PROJECT 1(LAMP STACK)

Aim: To build a web stack used to develop a software product using Lamp stack. LAMP which stands for Linux, Apache, MysQl and PHP. Each component contributes essential capabilities to the stack.

-The LAMP stack is a popular software bundle using open source components to build and deliver web applications
-With open-source tools for an operating system (Linux), web server (Apache), database server (MySQL), and programming language (PHP), the LAMP stack is an efficient and flexible method that enables competition with commercial software developers.

The Web Server: Apache HTTP server

The open-source Apache web server manages the traffic a website or web application receives. Like other web servers, Apache processes online client requests and communicates with an internal database (MySQL) and transmit information through internet using HTTP

The Database Server: MySQL

The open-source relational database management system (RDBMS) MySQL, is the most popular free option for storing application data. Administrators can seamlessly query data with the SQL language while managing information for client use.

The Programming Language: PHP

The Hypertext Preprocessor, more commonly known as PHP, is a general-purpose scripting language used for creating dynamic web content. Because HTML is a static processor, inserting PHP scripting into pages enables dynamic web applications, tools, and processes. In addition to PHP, Perl and Python are also widely used for the LAMP stack.
When a visitor opens the webpage, the server processes the PHP commands and sends the results to the visitor's browser.

HOW DOES LAMP STACK WORK?

1. Clients navigate to a website and make a request on their web browser.
2. If the client request is for a PHP file, Apache uses PHP to execute the request.
3. With other data requests, the Apache fetches MySQL data.
4. The Apache web server receives granted resources from PHP and MySQL.
5. Apache processes the resource to present HTML content to the client.

Software Development Life Cycle(SDLC)- defines the complete cycle of development from planning to creating to testing to deploying a software product.

CHMOD (Change mode)- A command in Linux/Unix used to change access permission of a file system. (Directory files, regular files, etc) using either a numeric or symbolic code

CHOWN(Change Ownership)-A command used in linux/Unix operating systems to change access ownership of a file system.

Both TCP and UDP are transport layer protocols, TCP ensures data can be transmitted between server and client with reliability while UDP is a connection less protocol without guarantee if a message would be delivered or not.

Ports commonly used in web are listed as follows: http-port 80, https-port 443, ftp, ssh.

Preparatory Steps

A. I created AWS account using aws.amazon.com and chose ubuntu 20.0 64 bits as server

B. Connected to my EC2 instance by launching

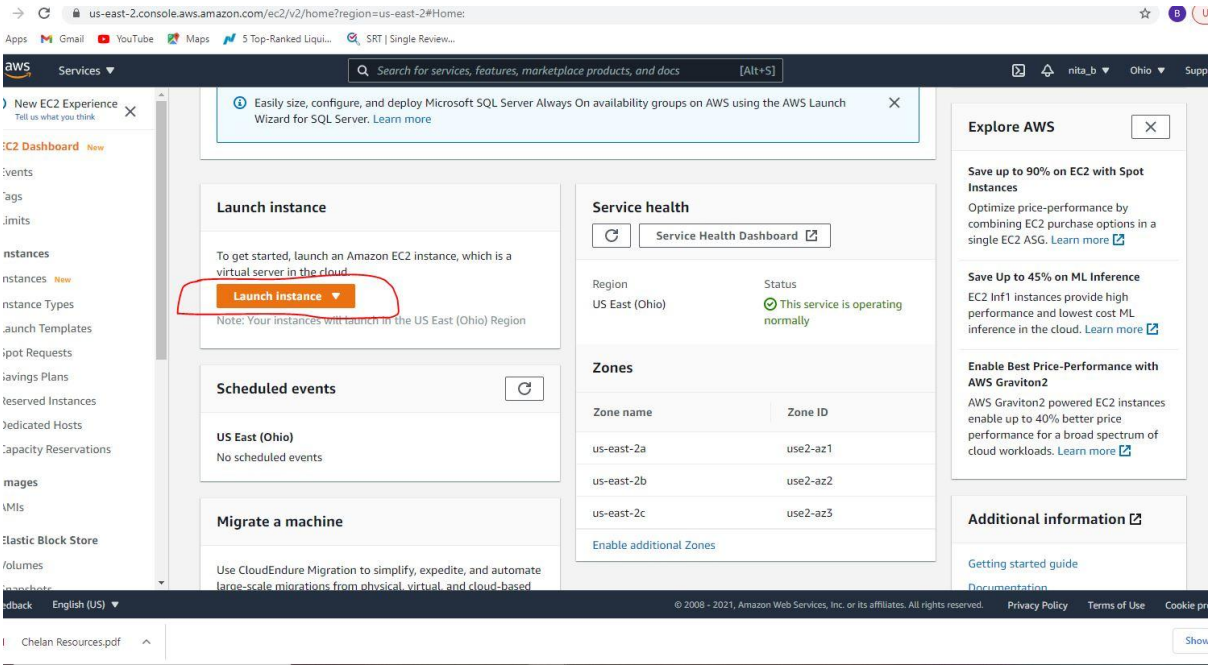
Congratulations!

Thank you for signing up with AWS.

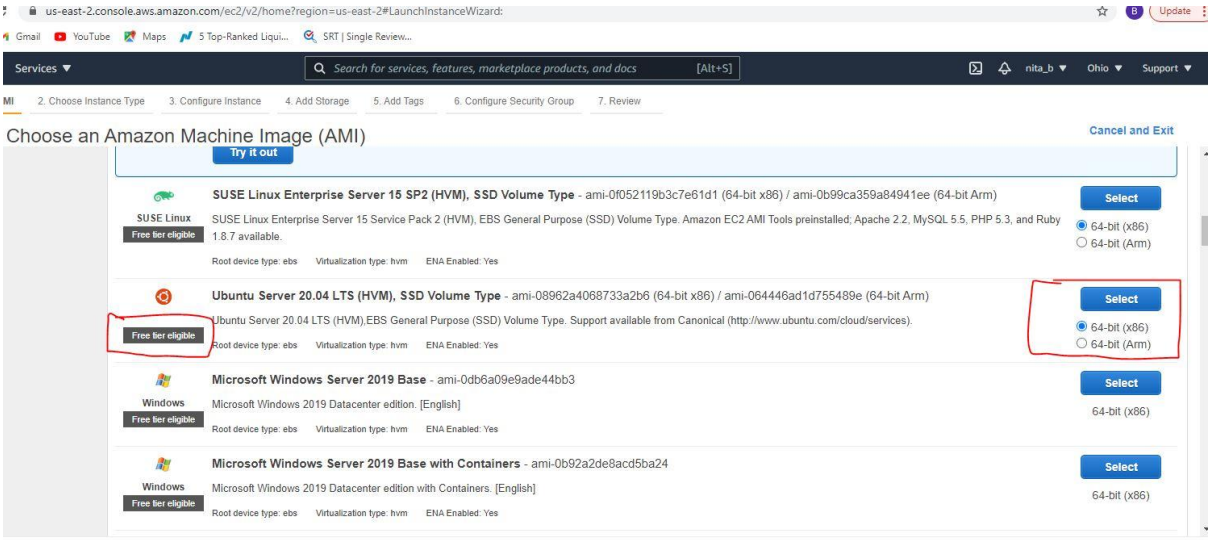
We are activating your account, which should take a few minutes. You will receive an email when this is complete.

Go to the AWS Management Console

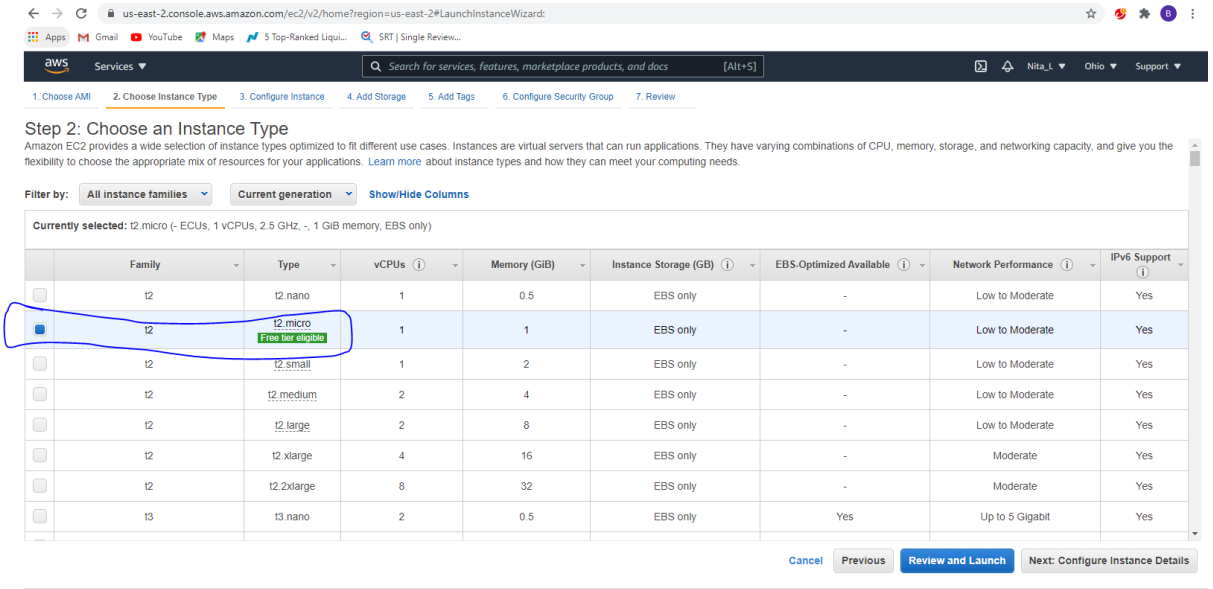
Sign up for another account or Contact Sales



C. I chose 64 bits as my system size is 64 bits operating system

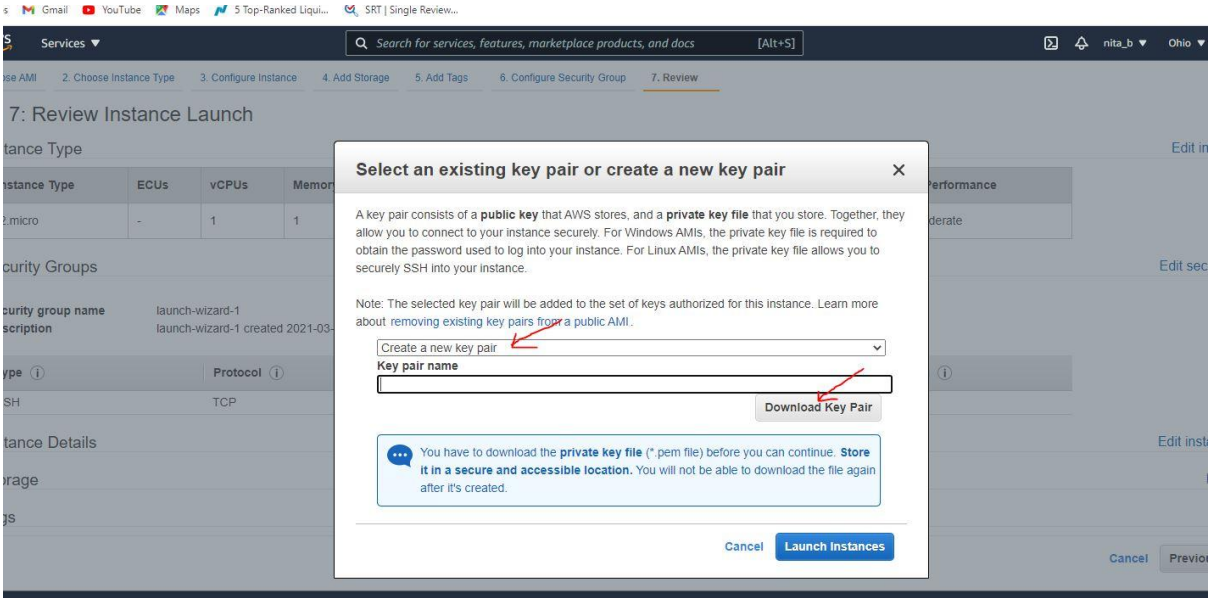


D. I chose t2 micro free tier

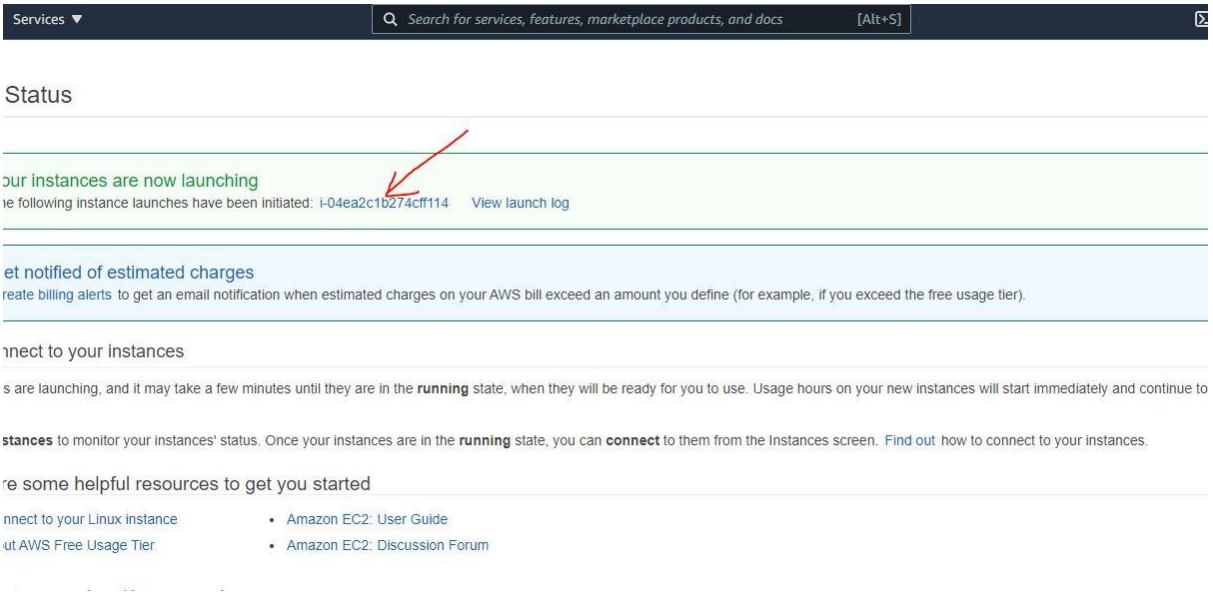


C. You can either choose a new pair key and download or an existing one i chose a new one and saved securely as this is my PEM file, i chose a new security group and

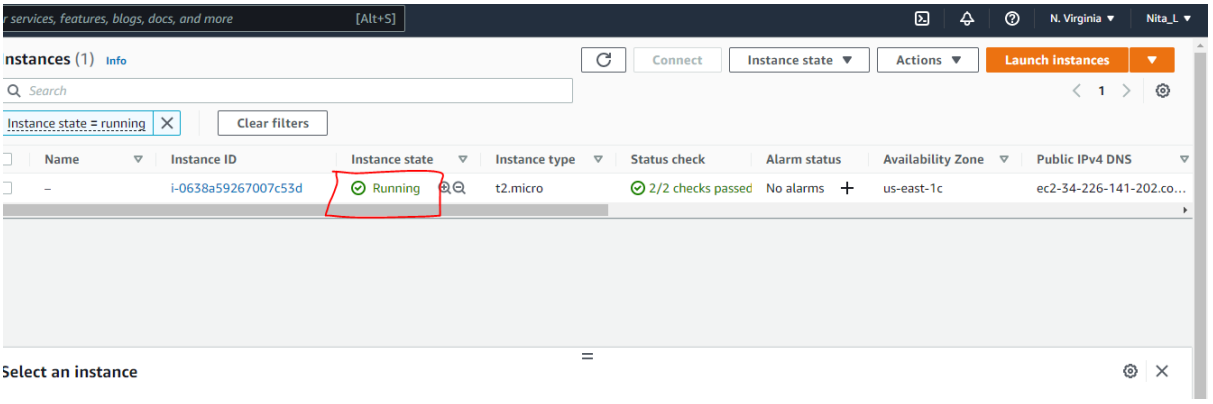
clicked next all the way till the end and launched instance



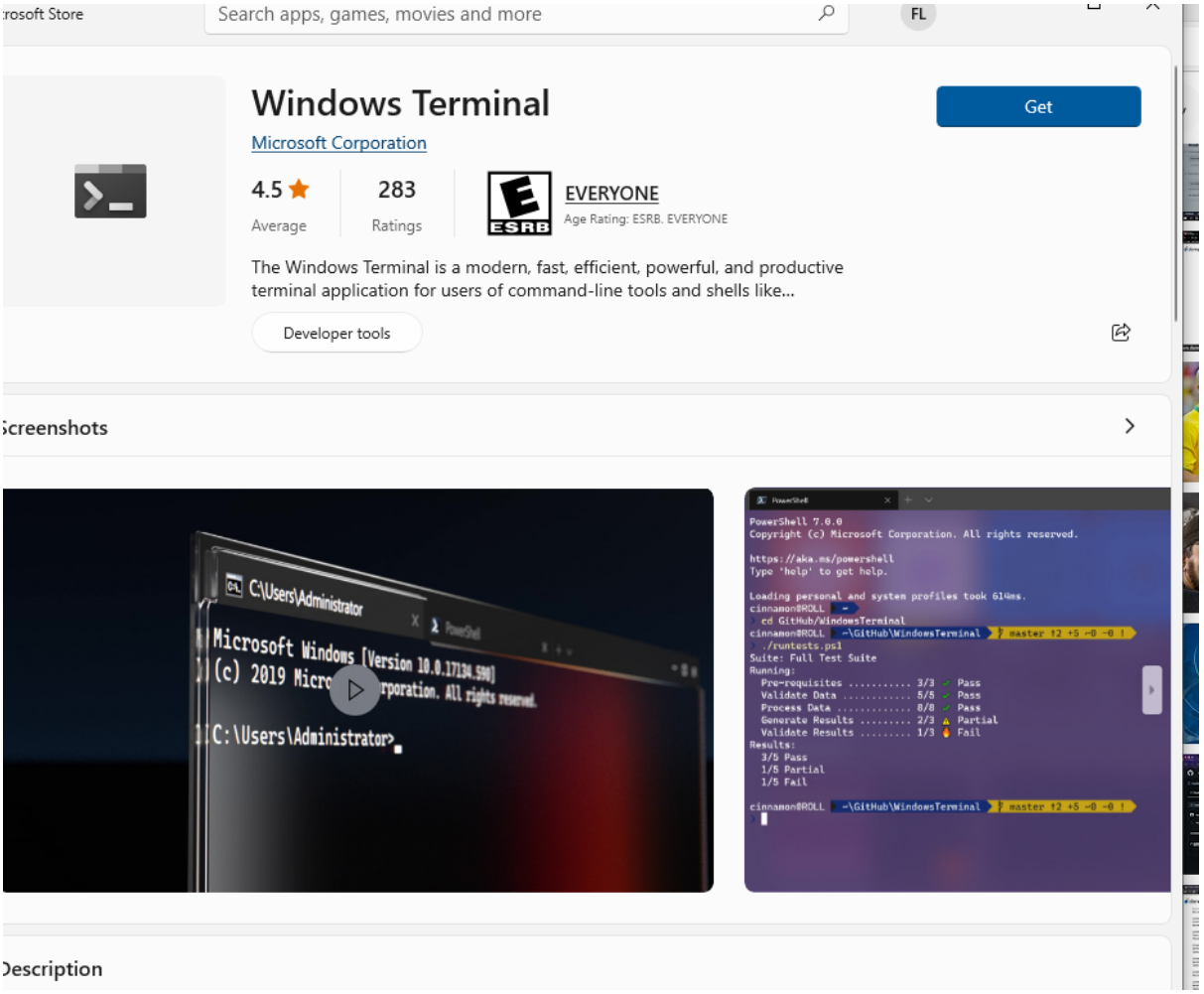
E. Instance is launching now



F. Instance is running now

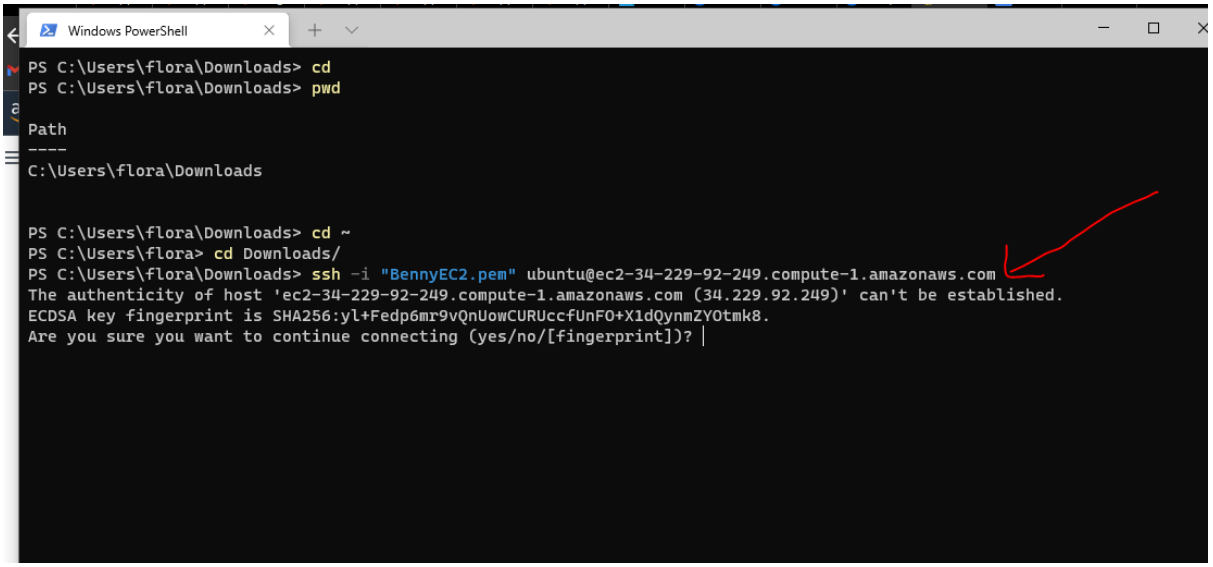


G: i downloaded windows terminal from microsoft store



H: checking to see if my private key file is in downloads using **cd Downloads/** and connect to ssh using **ssh -i "BennyEC2.pem" ubuntu@ec2-34-229-92-249.compute-1.amazonaws.com**

Note: i got an error here and tried to rerun the command again



I: I tried to connect but without changing directory to where my PEM file is and it was denied from the red arrow below

```
Path
PS C:\Users\flora> ssh -i "BennyEC2.pem" ubuntu@ec2-34-229-92-249.compute-1.amazonaws.com
Warning: Identity file BennyEC2.pem not accessible: No such file or directory.
ubuntu@ec2-34-229-92-249.compute-1.amazonaws.com: Permission denied (publickey).
PS C:\Users\flora> cd Downloads/
PS C:\Users\flora\Downloads> ssh -i "BennyEC2.pem" ubuntu@ec2-34-229-92-249.compute-1.amazonaws.com
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Mar 22 18:20:24 UTC 2022

  System load:  0.0               Processes:             99
  Usage of /:   18.4% of 7.69GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 172.31.22.44
  Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
```

STEP 1: INSTALLING APACHE AND UPDATING FIREWALL

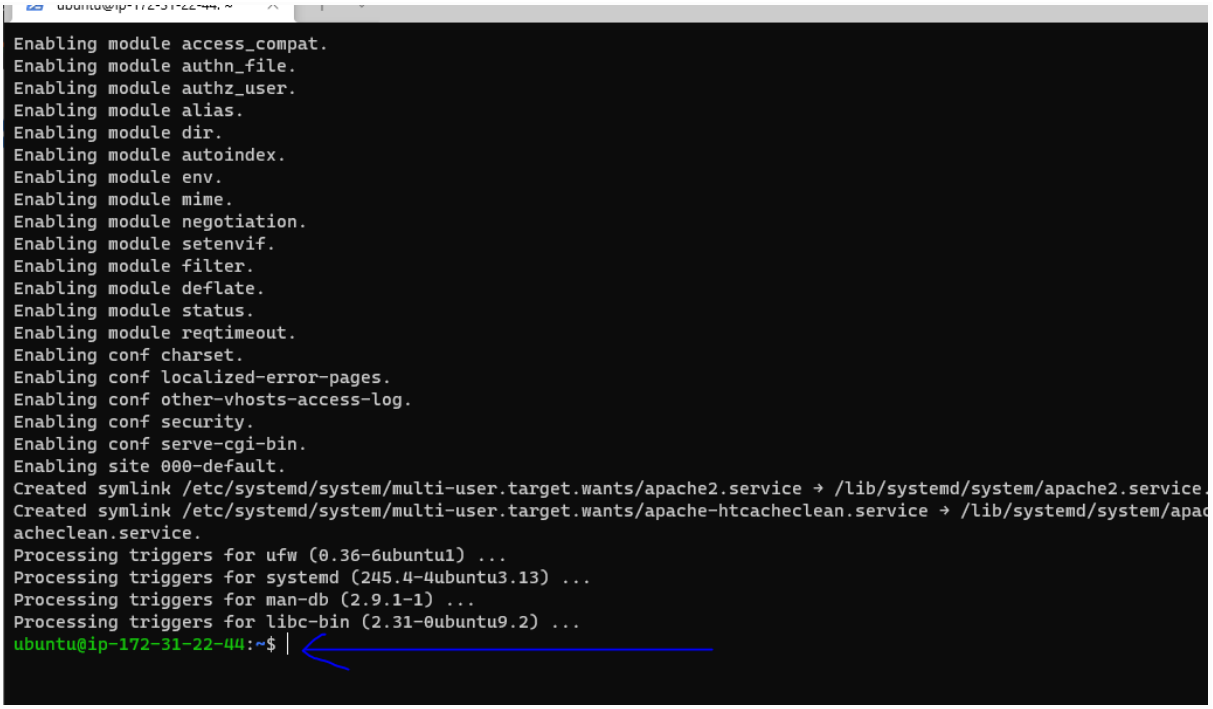
Apache HTTP Server is the most widely used web server software, Its purpose is to create a connection between a server and the browsers of website visitors (Firefox, Google Chrome, Safari, etc.) while delivering files between them.

Steps:

A- Update app using **sudo apt update**

```
ubuntu@ip-172-31-22-44: ~
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [20.3 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.5 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [7336 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [592 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [42.2 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [10.1 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [864 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [22.7 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [15.4 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [804 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1342 kB]
Get:32 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [233 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [9800 B]
Get:34 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [825 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [117 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [532 B]
Get:37 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [692 kB]
Get:38 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [121 kB]
Get:39 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [14.1 kB]
Get:40 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [20.7 kB]
Get:41 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5196 B]
Get:42 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [500 B]
Fetched 22.3 MB in 4s (5697 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
89 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-22-44:~$
```

B: Install Apache using **sudo apt install apache2**

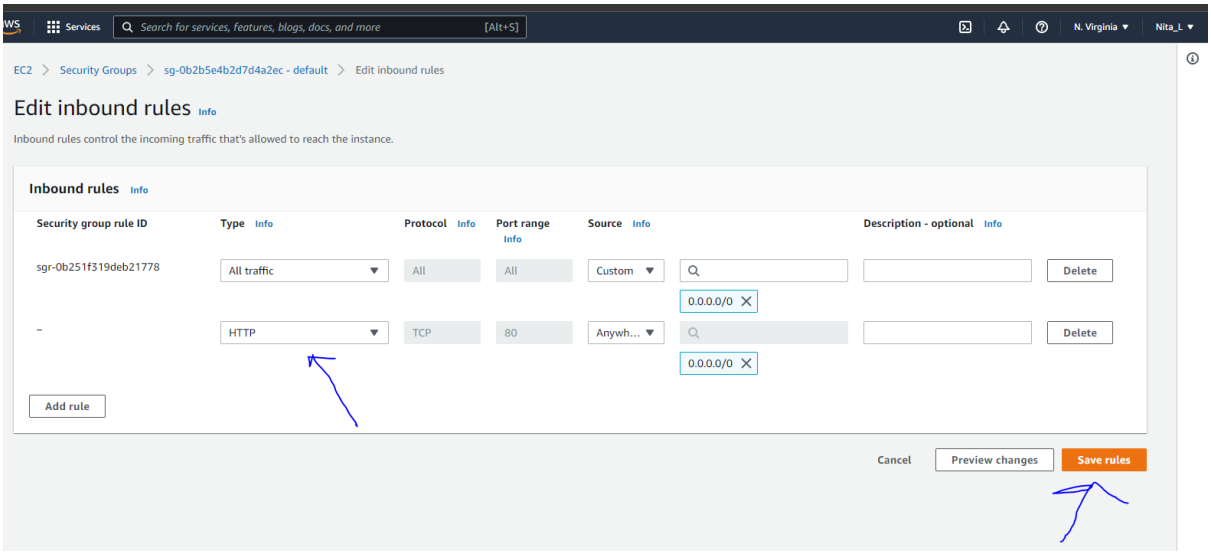


C: To verify that apache2 is running **sudo systemctl status apache2**

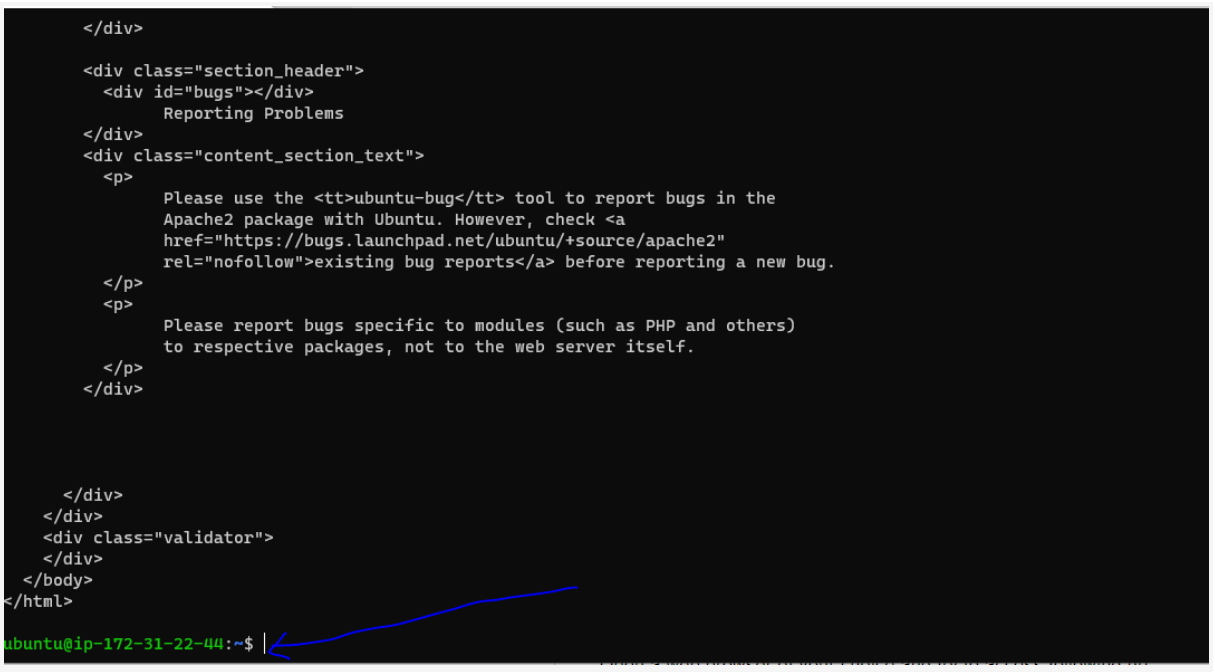
D: If it is green and running, then you did everything correctly and yes it **GREEN**.

```
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
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/sy
acheclean.service.
Processing triggers for ufw (0.36-6ubuntu1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
ubuntu@ip-172-31-22-44:~$ 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 2022-03-22 18:34:46 UTC; 3min 54s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2444 (apache2)
      Tasks: 55 (limit: 1147)
     Memory: 4.8M
     CGroup: /system.slice/apache2.service
             ├─2444 /usr/sbin/apache2 -k start
             ├─2447 /usr/sbin/apache2 -k start
             └─2448 /usr/sbin/apache2 -k start

Mar 22 18:34:46 ip-172-31-22-44 systemd[1]: Starting The Apache HTTP Server...
Mar 22 18:34:46 ip-172-31-22-44 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-22-44:~$
```

E:I edited the security group and added a new connection through port 80

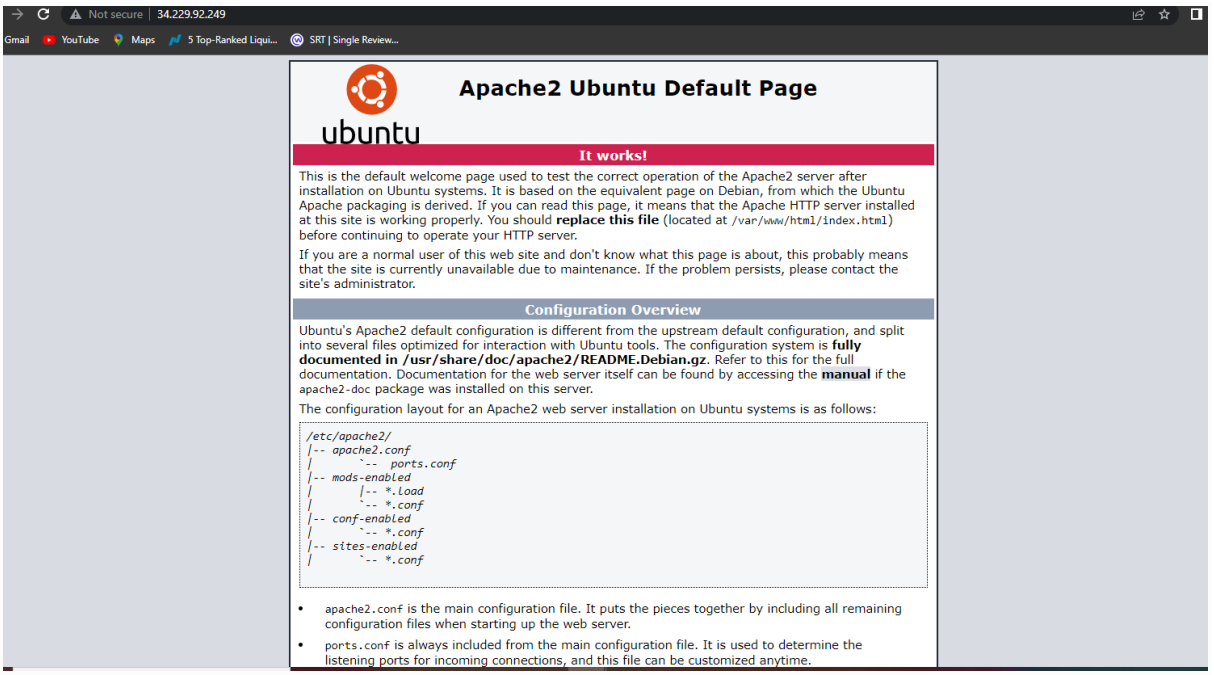


F: To access server locally in ubuntu shell can be done either by
Dns name using
curl http://localhost:80 or by
IP address **using curl http://127.0.0.1:80** but i used IP address



G: To test if Apache HTTP server is responding to request from internet using **http://<Public-IP-Address>:80 which is http://34.229.92.249:80 (public IP address used is 34.229.92.249)**

Side note: I added a new security rule and chose HTTP type and anywhere as source)



If you see the following page, then your web server is now correctly installed and accessible through your firewall.

STEP 2: Installing MySQL

Steps:

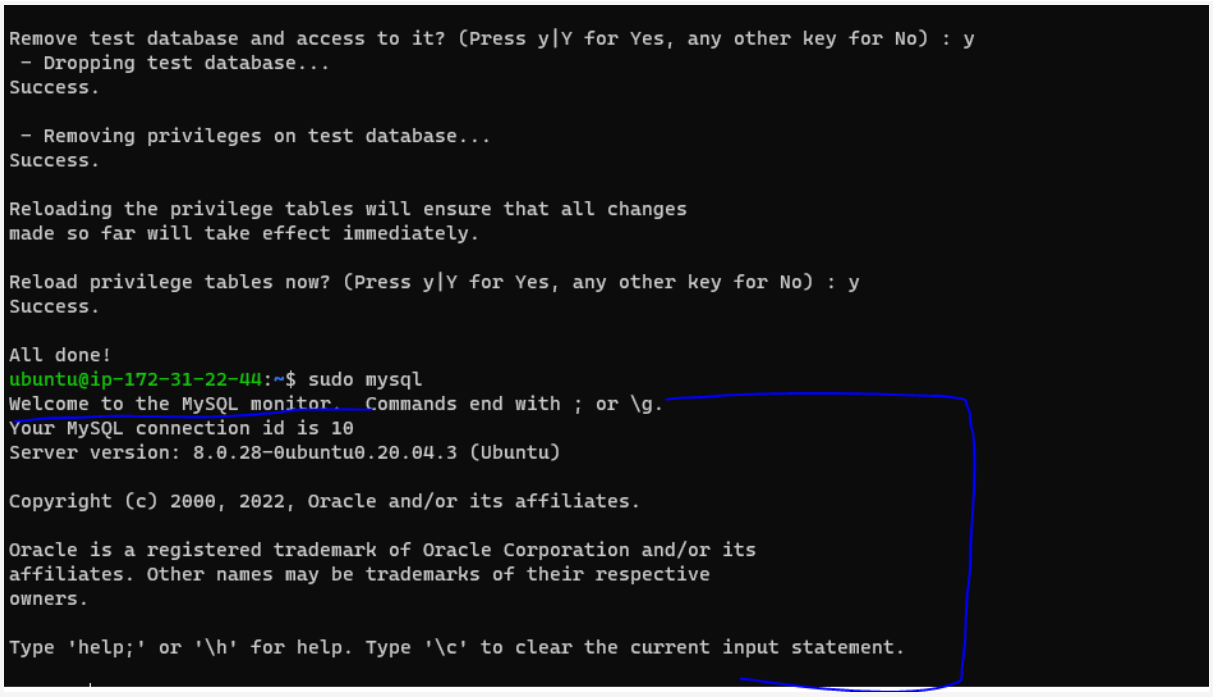
A: Install MySQL using **sudo apt install mysql-server**

```
reading /usr/share/mecab/dic/ipadic/Filler.csv ... 19
reading /usr/share/mecab/dic/ipadic/Auxil.csv ... 199
reading /usr/share/mecab/dic/ipadic/Others.csv ... 2
reading /usr/share/mecab/dic/ipadic/Adverb.csv ... 3032
reading /usr/share/mecab/dic/ipadic/Noun.proper.csv ... 27328
emitting double-array: 100% |###########################################|
reading /usr/share/mecab/dic/ipadic/matrix.def ... 1316x1316
emitting matrix      : 100% |###########################################|

done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/lib/mecab/dic/debian (mecab-dictionary)
o mode
Setting up libhtml-parser-perl (3.72-5) ...
Setting up libhttp-message-perl (6.22-1) ...
Setting up mysql-server-8.0 (8.0.28-0ubuntu0.20.04.3) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 14705
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up libcgi-pm-perl (4.46-1) ...
Setting up libhtml-template-perl (2.97-1) ...
Setting up mysql-server (8.0.28-0ubuntu0.20.04.3) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
ubuntu@ip-172-31-22-44:~$
```

B: Running a security script to remove some insecure default settings using **sudo mysql_secure_installation**

```
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-172-31-22-44:~$
```

C: To test if i can login to mysql using **sudo mysql**



STEP 3: InSTALLING PHP

Steps:

A: Install php using **sudo apt install php libapache2-mod-php php-mysql**

Side Note: In order for Apache to handle PHP files, we need **Php-mysql** and **libapache2-mod-php** install

```
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-172-31-22-44:~$
```

B: To confirm PHP version using
php -v

```
ubuntu@ip-172-31-22-44:~$ php -v
PHP 7.4.3 (cli) (built: Mar  2 2022 15:36:52) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-22-44:~$
```

STEP 4: Creating a Virtual Host for your Website using Apache in order to test PHP script

Steps:

A: Make a directory name “projectone” using
sudo mkdir /var/www/projectone

```
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-22-44:~$ sudo mkdir /var/www/projectone
ubuntu@ip-172-31-22-44:~$
```

B: Assigning ownership of the directory with your current system user
sudo chown -R $USER:$USER /var/www/projectone

```
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-172-31-22-44:~$ php -v
PHP 7.4.3 (cli) (built: Mar  2 2022 15:36:52) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-22-44:~$ sudo mkdir /var/www/projectone
ubuntu@ip-172-31-22-44:~$ sudo chown -R $USER:$USER /var/www/projectone
ubuntu@ip-172-31-22-44:~$
```

C: create and open a new configuration file in Apache's
Sites-available directory using Vi editor with the command below
sudo vi /etc/apache2/sites-available/projectone.conf

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

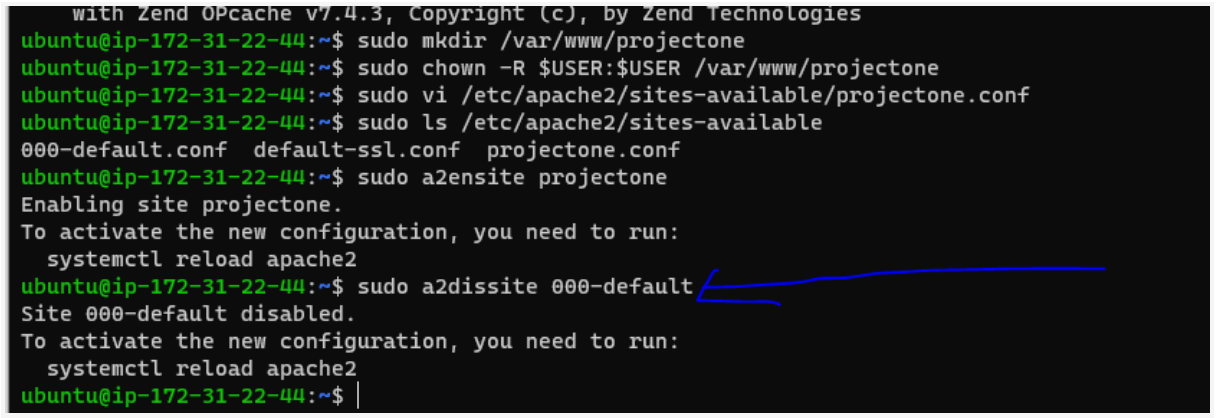
D: To check file saved on Vi using **sudo ls /etc/apache2/sites-available**

```
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-172-31-22-44:~$ php -v
PHP 7.4.3 (cli) (built: Mar  2 2022 15:36:52) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-22-44:~$ sudo mkdir /var/www/projectone
ubuntu@ip-172-31-22-44:~$ sudo chown -R $USER:$USER /var/www/projectone
ubuntu@ip-172-31-22-44:~$ sudo vi /etc/apache2/sites-available/projectone.conf
ubuntu@ip-172-31-22-44:~$ sudo ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf  projectone.conf
ubuntu@ip-172-31-22-44:~$
```

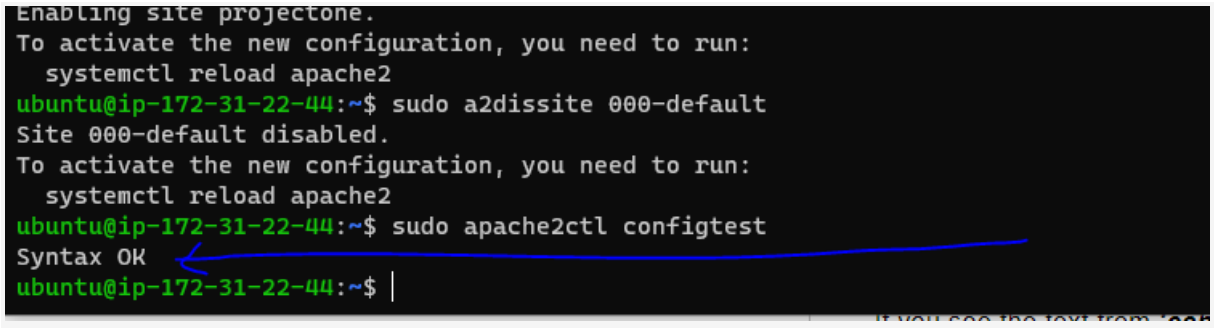
E: To enable the new virtual host using **sudo a2ensite projectone**

```
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
ubuntu@ip-172-31-22-44:~$ sudo mkdir /var/www/projectone
ubuntu@ip-172-31-22-44:~$ sudo chown -R $USER:$USER /var/www/projectone
ubuntu@ip-172-31-22-44:~$ sudo vi /etc/apache2/sites-available/projectone.con
ubuntu@ip-172-31-22-44:~$ sudo ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf  projectone.conf
ubuntu@ip-172-31-22-44:~$ sudo a2ensite projectone
Enabling site projectone.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-22-44:~$
```

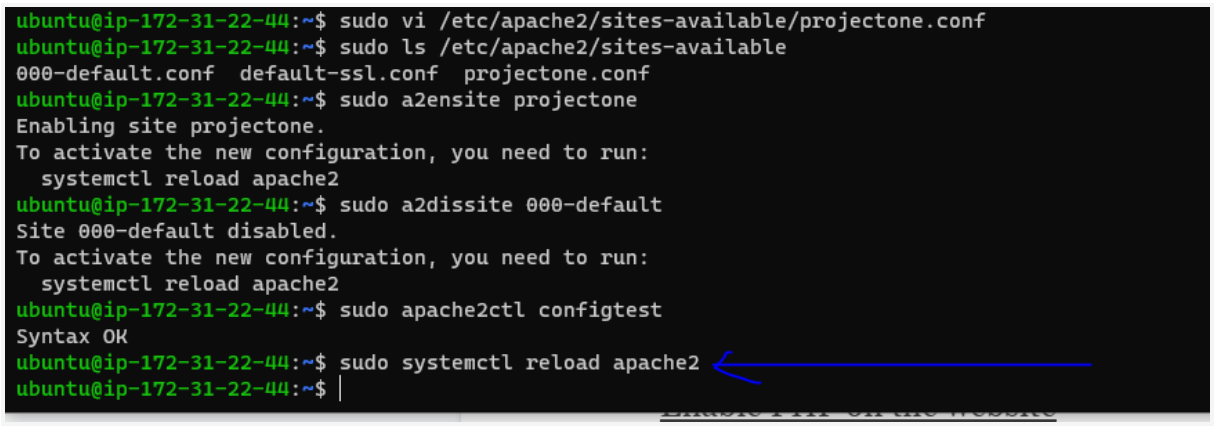
F: To disable Apache's default website since i did not use a custom domain name using **sudo a2dissite 000-default**



G: To make sure your configuration file doesn't contain syntax errors Using **sudo apache2ctl configtest**

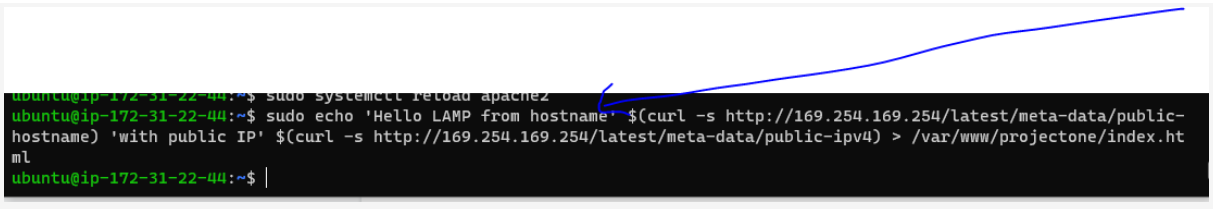


H: Reload Apache to let changes take effect using **sudo systemctl reload apache2**



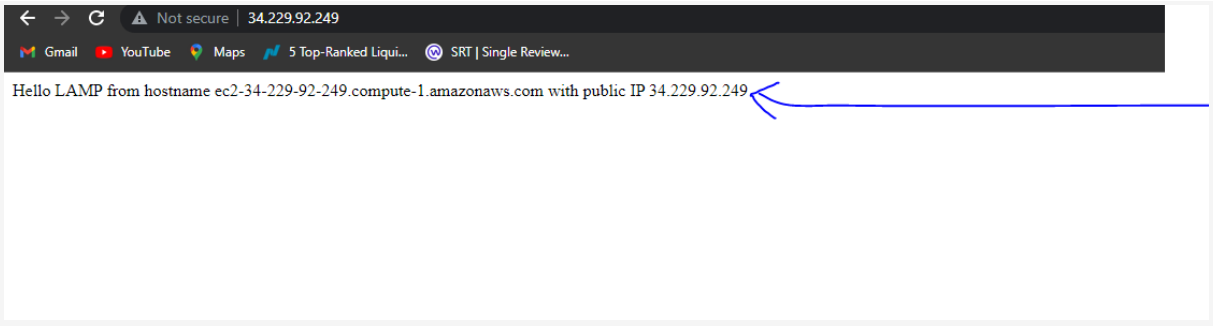
I: Creating an index.html file in this location /var/www/projectone to test if virtual hosts worked using

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/projectone/index.html



J: I opened my browser and try to open your website URL using IP address:

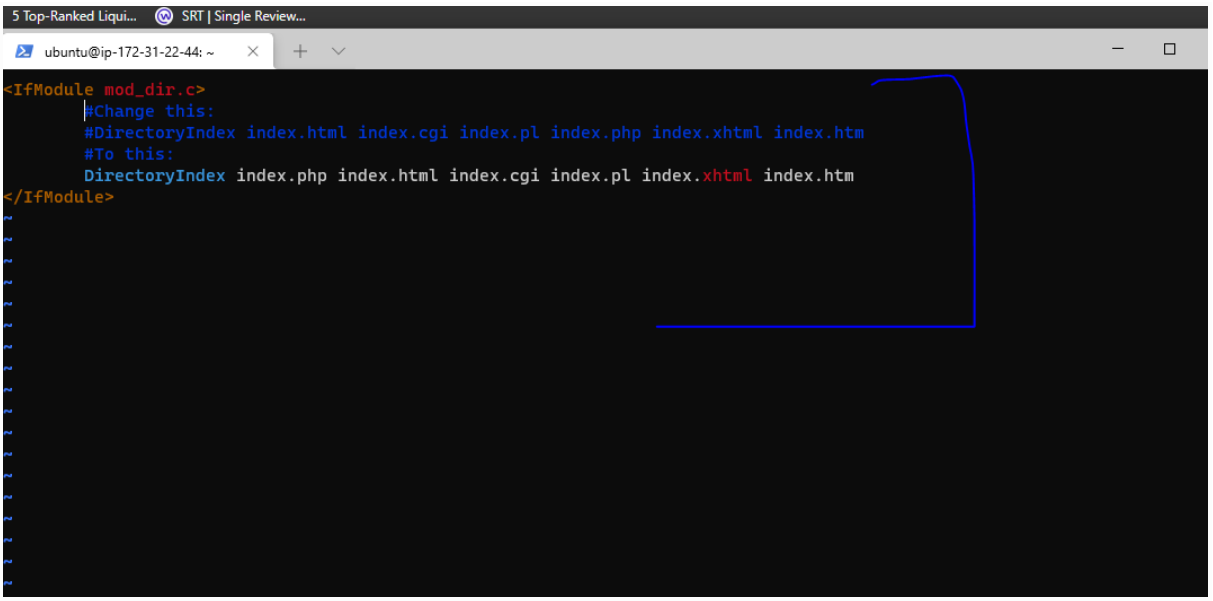
http://<Public-IP-Address>:80 using my public IP address and got this "Echo"output



STEP 5: Enable PHP on the website

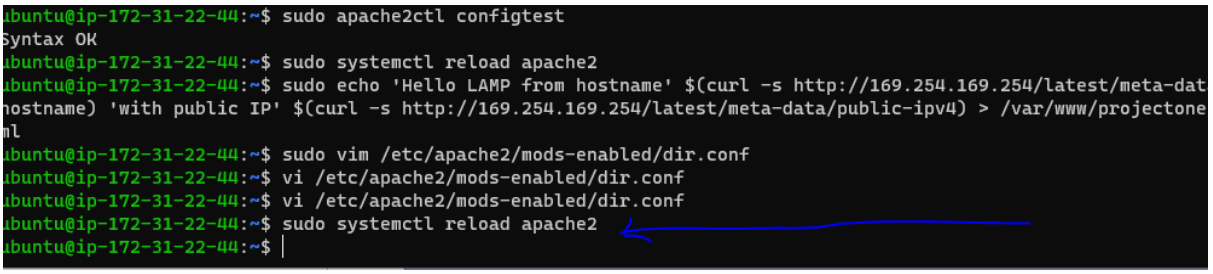
steps:

A: 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 using sudo vim /etc/apache2/mods-enabled/dir.conf



B: Reload Apache for changes to take place using

sudo systemctl reload apache2



C: I created a PHP script to test that PHP is correctly installed and configured properly on the server and also create a test script to see that Apache is able handle PHP test files

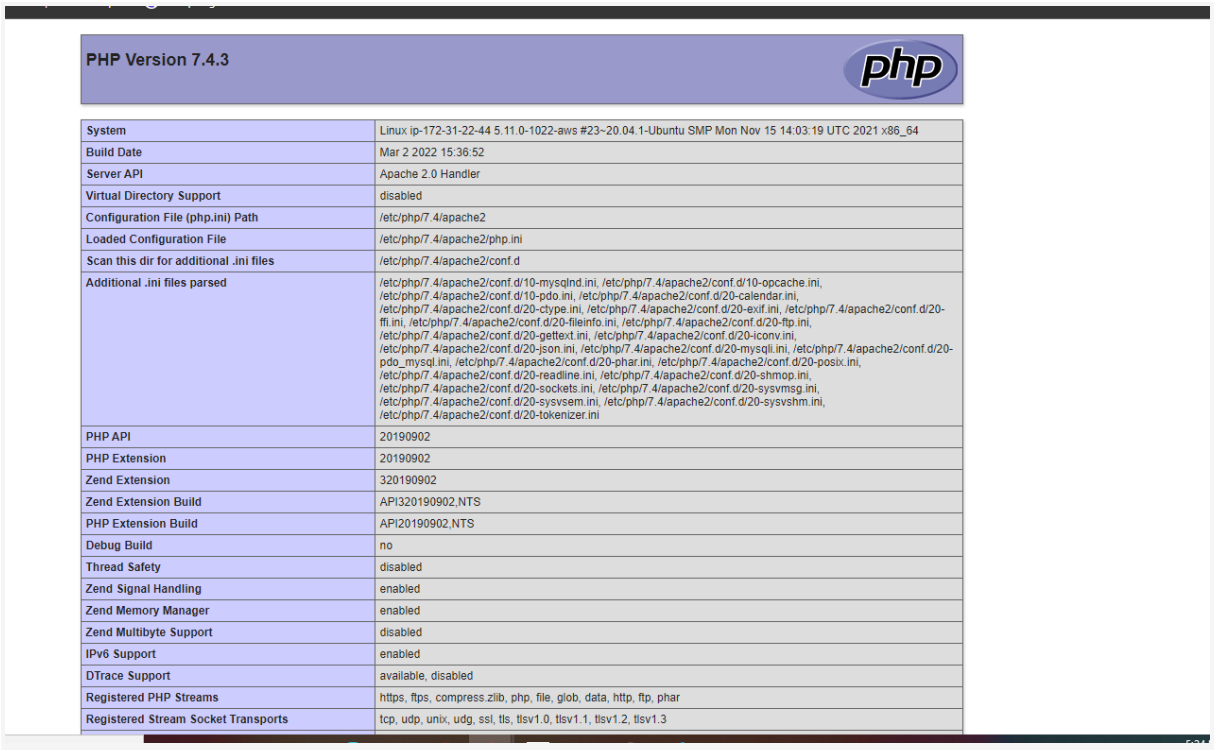
- Create a new file named index.php inside your custom web root folder:

vim /var/www/projectlone/index.php and paste the content below into the file and save file.

<?php

phpinfo();

D: Reload webpage and i got this page below which means my PHP installation is working properly



PHP Version 7.4.3

System	Linux ip-172-31-22-44 5.11.0-1022-aws #23~20.04.1-Ubuntu SMP Mon Nov 15 14:03:19 UTC 2021 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-mysqlnd.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/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.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-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-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-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
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	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

E: For security reasons, it is best to remove the file created as it contains sensitive information my ubuntu server, using the command below

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