**MONGODB 3 NODE REPLICATION**

The following steps need to be taken for setting up 3 node replications for MongoDB:

Step 1: Launch 3 separate servers as 3 nodes of MongoDB.

Step 2: Open port 27017 globally in the inbound traffic for all the servers.

Step 3: Open port 27017 in firewall-cmd with the following command:

**sudo firewall-cmd --zone=public --permanent --add-port=27017/tcp**

**sudo firewall-cmd --reload**

Step 4: Install MongoDB on three servers by following the steps outlined below:

1. **curl -fsSL https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add –**
2. **apt-key list**
3. **echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list**
4. **sudo apt update**
5. **sudo apt install mongodb-org**

Step 5: Navigate to the /etc directory.

**sudo nano /etc/mongod.conf**

Step 6: Replace the content of mongod.conf with the following content.

**storage:**

**dbPath: /var/lib/mongodb**

**net:**

**bindIp: 0.0.0.0**

**port: 27017**

**systemLog:**

**destination: file**

**path: /var/log/mongodb/mongod.log**

**logAppend: true**

**replication:**

**replSetName: <replicaset-name>**

Step 7: Start the MongoDB server.

**sudo systemctl start mongod.service**

Step 8: Verify the status of the MongoDB server.

**sudo systemctl status mongod**

Step 9: Enable MongoDB service to start up the boot.

**sudo systemctl enable mongod**

Step 10: Connect the mongo shell with the public ip and the port in the 1st node with the following command.

**mongo --host <public-ip> --port 27017**

Step 11: Run the following commands to initiate the replica set.

**rs.initiate(**

**{**

**\_id: "<replicaset-name>",**

**members: [**

**{ \_id: 0, host: "<public-ip>:27017" },**

**{ \_id: 1, host: "<public-ip>:27017" },**

**{ \_id: 2, host: "<public-ip>:27017" }**

**]**

**}**

**);**

Step 12: Verify the replica set status.

**rs.status()**

Step 13: Connect to 2nd MongoDB node.

**mongo --host <public-ip> --port 27017**

Step 14: Verify the replica set status.

rs.status()

Step 15: Repeat step 13 and step 14 for the 3rd MongoDB node.

Once the replica set is created for all the 3 nodes, we need to create the respective database with username and password and give them the appropriate read and write permissions in the primary node. It will be replicated in all the other nodes automatically.

Step 16: Connect the primary node with host and port.

**mongo --host <public-ip> --port 27017**

Step 17: Create the database with the user and password and give them the read and write access with the following command.

**use <database-name>**

**db.createUser(**

**{**

**user: "<username>",**

**pwd: "<password>",**

**roles: [ { role: "readWrite", db: "<database-name>" } ]**

**}**

**)**

Step 18: Create the collections in the database.

Step 19: Exit from the mongo shell with the **exit** command**.**

Step 20: Connect the primary node again with host port username and password.

**mongo --host "<public-ip>:<port>" --username <username> --password <pwd>**

Step 21: Check the databases that have been createdwith the following command.

**show dbs;**

Step 22**:** Connect the secondary nodes with the same username and password.

**mongo --host "<public-ip>:<port>" --username <username> --password <pwd>**

Step 23: Check if the database created in the primary node has been replicated to all the secondary nodes.

**rs.secondaryOk()**

**show dbs;**

Step 24: If all the database created in primary nodes are replicated to both the secondary nodes then the MongoDB 3 node replication is successful.