

Hosting Wordpress Step-by-Step

Introduction

This project offers a detailed walkthrough for launching a WordPress website on a server with Apache as the web server and MariaDB as the database. It covers setting up the environment, installing necessary components, configuring WordPress, and completing the deployment to make the site accessible online. By following this guide, you can achieve a secure, scalable, and production-ready WordPress setup.

Prerequisites

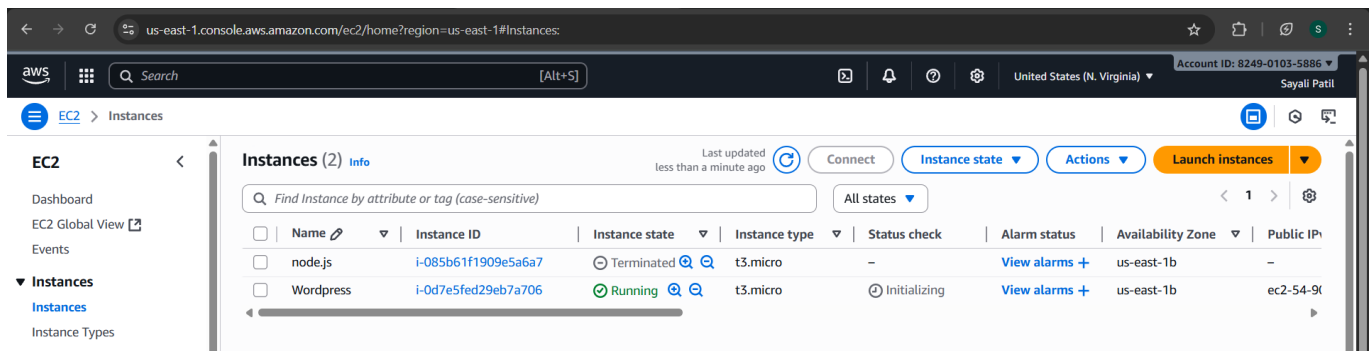
Before launching the WordPress website, confirm that the following are set up and operational:

- Linux Server – Amazon Linux operating system
- Web Server – Apache installed and actively running
- PHP – Including all necessary PHP extensions
- Database – MariaDB installed for managing WordPress data

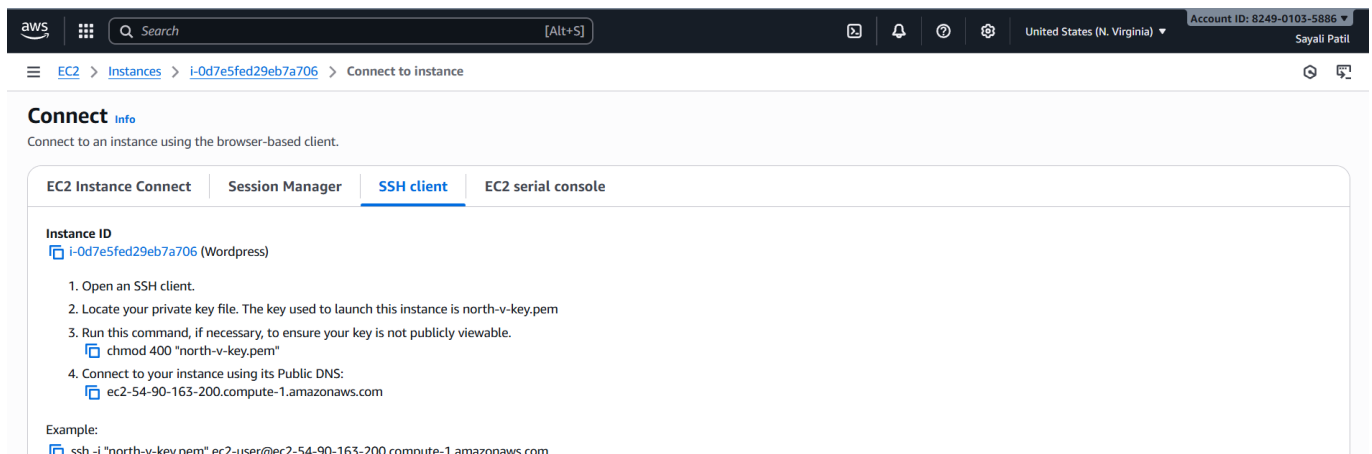
Step-by-Step Setup

Step 1: Launch an EC2 Instance and Connect Securely via SSH

1. Launch EC2 Instance



2. Copy SSH Command



3. Paste it on Git bash terminal

```
TUSHAR@LAPTOP-C5S5ASDO MINGW64 /c/sayali c\oud  
$ ssh -i "north-v-key.pem" ec2-user@ec2-54-90-163-200.compute-1.amazonaws.com  
The authenticity of host 'ec2-54-90-163-200.compute-1.amazonaws.com (64:ff9b::365a:a3c8)' can'  
ED25519 key fingerprint is SHA256:qE18+b8p4100eFz5q3m9js+8Ka9SWZrrjVnCPmKDgIW.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-54-90-163-200.compute-1.amazonaws.com' (ED25519) to the list o  
  
#_ Amazon Linux 2023  
~ \#####  
~~ \#####\  
~~ \###|  
~~ \#/_____  
~~ V~'|'-> https://aws.amazon.com/linux/amazon-linux-2023  
~~~~  
~~.-.-  
~/-'/_-/  
/m/'_-/-
```

Step 2: Automating the Installation of the LAMP Stack on an AWS EC2 Instance

1. Create a lamp.sh file

```
#sudo vim lamp.sh
```

```
[ec2-user@ip-172-31-21-121 ~]$ sudo vim lamp.sh
```

2. Insert the script to install Apache, MySQL, and PHP on the system.

```
ec2-user@ip-172-31-21-121:~$ 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. Open and run the file.

```
[ec2-user@ip-172-31-21-121 ~]$ sudo bash lamp.sh
Amazon Linux 2023 Kernel Livepatch repository                200 kB/s | 23 kB    00:00
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:00:02 ago on Fri Sep 19 15:50:57 2025.
Dependencies resolved.
```

Package	Arch	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.65-1.amzn2023.0.1	amazonlinux	47 k
mariadb105-server	x86_64	3:10.5.29-1.amzn2023.0.1	amazonlinux	10 M
php8.4	x86_64	8.4.10-1.amzn2023.0.1	amazonlinux	17 k
Installing dependencies:				
apr	x86_64	1.7.5-1.amzn2023.0.4	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.65-1.amzn2023.0.1	amazonlinux	1.4 M
httpd-filesystem	noarch	2.4.65-1.amzn2023.0.1	amazonlinux	13 k
httpd-tools	x86_64	2.4.65-1.amzn2023.0.1	amazonlinux	81 k
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
libsodium	x86_64	1.0.19-4.amzn2023	amazonlinux	176 k
libxslt	x86_64	1.1.43-1.amzn2023.0.2	amazonlinux	183 k
mailcap	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	33 k
mariadb-connector-c	x86_64	3.3.10-1.amzn2023.0.1	amazonlinux	211 k
mariadb-connector-c-config	noarch	3.3.10-1.amzn2023.0.1	amazonlinux	9.9 k
mariadb105	x86_64	3:10.5.29-1.amzn2023.0.1	amazonlinux	1.5 M
mariadb105-common	x86_64	3:10.5.29-1.amzn2023.0.1	amazonlinux	28 k
mariadb105-errmsg	x86_64	3:10.5.29-1.amzn2023.0.1	amazonlinux	212 k
mysql-selinux	noarch	1.0.4-2.amzn2023.0.3	amazonlinux	36 k
nginx-filesystem	noarch	1:1.28.0-1.amzn2023.0.2	amazonlinux	9.6 k
perl-B	x86_64	1.80-477.amzn2023.0.7	amazonlinux	177 k
perl-DBD-MariaDB	x86_64	1.22-1.amzn2023.0.4	amazonlinux	153 k

Step 3: Downloading and Setting Up WordPress

1. Go to the html file

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

2. Download Wordpress

```
#sudo wget https://wordpress.org/latest.tar.gz
```

3. Extract the archive

```
#sudo tar -xvzf latest.tar.gz
```

```
[ec2-user@ip-172-31-21-121 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-21-121 html]$ sudo wget https://wordpress.org/latest.tar.gz
--2025-09-19 15:53:47-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252, 2607:f978:5:8002::c68f:a4fc
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%[=====] 25.68M  75.7MB/s   in 0.3s

2025-09-19 15:53:47 (75.7 MB/s) - 'latest.tar.gz' saved [26925441/26925441]

[ec2-user@ip-172-31-21-121 html]$ ls
latest.tar.gz
[ec2-user@ip-172-31-21-121 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
wordpress/wp-admin/admin-footer.php
```

Step 4: Remove latest.tar.gz

```
#sudo rm -rf latest.tar.gz
```

```
[ec2-user@ip-172-31-21-121 html]$ ls
latest.tar.gz wordpress
[ec2-user@ip-172-31-21-121 html]$ sudo rm -rf latest.tar.gz
[ec2-user@ip-172-31-21-121 html]$ ls
wordpress
[ec2-user@ip-172-31-21-121 html]$ |
```

Step 5: Go to the wordpress folder

```
#cd wordpress/
```

```
[ec2-user@ip-172-31-21-121 html]$ cd wordpress/
[ec2-user@ip-172-31-21-121 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-172-31-21-121 wordpress]$ |
```

Step 6: Create a Database for WordPress

1. Create Username and Password

```
#sudo mysql
```

```
#alter user root@localhost identified by 'root';
```

```
[ec2-user@ip-172-31-21-121 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 (mariadb 105-server)

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

```
[ec2-user@ip-172-31-21-121 wordpress]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
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 & Show 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)
```

Step 7: Install Connector

```
#sudo yum install php8.4-mysqldb.x86_64
```

```
[ec2-user@ip-172-31-21-121 wordpress]$ sudo yum install php8.4-mysqldb.x86_64
Last metadata expiration check: 0:18:25 ago on Fri Sep 19 15:50:57 2025.
Dependencies resolved.
=====
Package                        Architecture      Version           Repository        Size
=====
Installing:
php8.4-mysqldb                x86_64            8.4.10-1.amzn2023.0.1  amazonlinux      156 k
Transaction Summary
=====
Install 1 Package
Total download size: 156 k
Installed size: 438 k
Is this ok [y/N]: y
Downloading Packages:
php8.4-mysqldb-8.4.10-1.amzn2023.0.1.x86_64.rpm                                3.4 MB/s | 156 kB    00:00
-----
Total                                                                           1.9 MB/s | 156 kB    00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : php8.4-mysqldb-8.4.10-1.amzn2023.0.1.x86_64                1/1
  Installing     : php8.4-mysqldb-8.4.10-1.amzn2023.0.1.x86_64                1/1
  Running scriptlet: php8.4-mysqldb-8.4.10-1.amzn2023.0.1.x86_64                1/1
  Verifying      : php8.4-mysqldb-8.4.10-1.amzn2023.0.1.x86_64                1/1

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

Complete!
```

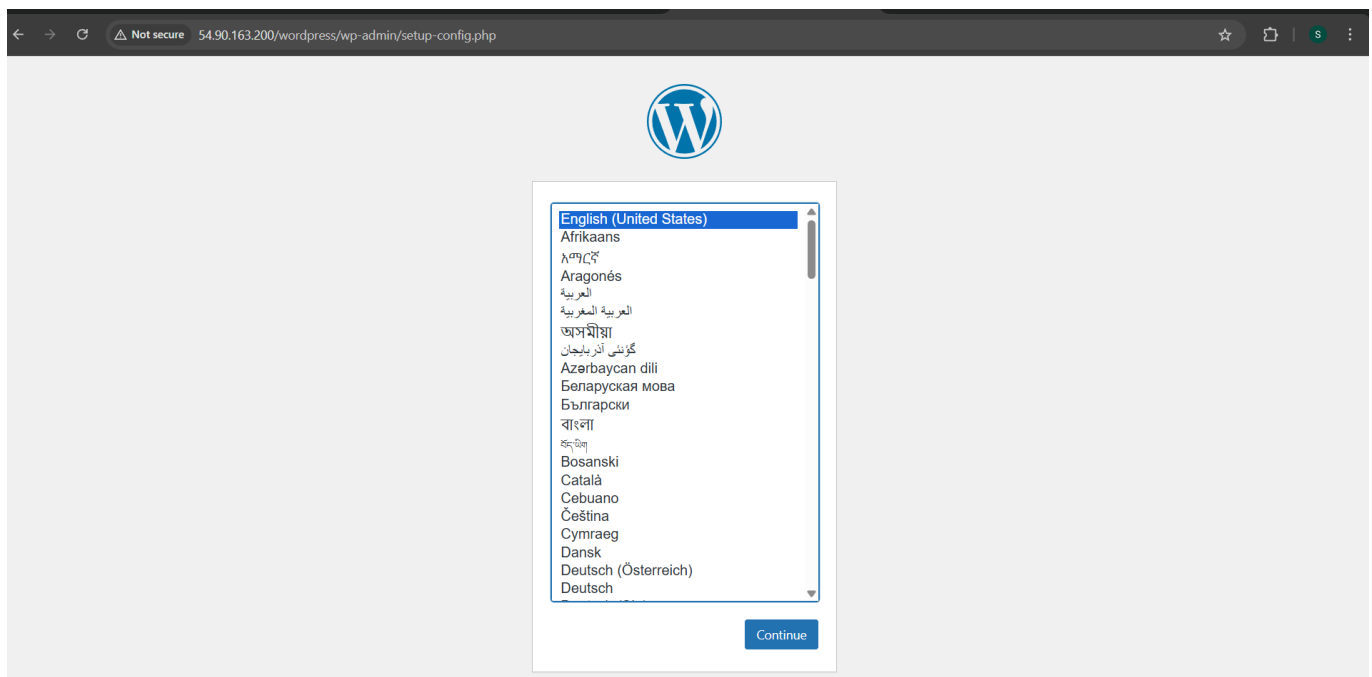
Step 8: Change ownership of the files

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

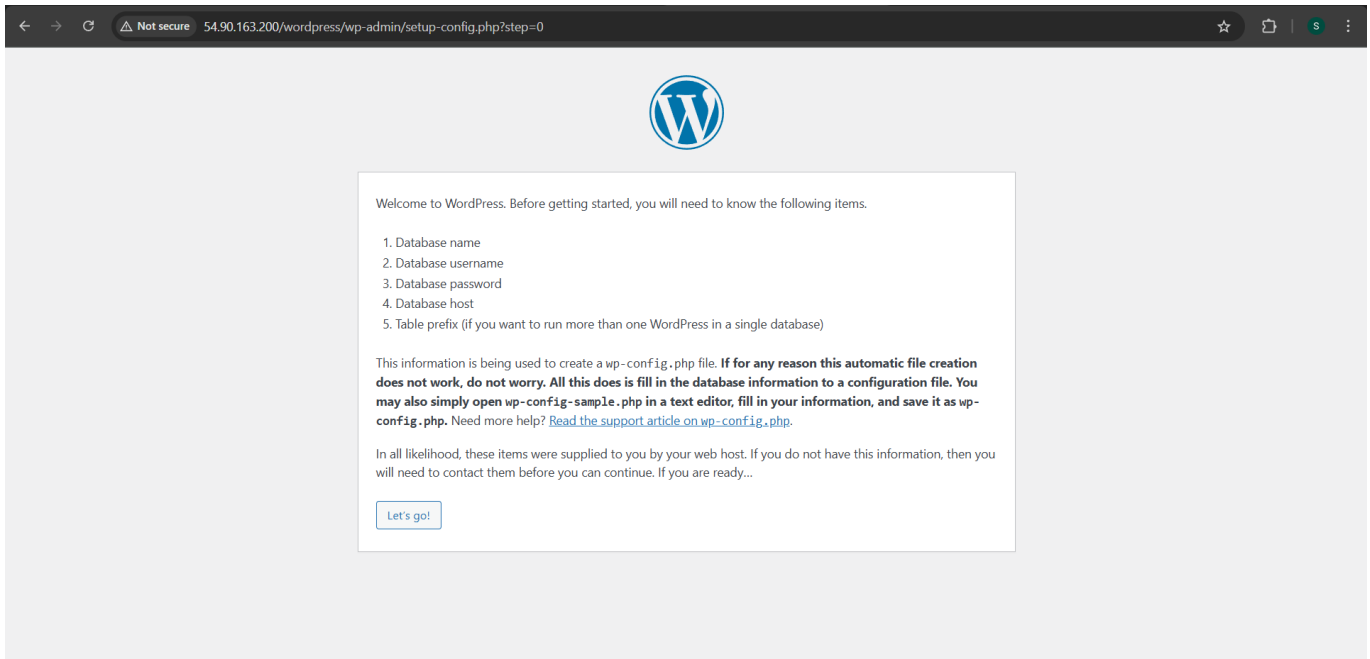
```
[ec2-user@ip-172-31-21-121 html]$ sudo chown -R apache:apache wordpress/
[ec2-user@ip-172-31-21-121 html]$
```

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

1. Hit Public IP on Browser
2. Click on continue



3. Click on Let's go



Welcome to WordPress. Before getting started, you will need to know the following items.

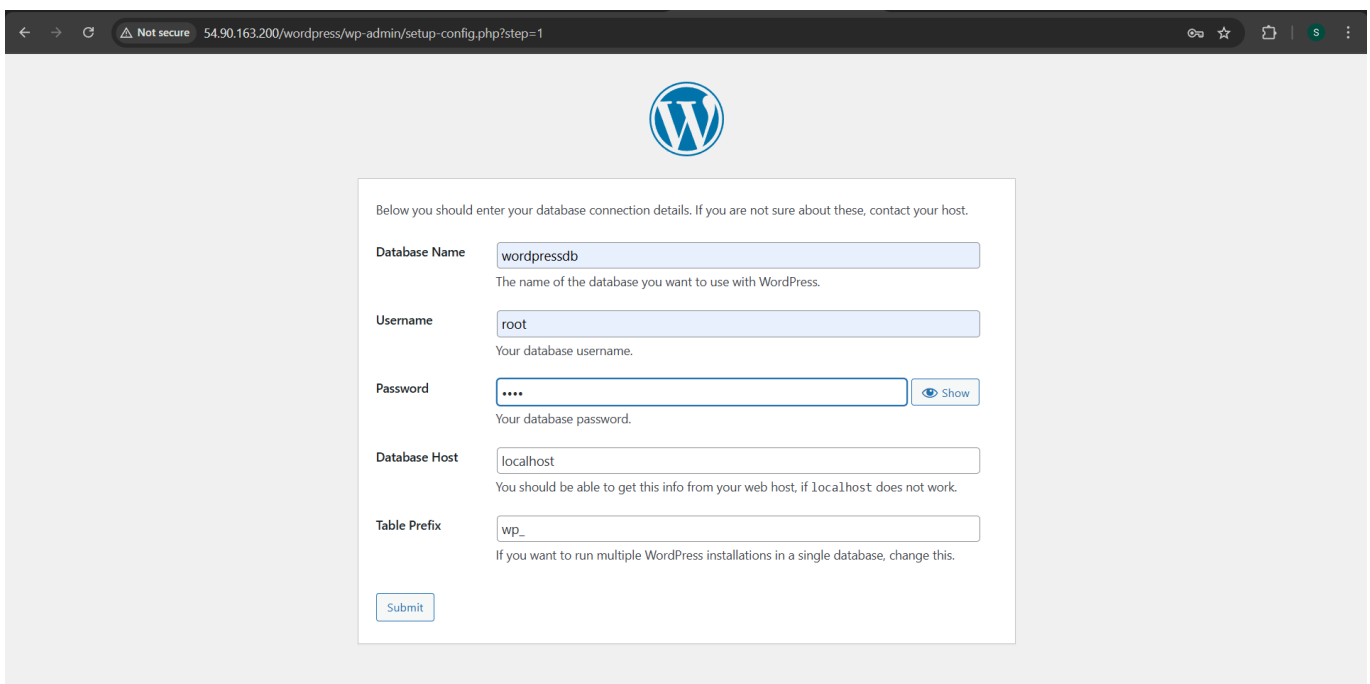
1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

This information is being used to create a `wp-config.php` file. **If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? [Read the support article on wp-config.php.](#)

In all likelihood, these items were supplied to you by your web host. If you do not have this information, then you will need to contact them before you can continue. If you are ready...

[Let's go!](#)

4. Fill all details 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.

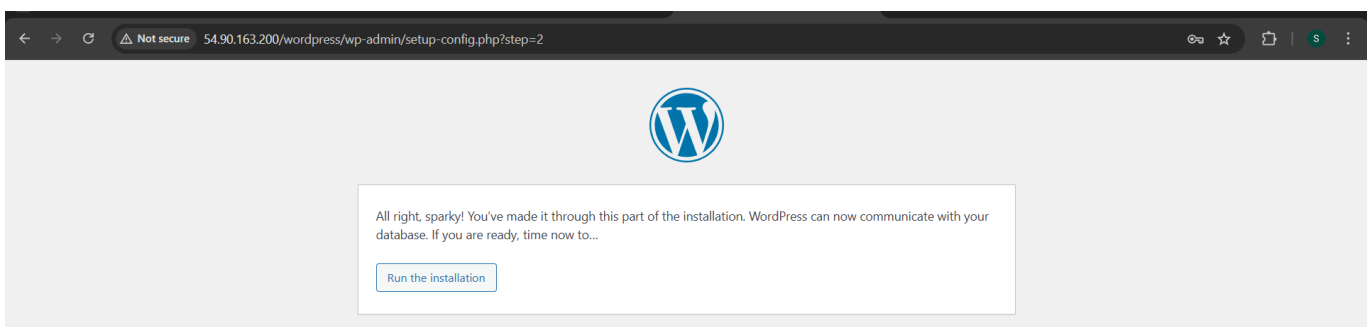
Password [Show](#)
Your database password.

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](#)

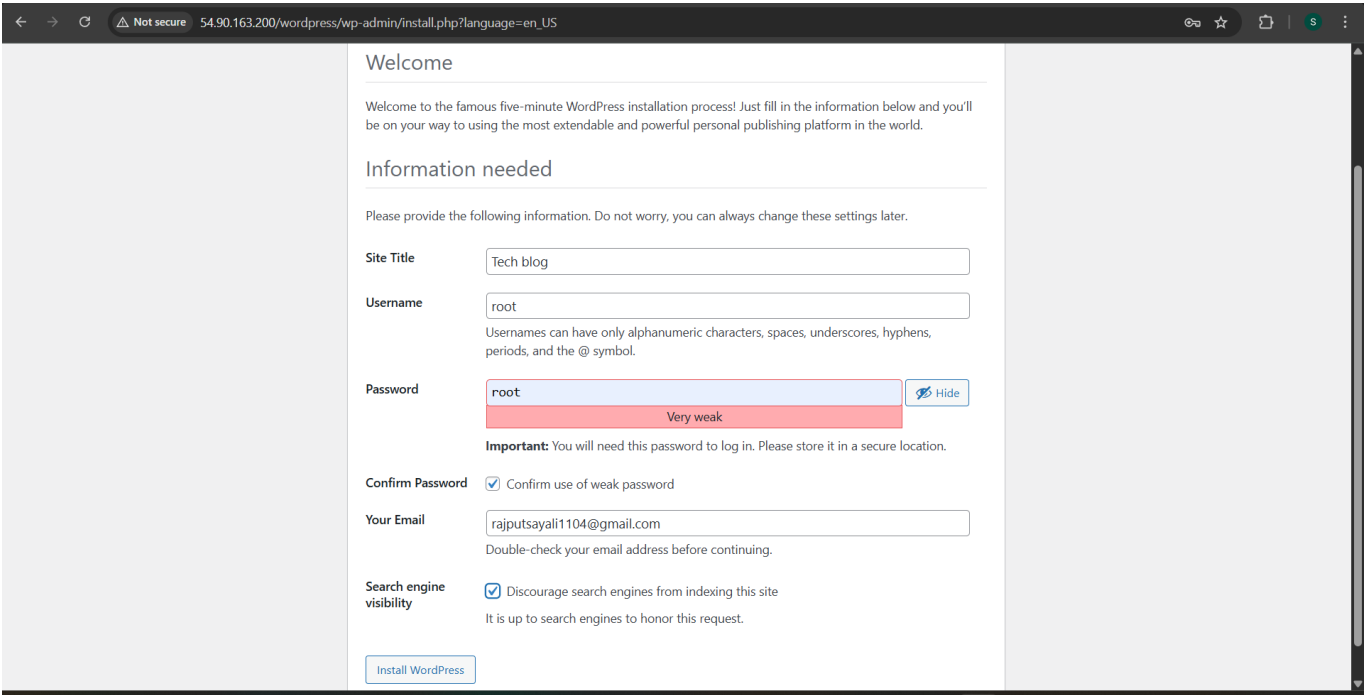
5. Run the Installation



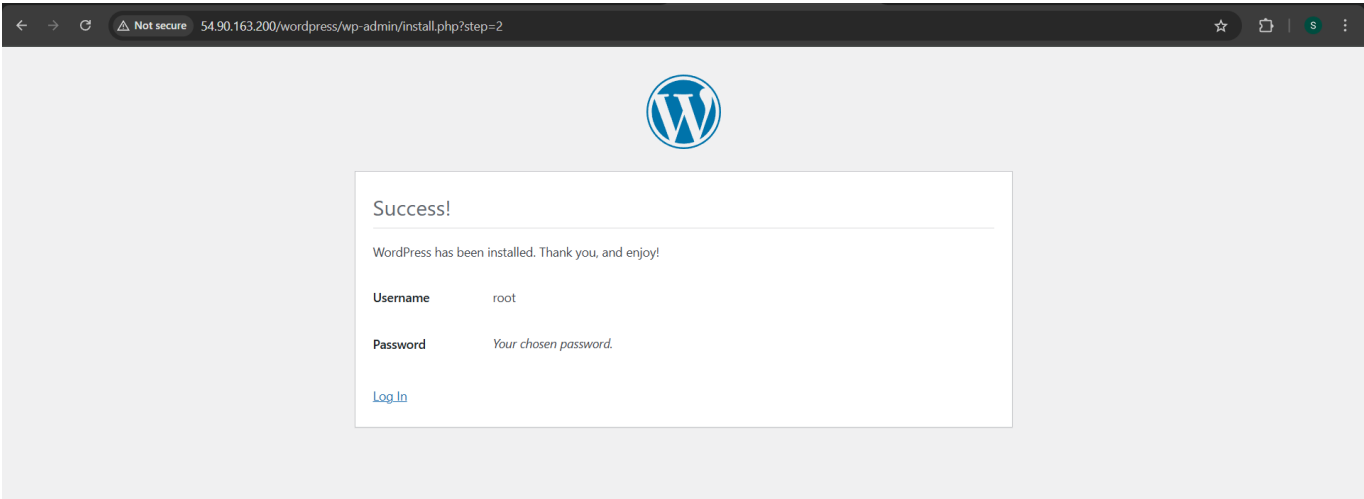
All right, sparky! You've made it through this part of the installation. WordPress can now communicate with your database. If you are ready, time now to...

[Run the installation](#)

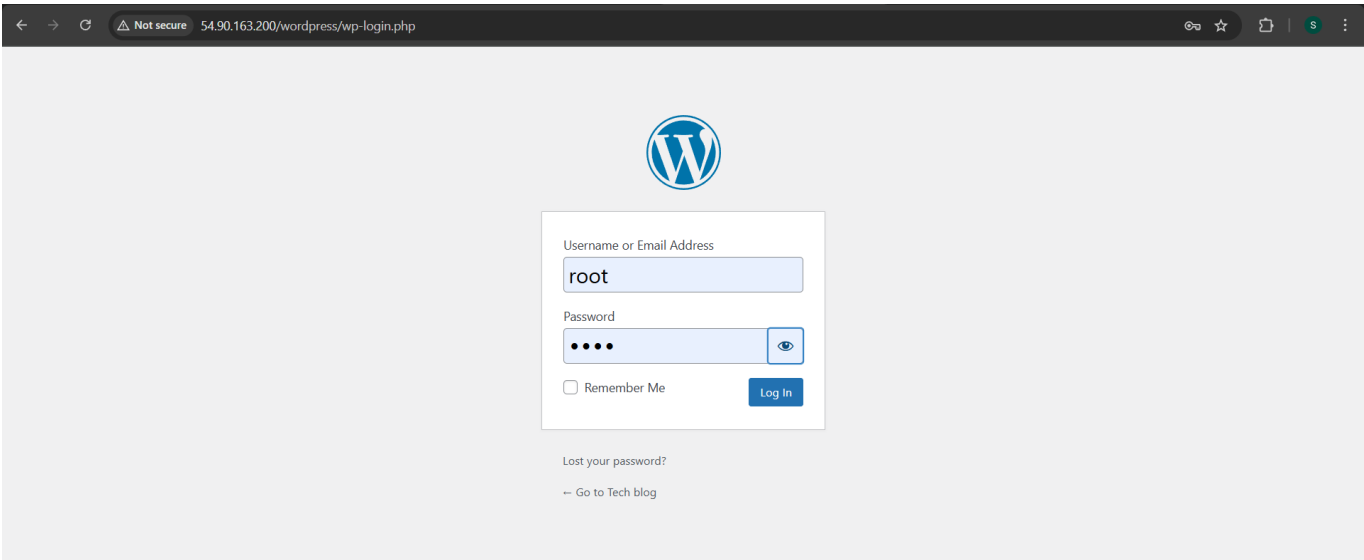
6. Fill up all information and click on install wordpress



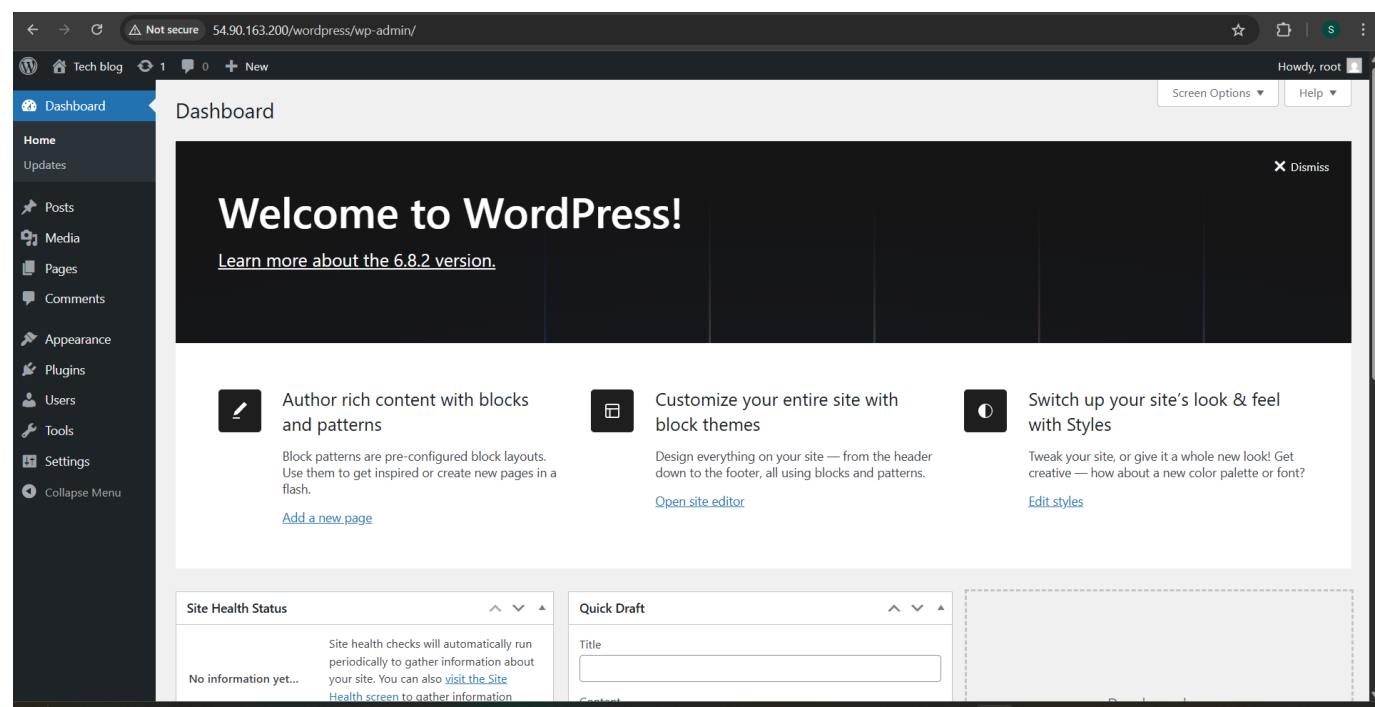
7. Login to wordpress



8. Click on Login



9. Deployment of WordPress was successfull.



10. A new table was automatically inserted into the 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)
```

Project Summary

This project involves deploying a WordPress website on a Linux-based server using Apache as the web server and MariaDB as the database. It covers the complete setup process, including downloading WordPress, configuring file permissions, setting up the database, and optimizing the server for production. The result is a secure, scalable, and fully functional WordPress site accessible to the public.