LAMP SETUP on AWS

A Comprehensive Guide to LAMP Stack Deployment on AWS

This guide provides a step-by-step approach to setting up a robust LAMP (Linux, Apache, MySQL, PHP) stack on your Amazon Web Services (AWS) EC2 instance. Each step is explained in detail, emphasizing professional best practices for security and performance.

Prerequisites:

- An AWS account
- A running EC2 instance with Amazon Linux or Ubuntu
- SSH access to the instance
- Basic knowledge of Linux command line

Introduction :

The LAMP stack (Linux, Apache, MySQL, PHP) is a popular combination for dynamic web applications. Its open-source nature, flexibility, and scalability make it an attractive choice for developers and enterprises alike. When it comes to hosting LAMP applications, Amazon Web Services (AWS) offers a robust and cost-effective platform. This report explores the various approaches to setting up a LAMP stack on AWS, highlighting the benefits and considerations for each.

Start by setting up an Amazon (EC2) Ubuntu instance keeping all the default settings (Check GitHub for steps and AWS setup)
Start by :-

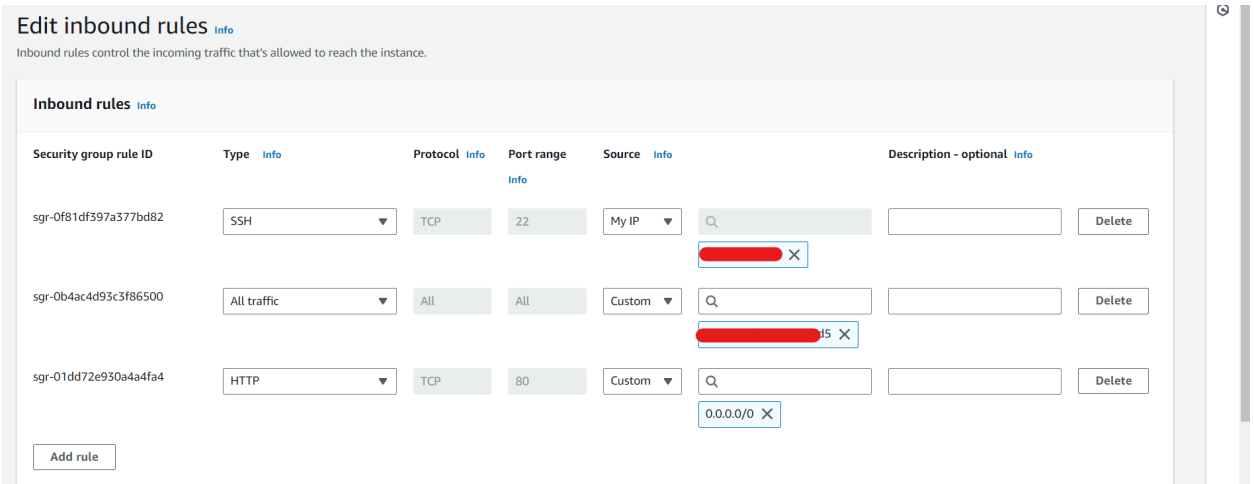
1. Installing Apache2:

1.1. Package Updates: Ensure your system is up-to-date by running sudo apt update.

```
ubuntu@ip-172-31-4-56:~$ sudo apt-update
sudo: apt-update: command not found
ubuntu@ip-172-31-4-56:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1268 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [260 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1257 kB
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [205 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1021 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [227 kB]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB
```

1.2. Security Groups: Configure security groups to restrict access to essential ports (80 for HTTP, 3306 for MySQL). This adds a vital layer of defense against unauthorized access.

1.3. IAM Roles: Assign IAM roles with minimal permissions to grant your LAMP instance access to relevant AWS services, adhering to the principle of least privilege.



2. Installing Apache2:

2.1. Installation: Execute **>> sudo apt install apache2** to install and configure Apache2, the web server that processes and delivers web content.

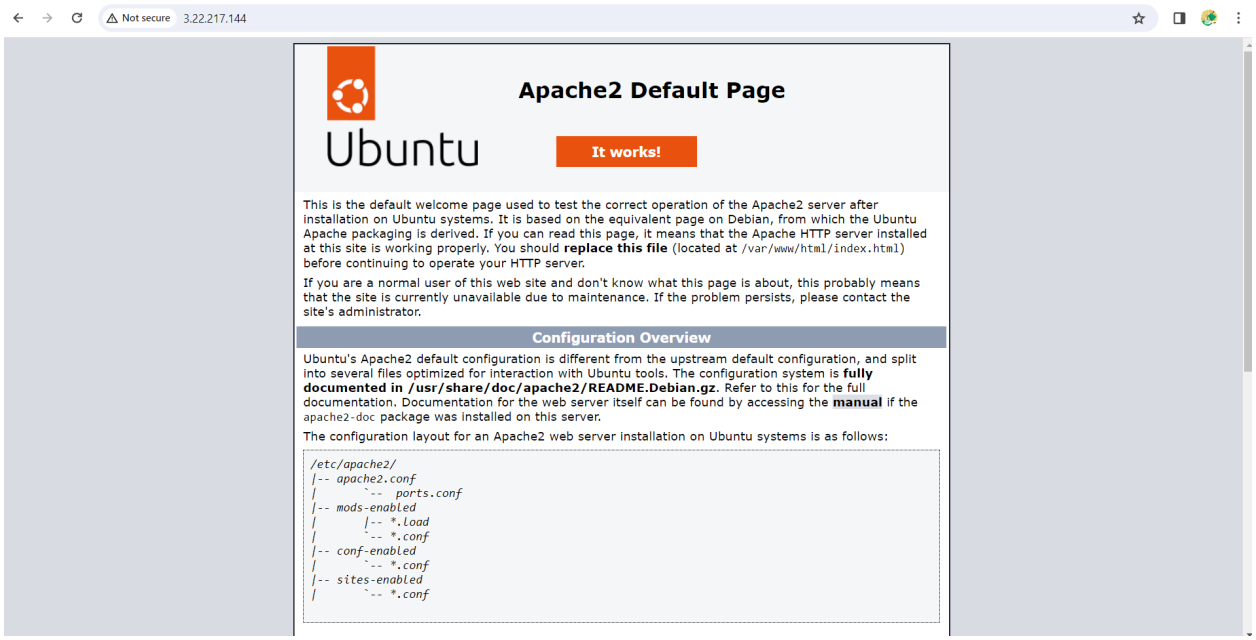
```
ubuntu@ip-172-31-4-56:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.3-0 mailcap mime-support ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 32 not upgraded.
Need to get 2139 kB of archives.
After this operation, 8518 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2 [92.
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.2
```

2.2. Start and Enable: Use **>> sudo apt install apache2** and **>> sudo start apache2 OR >> sudo systemctl start apache2** enable apache2 to start and configure Apache2 to automatically launch at system boot. It's also good practice to enable the it by doing *#enable automatic apache2 start service at boot time* **>> sudo systemctl enable apache2**

```
ubuntu@ip-172-31-4-56:~$ sudo start apache2
sudo: start: command not found
ubuntu@ip-172-31-4-56:~$ sudo systemctl start apache2
ubuntu@ip-172-31-4-56:~$ 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 Fri 2024-01-05 19:08:29 UTC; 1min 20s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2220 (apache2)
      Tasks: 55 (limit: 1121)
     Memory: 4.8M
        CPU: 36ms
     CGroup: /system.slice/apache2.service
             ├─2220 /usr/sbin/apache2 -k start
             ├─2222 /usr/sbin/apache2 -k start
             └─2223 /usr/sbin/apache2 -k start

Jan 05 19:08:29 ip-172-31-4-56 systemd[1]: Starting The Apache HTTP Server...
Jan 05 19:08:29 ip-172-31-4-56 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-4-56:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-s
Executing: /lib/systemd/systemd-sysv-install enable apache2
ubuntu@ip-172-31-4-56:~$ 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 Fri 2024-01-05 19:08:29 UTC; 3min 49s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2220 (apache2)
      Tasks: 55 (limit: 1121)
     Memory: 5.3M
        CPU: 47ms
     CGroup: /system.slice/apache2.service
             ├─2220 /usr/sbin/apache2 -k start
             ├─2222 /usr/sbin/apache2 -k start
             └─2223 /usr/sbin/apache2 -k start
```

2.3. Verification: Ensure Apache2 is running with >> >> **sudo systemctl status apache2**.



3. Securing MySQL Server:

3.1. Installation: Install MySQL Server with >> **sudo apt install mysql-server**.

```
ubuntu@ip-172-31-4-56:~$ 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-pth
  libfcgi0ldbl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libht
  libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23 libtimedate-
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-commo
```

3.2. Secure Installation: Run ***sudo mysql_secure_installation*** to set a strong password for the root user and disable remote access for enhanced security.

```
ubuntu@ip-172-31-4-56:~$ sudo systemctl status mysql-server
Unit mysql-server.service could not be found.
ubuntu@ip-172-31-4-56:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled
     Active: active (running) since Fri 2024-01-05 19:13:11 UTC; 34s ago
    Process: 3319 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, s
   Main PID: 3327 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 1121)
     Memory: 356.9M
        CPU: 1.201s
     CGroup: /system.slice/mysql.service
             └─3327 /usr/sbin/mysqld
```

3.3. Confirmation: Use **>> *sudo systemctl status mysql*** to verify MySQL is running.

4. Implementing PHP:

4.1. Package Installation: Install PHP and required modules for Apache2 and MySQL integration with

#Installing PHP with other packages required for integration with apache2 and mysql

>> sudo apt install php libapache2-mod-php php-mysql

*# **php**: Installs the PHP scripting language.*

*# **libapache2-mod-php**: Installs the Apache module for integrating PHP with the Apache web server.*

*# **php-mysql**: Installs the MySQL extension for PHP, allowing PHP to interact with MySQL databases.*

```
ubuntu@ip-172-31-4-56:~$ 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:
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php8.1 php php-common php-mysql php8.1 php8.1-cli php8.1-common php8.1-mysql
  php8.1-opcache php8.1-readline
0 upgraded, 11 newly installed, 0 to remove and 32 not upgraded.
Need to get 5265 kB of archives.
After this operation, 21.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:92ubuntu1 [12.4 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-common amd64 8.1.2-1ubuntu2.
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-opcache amd64 8.1.2-1ubuntu2
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-readline amd64 8.1.2-1ubuntu
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-cli amd64 8.1.2-1ubuntu2.14
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapache2-mod-php8.1 amd64 8.1.2-1
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapache2-mod-php all 2:8.1+92ubuntu1 [289
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1 all 8.1.2-1ubuntu2.14 [9158
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php all 2:8.1+92ubuntu1 [2756 B]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-mysql amd64 8.1.2-1ubuntu2.
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-mysql all 2:8.1+92ubuntu1 [1834 B]
Fetched 5265 kB in 0s (31.9 MB/s)
Selecting previously unselected package php-common.
(Reading database ... 66170 files and directories currently installed.)
Preparing to unpack .../00-php-common_2%3a92ubuntu1_all.deb ...
Unpacking php-common (2:92ubuntu1) ...
Selecting previously unselected package php8.1-common.
Preparing to unpack .../01-php8.1-common_8.1.2-1ubuntu2.14_amd64.deb ...
```

4.2. Apache2 Configuration: Edit /etc/apache2/mods-enabled/dir.conf with **>> sudo nano /etc/apache2/mods-enabled/dir.conf** to the DirectoryIndex line, ensuring PHP files are prioritized for execution.

*#Edit the **dir.conf** file by adding **index.php** as shown below:*
<IfModule mod_dir.c>
DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

4.3. Apache2 Restart: Execute **>>sudo systemctl restart apache2** to apply the configuration changes.

```
ubuntu@ip-172-31-4-56:~$ sudo systemctl restart apache2
ubuntu@ip-172-31-4-56:~$ sudo systemctl restart mysql
ubuntu@ip-172-31-4-56:~$ sudo nano /etc/apache2/mods-enabled/dir.conf
ubuntu@ip-172-31-4-56:~$ sudo systemctl restart apache2
ubuntu@ip-172-31-4-56:~$ cd ../../../
ubuntu@ip-172-31-4-56:/$ ls
bin   dev  home  lib32  libx32      media  opt   root  sbin  srv  tmp  var
boot  etc  lib   lib64  lost+found  mnt    proc  run   snap  sys  usr
```

Also add an php file in the HTML folder

```
ubuntu@ip-172-31-4-56:/var$ cd www
ubuntu@ip-172-31-4-56:/var/www$ ls
html
ubuntu@ip-172-31-4-56:/var/www$ cd html
ubuntu@ip-172-31-4-56:/var/www/html$ ls
index.html
ubuntu@ip-172-31-4-56:/var/www/html$ nano index.php
ubuntu@ip-172-31-4-56:/var/www/html$ sudo nano index.php
```

5. Deploying phpMyAdmin:

5.1. Installation: Install phpMyAdmin, the web-based interface for managing MySQL databases, with ***>> sudo apt install phpmyadmin***.

5.2. Apache2 Integration: Edit /etc/apache2/apache2.conf with **>>sudo nano and add Include /etc/phpmyadmin/apache.conf** at the end, linking phpMyAdmin to Apache2.

5.3. Apache2 Restart: Execute **>>sudo systemctl restart apache2** to activate the integration.

```
ubuntu@ip-172-31-4-56:/etc$ cd phpmyadmin/
ubuntu@ip-172-31-4-56:/etc/phpmyadmin$ ls
apache.conf  conf.d  config-db.php  config.footer.inc.php  config.header.inc.php  config.inc.php  lighttpd.conf  phpmyadmin.desktop  phpmyadmin.service
ubuntu@ip-172-31-4-56:/etc/phpmyadmin$ cd ..
ubuntu@ip-172-31-4-56:/etc$ cd apache2
ubuntu@ip-172-31-4-56:/etc/apache2$ ls
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available  sites-enabled
ubuntu@ip-172-31-4-56:/etc/apache2$ sudo nano apache2.conf
ubuntu@ip-172-31-4-56:/etc/apache2$ sudo systemctl restart apache2
ubuntu@ip-172-31-4-56:/etc/apache2$ cd ..
ubuntu@ip-172-31-4-56:/etc$ cd php
php/        phpmyadmin/
ubuntu@ip-172-31-4-56:/etc$ cd phpmyadmin/
ubuntu@ip-172-31-4-56:/etc/phpmyadmin$ ls
apache.conf  conf.d  config-db.php  config.footer.inc.php  config.header.inc.php  config.inc.php  lighttpd.conf  phpmyadmin.desktop  phpmyadmin.service
ubuntu@ip-172-31-4-56:/etc/phpmyadmin$ cat config
config-db.php          config.footer.inc.php  config.header.inc.php  config.inc.php
ubuntu@ip-172-31-4-56:/etc/phpmyadmin$ cat config.inc.php
<?php
/**
 * Debian local configuration file
 *
 * This file overrides the settings made by phpMyAdmin interactive setup
 * utility.
 *
 * For example configuration see
 *   /usr/share/doc/phpmyadmin/examples/config.sample.inc.php
 * or
 *   /usr/share/doc/phpmyadmin/examples/config.manyhosts.inc.php
 *
```

```
ubuntu@ip-172-31-4-56:/var/www$ cd html
ubuntu@ip-172-31-4-56:/var/www/html$ ls
index.html  index.php
ubuntu@ip-172-31-4-56:/var/www/html$ cat index.php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lamp Setup</title>
</head>
<body>

<div style="text-align: center; padding: 50px;">
    <h1>Hey, it's a Lamp Setup!</h1>
    <img src="lamp_image.jpg" alt="Lamp Image" width="300">
</div>

</body>
</html>

ubuntu@ip-172-31-4-56:/var/www/html$
```

Hey, it's a Lamp Setup!

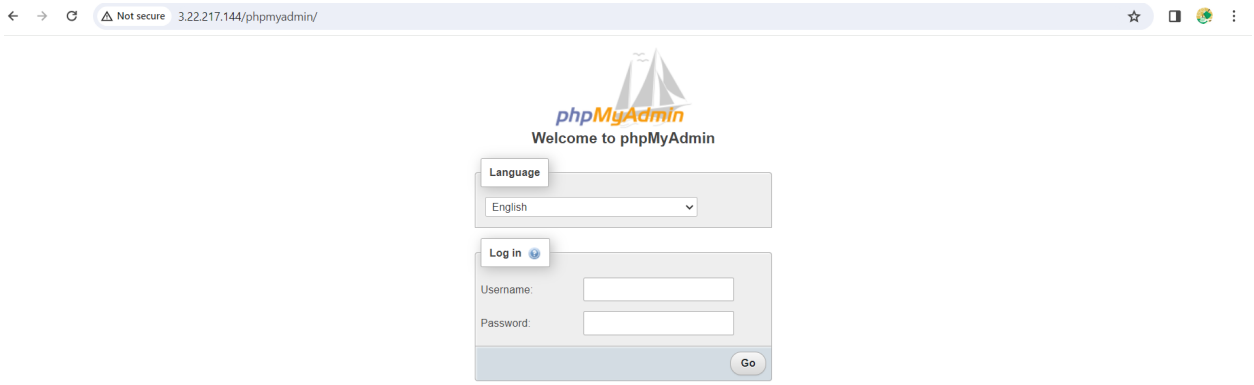
Lamp Image

6. Functional Validation:

6.1. Server Accessibility: Access your server's IP address or domain name in a web browser. The default Apache2 page should appear, confirming basic functionality.

6.2. PHP Verification: Create a PHP file (e.g., info.php) with phpinfo() content, upload it to the web server document root, and access it to validate PHP execution.

6.3. phpMyAdmin Access: Open http://your-server-ip/phpmyadmin in your browser. You should be able to log in and manage your MySQL databases using the phpMyAdmin interface.



Conclusion :

Having authored by me, Neil Machado, this comprehensive guide walks you through the meticulous steps of setting up an Amazon EC2 instance and configuring it for a LAMP stack. With an emphasis on clarity and precision, I've crafted this resource as a valuable companion for anyone looking to effortlessly deploy web development environments on AWS. My expertise ensures that the provided instructions are insightful and user-friendly, making this report an indispensable tool for those venturing into AWS-based LAMP stack setups.