

How to Install phpMyAdmin with Apache (LAMP) on Ubuntu 20.04 LTS

This tutorial will be showing you how to install phpMyAdmin with Apache, MariaDB, PHP7.4 (LAMP stack) on Ubuntu 20.04 LTS. phpMyAdmin is a free and open-source web-based database management tool written in PHP. It provides a graphical web interface for users to manage MySQL or MariaDB database.

phpMyAdmin allows administrators to:

- browse through databases and tables;
- create, copy, rename, alter and drop databases;
- create, copy, rename, alter and drop tables;
- perform table maintenance;
- add, edit and drop fields;
- execute any SQL-statement, even multiple queries;
- create, alter and drop indexes;
- load text files into tables;
- create and read dumps of tables or databases;
- export data to SQL, CSV, XML, Word, Excel, PDF and LaTeX formats;
- administer multiple servers;
- manage MySQL users and privileges;
- check server settings and runtime information with configuration hints;
- check referential integrity in MyISAM tables;
- create complex queries using Query-by-example (QBE), automatically connecting required tables;
- create PDF graphics of database layout;
- search globally in a database or a subset of it;
- transform stored data into any format using a set of predefined functions, such as displaying BLOB-data as image or download-link;
- manage InnoDB tables and foreign keys;

Prerequisites

To follow this tutorial, you need a Ubuntu 20.04 OS running on your local computer or on a remote server.

If you are looking for a VPS (Virtual Private Server), then you can create an account at Vultr via [my referral link](#) to get \$50 free credit (for new users only). And if you need to set up phpMyAdmin with a domain name, I recommend buying domain names from [NameCheap](#) because the price is low and they give whois privacy protection for free.

It is assumed that you have already installed LAMP stack on Ubuntu 20.04. If not, please check out the following tutorial.

- [How to Install LAMP stack \(Apache, MariaDB, PHP7.4\) on Ubuntu 20.04 LTS](#)

With that out of the way, let's get started with installing phpMyAdmin.

Step 1: Download and Install phpMyAdmin on Ubuntu 20.04

phpMyAdmin is included in Ubuntu 20.04 software repository. so we can easily install it with the command below.

```
sudo apt update
sudo apt install phpmyadmin
```

The above command will install all necessary dependencies including PHP7 extensions. During the installation, it will prompt you to select a web server to configure. Hit the `|space|` bar to select `apache2`. (An asterisk indicates the item is selected). Then hit `|Tab|` key and `|Enter|` to confirm your choice.

Configuring phpmyadmin

Please choose the web server that should be automatically configured to run phpMyAdmin.

Web server to reconfigure automatically:

apache2
 lighttpd

<Ok>

LinuxBabe.Com

In the next screen, select `Yes` to configure a database for phpMyAdmin with `dbconfig-common`.

Configuring phpmyadmin

The phpmyadmin package must have a database installed and configured before it can be used. This can be optionally handled with `dbconfig-common`.

If you are an advanced database administrator and know that you want to perform this configuration manually, or if your database has already been installed and configured, you should refuse this option. Details on what needs to be done should most likely be provided in `/usr/share/doc/phpmyadmin`.

Otherwise, you should probably choose this option.

Configure database for phpmyadmin with `dbconfig-common`?

<Yes>

<No>

Then set a password for the `phpmyadmin` user in MariaDB/MySQL.


```
/etc/apache2/conf-enabled/phpmyadmin.conf: symbolic link to ../conf-available/phpmyadmin.conf
```

If this file doesn't exist on your server, it's likely that you didn't select Apache web server in the phpMyAdmin setup wizard. You can fix it with the following commands.

```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf
```

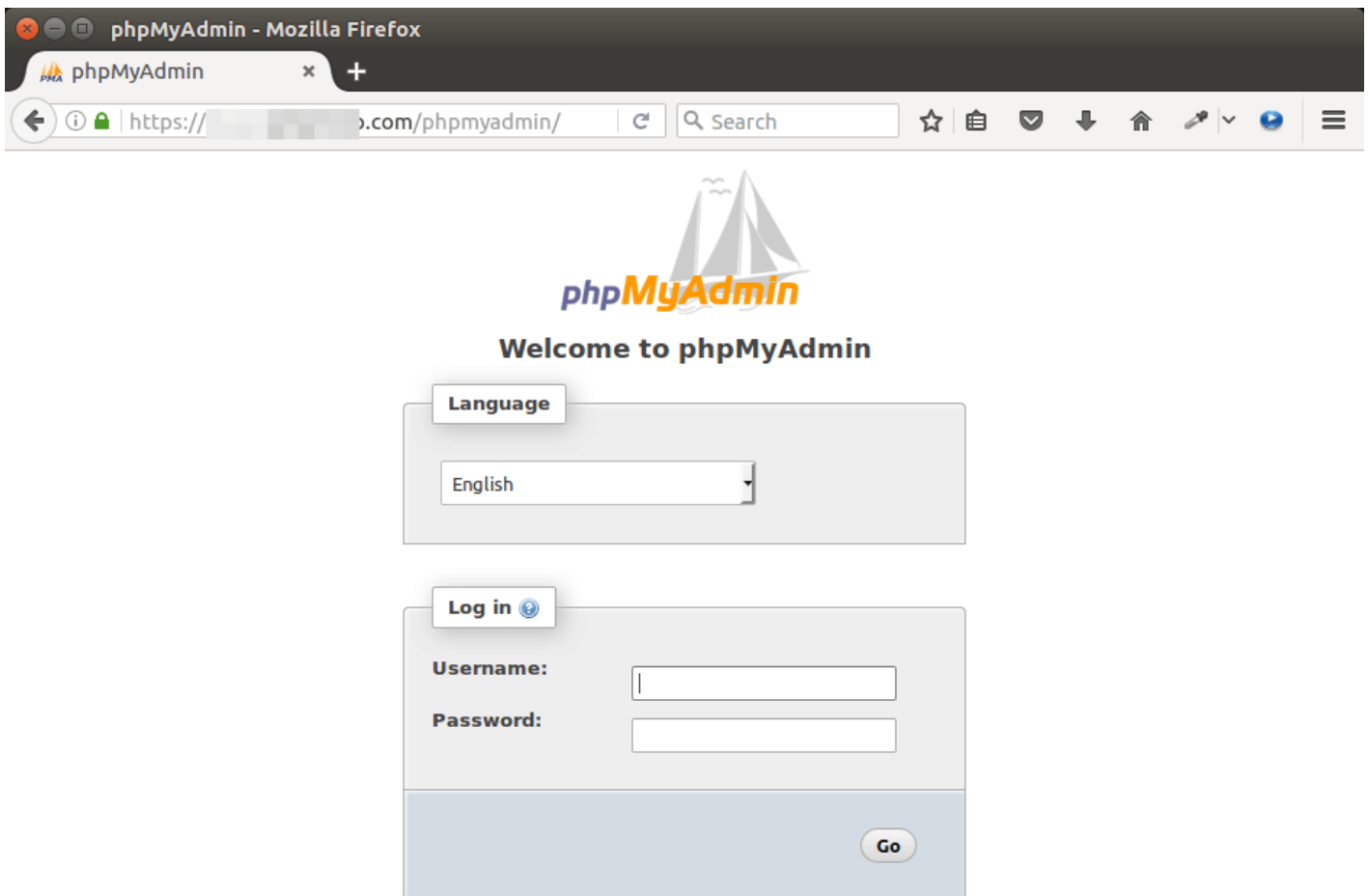
```
sudo a2enconf phpmyadmin
```

```
sudo systemctl reload apache2
```

Step 2: Accessing phpMyAdmin from a Sub-directory

When installing phpMyAdmin, a configuration file is placed at `/etc/apache2/conf-enabled/phpmyadmin.conf`, which allows us to access phpMyAdmin via sub-directory. If you have installed other web applications like [WordPress](#) on the same Ubuntu 20.04 server, then simply add `/phpmyadmin` after your domain name in the browser address bar.

```
example.com/phpmyadmin
```



If phpMyAdmin is installed on your local Ubuntu computer, then you can access phpMyAdmin web interface by typing in the following text in the browser address bar.

```
localhost/phpmyadmin
```

If the connection is refused or failed to complete, there might be a firewall preventing HTTP requests. If you are using iptables firewall, then you need to run the following commands to open TCP port 80 and 443.

```
sudo iptables -I INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -I INPUT -p tcp --dport 443 -j ACCEPT
```

If you are using [UFW firewall](#), then run this command to open TCP port 80 and 443.

```
sudo ufw allow 80,443/tcp
```

Now You can use MariaDB/MySQL username and password to log into phpMyAdmin. If you can't login as root, go to step 5 to troubleshoot this problem.

Step 3: Access phpMyAdmin From a Sub-domain

If your Ubuntu 20.04 server doesn't have other Apache virtual host, then you may want to use a sub-domain to access phpMyAdmin web interface. This way, you can enable HTTPS to encrypt the traffic.

First, we need to create an Apache virtual host for phpMyAdmin. The existing phpMyAdmin configuration snippet can be used as a template. Let's copy it to a new file.

```
sudo cp /etc/apache2/conf-enabled/phpmyadmin.conf /etc/apache2/sites-available/phpmyadmin.conf
```

Then edit the new file with a command line text editor, such as Nano.

```
sudo nano /etc/apache2/sites-available/phpmyadmin.conf
```

This file lacks `<VirtualHost>` tags, so we need to add the following lines at the beginning of this file. Replace `pma.example.com` with your preferred sub-domain for phpMyAdmin. Don't forget to [create DNS A record](#) for this sub-domain.

```
<VirtualHost *:80>
    ServerName pma.example.com
    DocumentRoot /usr/share/phpmyadmin

    ErrorLog ${APACHE_LOG_DIR}/pma.error.log
    CustomLog ${APACHE_LOG_DIR}/pma.access.log combined
```

Add the following line at the end of this file to close the `<VirtualHost>` tag.

```
</VirtualHost>
```

```

<VirtualHost *:80>
  ServerName pma.linuxbabe.com
  DocumentRoot /usr/share/phpmyadmin

  ErrorLog ${APACHE_LOG_DIR}/pma.error.log
  CustomLog ${APACHE_LOG_DIR}/pma.access.log combined

# phpMyAdmin default Apache configuration

Alias /phpmyadmin /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
  Options SymLinksIfOwnerMatch
  DirectoryIndex index.php

# limit libapache2-mod-php to files and directories necessary by pma
<IfModule mod_php7.c>
  php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
  php_admin_value open_basedir /usr/share/phpmyadmin:/etc/phpmyadmin:/var/lib/phpmyadmin:/usr/share/doc/phpmyadmin:/usr/share/php/phpseclib:/usr/share/php/PhpMyAdmin:/usr/share/php/pear:/usr/share/php/Psr/Cache:/usr/share/php/Psr/Log:/usr/share/php/Psr/SimpleCache/
</IfModule>

</Directory>

# Disallow web access to directories that don't need it
<Directory /usr/share/phpmyadmin/templates>
  Require all denied
</Directory>
<Directory /usr/share/phpmyadmin/libraries>
  Require all denied
</Directory>

</VirtualHost>

```

Save and close the file. (To save a file in Nano text editor, press `Ctrl+O`, then press Enter to confirm. To exit, press `Ctrl+X`.) Then enable this virtual host.

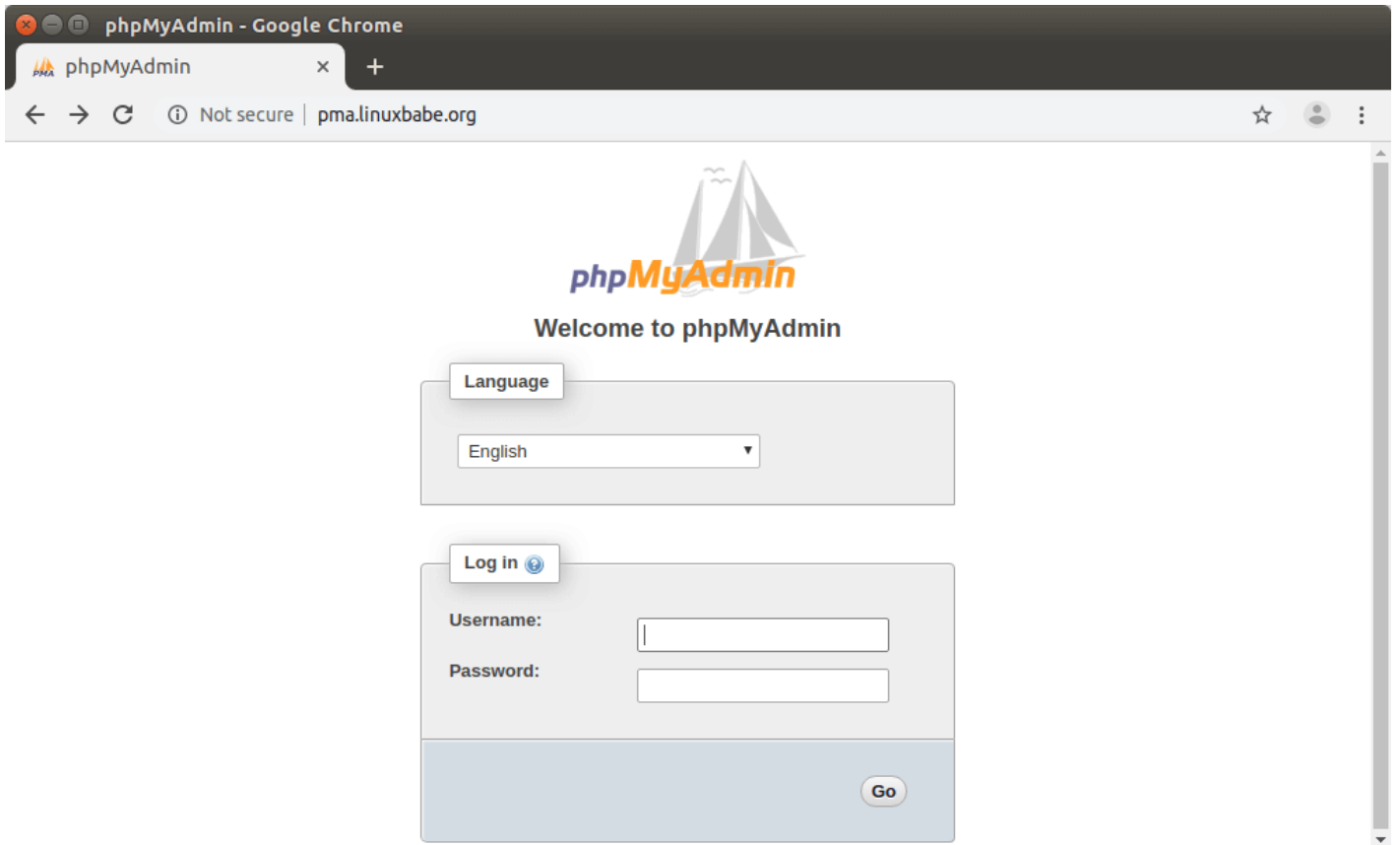
```
sudo a2ensite phpmyadmin.conf
```

Reload Apache web server for this change to take effect.

```
sudo systemctl reload apache2
```

Now you should be able to access phpMyAdmin web interface via

```
pma.example.com
```



Before entering user credentials in the login form, let's enable HTTPS.

Step 4: Enable HTTPS on phpMyAdmin with Apache

To secure the phpMyadmin web interface, we can install a free Let's Encrypt TLS certificate. Run the following command to install the Let's Encrypt client (certbot) from Ubuntu 20.04 software repository.

```
sudo apt install certbot python3-certbot-apache
```

`python3-certbot-apache` is the Apache plugin for Certbot. Now run the following command to obtain and install TLS certificate.

```
sudo certbot --apache --agree-tos --redirect --hsts --staple-ocsp --must-staple -d  
pma.example.com --email you@example.com
```

Where:

- **-apache:** Use the Apache authenticator and installer

- **-agree-tos:** Agree to Let's Encrypt terms of service
- **-redirect:** Enforce HTTPS by adding 301 permanent redirect.
- **-hsts:** Add the Strict-Transport-Security header to every HTTP response.
- **-staple-ocsp:** Enables OCSP Stapling.
- **-must-staple:** Adds the OCSP Must Staple extension to the certificate.
- **-d** flag is followed by a list of domain names, separated by comma. You can add up to 100 domain names.
- **-email:** Email used for registration and recovery contact.

You will be asked if you want to receive emails from EFF(Electronic Frontier Foundation). After choosing Y or N, your TLS certificate will be automatically obtained and configured for you, which is indicated by the message below.

```
IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/pma.██████████/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/pma.██████████/privkey.pem
  Your cert will expire on 2020-07-29. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot again
  with the "certonly" option. To non-interactively renew *all* of
  your certificates, run "certbot renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
  Donating to EFF:                  https://eff.org/donate-le
```

Step 5: Troubleshoot phpMyAdmin Login Error

If you try to login to phpMyAdmin with MariaDB root account, you may see the following error.

```
#1698 - Access denied for user 'root'@'localhost'
```

or

```
mysqli_real_connect(): (HY000/1698): Access denied for user 'root'@'localhost'
```

If you log in with user `|phpmyadmin|`, you won't see the above error. However, user `|phpmyadmin|` can only be used to manage the `|phpmyadmin|` database. The cause of the error is that by default MariDB root user is authenticated via the `unix_socket` plugin, instead of using the `|mysql_native_password|` plugin. To solve this problem, we can create another admin user and grant all privileges to the

new admin user.

Log into MariaDB server from the command line.

```
sudo mysql -u root
```

Create an admin user with password authentication.

```
create user admin@localhost identified by 'your-preferred-password';
```

Grant all privileges on all databases.

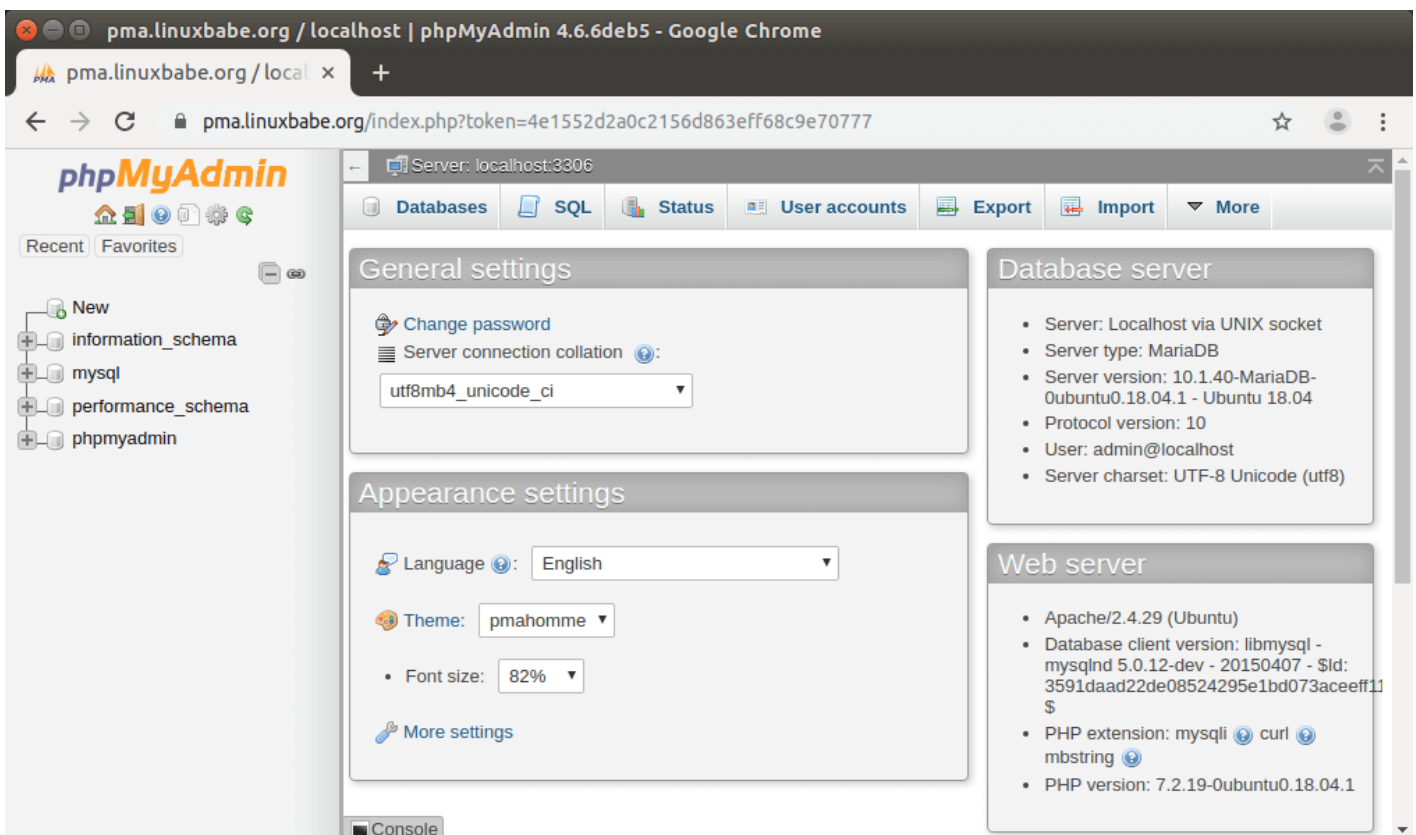
```
grant all privileges on *.* to admin@localhost with grant option;
```

Flush privileges and exit;

```
flush privileges;
```

```
exit;
```

Now you can log into phpMyAdmin with the `admin` account and manage all databases.



The screenshot displays the phpMyAdmin web interface in a Google Chrome browser. The address bar shows the URL `pma.linuxbabe.org / localhost | phpMyAdmin 4.6.6deb5 - Google Chrome`. The page title is `pma.linuxbabe.org / localhost`. The main content area is titled `Server: localhost:3306` and features a navigation menu with options: `Databases`, `SQL`, `Status`, `User accounts`, `Export`, `Import`, and `More`. The interface is divided into several panels:

- General settings:** Includes a `Change password` link and a `Server connection collation` dropdown menu set to `utf8mb4_unicode_ci`.
- Appearance settings:** Includes a `Language` dropdown set to `English`, a `Theme` dropdown set to `pmahomme`, and a `Font size` dropdown set to `82%`. A `More settings` link is also present.
- Database server:** Lists server details:
 - Server: Localhost via UNIX socket
 - Server type: MariaDB
 - Server version: 10.1.40-MariaDB-0ubuntu0.18.04.1 - Ubuntu 18.04
 - Protocol version: 10
 - User: admin@localhost
 - Server charset: UTF-8 Unicode (utf8)
- Web server:** Lists web server details:
 - Apache/2.4.29 (Ubuntu)
 - Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 3591daad22de08524295e1bd073aceff11\$
 - PHP extension: mysqli, curl, mbstring
 - PHP version: 7.2.19-0ubuntu0.18.04.1

A `Console` tab is visible at the bottom left of the interface.

Created 11 July 2021 19:00:12 by Admin
Updated 11 July 2021 19:00:41 by Admin