

# **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.

EC2 > Instances > Launch an instance

### Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

#### Name and tags Info

Name

 [Add additional tags](#)

#### ▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

**Quick Start**

macOS

**Ubuntu**

Windows

Red Hat

SUSE Linux

Mac

**ubuntu**

Microsoft

Red Hat

SUSE

[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

#### ▼ Summary

Number of instances Info

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more  
ami-0c7217cdde317cfc

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 I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Launch instance](#) [Review commands](#)

### 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.

### Create key pair

×

Key pair name

Key pairs allow you to connect to your instance securely.

The name can include upto 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

### 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.

**Network settings** [Info](#) [Edit](#)

Network [Info](#)  
vpc-07af334504c75bebc

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

Common security groups [Info](#)  
Select security groups

default sg-0dc903886a12e1f38 X  
VPC: vpc-07af334504c75bebc

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

**Configure storage** [Info](#) [Advanced](#)

1x 8 GiB gp2 Root volume (Not encrypted)

1

Software Image (AMI)  
Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-0c7217cdde317c7ec

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
default

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](#)

### 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

Instances (1/3) [info](#)

[Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
lampsetup	i-0a5ef1b5dd1ee670d	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1b

Instance: i-0a5ef1b5dd1ee670d (lampsetup)

[Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

▼ Security details

IAM Role: -

Owner ID: 999403824284

Launch time: Fri Jan 05 2024 15:53:38 GMT-0800 (Pacific Standard Time)

Security groups: [sg-0dc903886a12e1f38 \(default\)](#)

▼ Inbound rules

[EC2](#) > [Security Groups](#) > sg-0dc903886a12e1f38 - default

sg-0dc903886a12e1f38 - default [Actions](#)

Details

Security group name default	Security group ID sg-0dc903886a12e1f38	Description default VPC security group	VPC ID <a href="#">vpc-07af334504c75bebc</a>
Owner 999403824284	Inbound rules count 4 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) [Outbound rules](#) [Tags](#)

Inbound rules (4) [Refresh](#) [Manage tags](#) [Edit inbound rules](#)

Name	Security group rule...	IP version	Type	Protocol	Port range

Inbound rules [info](#)

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sg-08b0aea9e5d18fb75	All traffic	All	All	Custom	
-	SSH	TCP	22	My IP	
-	HTTP	TCP	80	Anyw...	
-	HTTPS	TCP	443	Anyw...	

[Add rule](#)

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

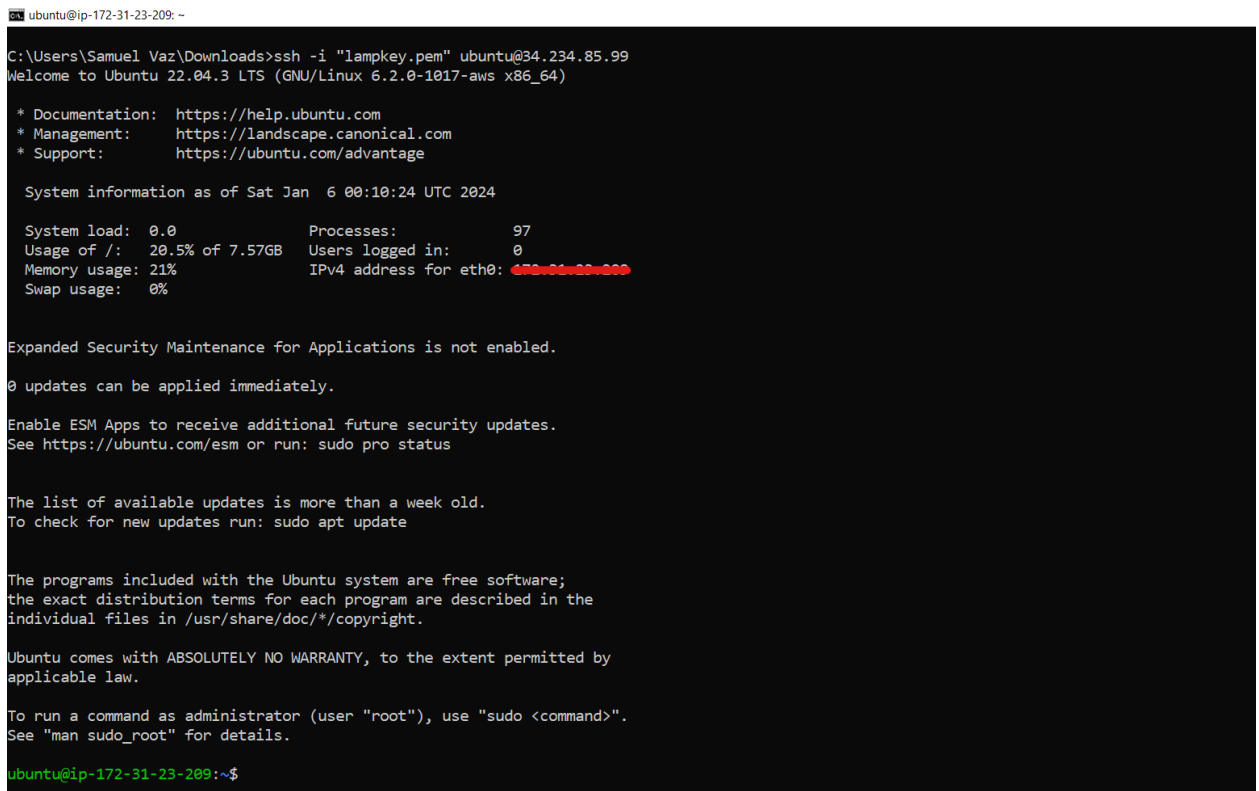
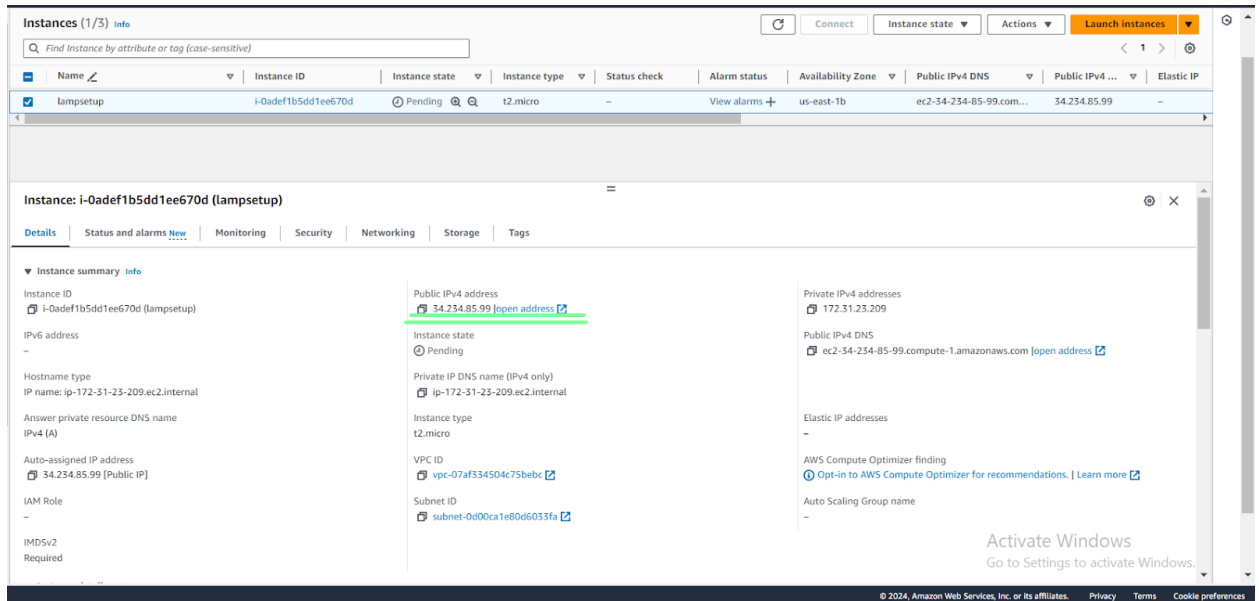
[Cancel](#) [Preview changes](#) [Save rules](#)

### 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***

```
ubuntu@ip-172-31-23-209:~$ 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-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [188 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2 [92.8 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.2 [11.3 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.2 [9170 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1build1 [140 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.7 [1346 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.7 [165 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.7 [88.8 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mailcap all 3.70+nmu1ubuntu1 [23.8 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mime-support all 3.66 [3696 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.7 [97.8 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bzip2 amd64 1.0.8-5build1 [34.8 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ssl-cert all 1.1.2 [17.4 kB]
Fetched 2139 kB in 0s (22.8 MB/s)
Preconfiguring packages ...
```

#### 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
```

```

ubuntu@ip-172-31-23-209:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbcbi-fast-perl libbcbi-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-
  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 tinyca
The following NEW packages will be installed:
  libbcbi-fast-perl libbcbi-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-
  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 32 not upgraded.
Need to get 29.6 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7212 B]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.35-0ubuntu0.22.04.1 [2682 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-8.0 amd64 8.0.35-0ubuntu0.22.04.1 [22.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-pthreads-2.1-7 amd64 2.1.12-stable-1build3 [7642 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libmecab2 amd64 0.996-14build9 [199 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libprotobuf-lite23 amd64 3.12.4-1ubuntu7.22.04.1 [209 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-core-8.0 amd64 8.0.35-0ubuntu0.22.04.1 [17.6 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-8.0 amd64 8.0.35-0ubuntu0.22.04.1 [1438 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-tagset-perl all 3.20-4 [12.5 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liburi-perl all 5.10-1 [78.8 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-parser-perl amd64 3.76-1build2 [88.4 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libbcbi-pm-perl all 4.54-1 [188 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi0ldbl amd64 2.4.2-2build2 [28.0 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi-perl amd64 0.82+ds-1build1 [22.8 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libbcbi-fast-perl all 1:2.15-1 [10.5 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libclone-perl amd64 0.45-1build3 [11.0 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libencode-locale-perl all 1.05-1.1 [11.8 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi-bin amd64 2.4.2-2build2 [11.2 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-template-perl all 2.97-1.1 [59.1 kB]

```

- 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***

```

ubuntu@ip-172-31-23-209:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

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: y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

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) : 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) : n
... 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!
ubuntu@ip-172-31-23-209:~$

```

- 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
```

```
ubuntu@ip-172-31-23-209:~$ sudo apt install php libapache2-mod-php php-mysql
mysql> \q
Bye
ubuntu@ip-172-31-23-209:~$ 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
```

- **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>
```

```
ubuntu@ip-172-31-23-209: ~
GNU nano 6.2 /etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

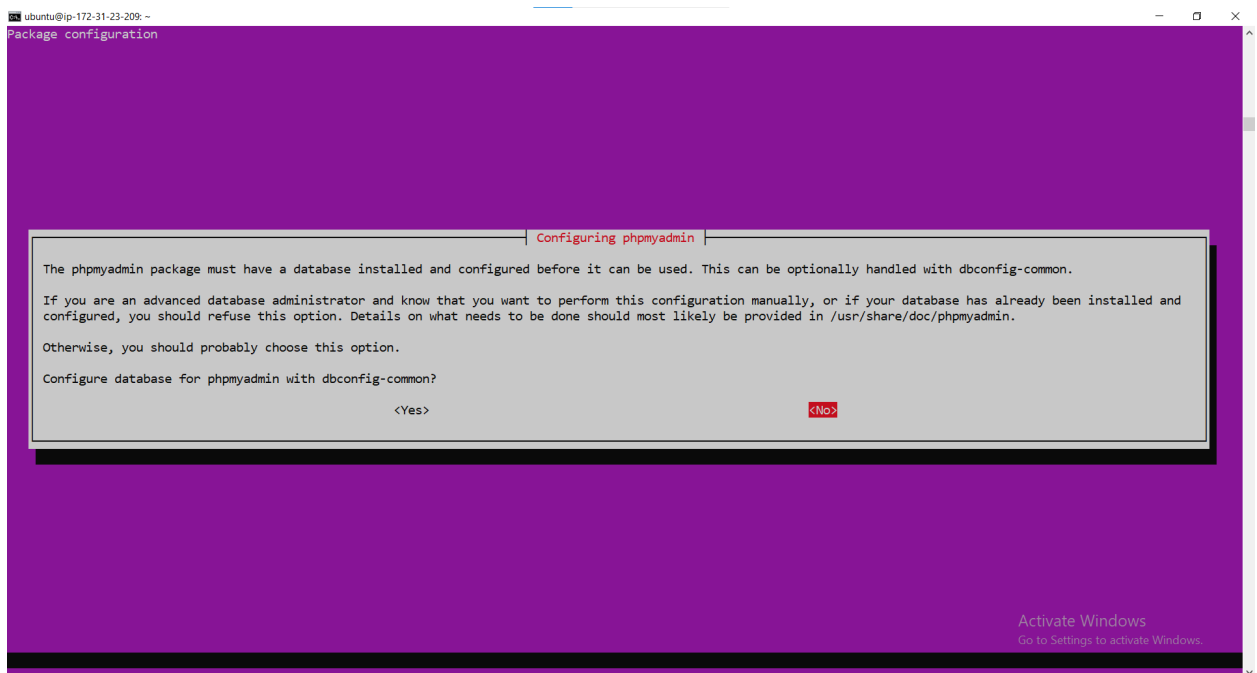
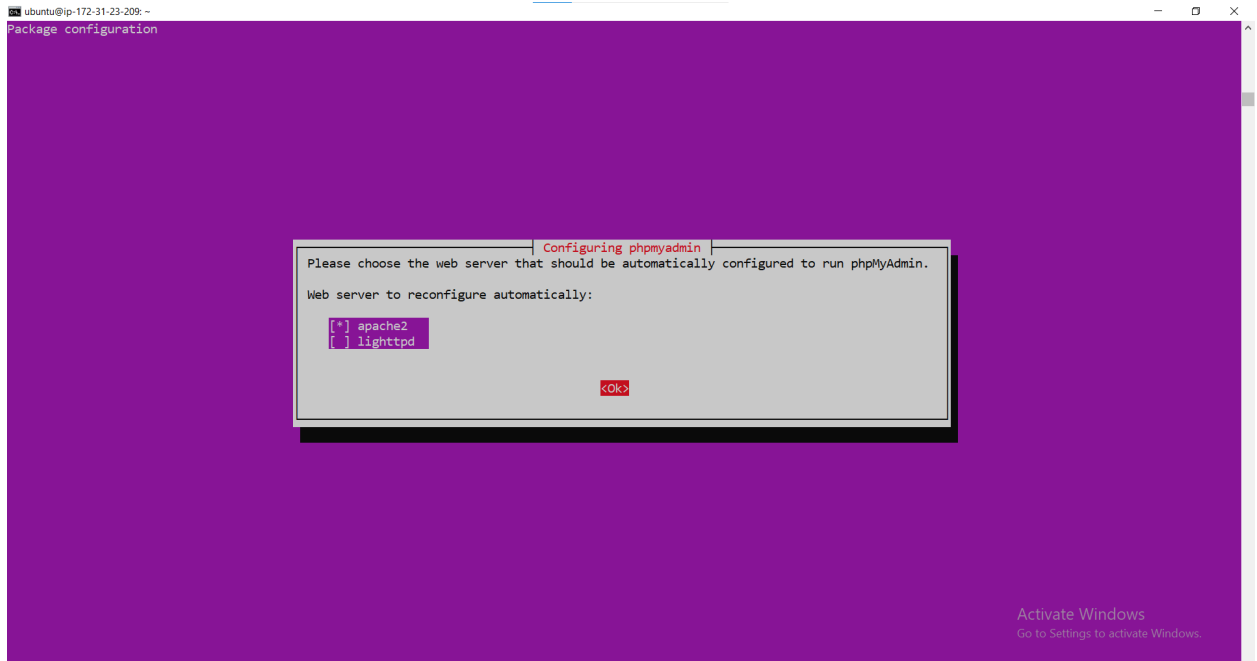
[ Read 5 lines ]
Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-B To Bracket M-] Previous
Exit Read File Replace Paste Justify Go To Line M-E Redo M-C Copy M-? Where Was M-N Next
```

## 4.6 Installing PHPMYADMIN:

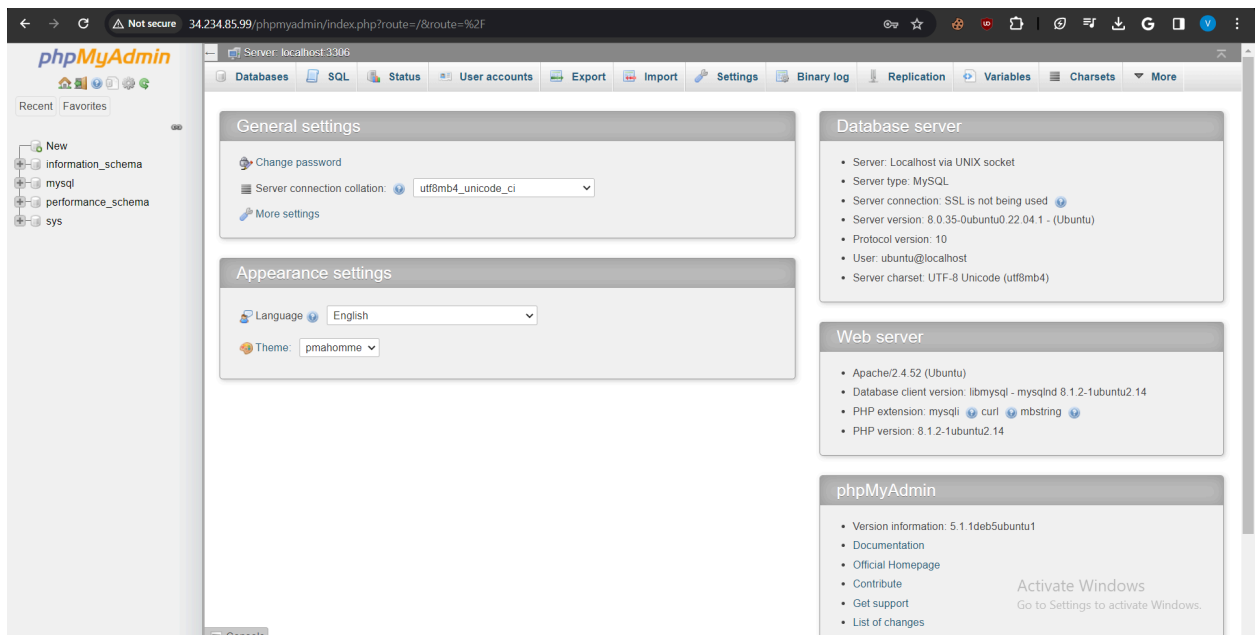
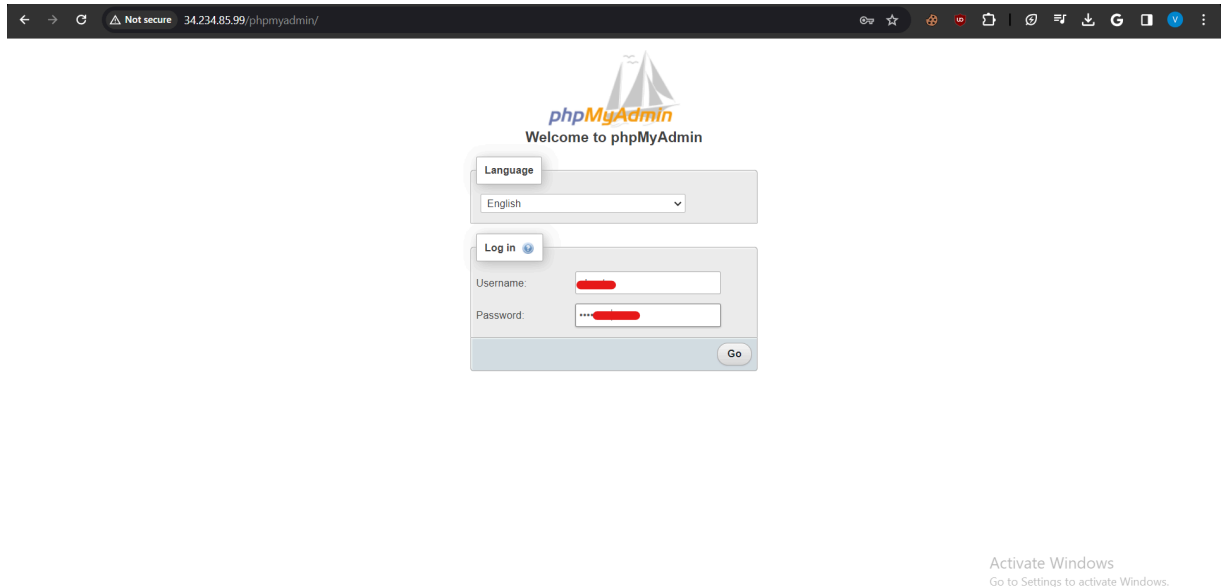
- Installing PHPMYADMIN

>> ***sudo apt install phpmyadmin***

```
ubuntu@ip-172-31-23-209:~$ sudo nano /etc/apache2/mods-enabled/dir.conf
ubuntu@ip-172-31-23-209:~$ sudo apt install phpmyadmin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core icc-profiles-free javascript-common libdeflate0 libfontconfig1 libgd3 libjpeg0 libjpeg-turbo8
libjpeg8 libjs-bootstrap4 libjs-codemirror libjs-jquery libjs-jquery-mousewheel libjs-jquery-timepicker libjs-jquery-ui libjs-popover.js libjs-sizzle libjs-sphinxdoc
libjs-underscore libonig5 libtiff5 libwebp7 libxpm4 libzip4 node-jquery php-bz2 php-curl php-gd php-google-recaptcha php-json php-mariadb-mysql-kbs php-mbstring
php-nikic-fast-route php-phpmyadmin-motranslator php-phpmyadmin-shapefile php-phpmyadmin-sql-parser php-phpseclib php-psr-cache php-psr-container php-psr-log
php-symfony-cache php-symfony-cache-contracts php-symfony-config php-symfony-dependency-injection php-symfony-deprecation-contracts php-symfony-expression-language
php-symfony-file-system php-symfony-polyfill-php80 php-symfony-polyfill-php81 php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twig
php-twig-i18n-extension php-xml php-zip php8.1-bz2 php8.1-curl php8.1-gd php8.1-mbstring php8.1-xml php8.1-zip
Suggested packages:
libgd-tools libjs-requirejs libjs-jquery-ui-docs php-dbase php-libsodium php-mcrypt php-gmp php-symfony-yaml php-symfony-finder php-symfony-proxy-manager-bridge
php-imagick php-twig-doc php-recode www-browser php-gd2 php-pragmarx-google2fa-qrcode php-samyoul-u2f-php-server
Recommended packages:
php-mcrypt
The following NEW packages will be installed:
dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core icc-profiles-free javascript-common libdeflate0 libfontconfig1 libgd3 libjpeg0 libjpeg-turbo8
libjpeg8 libjs-bootstrap4 libjs-codemirror libjs-jquery libjs-jquery-mousewheel libjs-jquery-timepicker libjs-jquery-ui libjs-popover.js libjs-sizzle libjs-sphinxdoc
libjs-underscore libonig5 libtiff5 libwebp7 libxpm4 libzip4 node-jquery php-bz2 php-curl php-gd php-google-recaptcha php-json php-mariadb-mysql-kbs php-mbstring
php-nikic-fast-route php-phpmyadmin-motranslator php-phpmyadmin-shapefile php-phpmyadmin-sql-parser php-phpseclib php-psr-cache php-psr-container php-psr-log
php-symfony-cache php-symfony-cache-contracts php-symfony-config php-symfony-dependency-injection php-symfony-deprecation-contracts php-symfony-expression-language
php-symfony-file-system php-symfony-polyfill-php80 php-symfony-polyfill-php81 php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twig
php-twig-i18n-extension php-xml php-zip php8.1-bz2 php8.1-curl php8.1-gd php8.1-mbstring php8.1-xml php8.1-zip phpmyadmin
0 upgraded, 66 newly installed, 0 to remove and 32 not upgraded.
Need to get 20.0 MB of archives.
After this operation, 89.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 dbconfig-common all 2.0.21 [597 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 dbconfig-mysql all 2.0.21 [938 B]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-core all 2.37-2build1 [1041 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fontconfig-config all 2.13.1-4.2ubuntu5 [29.1 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 icc-profiles-free all 2.0.14dfsg-1.1 [214 kB]
```



- 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
```



```
ubuntu@ip-172-31-23-209: ~
GNU nano 6.2 /etc/apache2/apache2.conf
include /etc/phpmyadmin/apache.conf
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.4/ for detailed information about
# the directives and /usr/share/doc/apache2/README.Debian about Debian specific
# hints.
#
# Summary of how the Apache 2 configuration works in Debian:
# The Apache 2 web server configuration in Debian is quite different to
# upstream's suggested way to configure the web server. This is because Debian's
# default Apache2 installation attempts to make adding and removing modules,
# virtual hosts, and extra configuration directives as flexible as possible, in
# order to make automating the changes and administering the server as easy as
# possible.
#
# It is split into several files forming the configuration hierarchy outlined
# below, all located in the /etc/apache2/ directory:
#
# /etc/apache2/
# |-- apache2.conf
# |   |-- ports.conf
# |   |-- mods-enabled
# |       |-- *.load
# |       |-- *.conf
# |   |-- conf-enabled
# |       |-- *.conf
# |   |-- sites-enabled
# |       |-- *.conf
#
# * apache2.conf is the main configuration file (this file). It puts the pieces
# together by including all remaining configuration files when starting the
# web server.
#
# * ports.conf is always included from the main configuration file. It is
# supposed to determine listening ports for incoming connections which can be
#
Read 228 lines
Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-J To Bracket M-C Previous
Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy M-W Where Was M-W Next
```

## 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
```

```
ubuntu@ip-172-31-23-209:~$ sudo nano /var/www/html/index.php
ubuntu@ip-172-31-23-209:~$
```

```
ubuntu@ip-172-31-23-209: ~
GNU nano 6.2 /var/www/html/index.php
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Bootstrap demo</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLRA9TneNe0a7RxnztjcdScmG1MxSR1GAsxE" crossorigin="anonymous">
</head>
<body>
  <div class="px-4 pt-5 my-5 text-center border-bottom">
    <h1 class="display-4 fw-bold text-body-emphasis">Centered screenshot</h1>
    <div class="col-lg-6 mx-auto">
      <p class="lead mb-4">Quickly design and customize responsive mobile-first sites with Bootstrap, the world's most popular front-end open source toolkit, featuring Sass, auto-prefixing, HTML5 validation, and much more.</p>
      <div class="d-grid gap-2 d-sm-flex justify-content-sm-center mb-5">
        <button type="button" class="btn btn-primary btn-lg px-4 me-sm-3">Primary button</button>
        <button type="button" class="btn btn-outline-secondary btn-lg px-4">Secondary</button>
      </div>
    </div>
    <div class="overflow-hidden" style="max-height: 30vh;">
      <div class="container px-5">
        
      </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kDrMNeT87bh95OGNyZPhcTNxj1Nw7RuBCsyN/o0jlpeV80yC" crossorigin="anonymous"></script>
  </body>
</html>
Read 27 lines
Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark M-J To Bracket M-C Previous
Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy M-W Where Was M-W Next
```

Final Website:

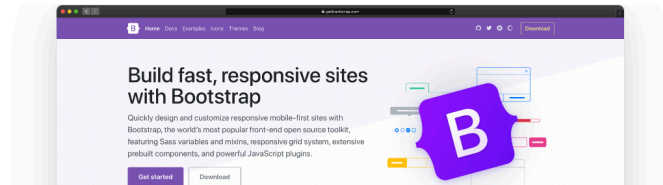


# Centered screenshot

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



Activate Windows  
Go to Settings to activate Windows.