# Installing MariaDB on Ubuntu 18.04 from MariaDB Repositories

-At the time of writing of this article the latest version of MariaDB available from the official MariaDB repositories is MariaDB version 10.3.

-Before continuing with the next step you should visit the MariaDB Repository page and check if there is a new version available.

-To install the latest stable version of MariaDB on your Ubuntu 18.04 from the MariaDB repositories, follow these steps:

## 1. Prerequisites

-Before continuing with this tutorial, make sure you are logged in as a user with sudo privileges.

## 2. Enable the MariaDB repository.

-First add the MariaDB GPG key to your system using the following command:

```
$ sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8
```

-Once the key is imported, add the MariaDB repository with:

```
$ sudo add-apt-repository 'deb [arch=amd64,arm64,ppc64el] http://ftp.utexas.edu/mariadb/repo/10.3/ubuntu bionic main'
```

-If you get an error message saying add-apt-repository command not found then you will need to install the software-properties-common package.

-To be able to install packages from the MariaDB repository you’ll need to update the packages list:

```
$ sudo apt update
```

## 3. Install MariaDB

-Now that the repository is added to install the MariaDB package simply type:

```
$ sudo apt install mariadb-server
```

## 4. Check MariaDB status

-The MariaDB service will start automatically, to verify it type:

```
$ sudo systemctl status mariadb

● mariadb.service - MariaDB 10.3.8 database server
Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
Drop-In: /etc/systemd/system/mariadb.service.d
        └─migrated-from-my.cnf-settings.conf
Active: active (running) since Sun 2018-07-29 19:36:30 UTC; 56s ago
   Docs: man:mysqld(8)
        https://mariadb.com/kb/en/library/systemd/
Main PID: 16417 (mysqld)
Status: "Taking your SQL requests now..."
   Tasks: 31 (limit: 507)
CGroup: /system.slice/mariadb.service
        └─16417 /usr/sbin/mysqld
```

-And print the MariaDB server version, with:

```
$ mysql -V

   mysql  Ver 15.1 Distrib 10.3.8-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2
```

## 5. Securing MariaDB

-Run the mysql_secure_installation command to improve the security of the MariaDB installation:

```
$ sudo mysql_secure_installation
```

-The script will prompt you to set up the root user password, remove the anonymous user, restrict root user access to the local machine and remove the test database.

-At the end the script will reload the privilege tables ensuring that all changes take effect immediately.

-All steps are explained in detail and it is recommended to answer “Y” (yes) to all questions.

```
--------------------------------------------------------------------------------------------------------
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
  SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
 ... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] n
 ... skipping.

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

6. Connect to MariaDB from the command line

-To connect to the MariaDB server through the terminal we can use the MariaDB client.

-To log in to the MariaDB server as the root user type:

```
mysql -u root -p
```

-You will be prompted to enter the root password you have previously set when the mysql_secure_installation script was run.
-Once you enter the password you will be presented with the MariaDB shell as shown below:

```
   Welcome to the MariaDB monitor.  Commands end with ; or \g.
   Your MariaDB connection id is 49
   Server version: 10.1.29-MariaDB-6 Ubuntu 18.04

   Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

   Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
-------------------------------------------------------------------------------------------
instructor@Ubuntu-00:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 51
Server version: 10.3.9-MariaDB-1:10.3.9+maria~bionic-log mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mysql]> select host,user,password from user;
+-----------+------------------+-------------------------------------------+
| host      | user             | password                                  |
+-----------+------------------+-------------------------------------------+
| localhost | root             | *F8480E475D8AEB0748BE8B0A89F0DA25D5F59F42 |
| 127.0.0.1 | root             | *F8480E475D8AEB0748BE8B0A89F0DA25D5F59F42 |
| ::1       | root             | *F8480E475D8AEB0748BE8B0A89F0DA25D5F59F42 |
| localhost | debian-sys-maint | *BDFE5FA56F72275F0678EF67F4DEA0C4797655CC |
+-----------+------------------+-------------------------------------------+
4 rows in set (0.000 sec)

MariaDB [mysql]> DELETE FROM user WHERE user = 'debian-sys-maint';
Query OK, 1 row affected (0.000 sec)

MariaDB [mysql]> GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.56.%'
   -> IDENTIFIED BY 'javamariadb';
Query OK, 0 rows affected (0.000 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.000 sec)

MariaDB [mysql]> SELECT host,user,password FROM user;
+--------------+------+-------------------------------------------+
| host         | user | password                                  |
+--------------+------+-------------------------------------------+
| localhost    | root | *F8480E475D8AEB0748BE8B0A89F0DA25D5F59F42 |
| 127.0.0.1    | root | *F8480E475D8AEB0748BE8B0A89F0DA25D5F59F42 |
| ::1          | root | *F8480E475D8AEB0748BE8B0A89F0DA25D5F59F42 |
| 192.168.56.% | root | *F8480E475D8AEB0748BE8B0A89F0DA25D5F59F42 |
+--------------+------+-------------------------------------------+
4 rows in set (0.000 sec)

MariaDB [mysql]>
```

7. Remote Access Login
   -By default, MySQL or MariaDB only listens for connections from the localhost.
   -All remote access to the server is denied by default.

-To enable remote access, run the commands below to open MySQL/MariaDB configuration file.

```
$ sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf
```

or

```
$ sudo vi /etc/mysql/mariadb.conf.d/my.cnf
```

-Then make the below change below from:

```
bind-address                    = 127.0.0.1
```

To

```
bind-address                     = 0.0.0.0
```

or

```
#bind-address                    = 127.0.0.1
```

-After making the change above, save the file and run the commands below to restart the server.

```
$ sudo systemctl restart mariadb.service
```

-To verify that the change happens, run the commands below

```
$ sudo netstat -anp | grep 3306
```

-and you should find the result that looks like the one below

```
tcp       0      0 0.0.0.0:3306          0.0.0.0:*        LISTEN         3213/mysqld
```

-Now the server is setup to listen to all IP addresses but individual IP needs to be explicitly configure to connect to a database.

-To enable a client to connect to a database, you must grant access to the remote server.

-It is common for people to want to create a "root" user that can connect from anywhere, so as an example, we'll do just that, but to improve on it we'll create a root user that can connect from anywhere on my local area network (LAN), which has addresses in the subnet 192.168.100.0/24.

-This is an improvement because opening a MariaDB server up to the Internet and granting access to all hosts is bad practice.

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.56.%' IDENTIFIED BY 'javamariadb' WITH GRANT OPTION;
(% is a wildcard)
```

-At this point we have accomplished our goal and we have a user 'root' that can connect from anywhere on the 192.168.56.0/24 LAN.

-After running the commands above, you should be able to access the server from the client computer with that assigned IP.

-To connect to the server from the IP, run the commands below

```
sudo mysql -uroot -pdatabaseuser_password -h server hostname or IP address
```

-That’s it! You’ve successfully configured a remote access to MySQL/MariaDB database server.

-You may want to open Ubuntu Firewall to allow IP address 192.168.56.1 to connect on port 3306.

```
sudo ufw allow from 192.168.56.1 to any port 3306
```