

Deployment of Wordpress

Introduction

This project demonstrate the deployment of a Wordpress websites on Linux server using the LAMP stack(Linux, Apache, MariaDB, PHP). Wordpress is a popular CMS that allows users to build dynamic and customizable websites.

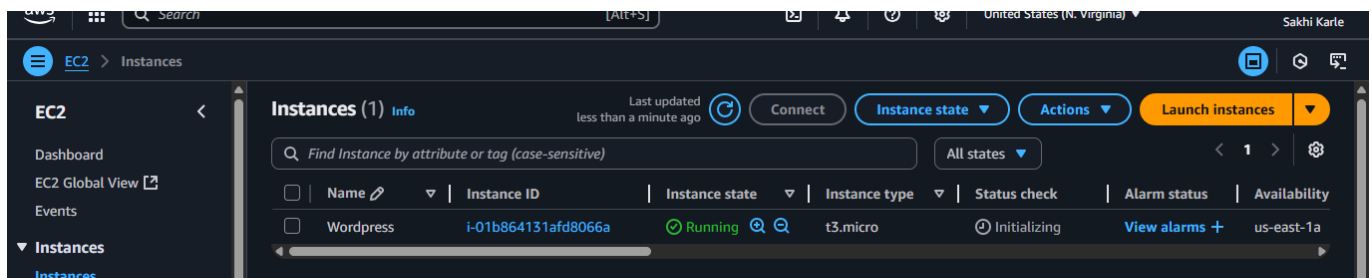
Prerequisites

Before deploying Wordpresss, ensure the following steps are installed and configured:

1. Linux Server- Amazon Linux/ Ubuntu (EC2 instance).
2. Web Server - Apache HTTP Server installed and running.
3. Database - Mariadb/MySQL for storing Wordpresss data.
4. PHP - With required extensions (php-mysql, php-fpm, etc).
5. Internet Access - To download Wordpress and dependencies.

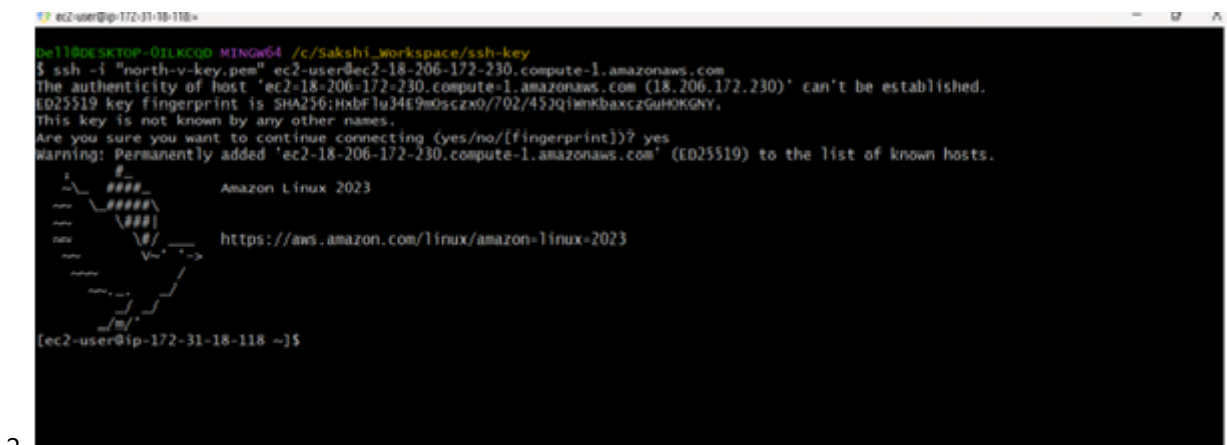
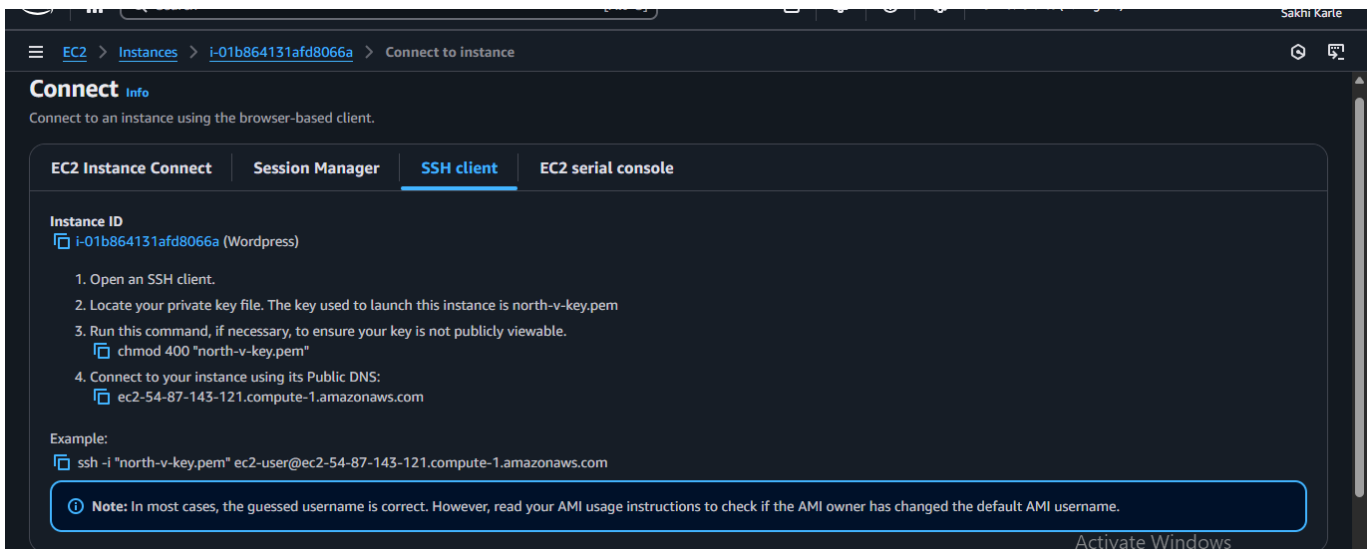
Deployment Steps

* Step 1: Launch EC2 Instance



* Step 2: Copy the ssh key and Connect to EC2 Instance

1.



2.

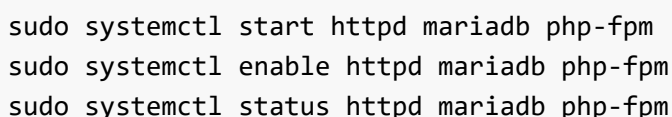
* Step 3: Install LAMP Stack with Shell Script

1. Create LAMP.sh file

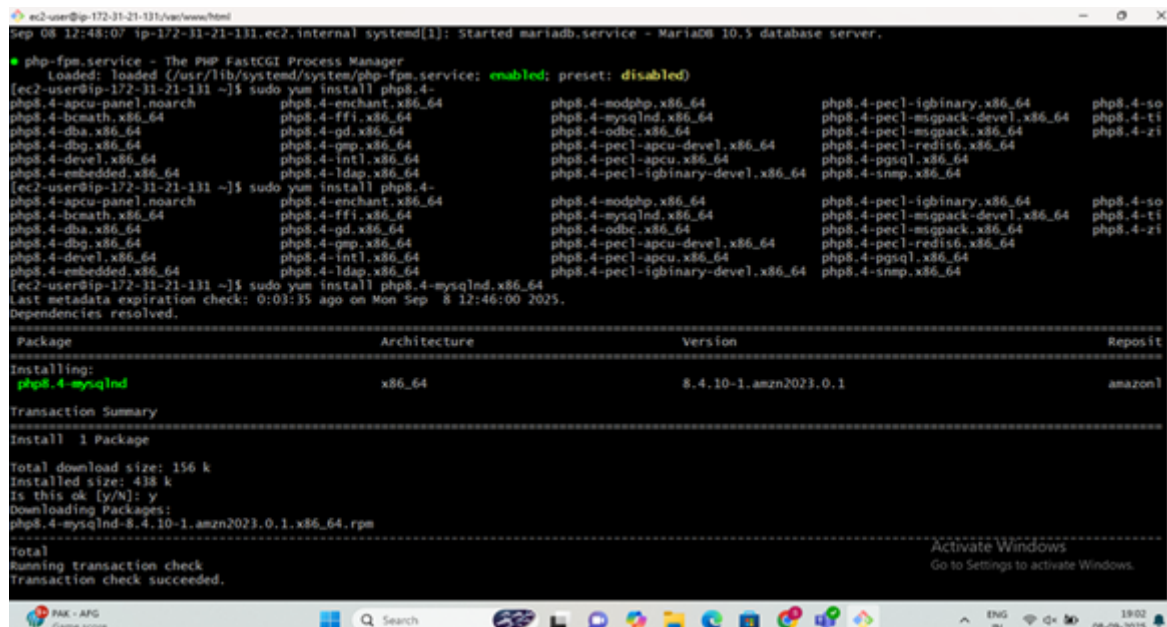
```
sudo vim LAMP.sh
```

2.Install LAMP

```
sudo yum update
```

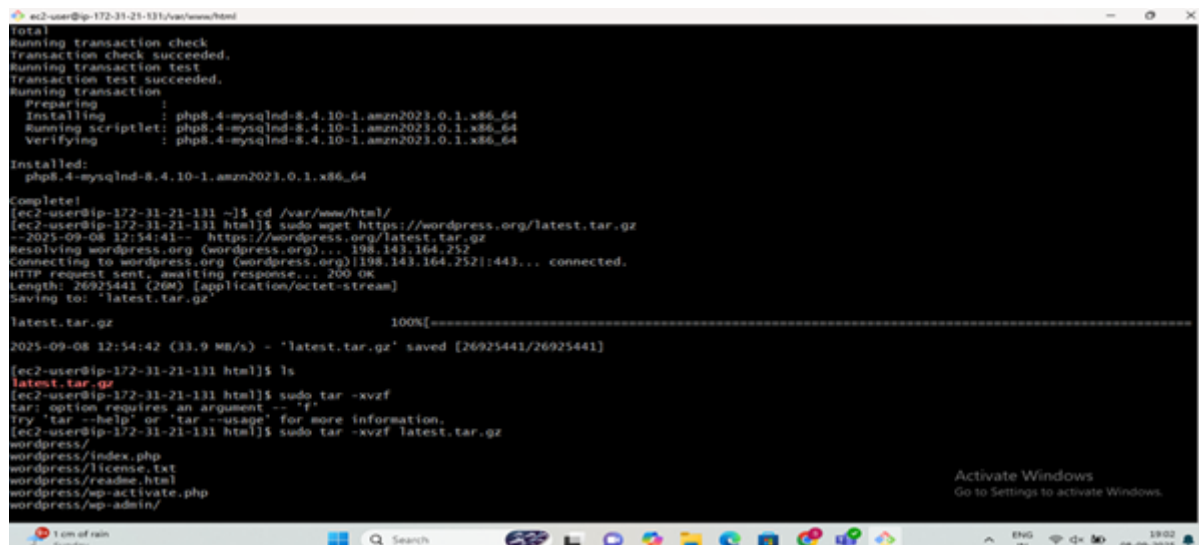


```
sudo yum install php8.4-mysqlnd.x86_64
```



Step 5: Download and Extract Wordpress

```
cd/var/www/html
```



```
sudo get https://wordpress.org/latest.tar.gz
tar -xvzf <latest.tar.gz>
```

```

ec2-user@ip-172-31-21-131:/var/www/html$ ls
wordpress/wp-includes/widgets/class-wp-widget-media-image.php
wordpress/wp-includes/widgets/class-wp-widget-media-video.php
wordpress/wp-includes/widgets/class-wp-widget-media.php
wordpress/wp-includes/widgets/class-wp-widget-meta.php
wordpress/wp-includes/widgets/class-wp-widget-pages.php
wordpress/wp-includes/widgets/class-wp-widget-recent-comments.php
wordpress/wp-includes/widgets/class-wp-widget-recent-posts.php
wordpress/wp-includes/widgets/class-wp-widget-rss.php
wordpress/wp-includes/widgets/class-wp-widget-search.php
wordpress/wp-includes/widgets/class-wp-widget-tag-cloud.php
wordpress/wp-includes/widgets/class-wp-widget-text.php
wordpress/wp-includes/widgets.php
wordpress/wp-includes/wp-db.php
wordpress/wp-includes/wp-diff.php
wordpress/wp-links-opml.php
wordpress/wp-load.php
wordpress/wp-login.php
wordpress/wp-mail.php
wordpress/wp-settings.php
wordpress/wp-signup.php
wordpress/wp-trackback.php
wordpress/xmlrpc.php
[ec2-user@ip-172-31-21-131 html]$ ls
latest.tar.gz  wordpress
[ec2-user@ip-172-31-21-131 html]$ sudo rm -rf latest.tar.gz
[ec2-user@ip-172-31-21-131 html]$ ls
wordpress
[ec2-user@ip-172-31-21-131 html]$ cd wordpress
[ec2-user@ip-172-31-21-131 wordpress]$ ls
index.php  readme.html  wp-admin  wp-comments-post.php  wp-content  wp-includes  wp-load.php  wp-mail.php  wp-signup.php
license.txt  wp-activate.php  wp-blog-header.php  wp-config-sample.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.p

```

Step 6: Create Wordpress database

1.Generate username and password

```

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

```

```

ec2-user@ip-172-31-18-118~$ sudo mysql
[ec2-user@ip-172-31-18-118 ~]$ sudo mysql
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)]> alter user root@localhost identified by 'root';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>

```

2.Login to Mysql (mariadb105-server)

```

sudo mysql -u root -p

```

```

ec2-user@ip-172-31-18-118~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
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)]> |

```

Step 7: Create Database

```

create database wordpressdb;
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 (0.002 sec)

MariaDB [(none)]> |
```

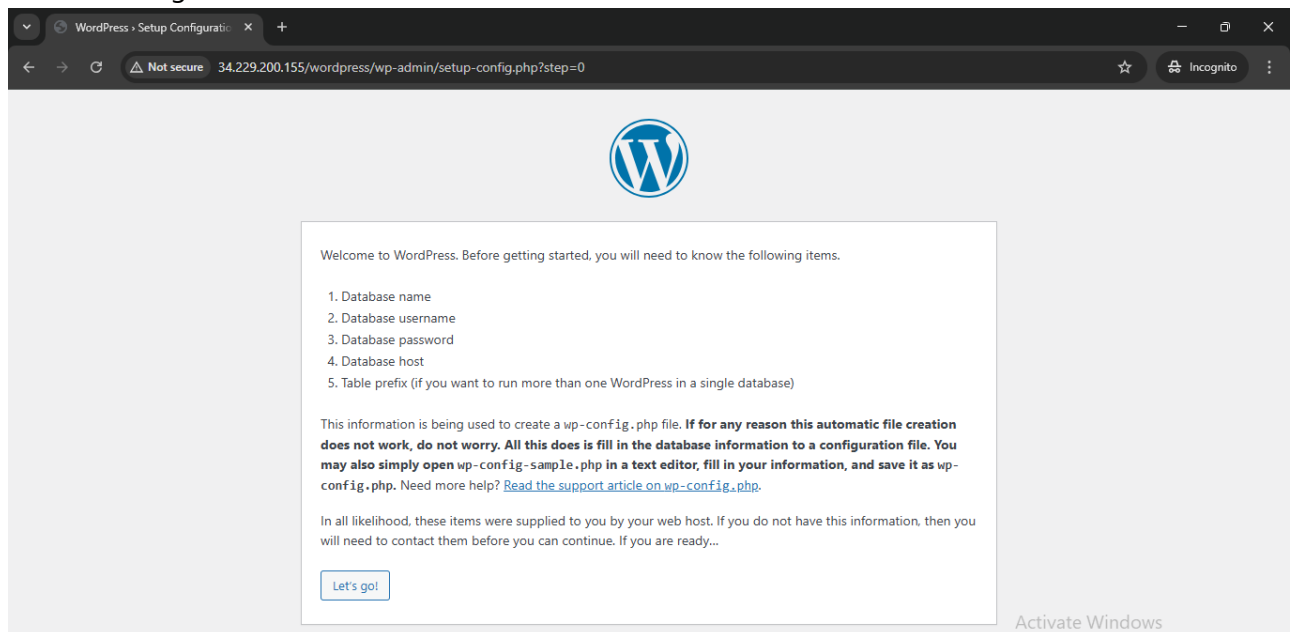
Activate Windows
Go to Settings to activate Windows.

Step 8: Grant permission to wordpress directory

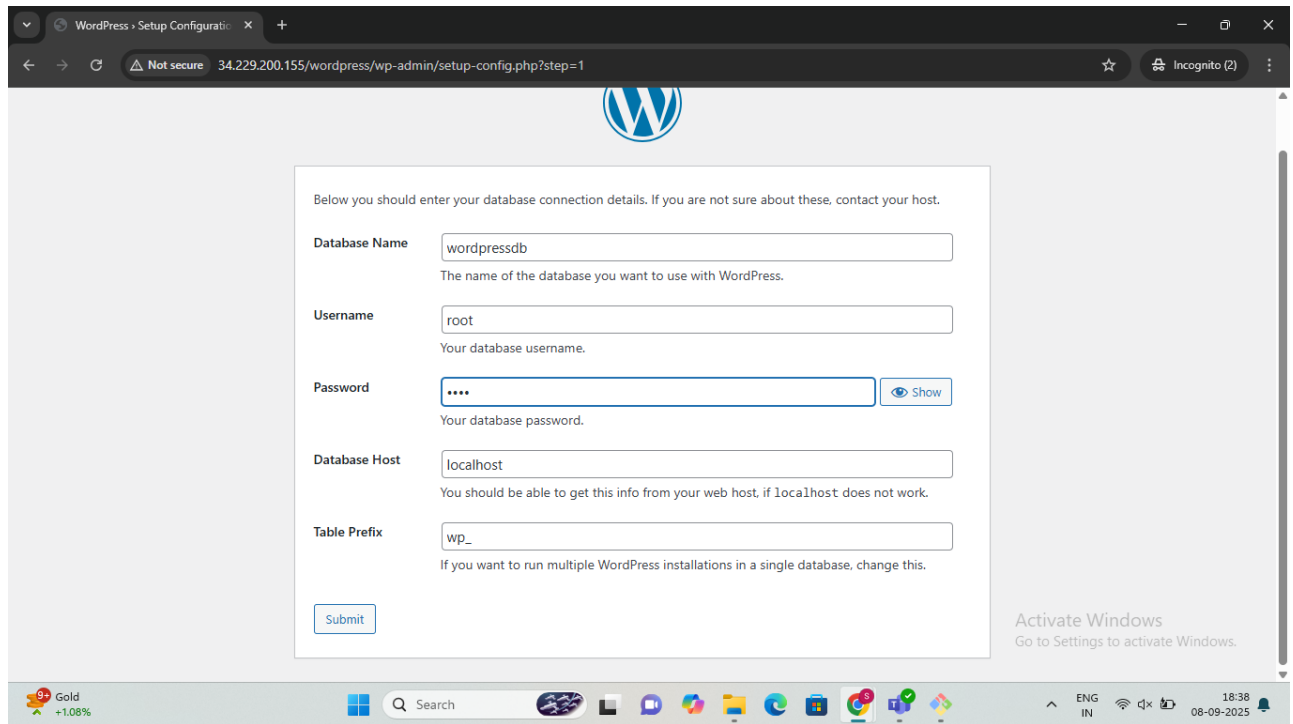
```
sudo chown -R apache:apache wordpress/
```

Step 9: Paste the public IP in any browser

1. Click on let's go



2. Fill the information and click on submit



WordPress Setup Configuration

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.

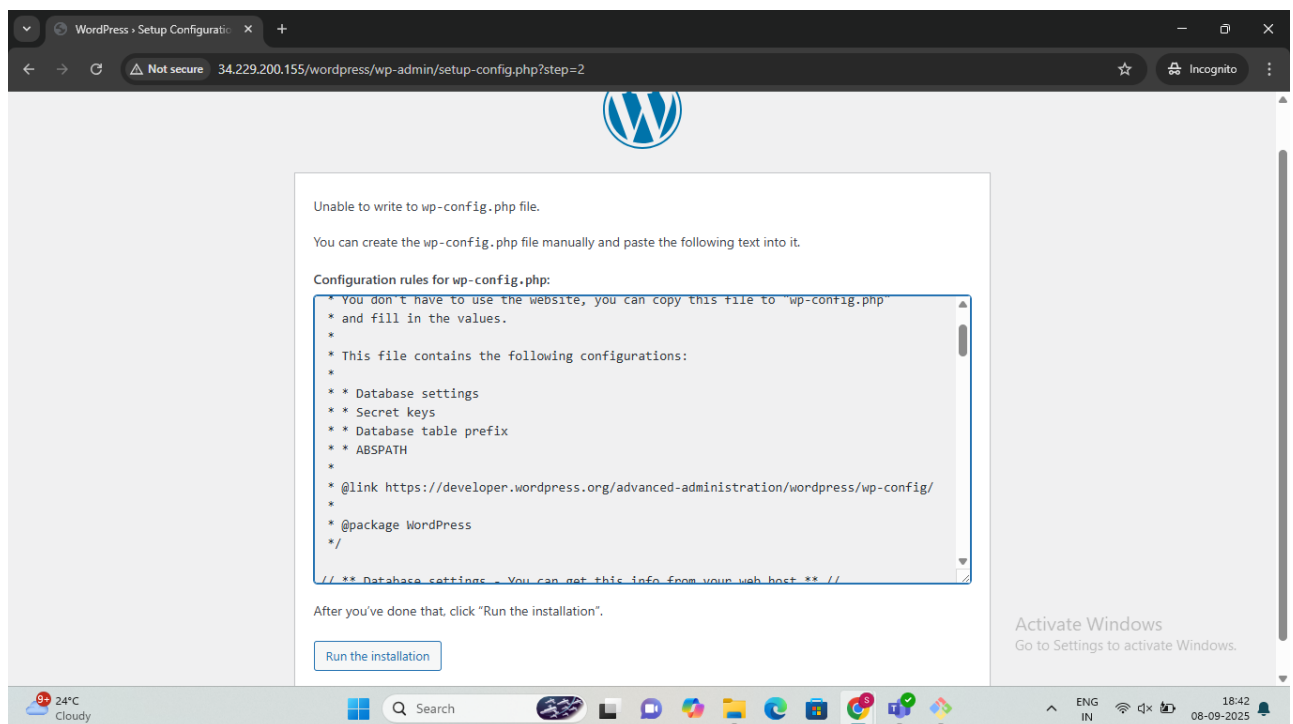
Database Host:
You should be able to get this info from your web host, if localhost does not work.

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

[Submit](#)

Activate Windows
Go to Settings to activate Windows.

3. Run and Installation



WordPress Setup Configuration

Unable to write to wp-config.php file.

You can create the wp-config.php file manually and paste the following text into it.

Configuration rules for wp-config.php:

```
* You don't have to use the website, you can copy this file to 'wp-config.php'
* and fill in the values.
*
* This file contains the following configurations:
*
* * Database settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
*
* @package WordPress
*/

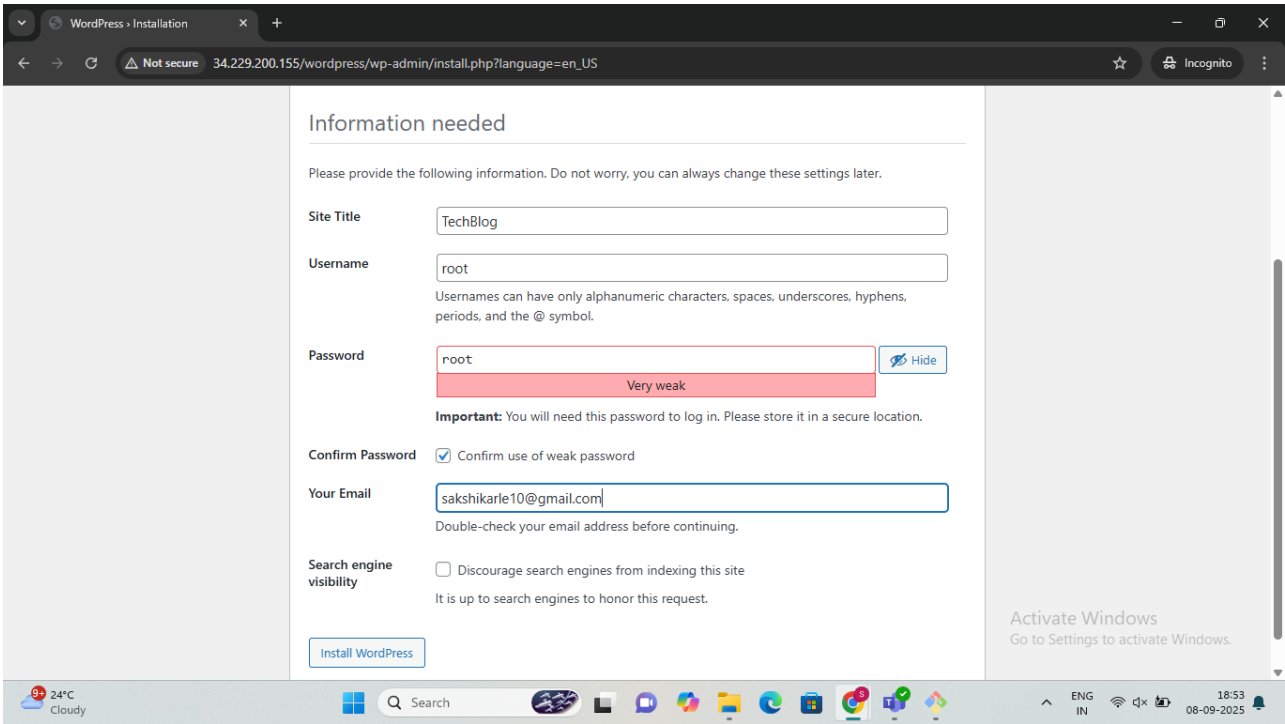
/** Database settings - You can get this info from your web host **/
```

After you've done that, click "Run the installation".

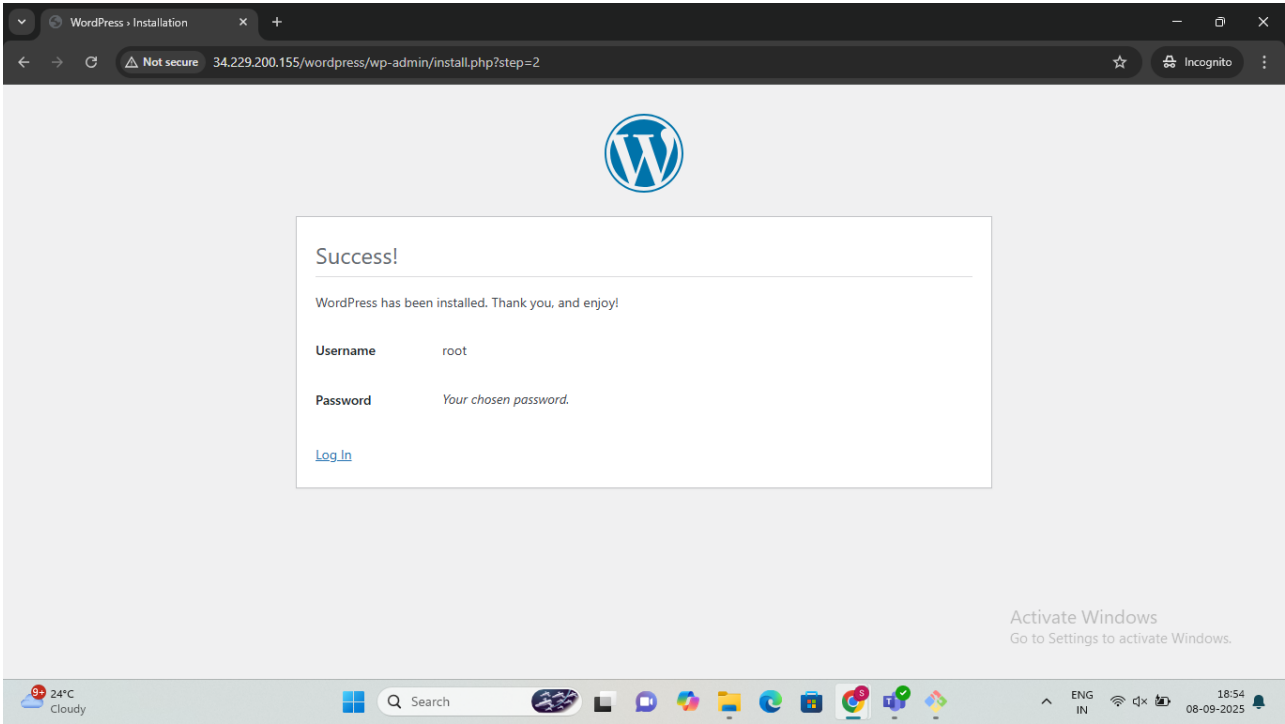
[Run the installation](#)

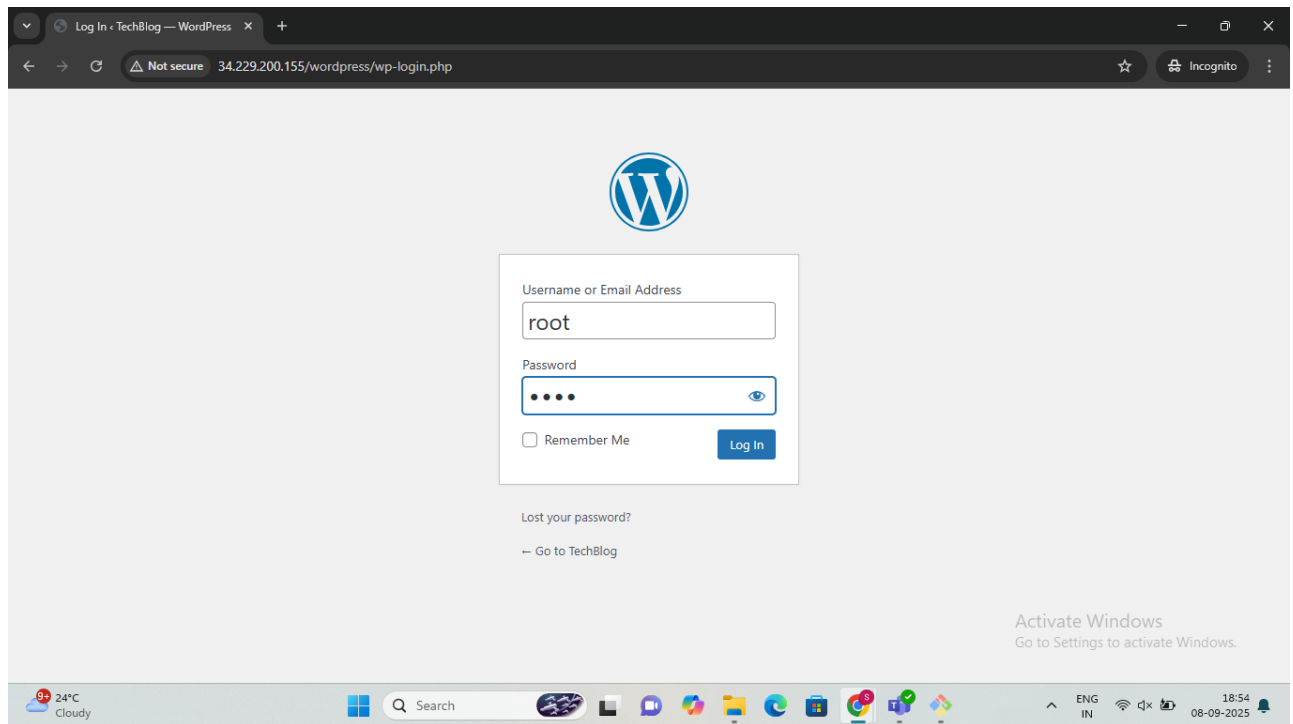
Activate Windows
Go to Settings to activate Windows.

4. Fill the information and click on install wordpress

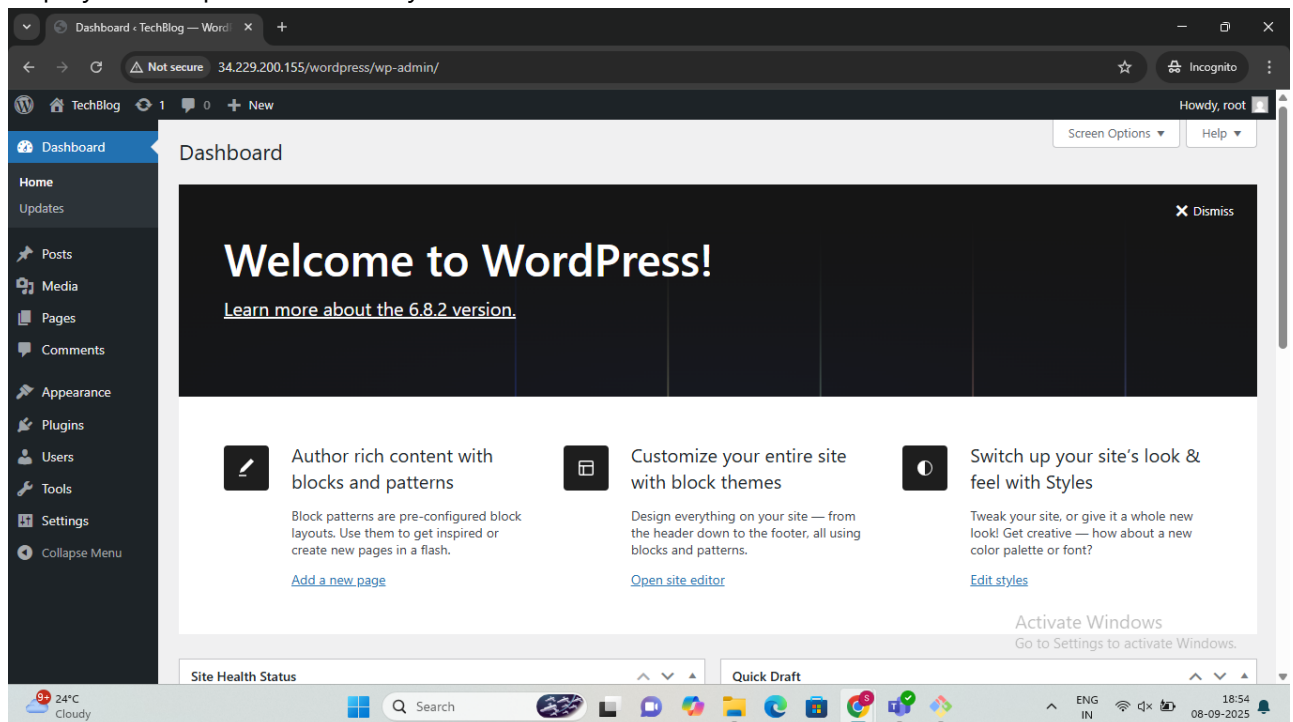


5. Login to wordpress





6. Deployed Wordpress Successfully



Summary

This project explains how to deploy a wordpress website on an Amazon Linux EC2 instance using the LAMP stack. It walks through installing Apache, PHP, and MariaDB, downloading and configuring wordpress, setting up the database, and adjusting file permissions. The result is a secure, production-ready wordpress site.