

# HOW TO INSTALL MYSQL/MARIADB SERVER ON RASPBERRY PI

I need to test Gammu with MySQL backend on Raspberry Pi. I've using `NULL` as backend before and I believe everything is set so it's time to use database backend. MySQL is chosen merely because of it's popularity. This tutorial will guide you through steps to install MySQL/MariaDB Server on your Raspberry Pi.

## MySQL/MariaDB Installation on Raspberry Pi

Please note that personally I don't recommend running a database server (MySQL/MariaDB in this case) on a Raspberry Pi unless you have a high-quality and high-speed USB flash drive MicroSd card from which you run the OS.

## Update packages on Raspbian

Luckily for us updating packages on Raspbian is incredibly easy, it involves typing two very simple commands into `terminal`. It is good to remember the two commands that are shown below as they will quickly become some of your most used Linux commands.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

The first of these commands (`sudo apt-get update`) makes a call to the Advanced Packaging Tool (`apt`) to update the package list, this is highly important as the install and upgrade commands only search the pre-grabbed package list and don't make any attempts to update it themselves.

The update command works this by searching the `/etc/apt/sources.list` file, and then polling all the websites in the list for all available packages creating a list of their download location and their current version.

The default location that Raspbian uses for its packages can be found on Raspbians mirror director, this mirror director is designed to automatically direct you to the closest download provider.

Failing to run the update command before running install or upgrade could also cause you to run into download errors especially when packages are moved around on the mirror server.

The second command (`sudo apt-get upgrade`) again utilizes the Advanced Packaging Tool (`apt`), but this time it uses it to check all currently installed packages against the package list, if there is a version miss-match for any it will attempt to update it by downloading the new version from the link in the list. The upgrade tool will never remove a package.

Once all the updates are installed, we're ready to install MySQL server.

## MySQL/MariaDB installation process

In this part of the article we will be exploring on how to install MySQL on Raspberry Pi. First, issue this command:

```
$ sudo apt-get install mysql-server
```

Alternatively you can use this command:

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

and press Enter. Note that on Raspbian, MariaDB is being used as MySQL drop-in replacement. This means that all MySQL command should be available on MariaDB.

```
$ sudo apt-get install mariadb-server
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

```
The following additional packages will be installed:
```

```
galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
libjemalloc1
liblwp-mediatypes-perl libreadline5 libsigsegv2 libterm-readkey-perl libtimedate-perl liburi-perl lsof mariadb-client-10.1
mariadb-client-core-10.1 mariadb-common mariadb-server-10.1 mariadb-server-core-10.1 socat
```

```
Suggested packages:
```

```
gawk-doc libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl
libwww-perl
mariadb-test tinycal
```

```
The following NEW packages will be installed:
```

```
galera-3 gawk libaio1 libcgi-fast-perl libcgi-pm-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
libjemalloc1
liblwp-mediatypes-perl libreadline5 libsigsegv2 libterm-readkey-perl libtimedate-perl liburi-perl lsof mariadb-client-10.1
mariadb-client-core-10.1 mariadb-common mariadb-server mariadb-server-10.1 mariadb-server-core-10.1 socat
```

```
0 upgraded, 30 newly installed, 0 to remove and 236 not upgraded.
```

```
Need to get 4279 kB/22.7 MB of archives.
```

```
After this operation, 169 MB of additional disk space will be used.
```

```
Do you want to continue? [Y/n] y
```

If you notice on the installation screen, the installation process will also install its `client`. This `client` will allow you to connect to your `server` from the command line which is always handy to do. After a short while, you will see this something like this:

```
Created symlink /etc/systemd/system/mysql.service → /lib/systemd/system/mariadb.service.
```

```
Created symlink /etc/systemd/system/mysqld.service → /lib/systemd/system/mariadb.service.
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service.
```

```
Setting up mariadb-server (10.1.23-9+deb9u1) ...
```

```
Processing triggers for libc-bin (2.24-11+deb9u1) ...
```

```
Processing triggers for systemd (232-25+deb9u1) ...
```

```
$
```

and you have your shell back (`$`). It means the installation is almost finished. Let's proceed to next steps.

## Setting MySQL/MariaDB root password

Next steps are database configuration, which the crucial thing is setting root database password. On your terminal, type this command:

```
$ sudo mysql_secure_installation
```

```
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):
```

Just press Enter because we just installed MySQL/MariaDB server. Next step is:

```
OK, successfully used password, moving on...
```

```
Setting the root password ensures that nobody can log into the MariaDB
```

```
root user without the proper authorisation.
```

```
Set root password? [Y/n]
```

Press ☐ Y and enter your password twice. The passwords aren't visible on the screen. This is normal behavior.

```
Set root password? [Y/n] Y
```

```
New password:
```

```
Re-enter new password:
```

```
Password updated successfully!
```

```
Reloading privilege tables..
```

```
... Success!
```

```
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]
```

Press ☐ Y to remove anonymous users.

```
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]
```

Press ☐ Y to disallow root login remotely.

```
Disallow root login remotely? [Y/n] Y
```

```
... Success!
```

```
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]
```

Press ☐ Y to remove test database and user

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

```
- Dropping test database...
```

```
... Success!

- Removing privileges on test database...

... Success!

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

Reload privilege tables now? [Y/n]
```

Press `Y` to reload privilege tables. This is the last step of MySQL/MariaDB interactive installation process.

```
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!
```

That's it. Installation process is complete. Next step we will try to connect to our newly installed server.

## Connect to Database

Here we are, the moment of truth. Now let's try connecting to our server using the client installed at the same time as the server. Type the following command:

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

The `-p` parameter means it will ask for password. Enter your password and if everything is installed correctly, you will see:

```
$ mysql -u root -p

Enter password:

Welcome to the MariaDB monitor.  Commands end with ; or \g.

Your MariaDB connection id is 9

Server version: 10.1.23-MariaDB-9+deb9u1 Raspbian 9.0

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

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

Congratulations, your MySQL/MariaDB server is installed successfully. Have fun!

## Access Denied Error

If you tried to login with correct credentials but it keeps saying like this:

```
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
```

You can try to disable MySQL from trying to authenticate `root` user using `plugin`:

```
$ sudo mysql -u root

[mysql] use mysql;

[mysql] update user set plugin='' where User='root';

[mysql] flush privileges;
```

```
[mysql] \q
```

## Final Words

I hope that you now know **how to install MySQL/MariaDB on Raspberry Pi**. If you run into any issues or have any feedback feel free to drop a comment below.