

Unit 1. Task 2: Apache + MySQL + PHP + phpMyAdmin installation in Ubuntu 18.04

Javier Aguilera Aguilera 2º CFGS DAW

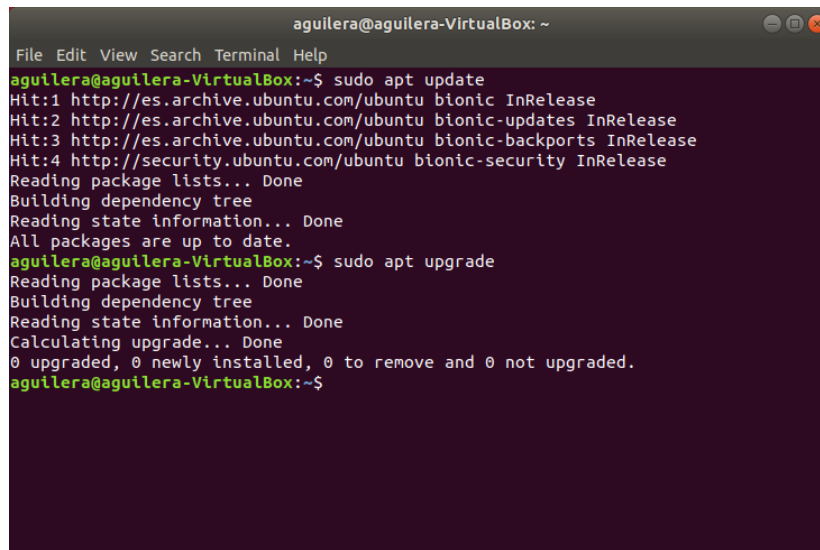
Contents

1	Previous steps	3
2	Apache	3
3	MySQL	5
3.1	Trying a connection with MySQL	6
4	PHP	7
4.1	Testing PHP	8
5	phpMyAdmin	9

1 Previous steps

Before making the installation, we are going to update our system executing in the terminal the following commands:

```
$ sudo apt update
$ sudo apt upgrade
```

A terminal window titled 'aguilera@aguilera-VirtualBox: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'sudo apt update' and 'sudo apt upgrade'. The output for 'update' shows hits for bionic, bionic-updates, bionic-backports, and bionic-security, followed by 'Reading package lists... Done', 'Building dependency tree', and 'Reading state information... Done'. The output for 'upgrade' shows 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', 'Calculating upgrade... Done', and '0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.'.

```
aguilera@aguilera-VirtualBox: ~
File Edit View Search Terminal Help
aguilera@aguilera-VirtualBox:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
aguilera@aguilera-VirtualBox:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
aguilera@aguilera-VirtualBox:~$
```

Figure 1: We have our system up to date

2 Apache

In order to install Apache we type in the terminal

```
$ sudo apt install apache2
```

Launching the web browser and typing in the search bar *localhost*, you can see if it works.

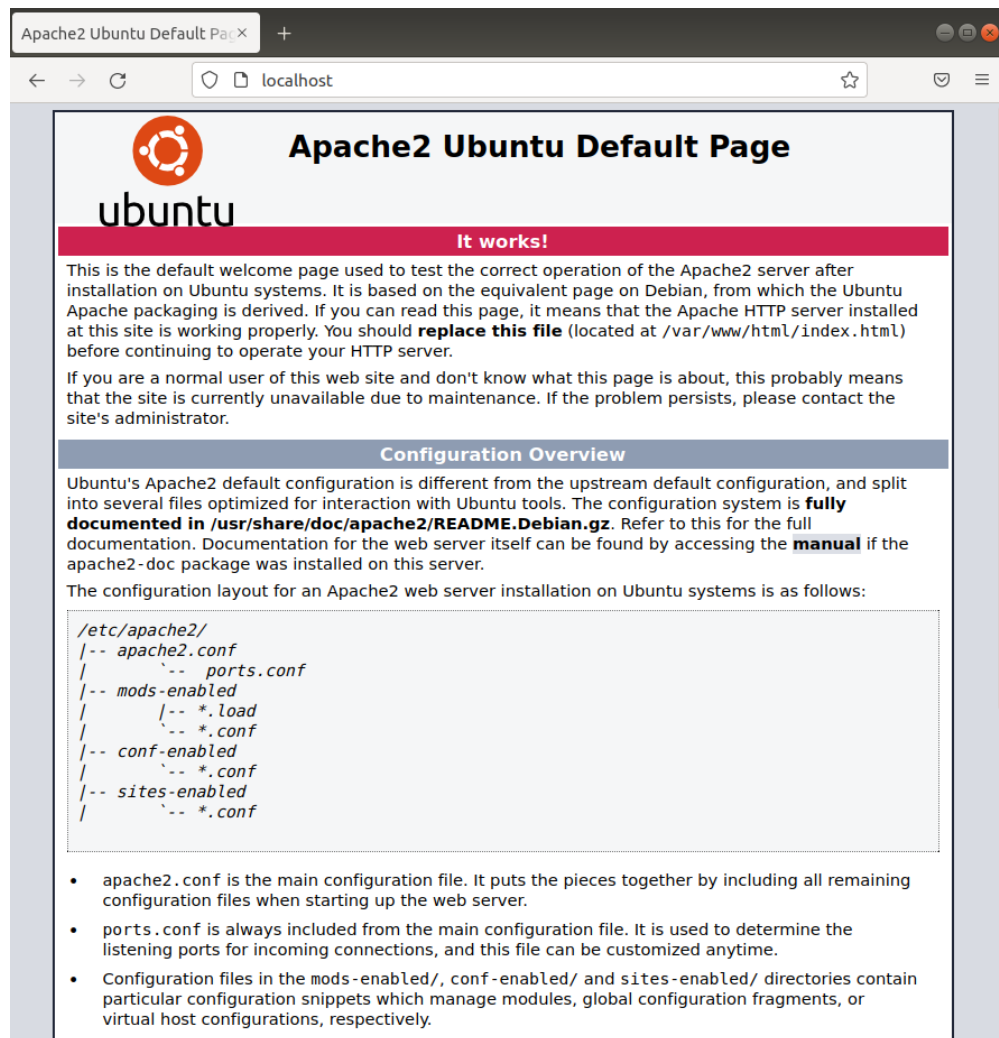
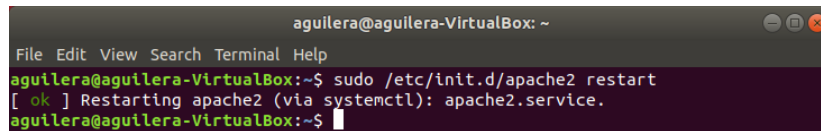


Figure 2: It works!

In case that it doesn't work you can reboot Apacher Server typing in the terminal

```
$ sudo /etc/init.d/apache2 restart
```

A terminal window titled 'aguilera@aguilera-VirtualBox: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'sudo /etc/init.d/apache2 restart' being executed. The output is '[ok] Restarting apache2 (via systemctl): apache2.service.' followed by a new prompt.

```
aguilera@aguilera-VirtualBox: ~  
File Edit View Search Terminal Help  
aguilera@aguilera-VirtualBox:~$ sudo /etc/init.d/apache2 restart  
[ ok ] Restarting apache2 (via systemctl): apache2.service.  
aguilera@aguilera-VirtualBox:~$
```

Figure 3: Restarting Apache

3 MySQL

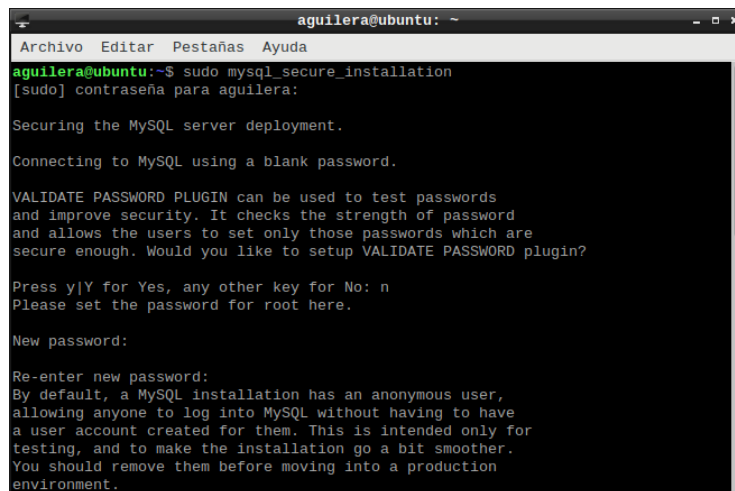
You have to type in the terminal

```
$ sudo apt install mysql-server
```

Later on, if you want to set a password for the administrator account (root) then type

```
$ sudo mysql_secure_installation
```

This command is going to ask us for several parameters to set. In the figures 4, 5 y 6 you can see the configuration that I have adopted:

A terminal window titled 'aguilera@ubuntu: ~' with a menu bar (Archivo, Editar, Pestañas, Ayuda). The terminal shows the output of the 'mysql_secure_installation' script. It prompts for a password for 'aguilera', then asks about the 'VALIDATE PASSWORD PLUGIN'. The user responds with 'n' for no. The script then prompts for a new password for 'root' and asks to re-enter it. Finally, it displays a warning about the anonymous user.

```
aguilera@ubuntu: ~  
Archivo Editar Pestañas Ayuda  
aguilera@ubuntu:~$ sudo mysql_secure_installation  
[sudo] contraseña para aguilera:  
  
Securing the MySQL server deployment.  
  
Connecting to MySQL using a blank password.  
  
VALIDATE PASSWORD PLUGIN can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD plugin?  
  
Press y|Y for Yes, any other key for No: n  
Please set the password for root here.  
  
New password:  
  
Re-enter new password:  
By default, a MySQL installation has an anonymous user,  
allowing anyone to log into MySQL without having to have  
a user account created for them. This is intended only for  
testing, and to make the installation go a bit smoother.  
You should remove them before moving into a production  
environment.
```

Figure 4: Screenshot_1

```
aguilera@ubuntu: ~  
Archivo  Editar  Pestañas  Ayuda  
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
Normally, root should only be allowed to connect from  
'localhost'. This ensures that someone cannot guess at  
the root password from the network.  
  
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
By default, MySQL comes with a database named 'test' that  
anyone can access. This is also intended only for testing,  
and should be removed before moving into a production  
environment.  
  
Remove test database and access to it? (Press y|Y for Yes, any other key for No)  
: y  
- Dropping test database...  
Success.  
  
- Removing privileges on test database...  
Success.
```

Figure 5: Screenshot_2

```
aguilera@ubuntu: ~  
Archivo  Editar  Pestañas  Ayuda  
Success.  
  
By default, MySQL comes with a database named 'test' that  
anyone can access. This is also intended only for testing,  
and should be removed before moving into a production  
environment.  
  
Remove test database and access to it? (Press y|Y for Yes, any other key for No)  
: y  
- Dropping test database...  
Success.  
  
- Removing privileges on test database...  
Success.  
  
Reloading the privilege tables will ensure that all changes  
made so far will take effect immediately.  
  
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
All done!  
aguilera@ubuntu:~$
```

Figure 6: Screenshot_3

3.1 Trying a connection with MySQL

Now we are going to try if the installation is right. To achieve that purpose, at the terminal we are going to run

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

```
aguilera@aguilera-VirtualBox: ~
File Edit View Search Terminal Help
aguilera@aguilera-VirtualBox:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Figure 7: Here we have MySQL prompt working

A easy way to try MySQL is executing at the command prompt:

> show databases ;

```
aguilera@ubuntu: ~
Archivo Editar Pestañas Ayuda
Server version: 5.7.35-0ubuntu0.18.04.2 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| phpmyadmin |
| sys       |
+-----+
5 rows in set (0,00 sec)

mysql>
```

Figure 8: Included databases in MySQL

If you want to comeback to bash we run

> exit

4 PHP

The generic command that install the newest version of PHP that our Linux based operating system support is:

```
$ sudo apt install php
```

Then we restart Apache:

```
$ sudo /etc/init.d/apache2 restart
```

4.1 Testing PHP

From Ubuntu 14.04 and later versions, */var/www/html* is the specific directory that Apache uses to find and serves archives. In previous versions was */var/www*. As we want to try if the system is working fine, we have to change the owner of the directory */var/www/* and its subdirectories because they belong to root user and we want to have permissions to make changes in them:

```
$ sudo chown -R $USER:root /var/www
```

After that, we are going to make a simple PHP file in */var/www/html*. First of all, we have to move into the directory:

```
$ cd /var/www/html
```

Then we type in the terminal

```
$ gedit info.php
```

If you don't have gedit installed you can use nano or vim in its place. When it is opened, you write the following content:

```
<?php
    phpinfo();
?>
```

Then save it and close the editor.

Now, you have to open your favorite web browser and in the search bar write *localhost/info.php*

System	Linux ubuntu 4.15.0-161-generic #169-Ubuntu SMP Fri Oct 15 13:41:54 UTC 2021 x86_64
Build Date	Aug 16 2021 05:46:32
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

Figure 9: info.php

5 phpMyAdmin

Using *apt* package system we download and install phpMyAdmin in our system:

```
$ sudo apt install phpmyadmin
```

Several popup windows are going to appear during the instalation:

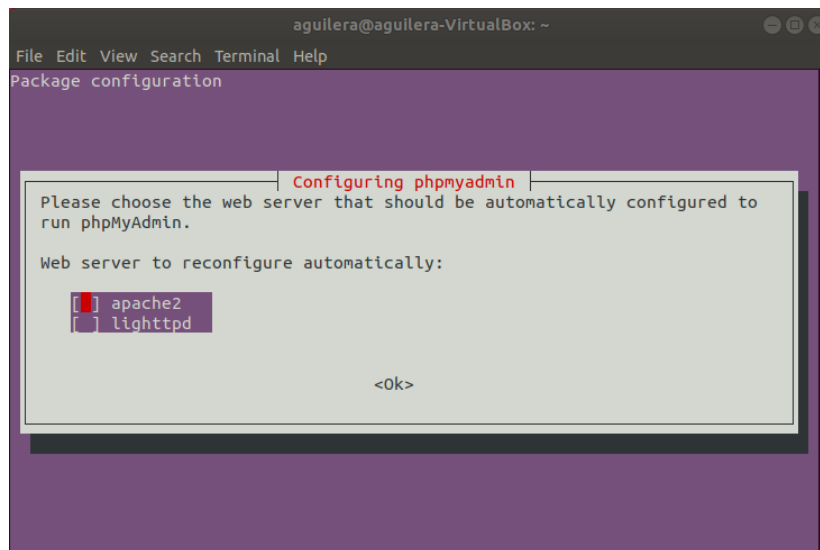


Figure 10: Choosing web server

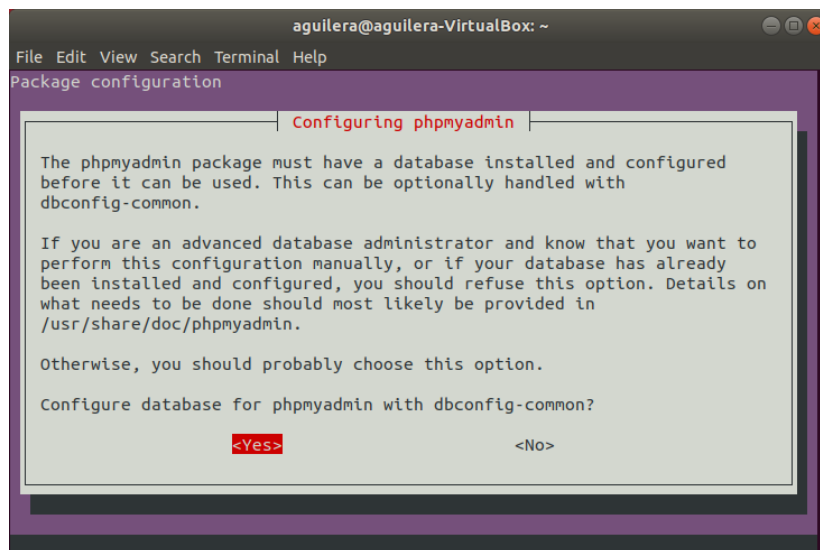


Figure 11: We have to choose “Yes”

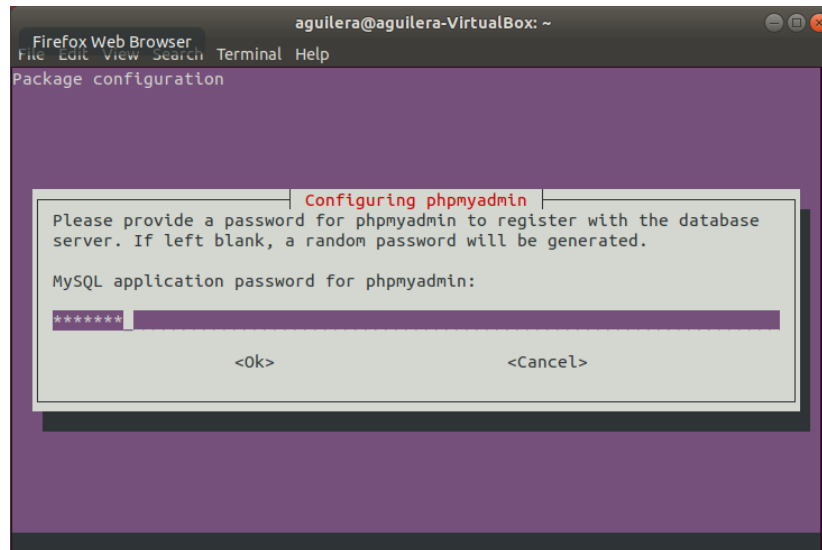


Figure 12: Then it asks for a password, that later we confirm

After the installation, you have to run the browser and type at the search bar */localhost/phpmyadmin*.

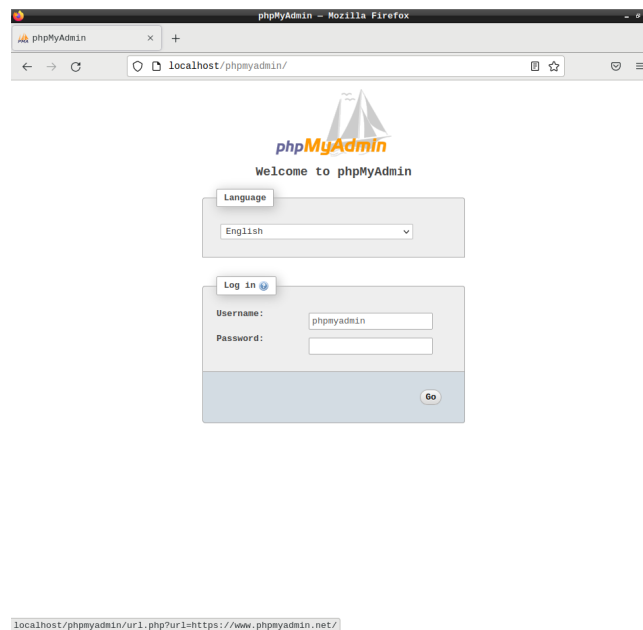


Figure 13: Login phpMyAdmin page

Login by the username *phpmyadmin* and the password that you set during the installation (Figure 12), it will charge phpMyAdmin home page.

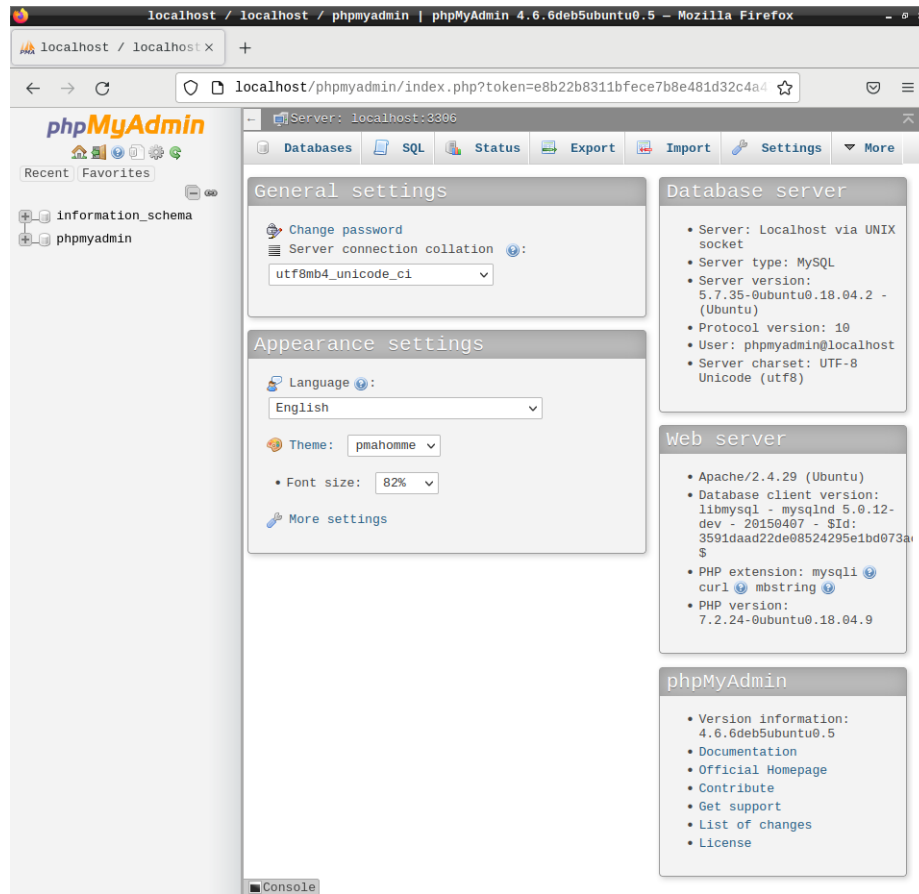


Figure 14: phpMyAdmin home page

If we want to enter as a root then we have to change the authentication method of root in MySQL.

First of all, we have to run MySQL prompt as root:

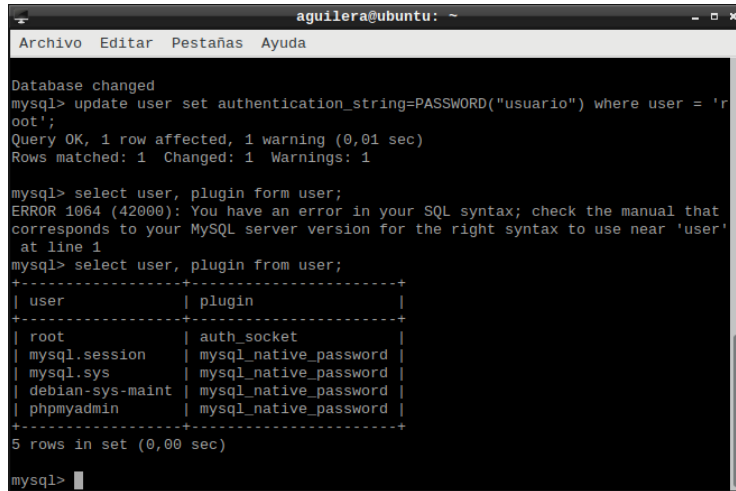
```
$ sudo mysql -u root -p
```

Then execute the following SQL statements:

```
> use mysql;
```

```
> update user set authentication_string=PASSWORD("your_root_password_here")  
where user = 'root';
```

```
> select user , plugin from user;
```



```
Database changed
mysql> update user set authentication_string=PASSWORD("usuario") where user = 'root';
Query OK, 1 row affected, 1 warning (0,01 sec)
Rows matched: 1 Changed: 1 Warnings: 1

mysql> select user, plugin form user;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'user'
at line 1
mysql> select user, plugin from user;
+-----+-----+
| user          | plugin          |
+-----+-----+
| root          | auth_socket     |
| mysql.session | mysql_native_password |
| mysql.sys     | mysql_native_password |
| debian-sys-maint | mysql_native_password |
| phpmyadmin    | mysql_native_password |
+-----+-----+
5 rows in set (0,00 sec)

mysql>
```

Figure 15: Changing root authentication method

As we can see at the table returned in figure 15, the root user has a different authentication method. We are going to change it because in a development environment is useful to have access to the root user. In a production environment we will set different users with different privileges. As we have to update the table we must execute an update statement:

```
> update user set plugin="mysql_native_password" where user = "root";
```

Now, let see if it works:

```
> select user , plugin from user;
```

```
> flush privileges;
```

```
> exit;
```

That's all, if we enter with the browser in */localhost/phpmyadmin* using the root account it will appear the phpMyAdmin home page.

References

[1] https://www.youtube.com/watch?v=YggXN_xJKbs