How To Install LAMP Stack on Ubuntu 22.04 LTS

The LAMP (Linux, Apache, MySQL, and PHP) stack is wildly used for deploying PHP-based applications on Linux systems. The LAMP server installation is pretty easy and straightforward. You need some basic knowledge of the Linux package manager to complete this setup.

Here Linux is an open-source operating system. Apache is a popular web server. MySQL is a relational database server and PHP is the programming language.

This tutorial will help you to install PHP, Apache & MySQL on Ubuntu 22.04 LTS Linux system.

Before We Begin

Assuming that you have a running Ubuntu 22.04 Linux system with sudo (or root) privileged access. Access your system and open a terminal. It will be good to update package manager cache and upgrade currently installed packages. To do this execute:

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

PHP Installation

The ondrej/php ppa contains all PHP version's for Ubuntu systems. So add this repository in your system with command below:

```
$ sudo add-apt-repository ppa:ondrej/php
```

Now update the apt cache and install PHP 8.1.

```
$ sudo apt update
$ sudo apt install php8.1
```

Apache Installation

Generally, the PHP installation also installs the Apache and its module on your system. Still, you can run the following commands to confirm the installations.

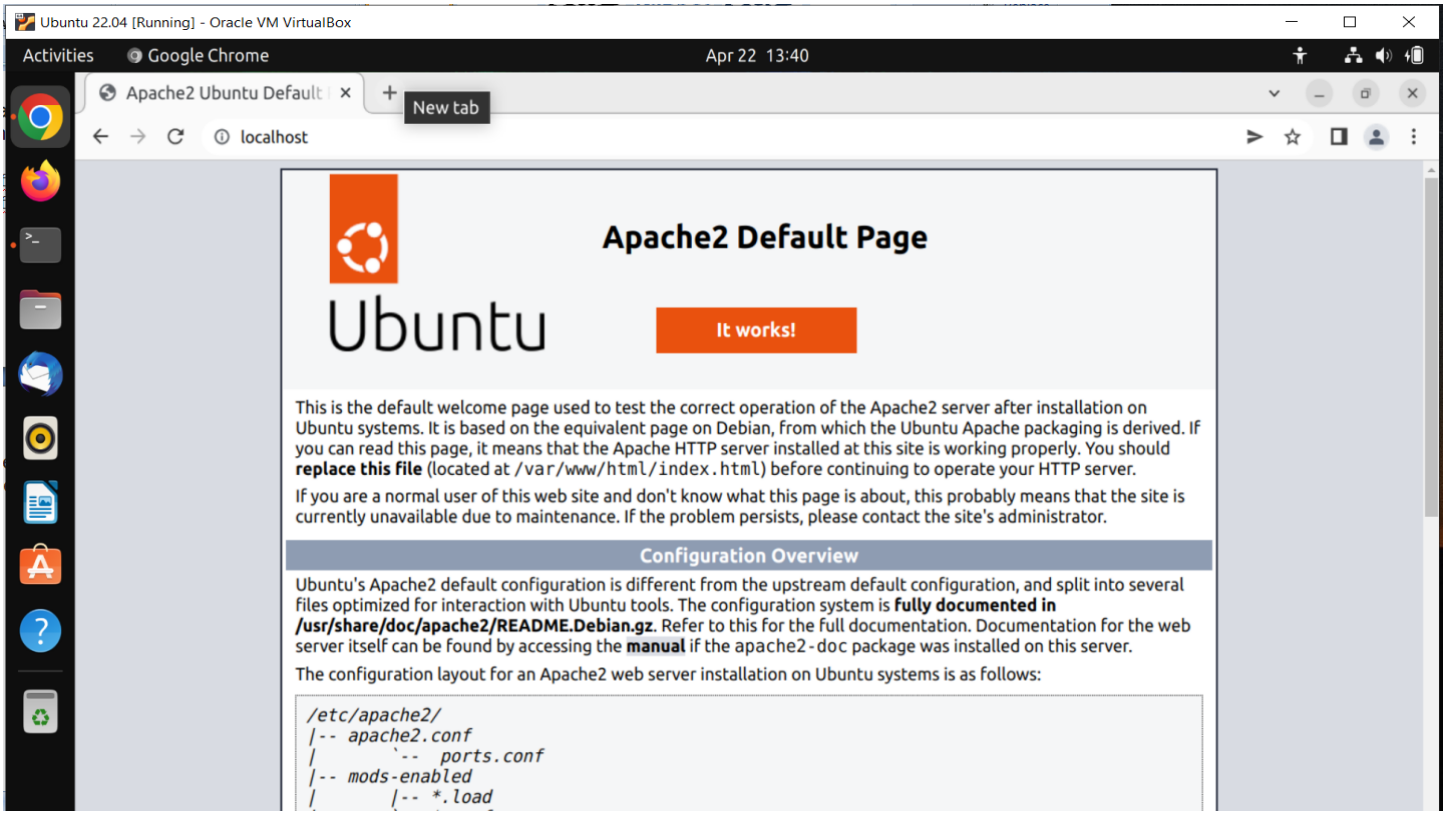
```
$ sudo apt install apache2 libapache2-mod-php8.1  -y
```

This will install Apache and start the service.

Now, you need to allow webserver ports in the firewall. To allow ports 80 and 443 in the UFW firewall, execute the following commands.

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

Open a web browser on your system and type the server's IP in the address bar. You will get the default Apache server page.



MySQL Installation

The default Ubuntu repositories contain MySQL 8.0. Which can be directly installed using the package manager. To install the available MySQL server version, execute the following command.

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

Once the installation is finished, you can secure the MySQL server by executing the following command.

```
$ sudo mysql_secure_installation
```

This will ask for a few questions to secure the MySQL server.

1. Press **'n'** to disable validate password plugin. This will allow you to set a simple password (**letmein**) for all user accounts. Remember that this virtual machine is only for educational purposes only. Simple password will save us lots of grieve and time. Of course, that will be a totally different story if this is a production server, strict password policy is absolutely required!

2. Enter a new password and re-enter it. Please enter a simple password: **letmein**

   New password: *******
   Re-enter new password: *******

3. Press 'y' to continue with provided password.

```
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No): y
```

4. Remove default anonymous users from MySQL server:

```
Remove anonymous users? (Press y|Y for Yes, any other key for No): y
```

5. Enable root login from remote systems

```
Disallow root login remotely? (Press y|Y for Yes, any other key for No): n
```

6. Remove test database form MySQL created by default during installation.

```
Remove test database and access to it? (Press y|Y for Yes, any other key for No): y
```

7. Reload all privileges to apply above changes immediately.

```
Reload privilege tables now? (Press y|Y for Yes, any other key for No): y
```

Remember that the above password set for the root accounts is used for remote users only. To log in from the same system, just type **mysql** on terminal.

```
$ sudo mysql
```

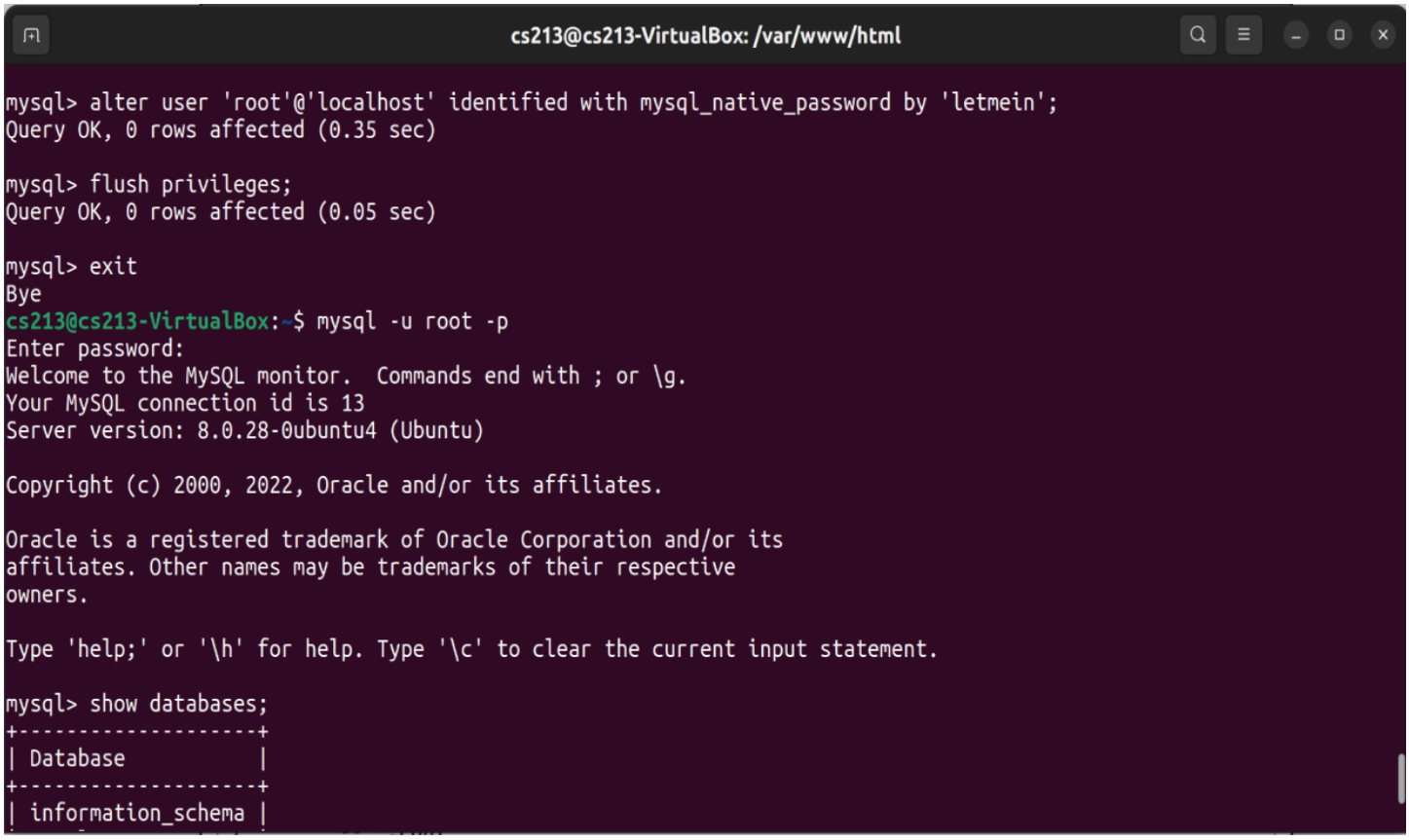
```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14 Server version: 8.0.28-0ubuntu4 (Ubuntu)
Copyright (c) 2000, 2022, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Now let’s create a local root user password for MySQL server with the same simple password (letmein) => then exit => then try logging in as root user with the password (letmein)

```
mysql>ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'letmein';
mysql>FLUSH PRIVILEGES;
mysql> exit

$ mysql –u root -p
```



If you can log in successfully, then you have successfully installed LAMP server using a simple password (letmein) for MySQL server.

Installing Other Required Packages

You may also need to install modules like MySQL and other extensions for PHP based on the application requirements. Let's begin installation of some other common modules:

```
$ sudo apt install php8.1-mysql php8.1-curl php8.1-xml php8.1-mbstring
php8.1-xdebug
```

Verify Setup

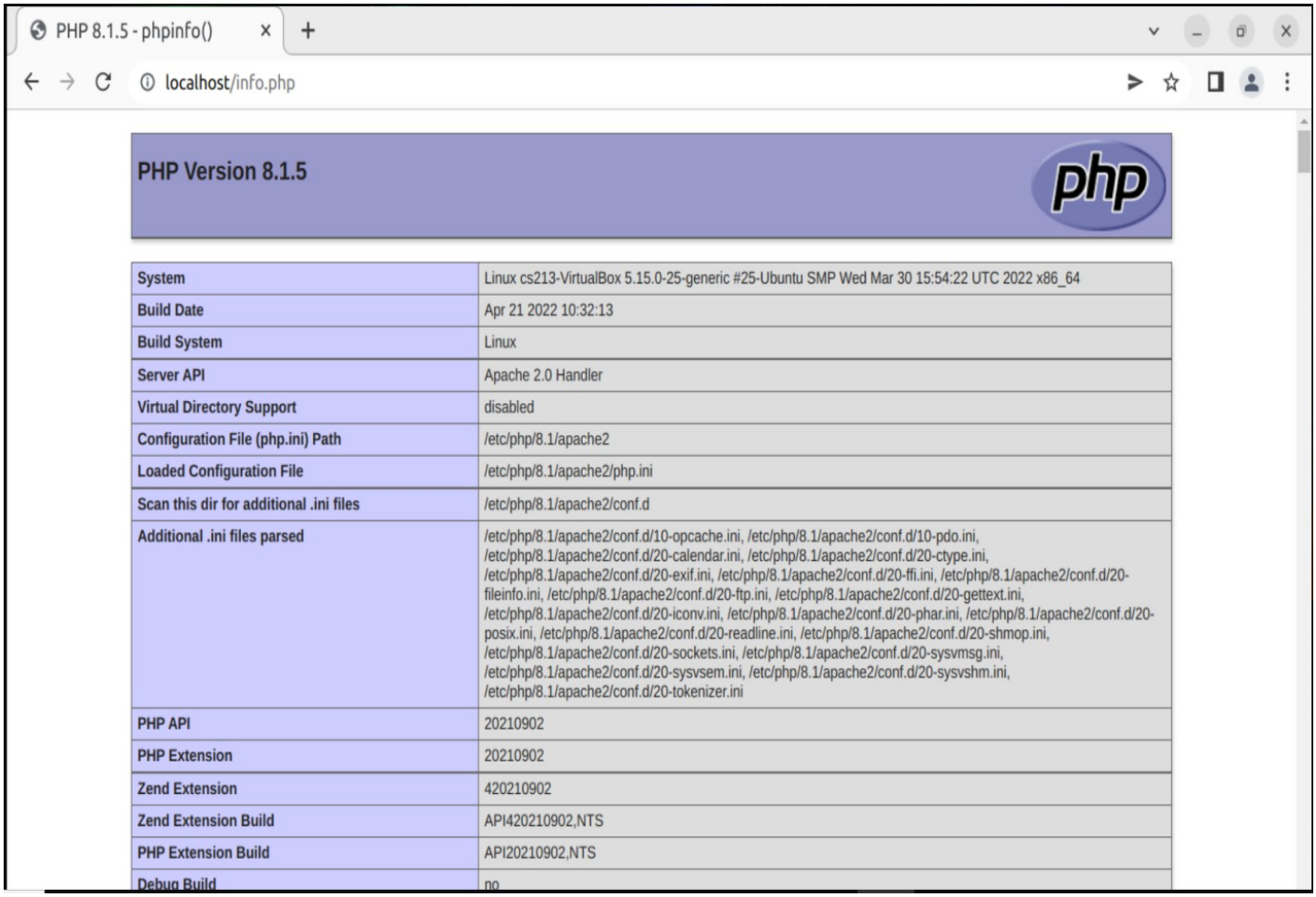
You have successfully completed the installation of Apache, MySQL, and PHP on the Ubuntu 22.04 Linux system. To verify the PHP integration with a php script file to be created by **gedit** editor:

```
$ gedit
```

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

Save the script file as **info.php**, and then access this file in the web browser (see below). It will show all the details about versions and installation.

http://localhost/info.php



System	Linux cs213-VirtualBox 5.15.0-25-generic #25-Ubuntu SMP Wed Mar 30 15:54:22 UTC 2022 x86_64
Build Date	Apr 21 2022 10:32:13
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no

PHP details by phpinfo() function.