**How to Install LAMP Stack with PhpMyAdmin in Ubuntu 18.04**

by [Aaron Kili](https://www.tecmint.com/author/aaronkili/) | Published: June 15, 2018 | Last Updated: June 18, 2018

**Download Your Free eBooks NOW** - [10 Free Linux eBooks for Administrators](https://www.tecmint.com/10-useful-free-linux-ebooks-for-newbies-and-administrators/" \t "_blank) | [4 Free Shell Scripting eBooks](https://www.tecmint.com/free-linux-shell-scripting-books/" \t "_blank)

A **LAMP** stack is composed of packages such as **Apache**, **MySQL/MariaDB** and **PHP** installed on a Linux system environment for hosting websites and apps.

**Read Also**: [Install Apache, MariaDB, PHP and PhpMyAdmin in Ubuntu 18.04](https://www.tecmint.com/install-lamp-with-phpmyadmin-in-ubuntu-18-04/" \t "_blank)

**PhpMyAdmin** is a free, open source, well known, fully-featured, and intuitive web-based frontend for administering **MySQL** and **MariaDB** database. It supports various database operations, and has many features that allow you to easily manage your databases from a web interface; such as importing and exporting data in various formats, generating complex and useful queries using Query-by-example (QBE), administering multiple servers, and much more.

**Requirements:**

1. Minimal Ubuntu 18.04 server Installation.
2. Access to server via SSH (if you don’t have direct access).
3. Root user privileges or use [sudo command](https://www.tecmint.com/su-vs-sudo-and-how-to-configure-sudo-in-linux/" \t "_blank) to run all commands.

In this article, we will explain how to install **LAMP** stack with **PhpMyAdmin** in **Ubuntu 18.04**.

**Step 1: Install Apache Web Server on Ubuntu 18.04**

**1.** First start by updating your software packages and then install **Apache** web server using following commands.

$ sudo apt update

$ sudo apt install apache2

**2.** After the installation process is complete, the apache service should start automatically and will be enabled to start at system boot time, you can check if it’s up and running using following command.

$ sudo systemctl status apache2

**3.** If you have a system firewall enabled and running, you need to open the ports **80** and **443** to allow client connection requests to apache web server via **HTTP** and **HTTPS** respectively, then reload the firewall settings as shown.

$ sudo ufw allow 80/tcp

$ sudo ufw allow 443/tcp

$ sudo ufw reload

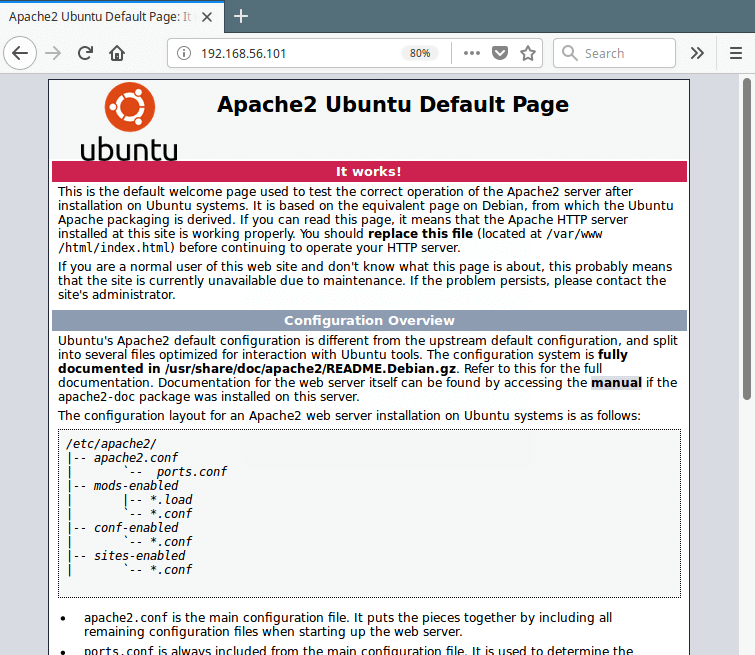
**4.** Now verify your Apache installation by testing default test page at the below URL from a web browser.

http://domain\_name/

OR

http://SERVER\_IP/

If you see the apache default web page, it means your installation is working fine.

[](https://www.tecmint.com/wp-content/uploads/2018/06/Check-Apache-Web-Page.png)

Check Apache Web Page

**Step 2: Install MariaDB on Ubuntu 18.04**

**5.** Now install **MariaDB**, is a free, open source database management system forked from **MySQL** and it is a community developed project being led by the original developers of **MySQL**.

$ sudo apt install mariadb-server mariadb-client

**6.** The **MariaDB** services should start automatically after installation, check its status to ensure that it is up and running.

$ sudo systemctl status mysql

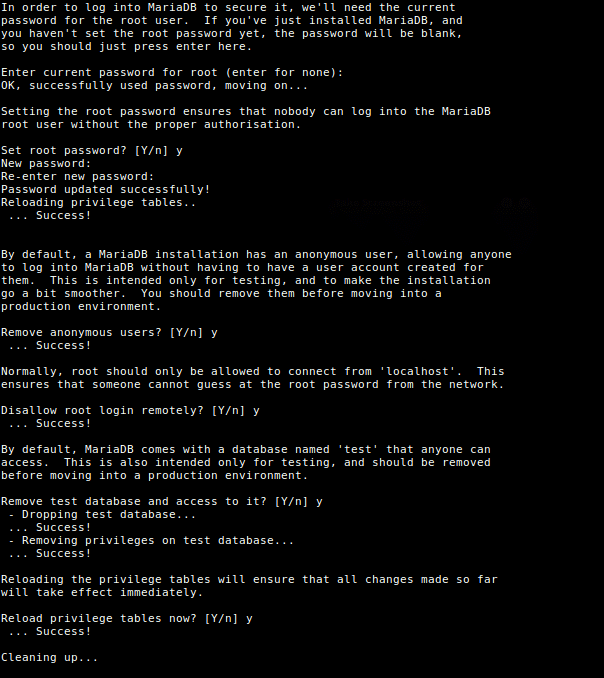
**7.** The **MariaDB** installation is not secure by default, you need to execute a security script that comes with the package. You will be asked to set a **root password** to ensure that nobody can log into the MariaDB.

$ sudo mysql\_secure\_installation

Once you execute the script, it will ask you to enter current password for root (**enter for none**):

Then enter yes/y to the following security questions:

* Set root password? [Y/n]: y
* Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
* Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
* Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
* Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y

[](https://www.tecmint.com/wp-content/uploads/2018/06/Secure-Mariadb-Installation.png)

Secure Mariadb Installation in Ubuntu 18.04

**Step 3: Install PHP on Ubuntu 18.04**

**8.** **PHP** is one of the most widely used server side scripting language used to generate dynamic content on websites and apps. You can install **PHP** (default version is **PHP 7.2**) and other modules for web deployments using following command.

$ sudo apt install php php-common php-mysql php-gd php-cli

**9.** Once PHP installed, you can test your PHP setup by creating a simple info.php page in your web server document root, using this single command.

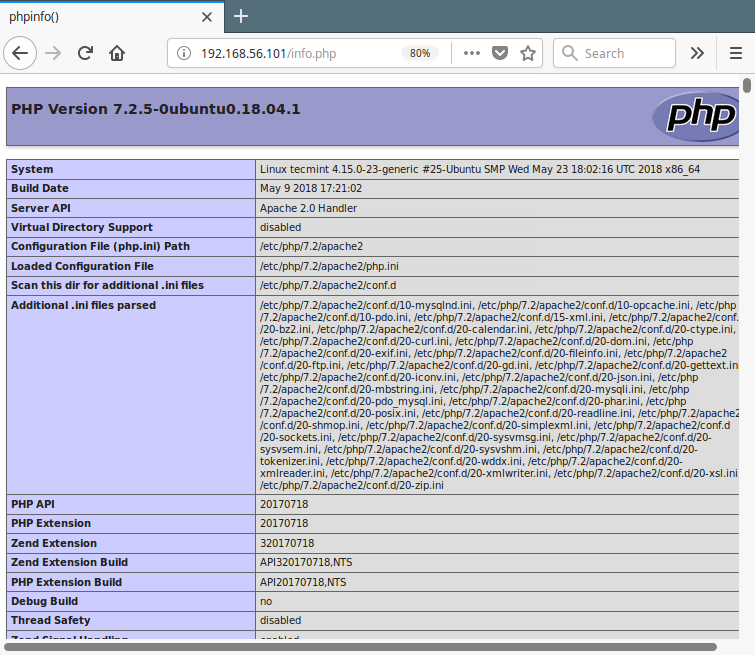
$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

**10.** Then open a web browser, and enter this URL to view the php information page.

http://domain\_name/info.php

OR

http://SERVER\_IP/info.php

[](https://www.tecmint.com/wp-content/uploads/2018/06/Test-PHP-Info.png)

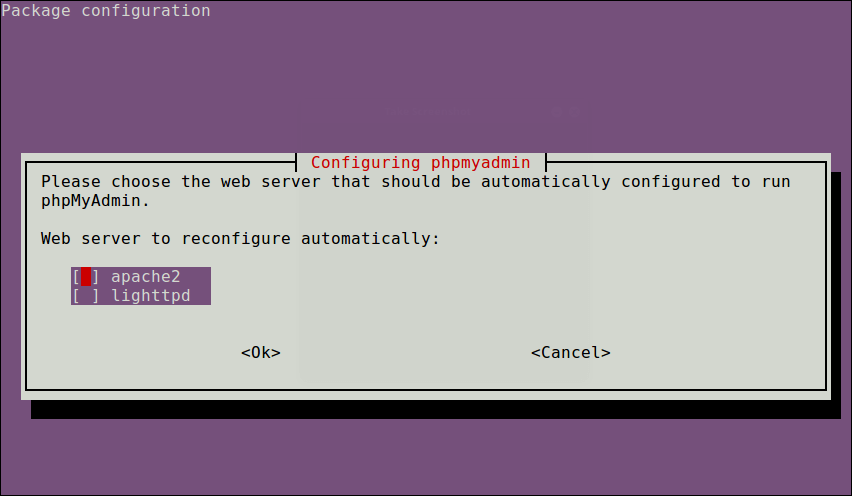
Test PHP Info in Ubuntu 18.04

**Step 4: Install PhpMyAdmin on Ubuntu 18.04**

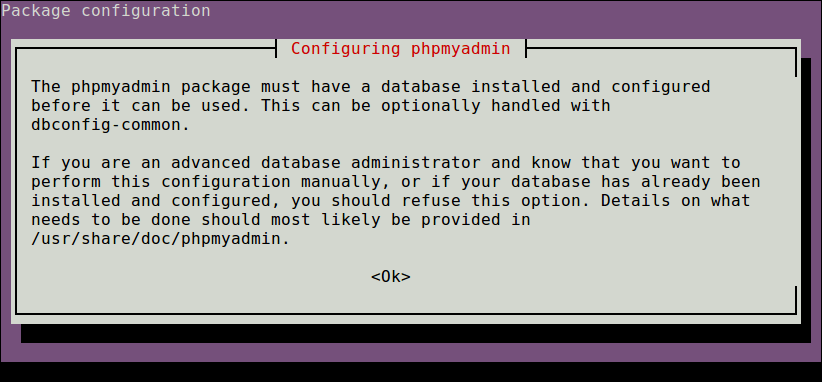
**11.** Finally, you can install **phpMyAdmin** for administrating **MySQL/MariaDB** databases from the comfort of a web browser, by running following command.

$ sudo apt install phpmyadmin

Through the package installation process, you will be asked to choose the web server that should be automatically configured to run phpMyAdmin, select **apache** by pressing the space bar and press **Enter**.

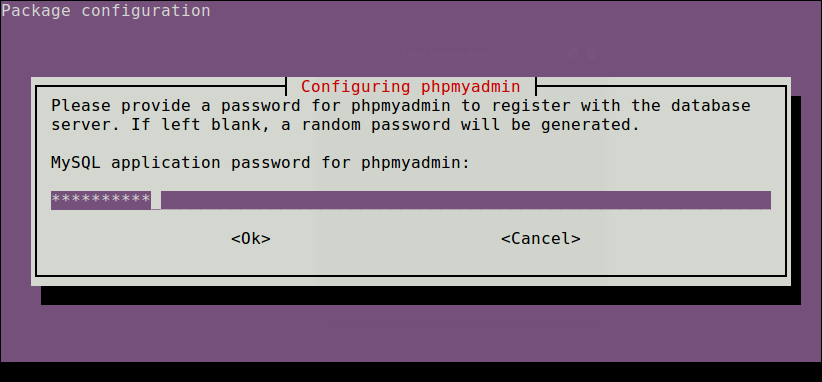
[](https://www.tecmint.com/wp-content/uploads/2018/06/Select-PhpMyAdmin-Web-Server.png)

Select PhpMyAdmin Web Server

[](https://www.tecmint.com/wp-content/uploads/2018/06/PhpMyAdmin-Configuration.png)

PhpMyAdmin Configuration

**12.** Next, enter the password for the **MySQL/MariaDB** administrative user so the installer can create database for phpmyadmin.

[](https://www.tecmint.com/wp-content/uploads/2018/06/Set-Password-for-Phpmyadmin.png)

Set Password for Phpmyadmin

**13.** Once everything installed, you can now restart the apache2 service to effect the recent changes.

$ sudo systemctl restart apache2

**Note**: If the **PhpMyAdmin** package has not been enable to work with apache web server automatically, run the following commands to copy the phpmyadmin apache configuration file located under **/etc/phpmyadmin/** to apache webserver available configurations directory **/etc/apache2/conf-available/** and then activate it using the **a2enconf** utility, and restart apache service effect the recent changes, as follows.

$ sudo cp /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf

$ sudo a2enconf phpmyadmin

$ sudo systemctl restart apache2

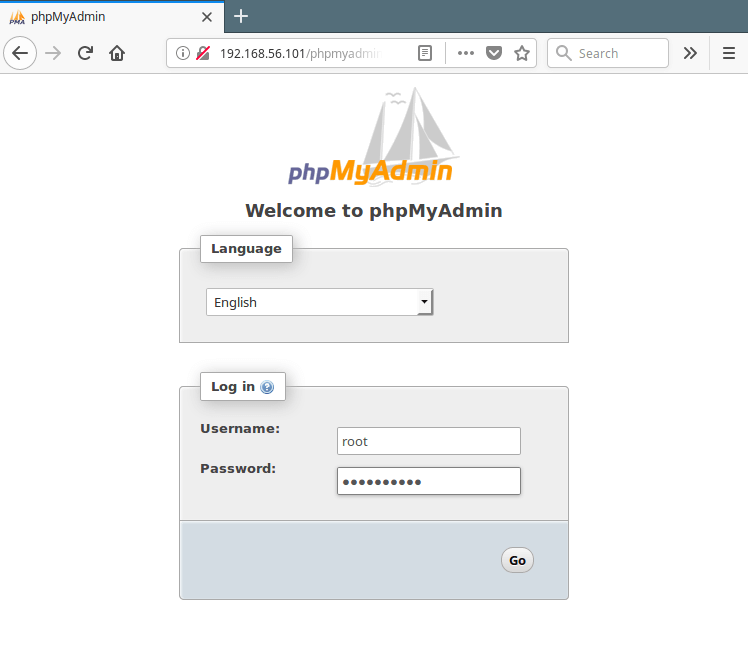
**14.** Lastly, from a web browser, and type the following URL to access you phpMyAdmin web frontend.

http://domain\_name/phpmyadmin

OR

http://SERVER\_IP/phpmyadmin

Use the **root** credentials to authenticate in the phpMyAdmin, as shown in the following screen shot.

[](https://www.tecmint.com/wp-content/uploads/2018/06/PhpMyAdmin-Login.png)

PhpMyAdmin Login

**Important**: Starting from **MySQL 5.7**, root login requires **sudo command**, therefore the root login will fail via phpmyadmin, you may need to create another admin user account. Access the mariadb shell using the root account from a terminal, and run the following commands to create a new user:

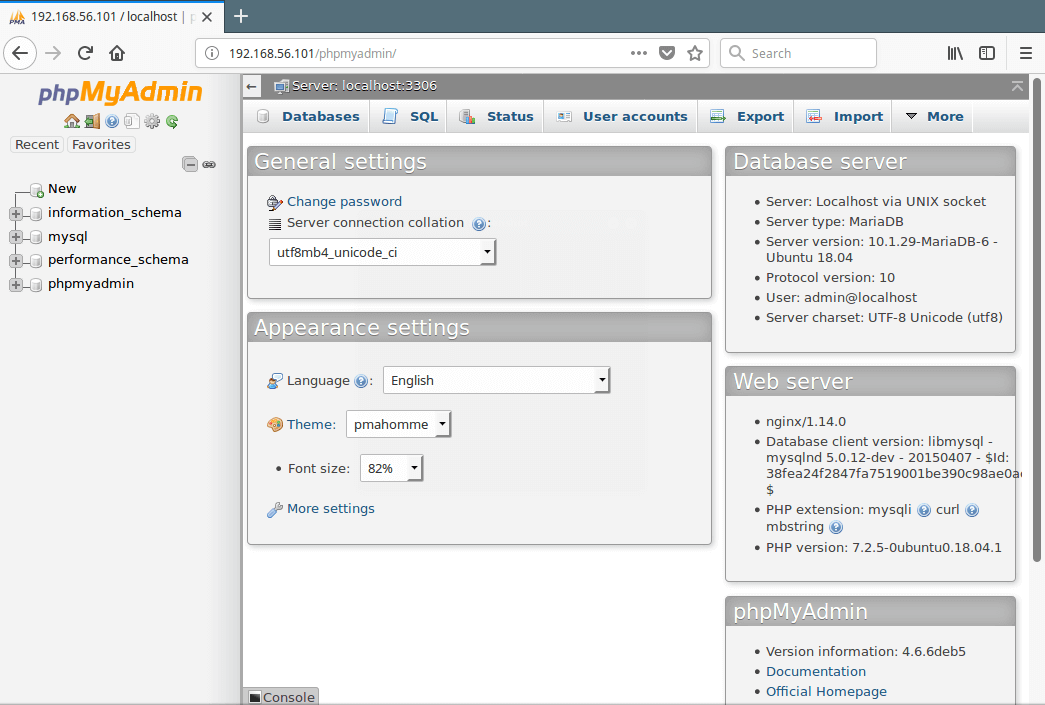
$ sudo mysql -u root -p

MariaDB [(none)]> CREATE USER '**admin**'@'localhost' IDENTIFIED BY '**=@!#254tecmint**';

MariaDB [(none)]> GRANT ALL PRIVILEGES ON \*.\* TO '**admin**'@'localhost' WITH GRANT OPTION;

MariaDB [(none)]> FLUSH PRIVILEGES;

Now log into **PhpMyAdmin** using the new **admin** credentials to administer your databases.

[](https://www.tecmint.com/wp-content/uploads/2018/06/PhpMyAdmin-MySQL-Database-Administration.png)

PhpMyAdmin MySQL Database Administration

Installing Wordpress

|  |
| --- |
| #Install WordPress CMS |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| wget -c http://wordpress.org/latest.tar.gz |
|  |

|  |
| --- |
| tar -xzvf latest.tar.gz |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #Then move the WordPress files from the extracted folder to the Apache default root directory, /var/www/html/: |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| sudo rsync -av wordpress/\* /var/www/html/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #Next, set the correct permissions on the website directory, that is give ownership of the WordPress files to the web server as follows: |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| sudo chown -R www-data:www-data /var/www/html/ |
|  |

|  |
| --- |
| sudo chmod -R 755 /var/www/html/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #Create WordPress Database |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| mysql -u root -p |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #At the mysql shell, type the following commands, pressing Enter after each line of a mysql command. Remember to use your own, valid values for database\_name, databaseuser, and also use a strong and secure password as databaseuser\_password: |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| mysql> CREATE DATABASE wp\_myblog; |
|  |

|  |
| --- |
| mysql> GRANT ALL PRIVILEGES ON wp\_myblog.\* TO 'your\_username\_here'@'localhost' IDENTIFIED BY 'your\_chosen\_password\_here'; |
|  |

|  |
| --- |
| mysql> FLUSH PRIVILEGES; |
|  |

|  |
| --- |
| mysql> EXIT; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #Go the /var/www/html/ directory and rename existing wp-config-sample.php to wp-config.php: |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| sudo mv wp-config-sample.php wp-config.php |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #then update it with your database information under the MySQL settings section (refer to the highlighted boxes in the image below): |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Vi /var/www/html/wp-config.php |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // \*\* MySQL settings - You can get this info from your web host \*\* // |
|  |

|  |
| --- |
| /\*\* The name of the database for WordPress \*/ |
|  |

|  |
| --- |
| define('DB\_NAME', 'database\_name\_here'); /\*\* MySQL database username \*/ define('DB\_USER', 'username\_here'); /\*\* MySQL database password \*/ define('DB\_PASSWORD', 'password\_here'); /\*\* MySQL hostname \*/ define('DB\_HOST', 'localhost'); /\*\* Database Charset to use in creating database tables. \*/ define('DB\_CHARSET', 'utf8'); /\*\* The Database Collate type. Don't change this if in doubt. \*/ define('DB\_COLLATE', ''); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #Afterwards, restart the web server and mysql service using the commands below: |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| http://server-address/wp-admin |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*conclusion\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |