

MySQL installation guide (Linux)

MySQL is a RDBMS software which use SQL like syntax to manage databases. Nowadays most of the major linux distributions come with maria db preinstalled, which is an open source drop in replacement for MySQL. I'll be writing about some ways to install MySQL in linux based operating systems,

XAMPP

Xampp is a popular tool which is an open source cross-platform web server solution stack package developed by Apache friends. It can be installed via the official website's installer. Here a `.run` file will be downloaded which can be installed by executing from a terminal. But it is not recommended to install in this way.

The most recommended way is to search for a similar package in distros native package manager. For example, in Arch Linux the package is available through AUR (Arch User Repository). Here's the git-clone URL,

- <https://aur.archlinux.org/packages/xampp>

To install it, we can use a AUR wrapper like `yay`. To do so, use the following command to query and install the latest version of `xampp`.

- `yay xampp`

After installing open the app, head over to the second tab and start database and web server. Web UI will be available under `localhost`.

Podman Container

One another good way to install MySQL is to use a podman or docker container. I personally prefer podman so I will be writing about it. Installing a container running only MySQL is pretty much easy. We just have to grab the image and run it in a container. It's volume will be created automatically. Or if we also want to include a phpmyadmin web app to manage our image then we actually have to use a pod to contain two different containers.

MySQL image

To setting up MySQL image, we can pull it from dockerhub. The command will be like,

- `podman pull mysql`

then, we can start and run our image with the following command,

- `podman run -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=tree --name mysql-db mysql:latest`

Here our root password is defined as tree by the environment variable `MYSQL_ROOT_PASSWORD`.

And if we try to do list running process we can execute,

```
- podman ps
```

It will see our image up and running. Now let's actually enter to our server!

```
- podman exec -it mysql-db mysql -u root -p
```

Let's run a command to verify,

```
- show databases;
```

It'll list all databases.

Now with `localhost:3306` you can access this database from mysql workbench or other clients.

Phpmyadmin image

Phpmyadmin is a web UI for managing MySQL databases. Let's pull it first,

```
- podman pull phpmyadmin
```

Now if run this image we won't be able to access another image (MySQL) because there's no connection in between them. So we will be using podman pod. Let's create a podman pod,

```
- podman pod create --name mysql -p 8080:8080 3306:3306
```

If we have previously created an image as per this guide and that is up and running, try the following command to stop and delete,

```
- podman pod create --name mysql && podman rm mysql-db
```

Now let's start our mysql server under this pod,

```
- podman run -d -e MYSQL_ROOT_PASSWORD=tree --pod mysql --name mysql-db mysql:latest
```

And finally let's open our phpmyadmin with this pod,

```
- podman run --name phpmyadmin -e PMA_ARBITRARY=1 -d --pod mysql phpmyadmin
```

It will be available under port 8080, as like we defined earlier. So let's head over to,

```
- http://localhost:8080/
```

Here, our,

Server = localhost:3306

Username = root

Password = tree

This can be also done graphically with the help of podman desktop.