

Step-by-Step WordPress Deployment.

Introduction

This project provides a comprehensive guide to deploying a WordPress website on a server using a web server **(Apache)** and a database **(MariaDB)**. It walks through environment setup, installation of required dependencies, WordPress configuration, and final deployment steps to make the site publicly accessible. Following this guide ensures a secure, scalable, and production-ready WordPress deployment.

Feature

- **Dynamic Website** – WordPress provides a fully functional CMS for blogs, portfolios, or business websites.
- **Database-Driven** – Uses MySQL/MariaDB to store content, user data, and settings.
- **Theme and Plugin Support** – Easily customize the site with themes and plugins.
- **Scalable** – Supports multiple users and high traffic when deployed on a proper server.

Prerequisites

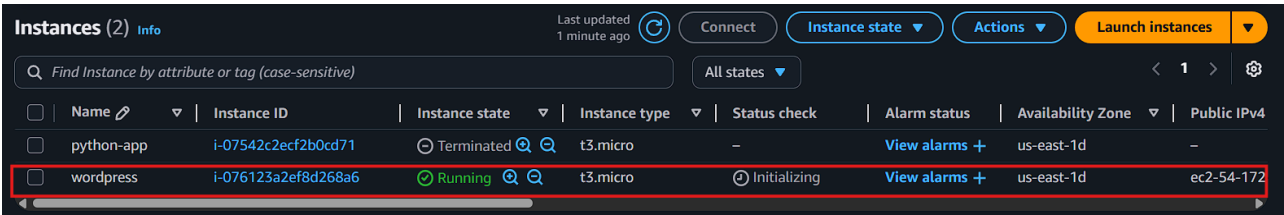
Before deploying the WordPress website, ensure the following are installed and configured:

- **Linux Server** – Amazon Linux
- **Web Server** – Apache installed and running.
- **PHP** – Along with required PHP extensions.
- **Database** – MariaDB for storing WordPress data.

Steps for Delpoyment

Step 1: Launch EC2 instance and Establishing a secure connection to your EC2 instance

1. Launch instance



Instances (2) Info								
Last updated 1 minute ago								
Find Instance by attribute or tag (case-sensitive)								
All states								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	python-app	i-07542c2ecf2b0cd71	Terminated	t3.micro	-	View alarms +	us-east-1d	-
<input type="checkbox"/>	wordpress	i-076123a2ef8d268a6	Running	t3.micro	Initializing	View alarms +	us-east-1d	ec2-54-172

2. Copy the SSH command


EC2 Instance Connect

Session Manager


SSH client


EC2 serial console

Instance ID


 i-076123a2ef8d268a6 (wordpress)


1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

 `chmod 400 "key.pem"`
4. Connect to your instance using its Public DNS:

 `ec2-54-172-145-151.compute-1.amazonaws.com`

Example:

 `ssh -i "key.pem" ec2-user@ec2-54-172-145-151.compute-1.amazonaws.com`

 **Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the

3. Paste command in Git bash

```

HP@LAPTOP-HFSFVC80 MINGW64 /d/dhanashri_workspace/ssh key
$ ssh -i "key.pem" ec2-user@ec2-54-172-145-151.compute-1.amazonaws.com
The authenticity of host 'ec2-54-172-145-151.compute-1.amazonaws.com (54.172.145.151)' can't be
ED25519 key fingerprint is SHA256:1lVw92fgAu7m4Jh001DQsvNgesWJXZ2IugxypXfs2h4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-172-145-151.compute-1.amazonaws.com' (ED25519) to
#_
~\##### Amazon Linux 2023
~~~\#####\
~~~\###|
~~~\#/
~~~V~'-'>
~~~~
~~~~-.
~~~\_/
~/m/'

```

Step 2: Automating LAMP Stack Setup on AWS EC2

1. Create a LAMP.sh file

```
sudo vim LAMP.sh
```

```

ec2-user@ip-172-31-23-131:~
[ec2-user@ip-172-31-23-131 ~]$ sudo vim LAMP.sh
[ec2-user@ip-172-31-23-131 ~]$

```

2. Insert the code for installing apache, mysql and php

```

sudo yum update
sudo yum install httpd mariadb105-server php -y

```

```
sudo systemctl start httpd mariadb php-fpm
sudo systemctl enable httpd mariadb php-fpm
```

ec2-user@ip-172-31-23-131:~

```
sudo yum update
sudo yum install httpd mariadb105-server php -y
sudo systemctl start httpd mariadb php-fpm
sudo systemctl enable httpd mariadb php-fpm
~
~
```

3. Run the file

ec2-user@ip-172-31-23-131:~

```
[ec2-user@ip-172-31-23-131 ~]$ sudo bash LAMP.sh
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:00:01 ago on Sun Sep  7 08:28:
Dependencies resolved.
=====
Package                                     Architecture
=====
Installing:
```

Step 3: Download and Configure WordPress

```
# Download WordPress
sudo wget https://wordpress.org/latest.tar.gz
# Extract the archive
sudo tar -xvzf latest.tar.gz
```

```

ec2-user@ip-172-31-23-131:/var/www/html
[ec2-user@ip-172-31-23-131 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-23-131 html]$ sudo wget https://wordpress.org/latest.tar.gz
--2025-09-07 08:31:44-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26925441 (26M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz                               100%[=====]
2025-09-07 08:31:45 (81.2 MB/s) - 'latest.tar.gz' saved [26925441/26925441]

[ec2-user@ip-172-31-23-131 html]$ ls
latest.tar.gz
[ec2-user@ip-172-31-23-131 html]$
[ec2-user@ip-172-31-23-131 html]$ sudo tar -xvzf latest.tar.gz
wordpress/
wordpress/index.php
wordpress/license.txt
wordpress/readme.html
wordpress/wp-activate.php
wordpress/wp-admin/
wordpress/wp-admin/about.php
wordpress/wp-admin/admin-ajax.php

```

Step 4: Remove latest.tar.gz

```

sudo rm -rf latest.tar.gz
ls

```

```

[ec2-user@ip-172-31-23-131 html]$ ls
latest.tar.gz  wordpress
[ec2-user@ip-172-31-23-131 html]$ sudo rm -rf latest.tar.gz
[ec2-user@ip-172-31-23-131 html]$ ls
wordpress

```

Step 5: Go to the wordpress folder

```

cd wordpress/

```

```

[ec2-user@ip-172-31-23-131 html]$ cd wordpress/
[ec2-user@ip-172-31-23-131 wordpress]$ ls
index.php      wp-activate.php      wp-comments-post.php  wp-cron.php          wp-load.php          wp-settings.php      xmlrpc.php
license.txt    wp-admin             wp-config-sample.php  wp-includes          wp-login.php         wp-signup.php
readme.html    wp-blog-header.php   wp-content            wp-links-opml.php    wp-mail.php          wp-trackback.php
[ec2-user@ip-172-31-23-131 wordpress]$

```

Step 6: Create WordPress Database

1. Generate the username and password.

```

sudo mysql
alter user root@localhost identified by 'root';

```

```
[ec2-user@ip-172-31-23-131 wordpress]$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.5.29-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> alter user root@localhost identified by 'root';
Query OK, 0 rows affected (0.002 sec)
```

2. Login to Mysql (mariadb105-server)

```
sudo mysql -u root -p
```

```
[ec2-user@ip-172-31-23-131 wordpress]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.5.29-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```

3. Create Database

```
# Create Database
create database wordpressdb;
# Show Database
show databases;
```

```
MariaDB [(none)]> create database wordpressdb;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| wordpressdb |
+-----+
4 rows in set (0.001 sec)
```

Step 7: Install Connector

```
sudo yum install php8.4-mysqlnd.x86_64
```

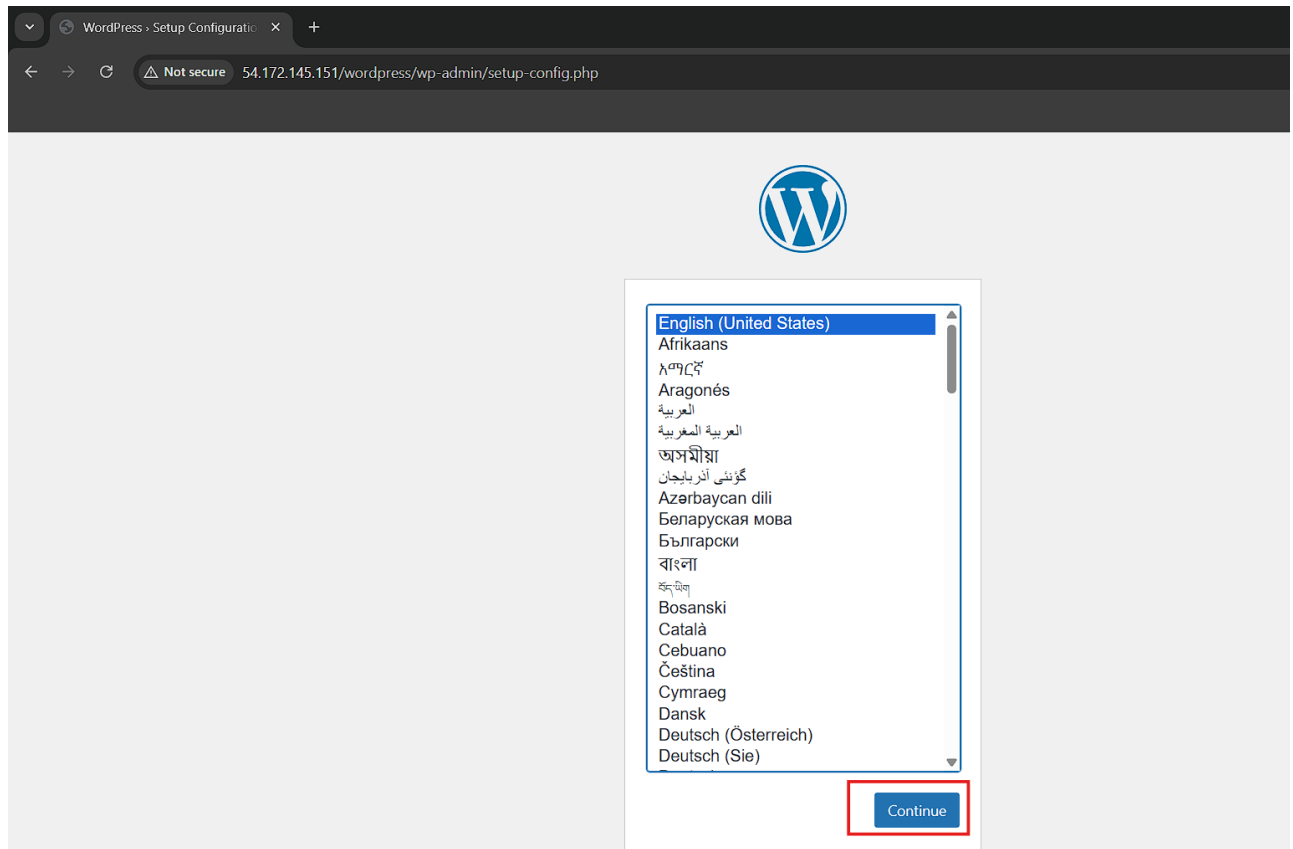
```
[ec2-user@ip-172-31-23-131 wordpress]$ sudo yum install php8.4-mysqlnd.x86_64
Last metadata expiration check: 0:17:42 ago on Sun Sep  7 08:28:43 2025.
Dependencies resolved.
=====
Package                                Architecture      Version
=====
Installing:
  php8.4-mysqlnd                        x86_64            8.4.10-1.amzn2023.0.1
Transaction Summary
=====
Install 1 Package
```

Step 8: Change ownership of the files

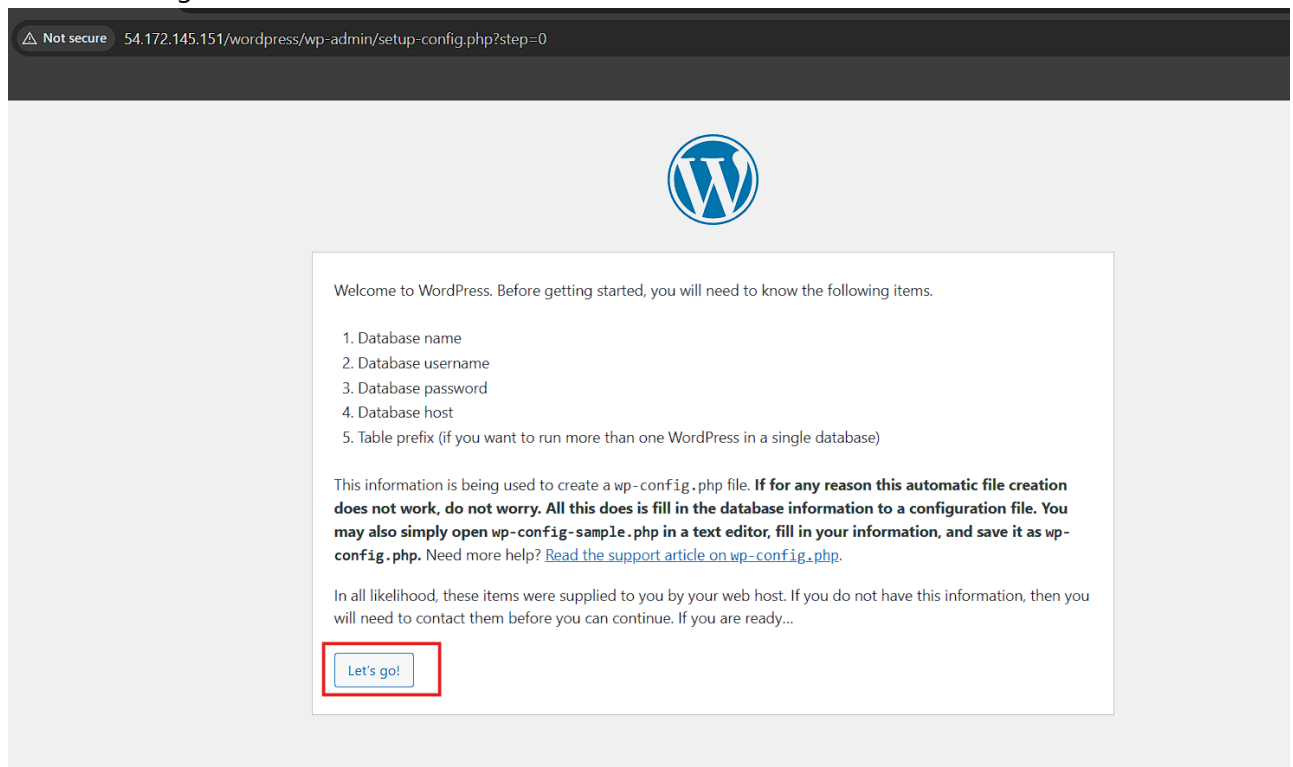
```
ec2-user@ip-172-31-23-131:/var/www/html
[ec2-user@ip-172-31-23-131 html]$ sudo chown -R apache:apache wordpress/
[ec2-user@ip-172-31-23-131 html]$ |
```

Step 9: Paste the Public IP and Paste it in any browser.

1. Click on Continue



2. Click on Let's go



3. Fill the information and click on Submit

Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name
The name of the database you want to use with WordPress.

Username
Your database username.

Password [Show](#)
Your database password.

Database Host
You should be able to get this info from your web host, if localhost does not work.

Table Prefix
If you want to run multiple WordPress installations in a single database, change this.

[Submit](#)

4. Run the Installation

All right, sparky! You've made it through this part of the installation. WordPress can now communicate with your database. If you are ready, time now to...

[Run the installation](#)


5. Fill the information and click on Install Wordpress

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title	<input type="text" value="Tech Blog"/>
Username	<input type="text" value="root"/> <small>Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.</small>
Password	<input type="password" value="root"/> Hide <div>Very weak</div> <p>Important: You will need this password to log in. Please store it in a secure location.</p>
Confirm Password	<input checked="" type="checkbox"/> Confirm use of weak password
Your Email	<input type="text" value="chaudharidhanashri2603@gmail.com"/> <small>Double-check your email address before continuing.</small>
Search engine visibility	<input checked="" type="checkbox"/> Discourage search engines from indexing this site <small>It is up to search engines to honor this request.</small>
<div>Install WordPress</div>	

6. Login to Wordpress

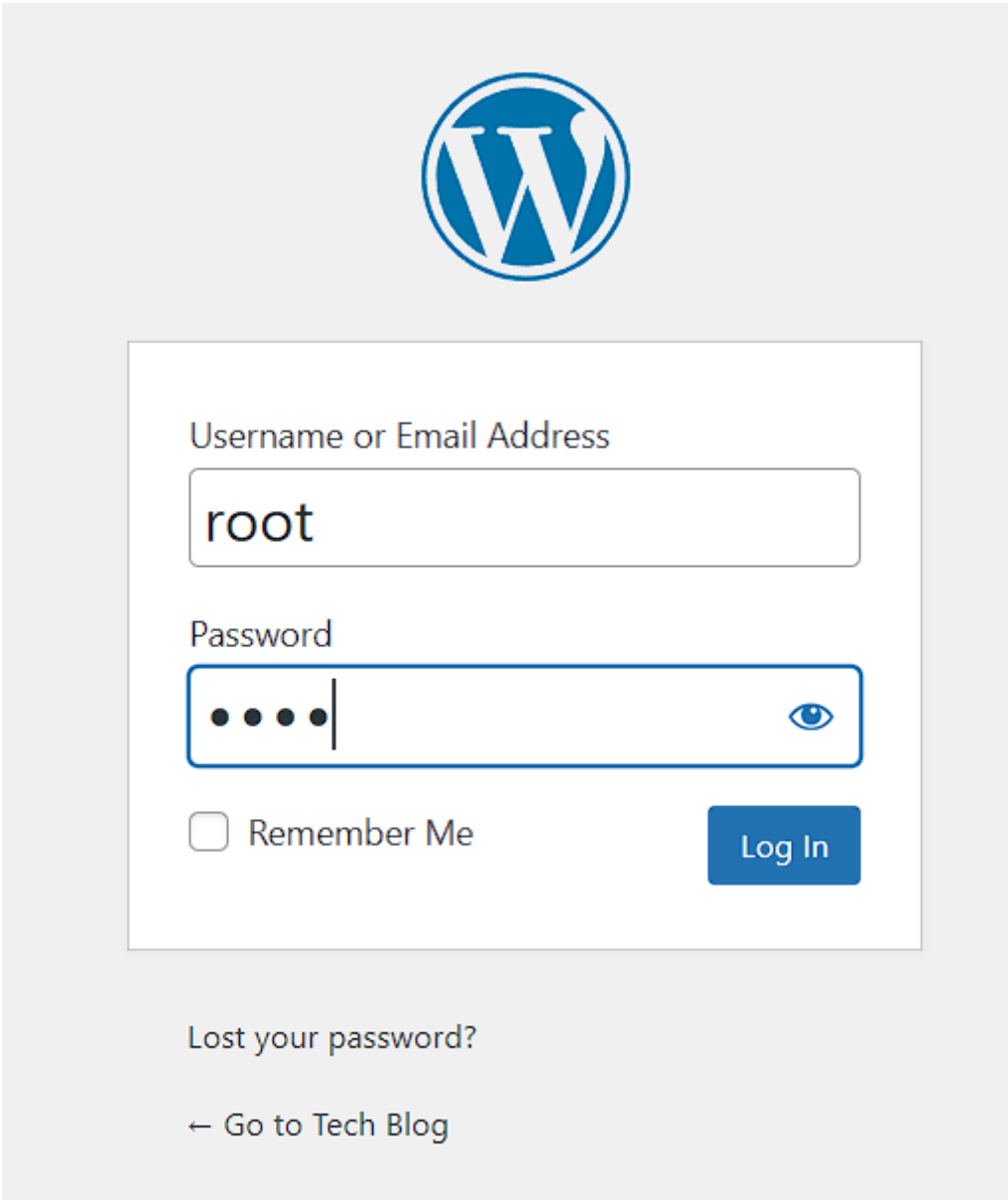


Success!

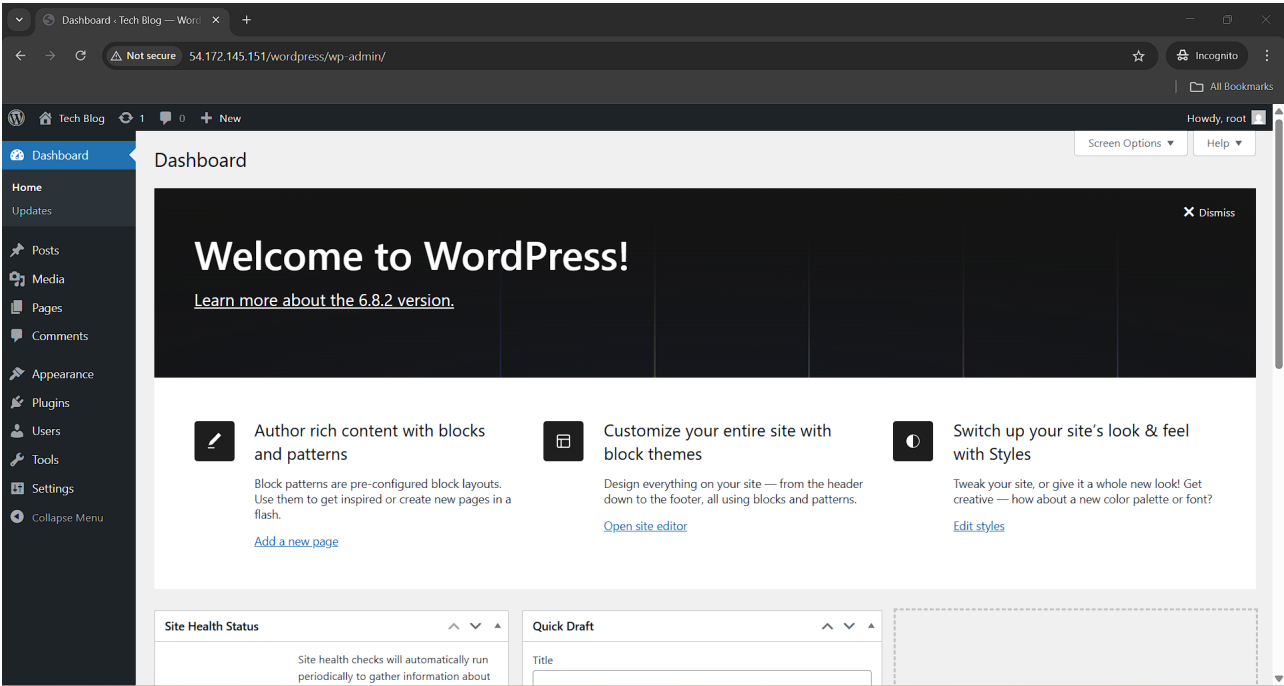
WordPress has been installed. Thank you, and enjoy!

Username	root
Password	Your chosen password.

Log In



7. Deployed Wordpress successfully



8. Table automatically added to Database

```
MariaDB [(none)]> use wordpressdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [wordpressdb]> show tables;
+-----+
| Tables_in_wordpressdb |
+-----+
| wp_commentmeta         |
| wp_comments            |
| wp_links               |
| wp_options             |
| wp_postmeta            |
| wp_posts               |
| wp_term_relationships  |
| wp_term_taxonomy       |
| wp_termmeta            |
| wp_terms               |
| wp_usermeta            |
| wp_users               |
+-----+
12 rows in set (0.000 sec)
```

Summary

This project provides a complete guide to deploying a WordPress website on a Linux server using a web server (Apache) and a database (MariaDB). It covers downloading and installing WordPress, configuring file permissions, setting up the database, and configuring the web server for production use. Following this guide ensures a secure, scalable, and fully functional WordPress site ready for public access.