

[How-To] Install MariaDB on Ubuntu 24.04 LTS

Purpose

The purpose of this document is to show explain in full detail the steps to take a freshly installed vm with Ubuntu 24.04 LTS as the operating system and install/configure MariaDB for use. It will not dive into clustering or changing of binding for external use. Those will be handled in their own How-To articles.

Prerequisites

List of prerequisites:

- Sudo user
- Ubuntu 24.04 LTS VM

Instructions

Step 1: Update the System

For the first step, we need to update the Ubuntu system OS:

```
sudo apt update && sudo apt upgrade -y
```

Step 2: Install Dependent Software

Next, we'll need some additional software packages to be able to continue with configuration:

```
sudo apt install -y software-properties-common
```

Step 3: Configure Hostname

Next, we'll configure the hostname of the server:

```
sudo hostnamectl set-hostname db-node1
```

Step 4: Install MariaDB

Now that we have our base system configured, updated, and prepped, we can move on to installing the MariaDB Server on the VM. We'll start by adding the MariaDB APT repository:

```
sudo apt install -y curl gnupg
```

```
sudo curl -Ls https://mariadb.org/mariadb_release_signing_key.asc | sudo gpg --dearmor -o  
/usr/share/keyrings/mariadb-keyring.gpg
```

```
sudo echo "deb [signed-by=/usr/share/keyrings/mariadb-keyring.gpg]  
https://mirror.mariadb.org/repo/10.11/ubuntu lunar main" | sudo tee /etc/apt/sources.list.d/mariadb.list
```

```
sudo apt update
```

Now that we have the repository, we can install MariaDB:

```
sudo apt install -y mariadb-server mariadb-client
```

At this point, you have an operational MariaDB Server. We need to do some additional steps to lock it down and configure it for the first time.

Step 5: First Time Installation

Now that we have our MariaDB Server installed, we need to set it up. Run the following command:

```
sudo mysql_secure_installation
```

This will take you into the start up installation for MariaDB. Follow the settings below to complete setup:

- **Enter root password:** Since one has not been set you, don't type a password and press enter.
- **Switch to unix_socket authentication?:** No, we don't need this.
- **Set root password?:** This is where you can set a strong password for the `root` user.
- **Remove anonymous users?:** Yes, it's best to remove them for security.
- **Disallow root login remotely?:** Yes, you should disallow remote login for the root user to prevent remote access.
- **Remove test database and access to it?:** Yes, it's a good idea to remove the test database, as it's not needed.
- **Reload privilege tables now?:** Yes, reload the privilege tables to apply all changes.

Once you are complete, try logging into MariaDB from terminal with the root MariaDB user you just set the password for:

```
sudo mariadb -u root -p
```

It will prompt for password, enter the one you set during installation. Then, try a command:

```
SHOW DATABASES;
```

```
QUIT;
```

There should be 3 or 4 system level databases already there for you to see.

Step 6: Configure UFW Firewall

This step is a security step and not needed to make your MariaDB server functional. But, it is always better to setup at the start and allow needed ports through as needed instead of leaving it open or trying to lock down later. Run the following command to install UFW:

```
sudo apt install ufw -y
```

Once installed, start by allowing ssh:

```
sudo ufw allow ssh
```

With SSH allowed, you can enable the firewall:

```
sudo ufw enable
```

It will warn you that you could get disconnected, continue as we know we just allowed ssh. Now, add our other rule for MariaDB:

```
sudo ufw allow 3306/tcp
```

```
sudo ufw reload
```

With this step completed, you have a secure and operational standalone MariaDB Server.

Revision #6

Created 6 December 2024 05:23:43 by Mike Leffring

Updated 6 December 2024 06:58:36 by Mike Leffring