Subscribe Share Contents

How To Install and Secure phpMyAdmin on Ubuntu 18.04

Posted May 15, 2018 215.4k APPLICATIONS PHP APACHE UBUNTU 18.04 UBUNTU

52

By: Mark Drake

Not using **Ubuntu 18.04**? Choose a different version:

An earlier version of this tutorial was written by Brennan Bearnes.

Introduction

While many users need the functionality of a database management system like MySQL, they may not feel comfortable interacting with the system solely from the MySQL prompt.

phpMyAdmin was created so that users can interact with MySQL through a web interface. In this guide, we'll discuss how to install and secure phpMyAdmin so that you can safely use it to manage your databases on an Ubuntu 18.04 system.

Prerequisites

Before you get started with this guide, you need to have some basic steps completed.

First, we'll assume that your server has a non-**root** user with `sudo` privileges, as well as a firewall configured with `ufw`, as described in the initial server setup guide for Ubuntu 18.04.

We're also going to assume that you've completed a LAMP (Linux, Apache, MySQL, and PHP) installation on your Ubuntu 18.04 server. If this is not completed yet, you can follow this guide on installing a LAMP stack on Ubuntu 18.04.

Finally, there are important security considerations when using software like phpMyAdmin, since it:

- Communicates directly with your MySQL installation
- Handles authentication using MySQL credentials
- Executes and returns results for arbitrary SQL queries

SCROLL TO TOP

For these reasons, and because it is a widely-deployed PHP application which is frequently targeted for attack, you should never run phpMyAdmin on remote systems over a plain HTTP connection. If you do not have an existing domain configured with an SSL/TLS certificate, you can follow this guide on securing Apache with Let's Encrypt on Ubuntu 18.04. This will require you to register a domain name, create DNS records for your server, and set up an Apache Virtual Host.

Once you are finished with these steps, you're ready to get started with this guide.

Step 1 — Installing phpMyAdmin

To get started, we will install phpMyAdmin from the default Ubuntu repositories.

This is done by updating your server's package index and then using the `apt` packaging system to pull down the files and install them on your system:

```
$ sudo apt update
$ sudo apt install phpmyadmin php-mbstring php-gettext
```

This will ask you a few questions in order to configure your installation correctly.

> **Warning:** When the prompt appears, "apache2" is highlighted, but **not** selected. If you do not hit `SPACE` to select Apache, the installer will *not* move the necessary files during installation. Hit `SPACE`, `TAB`, and then `ENTER` to select Apache.

- For the server selection, choose `apache2`
- Select `Yes` when asked whether to use `dbconfig-common` to set up the database
- You will then be asked to choose and confirm a MySQL application password for phpMyAdmin

The installation process adds the phpMyAdmin Apache configuration file into the `/etc/apache2/conf-enabled/` directory, where it is read automatically. The only thing you need to do is explicitly enable the `mbstring` PHP extension, which you can do by typing:

```
$ sudo phpenmod mbstring
```

Afterwards, restart Apache for your changes to be recognized:

```
$ sudo systemctl restart apache2
```

phpMyAdmin is now installed and configured. However, before you can log in and begin interacting with your MySQL databases, you will need to ensure that your MySQL users have the privileges required for interacting with the program.

SCROLL TO TOP

Step 2 — Adjusting User Authentication and Privileges

When you installed phpMyAdmin onto your server, it automatically created a database user called `phpmyadmin` which performs certain underlying processes for the program. Rather than logging in as this user with the administrative password you set during installation, it's recommended that you log in as either your **root** MySQL user or as a user dedicated to managing databases through the phpMyAdmin interface.

Configuring Password Access for the MySQL Root Account

In Ubuntu systems running MySQL 5.7 (and later versions), the **root** MySQL user is set to authenticate using the `auth_socket` plugin by default rather than with a password. This allows for some greater security and usability in many cases, but it can also complicate things when you need to allow an external program — like phpMyAdmin — to access the user.

In order to log in to phpMyAdmin as your **root** MySQL user, you will need to switch its authentication method from `auth_socket` to `mysql_native_password` if you haven't already done so. To do this, open up the MySQL prompt from your terminal:

```
$ sudo mysql
```

Next, check which authentication method each of your MySQL user accounts use with the following command:

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

```
Output
+------------------+-------------------------------------------+-----------------------+-----------+
| user             | authentication_string                     | plugin                | host      |
+------------------+-------------------------------------------+-----------------------+-----------+
| root             |                                           | auth_socket           | localhost |
| mysql.session    | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| mysql.sys        | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| debian-sys-maint | *8486437DE5F65ADC4A4B001CA591363B64746D4C | mysql_native_password | localhost |
| phpmyadmin       | *5FD2B7524254B7F81B32873B1EA6D681503A5CA9 | mysql_native_password | localhost |
+------------------+-------------------------------------------+-----------------------+-----------+
5 rows in set (0.00 sec)
```

In this example, you can see that the **root** user does in fact authenticate using the `auth_socket` plugin. To configure the **root** account to authenticate with a password, run the following `ALTER USER` command. Be sure to change `password` to a strong password of your choosing:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
```

SCROLL TO TOP

Then, run `FLUSH PRIVILEGES` which tells the server to reload the grant tables and put your new changes into effect:

```
mysql> FLUSH PRIVILEGES;
```

Check the authentication methods employed by each of your users again to confirm that **root** no longer authenticates using the `auth_socket` plugin:

```
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
```

```
Output
+------------------+-------------------------------------------+-----------------------+-----------+
| user             | authentication_string                     | plugin                | host      |
+------------------+-------------------------------------------+-----------------------+-----------+
| root             | *DE06E242B88EFB1FE4B5083587C260BACB2A6158 | mysql_native_password | localhost |
| mysql.session    | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| mysql.sys        | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| debian-sys-maint | *8486437DE5F65ADC4A4B001CA591363B64746D4C | mysql_native_password | localhost |
| phpmyadmin       | *5FD2B7524254B7F81B32873B1EA6D681503A5CA9 | mysql_native_password | localhost |
+------------------+-------------------------------------------+-----------------------+-----------+
5 rows in set (0.00 sec)
```

You can see from this output that the **root** user will authenticate using a password. You can now log in to the phpMyAdmin interface as your **root** user with the password you've set for it here.

Configuring Password Access for a Dedicated MySQL User

Alternatively, some may find that it better suits their workflow to connect to phpMyAdmin with a dedicated user. To do this, open up the MySQL shell once again:

```
$ sudo mysql
```

Note: If you have password authentication enabled, as described in the previous section, you will need to use a different command to access the MySQL shell. The following will run your MySQL client with regular user privileges, and you will only gain administrator privileges within the database by authenticating:

```
$ mysql -u root -p
```

From there, create a new user and give it a strong password:

SCROLL TO TOP

```
mysql> CREATE USER 'sammy'@'localhost' IDENTIFIED BY 'password';
```

Then, grant your new user appropriate privileges. For example, you could grant the user privileges to all tables within the database, as well as the power to add, change, and remove user privileges, with this command:

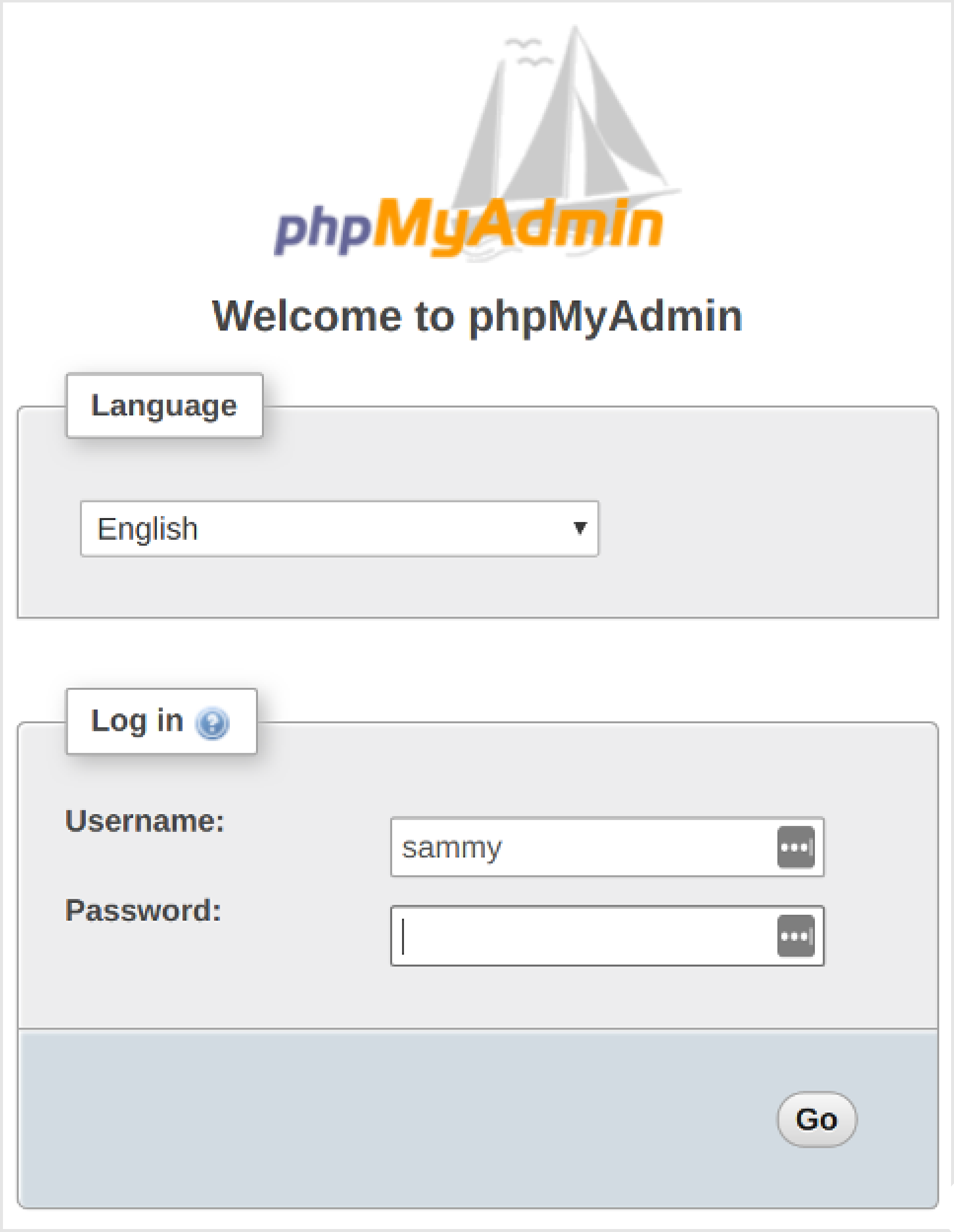
```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'sammy'@'localhost' WITH GRANT OPTION;
```

Following that, exit the MySQL shell:

```
mysql> exit
```

You can now access the web interface by visiting your server's domain name or public IP address followed by `/phpmyadmin`:

```
https://your_domain_or_IP/phpmyadmin
```



Welcome to phpMyAdmin

Language: English

Log in

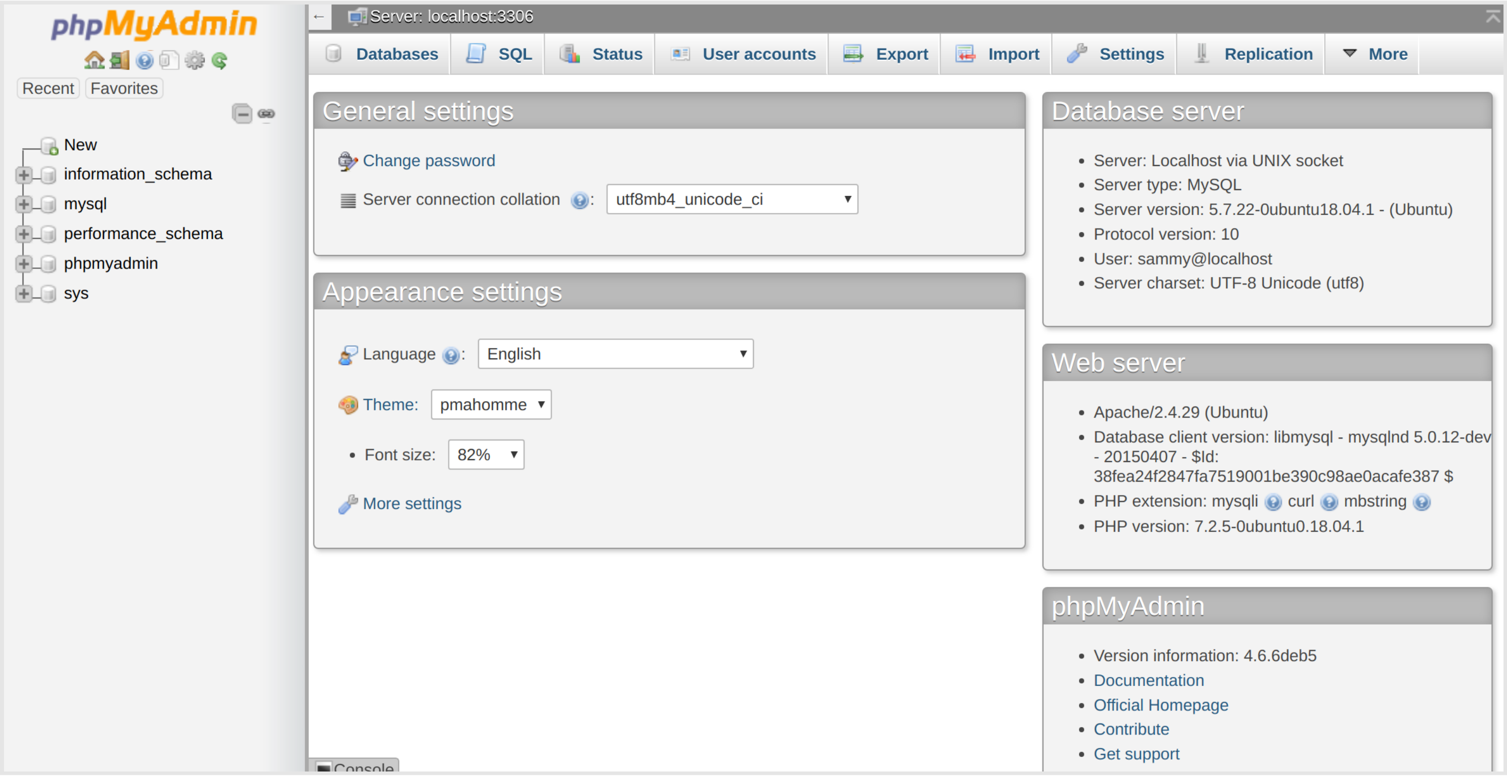
Username: sammy

Password:

SCROLL TO TOP

Log in to the interface, either as **root** or with the new username and password you just configured.

When you log in, you'll see the user interface, which will look something like this:



General settings

Change password

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language: English

Theme: pmahomme

- Font size: 82%

More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MySQL
- Server version: 5.7.22-0ubuntu18.04.1 - (Ubuntu)
- Protocol version: 10
- User: sammy@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.29 (Ubuntu)
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - $Id: 38fea24f2847fa7519001be390c98ae0acafe387 $
- PHP extension: mysqli curl mbstring
- PHP version: 7.2.5-0ubuntu0.18.04.1

phpMyAdmin

- Version information: 4.6.6deb5
- Documentation
- Official Homepage
- Contribute
- Get support

Now that you’re able to connect and interact with phpMyAdmin, all that’s left to do is harden your systems security to protect it from attackers.

Step 3 — Securing Your phpMyAdmin Instance

Because of its ubiquity, phpMyAdmin is a popular target for attackers, and you should take extra care to prevent unauthorized access. One of the easiest ways of doing this is to place a gateway in front of the entire application by using Apache's built-in `.htaccess` authentication and authorization functionalities.

To do this, you must first enable the use of `.htaccess` file overrides by editing your Apache configuration file.

Edit the linked file that has been placed in your Apache configuration directory:

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

Add an `AllowOverride All` directive within the `<Directory /usr/share/phpmyadmin>` section of the configuration file, like this:

/etc/apache2/conf-available/phpmyadmin.conf

```
<Directory /usr/share/phpmyadmin>
    Options FollowSymLinks
    DirectoryIndex index.php
    AllowOverride All
    . . .
```

SCROLL TO TOP

When you have added this line, save and close the file.

To implement the changes you made, restart Apache:

```
$ sudo systemctl restart apache2
```

Now that you have enabled `.htaccess` use for your application, you need to create one to actually implement some security.

In order for this to be successful, the file must be created within the application directory. You can create the necessary file and open it in your text editor with root privileges by typing:

```
$ sudo nano /usr/share/phpmyadmin/.htaccess
```

Within this file, enter the following information:

/usr/share/phpmyadmin/.htaccess

```
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /etc/phpmyadmin/.htpasswd
Require valid-user
```

Here is what each of these lines mean:

- `AuthType Basic`: This line specifies the authentication type that you are implementing. This type will implement password authentication using a password file.
- `AuthName`: This sets the message for the authentication dialog box. You should keep this generic so that unauthorized users won't gain any information about what is being protected.
- `AuthUserFile`: This sets the location of the password file that will be used for authentication. This should be outside of the directories that are being served. We will create this file shortly.
- `Require valid-user`: This specifies that only authenticated users should be given access to this resource. This is what actually stops unauthorized users from entering.

When you are finished, save and close the file.

The location that you selected for your password file was `/etc/phpmyadmin/.htpasswd`. You can now create this file and pass it an initial user with the `htpasswd` utility:

```
$ sudo htpasswd -c /etc/phpmyadmin/.htpasswd username
```

SCROLL TO TOP

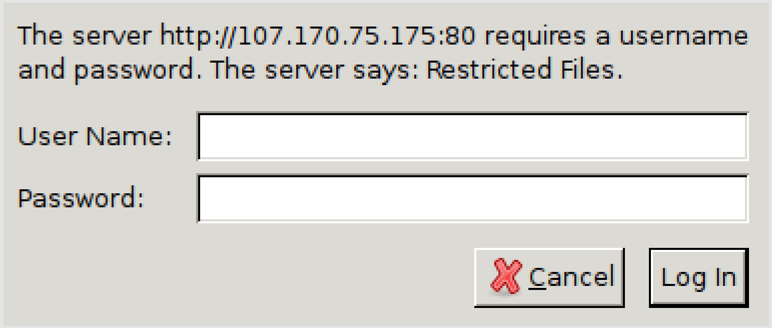
You will be prompted to select and confirm a password for the user you are creating. Afterwards, the file is created with the hashed password that you entered.

If you want to enter an additional user, you need to do so **without** the `-c` flag, like this:

```
$ sudo htpasswd /etc/phpmyadmin/.htpasswd additionaluser
```

Now, when you access your phpMyAdmin subdirectory, you will be prompted for the additional account name and password that you just configured:

```
https://domain_name_or_IP/phpmyadmin
```



The server http://107.170.75.175:80 requires a username and password. The server says: Restricted Files.

User Name:

Password:

After entering the Apache authentication, you'll be taken to the regular phpMyAdmin authentication page to enter your MySQL credentials. This setup adds an additional layer of security, which is desireable since phpMyAdmin has suffered from vulnerabilities in the past.

Conclusion

You should now have phpMyAdmin configured and ready to use on your Ubuntu 18.04 server. Using this interface, you can easily create databases, users, tables, etc., and perform the usual operations like deleting and modifying structures and data.

By: Mark Drake

Upvote (52) Subscribe Share

SCROLL TO TOP

Related Tutorials

How To Sync and Share Your Files with Seafile on Debian 9

How To Deploy a PHP Application with Kubernetes on Ubuntu 16.04

How To Install YunoHost on Debian 9

How To Install and Configure an Apache ZooKeeper Cluster on Ubuntu 18.04

How To Sync and Share Your Files with Seafile on Ubuntu 18.04

36 Comments

Leave a comment...

Log In to Comment

ballaquatics *May 22, 2018*

0 Something wrong with this tutorial..... I've run through it twice and I can't create database. Interface says No Privileges.

SCROLL TO TOP

Used the 16.04 tutorial with great success on 16.04 servers.....

mdrake MOD *May 25, 2018*

0 Hello ballaquatics!

Thank you for your comment and for pointing out this omission. I've updated the tutorial to include a step that describes how to adjust the **root** user's authentication method (allowing you to connect to phpMyAdmin as **root**) and create a new MySQL user with full privileges. I hope you find this new information helpful!

DemonMaestro *May 27, 2018*

2 I had noticed with ubuntu 18.04 that phpmyadmin v4.6.6deb5 keeps throwing these errors.

Warning in ./libraries/sql.lib.php#613
count(): Parameter must be an array or an object that implements Countable

How to fix that?

sudoRoot247 *June 10, 2018*

0 For the sql.lib.php error on line 613, please check thread below.

https://stackoverflow.com/questions/48001569/phpmyadmin-count-parameter-must-be-an-array-or-an-object-that-implements-co

lynnellemmanuel *June 26, 2018*

0 Here's the error my mine:

./libraries/plugin_interface.lib.php#551
count(): Parameter must be an array or an object that implements Countable

eugenegrechko *August 4, 2018*

0 **NVM Don't use this solution or the ones to fix the sql.lib.php warning.**
Both messages are warnings not errors. When you apply the given fixes they make phpmyadmin stop working which is much worse than a warning.

The solution to the problem can be found at the link below.

https://medium.com/@chaloemphonthipkasorn/%E0%B9%81%E0%B8%81%E0%B9%89-bug-phpmyadmin-php7-2-ubuntu-16-04-92b287090b01

Basically, follow these steps

1. sudo vi /usr/share/phpmyadmin/libraries/plugin_interface.lib.php
2. Got to line 551 (:set nu) then (:551)

SCROLL TO TOP

3. Find this `if ($options != null && count($options) > 0) {`
4. Make it look like this. Basically you put (array) in front of $options `if ($options != null && count((array)$options) > 0) {`

That's it! Good luck!

---

eugenegrechko *August 4, 2018*

0 I would leave it the way it is. When you run the fixes you get internal server errors which cause phpmyadmin to not work at all.

---

samhaiden *September 28, 2018*

0 Has anyone found a fix to these warnings/errors that work? Everything I try from this tutorial does not seem to work. Digital Ocean team - your articles are typically very good. Can you weigh in and help to correct these errors?

My Phpmyadmin is stuck at v. 4.6.6deb5 and am getting these warnings:

Warning in ./libraries/sql.lib.php#613
count(): Parameter must be an array or an object that implements Countable)

Backtrace

./libraries/display*import.lib.php#371: PMA*pluginGetOptions(
string 'Import',
array,
)
./libraries/display*import.lib.php#456: PMA*getHtmlForImportOptionsFormat(array)
./libraries/display*import.lib.php#691: PMA*getHtmlForImport(
string '5badbf9579854',
string 'server',
string '',
string '',
integer 16777216,
array,
NULL,
NULL,
string '',
)
./server*import.php#34: PMA*getImportDisplay(
string 'server',
string '',
string '',
integer 16777216,
)

Can anyone give some answers on how to successfully install this?

SCROLL TO TOP

linhadiretalipe *June 18, 2018*

3 great tutorial. I need to put one step to access phpmyadmin. I had to create a link in my ubuntu(18.04) :

sudo ln -s /usr/share/phpmyadmin /var/www/html

after that everything works good!

thanks!

---

perseus98 *July 6, 2018*

0 It really helped me lot.. Thank you to Team Digital Ocean.

---

ahmedalmulki *July 24, 2018*

1 What about Nginx , how can I choose nginx instead of apache2 ?

---

aarffy *August 8, 2018*

0 After these instructions you might find yourself needing this: https://askubuntu.com/questions/387062/how-to-solve-the-phpmyadmin-not-found-issue-after-upgrading-php-and-apache

---

fasalk *August 20, 2018*

0 How can we change default url ip/phpmyadmin to a customized example (ip/mydburl)?

---

justein230 *August 21, 2018*

0 I am trying to install php-gettext and php-mbstring, but they are not found in the default repositories. Any fixes for that?

---

busterftw *September 4, 2018*

0

```
sudo apt install phpmyadmin php-mbstring php-gettext
```

Gives
Unable to locate package phpmyadmin
Unable to locate package php-mbstring
Unable to locate package php-gettext

Can anyone give any insight on how to handle this?

SCROLL TO TOP

pablohalamaj *December 14, 2018*

0

Hello,

In ubuntu 18 the universe repo is disabled by default.

You can fix it running

$ sudo add-apt-repository universe

---

krupesh *September 20, 2018*

0 Thankyou So much for this detailed step by step process which was very easy to follow even for a Ubuntu Beginner like me.

---

louiearoy *September 26, 2018*

0 Hi I'm having these errors in phpmyadmin console

mysqli*real*connect(): (HY000/1045): Access denied for user 'phpmyadmin'@'localhost' (using password: YES) Connection for controluser as defined in your configuration failed.

And I don't see a phpmyadmin user when I run
mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;

---

samhaiden *October 4, 2018*

0 I found this blog very helpful: https://www.admintome.com/blog/install-phpmyadmin-on-ubuntu-18-04/

---

aubzp *October 21, 2018*

1 These instructions are good until the install phpmyadmin section, which does not work for me. All I get is the following when trying to load the http://###.###.###.###/phpmyadmin page.

Not Found

The requested URL /phpmyadmin was not found on this server.

Solution seems to be to put these 2 lines

---line 1
ServerName 127.0.0.1

--line 2
Include /etc/phpmyadmin/apache.conf

at the bottom of the /etc/apache2/apache2.conf file, save then restart apache.

Works for me now.

Just a note or request, as the passwords to phpmyadmin are sent without encryption per
how to quickly SSL these pages could be included?

SCROLL TO TOP

zy *October 22, 2018*

0 For some reasons the interface was returning a 404 error:(server_ip/phpmyadmin)

Fixed with

```
sudo vim /etc/apache2/apache2.conf
```

Then add the following line to the end of the file:

```
Include /etc/phpmyadmin/apache.conf
```

Then restart apache:

/etc/init.d/apache2 restart

from this forum

aubzp *October 25, 2018*

0 I used this sudo nano /usr/share/phpmyadmin/.htaccess and set a password to limit access to the phpmyadmin home page and it worked fine the first time I accessed that page, ie. it asked for a valid user and granted access once it was satisfied. I then created a database etc, logged out, went to another page, came back to the phpmyadmin home page and did not get the expected valid user challenge but was allowed straight into the phpmyadmin home page. Is this secure? Just what is happening? Is the .htaccess permissions storing a cookie I cant see or what? I would prefer that if I have left the page that when I returned I would be required to reenter my valid username and password creadentials before being able to access the phpmyadmin home page or am I just being too paranoid?

jeremc *October 25, 2018*

0 It seems the restriction page creates cookies once you are connected but they expire on session. This means that when your close your browser those cookies will be deleted. Cookies aren't dangerous on their own, if somebody take yours they will only have encrypted information so they probably won't even be able to connect to your account.
You can also setup your browser in the preferences section so he won't store any cookies.

daveOTlinux *November 2, 2018*

0 Another example of putting the warning after the command being warned about. It would be useful if the warning comes before the command, as it give the person following the instructions a chance to see the warning, before using the command. This document the way it is, the warning is of no help, except at least we know what we did wrong.

SCROLL TO TOP

aubzp *November 6, 2018*

2 Yes you were correct when I completely closed firefox the cookies were flushed and I needed to login again. All is cool. Good tute.

ahmadtahhan92 *November 8, 2018*

0 When i try to access phpmyadmin I am getting a 404 page, The requested URL /phpmyadmin was not found on this server.

how can I find what the problem is ?

mdrake MOD *November 8, 2018*

1 Hello @ahmadtahhan92!

There are a number of reasons why you may be seeing a `404 Not Found` page but, when it comes to phpMyAdmin, the most common cause is that `apache2` wasn't selected in the installation prompt. The most straightforward workaround I could find for this problem is as follows.

First, open the `apache2.conf` file:

```
$ sudo nano /etc/apache2/apache2.conf
```

Then add the following line somewhere in the file:

/etc/apache2/apache2.conf

```
. . .
Include /etc/phpmyadmin/apache.conf
. . .
```

Save and close the file, then restart Apache:

```
$ sudo systemctl restart apache2
```

Then refresh the page in your browser. Assuming that was indeed the problem, you will see the phpMyAdmin login screen.

ahmadtahhan92 *November 9, 2018*

1 Nice, worked perfectly, thank you so much :)

SCROLL TO TOP

Load More Comments