

Install MySQL on CentOS

Introduction

MySQL is an open-source database management system, commonly installed as part of the popular LEMP (Linux, Nginx, MySQL/MariaDB, PHP/Python/Perl) stack. It uses a relational database and SQL (Structured Query Language) to manage its data.

This tutorial will explain how to install MySQL version 5.7 on a CentOS server.

Step 1 — Installing MySQL

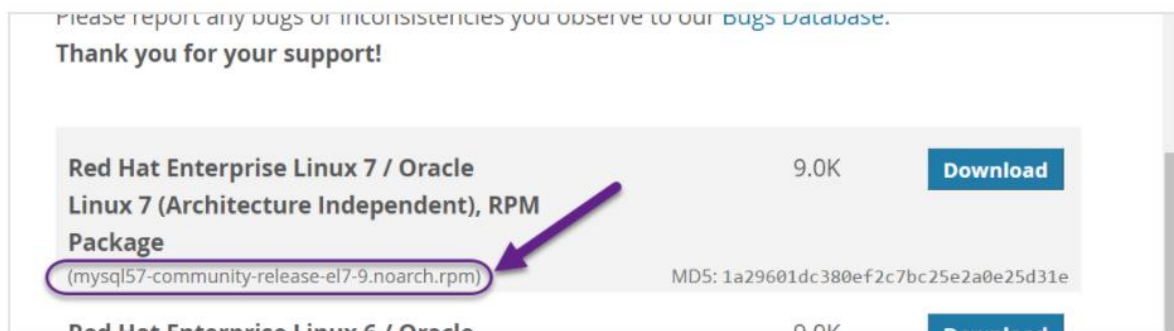
As mentioned in the introduction, the Yum command to install MySQL in fact installs MariaDB. To install MySQL, we'll need to visit the MySQL community Yum Repository which provides packages for MySQL.

In a web browser, visit:

```
https://dev.mysql.com/downloads/repo/yum/
```

Note that the prominent Download links don't lead directly to the files. Instead, they lead to a subsequent page where you're invited to log in or sign up for an account. If you don't want to create an account, you can locate the text "No thanks, just start my download", then right-click and copy the link location, or you can edit the version number in the commands below.

Locate the desired version, and update it as needed in the link below:

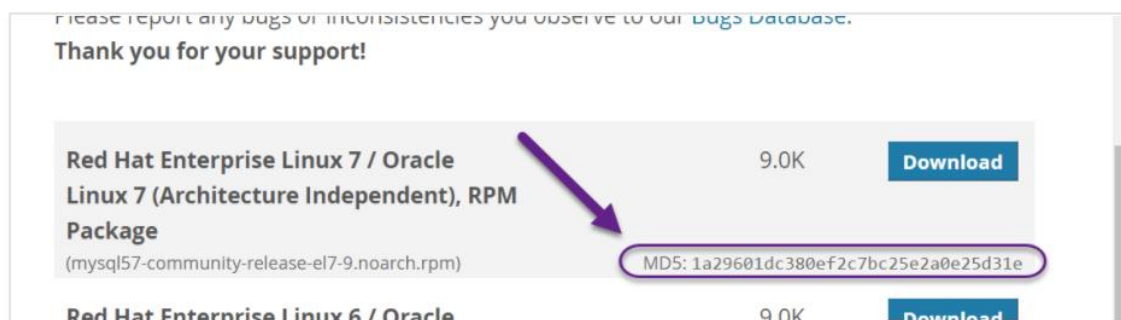


```
wget https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm
```

Once the rpm file is saved, we will verify the integrity of the download by running `md5sum` and comparing it with the corresponding MD5 value listed on the site:

```
md5sum mysql57-community-release-el7-9.noarch.rpm
Output
1a29601dc380ef2c7bc25e2a0e25d31e  mysql57-community-release-el7-9.noarch.rpm
```

Compare this output with the appropriate MD5 value on the site:



Now that we've verified that the file wasn't corrupted or changed, we'll install the package:

```
sudo rpm -ivh mysql57-community-release-el7-9.noarch.rpm
```

This adds two new MySQL yum repositories, and we can now use them to install MySQL server:

```
sudo yum install -y mysql-server
```

Press `y` to confirm that you want to proceed. Since we've just added the package, we'll also be prompted to accept its GPG key. Press `y` to download it and complete the install.

Step 2 — Starting MySQL

We'll start the daemon with the following commands:

```
wget https://raw.githubusercontent.com/gdraheim/docker-systemctl-replacement/master/files/docker/systemctl.py -O /usr/local/bin/systemctl
```

```
sudo /usr/local/bin/systemctl start mysqld
```

`systemctl` doesn't display the outcome of all service management commands, so to be sure we succeeded, we'll use the following command:

```
sudo /usr/local/bin/systemctl status mysqld
```

If MySQL has successfully started, the output should contain `Active: active (running)` and the final line should look something like:

```
Dec 01 19:02:20 centos-512mb-sfo2-02 systemd[1]: Started MySQL Server.
```

Note: MySQL is automatically enabled to start at boot when it is installed. You can change that default behavior with

```
sudo /usr/local/bin/systemctl disable mysqld
```

During the installation process, a temporary password is generated for the MySQL root user. Locate it in the `mysqld.log` with this command:

```
sudo grep 'temporary password' /var/log/mysqld.log
```

Output

```
2016-12-01T00:22:31.416107Z 1 [Note] A temporary password is generated for  
root@localhost: <your-password>
```

Make note of the password, which you will need in the next step to secure the installation and where you will be forced to change it. The default password policy requires 12 characters, with at least one uppercase letter, one lowercase letter, one number and one special character.

Step 3 — Configuring MySQL

MySQL includes a security script to change some of the less secure default options for things like remote root logins and sample users.

Use this command to run the security script.

```
sudo mysql_secure_installation
```

This will prompt you for the default root password. As soon as you enter it, you will be required to change it.

Output

```
The existing password for the user account root has expired. Please set a new password.
```

New password:

Enter a new 12-character password that contains at least one uppercase letter, one lowercase letter, one number and one special character. Re-enter it when prompted.

You'll receive feedback on the strength of your new password, and then you'll be immediately prompted to change it again. Since you just did, you can confidently say **No**:

Output

```
Estimated strength of the password: 100
```

```
Change the password for root ? (Press y|Y for Yes, any other key for No) :
```

After we decline the prompt to change the password again, we'll press **Y** and then **ENTER** to all the subsequent questions in order to remove anonymous users, disallow remote root login, remove the test database and access to it, and reload the privilege tables.

Now that we've secured the installation, let's test it.

Step 4 — Testing MySQL

We can verify our installation and get information about it by connecting with the `mysqladmin` tool, a client that lets you run administrative commands. Use the following command to connect to MySQL as **root** (`-u root`), prompt for a password (`-p`), and return the version.

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

You should see output similar to this:

Output

```
mysqladmin Ver 8.42 Distrib 5.7.16, for Linux on x86_64
```

Copyright (c) 2000, 2016, 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          5.7.16
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/lib/mysql/mysql.sock
Uptime:                 2 min 17 sec
Threads: 1  Questions: 6  Slow queries: 0  Opens: 107  Flush tables: 1  Open
tables: 100  Queries per second avg: 0.043
```

This indicates your installation has been successful.

Server: localhost

DatabaseName: mysql

Username: root

Password: <your-password>

Port number: 3306