MariaDB instructions

The instructions below are sourced from multiple internet forums.

All downloads for MariaDB are located in the download section of the official MariaDB foundation website. Click the link to the version you would like, and a list of downloads for multiple operating systems, architectures, and installation file types is displayed.

**Installing on windows**

**The following instructions are from tutorialspoint.**

After locating and downloading an automated install file (MSI), simply double click the file to start the installation. The installation wizard will walk you through every step of installation and any necessary settings.

Test the installation by starting it from the command prompt. Navigate to the location of the installation, typically in the directory, and type the following at the prompt −

mysqld.exe --console

If the installation is successful, you will see messages related to startup. If this does not appear, you may have permission issues. Ensure that your user account can access the application. Graphical clients are available for MariaDB administration in the Windows environment. If you find the command line uncomfortable or cumbersome, be sure to experiment with them.

Testing the Installation

Perform a few simple tasks to confirm the functioning and installation of MariaDB.

**Use the Admin Utility to Get Server Status**

View the server version with the mysqladmin binary.

[root@host]# mysqladmin --version

It should display the version, distribution, operating system, and architecture. If you do not see the output of that type, examine your installation for issues.

**Execute Simple Commands with a Client**

Bring up the command prompt for MariaDB. This should connect you to MariaDB and allow execution of commands. Enter a simple command as follows −

mysql> SHOW DATABASES;

Post- Installation

After successful installation of MariaDB, set a root password. A fresh install will have a blank password. Enter the following to set the new password −

mysqladmin -u root password "[enter your password here]";

Enter the following to connect to the server with your new credentials −

mysql -u root -p

Enter password:\*\*\*\*\*\*\*

Upgrading on Windows

If you already have MySQL installed on your Windows system, and want to upgrade to MariaDB; do not uninstall MySQL and install MariaDB. This will cause a conflict with the existing database. You must instead install MariaDB, and then use the upgrade wizard in the Windows installation file.

The options of your MySQL my.cnf file should work with MariaDB. However, MariaDB has many features, which are not found in MySQL.

Consider the following conflicts in your my.cnf file −

* MariaDB uses Aria storage engine by default for temporary files. If you have a lot of temporary files, modify key buffer size if you do not use MyISAM tables.
* If your applications connect/disconnect frequently, alter the thread cache size.
* If you use over 100 connections, use the thread pool.

**Installing on mac OSX**

**The following instructions are from the MariaDB blog on the official site.**

1. Install Xcode

Run xcode-select --install.

$ xcode-select --install

xcode-select: note: install requested for command line developer tools

2. Install Homebrew

Run /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)".

$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"

==> This script will install:

/usr/local/bin/brew

/usr/local/share/doc/homebrew

/usr/local/share/man/man1/brew.1

/usr/local/share/zsh/site-functions/\_brew

/usr/local/etc/bash\_completion.d/brew

/usr/local/Homebrew

==> The following new directories will be created:

/usr/local/sbin

/usr/local/Caskroom

Press RETURN to continue or any other key to abort

Password:

==> /usr/bin/sudo /bin/mkdir -p /usr/local/sbin /usr/local/Caskroom

==> /usr/bin/sudo /bin/chmod g+rwx /usr/local/sbin /usr/local/Caskroom

==> /usr/bin/sudo /usr/sbin/chown rob /usr/local/sbin /usr/local/Caskroom

==> /usr/bin/sudo /usr/bin/chgrp admin /usr/local/sbin /usr/local/Caskroom

==> Downloading and installing Homebrew...

remote: Enumerating objects: 5822, done.

remote: Counting objects: 100% (5822/5822), done.

remote: Compressing objects: 100% (43/43), done.

remote: Total 24553 (delta 5779), reused 5821 (delta 5779), pack-reused 18731

Receiving objects: 100% (24553/24553), 7.33 MiB | 1.09 MiB/s, done.

Resolving deltas: 100% (18413/18413), completed with 1257 local objects.

...

HEAD is now at 67d1bc6fb Merge pull request #7615 from Bo98/test-dep-satisfied

==> Downloading https://homebrew.bintray.com/bottles-portable-ruby/portable-ruby-2.6.3.mavericks.bottle.tar.gz

######################################################################## 100.0%

==> Pouring portable-ruby-2.6.3.mavericks.bottle.tar.gz

Updated 1 tap (homebrew/core).

==> New Formulae

...

==> Renamed Formulae

...

==> Deleted Formulae

...

==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.

Read the analytics documentation (and how to opt-out) here:

https://docs.brew.sh/Analytics

No analytics data has been sent yet (or will be during this `install` run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:

https://github.com/Homebrew/brew#donations

==> Next steps:

- Run `brew help` to get started

- Further documentation:

https://docs.brew.sh

3. Check Homebrew

Run brew doctor. Follow on-screen instructions to fix warnings if necessary.

$ brew doctor

Please note that these warnings are just used to help the Homebrew maintainers with debugging if you file an issue. If everything you use Homebrew for is working fine: please don't worry or file an issue; just ignore this. Thanks!

Warning: You have unlinked kegs in your Cellar.

Leaving kegs unlinked can lead to build-trouble and cause brews that depend on those kegs to fail to run properly once built. Run `brew link` on these:

gettext

git

Warning: Some installed formulae are missing dependencies.

You should `brew install` the missing dependencies:

brew install openssl@1.1

Run `brew missing` for more details.

$ brew link gettext git

Linking /usr/local/Cellar/gettext/0.19.8.1... 187 symlinks created

Linking /usr/local/Cellar/git/2.19.1...

Error: Could not symlink bin/git

Target /usr/local/bin/git

already exists. You may want to remove it:

rm '/usr/local/bin/git'

To force the link and overwrite all conflicting files:

brew link --overwrite git

To list all files that would be deleted:

brew link --overwrite --dry-run git

...

4. Update Homebrew

Run brew update.

$ brew update

Already up-to-date.

5. Verify MariaDB version in Homebrew repo

Run brew info mariadb.

$ brew info mariadb

mariadb: stable 10.4.13 (bottled)

Drop-in replacement for MySQL

https://mariadb.org/

Conflicts with:

mariadb-connector-c (because both install plugins)

mysql (because mariadb, mysql, and percona install the same binaries)

mytop (because both install `mytop` binaries)

percona-server (because mariadb, mysql, and percona install the same binaries)

/usr/local/Cellar/mariadb/10.2.14 (641 files, 168.6MB)

Poured from bottle on 2018-04-30 at 11:34:15

/usr/local/Cellar/mariadb/10.3.10 (652 files, 173.3MB) \*

Built from source on 2018-10-12 at 07:16:37

From: https://github.com/Homebrew/homebrew-core/blob/master/Formula/mariadb.rb

==> Dependencies

Build: cmake ✘, pkg-config ✘

Required: groonga ✘, openssl@1.1 ✔

==> Caveats

A "/etc/my.cnf" from another install may interfere with a Homebrew-built

server starting up correctly.

MySQL is configured to only allow connections from localhost by default

To have launchd start mariadb now and restart at login:

brew services start mariadb

Or, if you don't want/need a background service you can just run:

mysql.server start

==> Analytics

install: 15,161 (30 days), 36,985 (90 days), 172,584 (365 days)

install-on-request: 14,780 (30 days), 36,286 (90 days), 168,365 (365 days)

build-error: 0 (30 days)

6. Install MariaDB

Run brew install mariadb. Follow on-screen instructions to upgrade if necessary to upgrade a previously installed version.

$ brew install mariadb

Updating Homebrew...

==> Auto-updated Homebrew!

Updated 1 tap (homebrew/core).

==> Updated Formulae

fonttools timidity

==> Downloading https://homebrew.bintray.com/bottles/mecab-0.996.mojave.bottle.3.tar.gz

==> Downloading from https://akamai.bintray.com/ef/[...]?\_\_gda\_\_=exp=1590016

######################################################################## 100.0%

==> Downloading https://homebrew.bintray.com/bottles/mecab-ipadic-2.7.0-20070801.mojave.bottle.tar.gz

==> Downloading from https://akamai.bintray.com/30/[...]?\_\_gda\_\_=exp=1590016

######################################################################## 100.0%

==> Downloading https://homebrew.bintray.com/bottles/msgpack-3.2.1.mojave.bottle.tar.gz

==> Downloading from https://akamai.bintray.com/3b/[...]?\_\_gda\_\_=exp=1590016

######################################################################## 100.0%

==> Downloading https://homebrew.bintray.com/bottles/pcre-8.44.mojave.bottle.tar.gz

==> Downloading from https://akamai.bintray.com/ed/[...]?\_\_gda\_\_=exp=1590016

######################################################################## 100.0%

==> Downloading https://homebrew.bintray.com/bottles/groonga-10.0.2.mojave.bottle.tar.gz

==> Downloading from https://akamai.bintray.com/df/[...]?\_\_gda\_\_=exp=1590016

######################################################################## 100.0%

==> Downloading https://homebrew.bintray.com/bottles/mariadb-10.4.13.mojave.bottle.tar.gz

==> Downloading from https://akamai.bintray.com/e4/[...]?\_\_gda\_\_=exp=1590016

######################################################################## 100.0%

Error: mariadb 10.3.10 is already installed

To upgrade to 10.4.13, run `brew upgrade mariadb`.

$ brew upgrade mariadb

==> Upgrading 1 outdated package:

mariadb 10.3.10 -> 10.4.13

==> Upgrading mariadb 10.3.10 -> 10.4.13

==> Downloading https://homebrew.bintray.com/bottles/cmake-3.17.2.mojave.bottle.tar.gz

==> Downloading from https://akamai.bintray.com/ed/[...]?\_\_gda\_\_=exp=1590016

######################################################################## 100.0%

==> Downloading https://homebrew.bintray.com/bottles/pkg-config-0.29.2\_3.mojave.bottle.tar.gz

==> Downloading from https://akamai.bintray.com/0d/[...]?\_\_gda\_\_=exp=1590016

######################################################################## 100.0%

==> Downloading https://homebrew.bintray.com/bottles/mecab-0.996.mojave.bottle.3.tar.gz

Already downloaded: /Users/rob/Library/Caches/Homebrew/downloads/[...]--mecab-0.996.mojave.bottle.3.tar.gz

==> Downloading https://homebrew.bintray.com/bottles/mecab-ipadic-2.7.0-20070801.mojave.bottle.tar.gz

Already downloaded: /Users/rob/Library/Caches/Homebrew/downloads/[...]--mecab-ipadic-2.7.0-20070801.mojave.bottle.tar.gz

==> Downloading https://homebrew.bintray.com/bottles/msgpack-3.2.1.mojave.bottle.tar.gz

Already downloaded: /Users/rob/Library/Caches/Homebrew/downloads/[...]--msgpack-3.2.1.mojave.bottle.tar.gz

==> Downloading https://homebrew.bintray.com/bottles/pcre-8.44.mojave.bottle.tar.gz

Already downloaded: /Users/rob/Library/Caches/Homebrew/downloads/[...]--pcre-8.44.mojave.bottle.tar.gz

==> Downloading https://homebrew.bintray.com/bottles/groonga-10.0.2.mojave.bottle.tar.gz

Already downloaded: /Users/rob/Library/Caches/Homebrew/downloads/[...]--groonga-10.0.2.mojave.bottle.tar.gz

==> Downloading https://downloads.mariadb.com/MariaDB/mariadb-10.4.13/source/mariadb-10.4.13.tar.gz

######################################################################## 100.0%

==> Installing dependencies for mariadb: cmake, pkg-config, mecab, mecab-ipadic, msgpack, pcre and groonga

==> Installing mariadb dependency: cmake

==> Pouring cmake-3.17.2.mojave.bottle.tar.gz

==> Caveats

Emacs Lisp files have been installed to:

/usr/local/share/emacs/site-lisp/cmake

==> Summary

🍺 /usr/local/Cellar/cmake/3.17.2: 6,156 files, 58.1MB

==> Installing mariadb dependency: pkg-config

==> Pouring pkg-config-0.29.2\_3.mojave.bottle.tar.gz

🍺 /usr/local/Cellar/pkg-config/0.29.2\_3: 11 files, 623.6KB

==> Installing mariadb dependency: mecab

==> Pouring mecab-0.996.mojave.bottle.3.tar.gz

🍺 /usr/local/Cellar/mecab/0.996: 20 files, 4.2MB

==> Installing mariadb dependency: mecab-ipadic

==> Pouring mecab-ipadic-2.7.0-20070801.mojave.bottle.tar.gz

==> Caveats

To enable mecab-ipadic dictionary, add to /usr/local/etc/mecabrc:

dicdir = /usr/local/lib/mecab/dic/ipadic

==> Summary

🍺 /usr/local/Cellar/mecab-ipadic/2.7.0-20070801: 16 files, 50.6MB

==> Installing mariadb dependency: msgpack

==> Pouring msgpack-3.2.1.mojave.bottle.tar.gz

🍺 /usr/local/Cellar/msgpack/3.2.1: 757 files, 5.2MB

==> Installing mariadb dependency: pcre

==> Pouring pcre-8.44.mojave.bottle.tar.gz

🍺 /usr/local/Cellar/pcre/8.44: 204 files, 5.5MB

==> Installing mariadb dependency: groonga

==> Pouring groonga-10.0.2.mojave.bottle.tar.gz

🍺 /usr/local/Cellar/groonga/10.0.2: 886 files, 39.5MB

==> Installing mariadb

==> cmake . -DMYSQL\_DATADIR=/usr/local/var/mysql -DINSTALL\_INCLUDEDIR=include/mysql -DINSTALL\_MANDIR=share/man -DINSTALL\_DOCDIR=share/d

==> make

==> make install

==> Not running post\_install as we're building a bottle

You can run it manually using `brew postinstall mariadb`

==> Caveats

A "/etc/my.cnf" from another install may interfere with a Homebrew-built server starting up correctly.

MySQL is configured to only allow connections from localhost by default

To have launchd start mariadb now and restart at login:

brew services start mariadb

Or, if you don't want/need a background service you can just run:

mysql.server start

==> Summary

🍺 /usr/local/Cellar/mariadb/10.4.13: 737 files, 170.0MB, built in 8 minutes 53 seconds

Removing: /usr/local/Cellar/mariadb/10.2.14... (641 files, 168.6MB)

Removing: /usr/local/Cellar/mariadb/10.3.10... (652 files, 173.3MB)

Removing: /Users/rob/Library/Caches/Homebrew/mariadb--10.3.10.tar.gz... (67.2MB)

==> Checking for dependents of upgraded formulae...

==> No dependents found!

==> Caveats

==> cmake

Emacs Lisp files have been installed to:

/usr/local/share/emacs/site-lisp/cmake

==> mecab-ipadic

To enable mecab-ipadic dictionary, add to /usr/local/etc/mecabrc:

dicdir = /usr/local/lib/mecab/dic/ipadic

==> mariadb

A "/etc/my.cnf" from another install may interfere with a Homebrew-built

server starting up correctly.

MySQL is configured to only allow connections from localhost by default

To have launchd start mariadb now and restart at login:

brew services start mariadb

Or, if you don't want/need a background service you can just run:

mysql.server start

7. Run the database installer

Run mysql\_install\_db. Follow on-screen instructions to upgrade if necessary to upgrade a previously installed version.

$ mysql\_install\_db

WARNING: The host 'robs-MacBook-Pro-2.local' could not be looked up with /usr/local/Cellar/mariadb/10.4.13/bin/resolveip.

This probably means that your libc libraries are not 100 % compatible

with this binary MariaDB version. The MariaDB daemon, mysqld, should work normally with the exception that host name resolving will not work.

This means that you should use IP addresses instead of hostnames

when specifying MariaDB privileges !

mysql.user table already exists!

Run mysql\_upgrade, not mysql\_install\_db

$ mysql\_upgrade

Phase 1/7: Checking and upgrading mysql database

Processing databases

...

Phase 7/7: Running 'FLUSH PRIVILEGES'

OK

8. Start MariaDB

Run mysql.server start.

$ mysql.server start

Starting MySQL

. SUCCESS!

9. Secure the installation

If you are installing MariaDB 10.4.6 or later:  
Run mariadb-secure-installation.  
If you are installing an earlier version of MariaDB:  
Run mysql\_secure\_installation.

NOTE: If you are unsure about using unix\_socket, do not enable it when asked.  
NOTE: Set a root password even if the on-screen instructions tell you it is safe not to do so.

$ 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

haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):

OK, successfully used password, moving on

Setting the root password or using the unix\_socket ensures that nobody can log into the MariaDB root user without the proper authorisation.

Enable unix\_socket authentication? [Y/n] n

... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y

New password:

Re-enter new password:

Password updated successfully!

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] 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] 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] 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!

10. Connect to MariaDB

Run mariadb -u root -p.

If you’ve installed an older version of mariadb you may need to use “mysql -u root -p” in the above command.

**Importing the Database created by Group 5**

**The following is taken from DigitalOcean**

To import an existing dump file into MySQL or MariaDB, you will have to create the new database. This is where the contents of the dump file will be imported.

First, log in to the database as root or another user with sufficient privileges to create new databases:

mysql -u root -p

This will bring you into the MySQL shell prompt. Next, create a new database with the following command. In this example, the new database is called new\_database:

CREATE DATABASE new\_database;

You’ll see this output confirming that it was created.

Output

Query OK, 1 row affected (0.00 sec)

Then exit the MySQL shell by pressing CTRL+D. From the normal command line, you can import the dump file with the following command:

mysql -u username -p new\_database < data-dump.sql

username is the username you can log in to the database with

newdatabase is the name of the freshly created database

data-dump.sql is the data dump file to be imported, located in the current directory

If the command runs successfully, it won’t produce any output. If any errors occur during the process, mysql will print them to the terminal instead. You can check that the database was imported by logging in to the MySQL shell again and inspecting the data. This can be done by selecting the new database with USE new\_database and then using SHOW TABLES; or a similar command to look at some of the data.

**If changes to the database are required, the user manuals on the MariaDB website provides full oversite of the full suite of functions available. Please reference this manual post-handover for changes to the system as required.**

**Link - https://mariadb.com/kb/en/documentation/**