

Creating MongoDB Replica Set

MacOS / Linux(only dbpath changes)

1. Setting Up MongoDB Data Directory

- The location of the MongoDB data directory varies based on your Apple Processor:

	Intel Processor	Apple Silicon Processor
configuration file	<code>/usr/local/etc/mongod.conf</code>	<code>/opt/homebrew/etc/mongod.conf</code>
log directory	<code>/usr/local/var/log/mongodb</code>	<code>/opt/homebrew/var/log/mongodb</code>
data directory	<code>/usr/local/var/mongodb</code>	<code>/opt/homebrew/var/mongodb</code>

- For Intel Processor: The path is `/usr/local/var/mongodb`.
- For Apple Silicon Processor: The path is `/opt/homebrew/var/mongodb`.

2. Preparing for Replica Set Configuration

Before creating the replica set, you need to close the standalone instance of your running MongoDB server. Follow these steps:

- Start the mongoshell server.

```
echo — mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelection...
echo@viveks-MacBook-Air ~ % mongosh
```

- To close the running instance, enter `db.shutdownServer()` and press Enter.

```
-----
test> db.shutdownServer()
```

- You'll receive a confirmation message.

```
-----
[test> db.shutdownServer()
MongoNetworkError: connection 2 to 127.0.0.1:27017 closed
test>
```

- Type `exit` and press Enter to exit the mongosh shell.

```
-----
[test> exit
echo@viveks-MacBook-Air ~ %
```

3. Starting the Replica Set Server

1. To start the replica set, use the following command:
 - For Apple Silicon Processor: `mongod --replSet rs0 --dbpath /opt/homebrew/var/mongod`
 - For Intel Processor: `mongod --replSet rs0 --dbpath /usr/local/var/mongod`

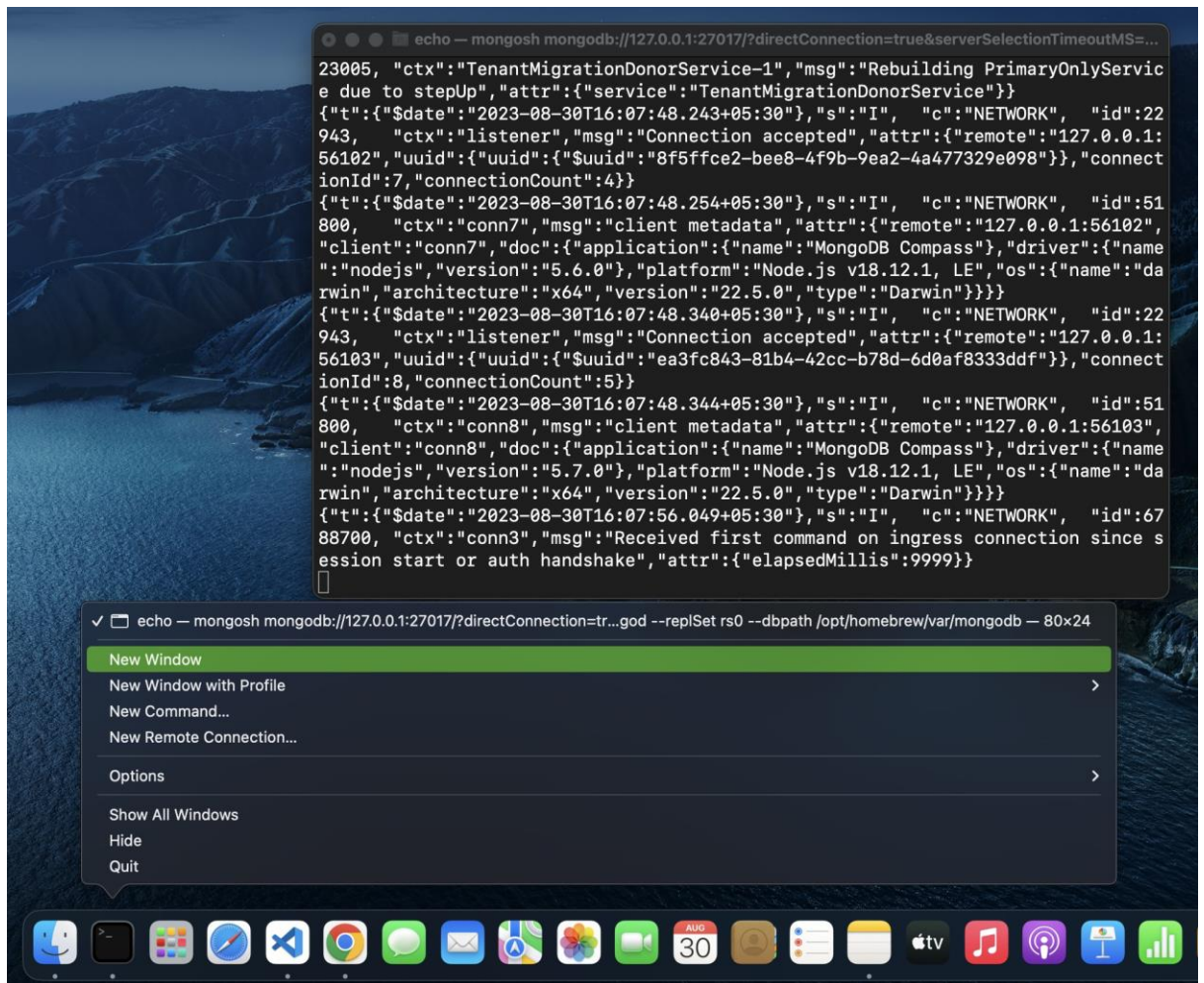
```
echo — mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=...
echo@viveks-MacBook-Air ~ % mongod --replSet rs0 --dbpath /opt/homebrew/var/mong
odb
```

2. Press Enter to start the server for the replica set.

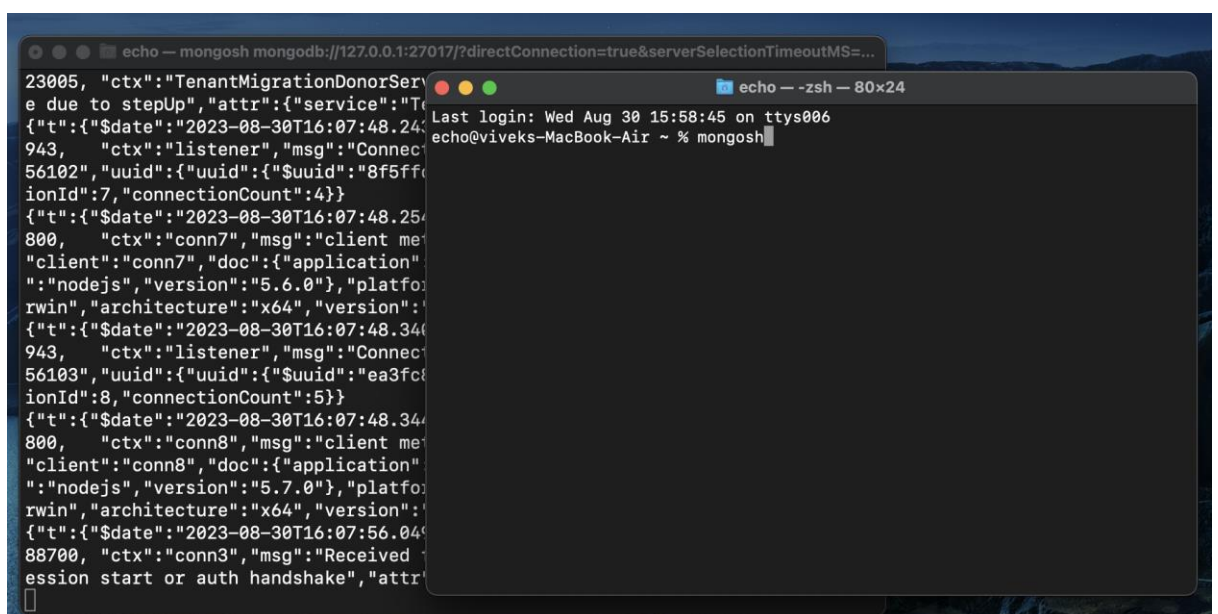
```
echo — mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=...
23005, "ctx":"TenantMigrationDonorService-1","msg":"Rebuilding PrimaryOnlyService due to stepUp","attr":{"service":"TenantMigrationDonorService"}}
{"t":{"$date":"2023-08-30T16:07:48.243+05:30"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:56102","uuid":{"uuid":{"$uuid":"8f5ffce2-bee8-4f9b-9ea2-4a477329e098"}}},"connectionId":7,"connectionCount":4}}
{"t":{"$date":"2023-08-30T16:07:48.254+05:30"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn7","msg":"client metadata","attr":{"remote":"127.0.0.1:56102","client":"conn7","doc":{"application":{"name":"MongoDB Compass"},"driver":{"name":"nodejs","version":"5.6.0"},"platform":"Node.js v18.12.1, LE","os":{"name":"darwin","architecture":"x64","version":"22.5.0","type":"Darwin"}}}}
{"t":{"$date":"2023-08-30T16:07:48.340+05:30"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:56103","uuid":{"uuid":{"$uuid":"ea3fc843-81b4-42cc-b78d-6d0af8333ddf"}}},"connectionId":8,"connectionCount":5}}
{"t":{"$date":"2023-08-30T16:07:48.344+05:30"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn8","msg":"client metadata","attr":{"remote":"127.0.0.1:56103","client":"conn8","doc":{"application":{"name":"MongoDB Compass"},"driver":{"name":"nodejs","version":"5.7.0"},"platform":"Node.js v18.12.1, LE","os":{"name":"darwin","architecture":"x64","version":"22.5.0","type":"Darwin"}}}}
{"t":{"$date":"2023-08-30T16:07:56.049+05:30"},"s":"I", "c":"NETWORK", "id":6788700, "ctx":"conn3","msg":"Received first command on ingress connection since session start or auth handshake","attr":{"elapsedMillis":9999}}
```

4. Initializing the Replica Set

- Keep the previous terminal with the running server open, and open a new terminal.



- In the new terminal, enter *[mongosh](#)* and press Enter.



- The mongoshell will start. Enter `rs.initiate()` and press Enter.

```
test> rs.initiate()
```

- The 'rs0' instance of the replica set will begin initialization.

```
echo — mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelection...
Last login: Wed Aug 30 15:58:45 on ttys006
echo@viveks-MacBook-Air ~ % mongosh
Current Mongosh Log ID: 64ef1c7ece4a14c9f307ec9c
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverS
electionTimeoutMS=2000&appName=mongosh+1.10.6
Using MongoDB:      7.0.0
Using Mongosh:      1.10.6

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

-----
The server generated these startup warnings when booting
  2023-08-30T16:07:45.878+05:30: Access control is not enabled for the database
. Read and write access to data and configuration is unrestricted
  2023-08-30T16:07:45.879+05:30: This server is bound to localhost. Remote syst
ems will be unable to connect to this server. Start the server with --bind_ip <a
ddress> 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 se
rver with --bind_ip 127.0.0.1 to disable this warning
  2023-08-30T16:07:45.879+05:30: Soft rlimits for open file descriptors too low
-----

rs0 [direct: primary] test> 
```

5. Note:

- If you encounter any errors during the process, consider uninstalling MongoDB, then reinstall and follow the steps outlined above.
- Make sure to remove all existing MongoDB-related files, including configuration and data directory files, after uninstallation.

6. Closing the MongoDB Replica Set

- To close the running replica set instance, enter `db.shutdownServer()` in the mongo shell and press Enter.

```
rs0 [direct: primary] test> db.shutdownServer()
```

- Exit the mongoshell.

```
rs0 [direct: primary] test> db.shutdownServer()  
MongoNetworkError: connection 2 to 127.0.0.1:27017 closed  
test> exit
```

Starting the Standalone MongoDB Server

- Start the standalone MongoDB server using the command `mongod --dbpath /opt/homebrew/var/mongodb/`.

```
echo — mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelection...  
echo@viveks-MacBook-Air ~ % mongod --dbpath /opt/homebrew/var/mongodb/
```

- Open a new terminal window while keeping the previous mongosh shell open.

```
echo — mongosh mongodb://127.0.0.1:27017/?directConnection=t...
{ "t": { "$date": "2023-08-30T18:58:33.201+05:30" }, "s": "WORK", "id": 22944, "ctx": "conn6", "msg": "Conne", "attr": { "remote": "127.0.0.1:58278", "uuid": { "uid": { "bb-8391-4149-b33a-ad7c2dc0b943" }, "connectionId": 1 }, "connectionCount": 5 } } }
{ "t": { "$date": "2023-08-30T18:58:33.201+05:30" }, "s": "WORK", "id": 22944, "ctx": "conn4", "msg": "Conne", "attr": { "remote": "127.0.0.1:58276", "uuid": { "uid": { "72-504f-4aea-87cb-379cd79b12e9" }, "connectionId": 1 }, "connectionCount": 3 } } }
{ "t": { "$date": "2023-08-30T18:58:34.051+05:30" }, "s": "WORK", "id": 4615610, "ctx": "conn3", "msg": "Failed to connect to the server", "attr": { "error": { "code": 6, "codeName": "NotWritable", "errmsg": "peekASIOStream :: caused by :: read by peer" } }, "connectionCount": 3 } } }
{ "t": { "$date": "2023-08-30T18:58:34.052+05:30" }, "s": "WORK", "id": 20883, "ctx": "conn3", "msg": "Interrupted as its client disconnected", "attr": { "opId": 191 }, "connectionCount": 2 } } }
{ "t": { "$date": "2023-08-30T18:58:34.055+05:30" }, "s": "WORK", "id": 22944, "ctx": "conn3", "msg": "Conne", "attr": { "remote": "127.0.0.1:58275", "uuid": { "uid": { "3e-a0f1-4adc-ba17-d5c79bdea0ea" }, "connectionId": 3 }, "connectionCount": 2 } } }
[ ]
```

- You will now be in the standalone server context.

```
echo — mongosh mongodb://127.0.0.1:27017/?directConnection=t...
cations": { "name": "mongosh 1.10.6", "driver": { "name": "mongodb", "version": "5.8.1|1.10.6", "platform": "Node.js", "os": { "name": "darwin", "architecture": "arm64", "version": "Darwin" } } } }
{ "t": { "$date": "2023-08-30T19:01:12.686+05:30" }, "s": "WORK", "id": 6788700, "ctx": "conn10", "msg": "Received on ingress connection since session start or", "attr": { "elapsedMillis": 0 } } }
{ "t": { "$date": "2023-08-30T19:01:12.752+05:30" }, "s": "WORK", "id": 6788700, "ctx": "conn9", "msg": "Received on ingress connection since session start or", "attr": { "elapsedMillis": 68 } } }
{ "t": { "$date": "2023-08-30T19:01:23.185+05:30" }, "s": "WORK", "id": 22943, "ctx": "listener", "msg": "Connected", "attr": { "remote": "127.0.0.1:58349", "uuid": { "uid": { "a41a2da5-48a1-4d71-aec4-0af7a6a7454c" }, "connectionId": 7 }, "connectionCount": 7 } } }
{ "t": { "$date": "2023-08-30T19:01:23.187+05:30" }, "s": "WORK", "id": 51800, "ctx": "conn11", "msg": "client connection", "attr": { "remote": "127.0.0.1:58349", "client": "conn11", "cations": { "name": "mongosh 1.10.6", "driver": { "name": "mongodb", "version": "5.8.1|1.10.6", "platform": "Node.js", "os": { "name": "darwin", "architecture": "arm64", "version": "Darwin" } } } } }
[ ]
```

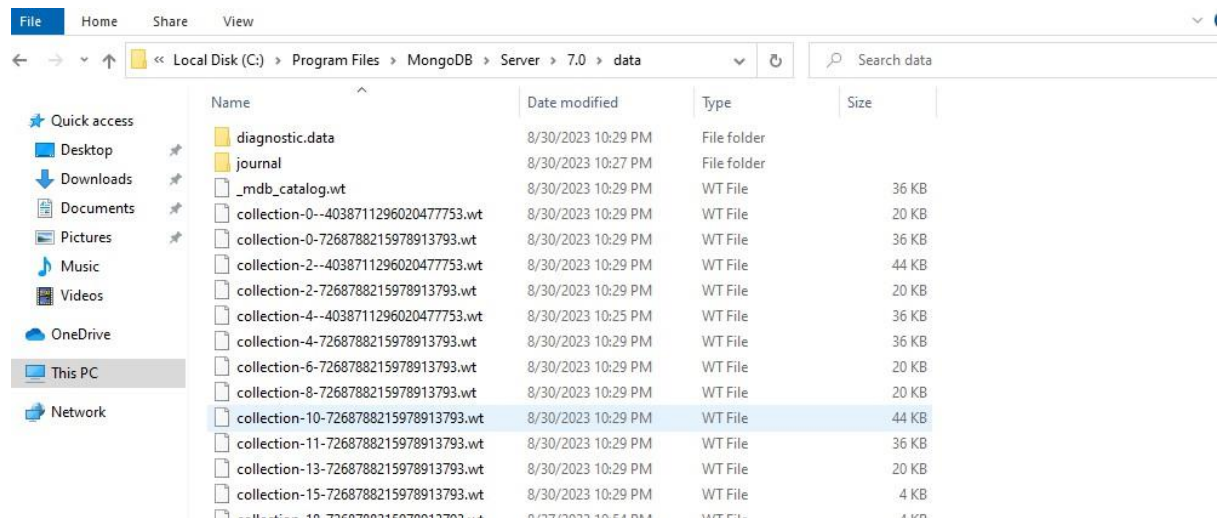
```
echo — mongosh mongodb://127.0.0.1:27017/?directConne
-----
The server generated these startup warnings when booting
2023-08-30T18:58:11.870+05:30: Access control is not enabled
- Read and write access to data and configuration is unrestricted
2023-08-30T18:58:11.870+05:30: This server is bound to localhost
ems will be unable to connect to this server. Start the server with
ddress> to specify which IP addresses it should serve responses
bind_ip_all to bind to all interfaces. If this behavior is desired
rver with --bind_ip 127.0.0.1 to disable this warning
2023-08-30T18:58:11.870+05:30: Soft rlimits for open file descriptors
2023-08-30T18:58:11.885+05:30: Document(s) exist in 'system.repl
rted without --replSet. Database contents may appear inconsistent
s that were visible when this node was running as part of a repl
with --replSet unless you are doing maintenance and no other cli
ted. The TTL collection monitor will not start because of this.
e http://dochub.mongodb.org/core/ttlcollections
2023-08-30T18:58:11.885+05:30: Replica set member is in standalone
orming any writes will result in them being untimestamped. If a v
xisting document, the document's history will be overwritten with
since the beginning of time. This can break snapshot isolation w
e engine.
-----
test> [ ]
```

Windows

1. Setting Up MongoDB Data Directory

- The "data directory" specifically refers to the location where MongoDB stores its data.

In Windows, the location might vary, but you need to get to this “**data**” directory and note down the path.



2. Setting Up Replication:

- Firstly, open the command prompt as **Administrator**.


```
C:\> Administrator: Command Prompt

Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

- Get inside the Mongo shell by using the command: **mongosh**
- Then we need to stop the running server by using the command:
db.shutdownServer()

```
C:\> mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeout

Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>mongosh
Current Mongosh Log ID: 64f0241c164ee77f2405953f
Connecting to:      mongodb://127.0.0.1:27017/?directConn
.10.6
Using MongoDB:      7.0.0
Using Mongosh:      1.10.6

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

-----
The server generated these startup warnings when booting
2023-08-30T22:20:10.682-07:00: Access control is not enabled
figuration is unrestricted
2023-08-30T22:20:11.497-07:00: Document(s) exist in 'system
s may appear inconsistent with the writes that were visible w
with --replSet unless you are doing maintenance and no other
start because of this. For more info see http://dochub.mongod
2023-08-30T22:20:11.497-07:00: Replica set member is in sta
ing untimestamped. If a write is to an existing document, the
since the beginning of time. This can break snapshot isolation
-----

test> db.shutdownServer()
```

- Now exit the Mongo shell using the command: **exit**

- You can now start the new “mongod” relocation server using the below command:

`mongod --replSet rs0 --dbpath "C:\Program Files\MongoDB\Server\7.0\data"`

Note: dbpath in the above command might vary. “**rs0**” is the replica set name, and dbpath is the location where replica data is going to be. **DON'T**

CLOSE THIS RUNNING COMMAND LINE!!!

```
test> db.shutdownServer()  
MongoNetworkError: connection 2 to 127.0.0.1:27017 closed  
test> exit  
  
C:\Windows\system32>mongod --replSet rs0 --dbpath "C:\Program Files\MongoDB\Server\7.0\data"
```

- Now, your replica server is running. Now you can open in a new Mongo shell and start using rs0 by typing in the command:

`rs.initiate()`

- You should reach the original state now.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>net stop MongoDB
The MongoDB Server (MongoDB) service is not started.

More help is available by typing NET HELPMSG 3521.

C:\Windows\system32>net start MongoDB
The MongoDB Server (MongoDB) service is starting.
The MongoDB Server (MongoDB) service was started successfully.

C:\Windows\system32>mongosh
Current Mongosh Log ID: 64f02668c9bd66c6563c0edd
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.6
Using MongoDB:      7.0.0
Using Mongosh:      1.10.6

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

-----
The server generated these startup warnings when booting
2023-08-30T22:34:24.265-07:00: Access control is not enabled for the database. Read and write access to data and configura
tion is unrestricted
2023-08-30T22:34:24.851-07:00: Document(s) exist in 'system.replset', but started without --replSet. Database contents may
appear inconsistent with the writes that were visible when this node was running as part of a replica set. Restart with --re
plSet unless you are doing maintenance and no other clients are connected. The TTL collection monitor will not start because
of this. For more info see http://dochub.mongodb.org/core/ttlcollections
2023-08-30T22:34:24.851-07:00: Replica set member is in standalone mode. Performing any writes will result in them being u
ntimestamped. If a write is to an existing document, the document's history will be overwritten with the new value since the
beginning of time. This can break snapshot isolation within the storage engine.
-----

test>
```