LINUX, APACHE, MYSQL, PHP(LAMP) Stack Setup on AWS Cloud Environment

Authored by Samuel Vaz

1. Introduction

A strong web server infrastructure is essential for any online endeavor in the modern digital world. This document explores the world of Linux, Apache, MySQL, and PHP (LAMP) stacks, an open-source powerhouse, and walks you through the smooth setup and use of these stacks on the Amazon Web Services (AWS) network. This report gives you the information and resources you need to launch your first website, regardless of your level of experience as a developer or as a web enthusiast.

This comprehensive guide tackles the intricacies of setting up and configuring a LAMP stack on AWS. From choosing the right instance type to securing your database, we delve into each step with practical instructions and best practices.

This report is your roadmap to:

- Understanding the benefits of hosting LAMP on AWS
- Selecting the optimal AWS services for your LAMP stack
- Step-by-step instructions for setting up and configuring each component

2. Disclaimer and Legal

2.1 Disclaimer:

This report, authored by Mr. Samuel Vaz, provides general information on setting up a LAMP stack on AWS. It is intended for educational purposes only and should not be interpreted as professional advice or specific configuration instructions. The author disclaims all warranties, express or implied, of accuracy, completeness, or fitness for a particular purpose. Always conduct your own research and consult with qualified professionals before implementing any technical solutions.

2.2 Legal:

- Software Licensing: You are responsible for complying with all applicable software licenses for the components of your LAMP stack.
- AWS Terms of Service: Your use of AWS is governed by the AWS Service Terms and Conditions. Please review these terms carefully before proceeding.
- Security: You are solely responsible for the security of your LAMP stack and any data it stores. It is your obligation to implement appropriate security measures to protect your data and systems.
- Unethical Use: Mr. Samuel Vaz, the author of this document, expressly condemns any unethical use of the information presented. This includes, but is not limited to, any activity that may harm individuals, organizations, or systems. The author will not be held liable for any consequences arising from such misuse.
- Personal Responsibility: Users of this document assume full responsibility for any harm or damage caused by their actions, including harm to individuals, organizations, or any other entities. The author takes no responsibility for any consequences arising from the use or misuse of this information.

3. Launching AWS EC2 Instance

In this section of the report we'll embark on a step-by-step journey to create and configure your EC2 instance, specifically tailored for hosting a LAMP(LINUX, APACHE, MYSQL, PHP) stack. Visual aids will accompany each step to enhance clarity and ensure a seamless setup experience.

Step 1: AWS Console

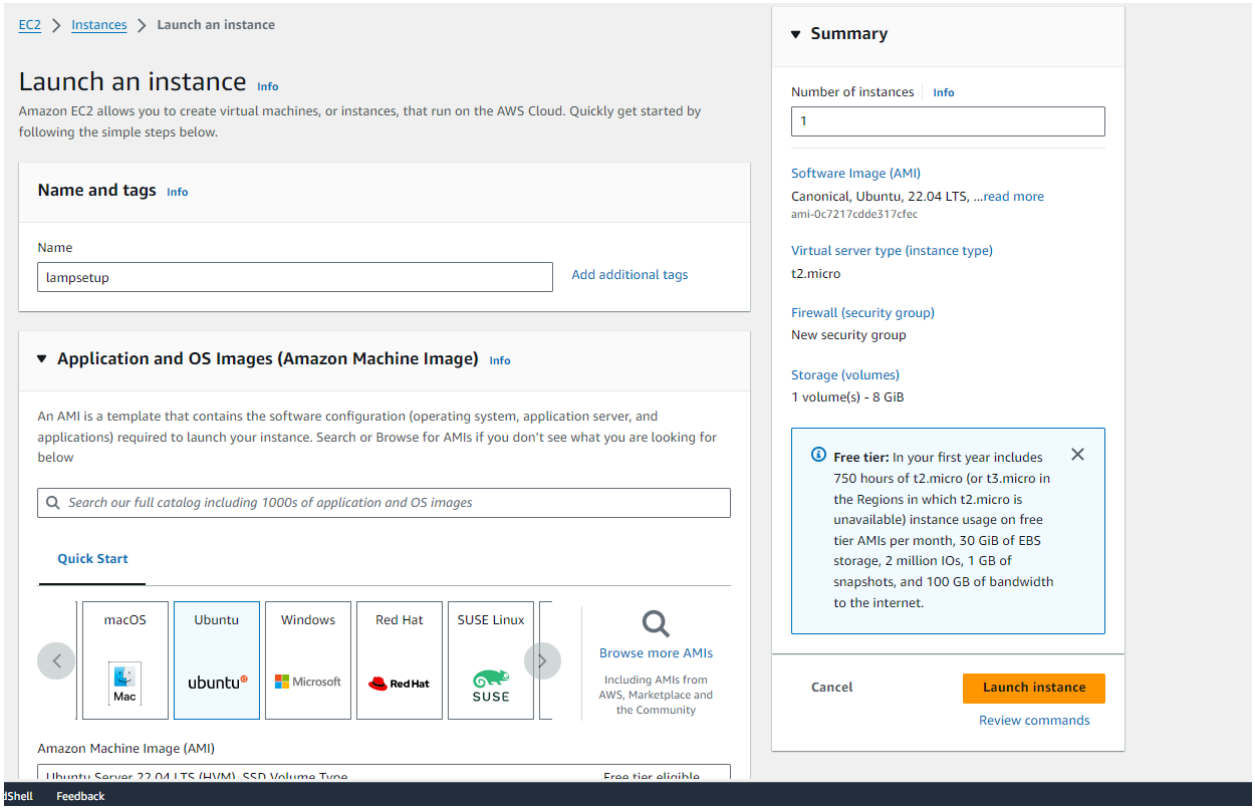
- Go to the AWS Management Console.
- Sign in to your AWS account or create a new one.

Step 2: Navigate to EC2 Dashboard and Launch an Instance

- In the AWS Management Console, locate the "Services" dropdown and select "EC2" under the "Compute" section.
- In the EC2 Dashboard, click on the "Instances" link in the left navigation pane.
- Click the "Launch Instances" button.

Step 4: Choose an Amazon Machine Image (AMI)

- Choose "Ubuntu Server" as your AMI. Select the version that is eligible for the AWS Free Tier.

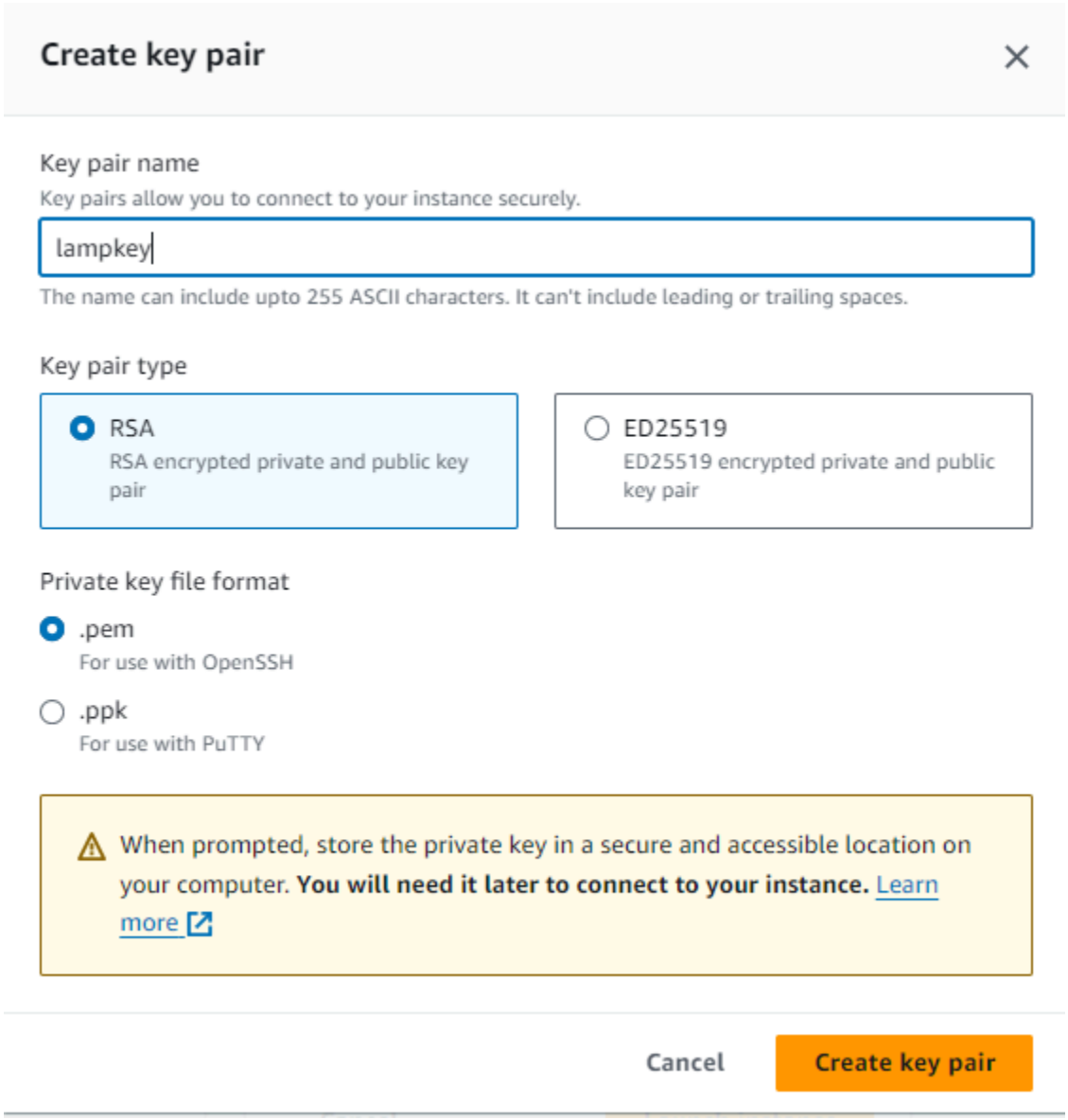


Step 5: Choose an Instance Type

- Select the "t2.micro" instance type. This type is eligible for the AWS Free Tier.

Step 6: Create a Key Pair

- Choose "Create a new key pair" and provide a name. Click on "Create Key Pair" to download the private key file (e.g., your-key-pair.pem). Store it in a secure location.



Step 7: Configure Security Group

- Use the default security group provided by AWS. It allows basic inbound SSH access.

Step 8: Configure Instance Details

- For the number of instances, network settings, and user data, use default settings or adjust as needed for your use case.

Step 9: Add Storage

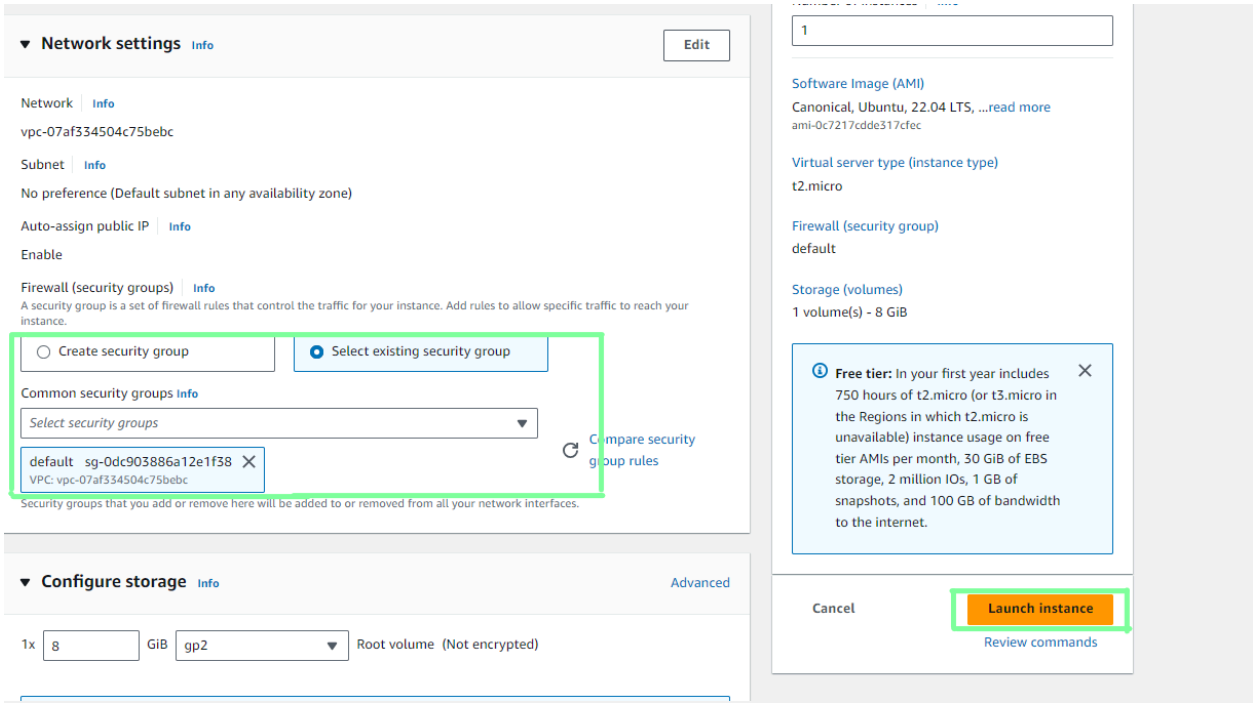
- Specify the size and type of storage for your instance. The default storage size is eligible for the AWS Free Tier.

Step 10: Add Tags (Optional)

- Add tags to your instance for better organization and identification. Tags are key-value pairs.

Step 11: Review and Launch

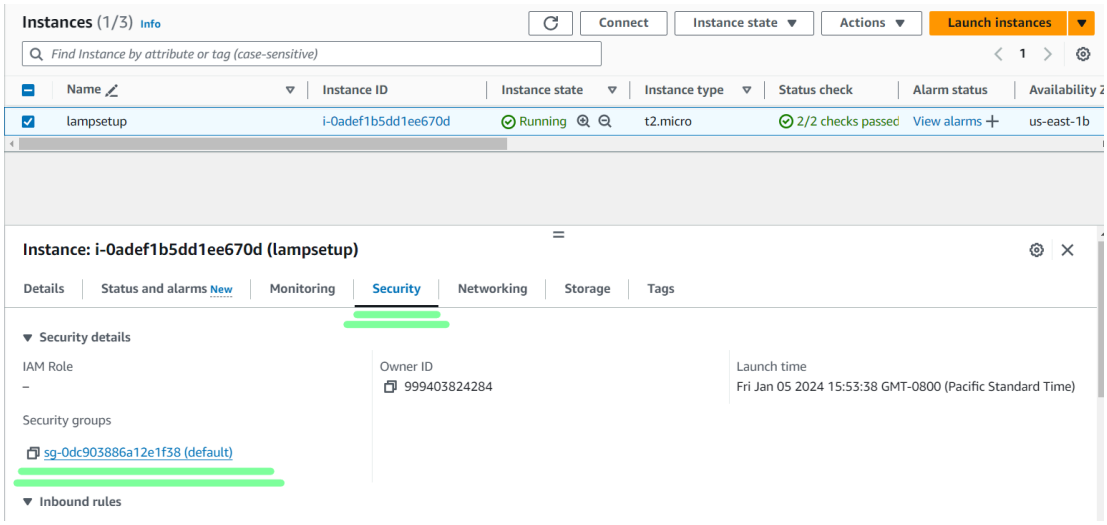
- Review your configuration settings. Click "Launch" when you are ready.

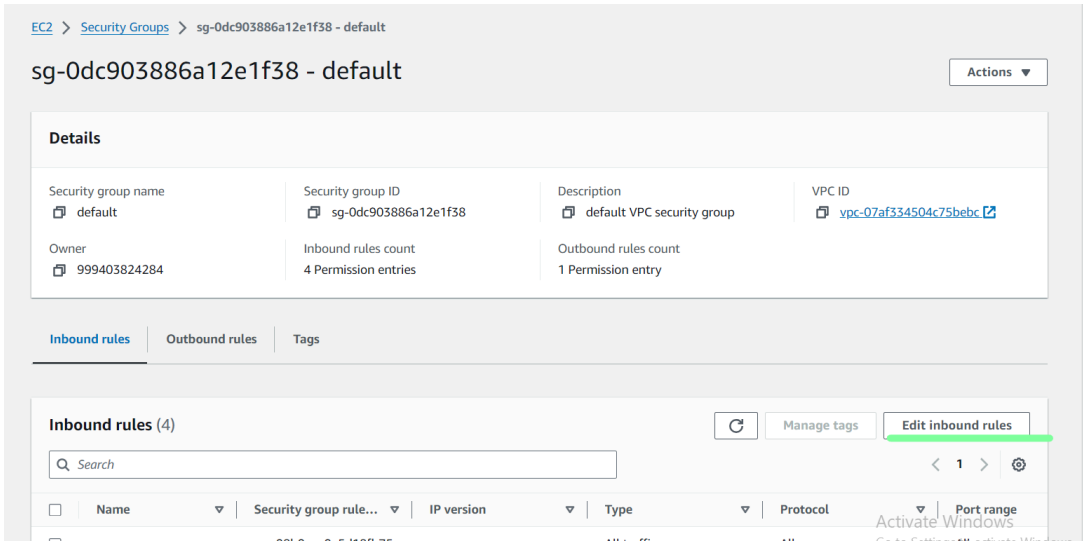


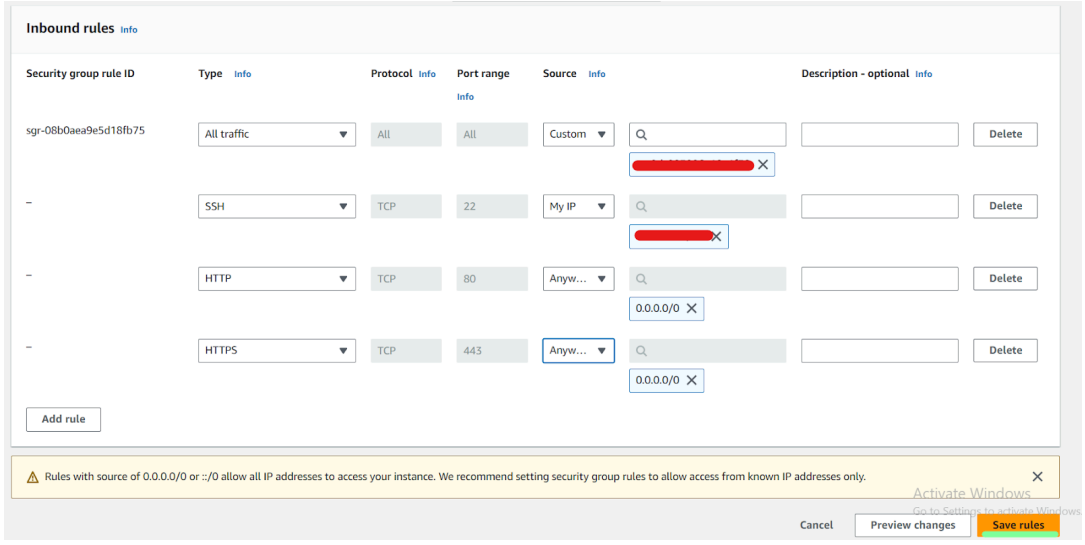
Step 13: Add Inbound Rules

- Create Inbound rules for HTTP, HTTPS, and SSH.
- Create a SSH rule for allowing connection only from your IPv4
- Create HTTP and HTTPS rules for allowing traffic from anywhere

- Refer below screenshots





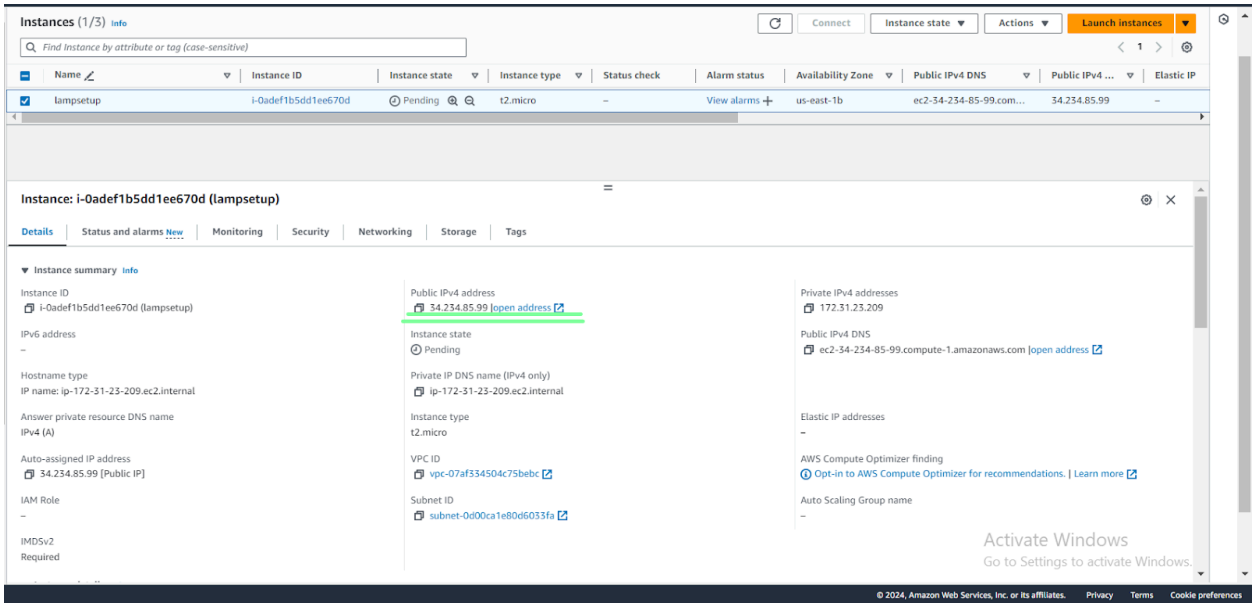


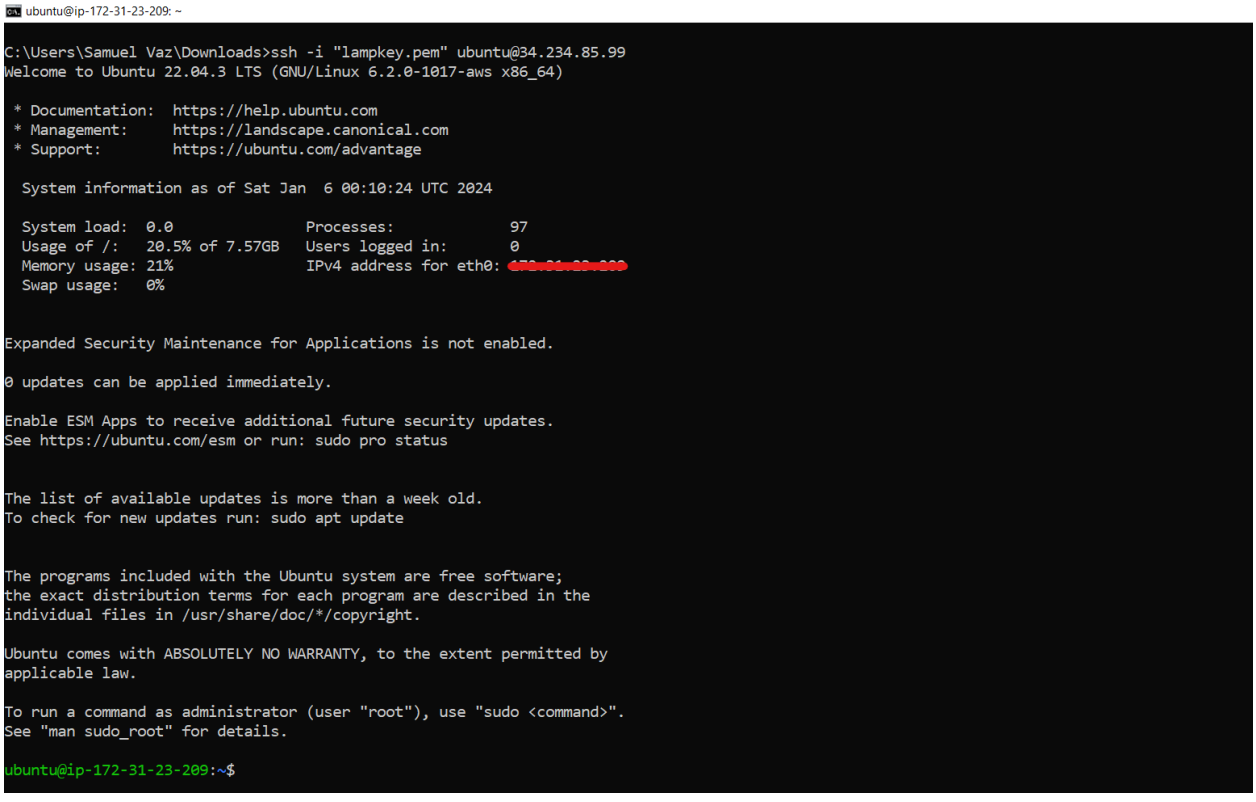
Step 13: Connect to Your Instance

- Use the key pair you generated to connect to your instance using SSH. The connection command will look like this:

```
>> ssh -i "your-key-pair.pem" ubuntu@your-instance-ip
```

- Refer below screenshots





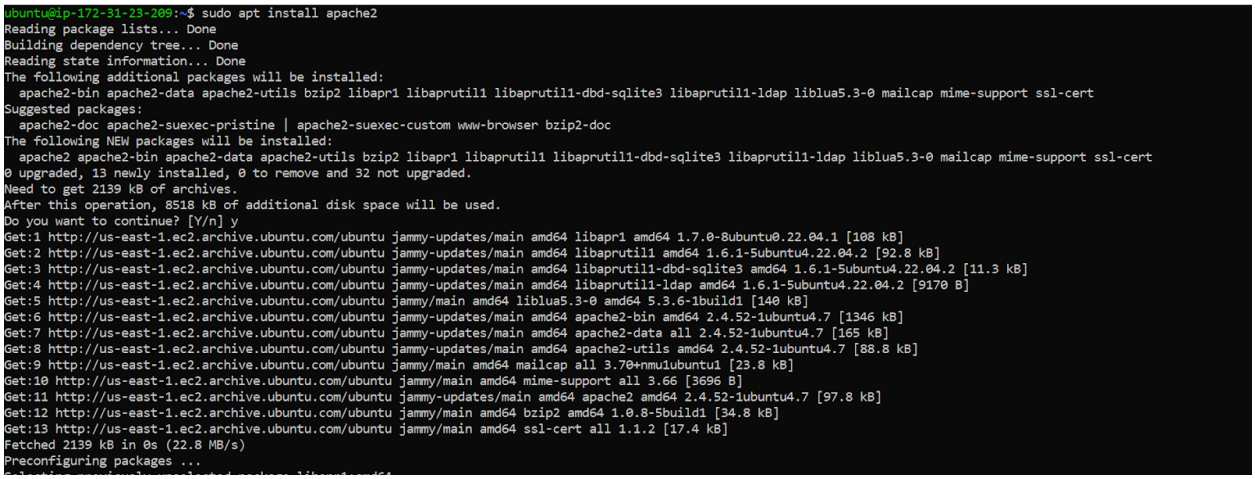
4. LAMP Stack Setup

4.1 Installing Apache2:

Step 1: Installing Apache2

- Install Apache2 Webserver using below command

  >> *sudo apt install apache2*



Step 2: Starting apache2 webserver

- After Installing the Apache2, start Apache2 service using below command

  >> *sudo systemctl start apache2*

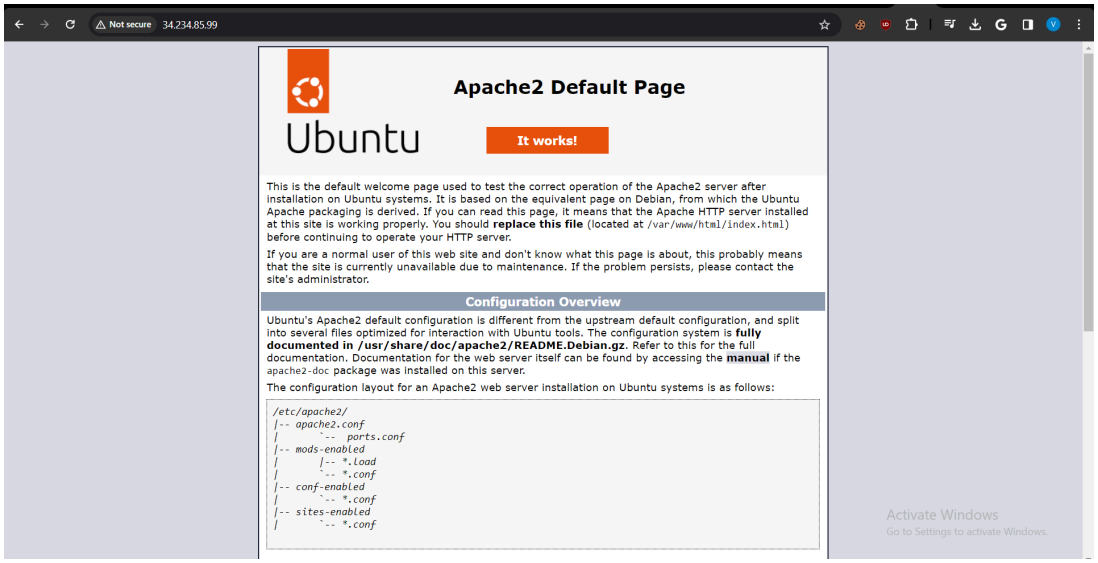
- Checking whether the apache2 is running or not

  >> *sudo systemctl status apache2*

- Enable automatic apache2 start service at boot time

  >> *sudo systemctl enable apache2*

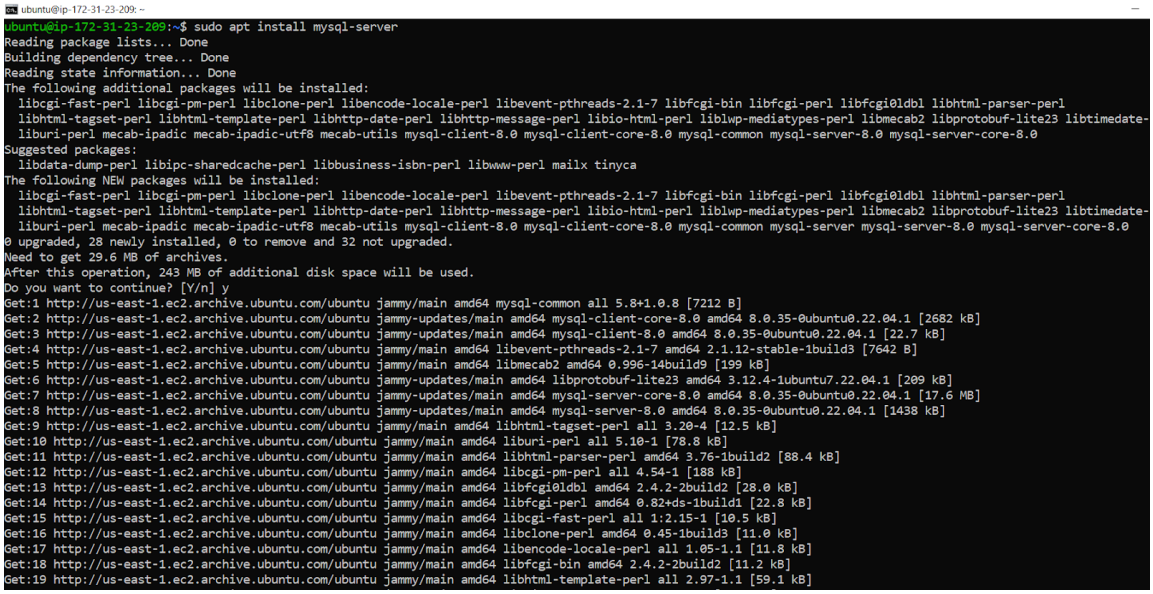
```
ubuntu@ip-172-31-23-209: ~
ubuntu@ip-172-31-23-209:~$ sudo systemctl start apache2
ubuntu@ip-172-31-23-209:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-01-06 00:22:44 UTC; 1min 17s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2407 (apache2)
      Tasks: 55 (limit: 1121)
     Memory: 4.8M
        CPU: 36ms
     CGroup: /system.slice/apache2.service
             ├─2407 /usr/sbin/apache2 -k start
             ├─2409 /usr/sbin/apache2 -k start
             └─2410 /usr/sbin/apache2 -k start

Jan 06 00:22:44 ip-172-31-23-209 systemd[1]: Starting The Apache HTTP Server...
Jan 06 00:22:44 ip-172-31-23-209 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-23-209:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
ubuntu@ip-172-31-23-209:~$
```



4.2 Installing MYSQL-SERVER

- Installing MySQl Server

  `>> sudo apt install mysql-server`



- Check if the mysql-server is running

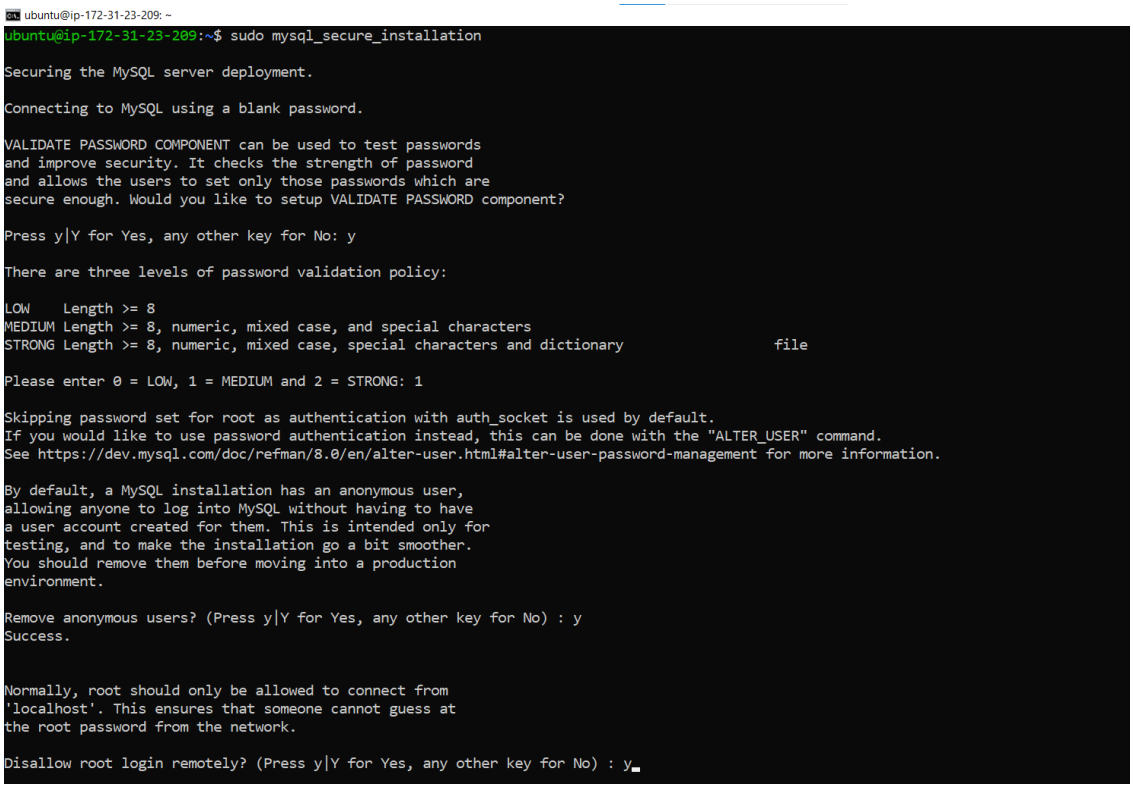
  >> *sudo systemctl status mysql*

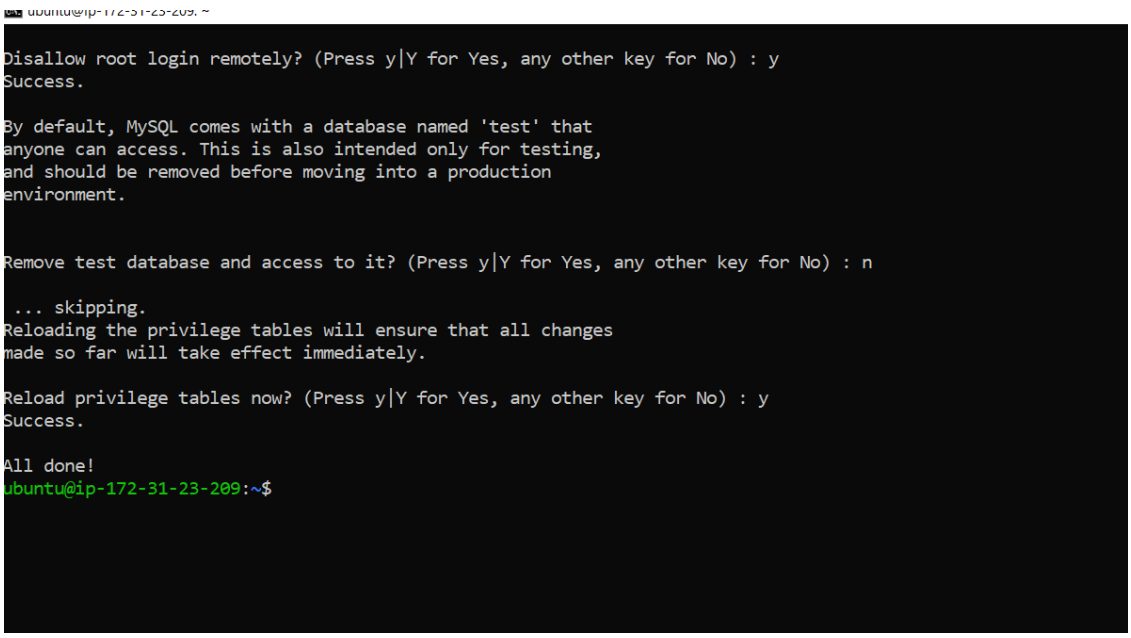
```
ubuntu@ip-172-31-23-209:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-01-06 00:25:50 UTC; 6min ago
    Process: 3502 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 3510 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 1121)
     Memory: 357.2M
        CPU: 2.666s
     CGroup: /system.slice/mysql.service
             └─3510 /usr/sbin/mysqld

Jan 06 00:25:49 ip-172-31-23-209 systemd[1]: Starting MySQL Community Server...
Jan 06 00:25:50 ip-172-31-23-209 systemd[1]: Started MySQL Community Server.
```

4.3 MYSQL SECURE INSTALLATION

- Securing MySql Installation

  >> *sudo mysql_secure_installation*





- Create a new user in MySQL, you can use the CREATE USER statement followed by the IDENTIFIED BY clause to set the password for the user. Here is the basic syntax:

sql> CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'newpassword';

```
ubuntu@ip-172-31-23-209:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.35-0ubuntu0.22.04.1 (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> CREATE USER '[redacted]'@'localhost' IDENTIFIED BY '[redacted]';
Query OK, 0 rows affected (0.02 sec)
```

Note: Remember the user credentials to log into phpmyadmin

- After creating the user, you need to grant the necessary privileges to the user. The GRANT statement is used for this purpose. For example, to grant all privileges on all databases to the new user:

sql> GRANT ALL PRIVILEGES ON *.* TO 'newuser'@'localhost' WITH GRANT OPTION;

```
mysql> GRANT ALL PRIVILEGES ON *.* TO '[redacted]'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

- Finally, you need to apply the changes by running the FLUSH PRIVILEGES statement:

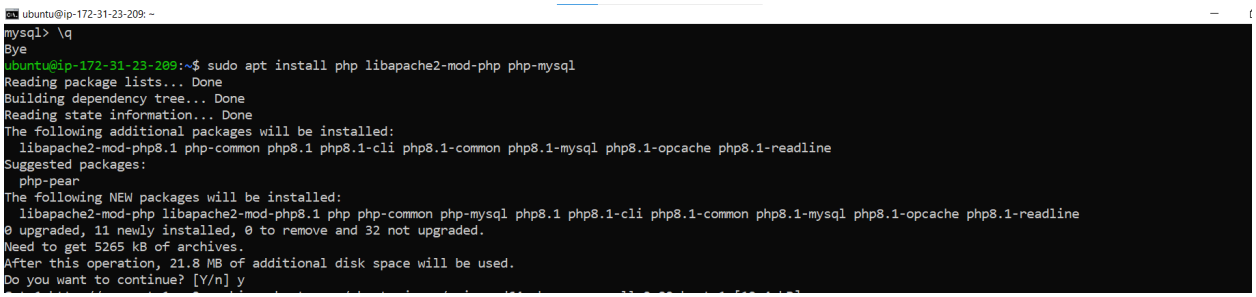
sql> FLUSH PRIVILEGES;
sql> \q

```
mysql> GRANT ALL PRIVILEGES ON *.* TO '[redacted]'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

4.4 Installing PHP

- 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.

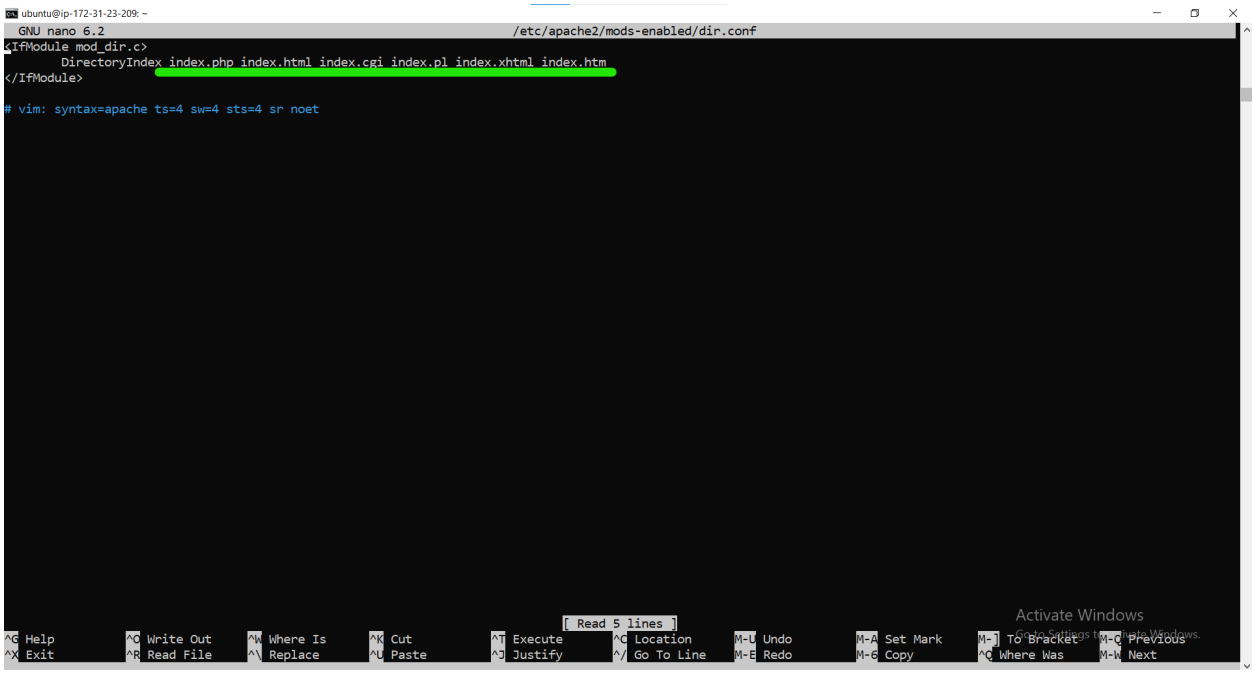
4.5 Configuring PHP:

- Configuring apache2 to use php:

```
>> sudo nano /etc/apache2/mods-enabled/dir.conf
```

- Edit the **dir.conf** file by adding **index.php** as shown below:

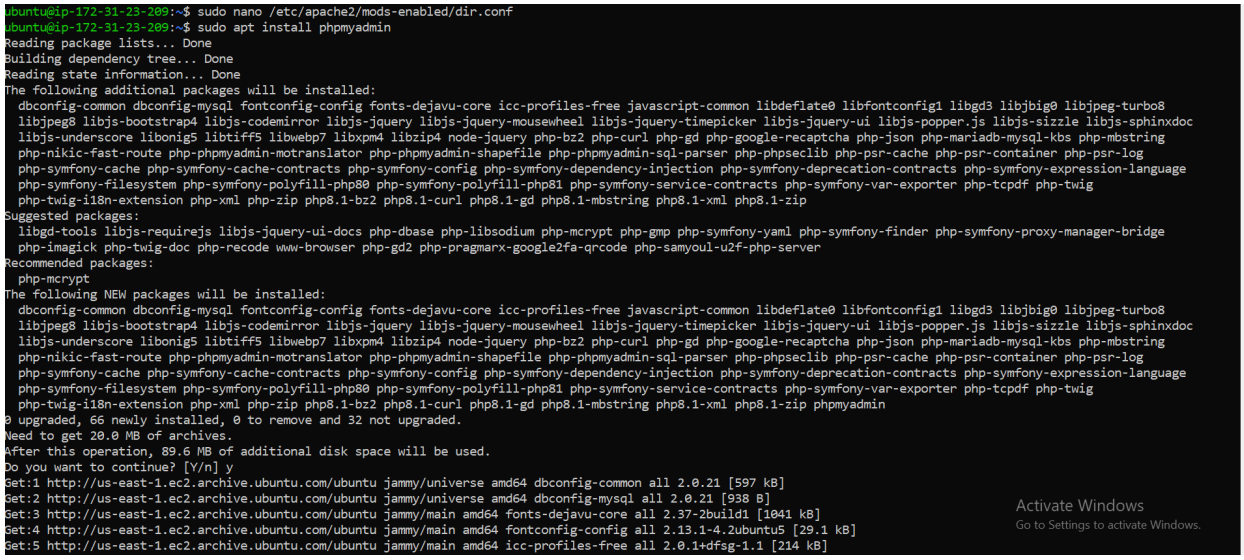
```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl
index.xhtml index.htm
</IfModule>
```

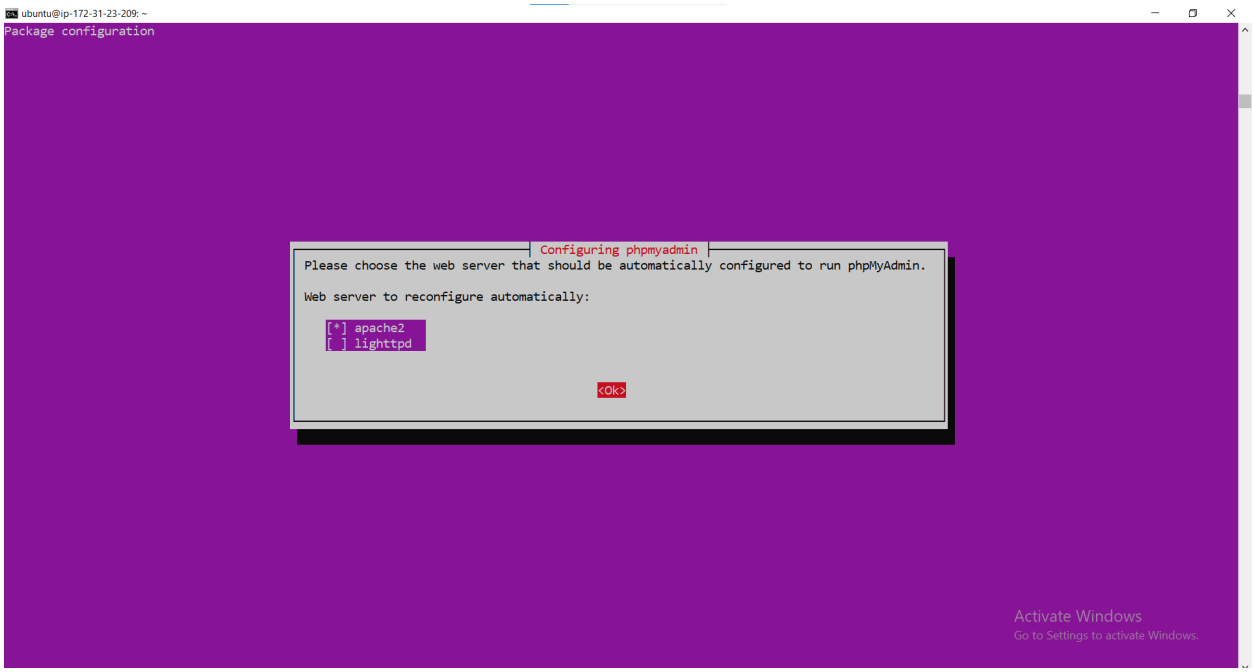


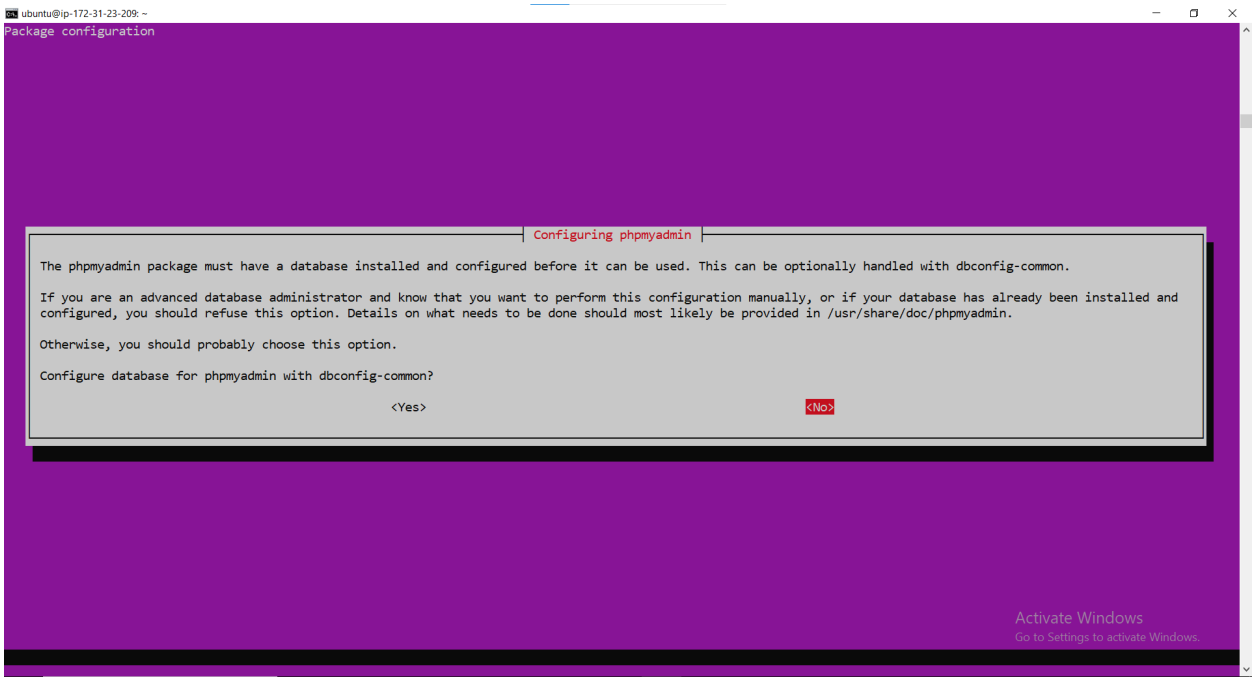
4.6 Installing PHPMYADMIN:

- Installing PHPMYADMIN

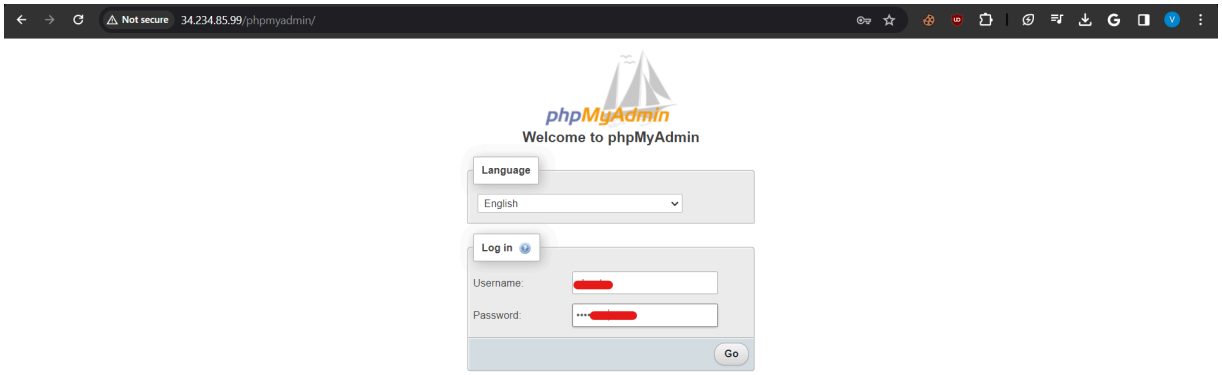
  >> *sudo apt install phpmyadmin*

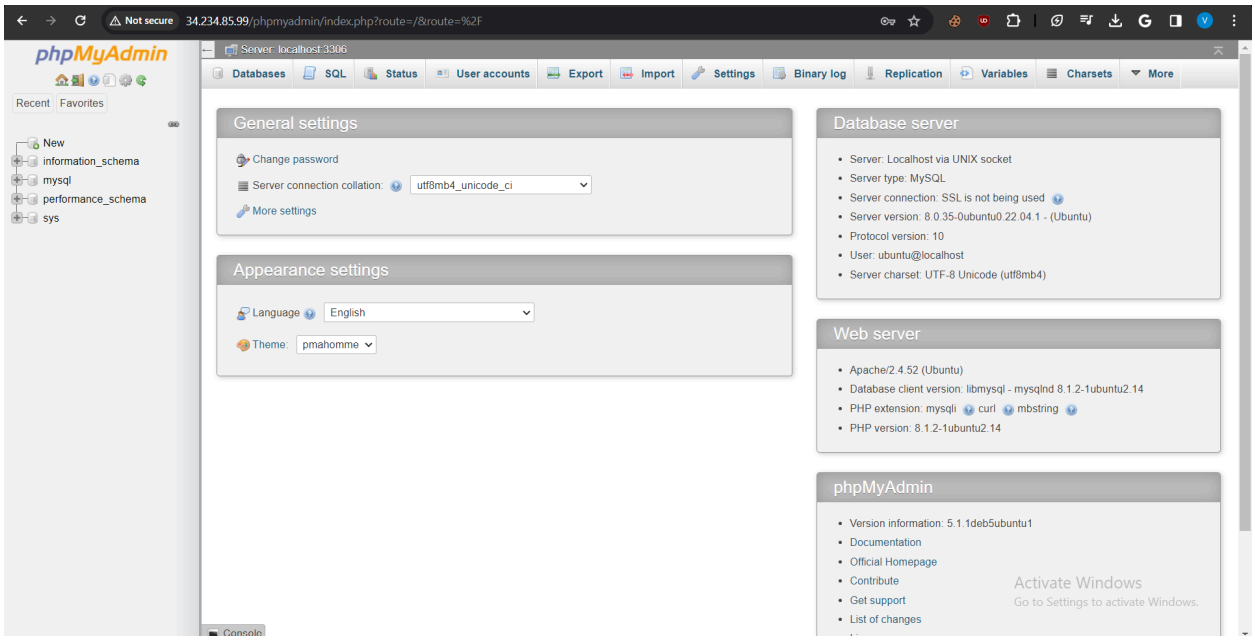






- Use the user credentials creating after **mysql_secure_installation** to login into phpmyadmin





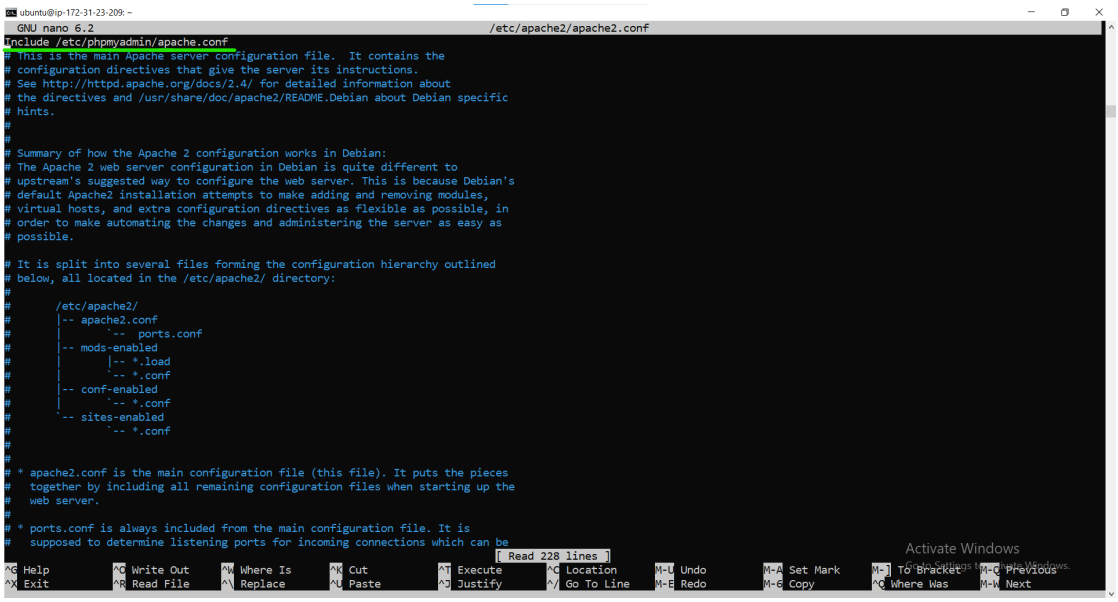
- Configuring the phpmyadmin with apache2

  >> *sudo nano /etc/apache2/apache2.conf*

```
ubuntu@ip-172-31-23-209:~$ sudo nano /etc/apache2/apache2.conf
```

- Add below statement in the apache.conf

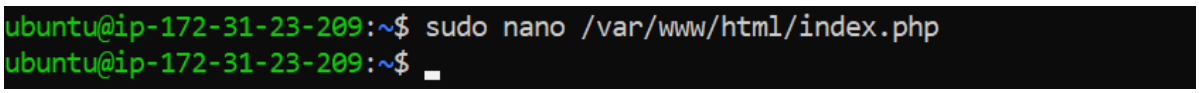
  Include /etc/phpmyadmin/apache.conf

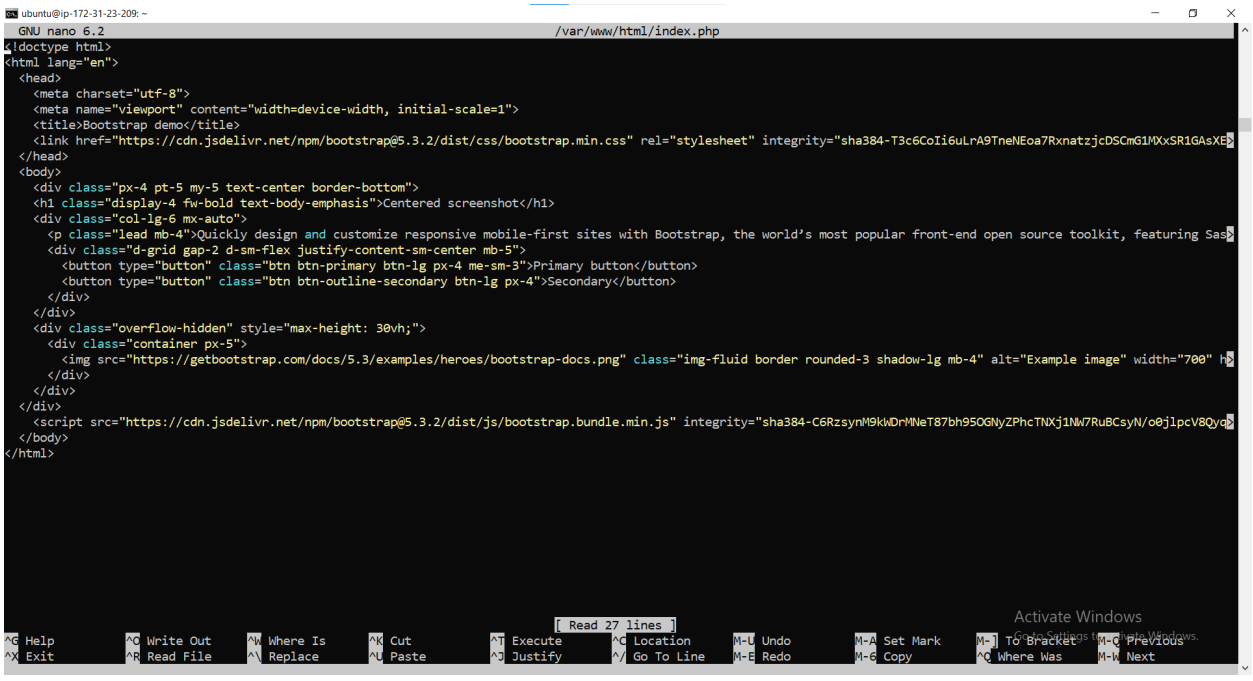


4.7 Creating a Web Page:

- Create an '**index.php**' file in **/var/www/html/** and Paste any sample html/php script in it.

>> sudo nano /var/www/html/index.php





Final Website:

Not secure 34.234.85.99

Centered screenshot

Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

Primary button Secondary