Aws(cloud)-monolithic architecture

Introduction:

Welcome to this report detailing the meticulous setup of MySQL and WordPress on an Ubuntu EC2 instance. This document chronicles the systematic installation and configuration process aimed at establishing a resilient database management system and a versatile content management system for hosting websites or web applications.

The steps should be followed to create an instance

Log in to the AWS Management Console using your credentials. Ensure that you are logged in securely, and take necessary precautions to protect your login information.

Once logged in, navigate to the EC2 service dashboard. This is where you can manage your virtual servers, also known as instances.

through a series of steps to configure your instance.

Select an AMI that suits your requirements. This is essentially the operating system and software stack that will be pre-installed on your instance.

Choose the instance type based on your workload. Instances come in various sizes with different CPU, memory, storage, and networking capacities.

Configure additional settings such as instance details, network settings, and storage options. You can customize parameters like instance name, network settings (VPC, subnet, IP addressing), and storage volumes.

(Optionally) you can add tags to your instance for better organization and management. Tags are key-value pairs that help you identify resources easily.

Create or select an existing security group. This acts as a virtual firewall for your instance, controlling inbound and outbound traffic.

Review your instance configuration to ensure everything is set up as desired. Once confirmed, click "Launch" to initiate the instance creation process.

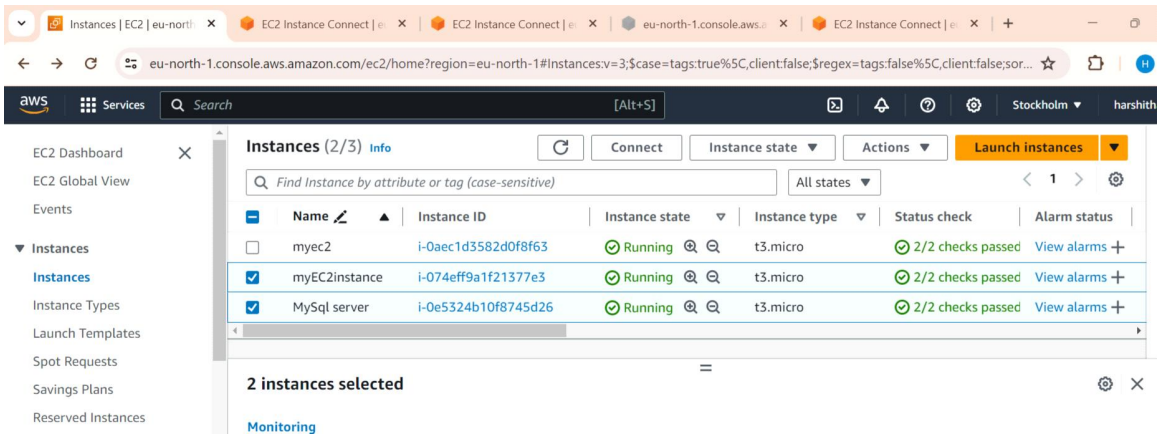
Select an existing key pair or create a new one. This key pair will be used to securely connect to your instance via SSH (for Linux instances) or RDP (for Windows instances).

After selecting the key pair, click "Launch Instances" to finalize the process. Your instance will now be provisioned and launched.

Once the instance is running, you can access it via SSH (for Linux) or RDP (for Windows) using the appropriate credentials and the private key associated with your key pair eating an instance on a cloud platform like AWS can be done as follows:

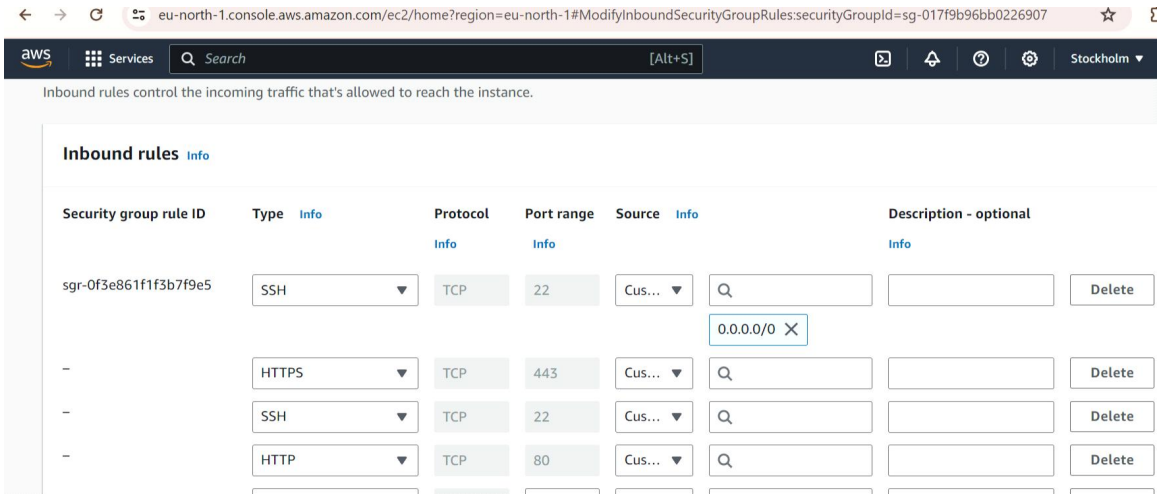
THE UPCOMING SHOTS SHOWS THE STEPS TO BE FOLLOWED

Select all requirements



Launch you instance as shown in above screenshot

Later it will display like this which is connected to cmd prompt of AWS



Installation and Configuration of MySQL:

Installation of MySQL Server and Client: Executed commands to update package index and install MySQL server and client, ensuring the latest version is obtained.

Run the mysql_ secure_installation script to enhance security by setting a root password, removing anonymous users, disallowing root login remotely, and removing the test database and access to it.

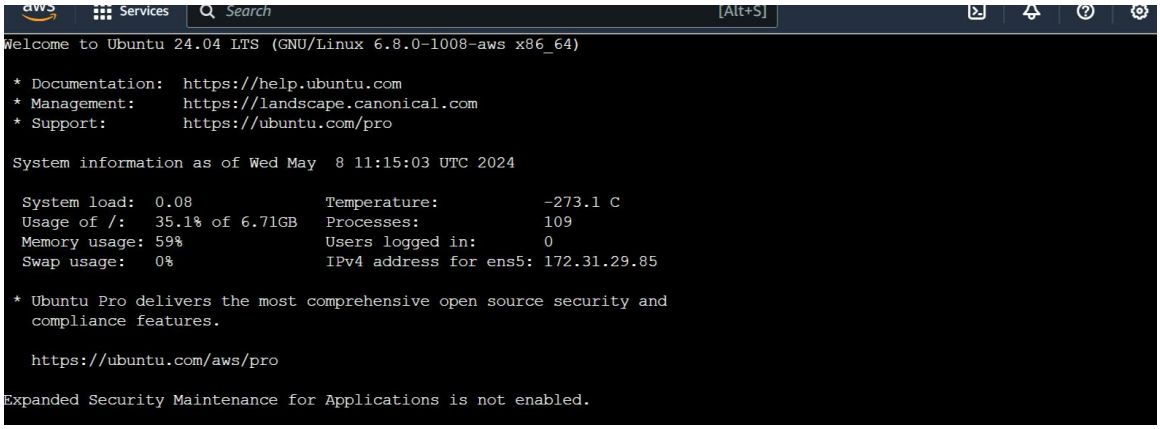
Logged into MySQL shell, created a database named wordpress', created a MySQL user with privileges on the 'wordpress' database, and verified the successful creation of the database and user.

The installation and configuration process were guided by prior knowledge of Linux server administration and MySQL management. Additionally, the official documentation of MySQL provided valuable insights and best practices during the setup process.

Use the command

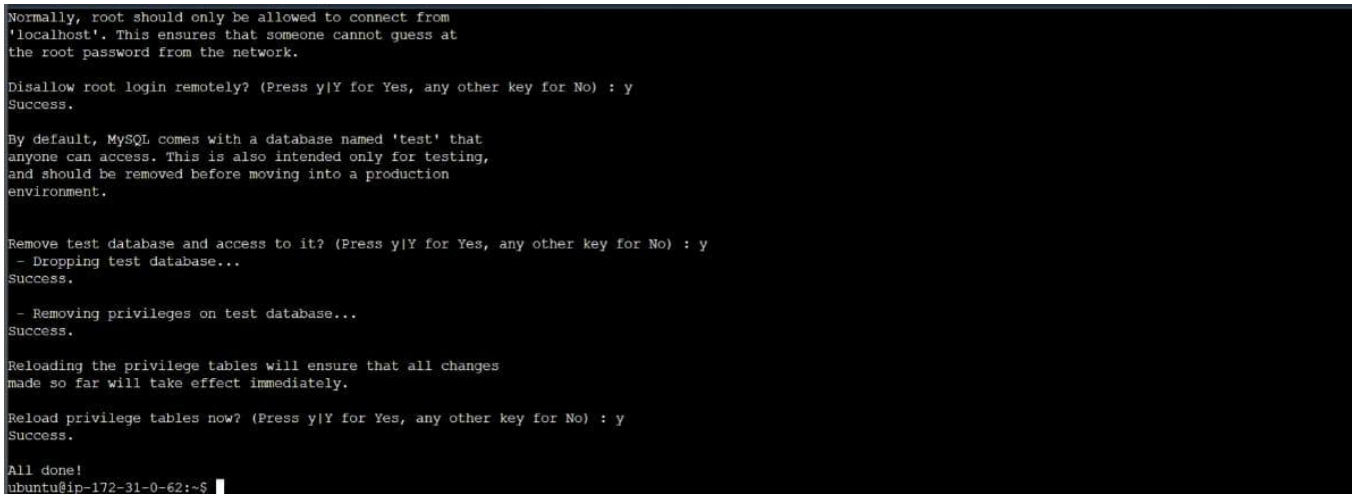
sudo apt update

sudo apt install mysql-server mysql-client (to install)



Secure mysql installation

sudo mysql_secure_installation



show these steps for secure installation

Set root password

Remove anonymous users

Disallow root login remotely

Remove test database and access to it

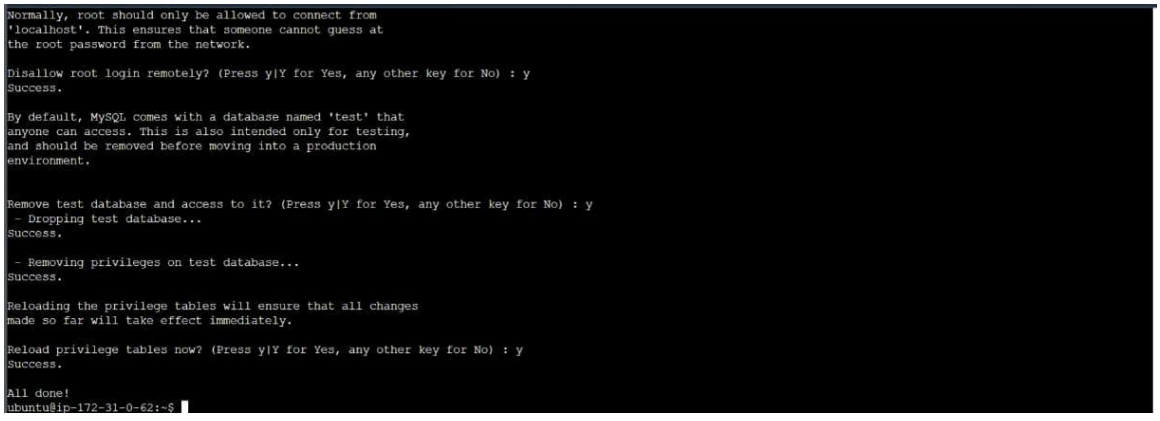
Reload privilege tables now

Create a MySQL Database and User for WordPress

Log into the MySQL shell as the root user

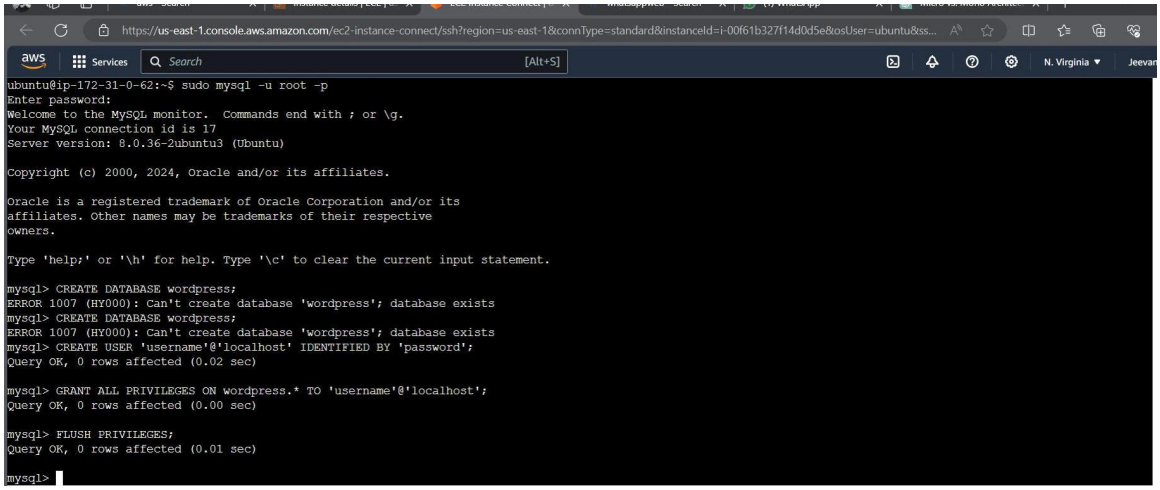
sudo mysql -u root -p

Enter the root password when prompted



Once logedin create a DB

CREATE DATABASE wordpress;



Next follow the steps

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

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

FLUSH PRIVILEGES;

After this exit command should be used

EXIT;

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.36-2ubuntu3 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE DATABASE wordpress;
ERROR 1007 (HY000): Can't create database 'wordpress'; database exists
mysql> CREATE DATABASE wordpress;
ERROR 1007 (HY000): Can't create database 'wordpress'; database exists
mysql> CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'username'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> EXIT;
Bye
```

Show databases;

```
Bye
ubuntu@ip-172-31-0-62:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.36-2ubuntu3 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
5 rows in set (0.01 sec)

mysql>
```

Install Apache Web Server, PHP, and Required Extensions

sudo apt update

```
mysql> EXIT;
Bye
ubuntu@ip-172-31-0-62:~$ sudo apt update
sudo apt install apache2 php libapache2-mod-php php-mysql
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.1).
php is already the newest version (2:8.3+93ubuntu2).
libapache2-mod-php is already the newest version (2:8.3+93ubuntu2).
php-mysql is already the newest version (2:8.3+93ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
ubuntu@ip-172-31-0-62:~$
```

sudo apt install apache2 php libapache2-mod-php php-mysql

Download and Configure WordPress

cd /tmp

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

```
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/site-icon.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
ubuntu@ip-172-31-0-62:/tmp$ sudo mv wordpress/* /var/www/html/
mv: cannot overwrite '/var/www/html/wp-admin': Directory not empty
mv: cannot overwrite '/var/www/html/wp-content': Directory not empty
mv: cannot overwrite '/var/www/html/wp-includes': Directory not empty
ubuntu@ip-172-31-0-62:/tmp$
```

Extract the downloaded WordPress archive:

tar -xvzf latest.tar.gz

Move the extracted WordPress files to the Apache document root directory (/var/www/html):

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

Set appropriate permissions on the WordPress directory:

sudo chown -R www-data:www-data /var/www/html/ sudo chmod -R 755 /var/www/html/

Configure WordPress to Use MySQL Database

Rename:

cd /var/www/html

sudo mv wp-config-sample.php wp-config.php

Edit:

sudo nano wp-config.php

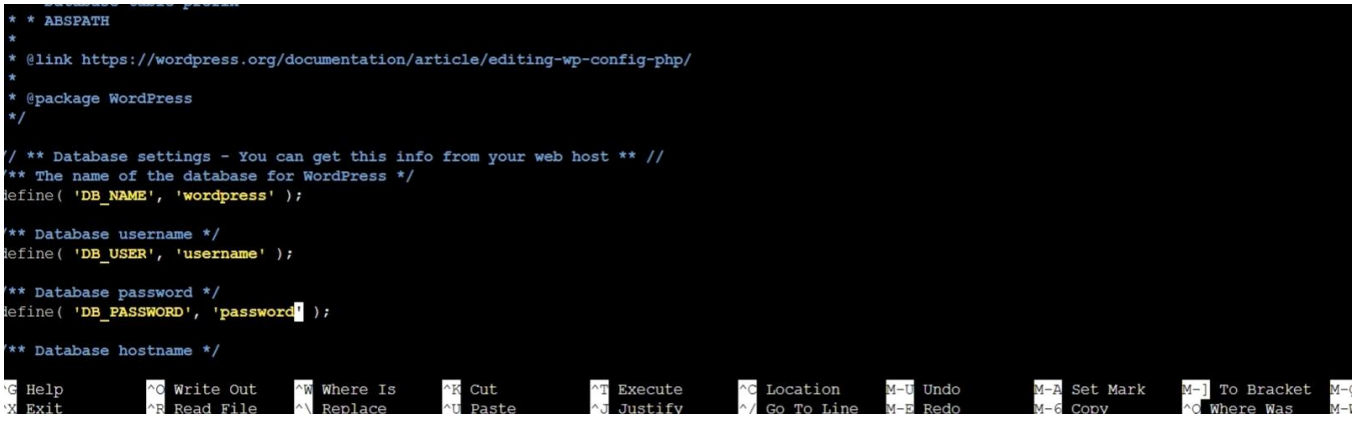
Update the following lines with your MySQL database information:

define('DB_NAME', 'wordpress');

define('DB_USER', 'username');

define('DB_PASSWORD', 'password');

define('DB_HOST', 'localhost');



Authentication Keys And Salts

define('AUTH_KEY', 'put your unique phrase here');

define('SECURE_AUTH_KEY', 'put your unique phrase here');

define('LOGGED_IN_KEY', 'put your unique phrase here');

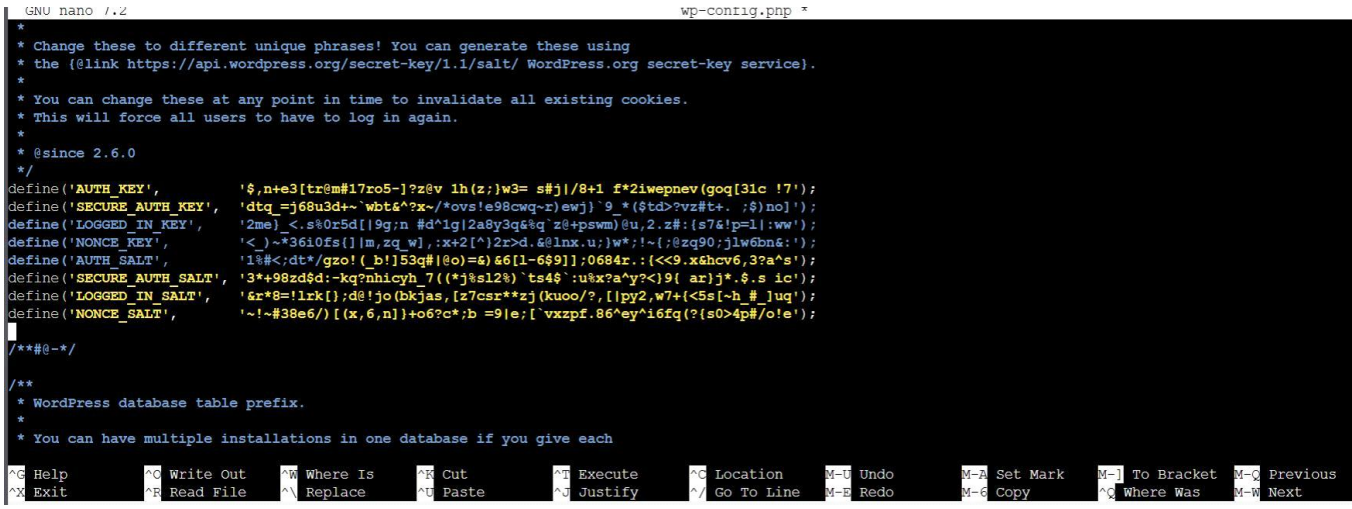
define('NONCE_KEY', 'put your unique phrase here');

define('AUTH_SALT', 'put your unique phrase here');

define('SECURE_AUTH_SALT', 'put your unique phrase here');

define('LOGGED_IN_SALT', 'put your unique phrase here');

defi

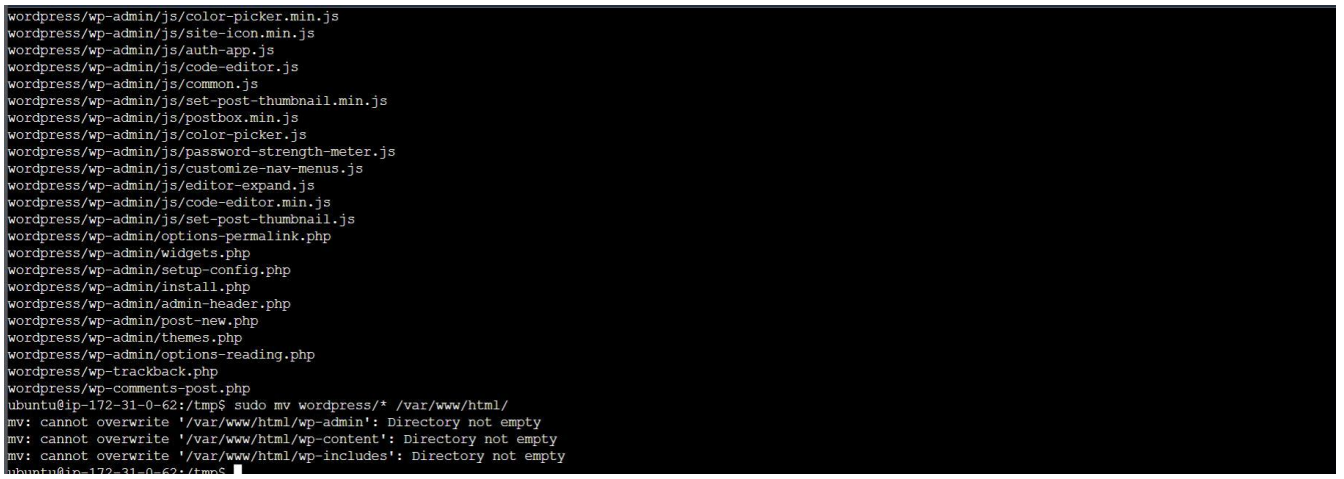


ne('NONCE_SALT', 'put your unique phrase here');

Set Up Virtual Hosts to Serve WordPress

Create a new virtual host configuration file for WordPress:

sudo nano /etc/apache2/sites-available/wordpress.conf



Paste the following configuration into the file:

<VirtualHost *:80>

ServerAdmin admin@example.com

DocumentRoot /var/www/html

ServerName your_domain.com (give your domain address here)

ServerAlias www.your_domain.com (give your domain address here)

<Directory /var/www/html>

Options FollowSymLinks

AllowOverride All

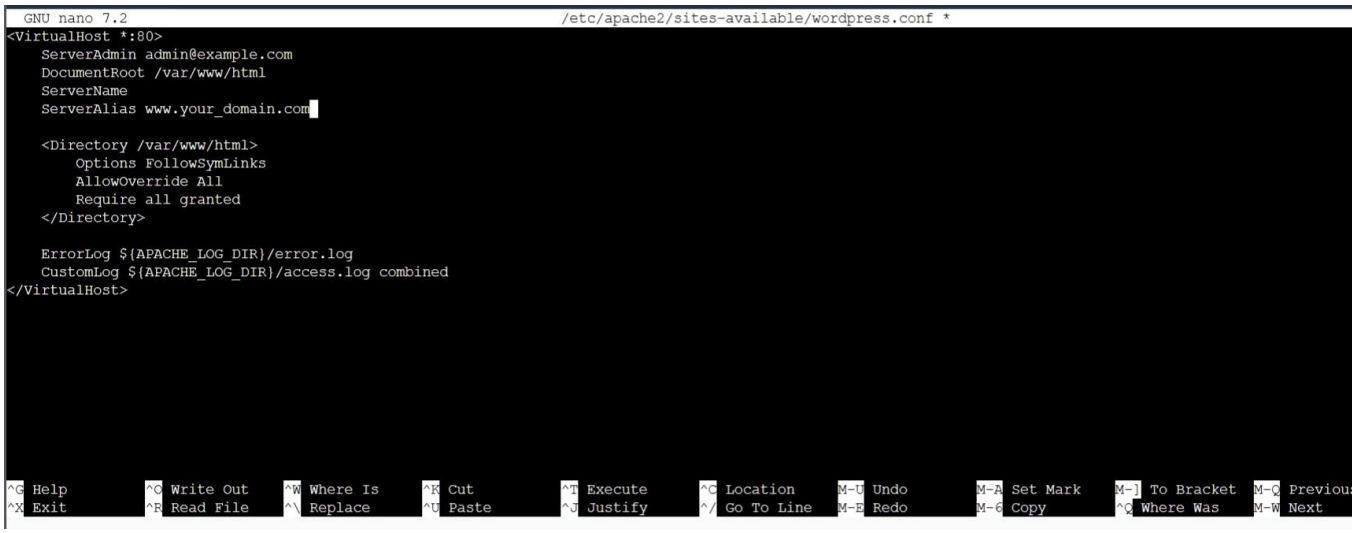
Require all granted

</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log

CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>



Enable the new virtual host configuration and disable the default one:

sudo a2ensite wordpress.conf

sudo a2dissite 000-default.conf

```
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/site-icon.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
ubuntu@ip-172-31-0-62:/tmp$ sudo mv wordpress/* /var/www/html/
mv: cannot overwrite '/var/www/html/wp-admin': Directory not empty
mv: cannot overwrite '/var/www/html/wp-content': Directory not empty
mv: cannot overwrite '/var/www/html/wp-includes': Directory not empty
```

Restart Apache for the changes to take effect:

sudo systemctl restart apache2

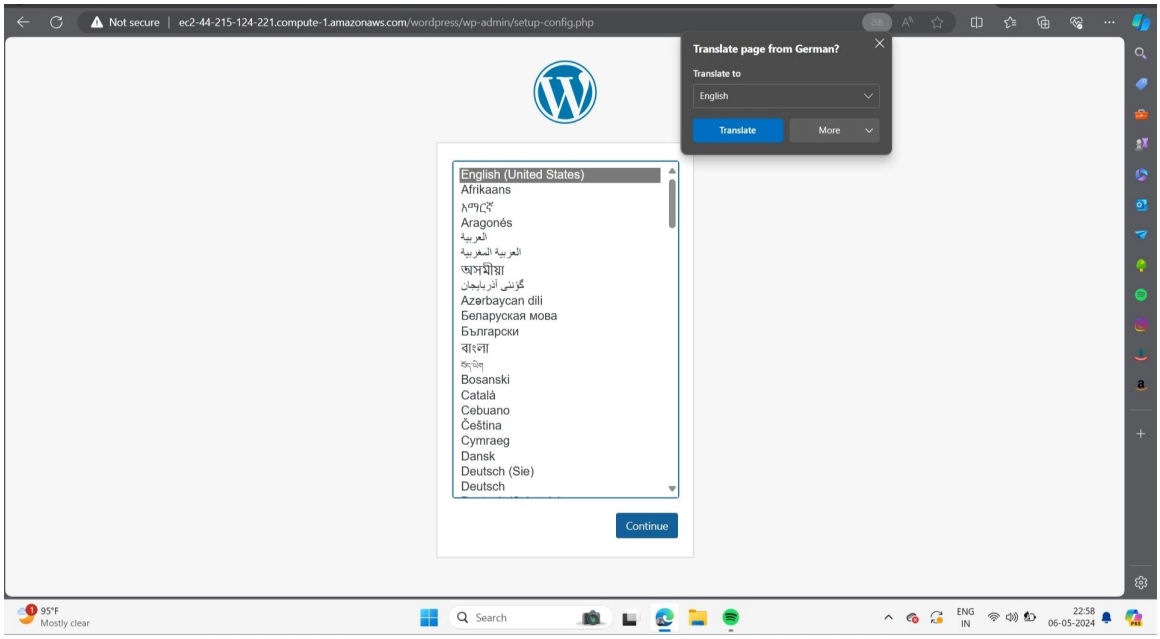

```
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/site-icon.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
ubuntu@ip-172-31-0-62:/tmp$ sudo mv wordpress/* /var/www/html/
mv: cannot overwrite '/var/www/html/wp-admin': Directory not empty
mv: cannot overwrite '/var/www/html/wp-content': Directory not empty
mv: cannot overwrite '/var/www/html/wp-includes': Directory not empty
```


Finally paste your Dns or instead paste IP address of your created instance

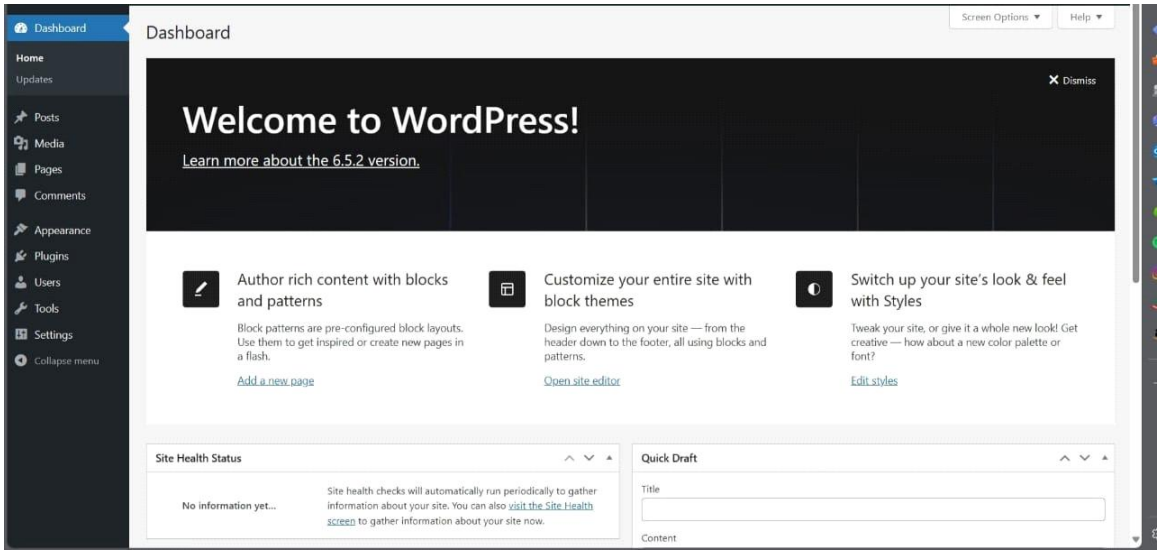
In my case ip address is : http://3.230.128.33/\wordpress

Further the wordpress page will get displayed and user can allow to do further steps: In my case I will adding images what I have done

1)



finally the page saying welcome to wordpress will be displayed



Here is the deployed page as welcome page in wordpress

Further microservices:

How to Deploy a Microservices Application on AWS EC2

Installing instances on AWS EC2 Instances and install on both instances how to initialize the master node and join the worker node to the cluster.

Deploying the Microservices Application to deploy a microservices application that consists of a MongoDB database and a taskmaster service. The taskmaster service is a web app that allows you to create and manage tasks. The app is written in Flask and uses MongoDB as the databaseBy the end of this blog post, you will have a working microservices application running on AWS EC2 using Kubernetes. You will also learn some important concepts and commands related to Kubernetes, such as pods, services, deployments, etc.

Setting up AWS EC2 Instances

In this , we will create two t3.micro instances on AWS EC2, one for the master node and one for the worker node. We will also configure the security groups and SSH keys for these instances.

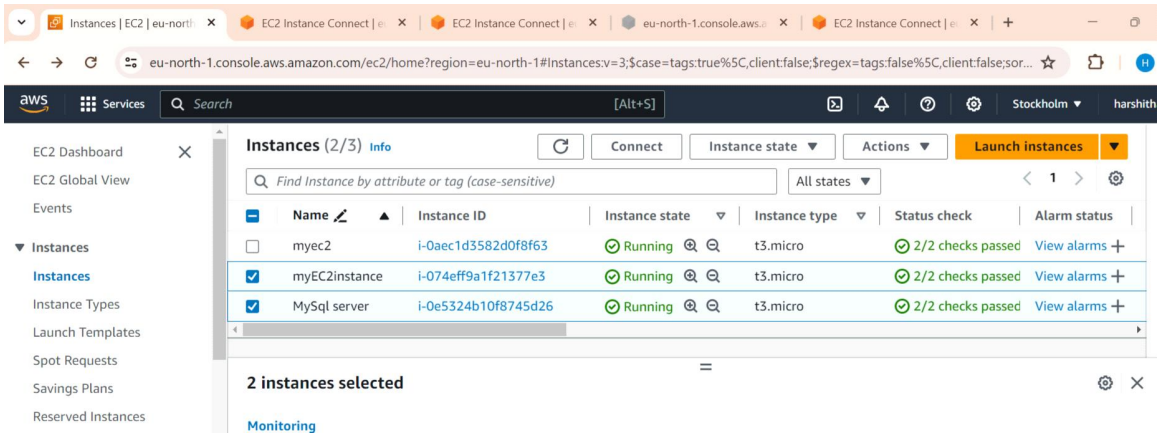
Creating Two t3.micro Instances

To create two t3.micro instances on AWS EC2, the following steps:

Log in to your AWS console and go to the EC2 dashboard.

Click on the Launch Instance button.

Choose Ubuntu Server



(HVM) as the AMI.

Choose t3.micro as the instance type.

Click on Next until you reach the Configure Security Group page.

Create a new security group with the following rules:

Allow SSH

Allow TCP

Allow TCP

Allow TCP

Allow all traffic from within the security group

7. Click on the Review and Launch button.

8. Create a new key pair or use an existing one and download it.

9. Click on the Launch Instances button.

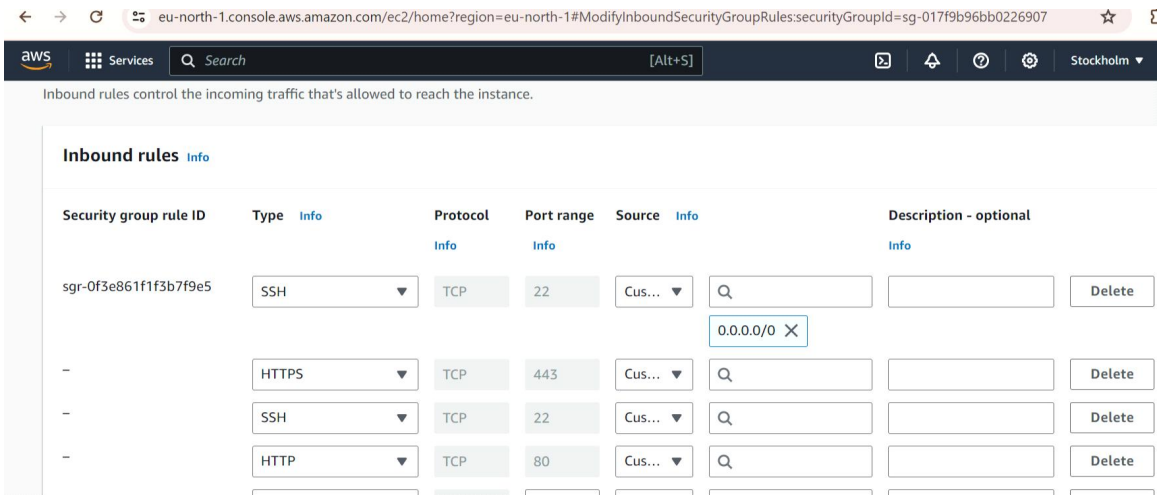
To configure the security groups and SSH keys for our instances, follow these steps:

Go to the Instances page on the EC2 dashboard and select one of your instances.

Click on Actions > Networking > Change Security Groups.

Select the security group that you created in the previous step and click on the Assign Security Groups button.

Repeat the same steps for the other instance.



This shows that you have successfully assigned the same security group to both instances

Installing Docker on Both Instances

Before we install , we need to install Docker on both instances. Docker is a software that allows us to run containers, which are isolated environments that contain our applications and their dependencies.

To install Docker on both instances, follow these steps:

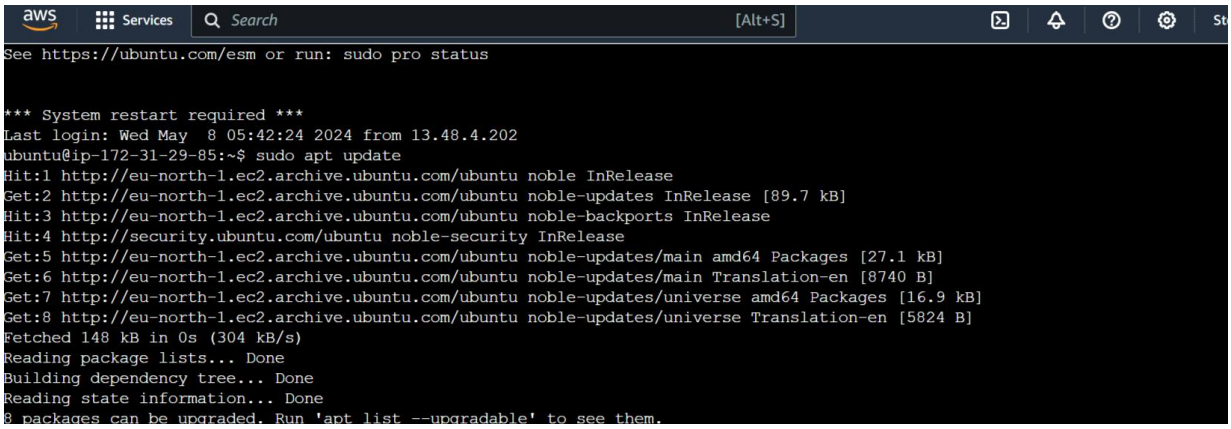
SSH into one of your instances by running:

ssh -i ~/.ssh/mykey.pem ubuntu@<instance-ip>

Replace ~/.ssh/mykey.pem with the path to your key pair file and <instance-ip> with the public IPv4 address of your instance.Replace ~/.ssh/mykey.pem with the path to your key pair file and <instance-ip> with the public IPv4 address of your instance.

update the packages:

sudo apt update



install docker by running:

sudo apt install docker.io-y

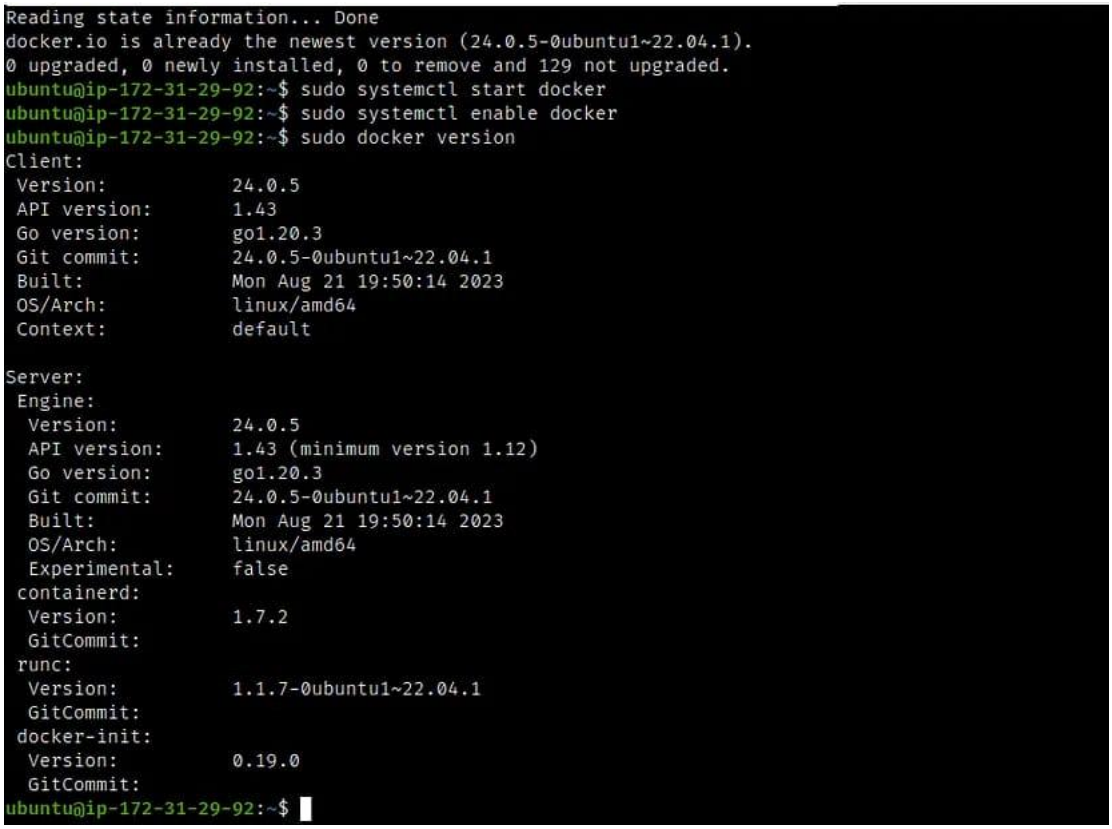
start and enable the docker service by running:

sudo systemctl start docker

sudo systemctl enable docker

verify that docker installing and running:

sudo docker version



have to follow the step by step process in micro services

later,

Installation and Configuration of MySQL:

Installation of MySQL Server and Client: Executed commands to update package index and install MySQL server and client, ensuring the latest version is obtained.

Run the mysql_ secure_installation script to enhance security by setting a root password, removing anonymous users, disallowing root login remotely, and removing the test database and access to it.

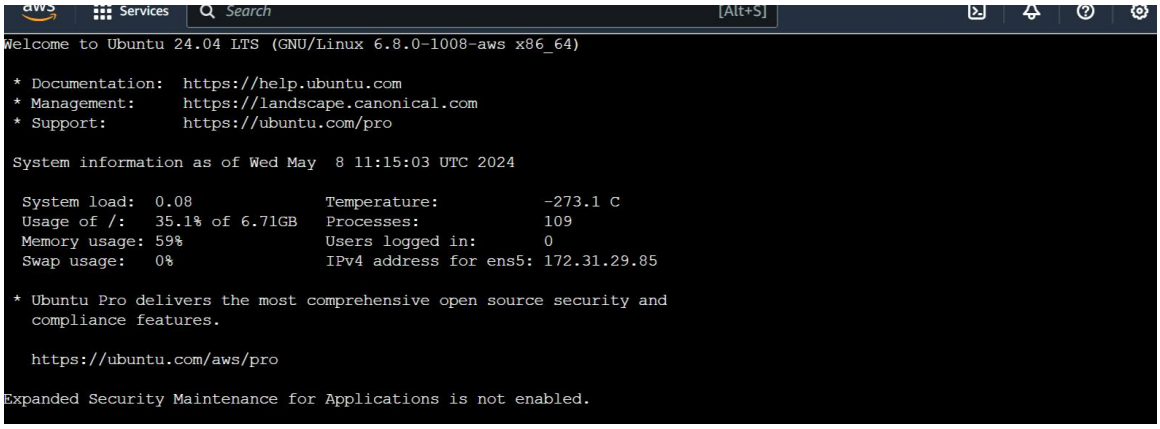
Logged into MySQL shell, created a database named wordpress', created a MySQL user with privileges on the 'wordpress' database, and verified the successful creation of the database and user.

The installation and configuration process were guided by prior knowledge of Linux server administration and MySQL management. Additionally, the official documentation of MySQL provided valuable insights and best practices during the setup process.

Use the command

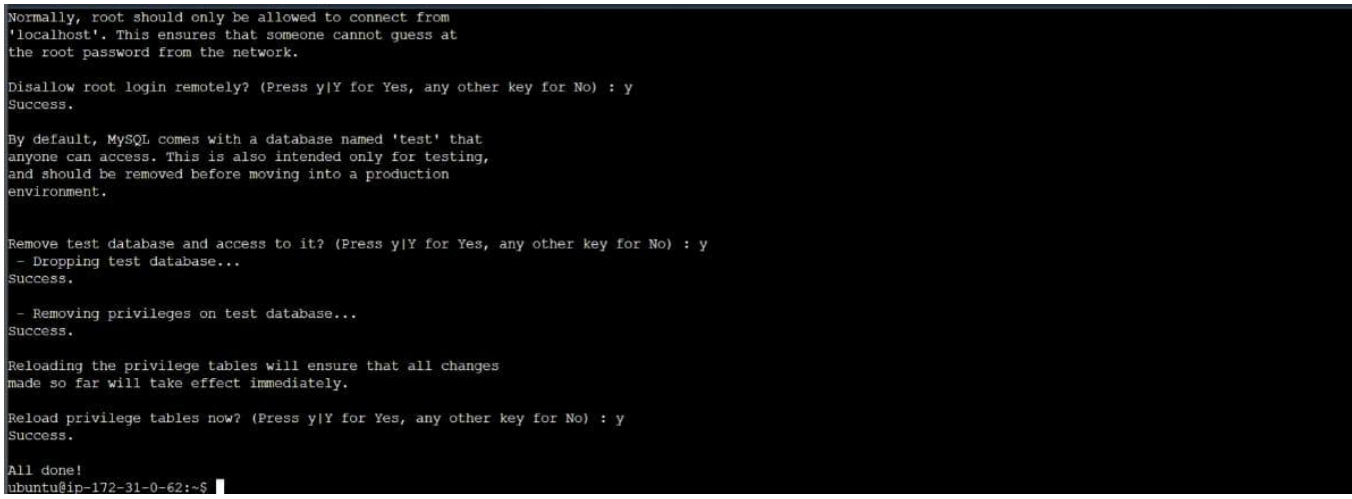
sudo apt update

sudo apt install mysql-server mysql-client (to install)



Secure mysql installation

sudo mysql_secure_installation



show these steps for secure installation

Set root password

Remove anonymous users

Disallow root login remotely

Remove test database and access to it

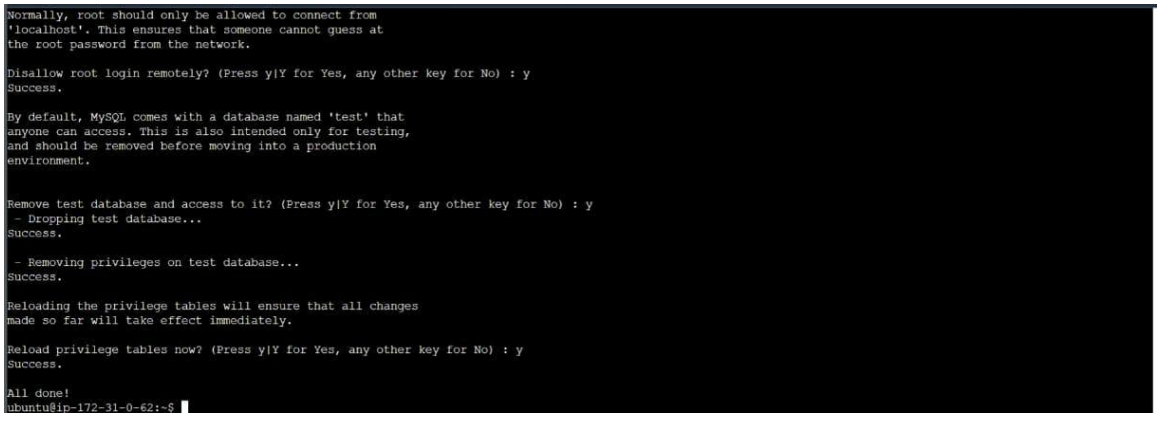
Reload privilege tables now

Create a MySQL Database and User for WordPress

Log into the MySQL shell as the root user

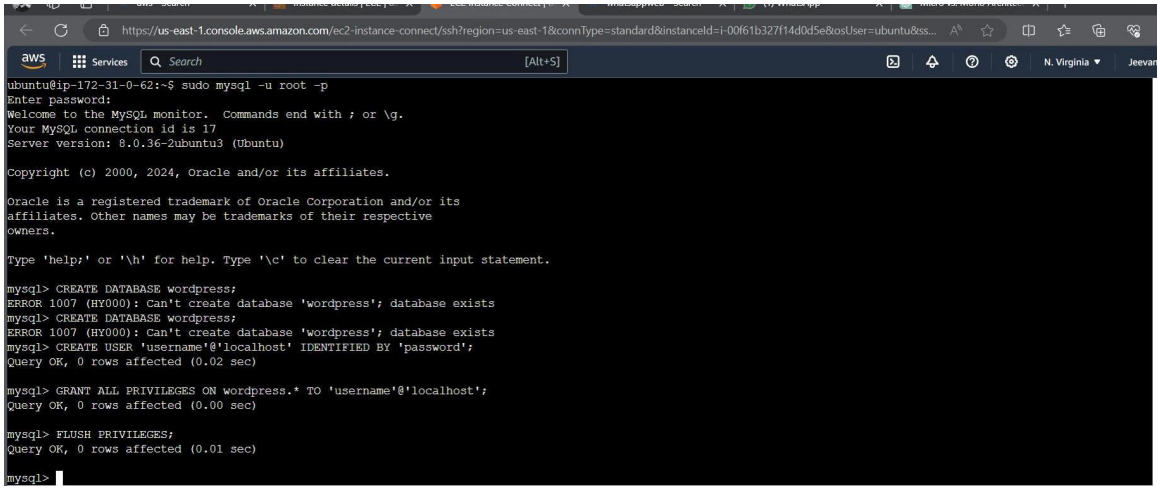
sudo mysql -u root -p

Enter the root password when prompted



Once logedin create a DB

CREATE DATABASE wordpress;



Next follow the steps

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

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

FLUSH PRIVILEGES;

After this exit command should be used

EXIT;

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.36-2ubuntu3 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE DATABASE wordpress;
ERROR 1007 (HY000): Can't create database 'wordpress'; database exists
mysql> CREATE DATABASE wordpress;
ERROR 1007 (HY000): Can't create database 'wordpress'; database exists
mysql> CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'username'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> EXIT;
Bye
```

Show databases;

```
Bye
ubuntu@ip-172-31-0-62:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.36-2ubuntu3 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
5 rows in set (0.01 sec)

mysql>
```

Install Apache Web Server, PHP, and Required Extensions

sudo apt update

```
mysql> EXIT;
Bye
ubuntu@ip-172-31-0-62:~$ sudo apt update
sudo apt install apache2 php libapache2-mod-php php-mysql
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.1).
php is already the newest version (2:8.3+93ubuntu2).
libapache2-mod-php is already the newest version (2:8.3+93ubuntu2).
php-mysql is already the newest version (2:8.3+93ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
ubuntu@ip-172-31-0-62:~$
```

sudo apt install apache2 php libapache2-mod-php php-mysql

Download and Configure WordPress

cd /tmp

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

```
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/site-icon.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
ubuntu@ip-172-31-0-62:/tmp$ sudo mv wordpress/* /var/www/html/
mv: cannot overwrite '/var/www/html/wp-admin': Directory not empty
mv: cannot overwrite '/var/www/html/wp-content': Directory not empty
mv: cannot overwrite '/var/www/html/wp-includes': Directory not empty
ubuntu@ip-172-31-0-62:/tmp$
```

Extract the downloaded WordPress archive:

tar -xvzf latest.tar.gz

Move the extracted WordPress files to the Apache document root directory (/var/www/html):

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

Set appropriate permissions on the WordPress directory:

sudo chown -R www-data:www-data /var/www/html/ sudo chmod -R 755 /var/www/html/

Configure WordPress to Use MySQL Database

Rename:

cd /var/www/html

sudo mv wp-config-sample.php wp-config.php

Edit:

sudo nano wp-config.php

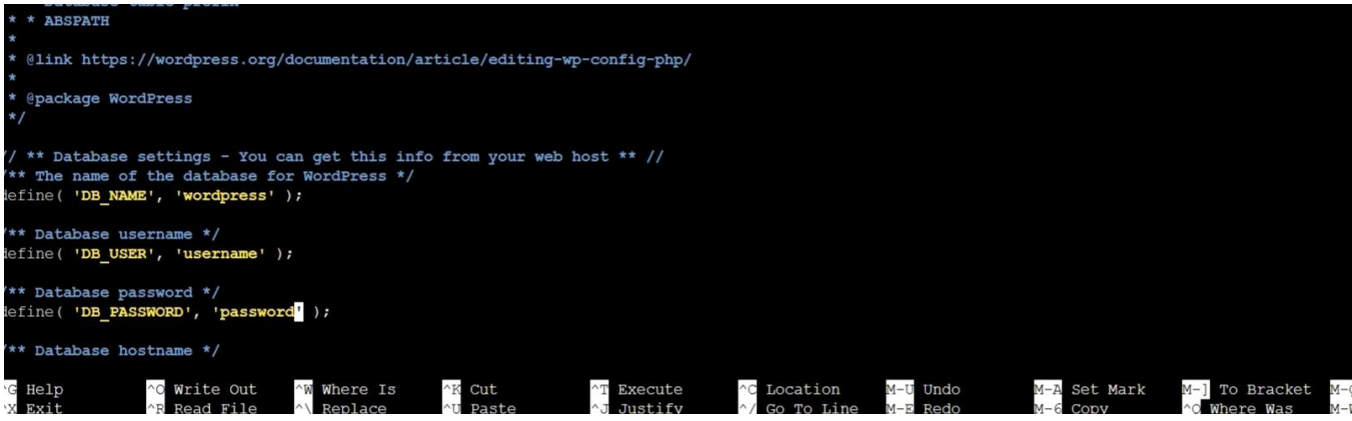
Update the following lines with your MySQL database information:

define('DB_NAME', 'wordpress');

define('DB_USER', 'username');

define('DB_PASSWORD', 'password');

define('DB_HOST', 'localhost');



Authentication Keys And Salts

define('AUTH_KEY', 'put your unique phrase here');

define('SECURE_AUTH_KEY', 'put your unique phrase here');

define('LOGGED_IN_KEY', 'put your unique phrase here');

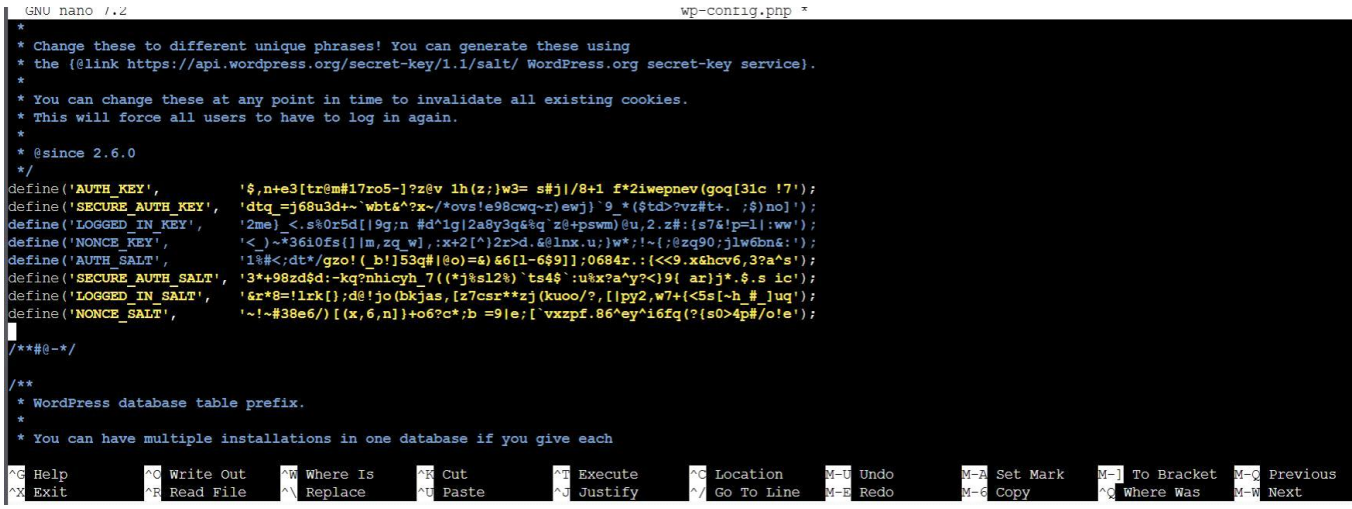
define('NONCE_KEY', 'put your unique phrase here');

define('AUTH_SALT', 'put your unique phrase here');

define('SECURE_AUTH_SALT', 'put your unique phrase here');

define('LOGGED_IN_SALT', 'put your unique phrase here');

defi

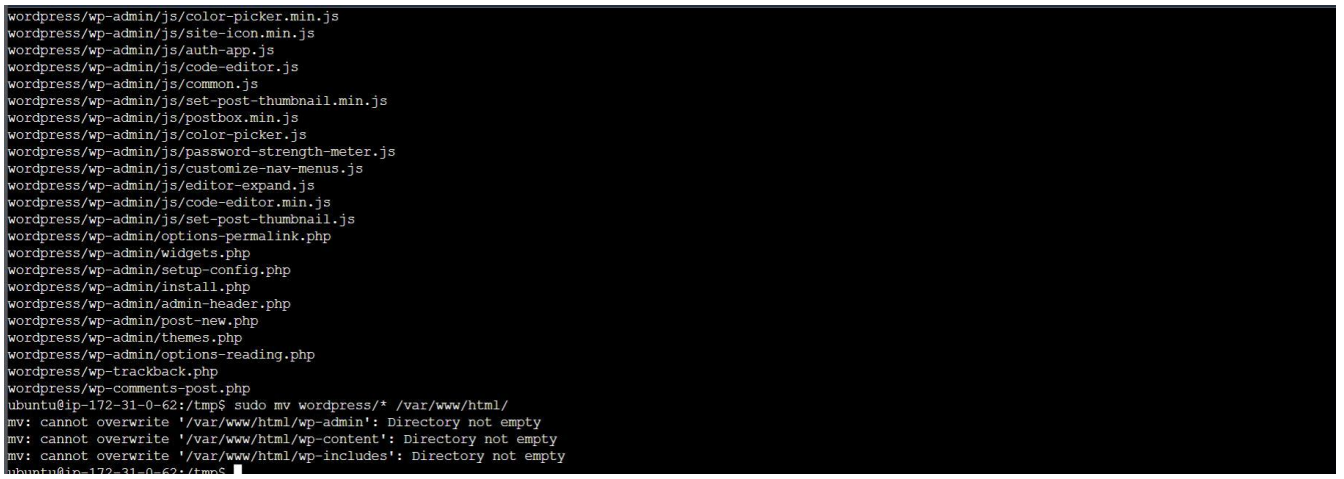


ne('NONCE_SALT', 'put your unique phrase here');

Set Up Virtual Hosts to Serve WordPress

Create a new virtual host configuration file for WordPress:

sudo nano /etc/apache2/sites-available/wordpress.conf



Paste the following configuration into the file:

<VirtualHost *:80>

ServerAdmin admin@example.com

DocumentRoot /var/www/html

ServerName your_domain.com (give your domain address here)

ServerAlias www.your_domain.com (give your domain address here)

<Directory /var/www/html>

Options FollowSymLinks

AllowOverride All

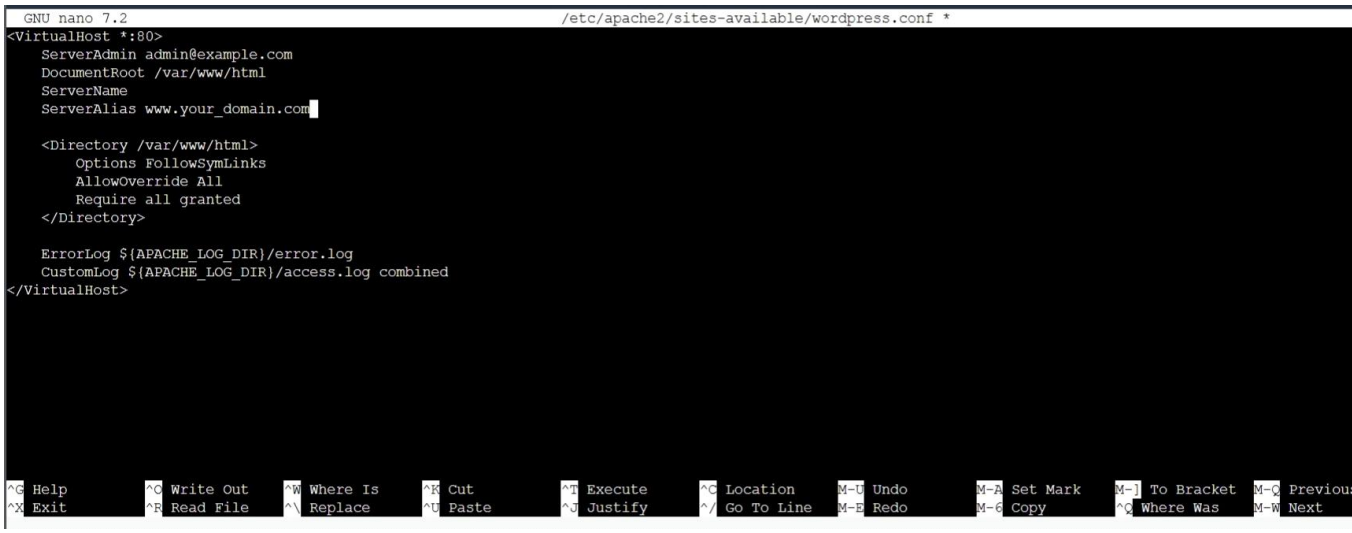
Require all granted

</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log

CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>



Enable the new virtual host configuration and disable the default one:

sudo a2ensite wordpress.conf

sudo a2dissite 000-default.conf

```
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/site-icon.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
ubuntu@ip-172-31-0-62:/tmp$ sudo mv wordpress/* /var/www/html/
mv: cannot overwrite '/var/www/html/wp-admin': Directory not empty
mv: cannot overwrite '/var/www/html/wp-content': Directory not empty
mv: cannot overwrite '/var/www/html/wp-includes': Directory not empty
ubuntu@ip-172-31-0-62:/tmp$
```

Restart Apache for the changes to take effect:

sudo systemctl restart apache2

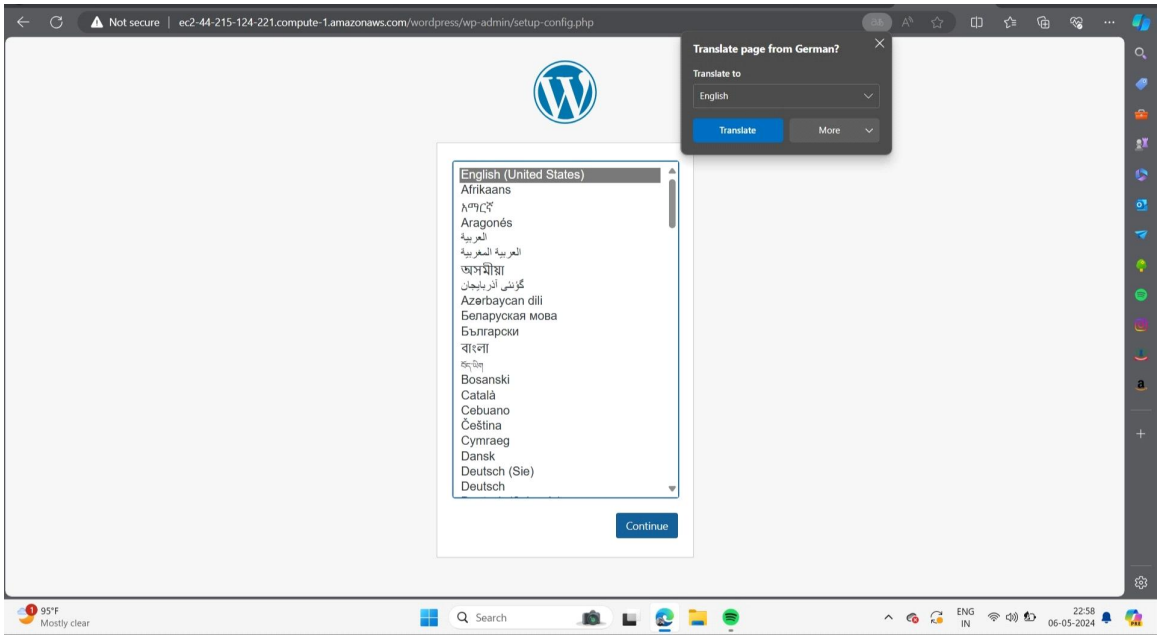

```
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/site-icon.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
ubuntu@ip-172-31-0-62:/tmp$ sudo mv wordpress/* /var/www/html/
mv: cannot overwrite '/var/www/html/wp-admin': Directory not empty
mv: cannot overwrite '/var/www/html/wp-content': Directory not empty
mv: cannot overwrite '/var/www/html/wp-includes': Directory not empty
ubuntu@ip-172-31-0-62:/tmp$
```


Finally paste your Dns or instead paste IP address of your created instance

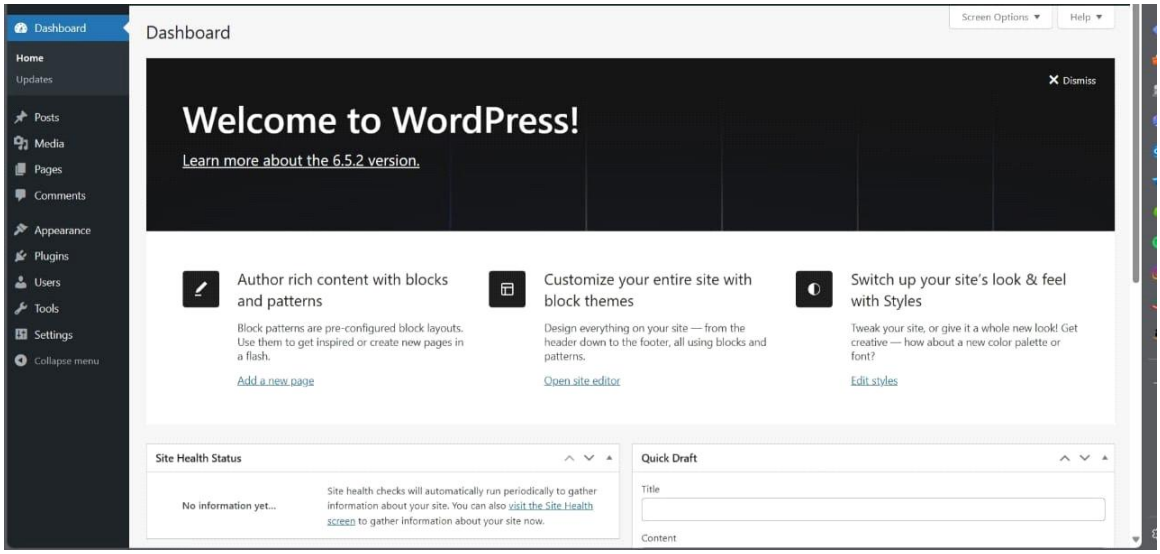
In my case ip address is http://3.230.128.33/ \wordpress

Further the wordpress page will get displayed and user can allow to do further steps: In my case I will adding images what I have done

1)



finally the page saying welcome to wordpress will be displayed



Here is the deployed page as welcome page in wordpress