

# 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.

Click on the "Launch Instance" button to begin the instance creation process. You'll be guided 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.

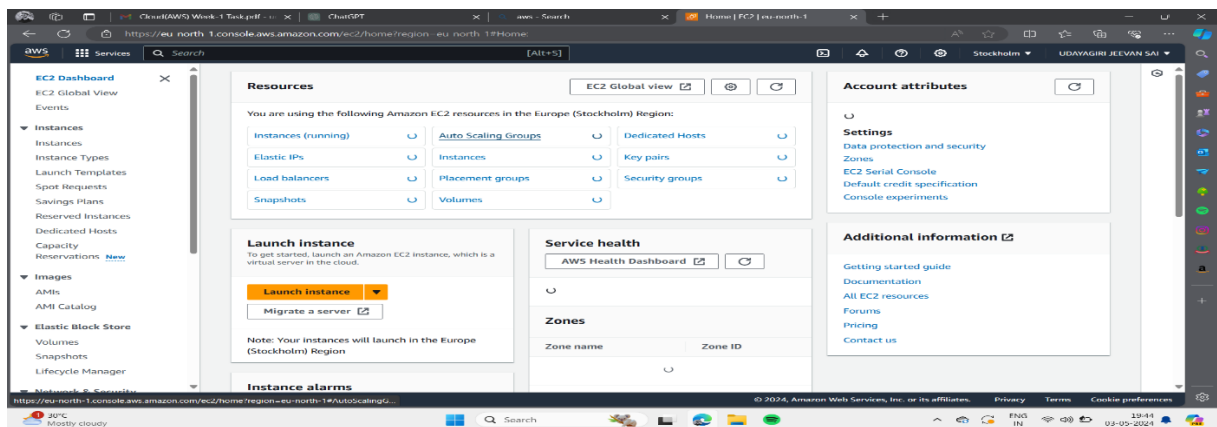
Review your instance configuration to ensure everything is set up as desired. Once confirmed, click "Launch" to initiate the instance creation process.

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).

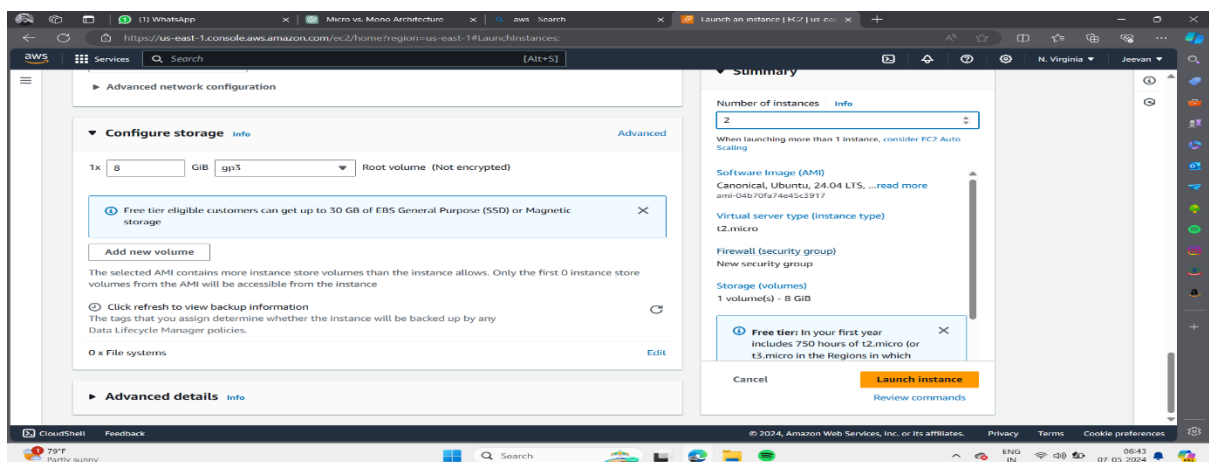
After selecting the key pair, click "Launch Instances" to finalize the process. Your instance will now be provisioned and launched.

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



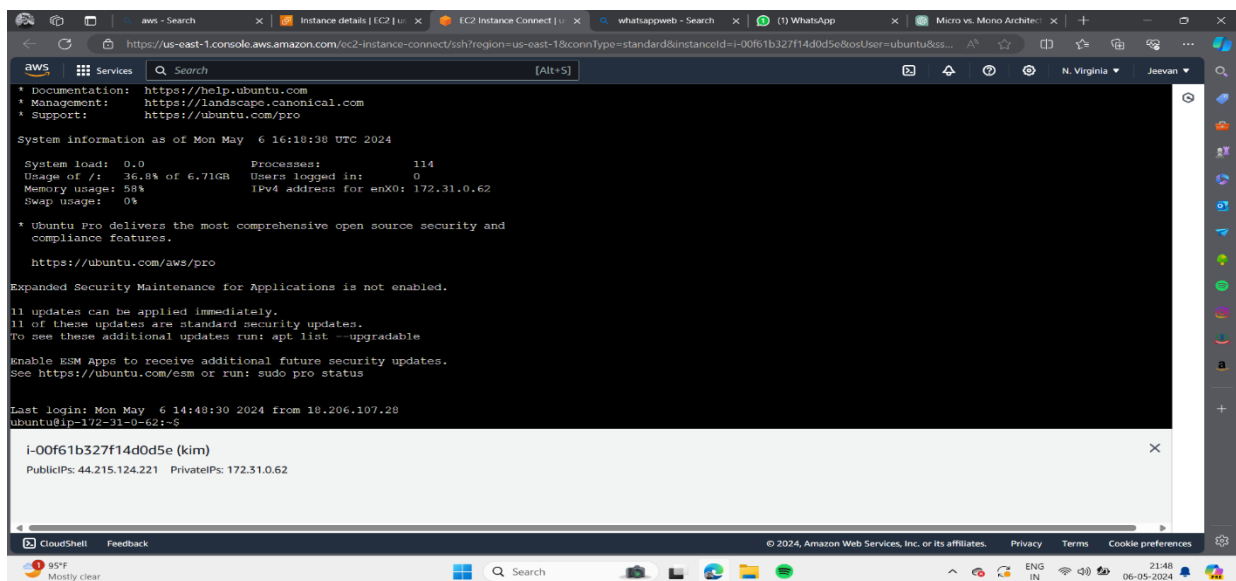
Select all requirements



Launch you instance as shown in above screenshot

Next

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.

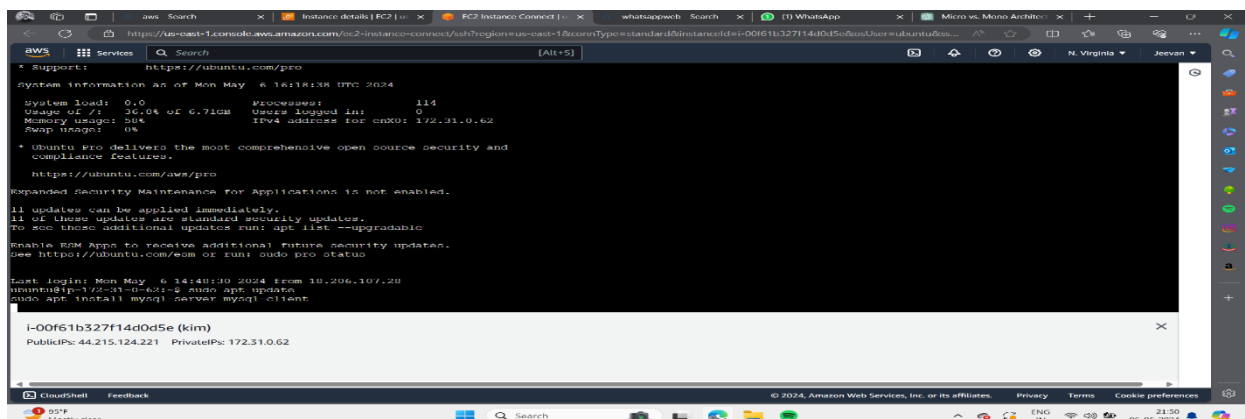
Logged into MySQL shell, created a database named 'wordpress', created a MySQL user with privileges on the 'wordpress' database, and verified the successful creation of the database and user.

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 Search
https://aws.amazon.com/ec2-instance-connect/sh?region=us-east-1&contentType=standard&instanceId=i-0061b327f14d0d5e5&userSubantua5e...
[Alt+S]
N. Virginia Jeevan

system information as of Mon May 6 16:18:38 UTC 2024
System load: 0.0 Processes: 114
Usage of /: 36.0% of 6.71GB
Memory usage: 50%
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
compliance features.
https://ubuntu.com/aws/pro

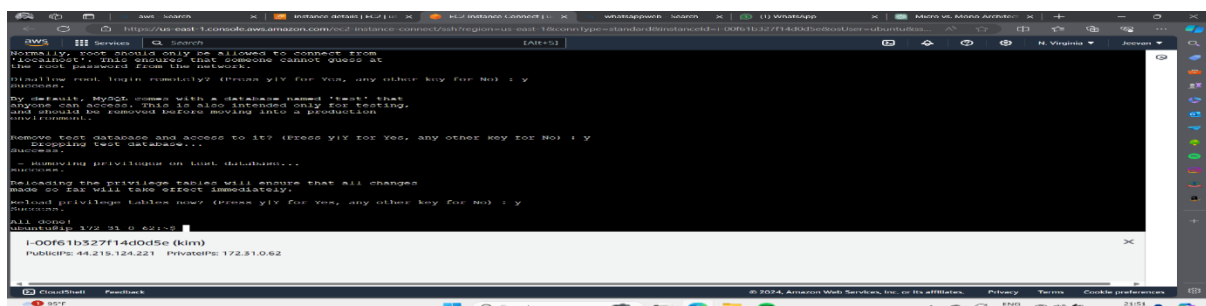
Expanded Security Maintenance for Applications is not enabled.
11 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Ubuntu Pro Apps to receive additional future security updates.
See https://ubuntu.com/pro or run: sudo pro status

Last login: Mon May 6 14:10:10 2024 from 18.206.107.20
ubuntu@ip-172-31-11-61:~$ sudo apt update
sudo apt install mysql-server mysql-client

i-0061b327f14d0d5e (kim)
PublicIPs: 44.215.124.221 PrivateIPs: 172.31.0.62
```

## Secure mysql installation

`sudo mysql_secure_installation`



```
aws Search
https://aws.amazon.com/ec2-instance-connect/sh?region=us-east-1&contentType=standard&instanceId=i-0061b327f14d0d5e5&userSubantua5e...
[Alt+S]
N. Virginia Jeevan

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

Disable 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-172-31-11-61:~$
```

## Follow these steps for secure installation

Set root password

Remove anonymous users

Disallow root login remotely

Remove test database and access to it

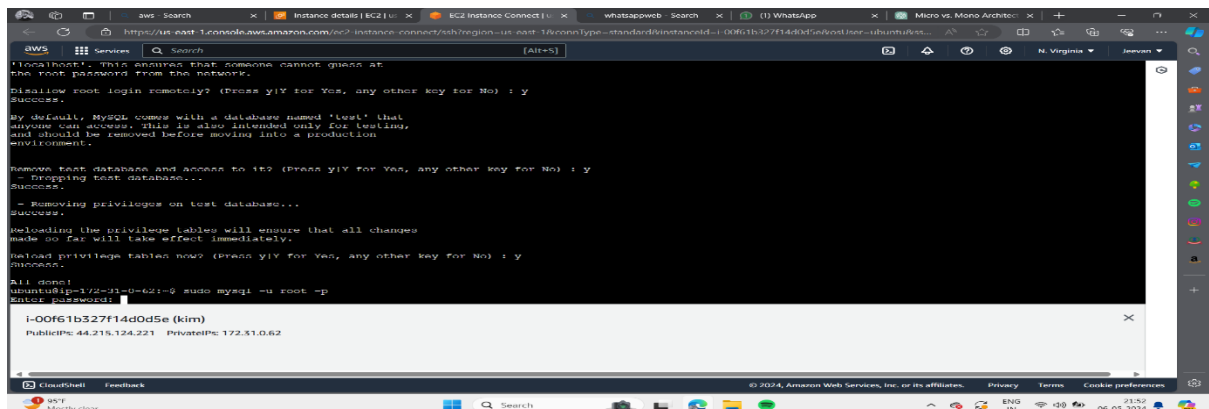
Reload privilege tables now

## Create a MySQL Database and User for WordPress

Log into the MySQL shell as the root user

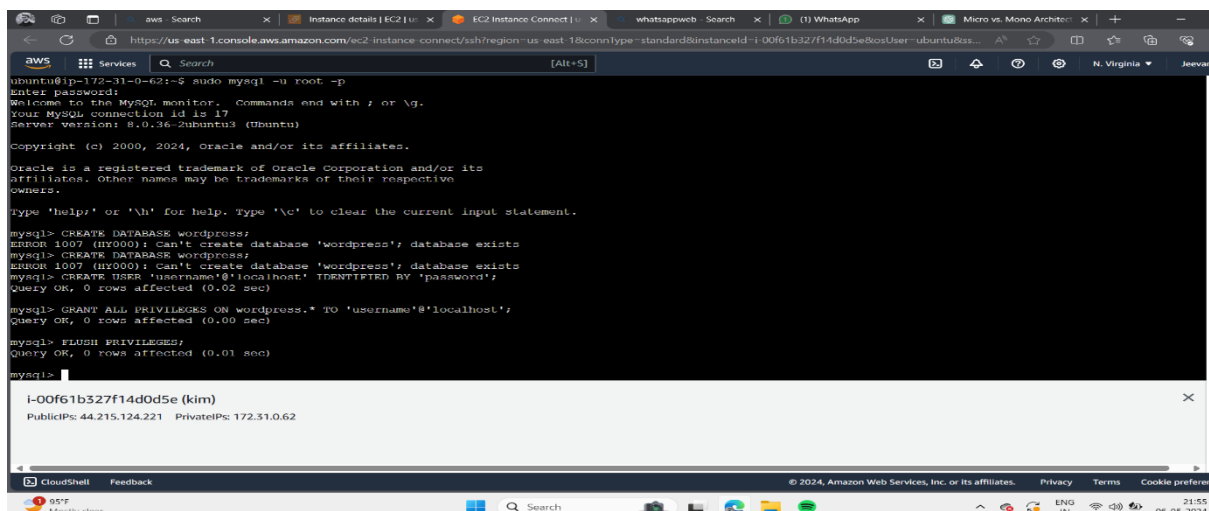
`sudo mysql -u root -p`

Enter the root password when prompted



Once logged in create a DB

`CREATE DATABASE wordpress;`



Next follow the steps

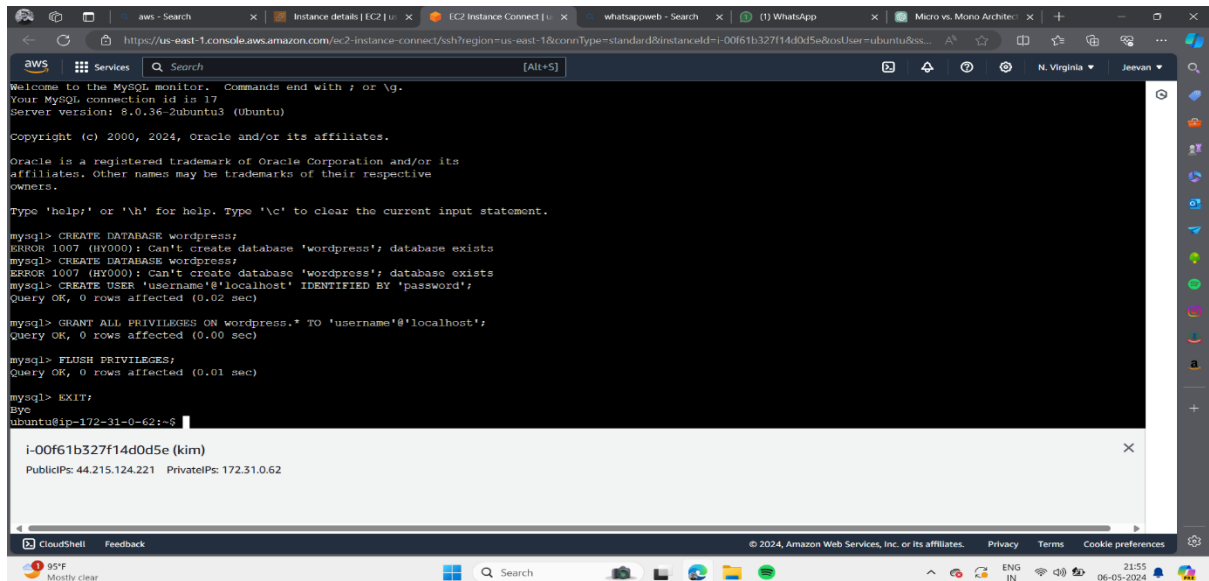
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';

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

FLUSH PRIVILEGES;

After this exit command should be used

EXIT;



```
aws Search
Instance details | EC2 | u
EC2 Instance Connect | u
whatsappweb - Search
(1) WhatsApp
Micro vs. Mono Architec
+
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-00f61b327f14d0d5e&osUser=ubuntu&sess...
[Alt+F5]
N. Virginia
Jeevan
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)

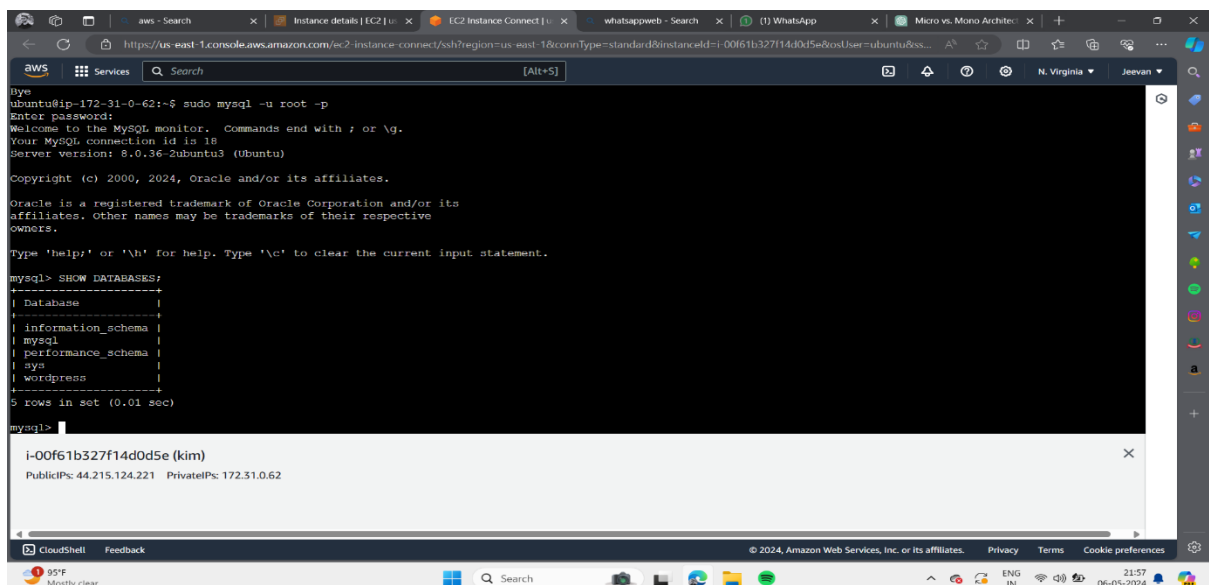
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> EXIT;
Bye
ubuntu@ip-172-31-0-62:~$

i-00f61b327f14d0d5e (kim)
PublicIPs: 44.215.124.221 PrivateIPs: 172.31.0.62

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
95°F Mostly clear
Search
ENG IN
21:55 06-05-2024
```

Show databases;



```
aws Search
Instance details | EC2 | u
EC2 Instance Connect | u
whatsappweb - Search
(1) WhatsApp
Micro vs. Mono Architec
+
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-00f61b327f14d0d5e&osUser=ubuntu&sess...
[Alt+F5]
N. Virginia
Jeevan
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>

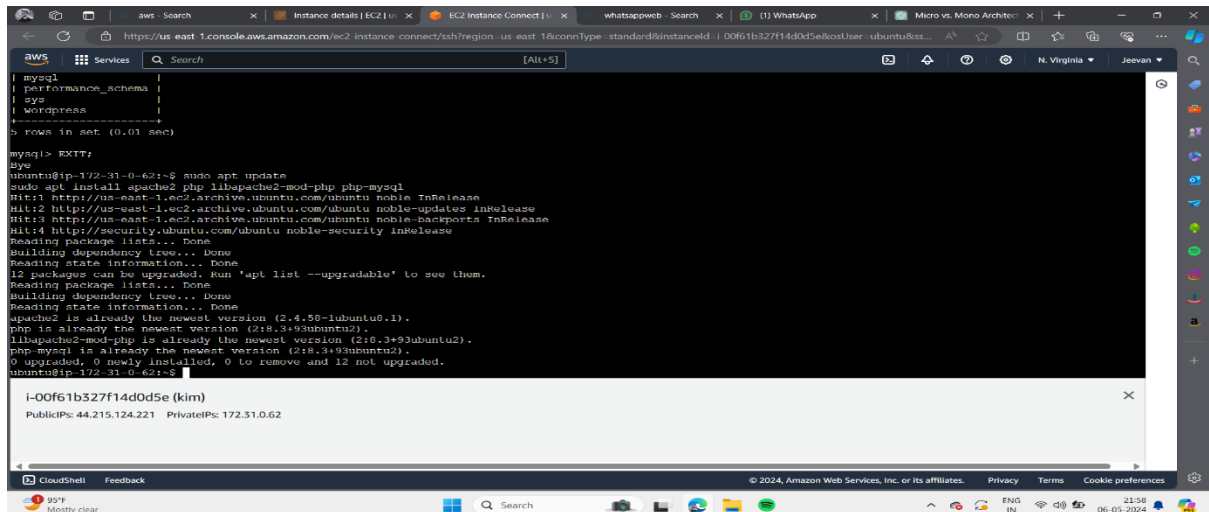
i-00f61b327f14d0d5e (kim)
PublicIPs: 44.215.124.221 PrivateIPs: 172.31.0.62

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
95°F Mostly clear
Search
ENG IN
21:57 06-05-2024
```

# Install Apache Web Server, PHP, and Required Extensions

```
sudo apt update
```

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



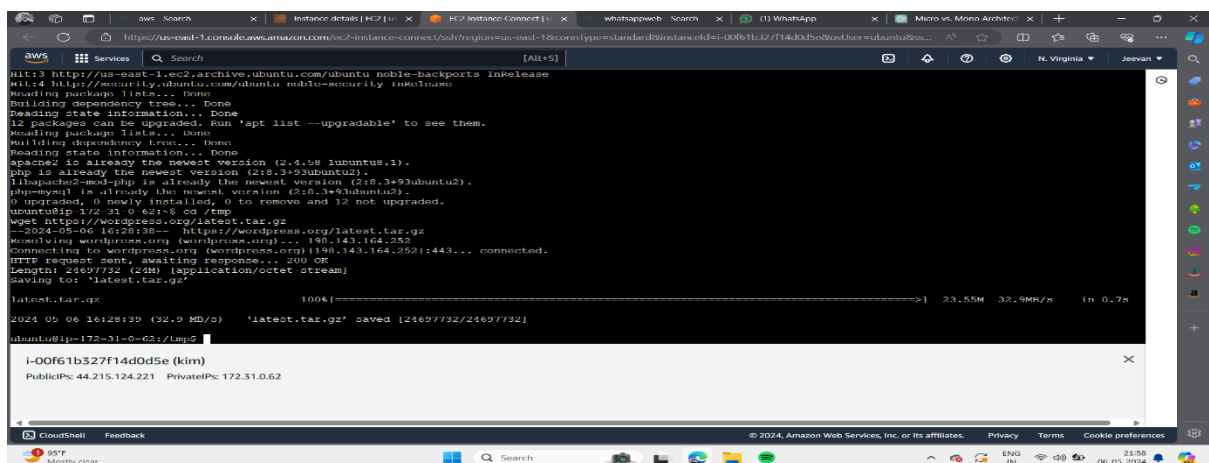
```
mysql |
+-----+
| performance_schema |
| sys                 |
| wordpress            |
+-----+
4 rows in set (0.01 sec)

mysql> EXIT;
bye
ubuntu@ip-172-31-0-62:~$ sudo apt update
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:~$
```

## Download and Configure WordPress

```
cd /tmp
```

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



```
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:~$ cd /tmp
ubuntu@ip-172-31-0-62:~/tmp$ wget https://wordpress.org/latest.tar.gz
--2024-05-06 16:26:38-- https://wordpress.org/latest.tar.gz
Connecting to wordpress.org (wordpress.org)... 198.143.164.252
HTTP request sent, awaiting response... 200 OK
Length: 2469732 (2.4M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz      100%[=====>] 2.355M  32.9MB/s  in 0.7s

2024 05 06 16:26:39 (32.9 MB/s) 'latest.tar.gz' saved [2469732/2469732]

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
```

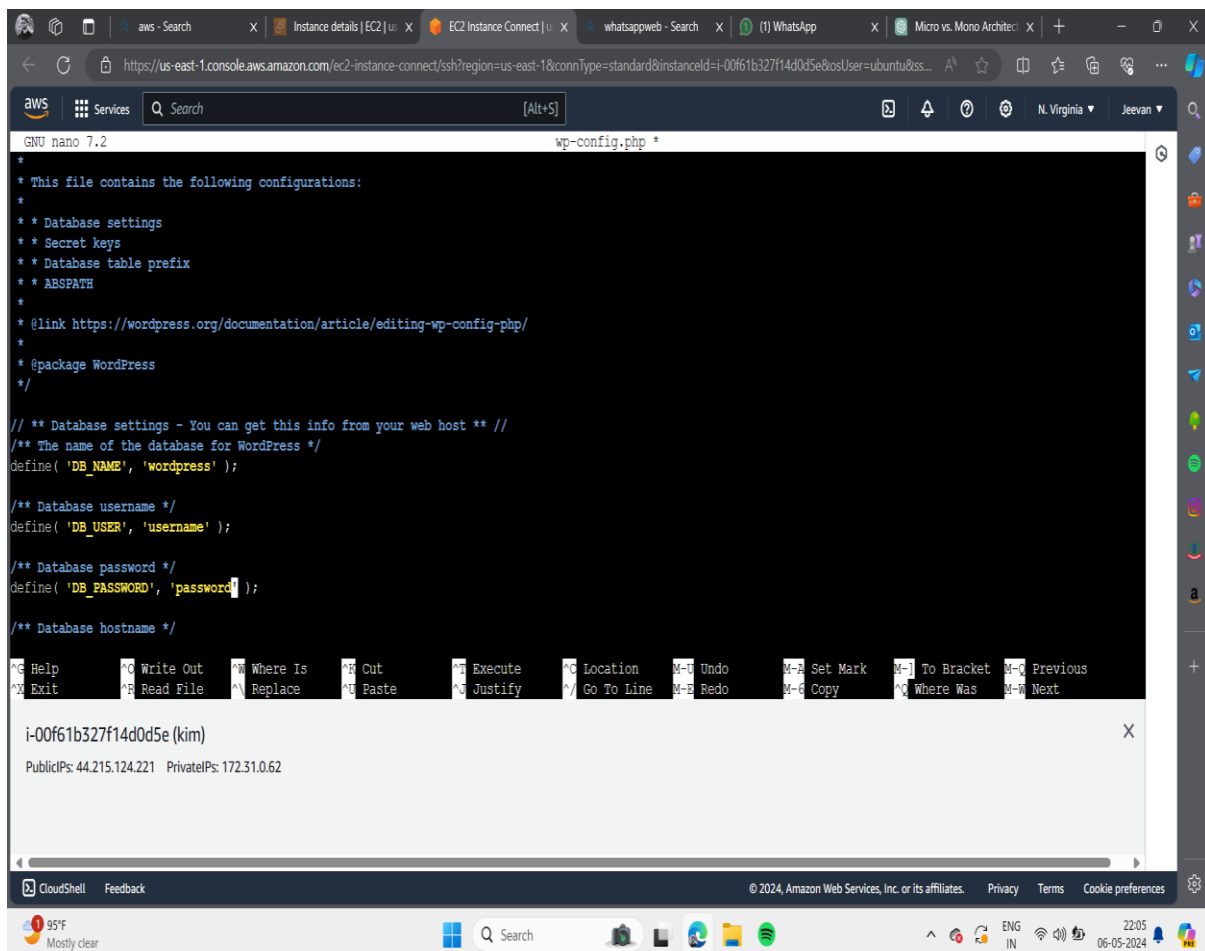
## Update the following lines with your MySQL database information:

```
define('DB_NAME', 'wordpress');
```

```
define('DB_USER', 'username');
```

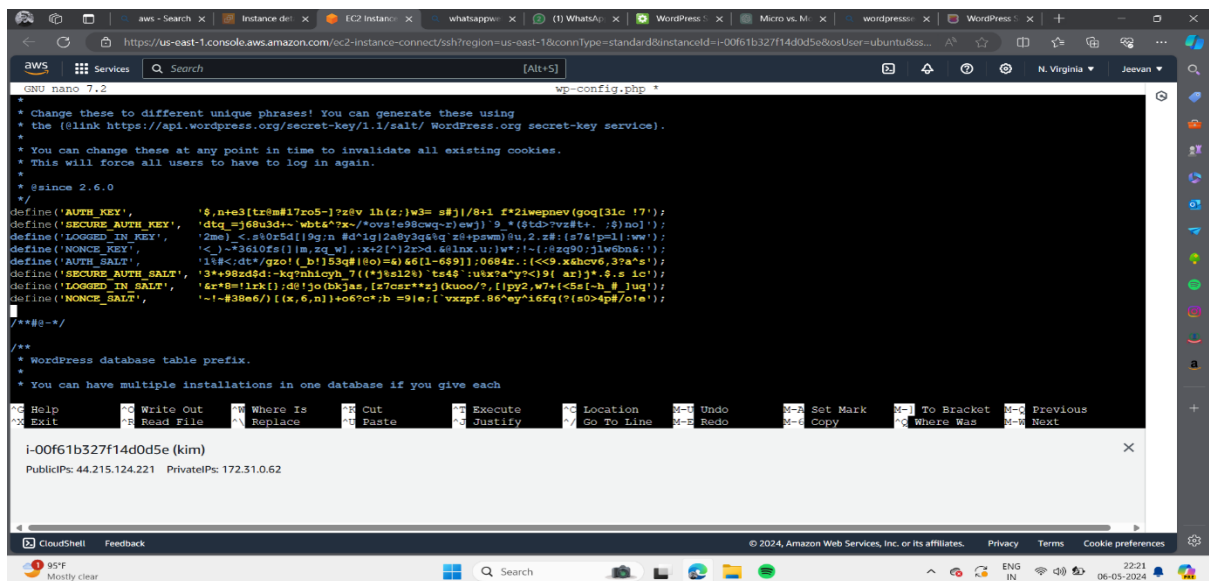
```
define('DB_PASSWORD', 'password');
```

```
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' );
define( 'NONCE_SALT',       'put your unique phrase here' );
```



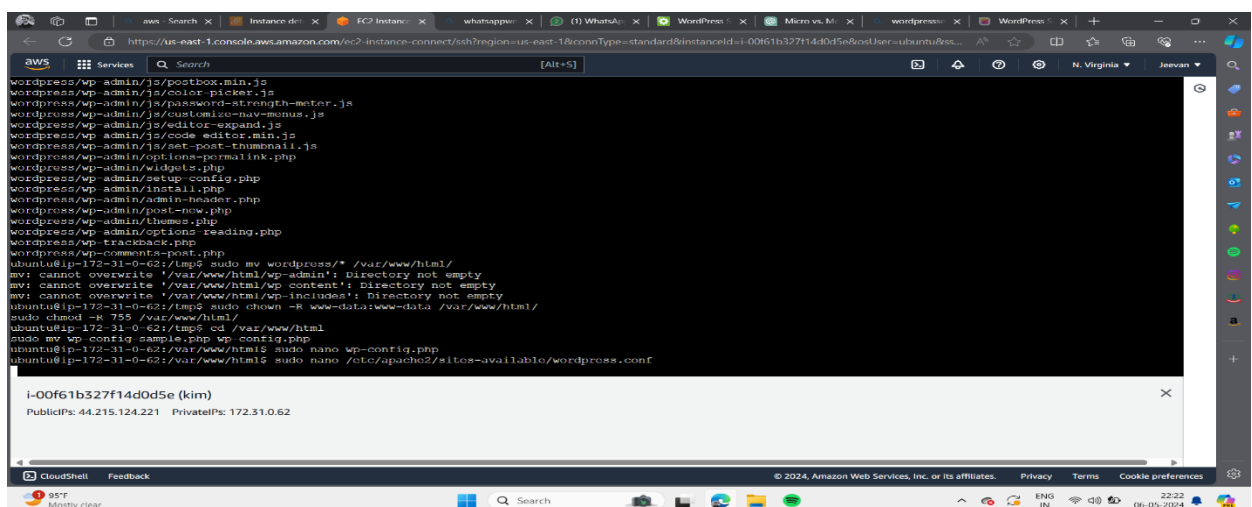
```
GNU nano 7.2 wp-config.php
* Change these to different unique phrases! You can generate these using
* the (link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service).
* You can change these at any point in time to invalidate all existing cookies.
* This will force all users to have to log in again.
*
* Since 2.6.0
*/
define('AUTH_KEY',         '6,n+e3{tr@#17ro5-}7z@v 1h(z;)}w3= s#j|/8+1 f*2iwepnev(gog{3lc !7'});
define('SECURE_AUTH_KEY',  'dtq_aj68u3d+~ wbt$^*x/*ovs!e98cwq-z)ewj} 9_~(8td~vz#t+. {8)noj!');
define('LOGGED_IN_KEY',    '2mcl_~_s!0s5dl!9grn #d*!q!8a9y2q8kq z84pwn)8u,2_#:(d7!tpsl!{wv');
define('NONCE_KEY',        '<_~3610fs(!m,zq w).:x+2[^!2r>d.60lnx.u;]w*!~(!:8zq90;]lw6bns!');
define('AUTH_SALT',        '1%#<;dt*/gzo!(_b!}53q#!(o)=6)66{1-699}};0684r.:!<<9.x6hcv6,3a^s!');
define('SECURE_AUTH_SALT', '3*+98pd8d~-kq7nhicyb_7((**!s12t) ts4$ :uix7a^y7<|9( az!)*.6.s 10');
define('LOGGED_IN_SALT',   '4et8~!1rk!;de!9o(bh3e8,if7car**z!kuoo/p, {!p2,v7*(<5a{ch_#_!ug');
define('NONCE_SALT',       '-!-#38e6/){z,6,n)!~o87c*;b =9!e;[ vxzpf.86^y~i6fq{?{s0>4p/o!e!');

/*
** WordPress database table prefix.
**
** You can have multiple installations in one database if you give each
** a unique prefix
**
** The only exception to the rule is the wp_options table, which must always be prefixed with wp_
*/
define('DB_PREFIX', 'wp_');
```

# 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/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/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:~$ sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
ubuntu@ip-172-31-0-62:~$ cd /var/www/html
sudo mv wp-config-sample.php wp-config.php
ubuntu@ip-172-31-0-62:~$ sudo nano /etc/apache2/sites-available/wordpress.conf
ubuntu@ip-172-31-0-62:~$ 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

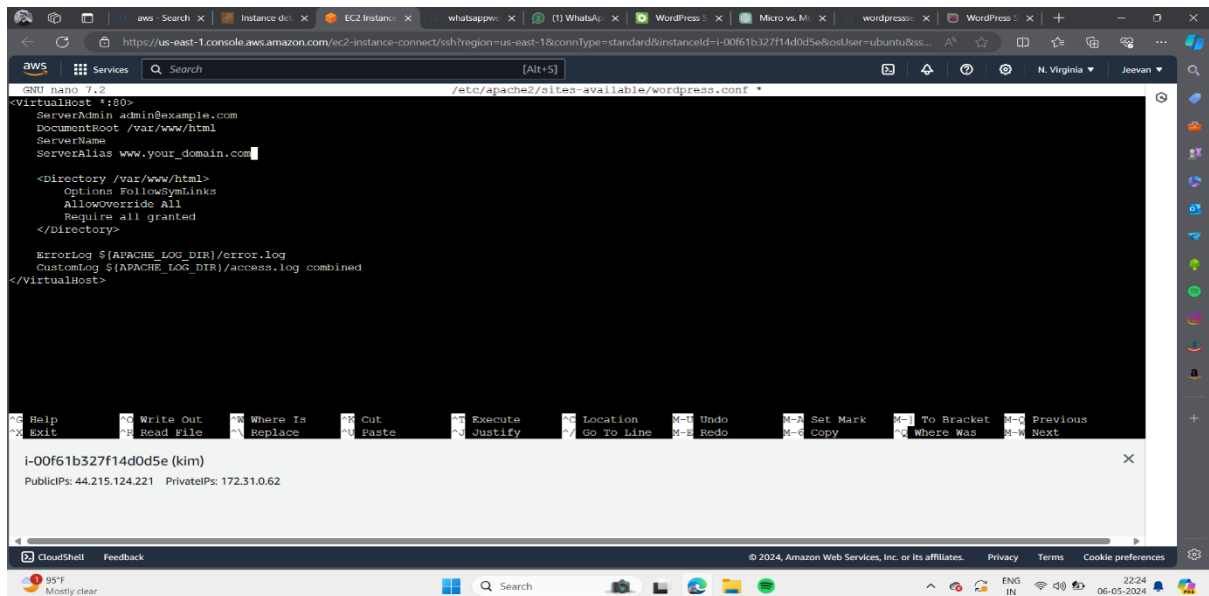
    Require all granted

</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/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$ sudo chown -R www-data:www-data /var/www/html/
chown: R 755 /var/www/html/
ubuntu@ip-172-31-0-62:/tmp$ cd /var/www/html
sudo mv wp-config-sample.php wp-config.php
ubuntu@ip-172-31-0-62:/var/www/html$ sudo nano wp-config.php
ubuntu@ip-172-31-0-62:/var/www/html$ sudo nano /etc/apache2/sites-available/wordpress.conf
ubuntu@ip-172-31-0-62:/var/www/html$
```

**Restart Apache for the changes to take effect:**

`sudo systemctl restart apache2`

```
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$ sudo chown -R www-data:www-data /var/www/html/
chown: R 755 /var/www/html/
ubuntu@ip-172-31-0-62:/tmp$ cd /var/www/html
sudo mv wp-config-sample.php wp-config.php
ubuntu@ip-172-31-0-62:/var/www/html$ sudo nano wp-config.php
ubuntu@ip-172-31-0-62:/var/www/html$ sudo nano /etc/apache2/sites-available/wordpress.conf
ubuntu@ip-172-31-0-62:/var/www/html$ sudo a2dissite 000-default.conf
Enabling site wordpress.
To activate the new configuration, you need to run:
  systemctl reload apache2
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-0-62:/var/www/html$ sudo systemctl restart apache2
ubuntu@ip-172-31-0-62:/var/www/html$
```

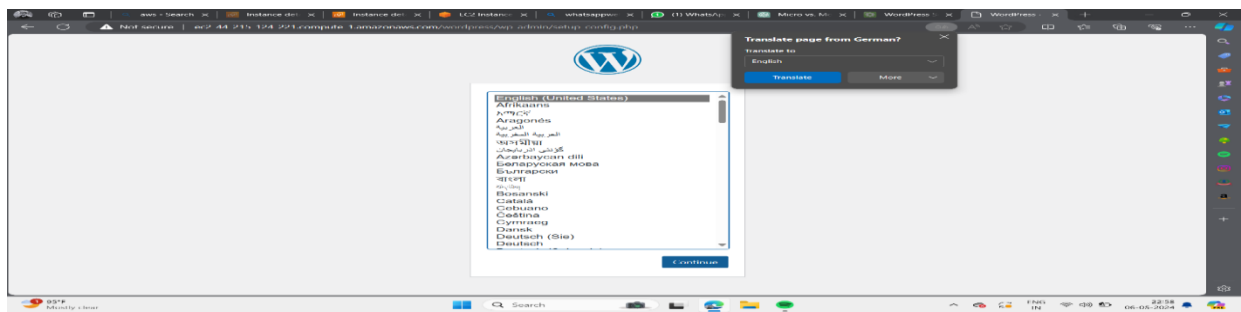
**Finally paste your Dns or instead paste IP address of your created instance**

**In my case ip address is : 44.215.124.221\wordpress**

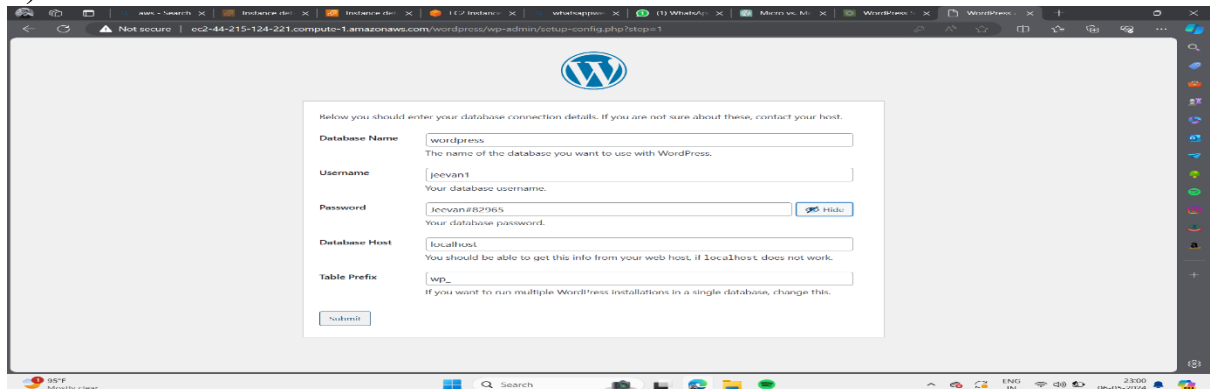
**In my case Dns is: ec2-44-215-124-221.compute-1.amazonaws.com\wordpress\wp-admin\**

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

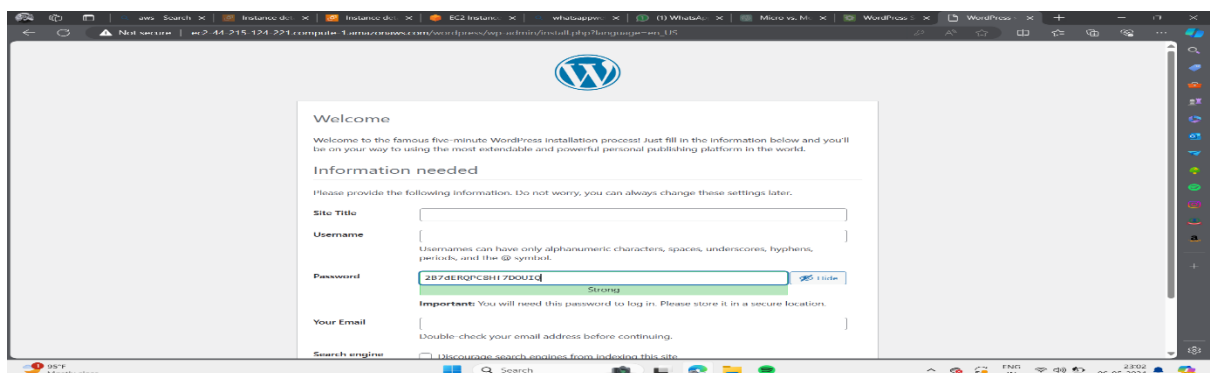
1)



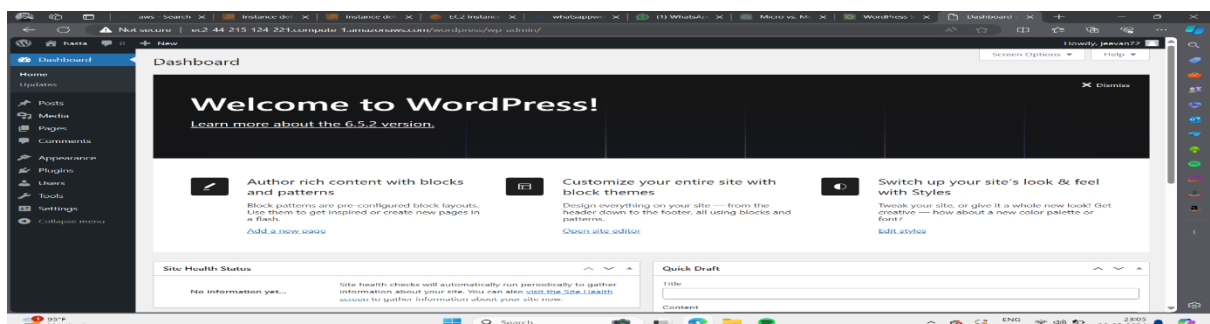
2)



3)



4)finally the page saying welcome to wordpress will be displayed



Here is the deployed page as welcome page in wordpress

# Microservices

Firstly we should create two instances that is 1 for wordpress and another for mysql

## 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.

Click on the "Launch Instance" button to begin the instance creation process. You'll be guided 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.

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

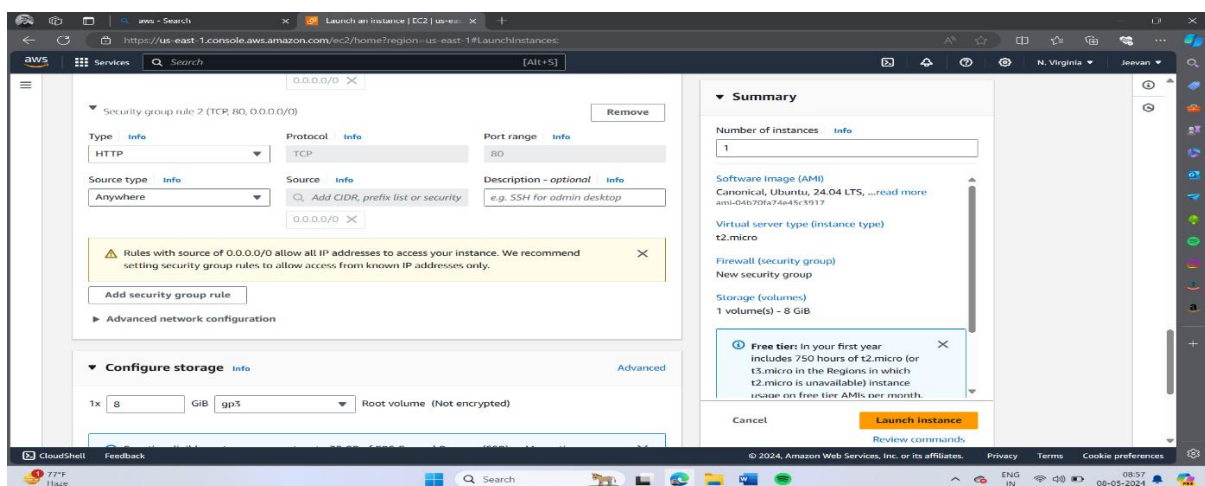
Review your instance configuration to ensure everything is set up as desired. Once confirmed, click "Launch" to initiate the instance creation process.

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

## 1 instance for wordpress



aws - Search

Launch an instance | EC2 | us-east-1

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#launchInstances:

Services Search [Alt+S]

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type ssh

Protocol TCP

Port range 22

Source type Anywhere

Source 0.0.0.0/0

Description - optional e.g. SSH for admin desktop

Remove

▼ Security group rule 2 (TCP, 3306, sg-09a8cff22ba6ec03c)

Type MYSQL/Aurora

Protocol TCP

Port range 3306

Source type Custom

Source sg-09a8cff22ba6ec03c

Description - optional e.g. SSH for admin desktop

Remove

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule

▼ Summary

Number of instances 1

Software Image (AMI) Canonical, Ubuntu, 24.04 LTS, ...read more ami-04b70fa74e45c3917

Virtual server type (instance type) t2.micro

Firewall (security group) New security group

Storage (volumes) 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month.

Cancel Launch instance

Review commands

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

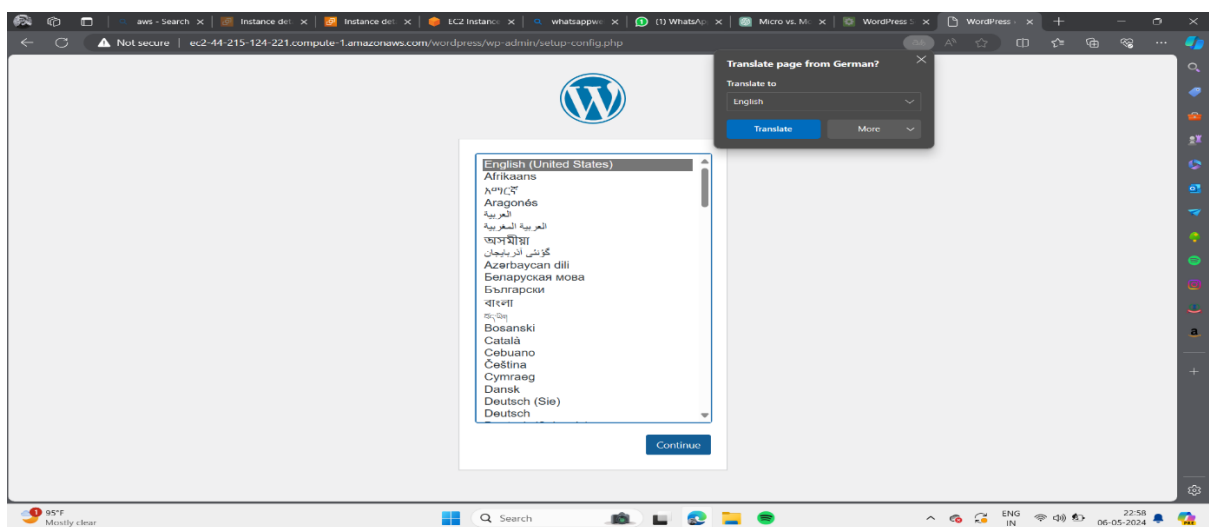
77°F Haze

Search

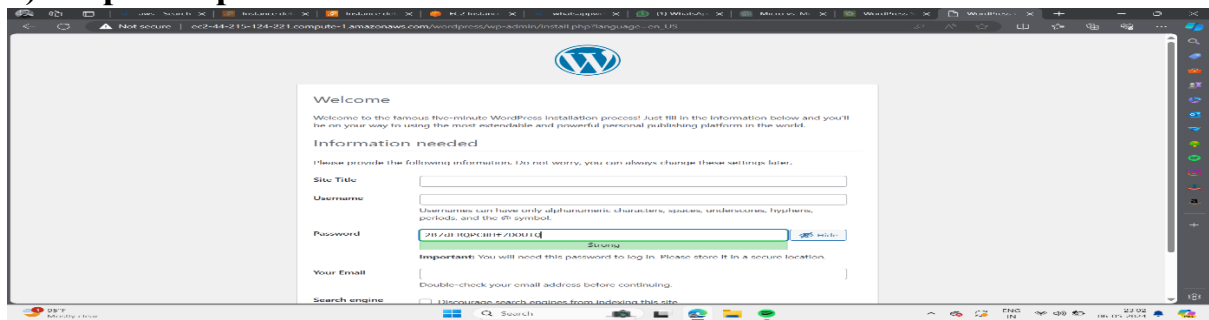
9:00 08-05-2024

## Follow monolithic architecture for further activities after creating instances

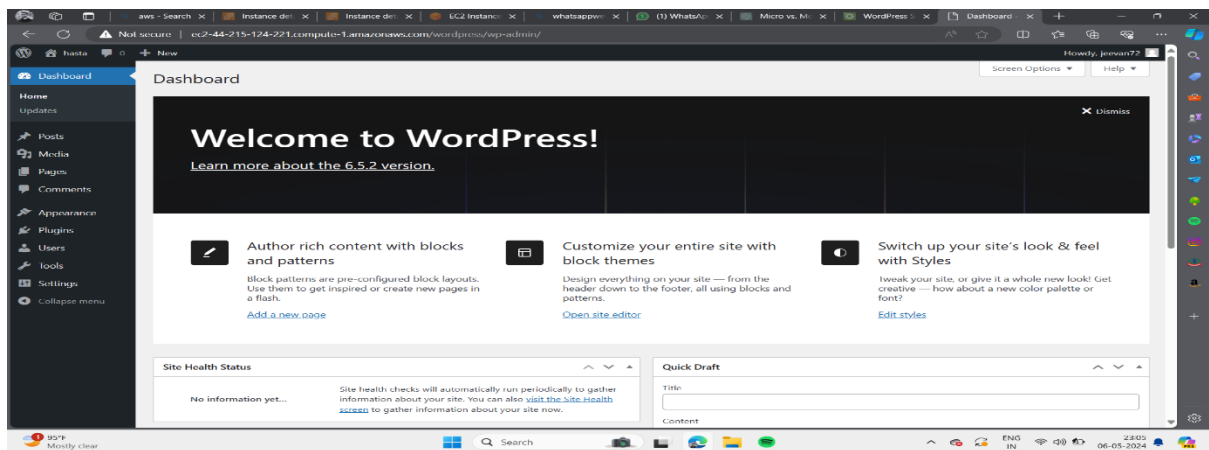
1)



## 2)setup wordpress



## 3)welcome to wordpress



Thank you