

## LAMP STACK PROJECT IMPLEMENTATION

The main aim for this project is to explain the DevOps concepts and processes using a LAMP web stack. Some developers use this set of framework and tools to develop a software products .We would be carrying out this project in the AWS platform

LAMP is an acronym of sets of technology used to develop a technical software product.

Linux

Apache

MySQL

PHP

Please note : ( P could also stand for Python or Perl )

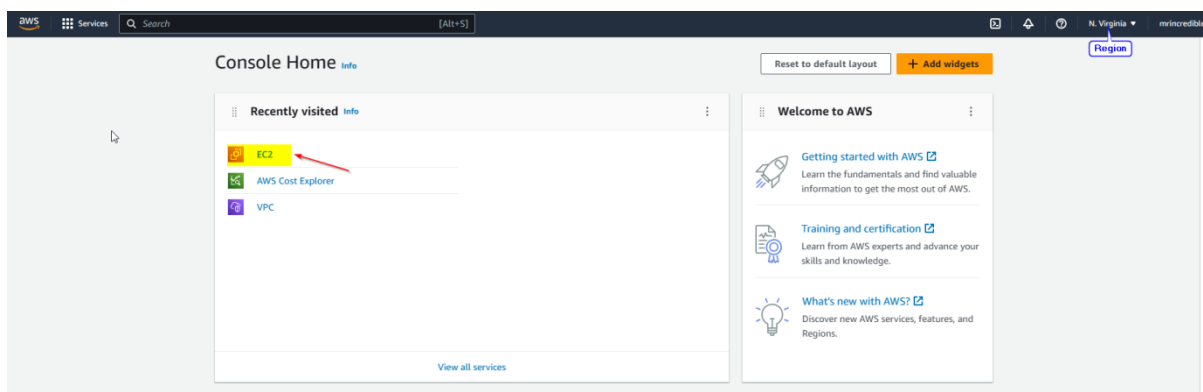
Apache server used is the apache2 version

Pre-requisite for the projects is the following.

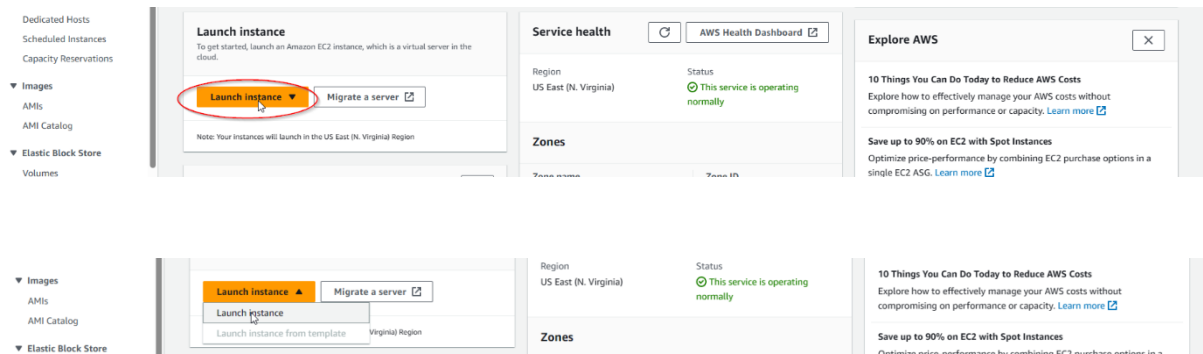
- 1) Fundamental Knowledge of Installing and downloading software
- 2) Basic Understanding of Linux Commands
- 3) AWS account login with EC2 instance
- 4) Internet connection

### IMPLEMENTATION STEPS:

- i) Ensure you login with your details to your AWS console via the <https://aws.amazon.com>
- ii) Click on the EC2 link to create instances.



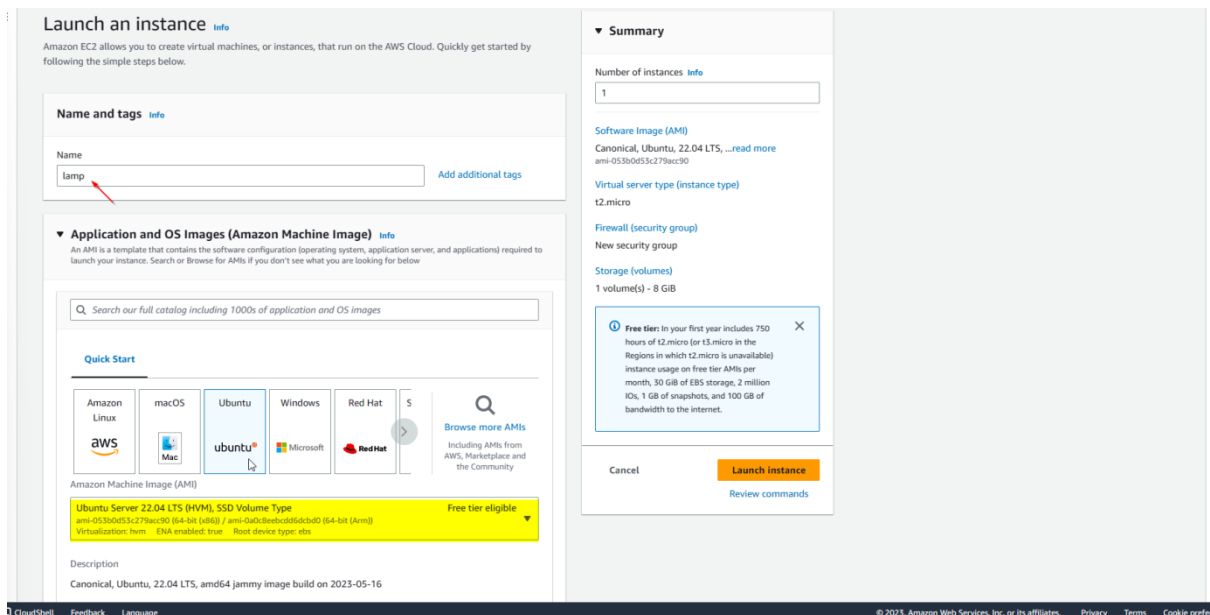
- iii) Click on launch instance dropdown button and select launch instance



iv) Fill in all relevant details to the lamp project such as :

Type in the name and additional tag to the project (lamp) .Selected ubuntu from the quick start option .Also note that the Amazon machine image selection varies from user to user

Select Ubuntu server 22.04 LTS (HVM), SSD Volume Type (Free Tier )



v) The instance type selected in the configuration is the t2 micro -free tier.

Click on the “Create new key pair” link.

Ensure the Checkbox remains on the “Create security group”.

▼ Instance type Info

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

Free tier eligible

All generations

Compare instance types

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select

Create new key pair

▼ Network settings Info

Edit

Network Info

vpc-0c3c371436c0dcd9d

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

We'll create a new security group called 'launch-wizard-16' with the following rules:

Allow SSH traffic from

▼ Summary

Number of instances Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-053b0d53c279acc90

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Review commands

vi) Typed in the key pair name, chose the default key pair type and private key file format (rsa and .pem) and clicked the “Create key pair button”

### Key pair name

Key pairs allow you to connect to your instance securely.

lamp

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

### Key pair type



RSA

RSA encrypted private and public key pair



ED25519

ED25519 encrypted private and public key pair

### Private key file format



.pem

For use with OpenSSH



.ppk

For use with PuTTY

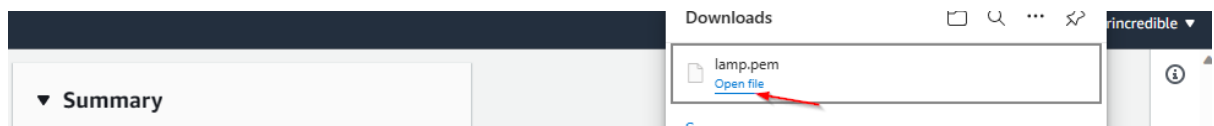


When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel

Create key pair

vii)The .pem file was downloaded successfully.



viii)I have deliberately chosen default settings to allow SSH traffic from anywhere as well as the storage volume given by AWS. Then proceed to launch our instance finally.



## x) Select checkboxes to view more details about the instance created.

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with options like EC2 Dashboard, Events, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, and Lifecycle Manager. The main area displays a table of EC2 instances. The 'lamp' instance is selected, and its details are shown in the 'Instance summary' tab. The public IP address 34.201.134.152 is highlighted with a red circle.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
nginx-server	i-09924c00d90fa7a9	Stopped	t2.micro	–	No alarms	us-east-1b	–	–	–
Demo-Apache2	i-01fde0d21739d518	Stopped	t2.micro	–	No alarms	us-east-1a	–	–	–
nginxserverpr...	i-0feb9757e77bb64ca	Stopped	t2.micro	–	No alarms	us-east-1a	–	–	–
stackmean	i-09f1e479ad3e0bdce	Stopped	t2.micro	–	No alarms	us-east-1a	–	–	–
stacklemp	i-0862f845ba4a743f2	Stopped	t2.micro	–	No alarms	us-east-1a	–	–	–
stacklamp	i-03ac707da751f0f22	Stopped	t2.micro	–	No alarms	us-east-1a	–	–	–
memstackmem	i-011b9e604e64ce6f4	Stopped	t2.micro	–	No alarms	us-east-1a	–	–	–
lamp	i-0985728291f6ac2e9	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-201-134-152.co...	34.201.134.152	–

Instance: i-0985728291f6ac2e9 (lamp)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary info

Instance ID  
i-0985728291f6ac2e9 (lamp)

IPv6 address  
–

Hostname type  
IP name: ip-172-31-91-102.ec2.internal

Answer private resource DNS name  
IPv4 (A)

Public IPv4 address  
34.201.134.152 open address

Instance state  
Running

Private IP DNS name (IPv4 only)  
ip-172-31-91-102.ec2.internal

Instance type  
t2.micro

Private IPv4 addresses  
172.31.91.102

Public IPv4 DNS  
ec2-34-201-134-152.compute-1.amazonaws.com | open address

Elastic IP addresses  
–

The public IP address shown on the screenshot should be copied as we would be using it on the console.  
Open git bash on visual studio code or whichever console is convenient to use .

We are using git bash here with Visual Studio Code

```
oshor@Oshority MINGW64 ~ (master)
$ cd Downloads/

oshor@Oshority MINGW64 ~/Downloads (master)
$ ssh -i "lamp.pem" ubuntu@ec2-34-201-134-152.compute-1.amazonaws.com
```

Type YES ,to connect

```
$ ssh -i "lamp.pem" ubuntu@ec2-34-201-134-152.compute-1.amazonaws.com
The authenticity of host 'ec2-34-201-134-152.compute-1.amazonaws.com (34.201.134.152)' can't be established.
ED25519 key fingerprint is SHA256:5I5neDj+0xU901oEwYZk1YP65Eqw3hm3qqQQWFEQT4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

You have successful connected to the EC2 instance launched on AWS via ssh

Type clear to have a clear console and proceed to updating the lists of packages in the package manager

```
ubuntu@ip-172-31-91-102:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
```

Then we run apache2 installation and click yes to complete installation

```
ubuntu@ip-172-31-91-102:~$ sudo apt install apache2
Reading package lists... Done
Enabling module authz_host.
```

**We have to verify that Apache is running in our Operating System.**

```
ubuntu@ip-172-31-91-102:~$ 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 2023-05-30 21:56:40 UTC; 58s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2404 (apache2)
```

**To proceed by launching the web server in the AWS Cloud, we need to navigate back to the security group on the platform to add a new rule for TCP port 80 which is the default for web browsers .**

**Once done we can access the web page on internet.**

**Click on security button.**

**Instance: i-0985728291f6ac2e9 (lamp)**

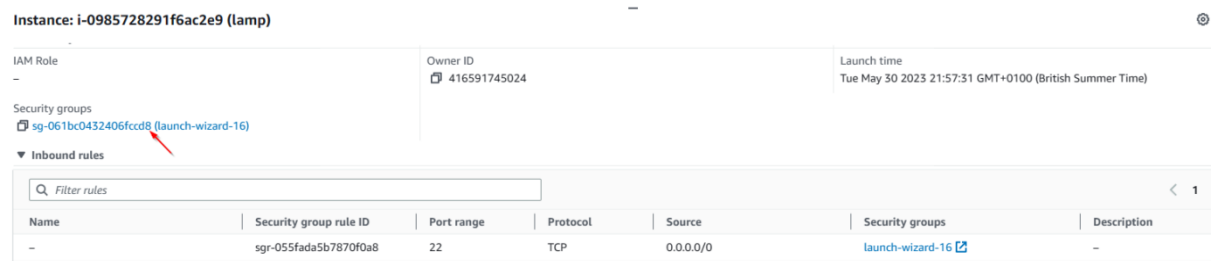


▼ Instance summary [Info](#)

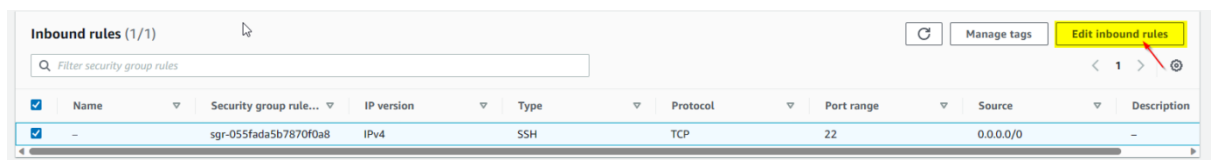
Instance ID

Public IPv4 address

**And click the security group link**



**Click on “Edit inbound rules “ in order to add a new rule for port 80**



## Add a new rule

Inbound rules [Info](#)

Security group rule ID: sgr-055fada5b7870f0a8

Type: SSH [Info](#)

Protocol: TCP [Info](#)

Port range: 22 [Info](#)

Source: Custom [Info](#)

Description - optional: [Info](#)

0.0.0.0 [X](#)

**Add rule**

Cancel [Preview changes](#) **Save rules**

## Type in the port range and click “Anywhere ipv4”

Security group rule ID: sgr-055fada5b7870f0a8

Type: SSH [Info](#)

Protocol: TCP [Info](#)

Port range: 22 [Info](#)

Source: Custom [Info](#)

Description - optional: [Info](#)

0.0.0.0 [X](#)

Custom TCP [Info](#)

Protocol: TCP [Info](#)

Port range: 80 [Info](#)

Source: Custom [Info](#)

Description - optional: [Info](#)

0.0.0.0 [X](#)

Custom [Info](#)

Anywhere-IPv4 [Info](#)

Anywhere-IPv6 [Info](#)

My IP [Info](#)

**Add rule**

Cancel [Preview changes](#) **Save rules**

## Click the “Save rules” Button

Inbound rules [Info](#)

Security group rule ID: sgr-055fada5b7870f0a8

Type: SSH [Info](#)

Protocol: TCP [Info](#)

Port range: 22 [Info](#)

Source: Custom [Info](#)

Description - optional: [Info](#)

0.0.0.0 [X](#)

Custom TCP [Info](#)

Protocol: TCP [Info](#)

Port range: 80 [Info](#)

Source: Anywhere-IPv4 [Info](#)

Description - optional: [Info](#)

0.0.0.0 [X](#)

**Add rule**

Cancel [Preview changes](#) **Save rules**

## Inbound rule successfully modified.

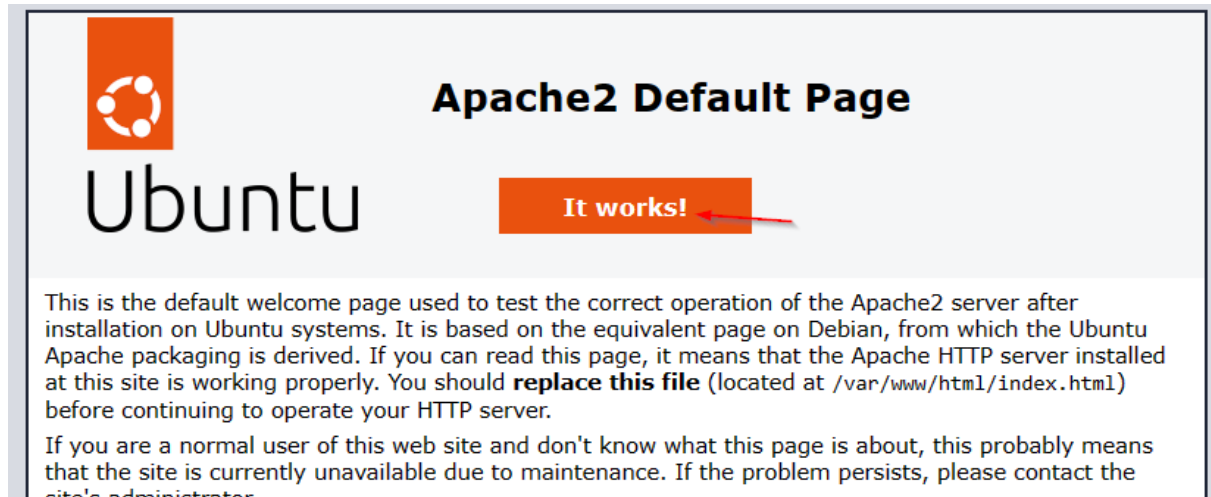
ⓘ Inbound security group rules successfully modified on security group (sg-061bc0432406fccd8 | launch-wizard-16)

► Details

EC2 > Security Groups > sg-061bc0432406fccd8 - launch-wizard-16

Open any browser of your choice and access the URL  
<http://34.201.134.152:80>





**Apache2 default page successfully displayed.**

**From the LAMP stack, we have implemented with Linux and got Apache ready .**

**Next step would be to get the MySQL installed.**

### **MYSQL INSTALLATION**

**Now that our web server is running, we need a relational database uses within the PHP environment hence we install MySQL server**

**Type “Y” and enter.**

```
ubuntu@ip-172-31-91-102:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
```

**When installation is finished, Log in to connect to the MySQL server as the administrator user root so that you can have access to the sudo command.**

```
ubuntu@ip-172-31-91-102:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.33-0ubuntu0.22.04.2 (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.
```

**It is important to set up a password for the user root using mysql\_native\_password as a default authentication method. Please note, Password not revealed for security purpose Exit MySQL**

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY
Query OK, 0 rows affected (0.02 sec)

mysql> exit
Bye
```

**Interactive script is started, and all modifications are answered with a Y/N response**

**Root user password was set Validate password: No  
Change password: No  
Remove anonymous user: No**

```
ubuntu@ip-172-31-91-102:~$ 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: No
Using existing password for root.
Change the password for root ? ((Press y|Y for Yes, any other key for No) : No

... skipping.
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.

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

... skipping.
```

**Disallow remote login: No**

**Remove test data base and access to it: No**

**Reload Privilege tables: Yes.**

```
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) : No

... skipping.
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) : No

... skipping.
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!
```

**Verify login details to ensure all details were inputted correctly and exiting MySQL**

```
ubuntu@ip-172-31-91-102:~$ sudo mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.33-0ubuntu0.22.04.2 (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> exit
Bye
```

**MySQL server was correctly installed and secured.**

**Next, we proceed to the PHP installation which is the final component of the LAMP STACK**

## **PHP INSTALLATION**

**PHP is the component that would process the codes to display dynamic content to the end user. Hence, we would need to install 3 packages namely :**

**1)PHP package 2) libapache2-mod-php 3) php-mysql .**

```
ubuntu@ip-172-31-91-102:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

**Installation continues.**

```
Created symlink /etc/systemd/system/timers.target.wants/phpsessionclean.timer → /lib/systemd/system/phpsessionclean.timer.
Setting up php8.1-common (8.1.2-1ubuntu2.11) ...

Creating config file /etc/php/8.1/mods-available/calendar.ini with new version

Creating config file /etc/php/8.1/mods-available/ctype.ini with new version
Progress: [ 56%] [#####.....]
```

**After installing, we check the PHP version.**

```
ubuntu@ip-172-31-91-102:~$ php -v
PHP 8.1.2-1ubuntu2.11 (cli) (built: Feb 22 2023 22:56:18) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
with Zend OPcache v8.1.2-1ubuntu2.11, Copyright (c), by Zend Technologies
```

**At this point the LAMP STACK implementation is completed and fully operational**

**We need to test our set up with a PHP script and this needs a proper APACHE virtual host to keep your website files and folder .Multiple website can be hosted on a single machine and the users would not notice**

## **CREATING AN APACHE VIRTUAL HOST FOR OUR WEBSITE TO USE .**

**Next step, making a directory for the site directory, running below**

```
ubuntu@ip-172-31-91-102:~$ sudo mkdir /var/www/projectlamp
ubuntu@ip-172-31-91-102:~$ sudo chown -R $USER:$USER /var/www/projectlamp
```

**Then proceed to edit a new site directory to input the virtual host information.**

```
ubuntu@ip-172-31-91-102:~$ sudo vi /etc/apache2/sites-available/projectlamp.conf
ubuntu@ip-172-31-91-102:~$
```

**Put the edited file in an insert mode by typing “i” without quotes and add the config files, press ESC ,save and exit with “ :wq” command**

```
<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>
```

Next check the content of the sites-available directory and you would see 3 configurations files on here .

```
ubuntu@ip-172-31-91-102:~$ sudo ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf  projectlamp.conf
```

With this configuration files, we would need to **DISABLE** the 000-default config file and **ENABLE** the new directory we created using the following command

```
ubuntu@ip-172-31-91-102:~$ sudo a2ensite projectlamp
Enabling site projectlamp.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-91-102:~$ sudo a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

After enabling and disabling done successfully, we would verify that there are no syntax errors with the command below

```
ubuntu@ip-172-31-91-102:~$ sudo apache2ctl configtest
Syntax OK
ubuntu@ip-172-31-91-102:~$
```

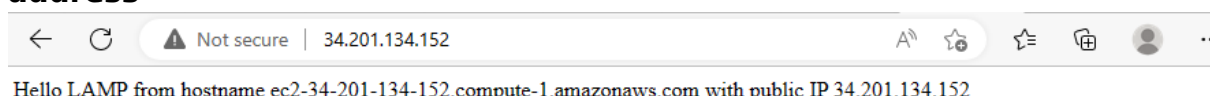
Then we proceed by reloading the Apache server to make these changes take effects.

```
ubuntu@ip-172-31-91-102:~$ sudo systemctl reload apache2
ubuntu@ip-172-31-91-102:~$
```

The new website is now active but the projectlamp has empty file .We create an index.html file in that location so that we can test our virtual host is performing as expected .

```
ubuntu@ip-172-31-91-102:~$ sudo echo "Hello LAMP from hostname $(curl -s http://169.254.169.254/latest/meta-data/public-hostname) 'with public IP' $(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)" > /var/www/projectlamp/index.html
ubuntu@ip-172-31-91-102:~$
```

Proceed to the browser and open the previous website using the ip address



The screenshot shows a web browser window with the address bar displaying "Not secure | 34.201.134.152". The page content reads: "Hello LAMP from hostname ec2-34-201-134-152.compute-1.amazonaws.com with public IP 34.201.134.152".

Echo successfully displayed but this is just to test the website.

Type "clear" command to clear screen.

## ENABLE PHP ON THE WEBSITE

We would need to set up an index.php file to replace the index.html file from the document root as it needs to override the default settings. This is a very useful maintenance page in PHP application

```
ubuntu@ip-172-31-91-102:~$ sudo vim /etc/apache2/mods-enabled/dir.conf
ubuntu@ip-172-31-91-102:~$
```

Files are edited correctly while index.php and index.html are in that order respectively.

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index
    x.shtml index.htm
</IfModule>
```

Edited successfully and the Apache needs to be reloaded again by the command below.

```
ubuntu@ip-172-31-91-102:~$ sudo systemctl reload apache2
ubuntu@ip-172-31-91-102:~$
```

Finally we would create the PHP script to test that PHP is correctly installed and configured on the server .The importance is to be able to handle and process request for PHP files with the command below

```
ubuntu@ip-172-31-91-102:~$ vim /var/www/projectlamp/index.php
ubuntu@ip-172-31-91-102:~$
```

Put the edited file in an insert mode by typing “i” without quotes and add the valid PHP code files, press ESC ,save and exit with “ :wq” command

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

Refresh the web page and you would see the web page server in a PHP perspective.



**This is the minimum requirement to set up an AWS instance with LINUX ,APACHE,MYSQL AND PHP for a web project.**

**Please note: Remember to terminate your EC2 instance.**