

# How To Install the Latest MySQL on Debian 9



Posted September 5, 2018  28.7k [MYSQL](#) [DATABASES](#) [DEBIAN 9](#)

By: Justin Ellingwood By: Brian Boucheron

Not using **Debian 9**? Choose a different version:

Ubuntu 18.04	<a href="#">&gt;</a>
Ubuntu 16.04	<a href="#">&gt;</a>
Automated: Docker	<a href="#">request</a>

## Introduction

MySQL is a prominent open source database management system used to store and retrieve data for a wide variety of popular applications. MySQL is the **M** in the *LAMP* stack, a commonly used set of open source software that also includes Linux, the Apache web server, and the PHP programming language.

In Debian 9, MariaDB, a community fork of the MySQL project, is packaged as the default MySQL variant. While, MariaDB works well in most cases, if you need features found only in Oracle's MySQL, you can install and use packages from a repository maintained by the MySQL developers.

To install the latest version of MySQL, we'll add this repository, install the MySQL software itself, secure the install, and finally we'll test that MySQL is running and responding to commands.

## Prerequisites

Before starting this tutorial, you will need:

- One Debian 9 server set up by following [this initial server setup guide](#), including a non-**root** user with `sudo` privileges and a firewall.

# Step 1 — Adding the MySQL Software Repository

The MySQL developers provide a `.deb` package that handles configuring and installing the official MySQL software repositories. Once the repositories are set up, we'll be able to use Ubuntu's standard `apt` command to install the software. We'll download this `.deb` file with `wget` and then install it with the `dpkg` command.

First, load the MySQL download page in your web browser. Find the **Download** button in the lower-right corner and click through to the next page. This page will prompt you to log in or sign up for an Oracle web account. We can skip that and instead look for the link that says **No thanks, just start my download**. Right-click the link and select **Copy Link Address** (this option may be worded differently, depending on your browser).

Now we're going to download the file. On your server, move to a directory you can write to. Download the file using `wget`, remembering to paste the address you just copied in place of the highlighted portion below:

```
$ cd /tmp
$ wget https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb
```

The file should now be downloaded in our current directory. List the files to make sure:

```
$ ls
```

You should see the filename listed:

Output

```
mysql-apt-config_0.8.10-1_all.deb
. . .
```

Now we're ready to install:

```
$ sudo dpkg -i mysql-apt-config*
```

`dpkg` is used to install, remove, and inspect `.deb` software packages. The `-i` flag indicates that we'd like to install from the specified file.

During the installation, you'll be presented with a configuration screen where you can specify which version of MySQL you'd prefer, along with an option to install repositories for other MySQL-related tools. The defaults will add the repository information for the latest stable version of MySQL and nothing else. This is what we want, so use the down arrow to navigate to the `Ok` menu option and hit `ENTER`.

The package will now finish adding the repository. Refresh your `apt` package cache to make the new software packages available:

```
$ sudo apt update
```

Now that we've added the MySQL repositories, we're ready to install the actual MySQL server software. If you ever need to update the configuration of these repositories, just run `sudo dpkg-reconfigure mysql-apt-config`, select new options, and then `sudo apt-get update` to refresh your package cache.

## Step 2 — Installing MySQL

Having added the repository and with our package cache freshly updated, we can now use `apt` to install the latest MySQL server package:

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

`apt` will look at all available `mysql-server` packages and determine that the MySQL provided package is the newest and best candidate. It will then calculate package dependencies and ask you to approve the installation. Type `y` then `ENTER`. The software will install.

You will be asked to set a **root** password during the configuration phase of the installation. Choose and confirm a secure password to continue. Next, a prompt will appear asking for you to select a default authentication plugin. Read the display to understand the choices. If you are not sure, choosing **Use Strong Password Encryption** is safer.

MySQL should be installed and running now. Let's check using `systemctl`:

```
$ sudo systemctl status mysql
```

### Output

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2018-09-05 15:58:21 UTC; 30s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
 Main PID: 12805 (mysqld)
    Status: "SERVER_OPERATING"
   CGroup: /system.slice/mysql.service

           └─12805 /usr/sbin/mysqld
```

```
Sep 05 15:58:15 mysql1 systemd[1]: Starting MySQL Community Server...
```

```
Sep 05 15:58:21 mysql1 systemd[1]: Started MySQL Community Server.
```

The **Active: active (running)** line means MySQL is installed and running. Now we'll make the installation a little more secure.

## Step 3 — Securing MySQL

MySQL comes with a command we can use to perform a few security-related updates on our new install. Let's run it now:

```
$ mysql_secure_installation
```

This will ask you for the MySQL **root** password that you set during installation. Type it in and press **ENTER**. Now we'll answer a series of yes or no prompts. Let's go through them:

First, we are asked about the **validate password plugin**, a plugin that can automatically enforce certain password strength rules for your MySQL users. Enabling this is a decision you'll need to make based on your individual security needs. Type **y** and **ENTER** to enable it, or just hit **ENTER** to skip it. If enabled, you will also be prompted to choose a level from 0–2 for how strict the password validation will be. Choose a number and hit **ENTER** to continue.

Next you'll be asked if you want to change the **root** password. Since we just created the password when we installed MySQL, we can safely skip this. Hit **ENTER** to continue without updating the password.

The rest of the prompts can be answered **yes**. You will be asked about removing the **anonymous** MySQL user, disallowing remote **root** login, removing the **test** database, and reloading privilege tables to ensure the previous changes take effect properly. These are all a good idea. Type **y** and hit **ENTER** for each.

The script will exit after all the prompts are answered. Now our MySQL installation is reasonably secured. Let's test it again by running a client that connects to the server and returns some information.

## Step 4 — Testing MySQL

**mysqladmin** is a command line administrative client for MySQL. We'll use it to connect to the server and output some version and status information:

```
$ mysqladmin -u root -p version
```

The **-u root** portion tells **mysqladmin** to log in as the MySQL **root** user, **-p** instructs the client to ask for a password, and **version** is the actual command we want to run.

The output will let us know what version of the MySQL server is running, its uptime, and some other status information:

Output

mysqladmin Ver 8.0.12 for Linux on x86\_64 (MySQL Community Server - GPL)  
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.

Server version 8.0.12  
Protocol version 10  
Connection Localhost via UNIX socket  
UNIX socket /var/run/mysqld/mysqld.sock  
Uptime: 6 min 42 sec

Threads: 2 Questions: 12 Slow queries: 0 Opens: 123 Flush tables: 2 Open tables: 99 Queries per second avg: 0.000

If you received similar output, congrats! You've successfully installed the latest MySQL server and secured it.

## Conclusion

You've now completed a basic install of the latest version of MySQL, which should work for many popular applications.

By: Justin Ellingwood    By: Brian Boucheron     Upvote (3)     Subscribe     Share



We just made it easier for you to deploy faster.

[TRY FREE](#)

### Related Tutorials

[How To Ensure Code Quality with SonarQube on Ubuntu 18.04](#)

How To Sync and Share Your Files with Seafile on Ubuntu 18.04

How To Troubleshoot Issues in MySQL

How To Set Up a Remote Database to Optimize Site Performance with MySQL on Ubuntu 18.04

How To Set Up Laravel, Nginx, and MySQL with Docker Compose

## 1 Comment

Leave a comment...

Log In to Comment

 [caslavsabani](#) *October 16, 2018*



0

Awesome article!!! Thank you!!!! I have installed MySQL 8.0.12 on Debian 9 with ease. It was awesome thanks.



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



