

# WEB STACK IMPLEMENTATION (LAMP STACK) IN AWS

## PROJECT 1

Login to AWS .amazon.com

Click on EC 2

Click on Instances (AMI)

Free tier eligible (t2.micro)

Review and Launch

Click on Launch

Create a new key pair and Download

Set up Security

Connect

Open terminal

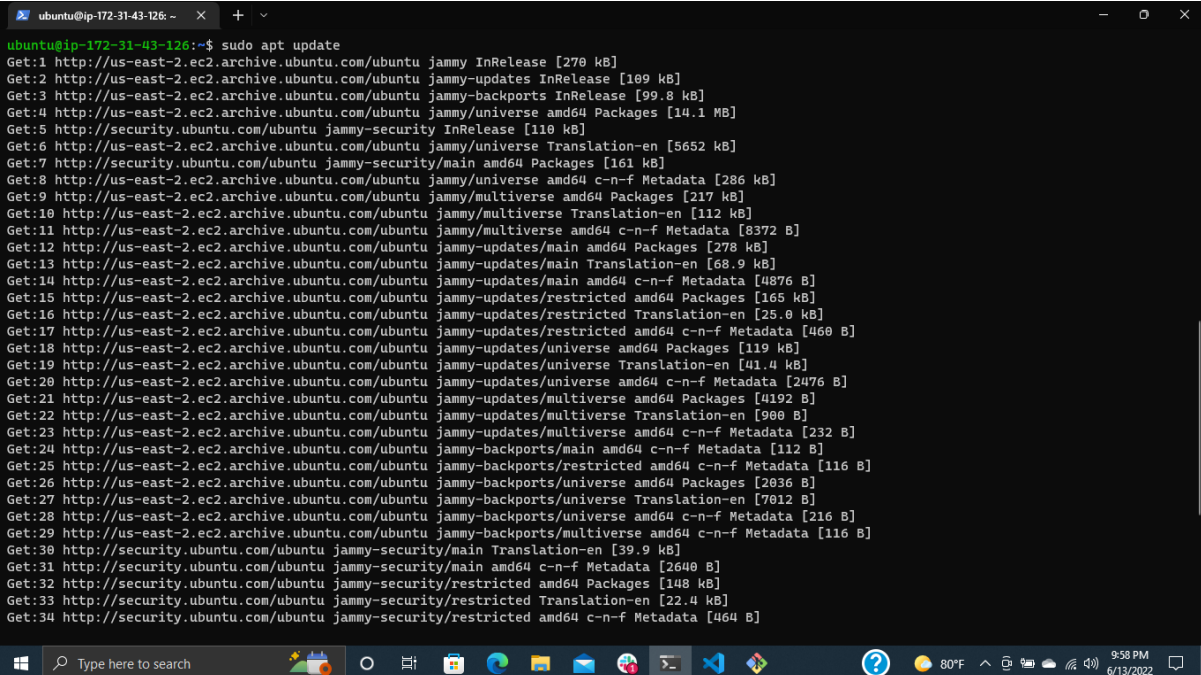
Type cd Downloads and press enter

Copy the ssh example on connecting the Instance and paste on the Terminal command.

### Step 1: Install Apache

Type: **sudo apt update**

This will update the package list for upgrades and new packages.



```
ubuntu@ip-172-31-43-126: ~$ sudo apt update
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [161 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [278 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [68.9 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [4876 B]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [165 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [25.0 kB]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [460 B]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [119 kB]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [41.4 kB]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [2476 B]
Get:21 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [4192 B]
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [900 B]
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [232 B]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [112 B]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [2836 B]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [7012 B]
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [216 B]
Get:29 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [39.9 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [2640 B]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [148 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [22.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [464 B]
```

Install Apache2 by typing this command on the Terminal: **sudo apt install apache2**

```
ubuntu@ip-172-31-43-126: ~$ sudo apt update
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [161 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [278 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [68.9 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [4876 B]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [165 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [25.0 kB]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [460 B]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [119 kB]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [41.4 kB]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [2476 B]
Get:21 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [4192 B]
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [900 B]
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [232 B]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [112 B]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [2836 B]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [7012 B]
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [216 B]
Get:29 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [39.9 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [2640 B]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [148 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [22.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [464 B]
```

```
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.1-4build1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-43-126:~$
```

Verify Apache2 is running as a service in the OS by typing: **sudo systemctl status apache2**  
Then press Enter.

```
ubuntu@ip-172-31-43-126: ~$ sudo systemctl status apache2
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.1-0ubuntu1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

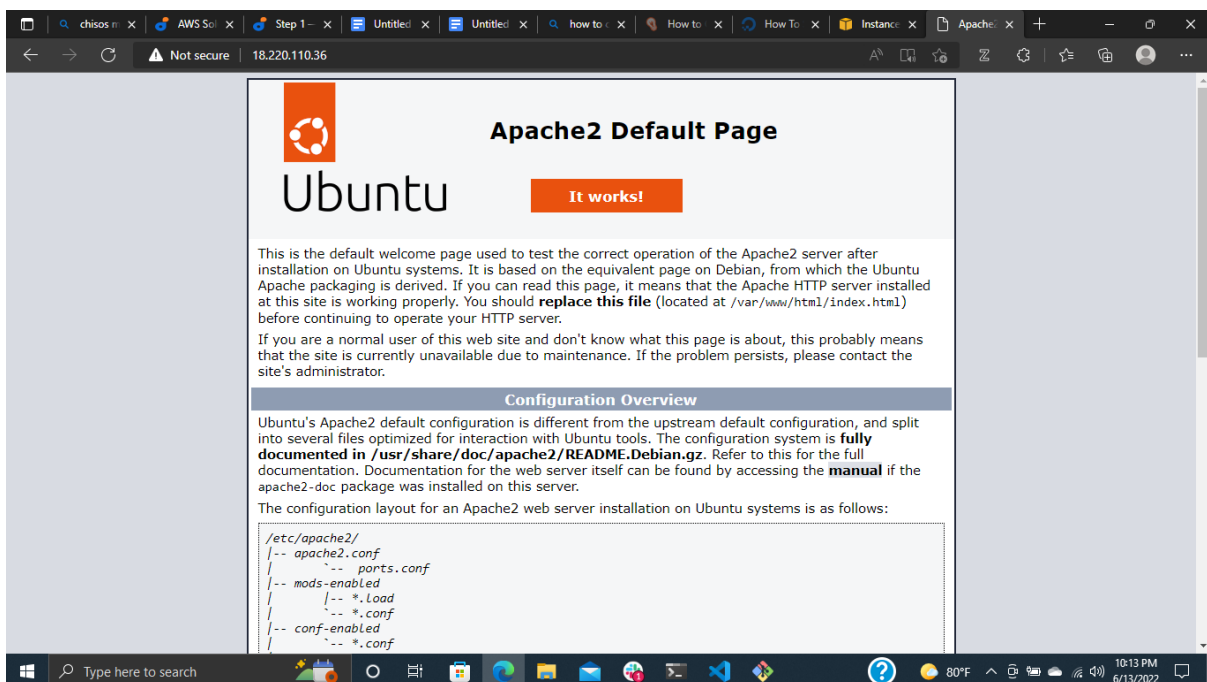
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-43-126:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-06-14 03:03:58 UTC; 3min 23s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2161 (apache2)
     Tasks: 55 (limit: 1146)
    Memory: 4.9M
       CPU: 39ms
   CGroup: /system.slice/apache2.service
           └─2161 /usr/sbin/apache2 -k start
             └─2163 /usr/sbin/apache2 -k start
               └─2164 /usr/sbin/apache2 -k start

Jun 14 03:03:57 ip-172-31-43-126 systemd[1]: Starting The Apache HTTP Server...
Jun 14 03:03:58 ip-172-31-43-126 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-43-126:~$
```

Copy the IPv4 Public IP from AWS EC2 and paste on your browser to open the Apache2 welcome page.



## Step 2: Installing MySQL

Type: `sudo apt install mysql-server`

```
ubuntu@ip-172-31-43-126: ~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl
  libmecab2 libprotobuf-lite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyc
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl
  libmecab2 libprotobuf-lite23 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 55 not upgraded.
Need to get 28.6 MB of archives.
After this operation, 240 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Type: **Y**

And press Enter.

Login to your MySQL console by typing: **sudo mysql** and then press enter.

Type: **ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'PassWord.1'**

Then exit MySQL shell by typing: **exit**

Press Enter

Note that after MySQL has been installed, you will need to set the root password of the database and secure it using the following command:

### **Sudo mysql\_secure\_installation**

This will take you to a command line for you to enter your password.

Enter the initial password: **PassWord.1**

Answer the proceeding questions with either a Y or N. But don't change the password.

```
ubuntu@ip-172-31-38-150: ~  
You should remove them before moving into a production environment.  
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.  
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-172-31-38-150:~$
```

## STEP 3: INSTALLING PHP

Copy and run this command: `sudo apt install php libapache2-mod-php php-mysql`

Enter

Type: **Y** and press Enter

```
ubuntu@ip-172-31-38-150: ~  
Creating config file /etc/php/7.4/mods-available/readline.ini with new version  
Setting up php7.4-opcache (7.4.3-4ubuntu2.10) ...  
Creating config file /etc/php/7.4/mods-available/opcache.ini with new version  
Setting up php7.4-json (7.4.3-4ubuntu2.10) ...  
Creating config file /etc/php/7.4/mods-available/json.ini with new version  
Setting up php-mysql (2:7.4+75) ...  
Setting up php7.4-cli (7.4.3-4ubuntu2.10) ...  
update-alternatives: using /usr/bin/php7.4 to provide /usr/bin/php (php) in auto mode  
update-alternatives: using /usr/bin/phar7.4 to provide /usr/bin/phar (phar) in auto mode  
update-alternatives: using /usr/bin/phar.phar7.4 to provide /usr/bin/phar.phar (phar.phar) in auto mode  
Creating config file /etc/php/7.4/cli/php.ini with new version  
Setting up libapache2-mod-php7.4 (7.4.3-4ubuntu2.10) ...  
Creating config file /etc/php/7.4/apache2/php.ini with new version  
Module mpm_event disabled.  
Enabling module mpm_prefork.  
apache2_switch_mpm Switch to prefork  
apache2_invoke: Enable module php7.4  
Setting up php7.4 (7.4.3-4ubuntu2.10) ...  
Setting up libapache2-mod-php (2:7.4+75) ...  
Setting up php (2:7.4+75) ...  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.10) ...  
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.10) ...  
ubuntu@ip-172-31-38-150:~$ ^C  
ubuntu@ip-172-31-38-150:~$
```

Run this command to confirm PHP Installation: `php -v`

The command will confirm PHP installation.

## STEP 4: CREATE A VIRTUAL HOST FOR THE WEBSITE USING APACHE

We will make use of PROJECTLAMP.

We'll create a directory by using the Make Directory command below:

```
sudo mkdir /var/www/projectlamp
```

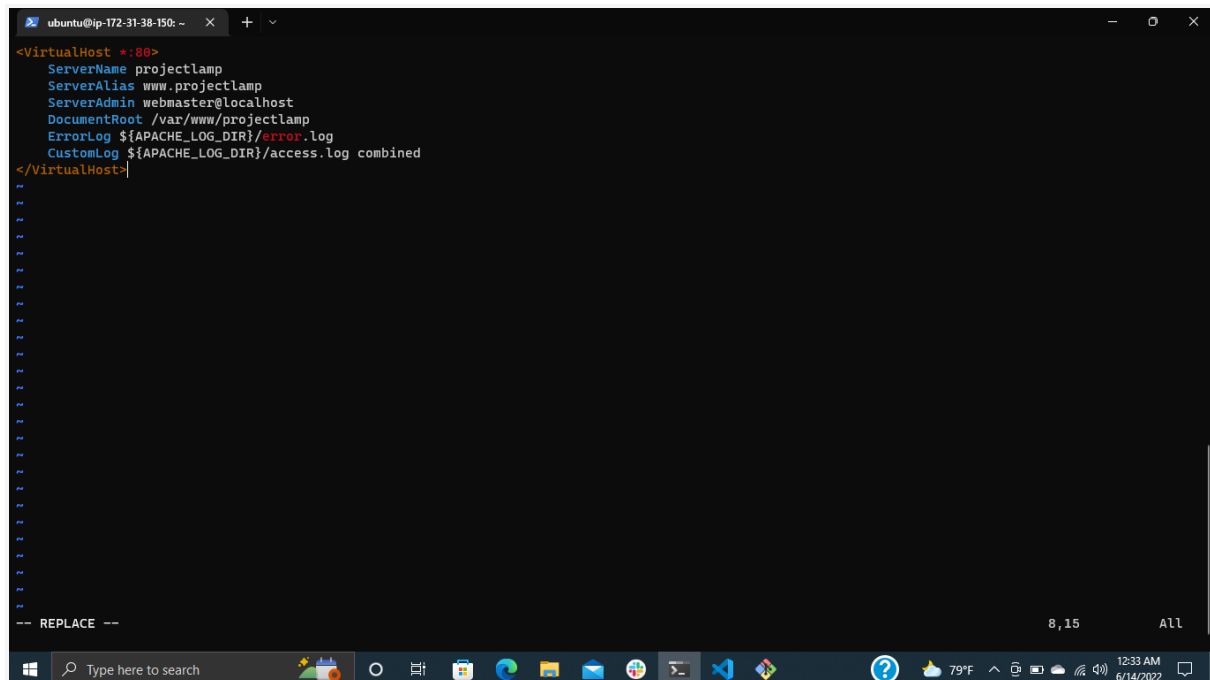
We're going to assign ownership to Projectlamp by typing the command below:

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

We will then use the vi editor command to create our Project lamp file by typing the command below:

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

To type into the vi Editor, we have to type the letter i and press enter. This will take us to INSERT. You can now type or insert into this page.

A screenshot of a terminal window titled 'ubuntu@ip-172-31-38-150: ~'. The terminal shows the configuration of a virtual host file in the vi editor. The content of the file is as follows:

```
<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>
```

The terminal also shows the vi editor's status bar at the bottom, indicating the current mode is REPLACE and the cursor is at line 8, column 15.

Now, use the ls command to show that it's a directory.

```
sudo ls /etc/apache2/sites-available
```

We'll then use the next two commands to enable our just created virtual host and disable the default domain by inputting the following commands:

```
sudo a2ensite projectlamp
```

```
sudo a2dissite 000-default
```

Now run the below command to confirm the validity of our file.

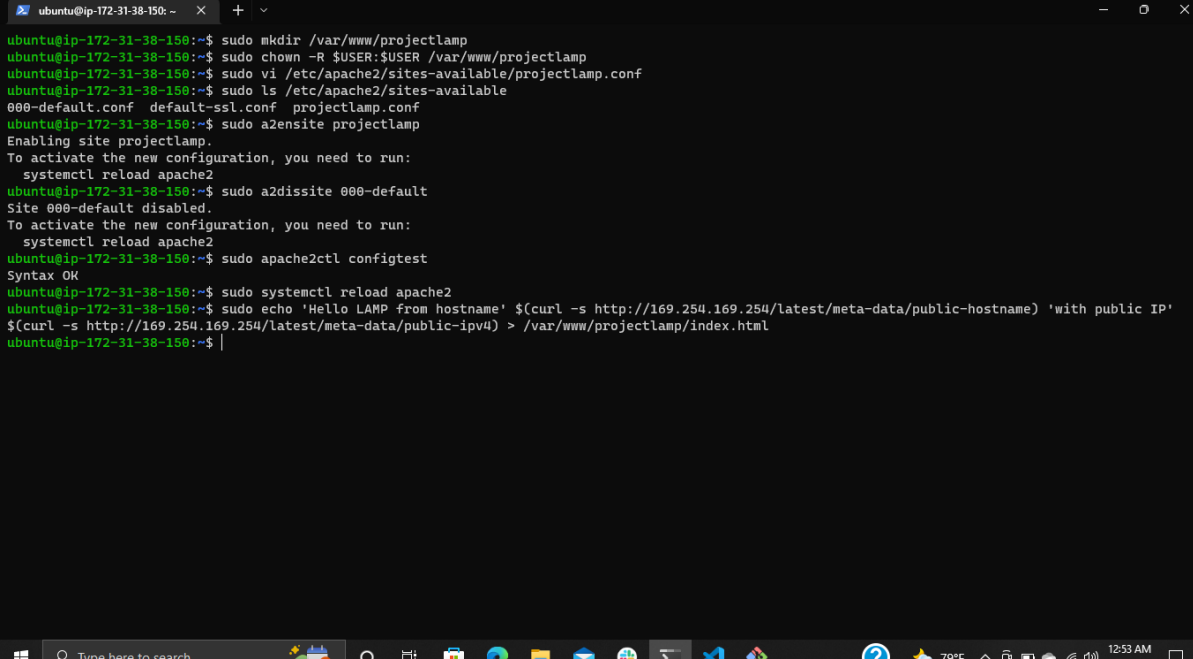
```
sudo apache2ctl configtest
```

Enter the next command to reload Apache2

**sudo systemctl reload apache2**

Now we have to create an index.html file by inserting the following command:

```
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
```

A terminal window screenshot showing the process of setting up ProjectLamp on Ubuntu. The user runs several commands: creating the directory /var/www/projectlamp, setting permissions, editing the site configuration file, enabling the site, disabling the default site, testing the configuration, and finally reloading Apache2. The terminal output shows the success of these commands and the creation of the index.html file.

```
ubuntu@ip-172-31-38-150: ~$ sudo mkdir /var/www/projectlamp
ubuntu@ip-172-31-38-150: ~$ sudo chown -R $USER:$USER /var/www/projectlamp
ubuntu@ip-172-31-38-150: ~$ sudo vi /etc/apache2/sites-available/projectlamp.conf
ubuntu@ip-172-31-38-150: ~$ sudo ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf  projectlamp.conf
ubuntu@ip-172-31-38-150: ~$ sudo a2ensite projectlamp
Enabling site projectlamp.
To activate the new configuration, you need to run:
    systemctl reload apache2
ubuntu@ip-172-31-38-150: ~$ sudo a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
ubuntu@ip-172-31-38-150: ~$ sudo apache2ctl configtest
Syntax OK
ubuntu@ip-172-31-38-150: ~$ sudo systemctl reload apache2
ubuntu@ip-172-31-38-150: ~$ sudo echo 'Hello LAMP from hostname' $(curl -s http://169.254.169.254/latest/meta-data/public-hostname) 'with public IP'
$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4) > /var/www/projectlamp/index.html
ubuntu@ip-172-31-38-150: ~$
```

## STEP 5: NOW WE'VE TO ENABLE PHP ON THE WEBSITE

We have to edit the Apache website and change the way it appears on our directory by inserting the following command:

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





Then insert the text:

```
<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>
```

Reload Apache for the changes to take effect. Type this command:

```
sudo systemctl reload apache2
```

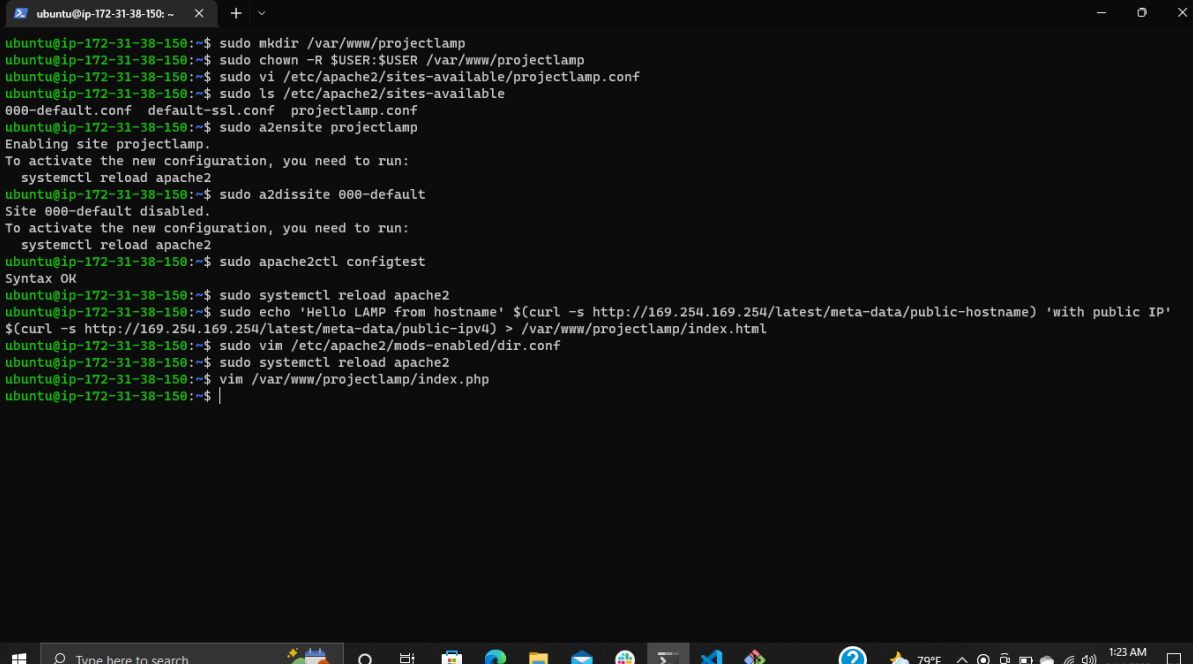
Create index.php file inside the web root folder by typing the command below:

```
vim /var/www/projectlamp/index.php
```

Enter the empty file text:

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

Click on colon sign and type wq and press enter.



```
ubuntu@ip-172-31-38-150: ~$ sudo mkdir /var/www/projectlamp
ubuntu@ip-172-31-38-150: ~$ sudo chown -R $USER:$USER /var/www/projectlamp
ubuntu@ip-172-31-38-150: ~$ sudo vi /etc/apache2/sites-available/projectlamp.conf
ubuntu@ip-172-31-38-150: ~$ sudo ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf  projectlamp.conf
ubuntu@ip-172-31-38-150: ~$ sudo a2ensite projectlamp
Enabling site projectlamp.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-38-150: ~$ sudo a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-38-150: ~$ sudo apache2ctl configtest
Syntax OK
ubuntu@ip-172-31-38-150: ~$ sudo systemctl reload apache2
ubuntu@ip-172-31-38-150: ~$ sudo echo 'Hello LAMP from hostname' $(curl -s http://169.254.169.254/latest/meta-data/public-hostname) 'with public IP'
$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4) > /var/www/projectlamp/index.html
ubuntu@ip-172-31-38-150: ~$ sudo vim /etc/apache2/mods-enabled/dir.conf
ubuntu@ip-172-31-38-150: ~$ sudo systemctl reload apache2
ubuntu@ip-172-31-38-150: ~$ vim /var/www/projectlamp/index.php
ubuntu@ip-172-31-38-150: ~$
```

Now, refresh the page by entering the initial IPv4 Public IP on your AWS EC2 Instance to refresh the page.



