Install phpMyAdmin with Apache (LAMP) on Debian 10 Buster

This tutorial will be showing you how to install phpMyAdmin with Apache, MariaDB, PHP7.3 (LAMP stack) on Debian 10 Buster. 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. We are also going to learn how to enable two-factor authentication on phpMyAdmin.

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 Debian 10 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 Debian 10. If not, please check out the following tutorial.

- How to Install LAMP stack (Apache, MariaDB, PHP7.3) on Debian 10 Buster

Please note that you need to have root privilege when installing software on Debian. You can add **sudo** at the beginning of a command, or use `su -` command to switch to root user.

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

Step 1: Download phpMyAdmin on Debian 10 Server

phpMyAdmin isn't included in Debian 10 software repository, so we have to manually download the software. Go to phpMyAdmin download page to check the latest stable version. Then run the following command to download it.

```
wget https://files.phpmyadmin.net/phpMyAdmin/4.9.0.1/phpMyAdmin-4.9.0.1-all-languages.zip
```

Hint: You can always use the above URL format to download the latest stable version of phpMyAdmin. Simply replace 4.9.0.1 with the latest version number.

Then extract it.

```
sudo apt install unzip


unzip phpMyAdmin-4.9.0.1-all-languages.zip
```

Move phpMyadmin 4.9 to `/usr/share/` directory.

```
sudo mv phpMyAdmin-4.9.0.1-all-languages /usr/share/phpmyadmin
```

Then make the web server user (`www-data`) as the owner of this directory.

```
sudo chown -R www-data:www-data /usr/share/phpmyadmin
```

Step 2: Create a MariaDB Database and User for phpMyAdmin

Now we need to log in to MariaDB console and create a database and user for phpMyAdmin. By default, the MaraiDB package on Debian uses *unix_socket* to authenticate user login, which basically means you can use username and password of the OS to log into MariaDB console. So you can run the following command to login without providing MariaDB root password.

```
sudo mysql -u root
```

Next, create a new database for phpMyAdmin using the following SQL command. This tutorial names it `phpmyadmin`, you can use whatever name you like for the database.

```
CREATE DATABASE phpmyadmin DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

The following SQL command will create the `phpmyadmin` database user and set a password, and at the same time grant all permission of the new database to the new user so later on phpMyAdmin can write to the database. Replace red texts with your perferred password.

```
GRANT ALL ON phpmyadmin.* TO 'phpmyadmin'@'localhost' IDENTIFIED BY 'your_preferred_password';
```

Flush privileges table and exit MariaDB console.

```

```

```
FLUSH PRIVILEGES;

EXIT;
```

Step 3: Install Required and Recommended PHP Modules.

Run the following command to install PHP modules required or recommended by phpMyAdmin.

```
sudo apt install php-imagick php-phpseclib php-php-gettext php7.3-common php7.3-mysql php7.3-gd php7.3-imap php7.3-json php7.3-curl php7.3-zip php7.3-xml php7.3-mbstring php7.3-bz2 php7.3-intl php7.3-gmp
```

Then restart Apache.

```
sudo systemctl restart apache2
```

Step 4: Create Apache Configuration for phpMyAdmin

If you would like to access phpMyAdmin web interface from a sub-directory, then create a configuration snippet with the following command.

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

Paste the following text into the file.

```
# phpMyAdmin default Apache configuration

Alias /phpmyadmin /usr/share/phpmyadmin

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

```
    <IfModule mod_php5.c>
        <IfModule mod_mime.c>
            AddType application/x-httpd-php .php
        </IfModule>
        <FilesMatch ".+\.php$">
            SetHandler application/x-httpd-php
        </FilesMatch>

        php_value include_path .
        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/php/php-
gettext/:/usr/share/php/php-php-
gettext/:/usr/share/javascript/:/usr/share/php/tcpdf/:/usr/share/doc/phpmyadmin/:/usr/share/php/
        php_admin_value mbstring.func_overload 0
    </IfModule>
    <IfModule mod_php.c>
        <IfModule mod_mime.c>
            AddType application/x-httpd-php .php
        </IfModule>
        <FilesMatch ".+\.php$">
            SetHandler application/x-httpd-php
        </FilesMatch>

        php_value include_path .
        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/php/php-
gettext/:/usr/share/php/php-php-
gettext/:/usr/share/javascript/:/usr/share/php/tcpdf/:/usr/share/doc/phpmyadmin/:/usr/share/php/
        php_admin_value mbstring.func_overload 0
    </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>
<Directory /usr/share/phpmyadmin/setup/lib>
    Require all denied
</Directory>
```

Save and close the file. Then enable this configuration snippet.

```
sudo a2enconf phpmyadmin.conf
```

We also need to create the phpMyAdmin temp folder.

```
sudo mkdir -p /var/lib/phpmyadmin/tmp
sudo chown www-data:www-data /var/lib/phpmyadmin/tmp
```

Reload Apache for the changes to take effect.

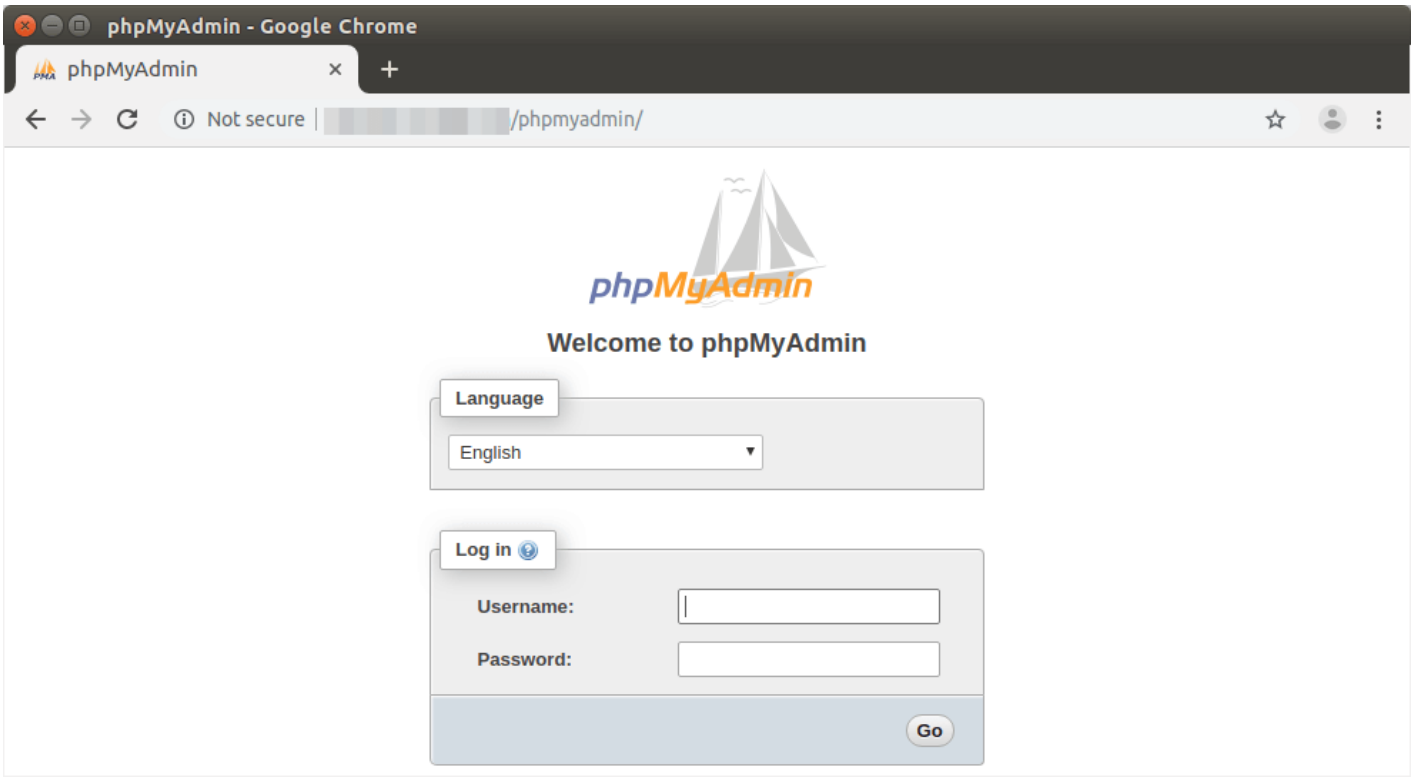
```
sudo systemctl reload apache2
```

Now you can access phpMyAdmin web interface at

```
your-server-ip/phpmyadmin
```

If phpMyAdmin is installed on your local Debian 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 command 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 http
sudo ufw allow https
```

Step 5: Access phpMyAdmin From a Sub-domain

Sometimes, 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
```

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.

```
</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.

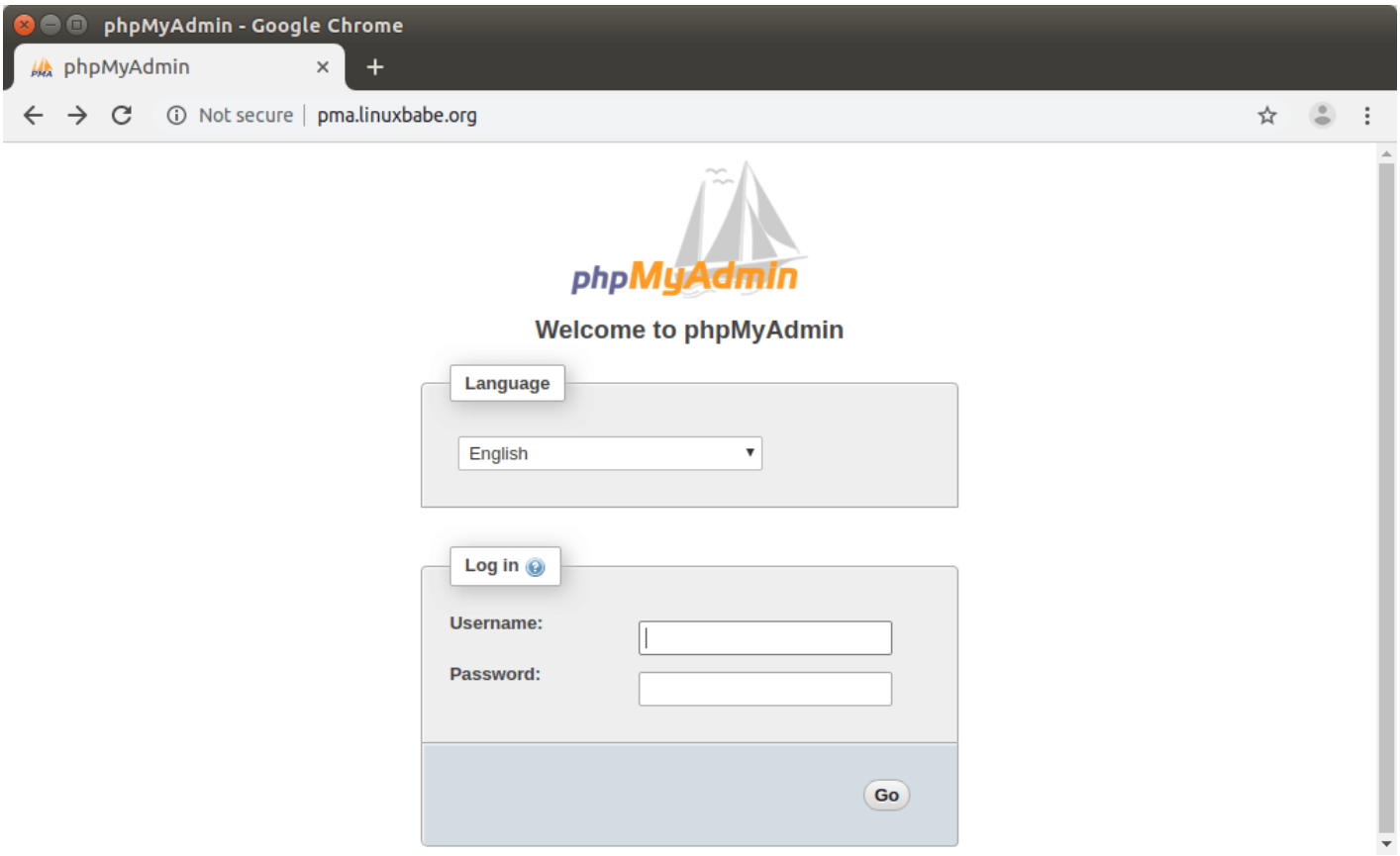
```
sudo a2ensite phpmyadmin.conf
```

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 6: 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 Debian 10 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
```

Explanation:

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

- **–agree-tos**: Agree to Let’s Encrypt terms of service
- **–redirect**: Add 301 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.linuxbabe.org/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/pma.linuxbabe.org/privkey.pem
   Your cert will expire on 2019-10-10. 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"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - 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 7: Run the phpMyAdmin Setup Script

Enter the following in your browser address bar.

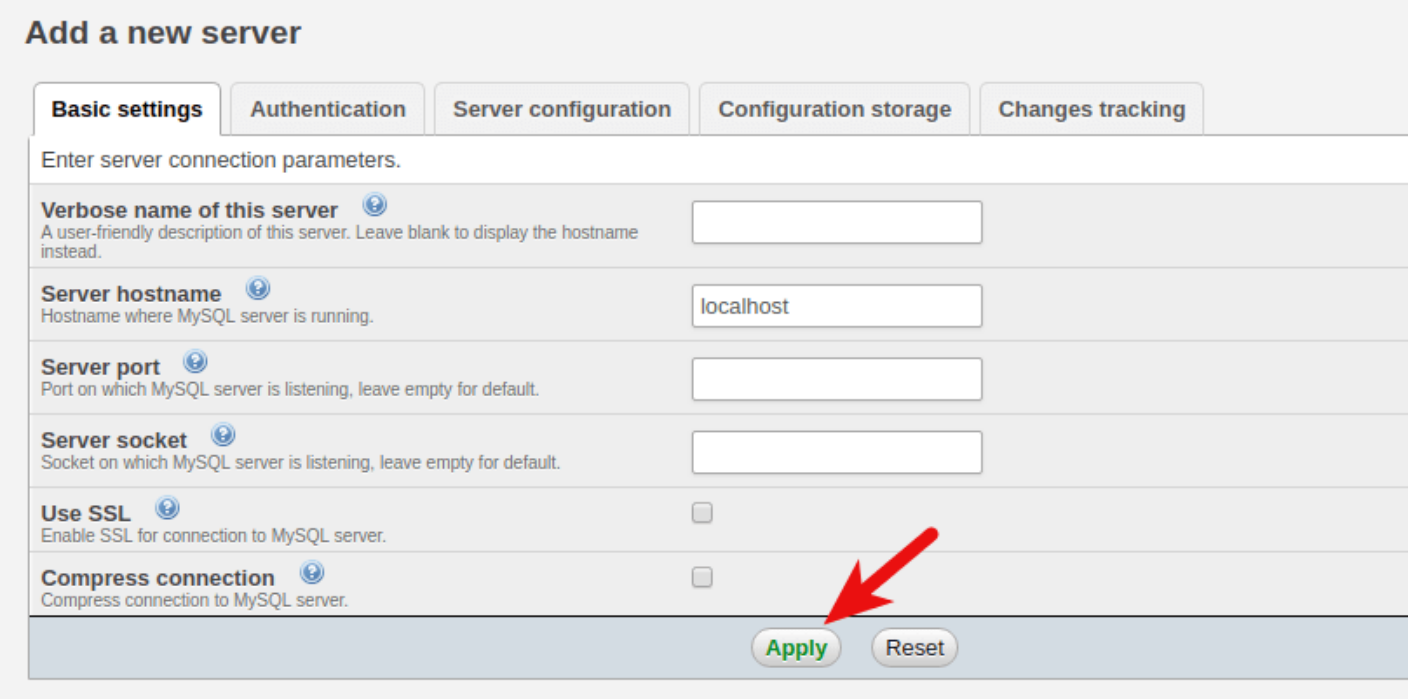
```
your-server-ip/phpmyadmin/setup
```

or

```
pma.example.com/setup
```

Click the `New Server` button to configure a server.

Then simply click the `Apply` button.

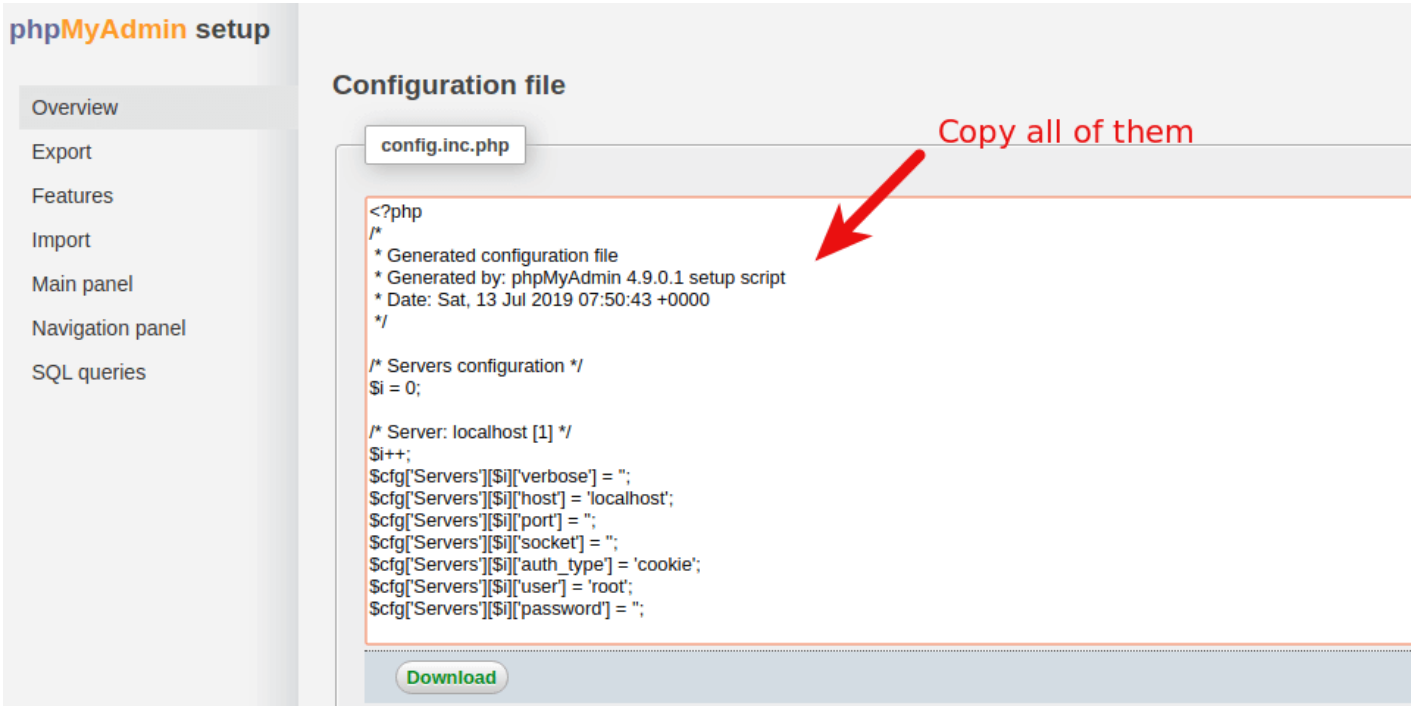


Next, click the Display button to show the configuration file.

In the `/usr/share/phpmyadmin/` directory, create the config.inc.php file.

```
sudo nano /usr/share/phpmyadmin/config.inc.php
```

Copy the content of `config.inc.php` from the phpMyAdmin setup page and paste them into `/usr/share/phpmyadmin/config.inc.php` file.



Step 8: Troubleshoot phpMyAdmin Login Error

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

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

and

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

If you login 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.

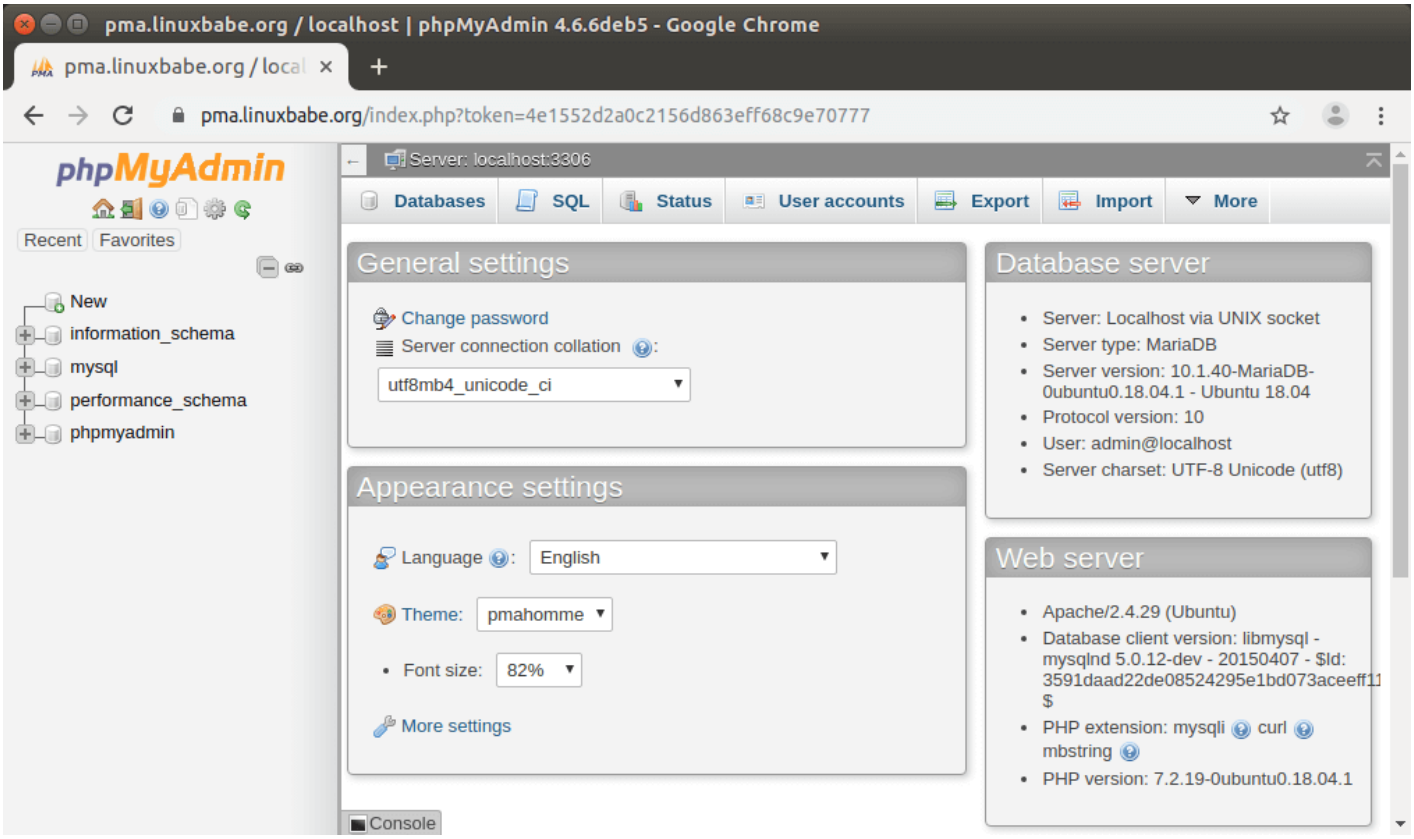
```
grant all privileges on *.* to admin@localhost with grant option;
```

Flush privileges and exit;

```
flush privileges;


exit;
```

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



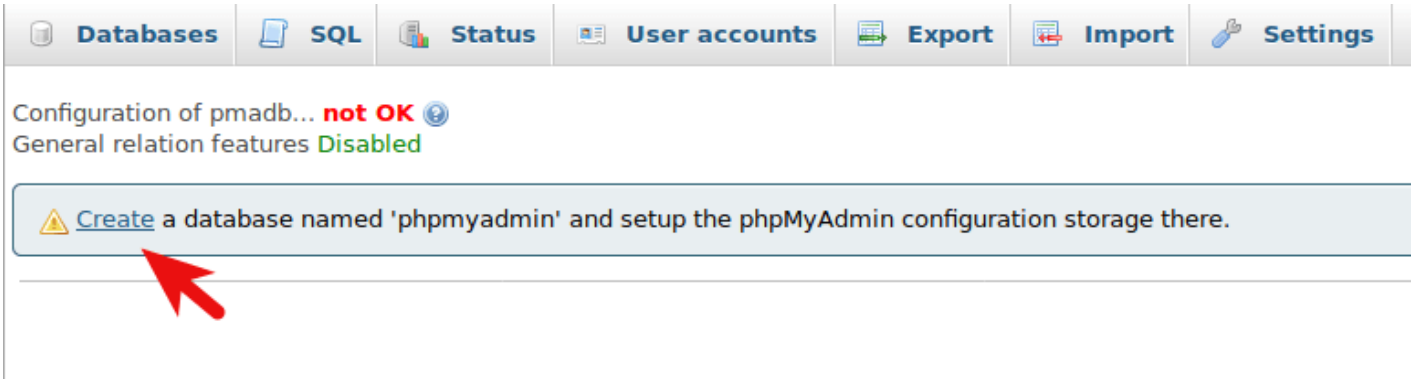
Step 9: Set Up phpMyAdmin

Configuration Storage

Now in the phpMyAdmin control panel, you can see a warning message.

```
The phpMyAdmin configuration storage is not completely configured, some extended features
have been deactivated. Find out why.
Or alternately go to 'Operations' tab of any database to set it up there.
```

Click the Find out why link. Then click the Create link to create tables in the `phpmyadmin` database.



Step 10: Restricting Access to the /setup Directory

To restrict access to the /setup directory, we can enable basic password authentication with Apache web server. Run the following command to set a password for user `admin`. `/etc/apache2/htpasswd` file is used to store usernames and password.

```
sudo htpasswd -c /etc/apache2/htpasswd admin
```

Then edit the Apache configuration file for phpMyAdmin

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

or

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

Add the following lines.

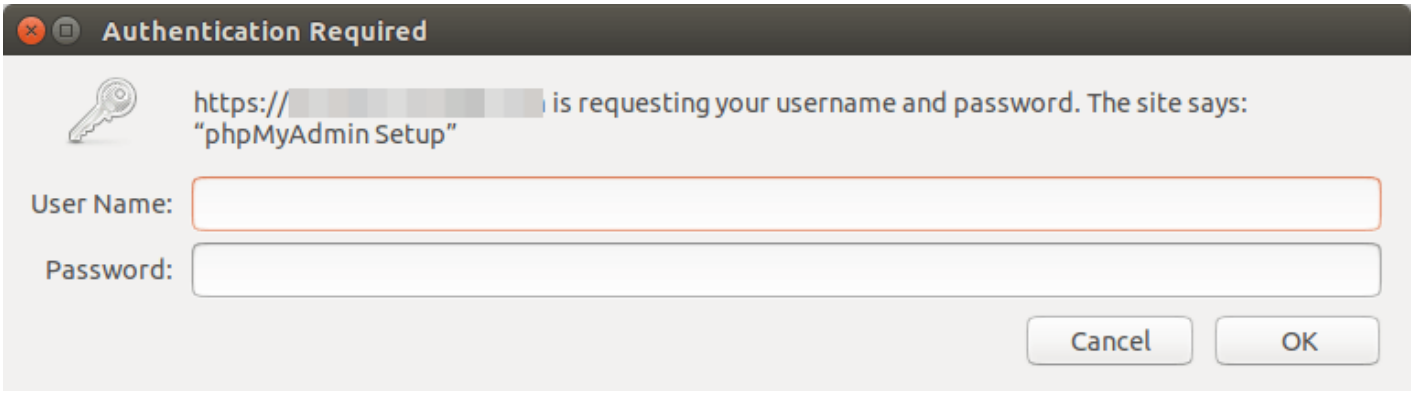
```
```

```
# Authorize for setup
<Directory /usr/share/phpmyadmin/setup>
    <IfModule mod_authz_core.c>
        <IfModule mod_authn_file.c>
            AuthType Basic
            AuthName "phpMyAdmin Setup"
            AuthUserFile /etc/apache2/htpasswd
        </IfModule>
        Require valid-user
    </IfModule>
</Directory>
```

Save and close the file. Then reload Apache for the changes to take effect.

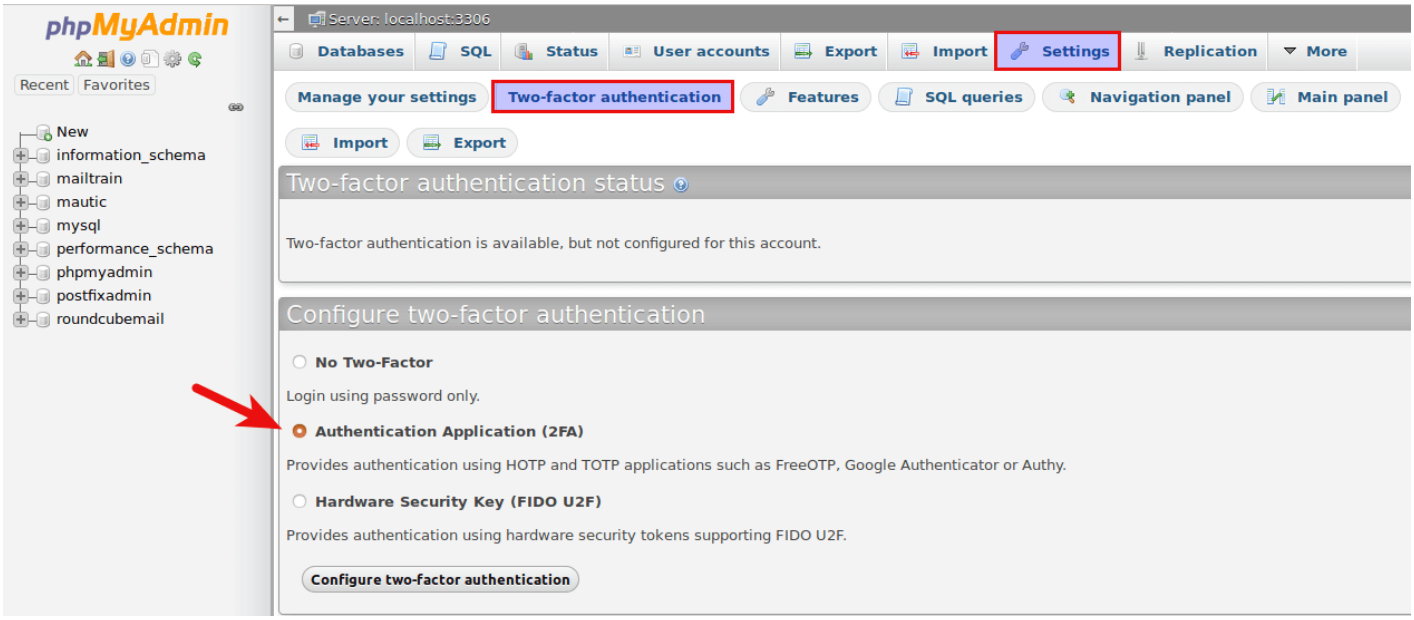
```
sudo systemctl reload apache2
```

If you access the phpMyAdmin setup script again, you will be asked to enter username and password.

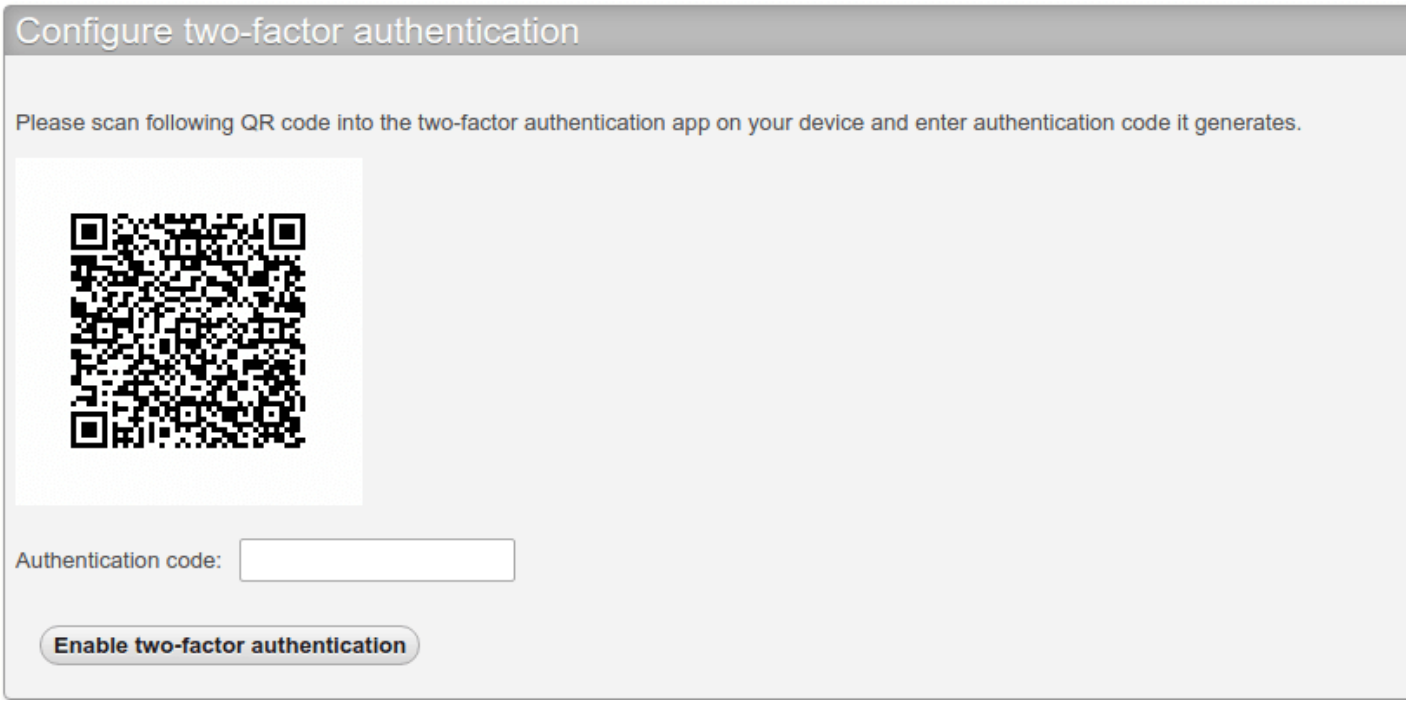


Enable Two-Factor Authentication

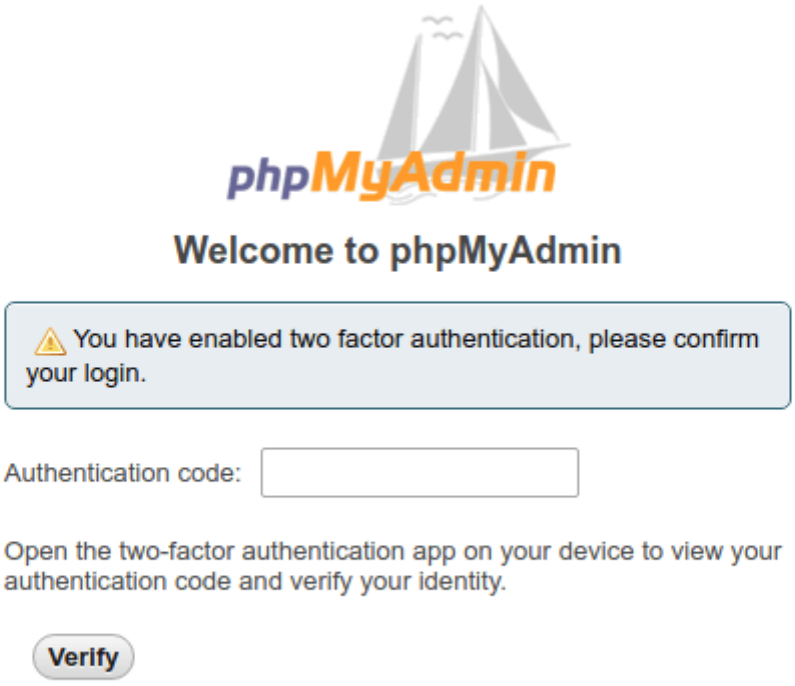
You can also harden phpMyAdmin by enabling two-factor authentication, which is a feature added in version 4.8. To enable it, log into phpMyAdmin. Then go to `Settings` -> `Two-factor authentication` and select **Authentication application (2FA)**.



After clicking the **Configure two-factor authentication** button, you will be presented with a QR code, which you need to scan with a two-factor authentication app on your phone.



Google Authenticator is a popular 2FA app, but I recommend FreeOTP, which is an open-source 2FA app developed by Red Hat. Once you enter the authentication code generated by your 2FA app, two-factor authentication is enabled. If you now log out and log back in, you need to enter the authentication code in addition to username and password.



Revision #1
Created 11 July 2021 19:00:48 by Admin
Updated 11 July 2021 19:01:13 by Admin