MongoDB

A concise guide to installing, configuring, and managing MongoDB on Ubuntu. Learn key commands, user management, and security best practices for efficient MongoDB deployment.

Install MongoDB on Ubuntu 22.04

This guide provides step-by-step instructions for installing MongoDB on Ubuntu. MongoDB is a NoSQL database that offers high performance, high availability, and easy scalability.

Prerequisites

- A system running Ubuntu 22.04
- A user account with sudo privileges
- An internet connection

Get Started

- Begin by updating your apt repository to ensure you have the latest package information and dependencies. Open a terminal and run the following command:

```
sudo apt update
```

This command will refresh the package lists and update any outdated packages on your system.

- Next, you need to install the necessary packages for MongoDB. These include `gnupg` for handling GPG keys and `curl` for downloading files. Execute the following command in the terminal:

```
sudo apt install gnupg curl
```

This command will download and install the required packages from the Ubuntu repositories.

- Add the MongoDB GPG key to your system to verify the authenticity of the MongoDB packages. Run the following command in the terminal:

```
curl -fsSL https://pgp.mongodb.com/server-7.0.asc | sudo apt-key add -
```

This command fetches the MongoDB GPG key and adds it to your list of trusted keys.

- Add the MongoDB repository to your system’s package source list. Execute the following command:

```
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu $(lsb_release -cs) mongodb-org 7.0" | sudo tee
/etc/apt/sources.list.d/mongodb-org-7.0.list
```

This command adds the MongoDB repository to your apt sources list, allowing you to install MongoDB from there.

- Update your package list to include the MongoDB packages from the newly added repository:

```
sudo apt update
```

- Install MongoDB by running the following command:

```
sudo apt install mongodb-org
```

This command will download and install MongoDB along with its dependencies.

- Once the installation is complete, start the MongoDB service and enable it to start on boot:

```
sudo systemctl start mongod
sudo systemctl enable mongod
```

- Verify that MongoDB is running correctly by checking its status:

```
sudo systemctl status mongod
```

You should see an output indicating that MongoDB is active and running.

- Optionally, you can check the MongoDB version installed by running:

```
mongod --version
```

This command displays the version of MongoDB installed on your system.

Finished

Congratulations! You have now installed MongoDB on your Ubuntu machine. MongoDB is ready to be used, allowing you to leverage its powerful NoSQL database capabilities for your projects and applications.

Create Users in MongoDB on Ubuntu 22.04

Prerequisites

- MongoDB installed on your system
- MongoDB service running
- Access to the MongoDB shell (`mongosh` or `mongo`)

Step 1: Connect to the MongoDB Shell

First, connect to your MongoDB shell by running:

```
mongosh
```

Or, if using the older shell version:

```
mongo
```

This will open the MongoDB shell where you can run commands.

Step 2: Switch to the Admin Database

To manage users, you need to switch to the `admin` database:

```
use admin
```

Step 3: Create a Super (Root) User

To create a user with full administrative privileges (root), use the following command:

```
db.createUser({
  user: "admin_user",
  pwd: "strong_password_here",
  roles: [ { role: "root", db: "admin" } ]
})
```

Replace `admin_user` with the desired username and `strong_password_here` with a strong password. This user will have full control over the entire MongoDB instance.

Step 4: Create a Normal User

Next, create a normal user with read and write permissions on a specific database. First, switch to the database where this user will operate:

```
use my_database
```

Then, create the user with the following command:

```
db.createUser({
  user: "normal_user",
  pwd: "normal_password_here",
  roles: [ { role: "readWrite", db: "my_database" } ]
})
```

Replace `normal_user` with the desired username, `normal_password_here` with the password, and `my_database` with the name of the database where this user will have access.

Step 5: Enable Authentication (Optional, but Recommended)

If you want to enforce user authentication, you need to enable it in the MongoDB configuration. To do this:

1. Open the MongoDB configuration file, typically located at `/etc/mongod.conf`:

   ```
   sudo nano /etc/mongod.conf
   ```

2. Find the following line and ensure `security` is uncommented and `authorization` is set to `enabled`:

   ```
   security:
     authorization: enabled
   ```

3. Save the file and restart MongoDB:

   ```
   sudo systemctl restart mongod
   ```

Step 6: Log in with the New User

To log in as the newly created users, you can use the following commands:

- For the root user:

  ```
  mongosh -u "admin_user" -p "strong_password_here" --authenticationDatabase "admin"
  ```

- For the normal user:

  ```
  mongosh -u "normal_user" -p "normal_password_here" --authenticationDatabase "my_database"
  ```

Finished

You have now successfully created both a root user and a normal user in MongoDB. The root user has full privileges, while the normal user has restricted access to a specific database.