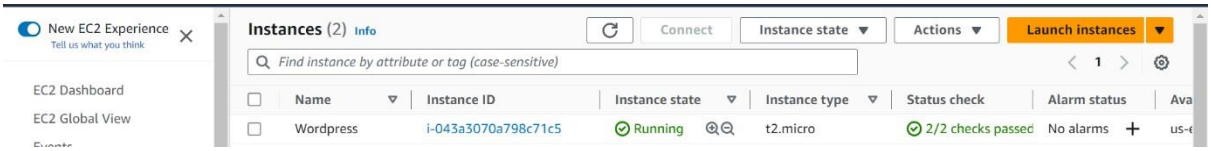
WordPress

Install wordpress website using ec2 instance

Step 1: create an instance with ubuntu as AMI

Here my instance name is Wordpress



Step 2:connect instance and deploy LAMP stack

-click on connect

Step 2:connect instance and deploy LAMP stack

-click on connect

-command:1.sudo apt-get update -y

2.sudo apt-get upgrade -y

3.sudo apt-get install apache2 -y

4.check status: sudo systemctl status apache2

5. If status in inactive than : sudo systemctl start apache2

After that: sudo systemctl enable apache2 :(this will keep the service running Even after reboot the machine)

-Deploy Database Server: run this command

a. sudo apt install mariadb-server mariadb-client -y

```
Jul 19 22:06:55 ip-172-31-82-201 systemd[1]: Starting The Apache HTTP Server...
Jul 19 22:06:55 ip-172-31-82-201 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-82-201:~$ sudo systemctl start apache2
ubuntu@ip-172-31-82-201:~$ 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-172-31-82-201:~$ sudo apt install mariadb-server  mariadb-client  -y
```

b. sudo systemctl start mariadb

```
 systemctl restart networkd-dispatcher.service
 systemctl restart unattended-upgrades.service
 systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-82-201:~$ sudo systemctl start mariadb
```

c. sudo systemctl status mariadb

```
ubuntu@ip-172-31-82-201:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.6.12 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-07-19 22:21:11 UTC; 3min 38s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 17616 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0/SUCCESS)
    Process: 17621 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
    Process: 17630 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||   VAR=`cd /usr/bin/..; /usr/bin/galera_recovery`; [ $? -e>
    Process: 17671 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
    Process: 17673 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 17660 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 1141)
     Memory: 67.1M
        CPU: 380ms
     CGroup: /system.slice/mariadb.service
             └─17660 /usr/sbin/mariadbd

ubuntu@ip-172-31-82-201:~$
```

d. set up ROOT password :

-command: sudo mysql_secure_installation

```
ubuntu@ip-172-31-82-201:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
```

-press enter

Switch to unix_socket authentication [Y/n] y

Change the root password? [Y/n] y

New password: (enter any password)

Re-enter new password:(re-enter that password)

Remove anonymous users? [Y/n] y

Disallow root login remotely? [Y/n] y

Remove test database and access to it? [Y/n] y

-Than restart the MariaDB: sudo systemctl restart mariadb

```
Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
ubuntu@ip-172-31-82-201:~$ ^C
ubuntu@ip-172-31-82-201:~$ sudo systemctl restart mariadb
ubuntu@ip-172-31-82-201:~$
```

-Install PHP:

Command: sudo apt install php php-mysql php-gd php-cli php-common -y

```
Thanks for using MariaDB!
ubuntu@ip-172-31-82-201:~$ ^C
ubuntu@ip-172-31-82-201:~$ sudo systemctl restart mariadb
ubuntu@ip-172-31-82-201:~$ sudo apt install php php-mysql php-gd php-cli php-common -y
Reading package lists... Done
Building dependency tree... Done
```

-Reload the page

-Install Wordpress :

Command: - sudo apt install wget unzip -y

```
Last login: Wed Jul 19 22:00:24 2023 from 18.206.107.29
ubuntu@ip-172-31-82-201:~$ sudo apt install wget unzip -y
```

i-043a3070a798c71c5 (Wordpress) ×

-sudo wget https://wordpress.org/latest.zip

```
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-82-201:~$ sudo wget https://wordpress.org/latest.zip
```

-ls

-sudo unzip lastest.zip

```
2023-07-19 22:50:19 (45.8 MB/s) - 'latest.zip' saved [24498824/24498824]

ubuntu@ip-172-31-82-201:~$ ls
latest.zip
ubuntu@ip-172-31-82-201:~$ sudo unzip latest.zip
```

-ls

```
  inflating: wordpress/wp-admin/post-new.php
  inflating: wordpress/wp-admin/themes.php
  inflating: wordpress/wp-admin/options-reading.php
  inflating: wordpress/wp-trackback.php
  inflating: wordpress/wp-comments-post.php
ubuntu@ip-172-31-82-201:~$ ls
latest.zip  wordpress
ubuntu@ip-172-31-82-201:~$
```

Here we got out wordpress directory

We copy all content in wordpress to our document root of apache server Command:

-sudo cp -r wordpress/* /var/www/html/

```
ubuntu@ip-172-31-82-201:~$ sudo cp -r wordpress/* /var/www/html/
ubuntu@ip-172-31-82-201:~$ cd /var/www/html/
ubuntu@ip-172-31-82-201:/var/www/html$ ls
index.html   readme.html      wp-blog-header.php    wp-content   wp-links-opml.php  wp-mail.php      wp-trackback.php
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
ubuntu@ip-172-31-82-201:/var/www/html$
```

your files are copied .

Than you to change the ownership of the file from root to www-data:

Command: for check ownership: - ls -l

```
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
ubuntu@ip-172-31-82-201:/var/www/html$ ls -l
total 240
-rw-r--r--  1 root root 10671 Jul 19 22:06 index.html
-rw-r--r--  1 root root   405 Jul 19 23:03 index.php
-rw-r--r--  1 root root 19915 Jul 19 23:03 license.txt
-rw-r--r--  1 root root  7402 Jul 19 23:03 readme.html
-rw-r--r--  1 root root  7205 Jul 19 23:03 wp-activate.php
drwxr-xr-x  9 root root  4096 Jul 19 23:03 wp-admin
-rw-r--r--  1 root root   351 Jul 19 23:03 wp-blog-header.php
-rw-r--r--  1 root root  2338 Jul 19 23:03 wp-comments-post.php
-rw-r--r--  1 root root  3013 Jul 19 23:03 wp-config-sample.php
drwxr-xr-x  4 root root  4096 Jul 19 23:03 wp-content
-rw-r--r--  1 root root  5536 Jul 19 23:03 wp-cron.php
drwxr-xr-x 28 root root 12288 Jul 19 23:03 wp-includes
-rw-r--r--  1 root root  2502 Jul 19 23:03 wp-links-opml.php
-rw-r--r--  1 root root  3792 Jul 19 23:03 wp-load.php
-rw-r--r--  1 root root 49330 Jul 19 23:03 wp-login.php
-rw-r--r--  1 root root  8541 Jul 19 23:03 wp-mail.php
-rw-r--r--  1 root root 24993 Jul 19 23:03 wp-settings.php
-rw-r--r--  1 root root 34350 Jul 19 23:03 wp-signup.php
-rw-r--r--  1 root root  4889 Jul 19 23:03 wp-trackback.php
-rw-r--r--  1 root root  3238 Jul 19 23:03 xmlrpc.php
ubuntu@ip-172-31-82-201:/var/www/html$
```

This are set as root

For changing type: -sudo chown www-data:www-data -R /var/www/html/

```
-rw-r--r--  1 root root  3238 Jul 19 23:03 xmlrpc.php
ubuntu@ip-172-31-82-201:/var/www/html$ sudo chown www-data:www-data  -R  /var/www/html/
ubuntu@ip-172-31-82-201:/var/www/html$ ls -l
total 240
-rw-r--r--  1 www-data www-data 10671 Jul 19 22:06 index.html
-rw-r--r--  1 www-data www-data   405 Jul 19 23:03 index.php
-rw-r--r--  1 www-data www-data 19915 Jul 19 23:03 license.txt
-rw-r--r--  1 www-data www-data  7402 Jul 19 23:03 readme.html
-rw-r--r--  1 www-data www-data  7205 Jul 19 23:03 wp-activate.php
drwxr-xr-x  9 www-data www-data  4096 Jul 19 23:03 wp-admin
-rw-r--r--  1 www-data www-data   351 Jul 19 23:03 wp-blog-header.php
-rw-r--r--  1 www-data www-data  2338 Jul 19 23:03 wp-comments-post.php
-rw-r--r--  1 www-data www-data  3013 Jul 19 23:03 wp-config-sample.php
drwxr-xr-x  4 www-data www-data  4096 Jul 19 23:03 wp-content
-rw-r--r--  1 www-data www-data  5536 Jul 19 23:03 wp-cron.php
drwxr-xr-x 28 www-data www-data 12288 Jul 19 23:03 wp-includes
-rw-r--r--  1 www-data www-data  2502 Jul 19 23:03 wp-links-opml.php
-rw-r--r--  1 www-data www-data  3792 Jul 19 23:03 wp-load.php
-rw-r--r--  1 www-data www-data 49330 Jul 19 23:03 wp-login.php
-rw-r--r--  1 www-data www-data  8541 Jul 19 23:03 wp-mail.php
-rw-r--r--  1 www-data www-data 24993 Jul 19 23:03 wp-settings.php
-rw-r--r--  1 www-data www-data 34350 Jul 19 23:03 wp-signup.php
-rw-r--r--  1 www-data www-data  4889 Jul 19 23:03 wp-trackback.php
-rw-r--r--  1 www-data www-data  3238 Jul 19 23:03 xmlrpc.php
ubuntu@ip-172-31-82-201:/var/www/html$
```

Thus ownership is changes.

Delete index.html file because they show at the top

So for delete:

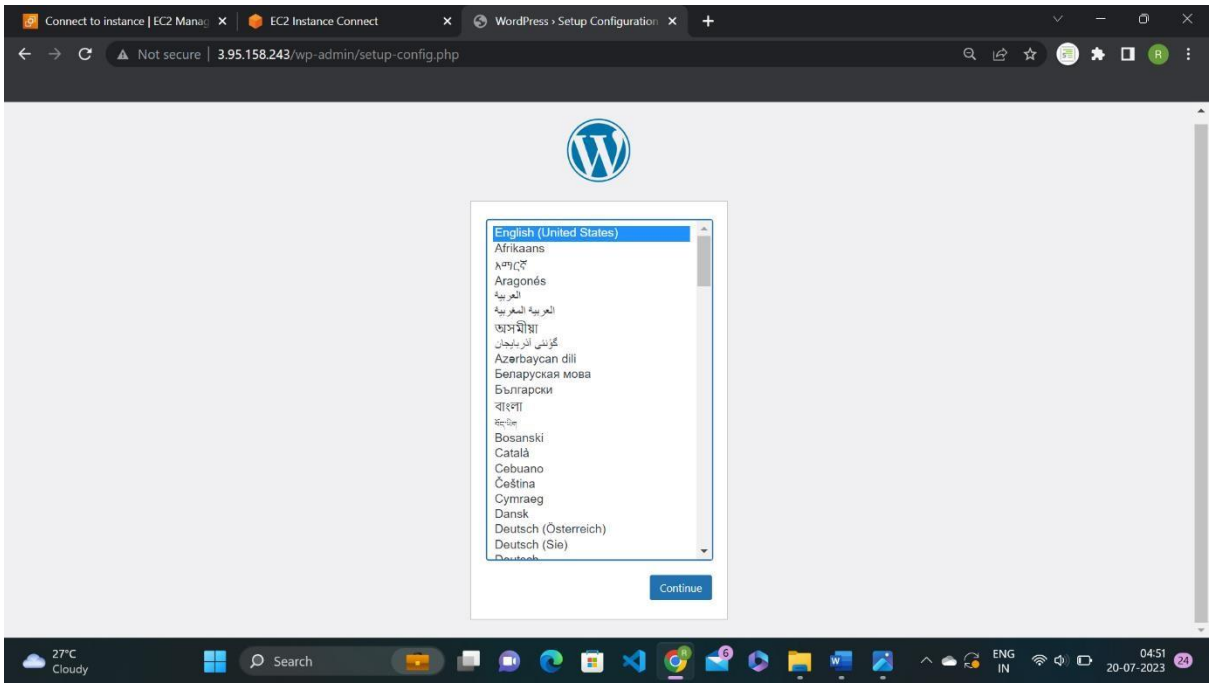
-sudo rm -rf index.html

```
-rw-r--r--  1 www-data www-data  4889 Jul 19 23:03 wp-trackback.php
-rw-r--r--  1 www-data www-data  3238 Jul 19 23:03 xmlrpc.php
ubuntu@ip-172-31-82-201:/var/www/html$ sudo rm -rf  index.html
ubuntu@ip-172-31-82-201:/var/www/html$ 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
ubuntu@ip-172-31-82-201:/var/www/html$
```

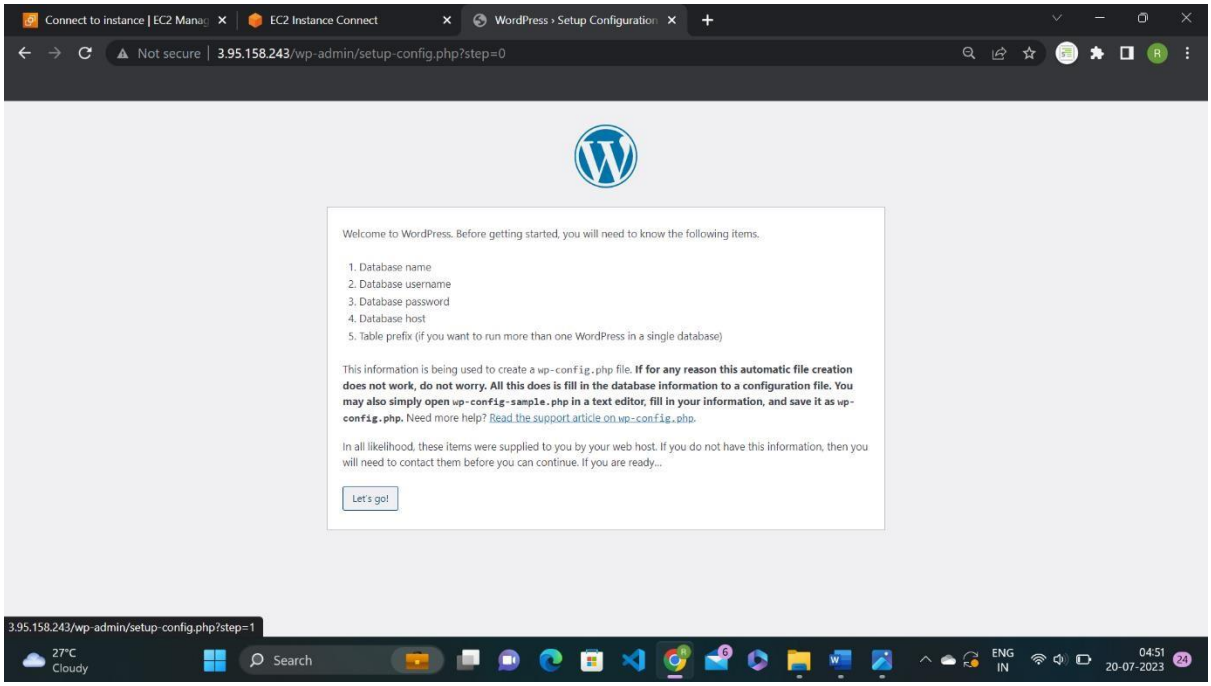
Our index.html successfully deleted Step

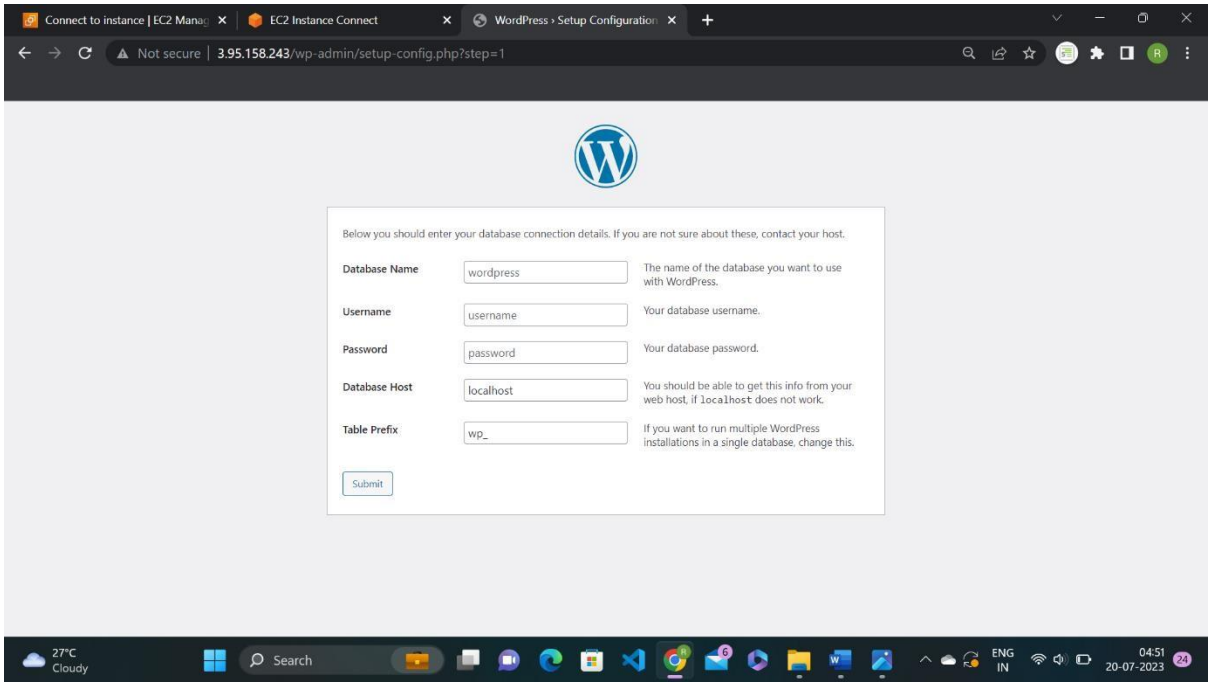
4:

-copy public ip paste on chrome tab



Click on continue <let’s go





Step 5: create database name ,username ,password

Go to -cd

Command: sudo mysql -u root -p

Type your password

```
ubuntu@ip-172-31-82-201:/var/www/html$ 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
ubuntu@ip-172-31-82-201:/var/www/html$ cd
ubuntu@ip-172-31-82-201:~$ sudo mysql-u root -p
sudo: mysql-u: command not found
ubuntu@ip-172-31-82-201:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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)]>
```

You are in a mariadb database

-for creating database : create database (name of database);

```
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)]> create database wordpress;
Query OK, 1 row affected (0.000 sec)
```

-for creating username and password: create user (username)@"%" identified by (password);

Error:syntax error

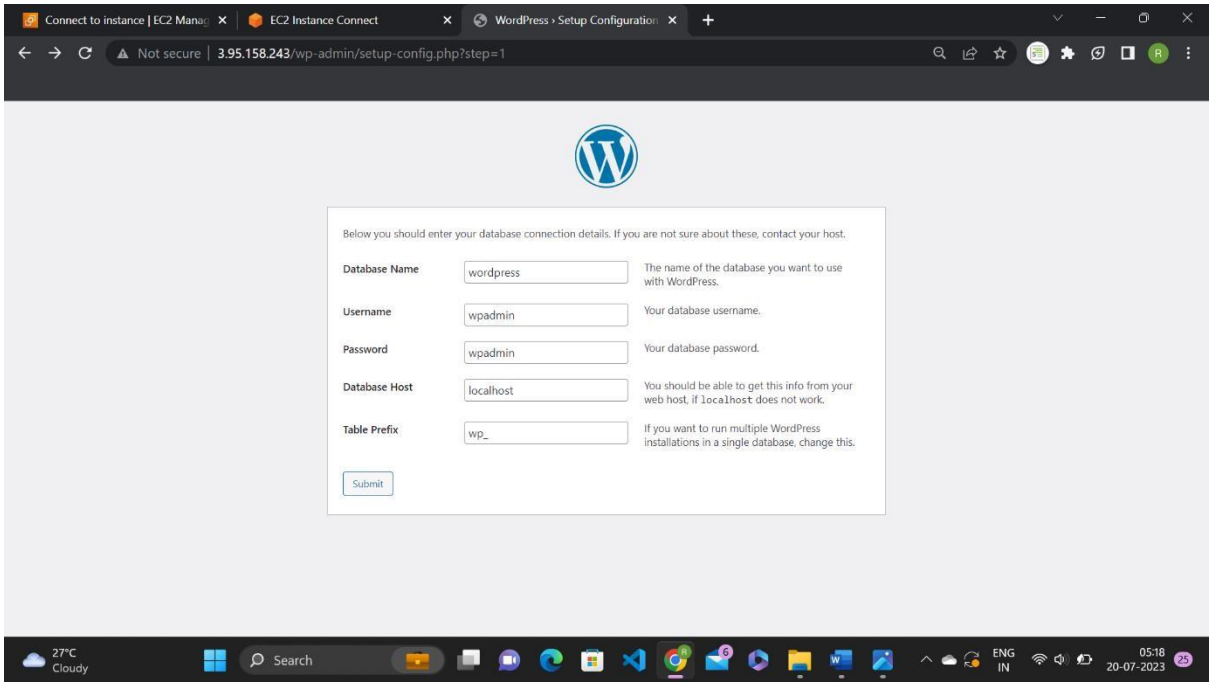
```
MariaDB [(none)]> create database wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user "wpadmin"@"%" indentified by "wpadmin";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to
 use near 'indentified by "wpadmin"' at line 1
MariaDB [(none)]> create user "wpadmin"@"%" identified by "wpadmin";
Query OK, 0 rows affected (0.003 sec)
```

-to grant privileges: grant all privileges on (database name).* to (username)@"%";

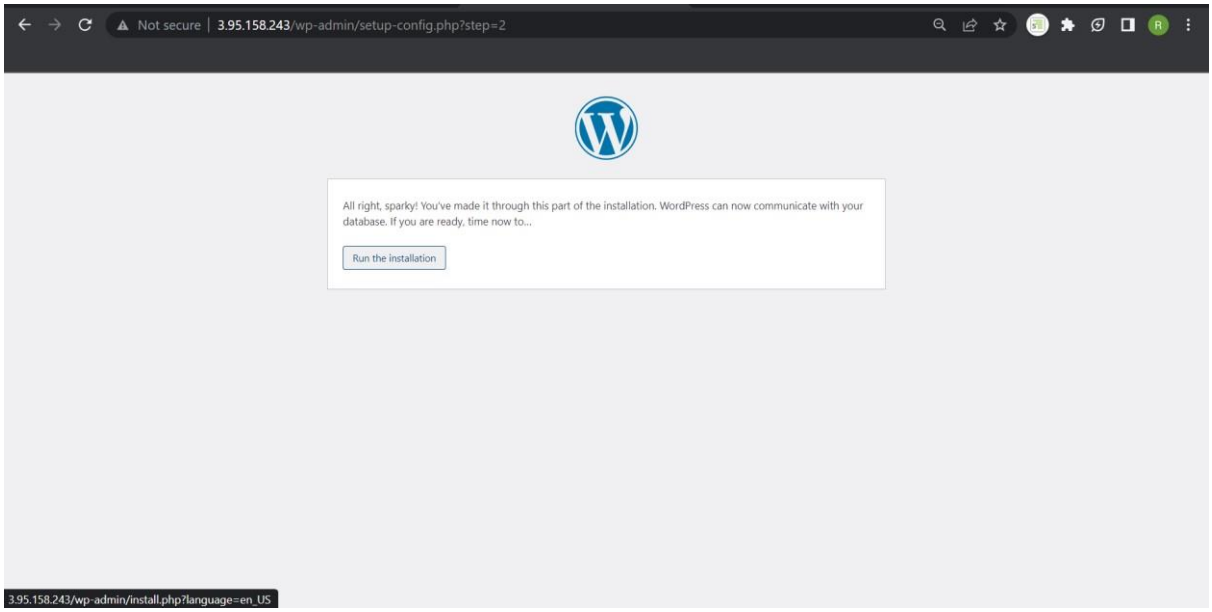
```
MariaDB [(none)]> grand all privileges on wordpress.* to "wpadmin"@"%";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to
 use near 'grand all privileges on wordpress.* to "wpadmin"@"%"' at line 1
MariaDB [(none)]> grant  all privileges on wordpress.* to "wpadmin"@"%";
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit
Bye
ubuntu@ip-172-31-82-201:~$
```

You are all set provide this information to your wordpress account

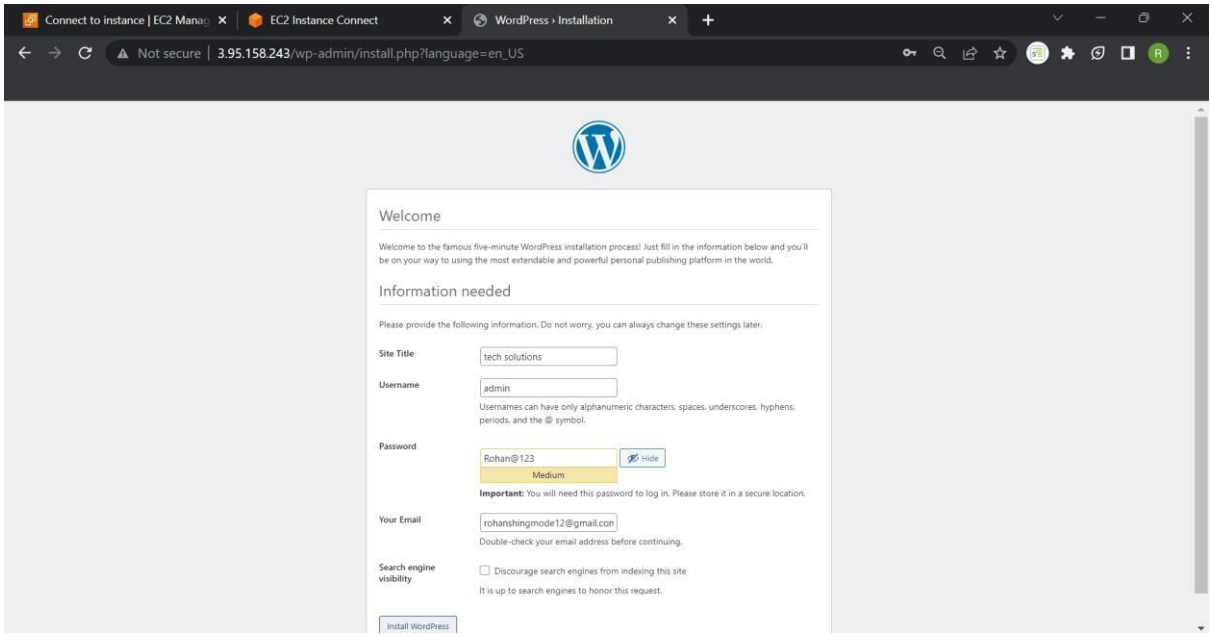
Setup all this information to your wordpress account



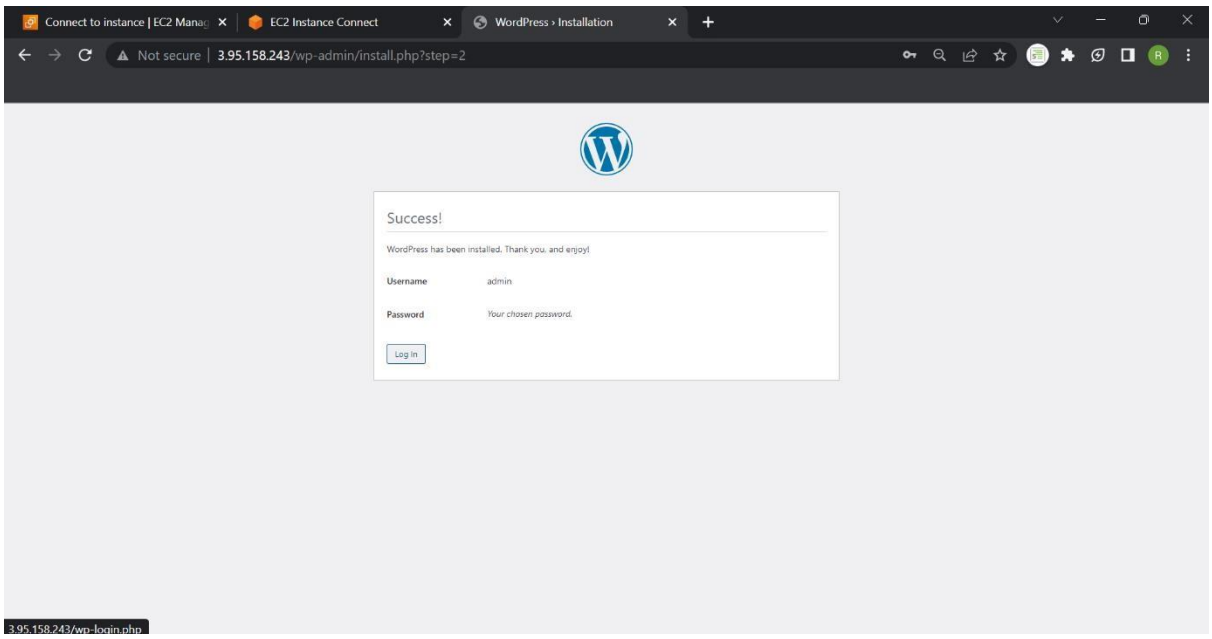
Click on submit



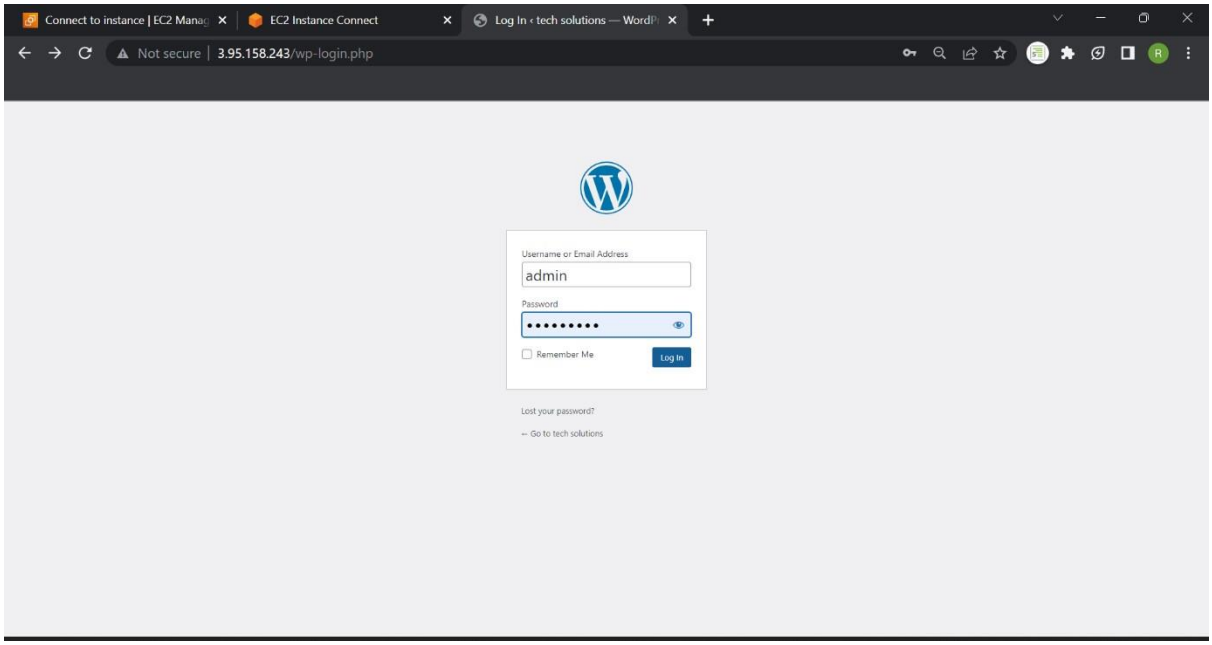
Click on run the installation



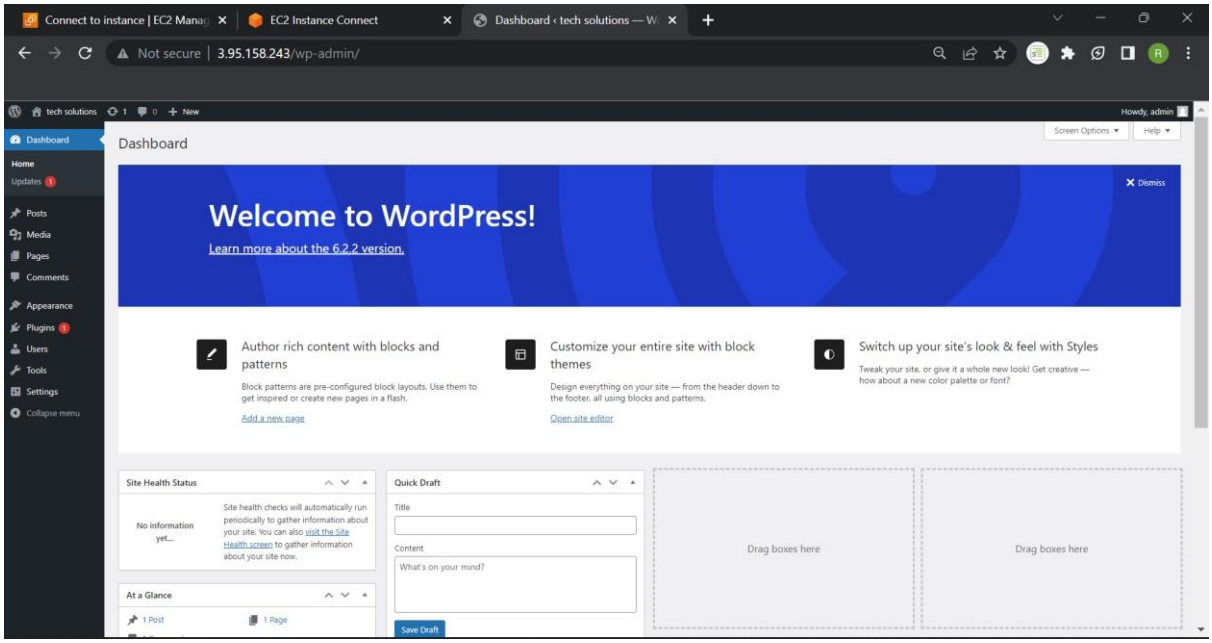
Type your username and click on install wordpress



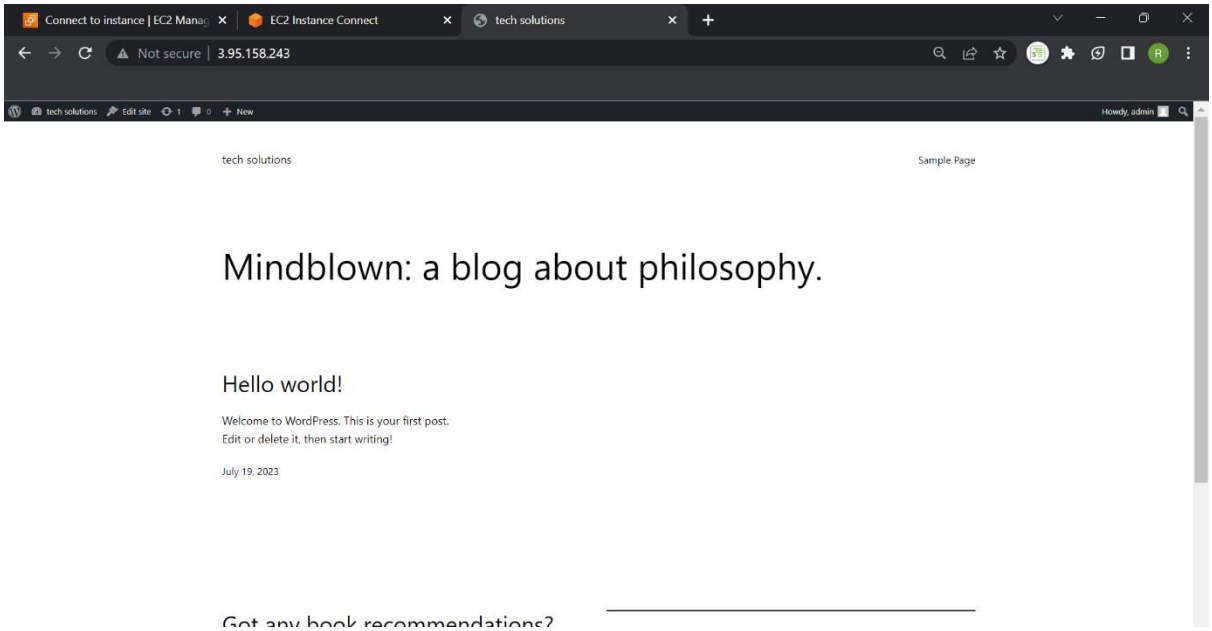
Click on login



Click on login



This is your dashboard



This your main website page.

To install WordPress on an AWS EC2 instance running Ubuntu, you can follow these steps:

Launch an EC2 Instance:

Log in to your AWS Management Console.

Navigate to the EC2 service and click on "Launch Instance."

Choose an Ubuntu AMI (Amazon Machine Image) for your instance. Select an instance type and configure other settings as needed.

Configure Security Group:

During the instance launch, configure the security group to allow inbound traffic on ports 80 (HTTP) and 443 (HTTPS) to allow web traffic.

Connect to the EC2 Instance:

Once the instance is launched, use SSH to connect to the EC2 instance securely. Use the .pem key provided during instance creation.

Update System Packages:

After connecting, update the system packages using the following commands:

sql

Copy code

sudo apt update

sudo apt upgrade

Install Apache Web Server:

Install Apache web server on the instance with the following command:

Copy code

sudo apt install apache2

Install MySQL Database:

Install MySQL to set up the database for WordPress with the following command:

Copy code

sudo apt install mysql-server

Secure MySQL Installation:

Run the MySQL secure installation script to set a root password and remove anonymous users:

Copy code

sudo mysql_secure_installation

Install PHP and Required Extensions:

Install PHP and necessary extensions with the following command:

lua

Copy code

sudo apt install php libapache2-mod-php php-mysql

Configure Apache for PHP:

Enable the PHP module in Apache and restart the server:

Copy code

sudo a2enmod php

sudo systemctl restart apache2

Download and Install WordPress:

Download the latest WordPress package and extract it:

arduino

Copy code

wget https://wordpress.org/latest.tar.gz

tar -xvf latest.tar.gz

Move the extracted files to the webserver root directory and give proper permissions:

bash

Copy code

sudo mv wordpress/* /var/www/html/

sudo chown -R www-data:www-data /var/www/html/

Create MySQL Database and User:

Log in to MySQL as the root user:

css

Copy code

sudo mysql -u root -p

Create a database and user for WordPress:

sql

Copy code

CREATE DATABASE wordpress;

CREATE USER 'wordpressuser'@'localhost' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpressuser'@'localhost';

FLUSH PRIVILEGES;

EXIT;

Configure WordPress:

Access your server's public IP address in a web browser.

Follow the WordPress setup wizard, providing the database information (database name, username, password) when prompted.

Complete Installation:

After completing the WordPress setup, you can log in to the WordPress admin dashboard and start building your website.

That's it! You now have WordPress installed on your AWS EC2 instance running Ubuntu. Remember to keep your instance and WordPress installation up to date with regular updates and security patches.