6
test-replica-set [direct: secondary] practical6> db.users.find()
[
  {
    _id: ObjectId('67acea1b0f9d5426634d7945'),
    name: 'efg',
    roll_no: 12,
    address: 'Delhi'
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7942'),
    name: 'Rehman',
    roll_no: 10,
    age: 21
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7943'),
    name: 'abc',
    roll_no: 18,
    age: 22
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7944'),
    name: 'xyz',
    roll_no: 21,
    age: 23,
    address: 'Mumbai'
  }
]
```

```
mongosh mongodb://localhost:2737/?directConnection=true&serverSelectionTimeoutMS=2000
test-replica-set [direct: secondary] practical> use practical6
switched to db practical6
test-replica-set [direct: secondary] practical6> db.users.find()
[
  {
    _id: ObjectId('67acea1b0f9d5426634d7945'),
    name: 'efg',
    roll_no: 12,
    address: 'Delhi'
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7942'),
    name: 'Rehman',
    roll_no: 10,
    age: 21
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7943'),
    name: 'abc',
    roll_no: 18,
    age: 22
  },
  {
    _id: ObjectId('67acea1b0f9d5426634d7944'),
    name: 'xyz',
    roll_no: 21,
    age: 23,
    address: 'Mumbai'
  }
]
test-replica-set [direct: secondary] practical6>
```

19. Performing more CRUD Operations

a. Insert a single record

db.users.insertOne({ name: "Nina", roll_no: 19, age: 20 });

```
test-replica-set [direct: primary] practical6>
{
  acknowledged: true,
  insertedId: ObjectId('67aceb800f9d5426634d7949')
}
```

b. Insert Multiple records

db.users.insertMany([{ name: "Sana", roll_no: 20, age: 22 }, { name: "Emad", roll_no: 21, age: 24 }]);

```
test-replica-set [direct: primary] practical6> db.users.insertMany([{ name: "Sana", roll_no: 20, age: 22 }, { name:
", roll_no: 21, age: 24 }]);
{ acknowledged: true,
  acknowledged: true, d('67aceb800f9d5426634d7949')
  insertedIds: {
    '0': ObjectId('67aceb9c0f9d5426634d794a'),
    '1': ObjectId('67aceb9c0f9d5426634d794b')
  }
}
```

c. Find record where name = "Rehmah"

db.users.find({ name: "Rehmah" }).pretty();

```
test-replica-set [direct: primary] practical6> db.users.find({ name: "Rehmah" }).pretty();
[
  {
    _id: ObjectId('67acea1b0f9d5426634d7942'),
    name: 'Rehmah',
    roll_no: 10,
    age: 21
  }
]
```

d. Update a document

db.users.updateOne({ name: "xyz" }, { \$set: { age: 23 } });

```
test-replica-set [direct: primary] practical6> db.users.updateOne( { name: "xyz" }, { $set: { age: 23 } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
```


g. Delete multiple records

```
db.users.deleteMany({ age: { $gt: 22 } });
```

```
test-replica-set [direct: primary] practical6> db.users.deleteMany({ age: { $gt: 22 } });  
{ acknowledged: true, deletedCount: 2 }
```

All the changes we made are reflected in the secondary server

First secondary server

| Second Secondary server:

```
mongosh mongodb://localhost:2727/?directConnection=true&serverSelectionT...  
test-replica-set [direct: secondary] practical6> db.users.find()  
[  
  {  
    _id: ObjectId('67acea1b0f9d5426634d7942'),  
    name: 'Rehman',  
    roll_no: 10,  
    age: 21,  
    status: 'young'  
  },  
  {  
    _id: ObjectId('67acea1b0f9d5426634d7943'),  
    name: 'abc',  
    roll_no: 18,  
    age: 22,  
    status: 'young'  
  },  
  {  
    _id: ObjectId('67aceb370f9d5426634d7946'),  
    name: 'Nina',  
    roll_no: 19,  
    age: 20,  
    status: 'young'  
  },  
]
```

```
mongosh mongodb://localhost:2737/?directConnection=true&serverSelectionT...  
test-replica-set [direct: secondary] practical6> db.users.find()  
[  
  {  
    _id: ObjectId('67acea1b0f9d5426634d7942'),  
    name: 'Rehman',  
    roll_no: 10,  
    age: 21,  
    status: 'young'  
  },  
  {  
    _id: ObjectId('67acea1b0f9d5426634d7943'),  
    name: 'abc',  
    roll_no: 18,  
    age: 22,  
    status: 'young'  
  },  
  {  
    _id: ObjectId('67aceb370f9d5426634d7946'),  
    name: 'Nina',  
    roll_no: 19,  
    age: 20,  
    status: 'young'  
  },  
  {  
    _id: ObjectId('67aceb680f9d5426634d7947'),  
    name: 'Nina',  
    roll_no: 19,  
    age: 20,  
    status: 'young'  
  },  
]
```