How to create replica set in mongodb version 8.2

Let's go step-by-step to create a **replica set in MongoDB 8.2** — the latest version as of 2025.

A **replica set** in MongoDB is a group of `mongod` instances that maintain the same data set, providing **redundancy** and **high availability**.

Step-by-Step Setup for MongoDB 8.2 Replica Set

We'll use **three nodes** (members) for demonstration:

```
Node 1: localhost:27017 (Primary)
Node 2: localhost:27018 (Secondary)
Node 3: localhost:27019 (Secondary)
```

Step 1: Create Data Directories

On your system, create separate data folders for each instance:

```
mkdir C:\data\r1
mkdir C:\data\r2
mkdir C:\data\r3
```

Step 2: Start MongoDB Instances

Open **three separate terminals** (or command prompts).

Terminal 1 (Primary)

```
mongod --port 27017 --dbpath "C:\data\r1" --replSet "rs0"
```

Terminal 2 (Secondary 1)

```
mongod --port 27018 --dbpath "C:\data\r2" --replSet "rs0"
```

Terminal 3 (Secondary 2)

```
mongod --port 27019 --dbpath "C:\data\r3" --replSet "rs0"
```

All must use the **same replica set name** (`rs0` here).

Step 3: Connect to Mongo Shell (mongosh)

Open a new terminal and connect to the first node:

```
mongosh --port 27017
```

Step 4: Initialize the Replica Set

In the Mongo shell, run:

```
rs.initiate({
  _id: "rs0",
  members: [
    { _id: 0, host: "localhost:27017" },
    { _id: 1, host: "localhost:27018" },
    { _id: 2, host: "localhost:27019" }
  ]
})
```

Step 5: Verify Replica Set Status

Run:

```
rs.status()
```

 You should see:

- One **PRIMARY**
- Two **SECONDARY** nodes

Step 6: Test Replication

- Connect to the **Primary** (port 27017).

  Mongosh - -port 27017

- Create a database and insert data:

  use testdb

  db.items.insertOne({ name: "replica test", time: new Date() })

- Connect to a **Secondary**:

  ```
  mongosh --port 27018
  ```

- Allow reading from secondary:

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

You'll see the same document replicated □

Optional: Run as Windows Services (MongoDB 8.2+)

If you want to run each instance as a **Windows service**:

1. Create 3 config files:
    - C:\Program Files\MongoDB\Server\8.2\bin\mongod1.cfg
    - C:\Program Files\MongoDB\Server\8.2\bin\mongod2.cfg
    - C:\Program Files\MongoDB\Server\8.2\bin\mongod3.cfg

Example config for node1:

```
systemLog:
  destination: file
  path: "C:\\data\\r1\\mongod.log"
storage:
  dbPath: "C:\\data\\r1"
net:
  bindIp: 127.0.0.1
  port: 27017
replication:
  replSetName: rs0
```

Then install each as a service:

```
mongod --config "C:\Program Files\MongoDB\Server\8.2\bin\mongod1.cfg" --
install --serviceName "MongoDB1"
mongod --config "C:\Program Files\MongoDB\Server\8.2\bin\mongod2.cfg" --
install --serviceName "MongoDB2"
mongod --config "C:\Program Files\MongoDB\Server\8.2\bin\mongod3.cfg" --
install --serviceName "MongoDB3"
```

Then start:

```
net start MongoDB1
net start MongoDB2
net start MongoDB3
```

□ Done!

You now have a **MongoDB 8.2 Replica Set** running locally with:

- Automatic data replication
- High availability
- Failover support
