

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.

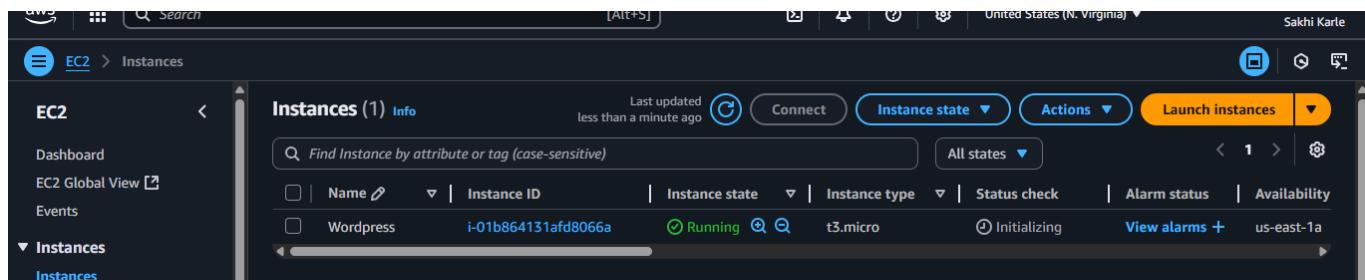
Prerequisites

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.

The screenshot shows the AWS EC2 Connect interface. At the top, there's a breadcrumb navigation: EC2 > Instances > i-01b864131af8066a > Connect to instance. Below this, a section titled "Connect" with a "Info" link is shown. A sub-section "EC2 Instance Connect" is selected, followed by "Session Manager", "SSH client" (which is currently active), and "EC2 serial console". Under "Instance ID", it shows "i-01b864131af8066a (Wordpress)". Below this, a numbered list of steps is provided:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is north-v-key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 'north-v-key.pem'
4. Connect to your instance using its Public DNS:
ec2-54-87-143-121.compute-1.amazonaws.com

Example:
ssh -i "north-v-key.pem" ec2-user@ec2-54-87-143-121.compute-1.amazonaws.com

A note in a blue-bordered box says: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username."

Activate Windows

The screenshot shows the AWS Lambda function execution interface. The log output shows the following:

```
SSH connection attempt from ec2-user@ip-172-31-18-118
[ec2-user@ip-172-31-18-118 ~]$ ssh -i "north-v-key.pem" ec2-user@ec2-18-206-172-230.compute-1.amazonaws.com
The authenticity of host 'ec2-18-206-172-230.compute-1.amazonaws.com (18.206.172.230)' can't be established.
EC2SS19 key fingerprint is SHA256:HxbfFlu34E9m0scex0/702/452QlmnKbxaczGUhOKGNY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-206-172-230.compute-1.amazonaws.com' (EC2SS19) to the list of known hosts.

[ec2-user@ip-172-31-18-118 ~]$
```

2.

* Step 3: Install LAMP Stack with Shell Script

1. Create LAMP.sh file

```
sudo vim LAMP.sh
```

2. Install LAMP

```
sudo yum update
```

```
ec2-user@ip-172-31-21-131:~$ ssh-keygen -t rsa -b 2048
ec2-user@ip-172-31-21-131:~$ ssh -i "north-v-key.pem" ec2-user@ec2-34-229-200-155.compute-1.amazonaws.com
The authenticity of host 'ec2-34-229-200-155.compute-1.amazonaws.com (34.229.200.155)' can't be established.
ECDSA key fingerprint is SHA256:7H0z13+VXx29633C6vzvRASePudfHmMNEtQH8.
Are you sure you want to continue connecting (yes/no/[Fingerprint])? yes
Warning: Permanently added 'ec2-34-229-200-155.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-21-131 ~]$ sudo yum update
Amazon Linux 2023 Kernel livepatch repository
Dependencies resolved.
Nothing to do.
[ec2-user@ip-172-31-21-131 ~]$ sudo yum install httpd -y
Last metadata expiration check: 0:00:25 ago on Mon Sep 8 12:46:00 2025.
Dependencies resolved.

Package                                         Architecture   Version          Repos
Installing:
httpd                                           x86_64         2.4.64-3.amzn2023.0.1
Installing dependencies:
apr                                             x86_64         1.6.5-1.amzn2023.0.1
apr-util                                         x86_64         1.6.6-1.amzn2023.0.1
generic-top                                     noarch        18.0.0-12.amzn2023.0.3
httpd-core                                       x86_64         2.4.64-3.amzn2023.0.1
httpd-fs-filesystem                            noarch        2.4.64-3.amzn2023.0.1
httpd-prefork                                    x86_64         2.4.64-3.amzn2023.0.1
libbrotli1                                      x86_64         1.0.9-4.amzn2023.0.2
mailcap                                         noarch        2.1.49-3.amzn2023.0.3
Activate Windows
Go to Settings to activate Windows.
amzn2023

[ec2-user@ip-172-31-21-131 ~]$
```

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

```
ec2-user@ip-172-31-21-131:~$ Installed:
apr-2.5-1.amzn2023.0.4.x86_64      apr-util-1.6.3-1.amzn2023.0.1.x86_64      apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
httpd-2.4-64-1.amzn2023.0.1.x86_64  httpd-core-2.4-64-1.amzn2023.0.1.x86_64  httpd-fs-filesystem-2.4-64-1.amzn2023.0.1.x86_64
libbrotli1-1.0-9-4.amzn2023.0.2.x86_64  mailcap-2.1.49-3.amzn2023.0.3.noarch  mod_http2-2.0-27-1.amzn2023.0.3.x86_64
Dependencies resolved.
Complete!
[ec2-user@ip-172-31-21-131 ~]$ sudo yum install httpd mariadb105-server php -y
Last metadata expiration check: 0:01:08 ago on Mon Sep 8 12:46:00 2025.
Package httpd-2.4-64-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.

Package                                         Architecture   Version          Repos
Installing:
mariadb105-server                           x86_64         3:10.5.29-1.amzn2023.0.1
php8.4                                         x86_64         8.4.10-1.amzn2023.0.1
Installing dependencies:
libbrotli1                                     x86_64         1.0.19-4.amzn2023
libedit                                         x86_64         1.1.43-1.amzn2023.0.2
mariadb-connector-c                           x86_64         3.3.10-1.amzn2023.0.1
mariadb-connector-c-config                   noarch        3.3.10-1.amzn2023.0.1
mariadb-common                                 x86_64         3:10.5.29-1.amzn2023.0.1
mariadb105-common                            x86_64         3:10.5.29-1.amzn2023.0.1
mysql-libs-linux                               noarch        3:10.5.29-1.amzn2023.0.1
nginx-fs-filesystem                           noarch        1.0.4-2.amzn2023.0.1
perl-DBI                                       x86_64         1:1.78-0-1.amzn2023.0.2
perl-DBD-MariaDB                             x86_64         1.22-1.amzn2023.0.4
perl-DBD-MariaDB                             x86_64         1.64-3-7.amzn2023.0.3
perl-Data-Dumper                             x86_64         2.174-460.amzn2023.0.2
perl-File-Copy                                noarch        2.30-477.amzn2023.0.7
perl-File-Sort                                noarch        1:1.9998-19-2.amzn2023.0.2
perl-Math-BigInt                             noarch        0.2624-500.amzn2023.0.2
perl-Math-BigInt                             noarch        1.59-477.amzn2023.0.7
perl-Math-Complex                            x86_64         3.27-477.amzn2023.0.7
perl-Scalar-List-Util                         noarch        2.27-477.amzn2023.0.7
php8.4-cgi                                   x86_64         8.4.10-1.amzn2023.0.1
php8.4-common                                x86_64         8.4.10-1.amzn2023.0.1
php8.4-process                               x86_64         8.4.10-1.amzn2023.0.1
Activate Windows
Go to Settings to activate Windows.
amzn2023

[ec2-user@ip-172-31-21-131 ~]$
```

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

Step 4: Install PHP Connector

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

```
ec2-user@ip-172-31-21-131:~$ curl www.html
Sep 08 12:48:07 ip-172-31-21-131.ec2.internal systemd[1]: Started mariadb.service - MariaDB 10.5 database server.
● php-fpm.service - The PHP FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; preset: disabled)
   Active: active (running) since Fri Sep  8 12:48:07 2023; 1s ago
     Docs: man:php-fpm(8)
     Tasks: 1 (limit: 128)
    CGroup: /system.slice/php-fpm.service
            └─1: php-fpm

Sep 08 12:48:07 ip-172-31-21-131 ~]$ sudo yum install php8.4-
php8.4-apcu-panel.noarch          php8.4-enchant.x86_64          php8.4-mod-php.x86_64          php8.4-pecl-igbinary.x86_64      php8.4-so
php8.4-bcmath.x86_64              php8.4-ffti.x86_64           php8.4-mysqlind.x86_64         php8.4-pecl-mspck-devel.x86_64  php8.4-ti
php8.4-dba.x86_64                php8.4-gd.x86_64            php8.4-odbc.x86_64           php8.4-pecl-mspck.x86_64       php8.4-zis
php8.4-dbg.x86_64                php8.4-gmp.x86_64          php8.4-pecl-apcu-devel.x86_64  php8.4-pecl-redis.x86_64
php8.4-devel.x86_64               php8.4-intl.x86_64         php8.4-pecl-apcu.x86_64        php8.4-pgsql.x86_64
php8.4-embedded.x86_64             php8.4-ldap.x86_64          php8.4-pecl-igbinary-devel.x86_64  php8.4-smp.x86_64
php8.4-apcu-panel.noarch          php8.4-enchant.x86_64          php8.4-mod-php.x86_64          php8.4-pecl-igbinary.x86_64      php8.4-so
php8.4-bcmath.x86_64              php8.4-ffti.x86_64           php8.4-mysqlind.x86_64         php8.4-pecl-mspck-devel.x86_64  php8.4-ti
php8.4-dba.x86_64                php8.4-gd.x86_64            php8.4-odbc.x86_64           php8.4-pecl-mspck.x86_64       php8.4-zis
php8.4-dbg.x86_64                php8.4-gmp.x86_64          php8.4-pecl-apcu-devel.x86_64  php8.4-pecl-redis.x86_64
php8.4-devel.x86_64               php8.4-intl.x86_64         php8.4-pecl-apcu.x86_64        php8.4-pgsql.x86_64
php8.4-embedded.x86_64             php8.4-ldap.x86_64          php8.4-pecl-igbinary-devel.x86_64  php8.4-smp.x86_64
php8.4-apcu-panel.noarch          php8.4-enchant.x86_64          php8.4-mod-php.x86_64          php8.4-pecl-igbinary.x86_64      php8.4-so
php8.4-bcmath.x86_64              php8.4-ffti.x86_64           php8.4-mysqlind.x86_64         php8.4-pecl-mspck-devel.x86_64  php8.4-ti
php8.4-dba.x86_64                php8.4-gd.x86_64            php8.4-odbc.x86_64           php8.4-pecl-mspck.x86_64       php8.4-zis
php8.4-dbg.x86_64                php8.4-gmp.x86_64          php8.4-pecl-apcu-devel.x86_64  php8.4-pecl-redis.x86_64
php8.4-devel.x86_64               php8.4-intl.x86_64         php8.4-pecl-apcu.x86_64        php8.4-pgsql.x86_64
php8.4-embedded.x86_64             php8.4-ldap.x86_64          php8.4-pecl-igbinary-devel.x86_64  php8.4-smp.x86_64
ec2-user@ip-172-31-21-131 ~]$ sudo yum install php8.4-mysqlind.x86_64
Last metadata expiration check: 0:03:35 ago on Mon Sep  8 12:46:00 2023.
Dependencies resolved.

=====


| Package         | Architecture | Version               | Repository |
|-----------------|--------------|-----------------------|------------|
| php8.4-mysqlind | x86_64       | 8.4.10-1.amzn2023.0.1 | amazon1    |


=====
Transaction Summary
=====
Install 1 Package

Total download size: 156 k
Installed size: 438 k
Is this ok [y/N]: y
Downloading Packages:
php8.4-mysqlind-8.4.10-1.amzn2023.0.1.x86_64.rpm

Total
Running transaction check
Transaction check succeeded.

Activate Windows
Go to Settings to activate Windows.
```

Step 5: Download and Extract Wordpress

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

```
ec2-user@ip-172-31-131-131:/var/www/html
```

Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
 php8.4-mysqlnd-8.4.10-1.amzn2023.0.1.x86_64
Running scriptlet: php8.4-mysqlnd-8.4.10-1.amzn2023.0.1.x86_64
Verifying : php8.4-mysqlnd-8.4.10-1.amzn2023.0.1.x86_64

Installed:
 php8.4-mysqlnd-8.4.10-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-131-131 ~]\$ cd /var/www/html/
[ec2-user@ip-172-31-131-131 html]\$ sudo wget https://wordpress.org/latest.tar.gz
--2025-09-08 12:54:41-- 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 (26MiB) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz 100%[==
2025-09-08 12:54:42 (33.9 MiB/s) - 'latest.tar.gz' saved [26925441/26925441]

[ec2-user@ip-172-31-131-131 html]\$ ls
latest.tar.gz
[ec2-user@ip-172-31-131-131 html]\$ sudo tar -xvzf
tar: option requires an argument -- 'f'
try 'tar --help' or 'tar --usage' for more information.
[ec2-user@ip-172-31-131-131 html]\$ sudo tar -xvzf latest.tar.gz
wordpress/
 wordpress/index.php
 wordpress/license.txt
 wordpress/readme.html
 wordpress/wp-activate.php
 wordpress/wp-admin/

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

```

ec2-user@ip-172-31-21-131:~/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
[ec2-user@ip-172-31-21-131 wordpress]$ cd ..

```

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:~]
[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:~]
[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)]> |
```

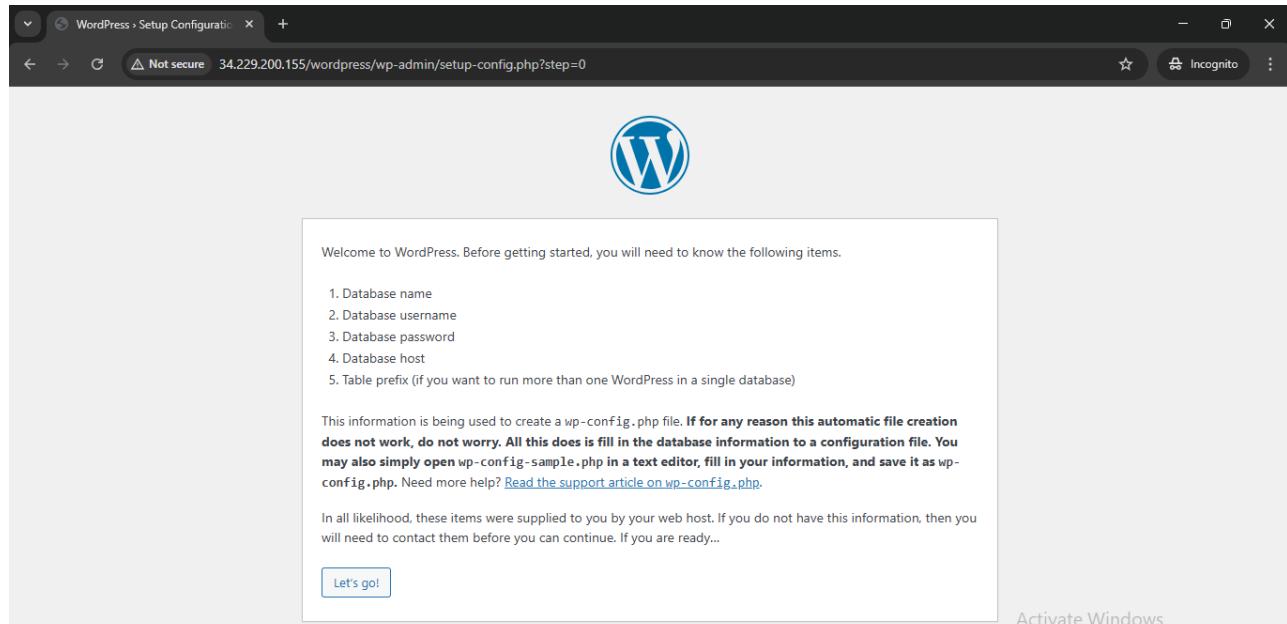
Activate Windows
Go to Settings to activate Windows.

Step 8: Grant permission to wordpress directory

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

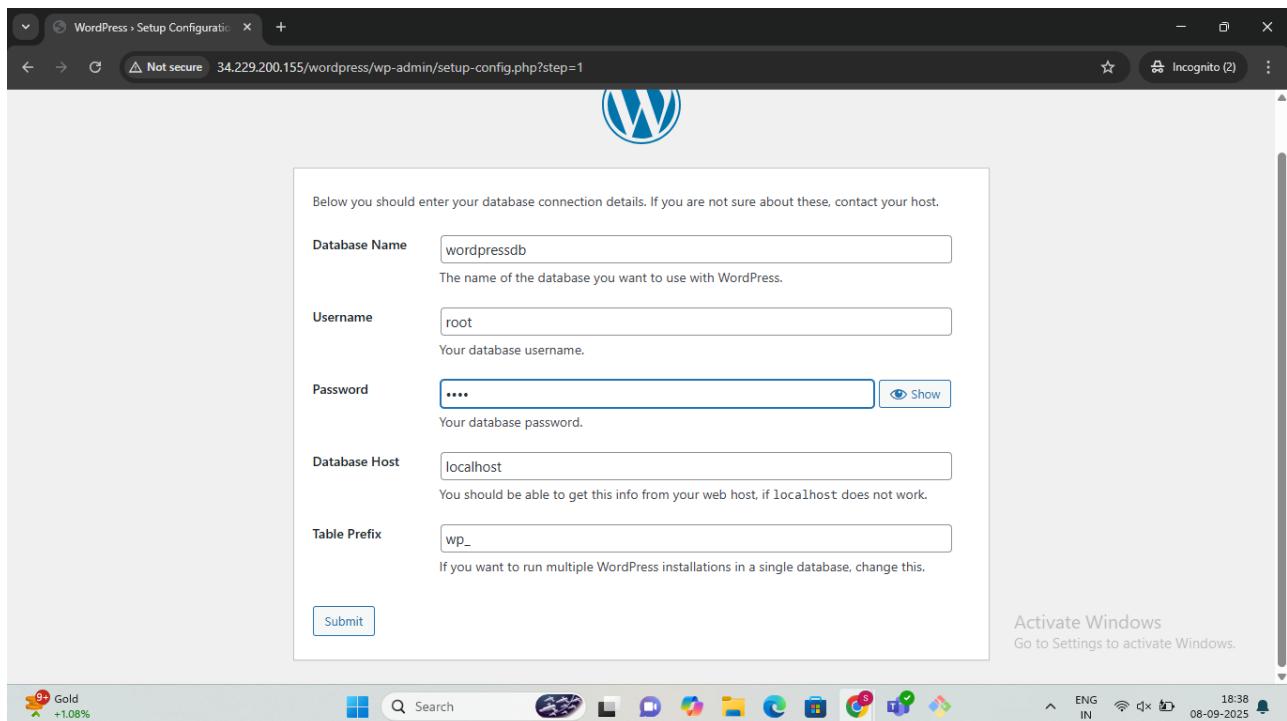
Step 9: Paste the public IP in any browser

1. Click on let's go

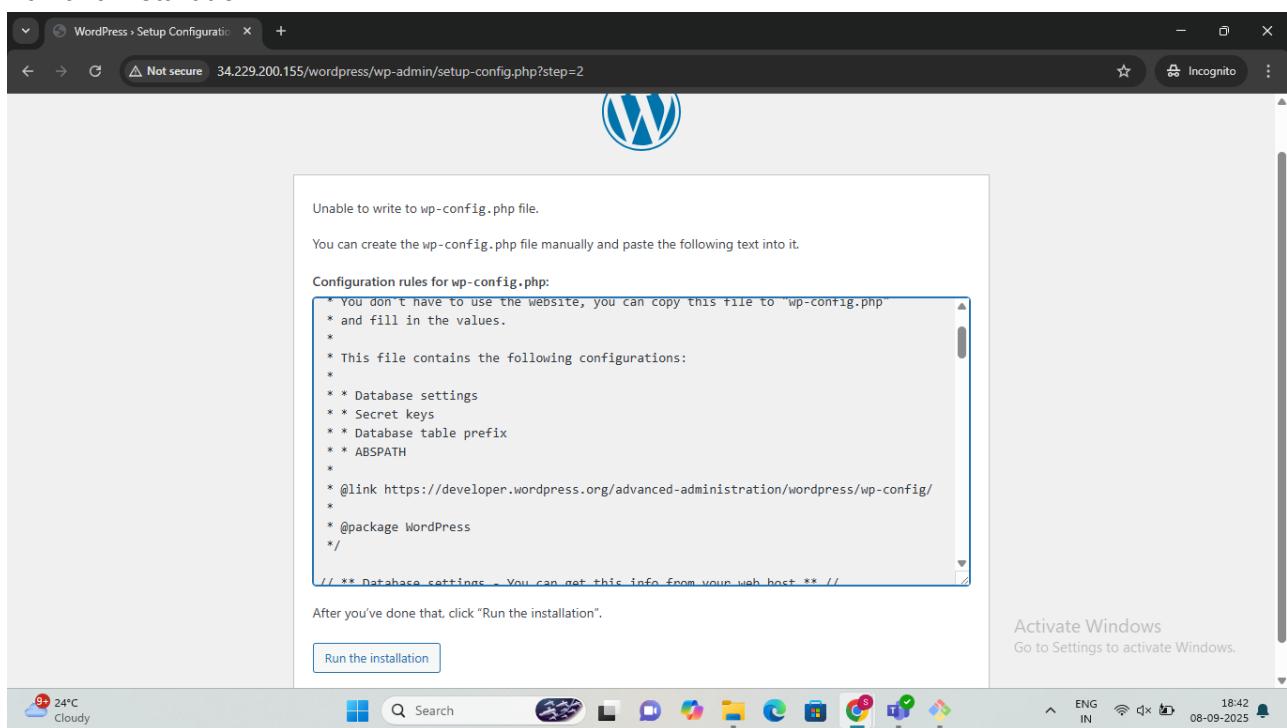


Activate Windows

2. Fill the information and click on submit



3. Run and Installation



4. Fill the information and click on install wordpress

The screenshot shows the 'Information needed' step of the WordPress installation process. The URL in the address bar is `34.229.200.155/wordpress/wp-admin/install.php?language=en_US`. The page title is 'WordPress > Installation'. The form fields are as follows:

- Site Title:** TechBlog
- Username:** root
- Password:** root (Strength: Very weak)
- Confirm Password:** Confirm use of weak password
- Your Email:** sakshikarle10@gmail.com
- Search engine visibility:** Discourage search engines from indexing this site (Note: It is up to search engines to honor this request)

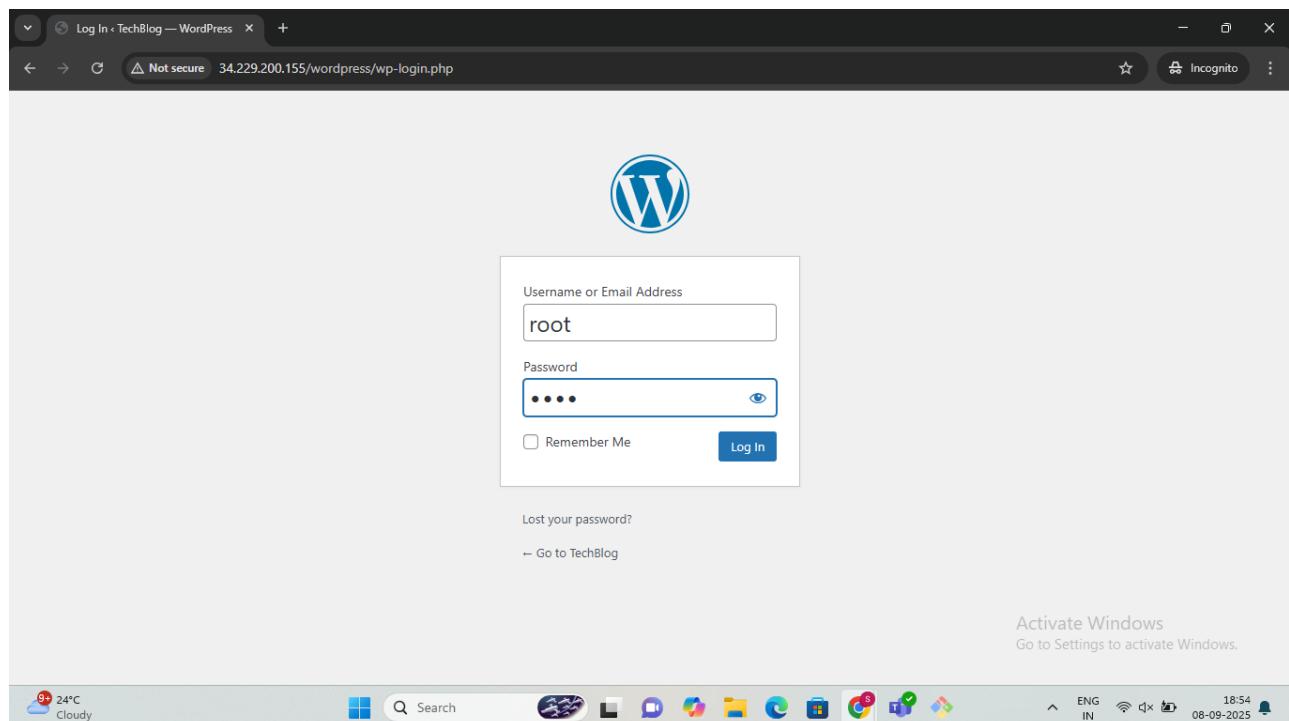
At the bottom is a blue 'Install WordPress' button. A status bar at the bottom of the screen shows 'Activate Windows Go to Settings to activate Windows.' and system icons.

5. Login to wordpress

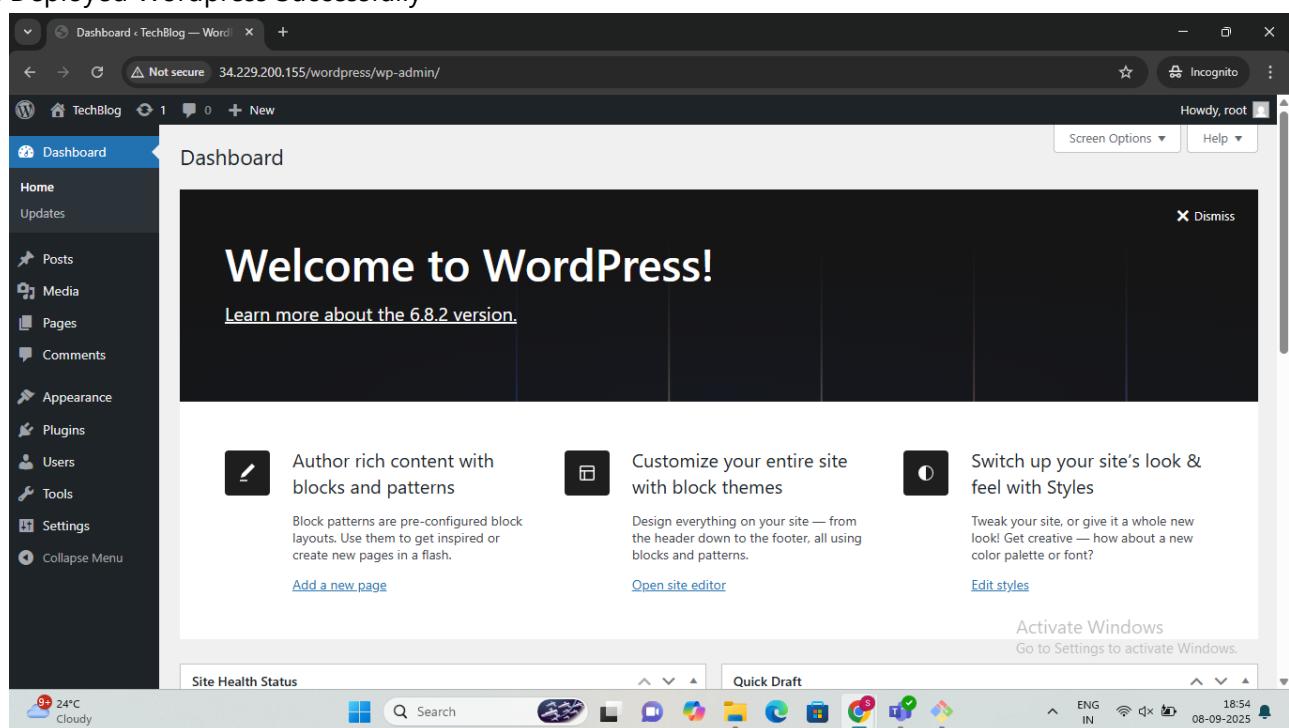
The screenshot shows the 'Success!' step of the WordPress installation process. The URL in the address bar is `34.229.200.155/wordpress/wp-admin/install.php?step=2`. The page title is 'WordPress > Installation'. The success message is: 'WordPress has been installed. Thank you, and enjoy!'. The login credentials are displayed:

Username: root
Password: Your chosen password.

A blue 'Log In' button is present. A status bar at the bottom of the screen shows 'Activate Windows Go to Settings to activate Windows.' and system icons.



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.