

# Installing MariaDB .deb Files

## Contents

- [Installing MariaDB with APT](#)
- [Adding the MariaDB APT repository](#)
- [Using the MariaDB Package Repository Setup Script](#)
- [Using the MariaDB Repository Configuration Tool](#)
- [Executing add-apt-repository](#)
- [Creating a Source List File](#)
- [Using Ubuntu Software Center](#)
- [Using Synaptic Package Manager](#)
- [Pinning the MariaDB Repository to a Specific Minor Release](#)
- [Updating the MariaDB APT repository to a New Major Release](#)
- [Updating the Major Release with the MariaDB Package Repository Setup Script](#)
- [Updating the Major Release with the MariaDB Repository Configuration Tool](#)
- [Updating a Repository with add-apt-repository](#)
- [Updating a Source List File](#)
- [Importing the MariaDB GPG Public Key](#)
- [Installing MariaDB Packages with APT](#)
- [Installing the Most Common Packages with APT](#)
- [Installing MariaDB Server with APT](#)
- [Installing MariaDB Galera Cluster with APT](#)
- [Installing MariaDB Clients and Client Libraries with APT](#)
- [Installing Mariabackup with APT](#)
- [Installing Plugins with APT](#)
- [Installing Older Versions from the Repository](#)
- [Installing MariaDB with dpkg](#)
- [After Installation](#)
- [Installation Issues](#)
- [Upgrading mariadb-server and mariadb-client packages](#)
- [Version Mismatch Between MariaDB and Ubuntu/Debian Repositories](#)
- [Pinning the MariaDB Repository](#)
- [Specifying Specific Versions to Install](#)
- [Holding Packages](#)
- [Available DEB Packages](#)
- [Available DEB Packages in MariaDB 10.4](#)

[Available DEB Packages in MariaDB 10.2 and MariaDB 10.3](#)

[Available DEB Packages in MariaDB 10.1](#)

[Available DEB Packages in MariaDB 5.5](#)

[Actions Performed by DEB Packages](#)

[Users and Groups Created](#)

[Installing MariaDB with APT](#)

On Debian, Ubuntu, and other similar Linux distributions, it is highly recommended to install the relevant .deb packages from MariaDB's repository using [apt](#), [aptitude](#), [Ubuntu Software Center](#), [Synaptic Package Manager](#), or another package manager.

This page walks you through the simple installation steps using apt.  
Adding the MariaDB APT repository

We currently have APT repositories for the following Linux distributions:

Debian 8 (Stretch)

Debian 9 (Jessie)

Debian Unstable (Sid)

Ubuntu 14.04 LTS (Trusty)

Ubuntu 16.04 LTS (Xenial)

Ubuntu 17.10 (Artful)

Ubuntu 18.04 LTS (Bionic)

Ubuntu 18.10 (Cosmic)

Mint 17 (Qiana)

Mint 17.1 (Rebecca)

Mint 18 (Sarah)

Mint 19 (Tara)

[Using the MariaDB Package Repository Setup Script](#)

If you want to install MariaDB with apt, then you can configure apt to install from MariaDB Corporation's MariaDB Package Repository by using the [MariaDB Package Repository setup script](#).

MariaDB Corporation provides a MariaDB Package Repository for several Linux distributions that use apt to manage packages. This repository contains software packages related to MariaDB Server, including the server itself, [clients and utilities](#), [client libraries](#), [plugins](#), and [Mariabackup](#). The MariaDB Package Repository setup script automatically configures your system to install packages from the MariaDB Package Repository.

To use the script, execute the following command:

```
curl -sS https://downloads.mariadb.com/MariaDB/mariadb_repo_setup | sudo bash
```

Note that this script also configures a repository for [MariaDB MaxScale](#) and a repository for MariaDB Tools, which currently only contains [Percona XtraBackup](#) and its dependencies.

Using the MariaDB Repository Configuration Tool

If you want to install MariaDB with apt, then you can configure apt to install from MariaDB Foundation's MariaDB Repository by using the [MariaDB Repository Configuration Tool](#).

The MariaDB Foundation provides a MariaDB repository for several Linux distributions that use apt-get to manage packages. This repository contains software packages related to MariaDB Server, including the server itself, [clients and utilities](#), [client libraries](#), [plugins](#), and [Mariabackup](#). The MariaDB Repository Configuration Tool can easily generate the appropriate commands to add the repository for your distribution.

There are several ways to add the repository.

Executing add-apt-repository

One way to add an apt repository is by using the [add-apt-repository](#) command. This command will add the repository configuration to /etc/apt/sources.list. For example, if you wanted to use the repository to install [MariaDB 10.3](#) on Ubuntu 18.04 LTS (Bionic), then you could use the following commands to add the MariaDB apt repository:

```
sudo apt-get install software-properties-common
sudo add-apt-repository 'deb [arch=amd64,arm64,ppc64el] http://sfo1.mirrors.digitalocean.com/mariadb/repo/10.3/ubuntu bionic main'
And then you would have to update the package cache by executing the following command:
sudo apt update
```

Creating a Source List File

Another way to add an apt repository is by creating a [source list](#) file in /etc/apt/sources.list.d/.

For example, if you wanted to use the repository to install [MariaDB 10.3](#) on Ubuntu 18.04 LTS (Bionic), then you could create the MariaDB.list file in /etc/apt/sources.list.d/ with the following contents to add the MariaDB apt repository:

```
# MariaDB 10.3 repository list - created 2019-01-27 09:50 UTC
# http://downloads.mariadb.org/mariadb/repositories/

deb [arch=amd64,arm64,ppc64el] http://sfo1.mirrors.digitalocean.com/mariadb/repo/10.3/ubuntu bionic main
```

```
deb-src http://sfo1.mirrors.digitalocean.com/mariadb/repo/10.3/ubuntu bionic main
```

And then you would have to update the package cache by executing the following command:

```
sudo apt update
```

#### Using Ubuntu Software Center

Another way to add an apt repository is by using [Ubuntu Software Center](#).

You can do this by going to the Software Sources window. This window can be opened either by navigating to Edit > Software Sources or by navigating to System > Administration > Software Sources.

Once the Software Sources window is open, go to the Other Software tab, and click the Add button. At that point, you can input the repository information provided by the [MariaDB Repository Configuration Tool](#).

#### Using Synaptic Package Manager

Another way to add an apt repository is by using [Synaptic Package Manager](#).

You can do this by going to the Software Sources window. This window can be opened either by navigating to System > Administrator > Software Sources or by navigating to Settings > Repositories.

Once the Software Sources window is open, go to the Other Software tab, and click the Add button. At that point, you can input the repository information provided by the [MariaDB Repository Configuration Tool](#).

#### Pinning the MariaDB Repository to a Specific Minor Release

If you wish to pin the apt repository to a specific minor release, or if you would like to downgrade to a specific minor release, then you can create a apt repository with the URL hard-coded to that specific minor release.

The MariaDB Foundation archives repositories of old minor releases at the following URL:

<http://archive.mariadb.org/>

For example, if you wanted to pin your repository to [MariaDB 10.3.14](#) on Ubuntu 18.04 LTS (Bionic), then you would have to first remove any existing MariaDB repository source list file from `/etc/apt/sources.list.d/`. And then you could use the following commands to add the MariaDB apt-get repository:

```
sudo add-apt-repository 'deb [arch=amd64,arm64,ppc64el] http://archive.mariadb.org/mariadb-10.3.14/repo/ubuntu/ bionic main'
```

And then you would have to update the package cache by executing the following command:

```
sudo apt update
```

Updating the MariaDB APT repository to a New Major Release

MariaDB's apt repository can be updated to a new major release. How this is done depends on how you originally configured the repository.

Updating the Major Release with the MariaDB Package Repository Setup Script

If you configured apt to install from MariaDB Corporation's MariaDB Package Repository by using the [MariaDB Package Repository setup script](#), then you can update the major release that the repository uses by running the script again.

Updating the Major Release with the MariaDB Repository Configuration Tool

If you configured apt to install from MariaDB Foundation's MariaDB Repository by using the [MariaDB Repository Configuration Tool](#), then you can update the major release in various ways, depending on how you originally added the repository.

Updating a Repository with add-apt-repository

If you added the apt repository by using the [add-apt-repository](#) command, then you can update the major release that the repository uses by using the [add-apt-repository](#) command again.

First, look for the repository string for the old version in `/etc/apt/sources.list`.

And then, you can remove the repository for the old version by executing the [add-apt-repository](#) command and providing the `--remove` option. For example, if you wanted to remove a [MariaDB 10.2](#) repository, then you could do so by executing something like the following:

```
sudo add-apt-repository --remove 'deb [arch=amd64,arm64,ppc64el] http://sfo1.mirrors.digitalocean.com/mariadb/repo/10.2/ubuntu bionic main'
```

After that, you can add the repository for the new version with the [add-apt-repository](#) command. For example, if you wanted to use the repository to install [MariaDB 10.3](#) on Ubuntu 18.04 LTS (Bionic), then you could use the following commands to add the MariaDB apt repository:

```
sudo apt-get install software-properties-common
sudo add-apt-repository 'deb [arch=amd64,arm64,ppc64el] http://sfo1.mirrors.digitalocean.com/mariadb/repo/10.3/ubuntu bionic main'
```

And then you would have to update the package cache by executing the following command:

```
sudo apt update
```

After that, the repository should refer to [MariaDB 10.3](#).

#### Updating a Source List File

If you added the apt repository by creating a [source list](#) file in `/etc/apt/sources.list.d/`, then you can update the major release that the repository uses by updating the source list file in-place. For example, if you wanted to change the repository from [MariaDB 10.2](#) to [MariaDB 10.3](#), and if the source list file was at `/etc/apt/sources.list.d/MariaDB.list`, then you could execute the following:

```
sudo sed -i 's/10.2/10.3/' /etc/apt/sources.list.d/MariaDB.list
```

And then you would have to update the package cache by executing the following command:

```
sudo apt update
```

After that, the repository should refer to [MariaDB 10.3](#).

#### Importing the MariaDB GPG Public Key

Before MariaDB can be installed, you also have to import the GPG public key that is used to verify the digital signatures of the packages in our repositories. This allows the apt utility to verify the integrity of the packages that it installs.

Prior to Debian 9 (Stretch), and Debian Unstable (Sid), and Ubuntu 16.04 LTS (Xenial), the id of our GPG public key is `0xc9cb082a1bb943db`. The full key fingerprint is:

```
1993 69E5 404B D5FC 7D2F E43B CBCB 082A 1BB9 43DB
```

The [apt-key](#) utility can be used to import this key. For example:

```
sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xc9cb082a1bb943db
```

Starting with Debian 9 (Stretch), and Debian Unstable (Sid), and Ubuntu 16.04 LTS (Xenial), the id of our GPG public key is `0xF1656F24C74CD1D8`. The full key fingerprint is:

```
177F 4010 FE56 CA33 3630 0305 F165 6F24 C74C D1D8
```

The [apt-key](#) utility can be used to import this key. For example:

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

Starting with Debian 9 (Stretch), the [dirmngr](#) package needs to be installed before the GPG public key can be imported. To install it, execute: `sudo apt install dirmngr`  
If you are unsure which GPG public key you need, then it is perfectly safe to import both keys.

The command used to import the GPG public key is the same on both Debian and Ubuntu. For example:

```
$ sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xcbc082a1bb943db
```

```
Executing: gpg --ignore-time-conflict --no-options --no-default-keyring --secret-keyring /tmp/tmp.ASyOPV87XC --trustdb-name /etc/apt/trustdb.gpg --keyring /etc/apt/trusted.gpg --primary-keyring /etc/apt/trusted.gpg --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xcbc082a1bb943db
```

```
gpg: requesting key 1BB943DB from hkp server keyserver.ubuntu.com
gpg: key 1BB943DB: "MariaDB Package Signing Key <package-signing-key@mariadb.org>" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:          imported: 1
```

Once the GPG public key is imported, you are ready to install packages from the repository.

#### Installing MariaDB Packages with APT

After the apt repository is configured, you can install MariaDB by executing the [apt-get](#) command. The specific command that you would use would depend on which specific packages that you want to install.

#### Installing the Most Common Packages with APT

To Install the most common packages, first you would have to update the package cache by executing the following command:

```
sudo apt update
```

MariaDB starting with [10.4](#)

In [MariaDB 10.4](#) and later, to Install the most common packages, execute the following command:

```
sudo apt-get install mariadb-server galera-4 mariadb-client libmariadb3 mariadb-backup mariadb-common
```

MariaDB [10.2](#) - [10.3](#)

In [MariaDB 10.2](#) and [MariaDB 10.3](#), to Install the most common packages, execute the following command:

```
sudo apt-get install mariadb-server galera mariadb-client libmariadb3 mariadb-backup mariadb-common
```

MariaDB until [10.1](#)

In [MariaDB 10.1](#) and before, to Install the most common packages, execute the following command:

```
sudo apt-get install mariadb-server galera mariadb-client libmysqlclient18 mariadb-backup mariadb-common
```

Installing MariaDB Server with APT

To Install MariaDB Server, first you would have to update the package cache by executing the following command:

```
sudo apt update
```

Then, execute the following command:

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

Installing MariaDB Galera Cluster with APT

The process to install MariaDB Galera Cluster with the MariaDB apt-get repository is practically the same as installing standard MariaDB Server.

In [MariaDB 10.1](#) and later, Galera Cluster support has been included in the standard MariaDB Server packages, so you will need to install the mariadb-server package, as you normally would.

In [MariaDB 10.4](#) and later, you also need to install the galera-4 package to obtain the [Galera 4](#) wsrep provider library.

In [MariaDB 10.3](#) and before, you also need to install the galera package to obtain the [Galera 3](#) wsrep provider library.

To install MariaDB Galera Cluster, first you would have to update the package cache by executing the following command:

```
sudo apt update
```

MariaDB starting with [10.4](#)

In [MariaDB 10.4](#) and later, to install MariaDB Galera Cluster, you could execute the following command:

```
sudo apt-get install mariadb-server mariadb-client galera-4
```



MariaDB until [10.3](#)

In [MariaDB 10.3](#) and before, to install MariaDB Galera Cluster, you could execute the following command:

```
sudo apt-get install mariadb-server mariadb-client galera
```

MariaDB Galera Cluster also has a separate package that can be installed on arbitrator nodes. In [MariaDB 10.4](#) and later, the package is called galera-arbitrator-

4 In [MariaDB 10.3](#) and before, the package is called galera-arbitrator-3. This package should be installed on whatever node you want to serve as the arbitrator. It can either run on a separate server that is not acting as a cluster node, which is the recommended configuration, or it can run on a server that is also acting as an existing cluster node.

MariaDB starting with [10.4](#)

In [MariaDB 10.4](#) and later, to install the arbitrator package, you could execute the following command:

```
sudo apt-get install galera-arbitrator-4
```

MariaDB until [10.3](#)

In [MariaDB 10.3](#) and before, to install the arbitrator package, you could execute the following command:

```
sudo apt-get install galera-arbitrator-3
```

Installing MariaDB Clients and Client Libraries with APT

In [MariaDB 10.2](#) and later, [MariaDB Connector/C](#) has been included as the client library.

To Install the clients and client libraries, first you would have to update the package cache by executing the following command:

```
sudo apt update
```

Then, in [MariaDB 10.2](#) and later, execute the following command:

```
sudo apt-get install mariadb-client libmariadb3
```

Or in [MariaDB 10.1](#) and before, execute the following command:

```
sudo apt-get install mariadb-client libmysqlclient18
```

Installing Mariabackup with APT

To install [Mariabackup](#), first you would have to update the package cache by executing the following command:

```
sudo apt update
```

Then, execute the following command:

```
sudo apt-get install mariadb-backup
```

Installing Plugins with APT

Some [plugins](#) may also need to be installed.

For example, to install the [cracklib\\_password\\_check](#) password validation plugin, first you would have to update the package cache by executing the following command:

```
sudo apt update
```

Then, execute the following command:

```
sudo apt-get install mariadb-cracklib-password-check
```

Installing Older Versions from the Repository

The MariaDB apt repository contains the last few versions of MariaDB. To show what versions are available, use the [apt-cache](#) command:

```
sudo apt-cache showpkg mariadb-server
```

In the output you will see the available versions.

To install an older version of a package instead of the latest version we just need to specify the package name, an equal sign, and then the version number.

However, when installing an older version of a package, if apt-get has to install dependencies, then it will automatically choose to install the latest versions of those packages. To ensure that all MariaDB packages are on the same version in this scenario, it is necessary to specify them all. Therefore, to install [MariaDB 10.3.14](#) from this apt repository, we would do the following:

```
sudo apt-get install mariadb-server=10.3.14-1 mariadb-client=10.3.14-1 libmariadb3=10.3.14-1 mariadb-backup=10.3.14-1 mariadb-common=10.3.14-1
```

The rest of the install and setup process is as normal.

### Installing MariaDB with dpkg

While it is not recommended, it is possible to download and install the .deb packages manually. However, it is generally recommended to use a package manager like apt-get.

A tarball that contains the .deb packages can be downloaded from the following URL:

<https://downloads.mariadb.com>

For example, to install the [MariaDB 10.4.8](#) .deb packages on Ubuntu 18.04 LTS (Bionic), you could execute the following:

```
sudo apt-get update
sudo apt-get install libdbi-perl libdbd-mysql-perl psmisc libaio1 socat
wget https://downloads.mariadb.com/MariaDB/mariadb-10.4.8/repo/ubuntu/mariadb-10.4.8-ubuntu-bionic-amd64-debs.tar
tar -xvf mariadb-10.4.8-ubuntu-bionic-amd64-debs.tar
cd mariadb-10.4.8-ubuntu-bionic-amd64-debs/
sudo dpkg --install ./mariadb-common*.deb \
    ./mysql-common*.deb \
    ./mariadb-client*.deb \
    ./libmariadb3*.deb \
    ./libmysqlclient18*.deb
sudo dpkg --install ./mariadb-server*.deb \
    ./mariadb-backup*.deb \
    ./galera-4*.deb
```

### After Installation

After the installation is complete, you can [start MariaDB](#).

If you are using [MariaDB Galera Cluster](#), then keep in mind that the first node will have to be [bootstrapped](#).

### Installation Issues

MariaDB starting with [5.5](#)

Upgrading mariadb-server and mariadb-client packages

As noted in [MDEV-4266](#), the mariadb-server and mariadb-client packages have a minor upgrade issue if you use 'apt-get install mariadb-server' or 'apt-get install mariadb-client' to upgrade them instead of the more common 'apt-get upgrade'. This is because those two packages depend on mariadb-server-5.5 and mariadb-client-5.5 with no specific version of those packages. For example, if you have the mariadb-server package installed, version 5.5.29 and you install version 5.5.30 of that package it will not automatically upgrade the mariadb-server-5.5 package to version 5.5.30 like you would expect because the 5.5.29 version of that package satisfies the dependency.

The mariadb-server and mariadb-client packages are virtual packages, they only exist to require the installation of the mariadb-server-5.5 and mariadb-client-5.5 packages, respectively. MariaDB will function normally with a, for example, version 5.5.30 version of the mariadb-server package and a version 5.5.29 version of the mariadb-server-5.5 package. No data is at risk. However, expected behavior is for 'apt-get install mariadb-server' to upgrade everything to the latest version (if a new version is available), so this is definitely a bug.

A fix is planned for this bug in a future version of MariaDB. In the mean time, when upgrading MariaDB, use 'apt-get upgrade' or 'apt-get install mariadb-server-5.5'.  
MariaDB [5.1](#) - [5.5](#)

#### Version Mismatch Between MariaDB and Ubuntu/Debian Repositories

As mentioned [here](#) (and in [MDEV-4080](#) and [MDEV-3882](#)) sometimes APT will refuse to install MariaDB. Or, if MariaDB is already installed, suggest the removal of MariaDB to apply an upgrade to the mysql-common or libmysqlclient packages. This happens whenever the version number of those two packages is higher in the distribution repositories than the versions in the MariaDB repositories. Most MariaDB packages have different names than their MySQL counterparts, but in order for upgrades from MySQL to MariaDB to be successful in APT, those two packages must be named the same. Because they have the same names, APT just checks the version numbers and tries to install what it considers to be the most recent.

It is rare for the version numbers of mysql-common or libmysqlclient to be higher in the official Ubuntu or Debian repositories than they are in the MariaDB repositories, but it has happened. Whenever it has it has been because of critical bug fix releases for bugs that existed in the version of MySQL in the distribution repositories but which had already been fixed in the version of MariaDB in the MariaDB repositories.

If a situation as described above exists when you try to install MariaDB you will get an error like this:

The following packages have unmet dependencies:

mariadb-server : Depends: mariadb-server-5.5 but it is not going to be installed

E: Unable to correct problems, you have held broken packages.

There are three primary ways of fixing this issue:

[Pinning the MariaDB Repository](#)

[Specifying Specific Versions to Install](#)

[Holding Packages](#)

Click on the links above to jump to directions for each (or simply scroll down).

## Pinning the MariaDB Repository

It is possible to *pin* the MariaDB repository used so the packages it provides will always have an higher priority over the ones from the system repositories.

This is done by creating a file with the `.pref` extension under `/etc/apt/preferences.d/` with the following contents:

```
Package: *
```

```
Pin: origin <mirror-domain>
```

```
Pin-Priority: 1000
```

Replace `<mirror-domain>` with the domain name of the MariaDB mirror you use. For example `ftp.osuosl.org`.

### Specifying Specific Versions to Install

A way to fix this is to specify the exact version of the two packages that you want to install. To do this, first determine the full version numbers of the affected packages.

An easy way to do so is with `'apt-cache show'`:

```
apt-cache show mysql-common | grep Version
```

```
apt-cache show libmysqlclient18 | grep Version
```

For each of the above you will be given a list of versions. The ones in the MariaDB repositories will have `"mariadb"` in the version strings and are the ones you want. With the version numbers in hand you will be able to install MariaDB by explicitly specifying the version numbers like so:

```
apt-get install mariadb-server-5.5 mariadb-client-5.5 \
```

```
  libmysqlclient18=<version-number> \
```

```
  mysql-common=<version-number>
```

Replace the two instances of `<version-number>` in the example above with the actual version number of MariaDB that you want to install.

Even after having installed these specific packages versions, running `'apt-get dist-upgrade'` will still try to upgrade the packages to the highest version available.

## Holding Packages

After MariaDB is installed, and as long as the version number issue exists, an `'apt-get dist-upgrade'` will try to remove MariaDB in order to install the "upgraded" `libmysqlclient` and `mysql-common` packages. To prevent this from happening you can *hold* them so that apt doesn't try to upgrade them. To do so, open a terminal, become root with `'sudo -s'`, and then enter the following:

```
echo libmysqlclient18 hold | dpkg --set-selections
```

```
echo mysql-common hold | dpkg --set-selections
```

The holds will prevent you from upgrading MariaDB, so when you want to remove the holds, open a terminal, become root with `'sudo -s'`, and then enter the following:

```
echo libmysqlclient18 install | dpkg --set-selections
```

```
echo mysql-common install | dpkg --set-selections
```

You will then be able to upgrade MariaDB as normal (e.g. with `'sudo apt-get update; sudo apt-get upgrade'`).

## Available DEB Packages

The available DEB packages depend on the specific MariaDB release series.  
Available DEB Packages in [MariaDB 10.4](#)  
MariaDB starting with [10.4](#)

In [MariaDB 10.4](#), the following DEBs are available:

Package Name	Description
galera-4	The WSREP provider for <a href="#">Galera 4</a> .
libmariadb3	Dynamic client libraries.
libmariadb-dev	Development headers and static libraries.
libmariadbclient18	Virtual package to satisfy external depends
libmysqlclient18	Virtual package to satisfy external depends
mariadb-backup	<a href="#">Mariabackup</a>
mariadb-client	Client tools like mysql CLI, mysqldump, and others.
mariadb-client-core	Core client tools
mariadb-common	Character set files and /etc/my.cnf
mariadb-plugin-connect	The <a href="#">CONNECT</a> storage engine.
mariadb-plugin-cracklib-password-check	The <a href="#">cracklib_password_check</a> password validation plugin.
mariadb-plugin-gssapi-client	The client-side component of the <a href="#">gssapi</a> authentication plugin.
mariadb-plugin-gssapi-server	The server-side component of the <a href="#">gssapi</a> authentication plugin.
mariadb-plugin-rocksdb	The <a href="#">MyRocks</a> storage engine.
mariadb-plugin-spider	The <a href="#">SPIDER</a> storage engine.

mariadb-plugin-tokudb	The <a href="#">TokuDB</a> storage engine.
mariadb-server	The server and server tools, like <a href="#">myisamchk</a> and <a href="#">mysqlhotcopy</a> are here.
mariadb-server-core	The core server.
mariadb-test	mysql-client-test executable, and mysql-test framework with the tests.
mariadb-test-data	MariaDB database regression test suite - data files

Available DEB Packages in [MariaDB 10.2](#) and [MariaDB 10.3](#)

MariaDB starting with [10.2](#)

In [MariaDB 10.2](#) and [MariaDB 10.3](#), the following DEBs are available:

Package Name	Description
galera	The WSREP provider for <a href="#">Galera 3</a> .
libmariadb3	Dynamic client libraries.
libmariadb-dev	Development headers and static libraries.
libmariadbclient18	Virtual package to satisfy external depends
libmysqlclient18	Virtual package to satisfy external depends
mariadb-backup	<a href="#">Mariabackup</a>
mariadb-client	Client tools like mysql CLI, mysqldump, and others.
mariadb-client-core	Core client tools
mariadb-common	Character set files and /etc/my.cnf
mariadb-plugin-connect	The <a href="#">CONNECT</a> storage engine.
mariadb-plugin-cracklib-password-check	The <a href="#">cracklib_password_check</a> password validation plugin.
mariadb-plugin-gssapi-client	The client-side component of the <a href="#">gssapi</a> authentication plugin.

mariadb-plugin-gssapi-server	The server-side component of the <a href="#">gssapi</a> authentication plugin.
mariadb-plugin-rocksdb	The <a href="#">MyRocks</a> storage engine.
mariadb-plugin-spider	The <a href="#">SPIDER</a> storage engine.
mariadb-plugin-tokudb	The <a href="#">TokuDB</a> storage engine.
mariadb-server	The server and server tools, like <a href="#">myisamchk</a> and <a href="#">mysqlhotcopy</a> are here.
mariadb-server-core	The core server.
mariadb-test	mysql-client-test executable, and mysql-test framework with the tests.
mariadb-test-data	MariaDB database regression test suite - data files

Available DEB Packages in [MariaDB 10.1](#)

MariaDB starting with [10.1](#)

In [MariaDB 10.1](#), the following DEBs are available:

Package Name	Description
galera	The WSREP provider for <a href="#">Galera 3</a> .
libmysqlclient18	Dynamic client libraries.
mariadb-backup	<a href="#">Mariabackup</a>
mariadb-client	Client tools like mysql CLI, mysqldump, and others.
mariadb-client-core	Core client tools
mariadb-common	Character set files and /etc/my.cnf
mariadb-plugin-connect	The <a href="#">CONNECT</a> storage engine.
mariadb-plugin-cracklib-password-check	The <a href="#">cracklib_password_check</a> password validation plugin.
mariadb-plugin-gssapi-client	The client-side component of the <a href="#">gssapi</a> authentication plugin.



mariadb-plugin-gssapi-server	The server-side component of the <a href="#">gssapi</a> authentication plugin.
mariadb-plugin-spider	The <a href="#">SPIDER</a> storage engine.
mariadb-plugin-tokudb	The <a href="#">TokuDB</a> storage engine.
mariadb-server	The server and server tools, like <a href="#">myisamchk</a> and <a href="#">mysqlhotcopy</a> are here.
mariadb-server-core	The core server.
mariadb-test	mysql-client-test executable, and mysql-test framework with the tests.
mariadb-test-data	MariaDB database regression test suite - data files

#### Available DEB Packages in [MariaDB 5.5](#)

MariaDB starting with [5.5](#)

In [MariaDB 5.5](#), the following DEBs are available:

Package Name	Description
libmysqlclient18	Dynamic client libraries.
mariadb-client	Client tools like mysql CLI, mysqldump, and others.
mariadb-client-core	Core client tools
mariadb-common	Character set files and /etc/my.cnf
mariadb-server	The server and server tools, like <a href="#">myisamchk</a> and <a href="#">mysqlhotcopy</a> are here.
mariadb-server-core	The core server.
mariadb-test	mysql-client-test executable, and mysql-test framework with the tests.
mariadb-test-data	MariaDB database regression test suite - data files

#### Actions Performed by DEB Packages

Users and Groups Created When the mariadb-server DEB package is installed, it will create a user and group named mysql, if they do not already exist.