

How To Install MariaDB 10.4 on Alpine Linux

By

[Josphat Mutai](#)

-

2512

[0](#)

In our previous guide we discussed on the [installation of MariaDB database server on Amazon Linux](#). Today's tutorial is geared towards setting up MariaDB Database Server on Alpine Linux instance. The machine can be running on Virtualized environment powered by KVM, VirtualBox, VMware, Parallels, Hyper-V or Cloud provider that provides alpine Linux servers. For those new to MariaDB it is a relational database management system that is developed by the community, but was forked from MySQL. The software is open source and released under the GNU General Public License.

As of this article update the latest stable release of MariaDB database server is version 10.5 but the version on the Alpine Linux repositories is 10.4. This guide will capture MariaDB 10.4 installation on Alpine Linux. Once a much newer release is available the title of this article will be updated to capture it.

Install MariaDB 10.4 on Alpine Linux

Before we even begin the actual installation we need to make sure our system can connect to the OS repositories for package installation.

Update APK application list cache:

```
$ sudo apk update
fetch http://nl.alpinelinux.org/alpine/v3.7/community/x86_64/APKINDEX.tar.gz
fetch http://nl.alpinelinux.org/alpine/edge/main/x86_64/APKINDEX.tar.gz
fetch http://nl.alpinelinux.org/alpine/edge/community/x86_64/APKINDEX.tar.gz
v3.7.3-183-gcc9ad2b48d [http://nl.alpinelinux.org/alpine/v3.7/community]
v20200626-271-gcbcf4f191f [http://nl.alpinelinux.org/alpine/edge/main]
v20200626-267-g3745709556 [http://nl.alpinelinux.org/alpine/edge/community]
OK: 16265 distinct packages available
```

Check for package names that match mariadb:

```
$ sudo apk search mariadb
mariadb-static-10.4.13-r1
mariadb-server-utils-10.4.13-r1
mariadb-connector-c-dev-3.1.8-r1
mariadb-bench-10.4.13-r1
mariadb-client-10.4.13-r1
mariadb-10.4.13-r1
mariadb-backup-10.4.13-r1
mariadb-doc-10.4.13-r1
mariadb-openrc-10.4.13-r1
mariadb-dev-10.4.13-r1
acf-mariadb-0.2.0-r2
mariadb-embedded-dev-10.4.13-r1
mariadb-test-10.4.13-r1
mariadb-connector-c-3.1.8-r1
```

```
mariadb-embedded-10.4.13-r1
mariadb-mysqld-10.4.13-r1
mariadb-plugin-rocksdb-10.4.13-r1
pdns-backend-mariadb-4.2.2-r0
mariadb-common-10.4.13-r1
```

Install MariaDB 10.4 on Alpine Linux

From the list we can confirm availability of MariaDB 10.4 on the Alpine Linux repositories.

```
$ sudo apk add mariadb mariadb-common mariadb-client
(1/12) Installing mariadb-common (10.4.13-r1)
(2/12) Installing libaio (0.3.112-r1)(3/11) Installing libcrypto1.1 (1.1.1g-r0)
ERROR: libcrypto1.1-1.1.1g-r0: trying to overwrite etc/ssl/openssl.cnf owned by
libressl2.6-libcrypto-2.6.4-r0.
(4/12) Installing libgcc (9.3.0-r3)
(5/12) Installing linux-pam (1.3.1-r4)
(6/12) Installing pcre (8.44-r0)
(7/12) Installing libssl1.1 (1.1.1g-r0)
(8/12) Installing libstdc++ (9.3.0-r3)
(9/12) Installing libxml2 (2.9.10-r4)
(10/12) Installing mariadb (10.4.13-r1)
Executing mariadb-10.4.13-r1.pre-install
(11/12) Installing mariadb-openrc (10.4.13-r1)
(12/12) Installing mariadb-client (10.4.13-r1)
Executing busybox-1.27.2-r8.trigger
1 error; 246 MiB in 62 packages
```

After installation you need to initialize the main mysql database, and the data dir as standardized to **/var/lib/mysql**. This can be done by executing the rc script below.

```
$ sudo /etc/init.d/mariadb setup
* Creating a new MySQL database ...
Installing MariaDB/MySQL system tables in '/var/lib/mysql' ...OK
```

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

Two all-privilege accounts were created.
One is root@localhost, it has no password, but you need to
be system 'root' user to connect. Use, for example, sudo mysql
The second is mysql@localhost, it has no password either, but
you need to be the system 'mysql' user to connect.
After connecting you can set the password, if you would need to be
able to connect as any of these users with a password and without sudo

Start MariaDB database service on Alpine Linux.

Alpine Linux uses [OpenRC](#) for its init system. The service can be started using the command:

```
$ sudo rc-service mariadb start
* Starting mariadb ...
200702 21:09:32 mysqld_safe Logging to syslog.200702 21:09:32 mysqld_safe
Starting mysqld daemon with databases from /var/lib/mysql
[ ok ]
```

Also set the service to be started at system boot.

```
$ sudo rc-update add mariadb default
* service mariadb added to runlevel default
```

Check service status with *rc-status* command:

```
$ rc-status default
Runlevel: default
  crond
[ started ]
  chronyd
[ started ]
  mariadb
[ started ]
  sshd
[ started ]
  acpid
[ started ]
```

Secure MariaDB database server on Alpine Linux

Now that the service is running the only pending item is securing it. Run the command below to set root password, remove test database and disallow remote login as root user.

```
$ 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 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.

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

```
Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
... Success!
```

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!
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] 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!

If you try accessing the database console without providing password an access denied error will be shown.

```
$ mysql -u root <ENTER>
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
```

Now try again but this time providing the password with **-p** command option.

```
$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 19
Server version: 10.4.13-MariaDB MariaDB Server
```

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)]>
```

Test MariaDB database installation on Alpine Linux

Create test database and user.

```
MariaDB [(none)]> CREATE DATABASE testdb;
Query OK, 1 row affected (0.000 sec)
```

```
MariaDB [(none)]> SHOW DATABASES;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
```

```
| performance_schema |
| testdb              |
+-----+
4 rows in set (0.000 sec)
```

Create a user and grant access to the database created.

```
MariaDB [(none)]> CREATE USER testuser@localhost IDENTIFIED BY "Str0ngDBP@ss";
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [(none)]> GRANT ALL ON testdb.* TO testuser@localhost;
Query OK, 0 rows affected (0.003 sec)
```

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
```

Quit from MariDB console and login as *testuser*.

```
MariaDB [(none)]> QUIT
Bye
```

```
$ mysql -u testuser -p'Str0ngDBP@ss'
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 21
Server version: 10.4.13-MariaDB MariaDB Server
```

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)]>
```

List the databases that the user has access to.

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database          |
+-----+
| information_schema |
| testdb            |
+-----+
2 rows in set (0.000 sec)
```

```
MariaDB [(none)]> QUIT
Bye
```

We have successfully installed, configured and started MariaDB database server on Alpine Linux virtual machine. We also demonstrated how you can create a database, user and grant permissions and we confirmed the user was able to see database when permissions were granted. You can now proceed to learn more about SQL and hack your projects with confidence.

<https://techviewleo.com/how-to-install-mariadb-on-alpine-linux/>