

Lab Cycle: 3

Experiment No.: 2

Date: 7-06-2022

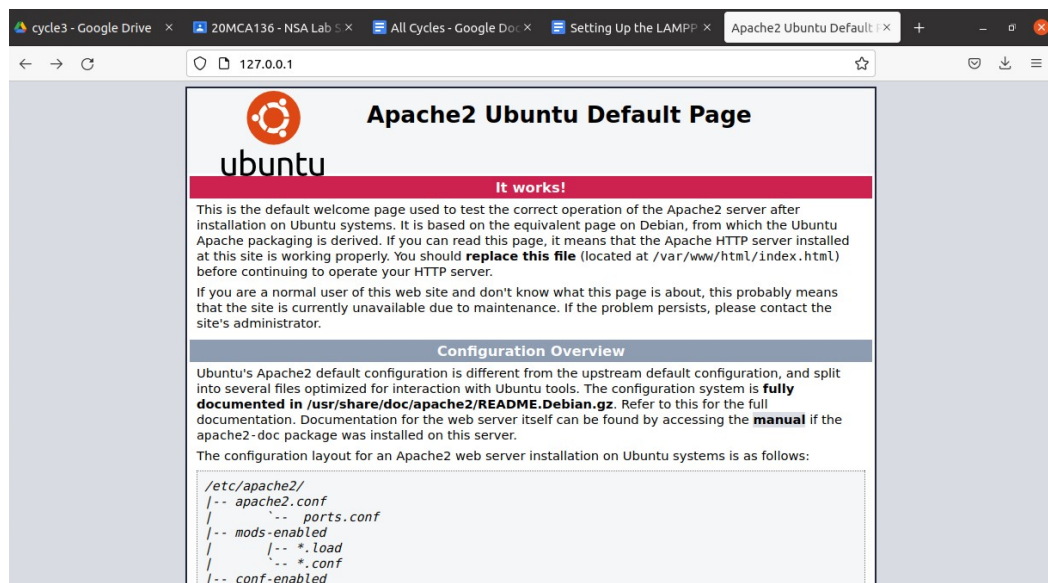
Aim: Illustrate the steps involved in installing the LAMP Stack on a Linux machine.
Deploy phpMyadmin.

Steps to installing the Lamp Stack

A. Steps to installing apache2 server, mariadb and php.

1. **sudo apt update** *[Update the repositories information]*
2. **sudo apt install apache2** *[Install apache2 Web Server]*
3. **sudo systemctl status apache2** *[Test the installation]*
4. From Guest check 127.0.0.1 from the browser Or From Host check the IP address of Guest (use **ifconfig** to get it)

You should see the start page of Apache



6. **sudo apt install mariadb-server mariadb-client** *[Install Mariadb Server and Client]*

7. **sudo mysql_secure_installation** *[Make Mariadb Secure]*

1. Set the root password
2. Disable Anonymous users
- c. Discard test database, etc

```

user@user-VirtualBox:~$ sudo mysql_secure_installation
[sudo] password for user:

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB 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.

Remove anonymous users? [Y/n] 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? [Y/n] y
... Success!

By default, MariaDB 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? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...

```

8. `sudo apt install php libapache2-mod-php php-opcache php-cli php-gd php-curl php-mysql` [*Install PHP and commonly used packages*]

9. `sudo systemctl restart apache2` [*Restart apache2*]

10. To test the PHP installation :

Create a file “test.php” with the following contents

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

Copy test.php to Web root

11. `sudo cp test.php /var/www/html`

12. Take the browser and check with `127.0.0.1/test.php`

You should get the output from `phpinfo()` as shown below

| | |
|---|---|
| 192.168.1.5/test.php | |
| PHP Version 7.4.3 | |
| System | Linux user-VirtualBox 5.13.0-41-generic #46--20.04.1-Ubuntu SMP Wed Apr 20 13:16:21 UTC 2022 x86_64 |
| Build Date | Mar 2 2022 15:36:52 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/7.4/apache2 |
| Loaded Configuration File | /etc/php/7.4/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php/7.4/apache2/conf.d |
| Additional .ini files parsed | /etc/php/7.4/apache2/conf.d/10-mysqld.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/15-xml.ini, /etc/php/7.4/apache2/conf.d/20-bc2.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-curl.ini, /etc/php/7.4/apache2/conf.d/20-dom.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gd.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mbstring.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-simplexml.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini, /etc/php/7.4/apache2/conf.d/20-xmlreader.ini, /etc/php/7.4/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.4/apache2/conf.d/20-xsl.ini, /etc/php/7.4/apache2/conf.d/20-zip.ini |
| PHP API | 20190902 |
| PHP Extension | 20190902 |
| Zend Extension | 320190902 |
| Zend Extension Build | API320190902.NTS |
| PHP Extension Build | API20190902.NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Signal Handling | enabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | provided by mbstring |
| IPv6 Support | enabled |
| DTrace Support | available, disabled |
| Registered PHP Streams | http, ftp, compress.zlib.php, file, glob, gpc, http, https, imap, ldap, ldaps, mysql, nntp, php, ssh, soap, tcp, tls, udp, unix, wddx, xml, xmlrpc, zip, zstd |

B. Preparations for installing phpmyadmin.

13. **sudo mysql** (-u root is assumed since you use sudo) [*First login to mysql*]

14. **CREATE USER 'phpmyadmin'@'localhost' IDENTIFIED BY 'phpmyadminpassword';**

15. **GRANT ALL PRIVILEGES ON *.* TO 'phpmyadmin'@'localhost';** [*Grant privileges to phpmyadmin*]

16. **FLUSH PRIVILEGES;**

17. To Change the Mysql root password:

1. **sudo mysql** [first login to mysql]

2. **ALTER USER root@localhost IDENTIFIED BY "rootpassword";**

3. **USE mysql;**

4. **UPDATE user SET plugin='mysql_native_password' WHERE user='root';**

5. **FLUSH PRIVILEGES;**

6. **exit;**

18. Steps to install phpmyadmin:

1. **sudo apt install phpmyadmin php-mbstring php-zip php-json**

2. For configuring the web server, select apache2.

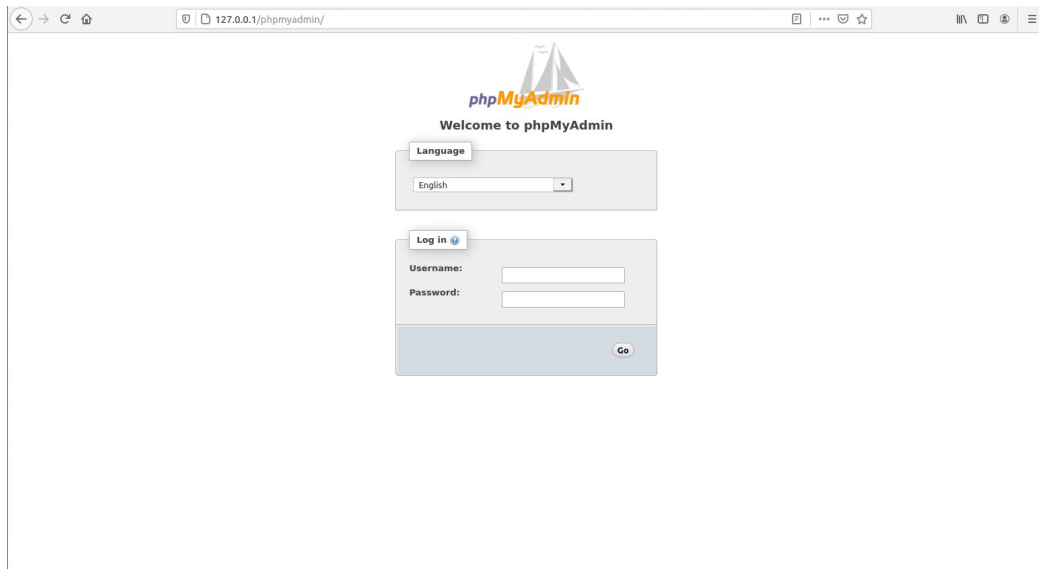
3. Select db-configuration and set the password for phpmyadmin.

4. **sudo systemctl restart apache2** [*Restart apache2*]

5. **sudo ln -s /usr/share/phpmyadmin/ /var/www/html/phpmyadmin**
[create a link to the phpmyadmin in the webroot folder]

6. Take the browser and check with *127.0.0.1/phpmyadmin*

You should get the phpmyadmin login page as shown below



You can use either phpmyadmin as Username and its password (phpmyadminpassword, in our case) as Password Or root as Username and rootpassword as Password.

On successful login you should get the following page where you can do any mysql activity.

