

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.

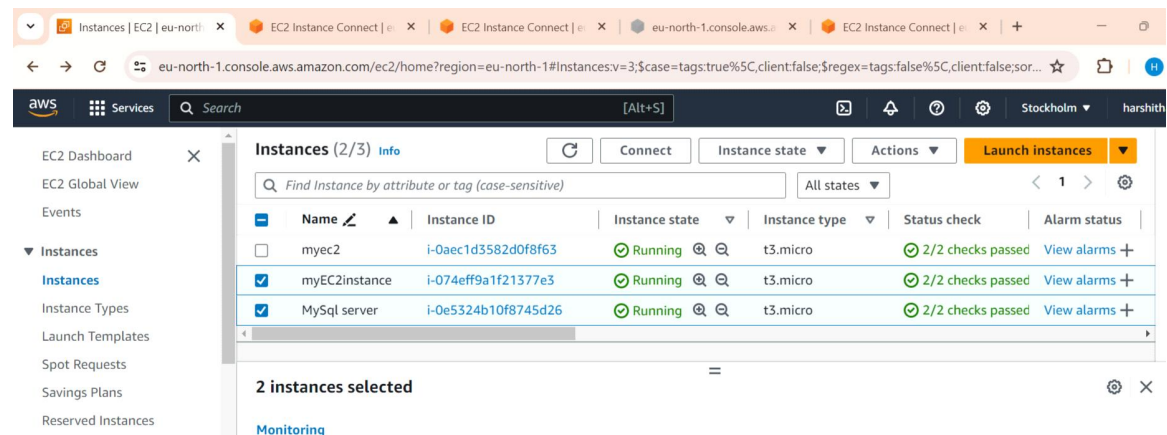
Select an existing key pair or create a new one. This key pair will be used to securely connect to your instance via SSH (for Linux instances) or RDP (for Windows instances).

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. Creating an instance on a cloud platform like AWS can be done as follows:

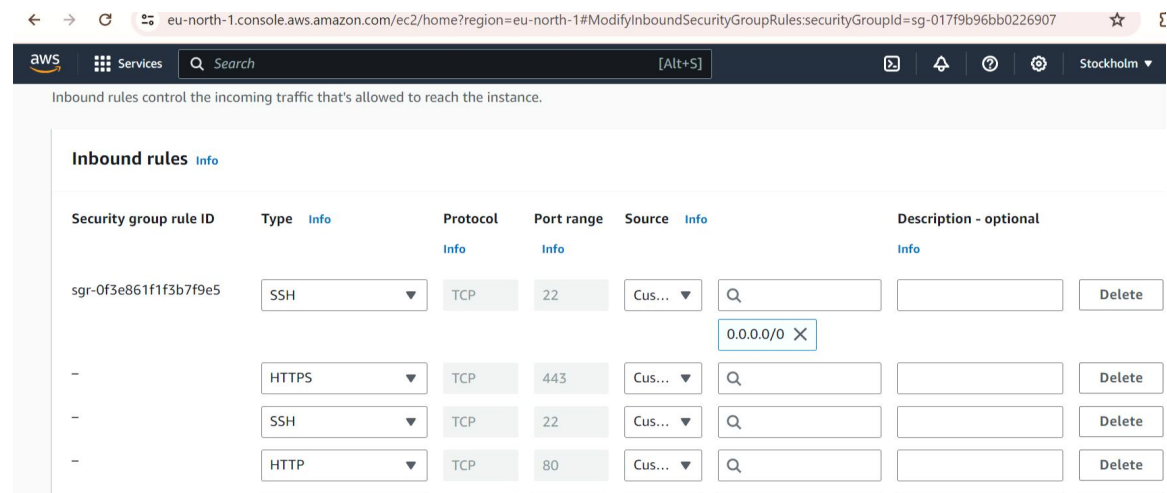
THE UPCOMING SHOTS SHOWS THE STEPS TO BE FOLLOWED

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.

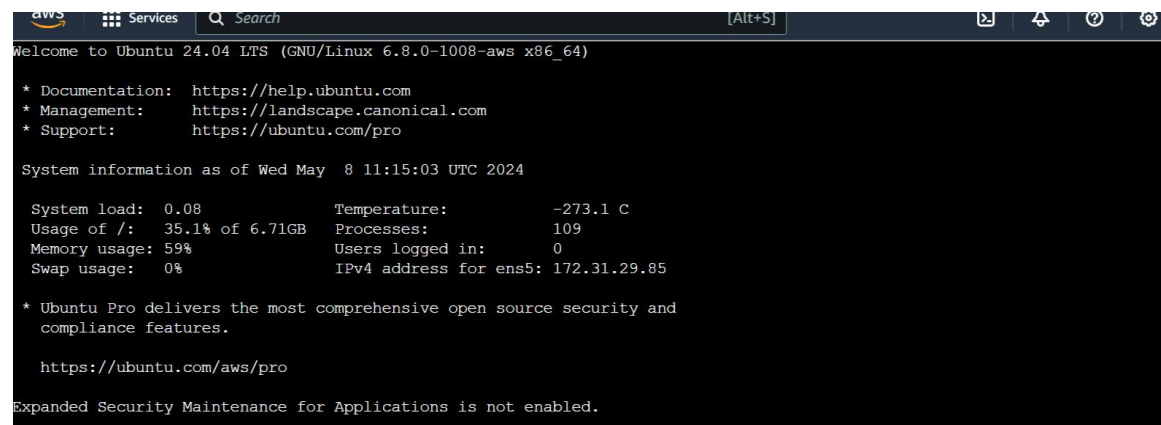
Logged into MySQL shell, created a database named 'wordpress', created a MySQL user with privileges on the 'wordpress' database, and verified the successful creation of the database and user.

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)
```



```
aws Services Search [Alt+S]
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of Wed May 8 11:15:03 UTC 2024

System load:  0.08           Temperature:   -273.1 C
Usage of /:   35.1% of 6.71GB Processes:      109
Memory usage: 59%           Users logged in: 0
Swap usage:  0%             IPv4 address for ens5: 172.31.29.85

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.
```

Secure mysql installation

```
sudo mysql_secure_installation
```

```

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-172-31-0-62:~$

```

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

```

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

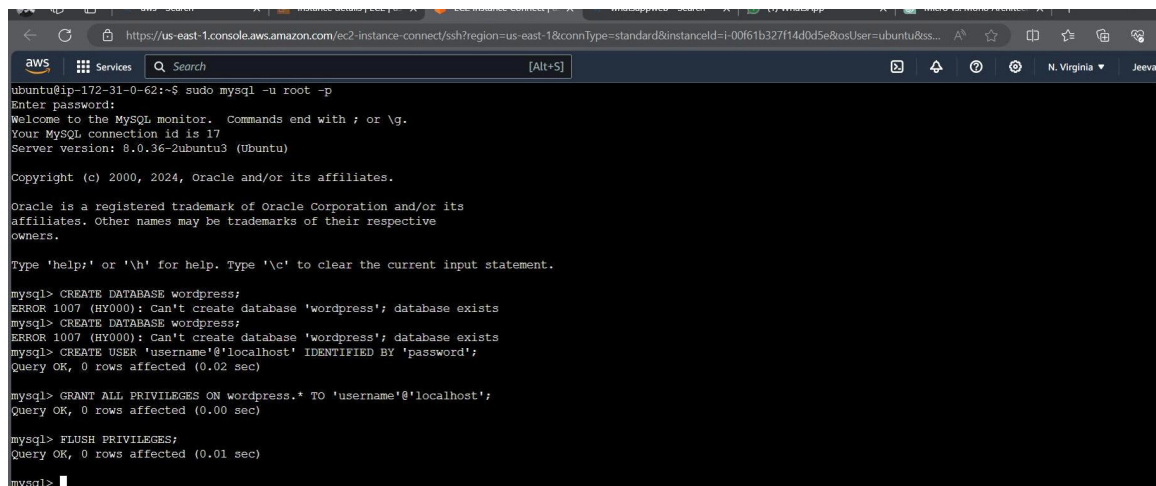
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-172-31-0-62:~$

```

Once logged in create a DB

`CREATE DATABASE wordpress;`

A screenshot of an AWS console terminal window. The terminal shows a user logging into an Ubuntu instance via SSH. The user runs the command 'sudo mysql -u root -p' to start the MySQL command-line interface. The MySQL prompt 'mysql>' is shown. The user enters the command 'CREATE DATABASE wordpress;'. The output is 'ERROR 1007 (HY000): Can't create database 'wordpress'; database exists'. The user then enters 'CREATE DATABASE wordpress;'. The output is 'ERROR 1007 (HY000): Can't create database 'wordpress'; database exists'. The user then enters 'CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';'. The output is 'Query OK, 0 rows affected (0.02 sec)'. The user then enters 'GRANT ALL PRIVILEGES ON wordpress.* TO 'username'@'localhost';'. The output is 'Query OK, 0 rows affected (0.00 sec)'. The user then enters 'FLUSH PRIVILEGES;'. The output is 'Query OK, 0 rows affected (0.01 sec)'. The terminal ends with the 'mysql>' prompt.

```
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 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>
```

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');
```

```
* * ABSOLUTE path to the database directory
*
* @link https://wordpress.org/documentation/article/editing-wp-config-php/
*
* @package WordPress
*/

/* Database settings - You can get this info from your web host */
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'username' );

/** Database password */
define( 'DB_PASSWORD', 'password' );

/** Database hostname */
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
```

```
GNU nano /./2 wp-config.php *
*
* Change these to different unique phrases! You can generate these using
* the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
*
* You can change these at any point in time to invalidate all existing cookies.
* This will force all users to have to log in again.
*
* @since 2.6.0
*/
define( 'AUTH_KEY',            '$_n+e3[tr@m#17ro5-]?z@v 1h(z;)w3= s#j|/8+1 f*2iwepnev(gog[3lc !7')';
define( 'SECURE_AUTH_KEY',    'dtq =j68u3d+~`wbt&^?x~/*ovs!e98cwq~r)ewj} `9_*($td>?vz#t+. ;$)no!')';
define( 'LOGGED_IN_KEY',      '2me) <.s%0r5d[|9g;n #d^lg|2a8y3q6%q`z0+pswm)@u,2.z#:{s76!p=l!:ww!');
define( 'NONCE_KEY',          '<_)~*36i0fs{||m,zq_w|,x+2(^)2r>d.60lnx.u;)w*;!~(:@zq90;jlw6bn&:')';
define( 'AUTH_SALT',          '1%#<;dt*/gzo!( _b!]53q#|@o)=6)66[1-6$9]];0684r.:{<<9.x6hcv6,3?a^s');
define( 'SECURE_AUTH_SALT',   '3*+98zd$d:-kq?nhicyh_7(((*j%sl2%)`ts4$ :u%x?a^y?<)9( ar)j*.$.s ic')';
define( 'LOGGED_IN_SALT',     '6r*8=!lrk[];d@!jo(bkjas,[z7csr**zj(kuoo/?,[|py2,w7+(<5s[-h_#_]uq')');
define( 'NONCE_SALT',         '~!~#38e6/) [(x,6,n)]+o6?c*;b =9|e;[ `vxzpf.86^ey^i6fq(?{s0>4p#o!e')';
/*#@-*/

/**
 * WordPress database table prefix.
 *
 * You can have multiple installations in one database if you give each
```

```
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
```

```
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:~$ 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:~$
```

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

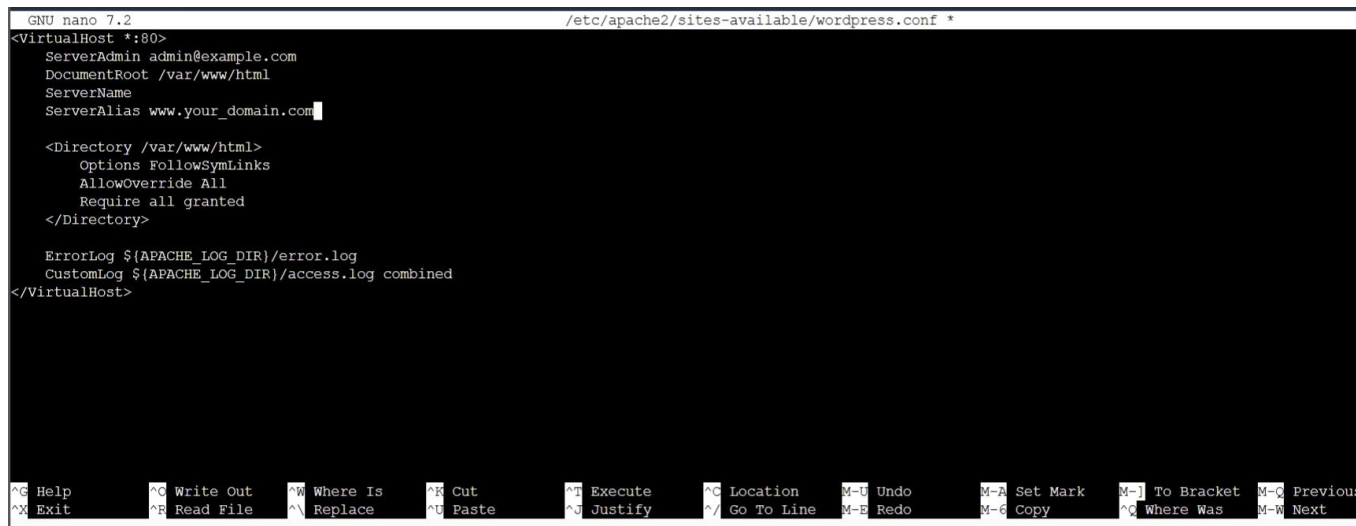
Require all granted

</Directory>

ErrorLog \${APACHE_LOG_DIR}/error.log

CustomLog \${APACHE_LOG_DIR}/access.log combined

</VirtualHost>



```
GNU nano 7.2 /etc/apache2/sites-available/wordpress.conf *
<VirtualHost *:80>
  ServerAdmin admin@example.com
  DocumentRoot /var/www/html
  ServerName
  ServerAlias www.your_domain.com

  <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>
```

Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket Previous
Exit Read File Replace Paste Justify Go To Line Redo Copy Where Was Next

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$

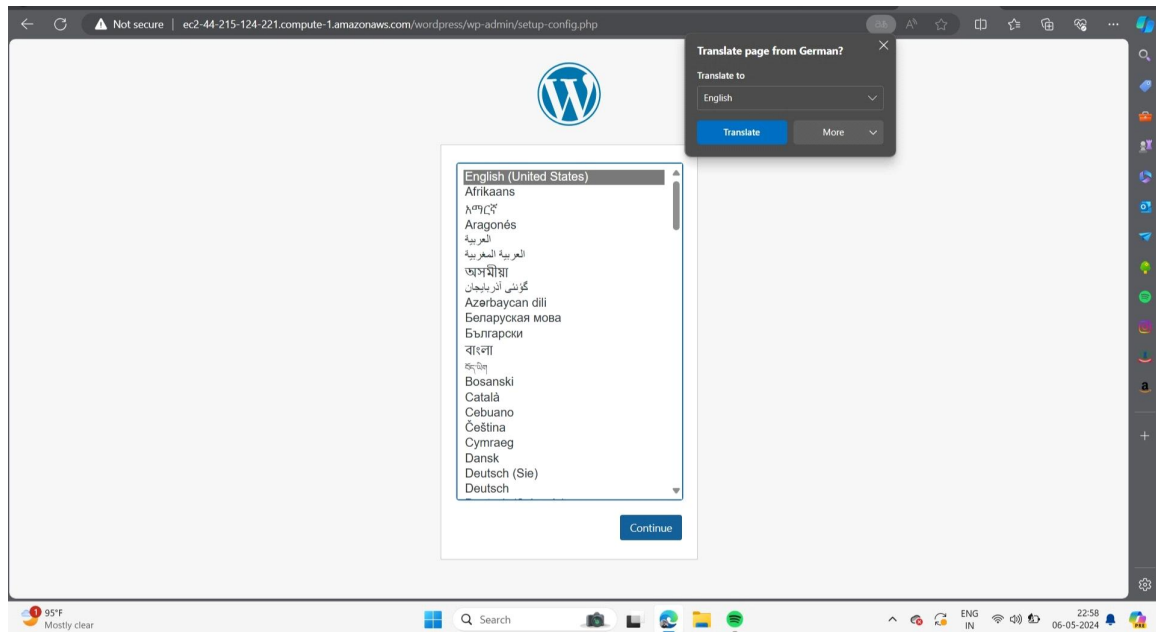
```

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

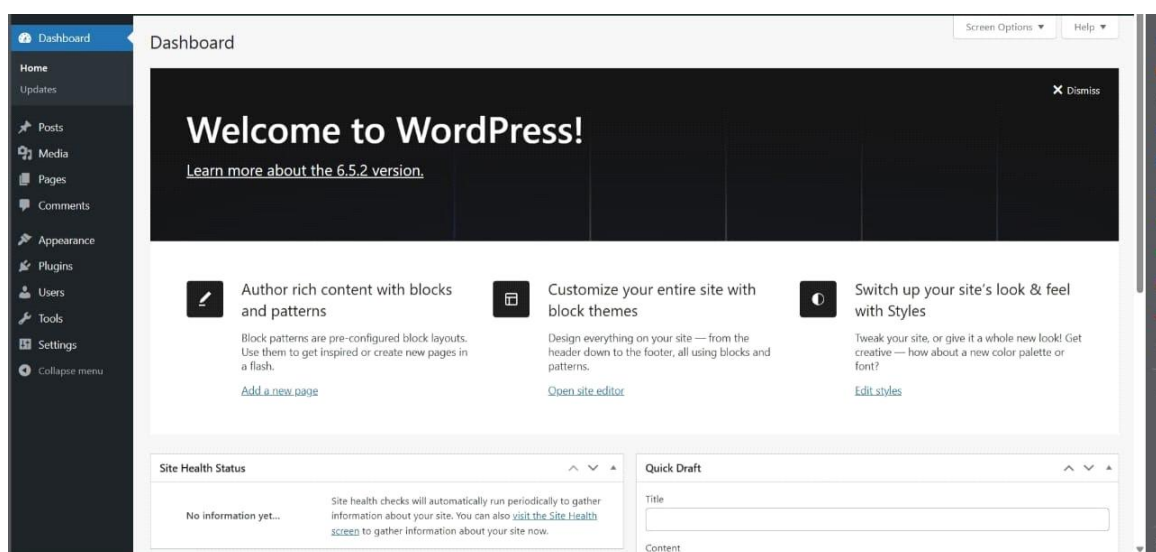
In my case ip address is : <http://3.230.128.33/\wordpress>

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

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 database
By 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.

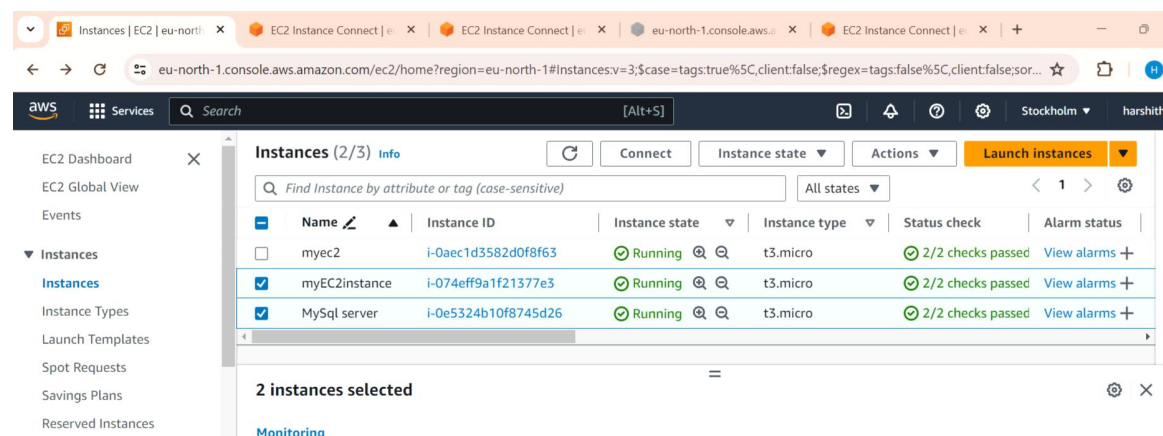
Creating Two t3.micro Instances

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.

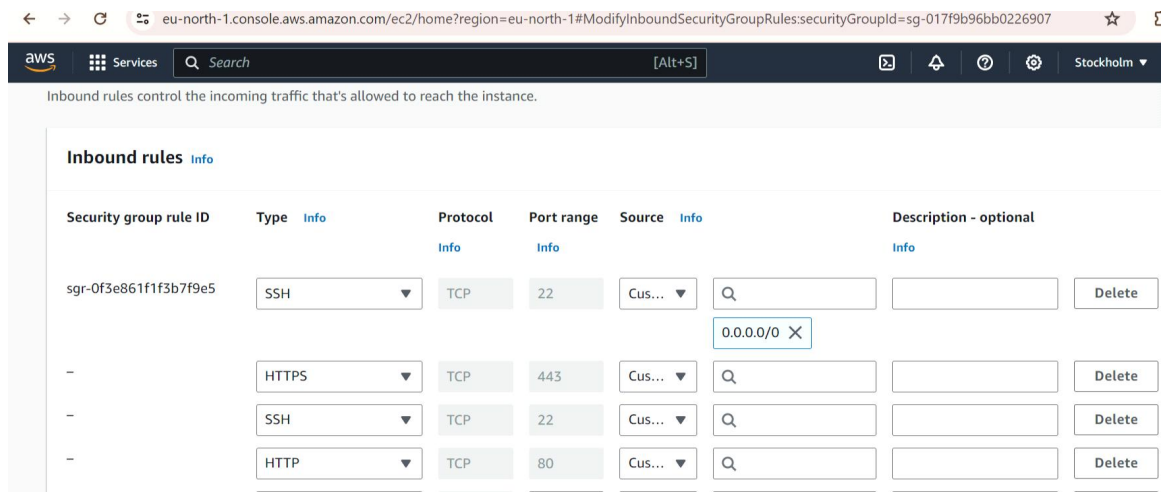
**To configure the security groups and SSH keys for our instances,
follow these steps:**

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:

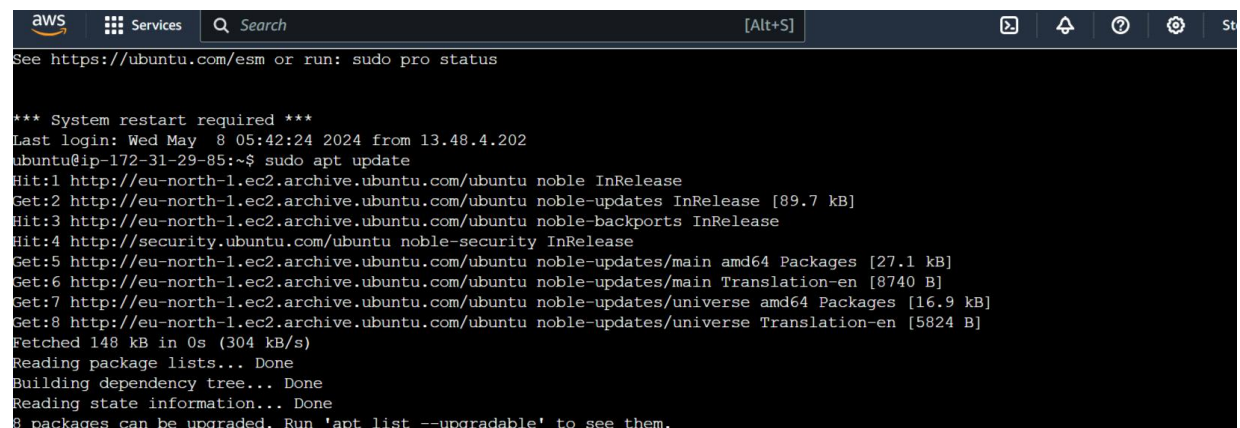
SSH into one of your instances by running:

```
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`



```
aws Services Search [Alt+S]
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Wed May  8 05:42:24 2024 from 13.48.4.202
ubuntu@ip-172-31-29-85:~$ sudo apt update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [89.7 kB]
Hit:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [27.1 kB]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [8740 B]
Get:7 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [16.9 kB]
Get:8 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [5824 B]
Fetched 148 kB in 0s (304 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
8 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

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

```
Reading state information... Done
docker.io is already the newest version (24.0.5-0ubuntu1~22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 129 not upgraded.
ubuntu@ip-172-31-29-92:~$ sudo systemctl start docker
ubuntu@ip-172-31-29-92:~$ sudo systemctl enable docker
ubuntu@ip-172-31-29-92:~$ sudo docker version
Client:
 Version:           24.0.5
 API version:       1.43
 Go version:        go1.20.3
 Git commit:        24.0.5-0ubuntu1~22.04.1
 Built:             Mon Aug 21 19:50:14 2023
 OS/Arch:           linux/amd64
 Context:           default

Server:
 Engine:
  Version:           24.0.5
  API version:       1.43 (minimum version 1.12)
  Go version:        go1.20.3
  Git commit:        24.0.5-0ubuntu1~22.04.1
  Built:             Mon Aug 21 19:50:14 2023
  OS/Arch:           linux/amd64
  Experimental:      false
 containerd:
  Version:           1.7.2
  GitCommit:
 runc:
  Version:           1.1.7-0ubuntu1~22.04.1
  GitCommit:
 docker-init:
  Version:           0.19.0
  GitCommit:
```

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.

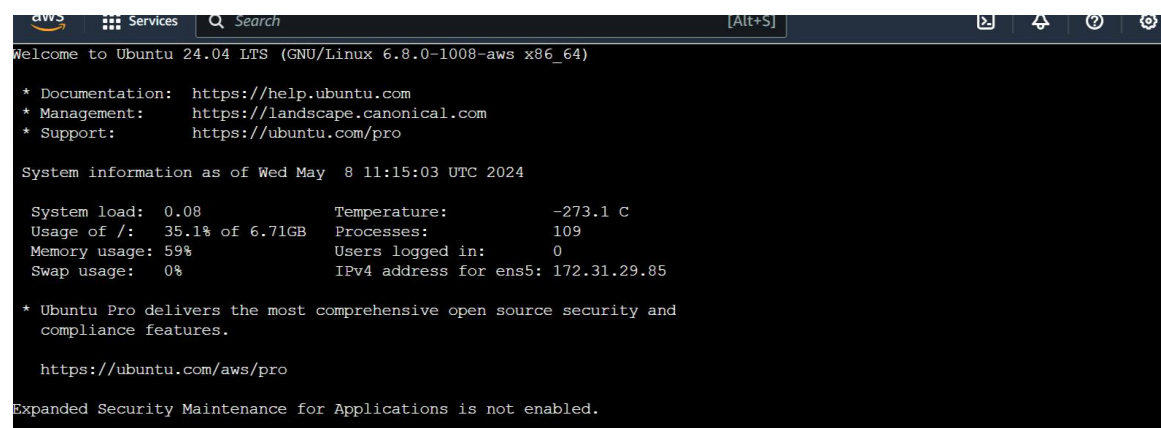
Logged into MySQL shell, created a database named 'wordpress', created a MySQL user with privileges on the 'wordpress' database, and verified the successful creation of the database and user.

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)
```



```
aws Services Search [Alt+S]
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1008-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

System information as of Wed May  8 11:15:03 UTC 2024

System load:  0.08          Temperature:   -273.1 C
Usage of /:   35.1% of 6.71GB Processes:       109
Memory usage: 59%          Users logged in: 0
Swap usage:   0%           IPv4 address for ens5: 172.31.29.85

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.
```

Secure mysql installation

```
sudo mysql_secure_installation
```

```

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-172-31-0-62:~$

```

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

```

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

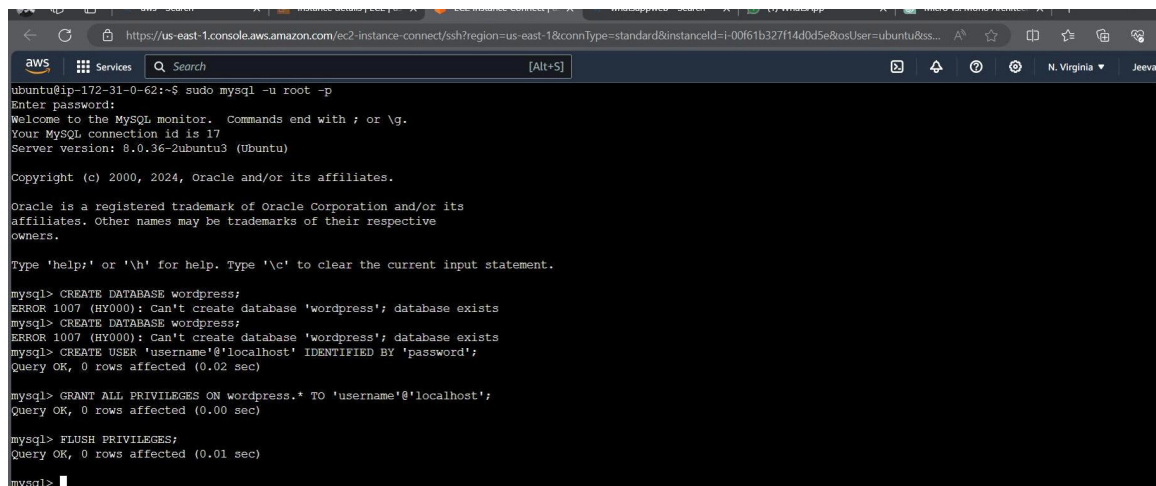
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-172-31-0-62:~$

```

Once logged in create a DB

`CREATE DATABASE wordpress;`

A screenshot of an AWS console terminal window. The terminal shows a user logging into an Ubuntu instance via SSH. The user runs the command 'sudo mysql -u root -p' to start the MySQL monitor. The terminal displays the MySQL welcome message, the connection ID (17), and the server version (8.0.36-2ubuntu3). The user then enters several SQL commands: 'CREATE DATABASE wordpress;', 'CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';', 'GRANT ALL PRIVILEGES ON wordpress.* TO 'username'@'localhost';', and 'FLUSH PRIVILEGES;'. The output shows that the database 'wordpress' already exists, the user was created successfully, and the privileges were granted and flushed. The terminal ends with the prompt 'mysql>'.

```
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 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>
```

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');
```

```
* * ABSOLUTE path to the database directory
*
* @link https://wordpress.org/documentation/article/editing-wp-config-php/
*
* @package WordPress
*/

/* Database settings - You can get this info from your web host */
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'username' );

/** Database password */
define( 'DB_PASSWORD', 'password' );

/** Database hostname */
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
```

```
GNU nano /./2                                     wp-config.php *
*
* Change these to different unique phrases! You can generate these using
* the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
*
* You can change these at any point in time to invalidate all existing cookies.
* This will force all users to have to log in again.
*
* @since 2.6.0
*/
define( 'AUTH_KEY',            '$_n+e3[tr@m#17ro5-]?z@v 1h(z;)w3= s#j|/8+1 f*2iwepnev(gog[3lc !7')';
define( 'SECURE_AUTH_KEY',    'dtq =j68u3d+~`wbt&^?x~/*ovs!e98cwq~r)ewj} `9_*($td>?vz#t+. ;$)no!')';
define( 'LOGGED_IN_KEY',      '2me) <.s%0r5d[|9g;n #d^lg|2a8y3q6%q`z0+pswm)@u,2.z#:{s76!p=l!;ww!');
define( 'NONCE_KEY',          '<_)~*36i0fs{||m,zq_w|,x+2(^)2r>d.60lnx.u;)w*;!~(:@zq90;jlw6bn&:')';
define( 'AUTH_SALT',          '1%#<;dt*/gzo!( _b!]53q#|@o)=6)66[1-6$9]];0684r.:{<<9.x6hcv6,3?a^s')';
define( 'SECURE_AUTH_SALT',   '3*+98zd$d:-kq?nhicyh_7(((*j%sl2%)`ts4$`u!x?a^y?<)9( ar)j*.$.s ic')';
define( 'LOGGED_IN_SALT',     '6r*8=!l!rk[];d@!jo(bkjas,[z7csr**zj(kuoo/?,[|py2,w7+(<5s[~h_#_!uq')';
define( 'NONCE_SALT',         '~!~#38e6/) [(x,6,n)]+o6?c*;b =9|e;[ `vxzpf.86^ey^i6fq(?{s0>4p#o!e')';
/*#@-*/

/**
 * WordPress database table prefix.
 *
 * You can have multiple installations in one database if you give each
```

```
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
```

```
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:~$ 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:~$
```

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

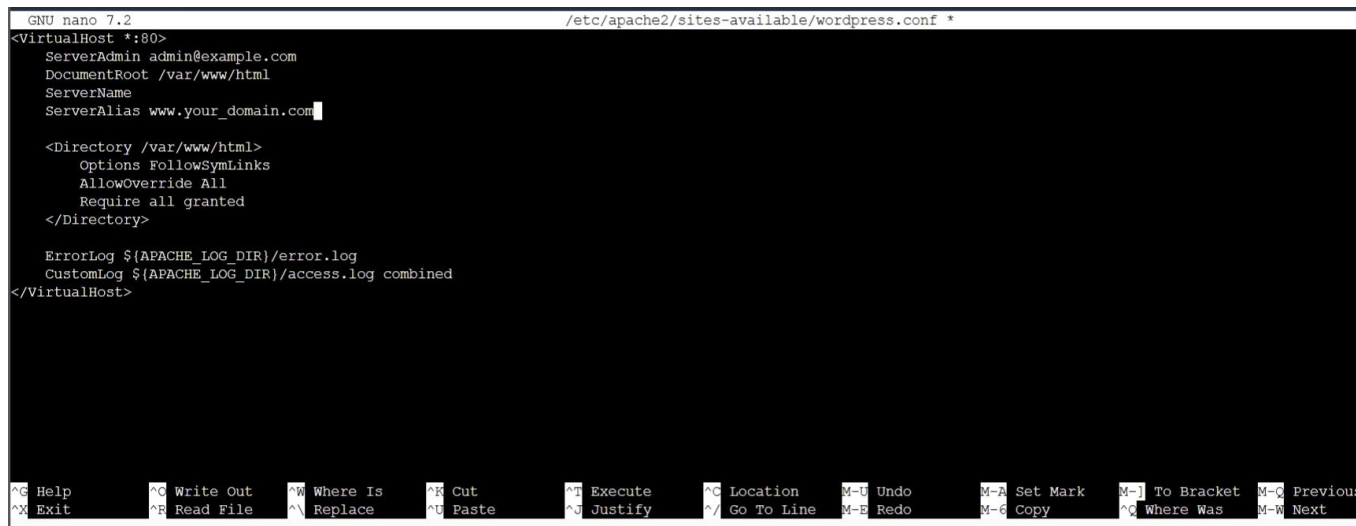
Require all granted

</Directory>

ErrorLog \${APACHE_LOG_DIR}/error.log

CustomLog \${APACHE_LOG_DIR}/access.log combined

</VirtualHost>



```
GNU nano 7.2 /etc/apache2/sites-available/wordpress.conf *
<VirtualHost *:80>
  ServerAdmin admin@example.com
  DocumentRoot /var/www/html
  ServerName
  ServerAlias www.your_domain.com

  <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>
```

^G Help ^O Write Out ^M Where Is ^R Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-^ Previous
^X Exit ^F Read File ^N Replace ^U Paste ^_ Justify ^/ Go To Line M-B Redo M-C Copy ^Q Where Was M-W Next

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$

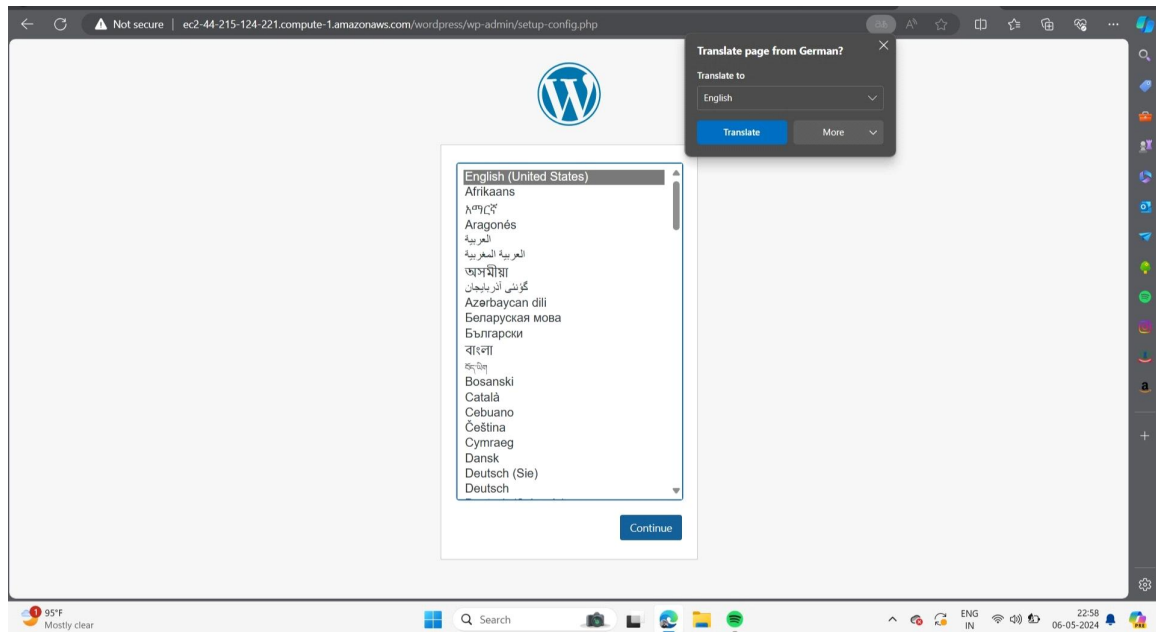
```

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

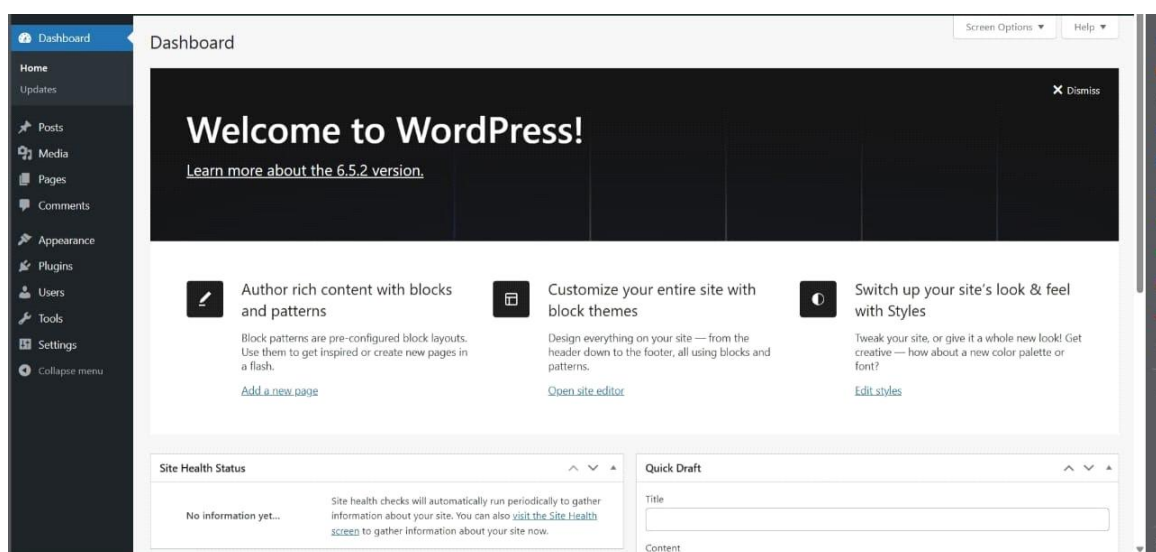
In my case ip address is <http://3.230.128.33/> \wordpress

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