What is a Technology stack?

A technology stack is a set of frameworks and tools used to develop a software product. This set of frameworks and tools are very specifically chosen to work together in creating well-functioning software. They are acronyms for individual technologies used together for a specific technology product. some examples are...

- **LAMP** (Linux, Apache, MySQL, PHP or Python, or Perl)
- **LEMP** (Linux, Nginx, MySQL, PHP or Python, or Perl)
- **MERN** (MongoDB, ExpressJS, ReactJS, NodeJS)
- **MEAN** (MongoDB, ExpressJS, AngularJS, NodeJS

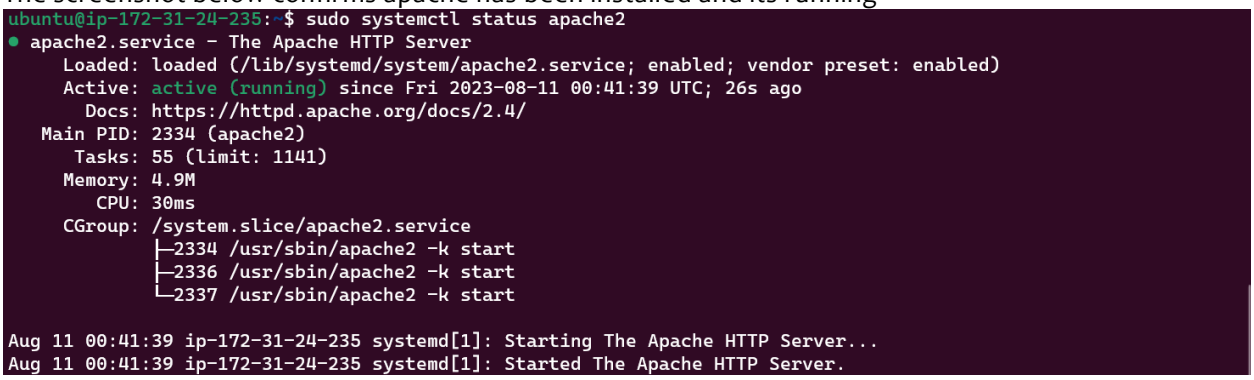
However, on this project, I will be working on LAMP(Linux, Apache, MySQL, and PHP) Stack

Step 1 — Install Apache and update the firewall.

I'm using the AWS platform for this where I have launched an ec2 instance and I can have access to it on my local machine.

Run the following commands to install and verify apache has been installed
sudo apt update
sudo apt install apache2
sudo systemctl status apache2

The screenshot below confirms apache has been installed and its running



let us try to check how we can access it locally in our Ubuntu shell, run:

curl http://localhost:80 or curl http://127.0.0.1:80

```
ubuntu@ip-172-31-24-235:~$ curl http://localhost:80
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
    padding: 3px 3px 3px 3px;

    background-color: #D8DBE2;

    font-family: Ubuntu, Verdana, sans-serif;
    font-size: 11pt;
    text-align: center;
  }

  div.main_page {
    position: relative;
    display: table;

    width: 800px;

    margin-bottom: 3px;
    margin-left: auto;
    margin-right: auto;
    padding: 0px 0px 0px 0px;

    border-width: 2px;
    border-color: #212738;
    border-style: solid;

    background-color: #FFFFFF;

    text-align: center;
  }

  div.page_header {
    height: 180px;
    width: 100%;
```

Now it is time for us to test how our Apache HTTP server can respond to requests from the Internet. Open a web browser of your choice and try to access the following URL *http://<Public-IP-Address>:80 or the Public IPv4 DNS(remember to remove the s in 'https' when doing that)*

Step 2 — Installing MySQL

sudo apt install mysql-server -y
Ran this command to install my sql server

sudo mysql
Used this command to log into the sql

```
ubuntu@ip-172-31-24-235:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.33-0ubuntu0.22.04.4 (Ubuntu)

Copyright (c) 2000, 2023, 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.
```

This command below will remove some insecure default settings and lock down access to your database system.
We're defining this user's password as PassWord.1 for the root user

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'PassWord.1';

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'PassWord.1';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Then exit the shell by using 'ctrl d' or type exit

Then we run this script sudo mysql_secure_installation

This script configures security settings and allows you to: Set a password for root accounts and remove the root accounts accessible from outside the local host, remove anonymous user accounts, delete the test database, accessible by anonymous users, and reload the user privileges tables.

```
ubuntu@ip-172-31-24-235:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

VALIDATE PASSWORD COMPONENT 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 component?

Press y|Y for Yes, any other key for No: Y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary                  file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1
Using existing password for root.

Estimated strength of the password: 100
Change the password for root ? ((Press y|Y for Yes, any other key for No) : y

New password:

Re-enter new password:

Estimated strength of the password: 50
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
 ... Failed! Error: Your password does not satisfy the current policy requirements

New password:

Re-enter new password:

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
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.
```

```
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.

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!
ubuntu@ip-172-31-24-235:~$
```

When you're finished, test if you're able to log in to the MySQL console by typing:

sudo mysql -p

```
ubuntu@ip-172-31-24-235:~$ sudo mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.33-0ubuntu0.22.04.4 (Ubuntu)

Copyright (c) 2000, 2023, 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>
```

Step 3 — Installing PHP

Apache is set up to deliver your content, while MySQL is installed to oversee your data. PHP is responsible for handling code to present interactive information to users. Besides the php package, you'll require php-mysql, a PHP module facilitating interaction with MySQL databases. Also, libapache2-mod-php is needed to empower Apache in managing PHP files. The essential PHP core packages will be automatically installed as prerequisites.

To install these 3 packages at once, run:

sudo apt install php libapache2-mod-php php-mysql -y and confirm version by running this php -v

```
ubuntu@ip-172-31-24-235:~$ php -v
PHP 8.1.2-1ubuntu2.13 (cli) (built: Jun 28 2023 14:01:49) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.13, Copyright (c), by Zend Technologies
```

Step 4 — Create a virtual host for your website using Apache

We will setup a virtual host to test the PHP script, virtual host enables you to setup multiple websites on a single server.

Create the directory for projectlamp using 'mkdir' command

sudo mkdir /var/www/projectlamp

Create and open a new configuration file in Apache's sites-available directory.

sudo vi /etc/apache2/sites-available/projectlamp.conf

This will create a new blank file, using 'i' to get to insert mode and paste the text below, the use :wq to write,save and quit

```
<VirtualHost *:80>
   ServerName projectlamp
   ServerAlias www.projectlamp
   ServerAdmin webmaster@localhost
   DocumentRoot /var/www/projectlamp
   ErrorLog ${APACHE_LOG_DIR}/error.log
   CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Use ls command to list the new file in the sites-available directory.

sudo ls /etc/apache2/sites-available

The output of the above command should be similar to this below

```
ubuntu@ip-172-31-24-235:~$ sudo ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf  projectlamp.conf
```

Let us enable the new virtual host with the a2ensite command:
sudo a2ensite projectlamp

disable Apache's default website use a2dissite command:
sudo a2dissite 000-default

To make sure your configuration file doesn't contain syntax errors, run:
sudo apache2ctl configtest

Finally, reload Apache so these changes take effect:
sudo systemctl reload apache2

```
ubuntu@ip-172-31-24-235:~$ sudo a2ensite projectlamp
Enabling site projectlamp.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-24-235:~$ sudo a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-24-235:~$ sudo apache2ctl configtest
Syntax OK
ubuntu@ip-172-31-24-235:~$ sudo systemctl reload apache2
ubuntu@ip-172-31-24-235:~$
```

Create an index.html file in that location so that we can test that the virtual host works as expected:

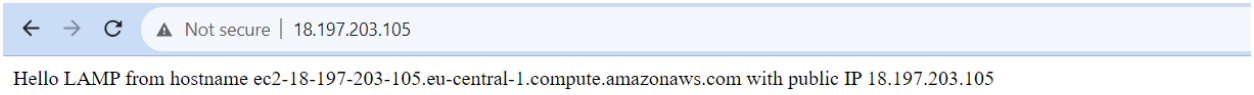
sudo echo 'Hello LAMP from hostname' $(curl -s http://169.254.169.254/latest/meta-data/public-hostname) 'with public IP' $(curl -s http://169.254.169.254/latest/meta-data/public-ipv4) > /var/www/projectlamp/index.html

Then use the command: sudo ls -l /var/www/html/ to confirm the index file has been created

```
ubuntu@ip-172-31-24-235:~$ sudo echo 'Hello LAMP from hostname' $(curl -s http://169.254.169.254/latest/meta-d
ata/public-hostname) 'with public IP' $(curl -s http://169.254.169.254/latest/meta-data/public-ipv4) > /var/ww
w/projectlamp/index.html
ubuntu@ip-172-31-24-235:~$ sudo ls -l /var/www/html/
total 12
-rw-r--r-- 1 root root 10671 Aug 11 00:41 index.html
ubuntu@ip-172-31-24-235:~$
```

Go to your browser and try to open your website URL using IP address:

http://<Public-IP-Address>:80



Step 5 — enable PHP on the website.

With the default DirectoryIndex settings on Apache, a file named index.html will always take precedence over an index.php file. This is useful for setting up maintenance pages in PHP applications, by creating a temporary index.html file containing an informative message to visitors. Because this page will take precedence over the index.php page, it will then become the landing page for the application. Once maintenance is over, the index.html is renamed or removed from the document root, bringing back the regular application page.
In case you want to change this behavior, you'll need to edit the /etc/apache2/mods-enabled/dir.conf file and change the order in which the index.php file is listed within the DirectoryIndex directive:

sudo vim /etc/apache2/mods-enabled/dir.conf

```
<IfModule mod_dir.c>
    #Change this:
    #DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
    #To this:
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

After saving and closing the file, you will need to reload Apache so the changes take effect:

sudo systemctl reload apache2

Finally, we will create a PHP script to test that PHP is correctly installed and configured on your server.

Create a new file named index.php inside your custom web root folder:

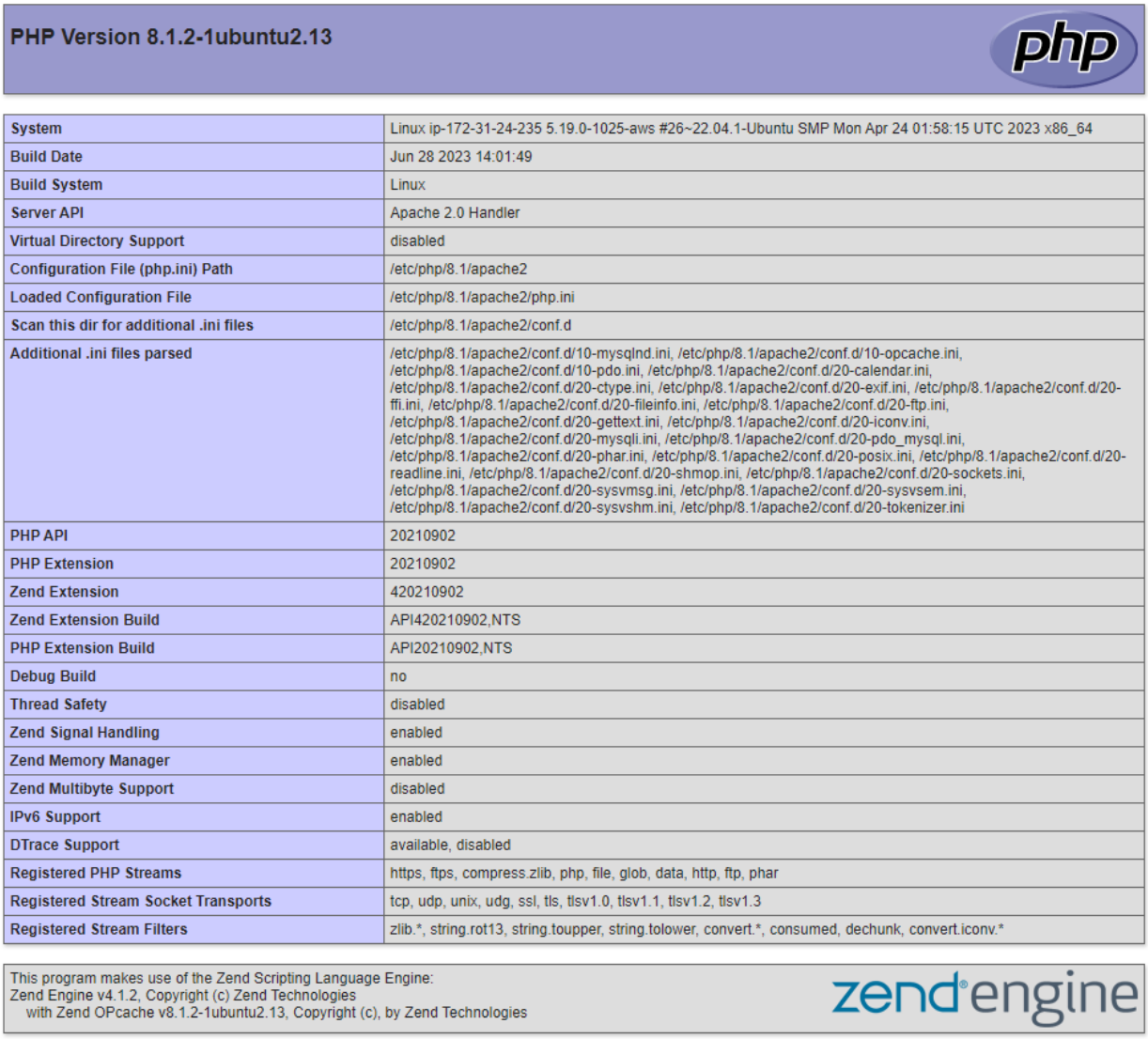
vim /var/www/projectlamp/index.php

This will open a blank file. Add the following text, which is valid PHP code, inside the file:

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

```
ubuntu@ip-172-31-24-235:~$ sudo vim /etc/apache2/mods-enabled/dir.conf
ubuntu@ip-172-31-24-235:~$ sudo systemctl reload apache2
ubuntu@ip-172-31-24-235:~$ vim /var/www/projectlamp/index.php
ubuntu@ip-172-31-24-235:~$
```

When you are finished, save and close the file, refresh the page and you will see a page similar to this shown below:



PHP Version 8.1.2-1ubuntu2.13

System	Linux ip-172-31-24-235 5.19.0-1025-aws #26~22.04.1-Ubuntu SMP Mon Apr 24 01:58:15 UTC 2023 x86_64
Build Date	Jun 28 2023 14:01:49
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-mysqlnd.ini, /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-mysqli.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.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
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, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.1.2, Copyright (c) Zend Technologies
with Zend OPcache v8.1.2-1ubuntu2.13, Copyright (c), by Zend Technologies

Configuration

After checking the relevant information about your PHP server through that page, it's best to remove the file you created as it contains sensitive information about your PHP environment -and your Ubuntu server. You can use rm to do so:

sudo rm /var/www/projectlamp/index.php