Deployment Report: WordPress and MySQL on AWS EC2

1. Introduction

This report outlines the process of deploying WordPress and MySQL on AWS EC2 instances using two different architectures:

1. Monolithic Architecture: A single EC2 instance hosting both WordPress and MySQL.
2. Microservices Architecture: Separate EC2 instances for WordPress and MySQL.

The report also details the security group configurations, installation steps, and relevant official documentation links.

2. AWS EC2 Instance Configuration

EC2 Instance Type: t2.micro

AMI: Ubuntu-*

Security Group: Configured for necessary inbound and outbound rules.

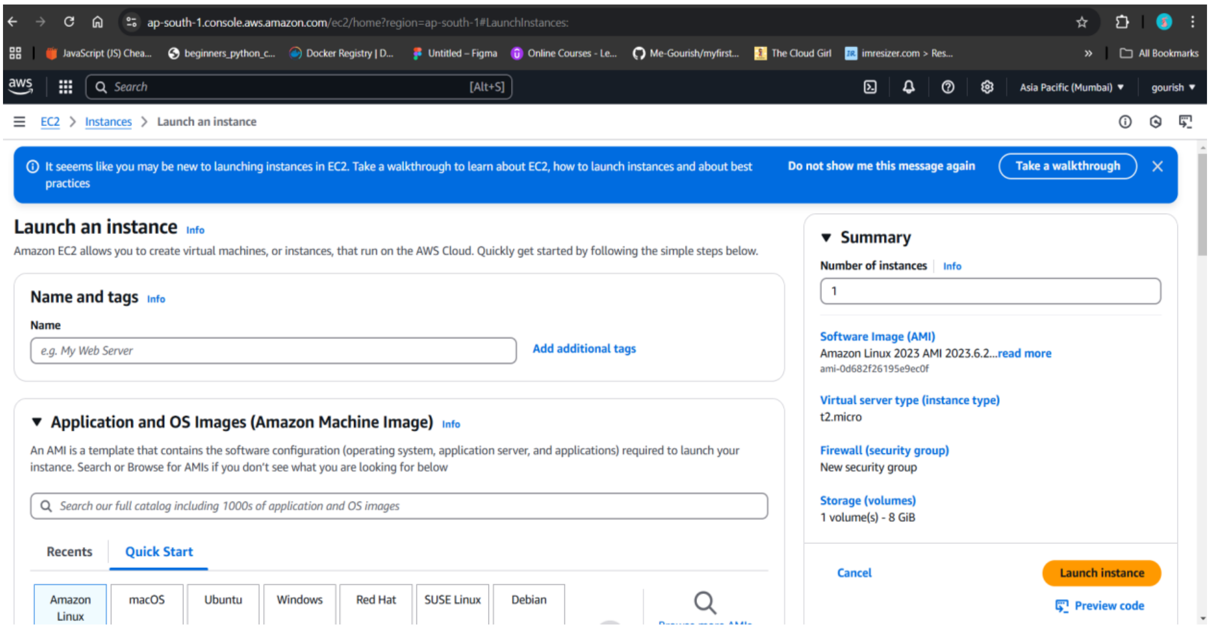
3. Monolithic Architecture Deployment

In this setup, a single EC2 instance will run both WordPress and MySQL.

Step-by-Step Guide

a. Launch an EC2 Instance:

- Go to the AWS EC2 console and launch a t2.micro instance with an Ubuntu AMI.



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

EC2 > Instances > Launch an instance

It seeems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: e.g. My Web Server

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Summary

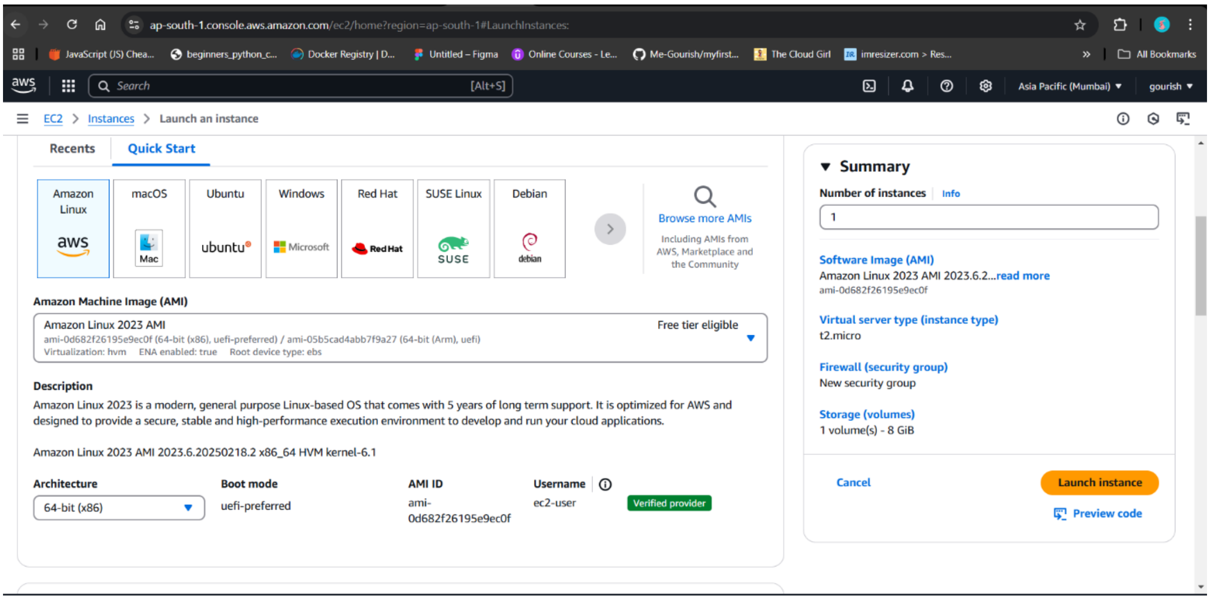
Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more
ami-0d682f26195e9ec0f

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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



Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0d682f26195e9ec0f (64-bit (x86), uefi-preferred) / ami-05b5cad4abb7f9a27 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.6.20250218.2 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Username
64-bit (x86)	uefi-preferred	ami-0d682f26195e9ec0f	ec2-user — Verified provider

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-0d682f26195e9ec0f

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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

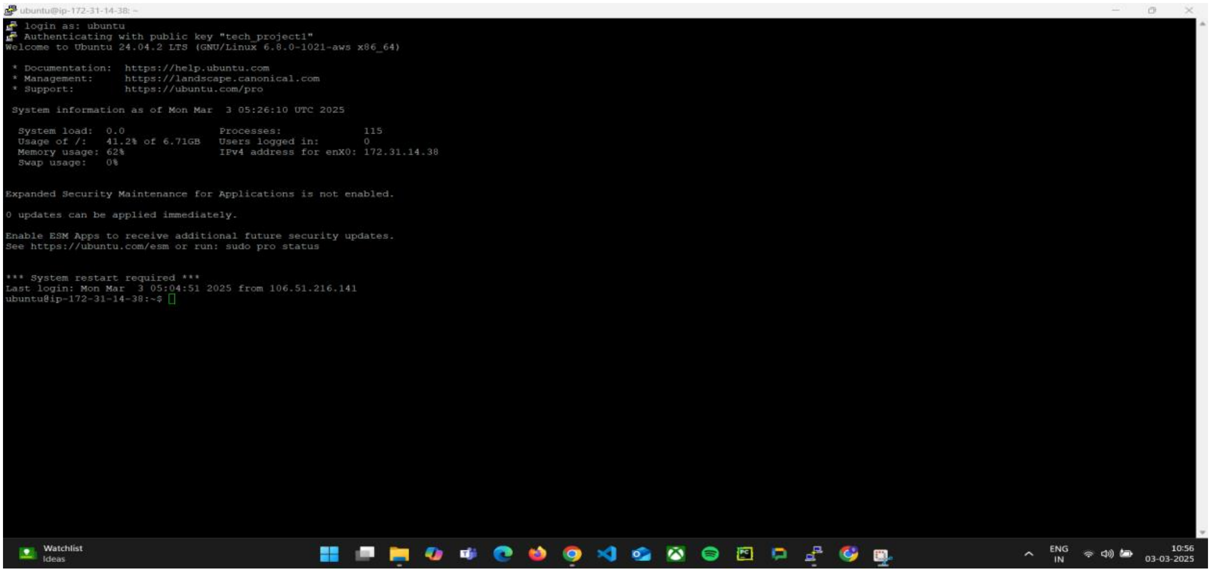
- Configure the security group:
  - Allow HTTP (port 80) and HTTPS (port 443) for web traffic.
  - Allow SSH (port 22) for secure access.
  - Allow MySQL (port 3306) only for local use (optional, for security reasons).
  - Attach a key pair for SSH access.

2.Connect to EC2 instance:

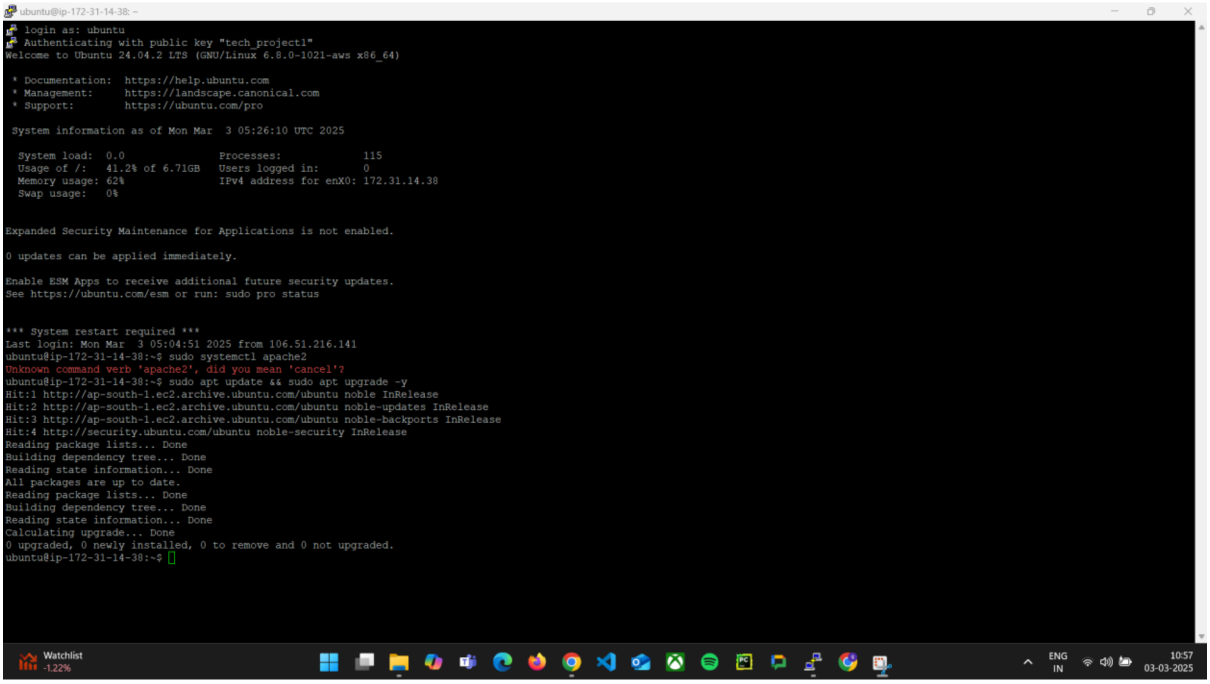
- ssh -i your-key.pem ubuntu@your-ec2-public-ip

3.Install Apache, MySQL, PHP

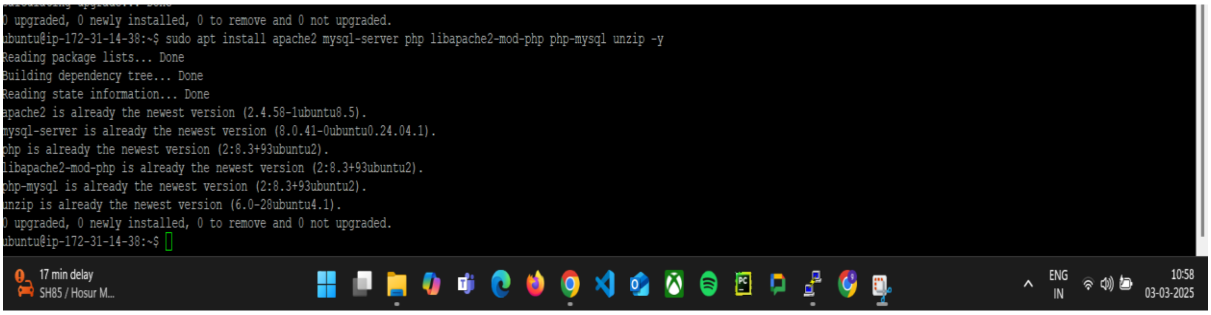
- sudo apt update
- sudo apt install apache2 mysql-server php php-mysql libapache2-mod-php -y



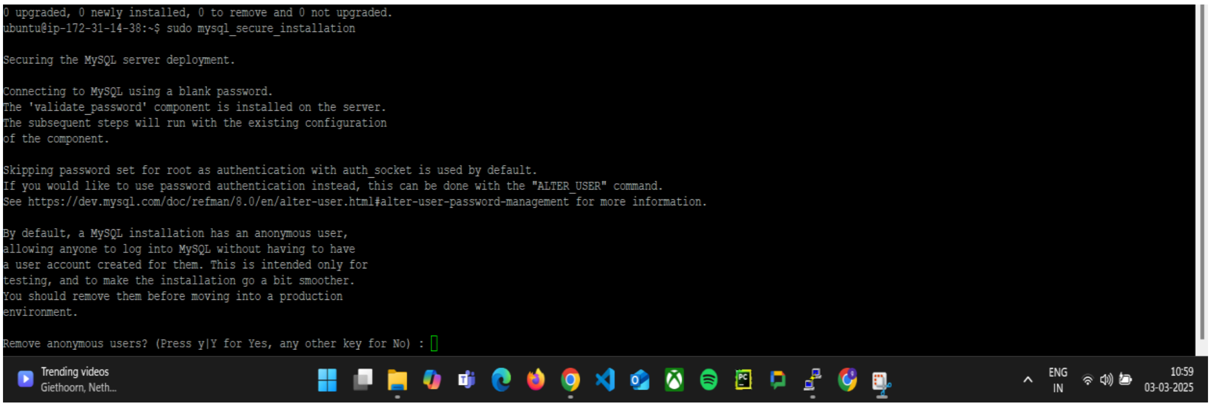
```
login as: ubuntu
Authenticating with public key "tech_project1"
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

 System information as of Mon Mar  3 05:26:10 UTC 2025

  System load:  0.0               Processes:             115
  Usage of /:   41.2% of 6.71GB   Users logged in:       0
  Memory usage: 62%               IPv4 address for enX0: 172.31.14.38
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
Last login: Mon Mar  3 05:04:51 2025 from 106.51.216.141
ubuntu@ip-172-31-14-38:~$
```



```
login as: ubuntu
Authenticating with public key "tech_project1"
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

 System information as of Mon Mar  3 05:26:10 UTC 2025

  System load:  0.0               Processes:             115
  Usage of /:   41.2% of 6.71GB   Users logged in:       0
  Memory usage: 62%               IPv4 address for enX0: 172.31.14.38
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
Last login: Mon Mar  3 05:04:51 2025 from 106.51.216.141
ubuntu@ip-172-31-14-38:~$ sudo systemctl apache2
Unknown command verb 'apache2', did you mean 'cancel'?
ubuntu@ip-172-31-14-38:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-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
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-14-38:~$
```



```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-14-38:~$ sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql unzip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.5).
mysql-server is already the newest version (8.0.41-0ubuntu0.24.04.1).
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).
unzip is already the newest version (6.0-28ubuntu4.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-14-38:~$
```

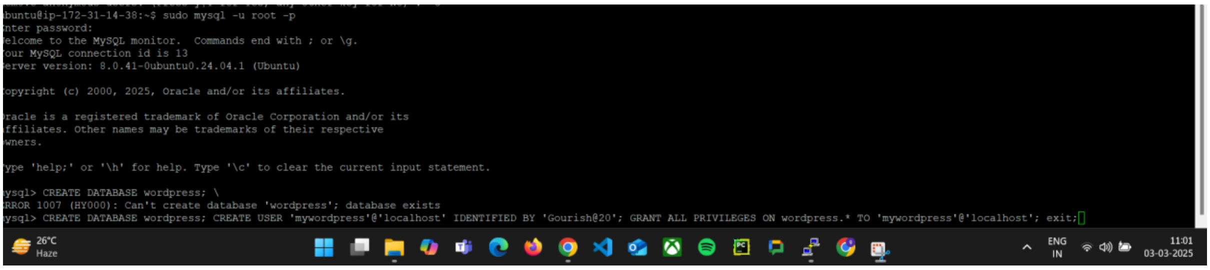


```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-14-38:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.
The 'validate_password' component is installed on the server.
The subsequent steps will run with the existing configuration
of the component.

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) :
```

- MySQL ask for security prompts so we have to say yes for all the questions.

4.Start and Enable Services:

- sudo systemctl enable apache2
- sudo systemctl enable mysql
- sudo systemctl start apache2
- sudo systemctl start mysql

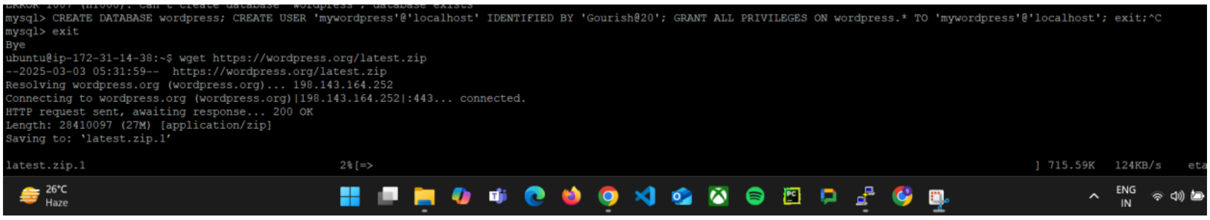


```
ubuntu@ip-172-31-14-38:~$ sudo mysql -u root -p
nter password:
elcome to the MySQL monitor.  Commands end with ; or \g.
our MySQL connection id is 13
erver version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

opyright (c) 2000, 2025, Oracle and/or its affiliates.

racle is a registered trademark of Oracle Corporation and/or its
ffiliates. Other names may be trademarks of their respective
wners.

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

ysql> CREATE DATABASE wordpress; \
RROR 1007 (HY000): Can't create database 'wordpress'; database exists
ysql> CREATE DATABASE wordpress; CREATE USER 'mywordpress'@'localhost' IDENTIFIED BY 'Gourish@20'; GRANT ALL PRIVILEGES ON wordpress.* TO 'mywordpress'@'localhost'; exit;
```

- Above image saya mysql has started and database has been created.

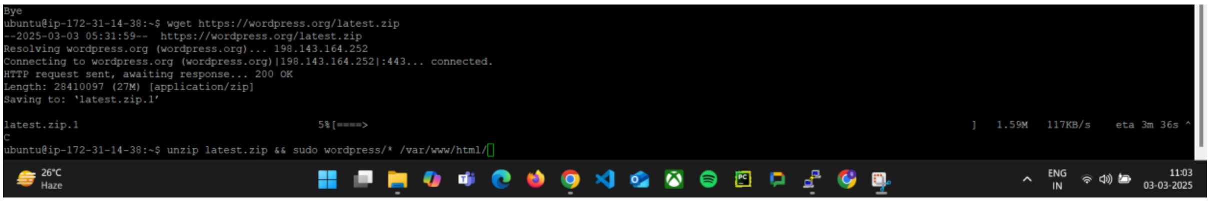
5.Download and Configure WordPress:

- cd /var/www/html
- sudo wget https://wordpress.org/latest.tar.gz
- sudo tar -xzf latest.tar.gz
- sudo chown -R www-data:www-data wordpress
- sudo chmod -R 755 wordpress



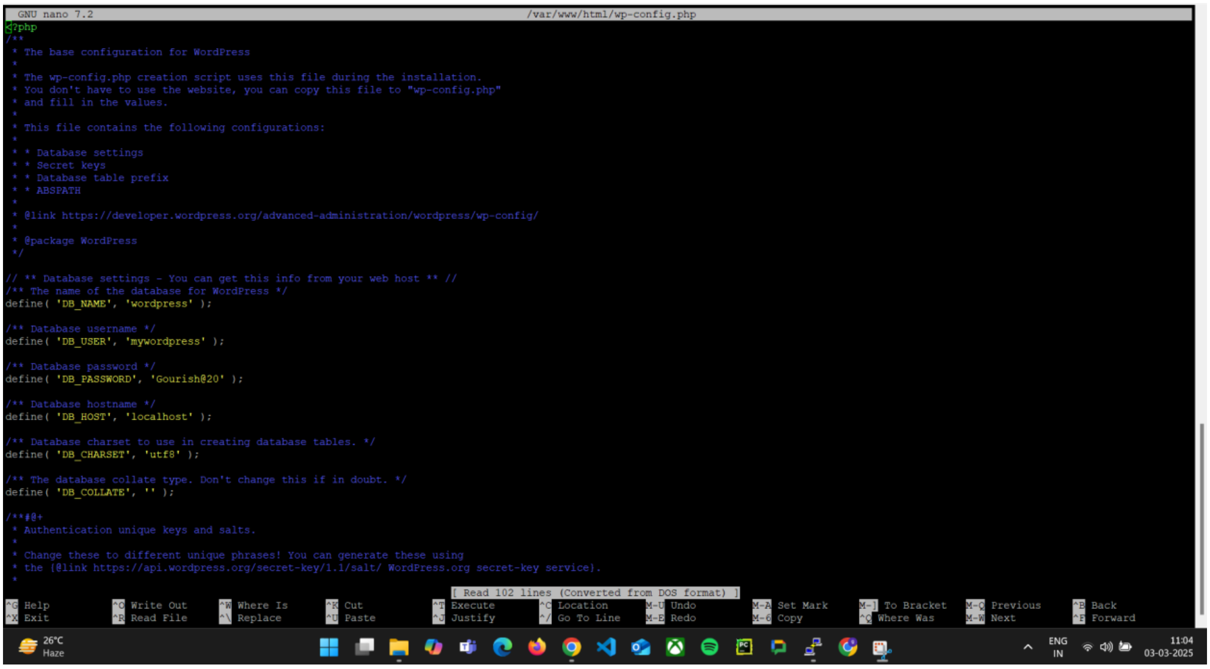
```
mysql> CREATE DATABASE wordpress; CREATE USER 'mywordpress'@'localhost' IDENTIFIED BY 'Gourish@20'; GRANT ALL PRIVILEGES ON wordpress.* TO 'mywordpress'@'localhost'; exit;^C
mysql> exit
Bye
ubuntu@ip-172-31-14-38:~$ wget https://wordpress.org/latest.zip
--2025-03-03 05:31:59--  https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 28410097 (27M) [application/zip]
Saving to: 'latest.zip.1'

latest.zip.1                    2%[=>                                                   ] 715.59K   124KB/s    eta
```

- The above image which download the wordpress in zip format then we have unzip it to install then it will install all its dependencies.



```
Bye
ubuntu@ip-172-31-14-38:~$ wget https://wordpress.org/latest.zip
--2025-03-03 05:31:59--  https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 28410097 (27M) [application/zip]
Saving to: 'latest.zip.1'

latest.zip.1                    5%[====>                                                ]   1.59M   117KB/s    eta 3m 36s ^
C
ubuntu@ip-172-31-14-38:~$ unzip latest.zip && sudo wordpress/* /var/www/html/
```

- As you can see the command which is installing the wordpress in var/www/html
- The above image says that we have to configure the wordpress using wp-config.php where we add username, password, as well hostname for deployment.



```
  GNU nano 7.2                                        /var/www/html/wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @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', 'mywordpress' );

/** Database password */
define( 'DB_PASSWORD', 'Gourish@20' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * 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}.
 *
                                        [ Read 102 lines (Converted from DOS format) ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-Q Previous  ^B Back
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was   M-W Next      ^F Forward
```

- The above image which shows that we configured the file by defining:
- DB_name, DB_USER, DB_Pasowrd and host

5.Configure MySQL Database:

- sudo mysql -u root -p
- CREATE DATABASE wordpress;
- CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'password';
- GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'localhost';
- FLUSH PRIVILEGES;
- EXIT;

6. Configure Apache for WordPress

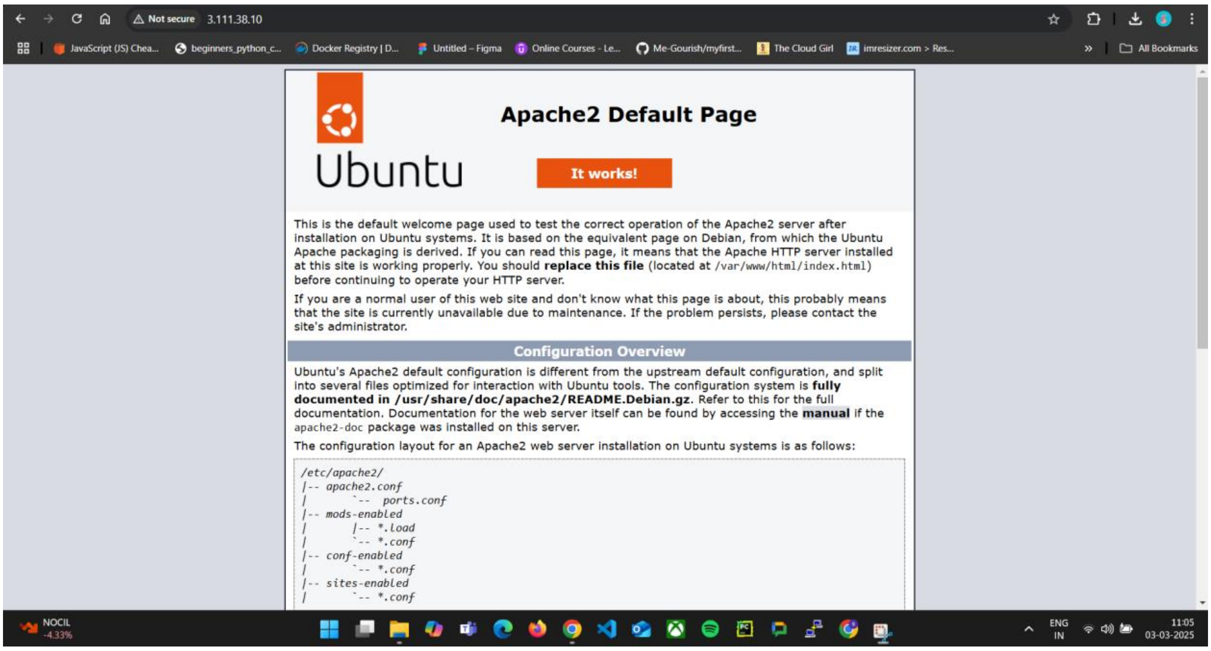
- sudo nano /etc/apache2/sites-available/wordpress.conf

Add the following content:

- <VirtualHost *:80>
-   DocumentRoot /var/www/html/wordpress
-   <Directory /var/www/html/wordpress>
-     AllowOverride All
-   </Directory>
- </VirtualHost>

7. Complete WordPress Setup:

- Visit http://your-ec2-public-ip/wordpress in a web browser.
- Follow the WordPress installation steps.
- Create a homepage using WordPress editor.



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

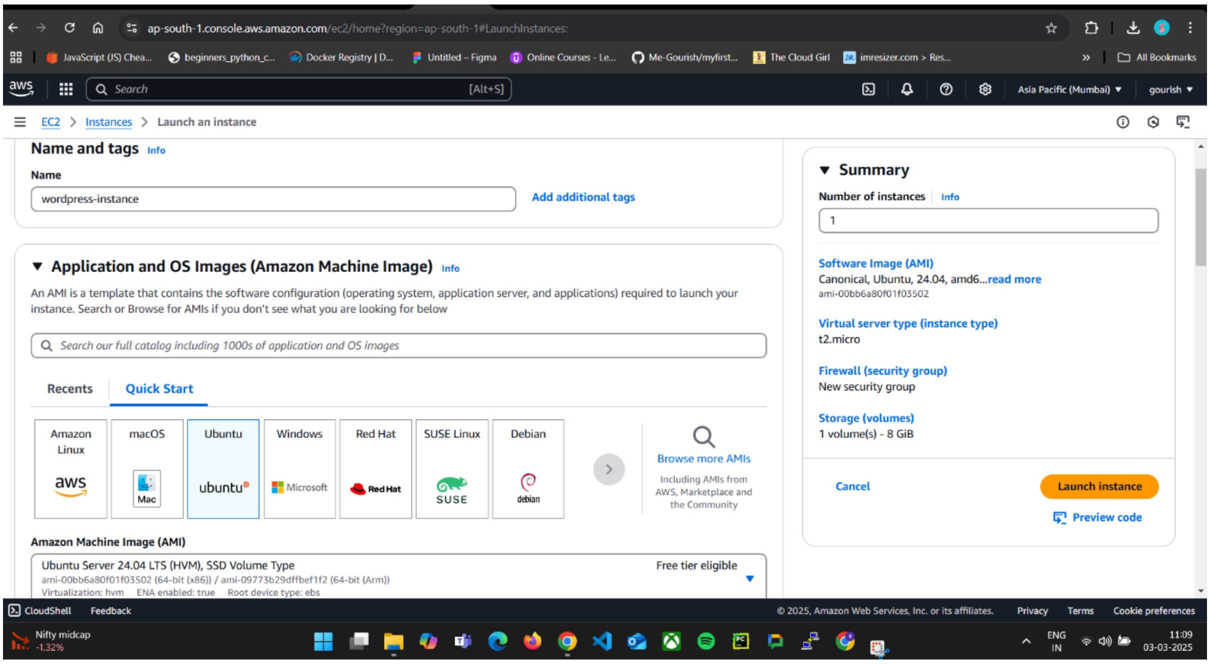
- The above image shows apache2 default page.

Microservices Architecture Deployment

This setup uses separate EC2 instances: one for WordPress and another for MySQL

A. MySQL Server Instance:

1. **Launch an EC2 instance** with Ubuntu.
2. **Configure the security group**:
   - Allow SSH (22) for access.
   - Allow MySQL (3306) **only from WordPress instance**



Name and tags

Name

wordpress-instance

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-00bb6a80f01f03502 (64-bit (x86)) / ami-09773b29dffbef1f2 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
EC2 > Instances > Launch an instance
Add additional tags
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Cancel
Launch instance
Preview code
Asia Pacific (Mumbai)
gourish
© 2025, Amazon Web Services, Inc. or its affiliates.
11:09
03-03-2025

3. **Install and Configure MySQL:**

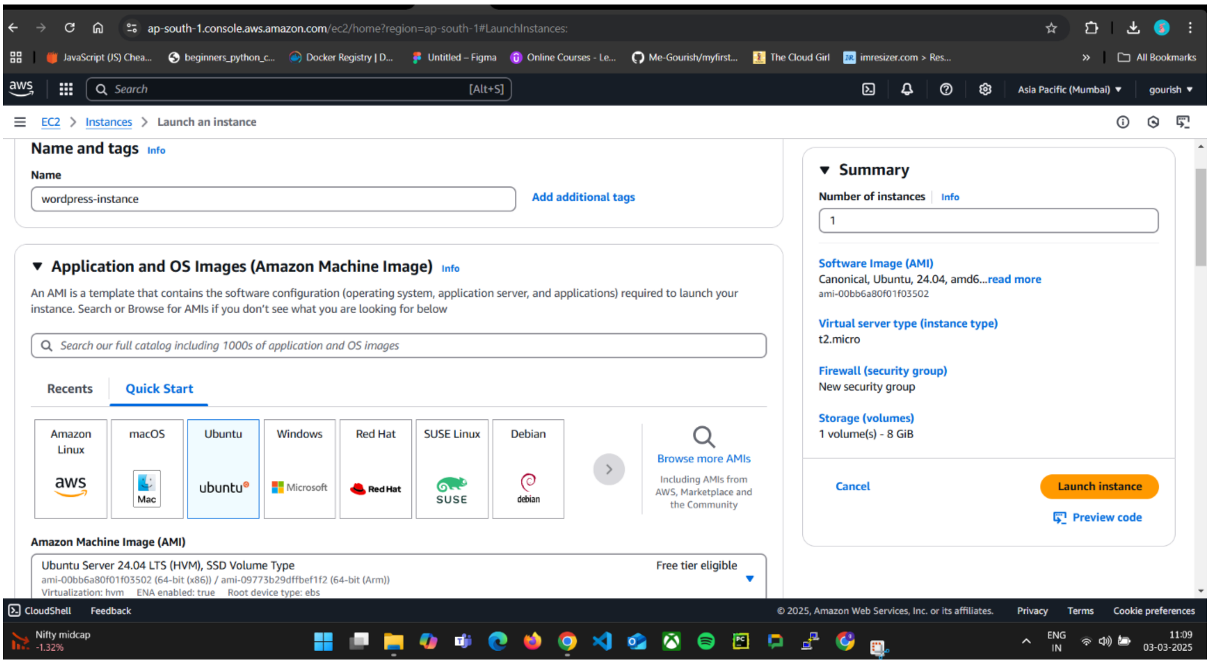
Installing mysql the commands as follows:

- sudo apt update
- sudo apt install mysql-server -y
- sudo mysql_secure_installation

4. **Create a Database and User:**
    - sudo mysql -u root -p
    - CREATE DATABASE wordpress;
    - CREATE USER 'wpuser'@'%' IDENTIFIED BY 'password';
    - GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'%';
    - FLUSH PRIVILEGES;
    - EXIT;
5. **Edit MySQL Configuration:**
    - sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
    - Change bind-address = 127.0.0.1 to bind-address = 0.0.0.0.Restart

Note:The connection step and creating user in database same as follows in the monolithic architecture so I had just showed here creating instance.

B. WordPress Server Instance:



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

EC2 > Instances > Launch an instance

Name and tags Info

Name: wordpress-instance

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian | Browse more AMIs (Including AMIs from AWS, Marketplace and the Community)

Amazon Machine Image (AMI): Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-00bb6a80f01f03502 (64-bit (x86)) / ami-09773b29dffbef1f2 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type): t2.micro

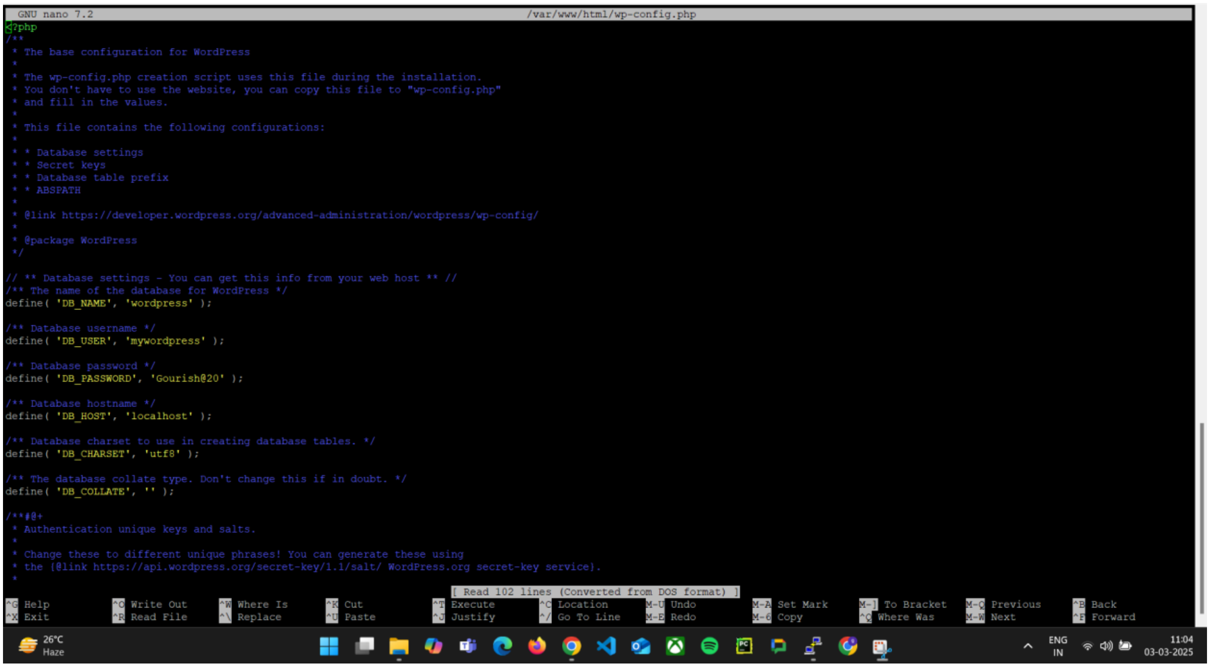
Firewall (security group): New security group

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

The above images shows creating ec2 instance for wrodpress.

1. **Launch another EC2 instance and install Apache, PHP, and WordPress (same steps as monolithic setup, but without MySQL).**
2. **Modify WordPress Configuration:**

- sudo nano /var/www/html/wordpress/wp-config.php



```
  GNU nano 7.2                                              /var/www/html/wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @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', 'mywordpress' );

/** Database password */
define( 'DB_PASSWORD', 'Gourish@20' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * 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}.
 *
                                                  [ Read 102 lines (Converted from DOS format) ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket M-Q Previous   ^B Back
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was   M-W Next       ^F Forward
```

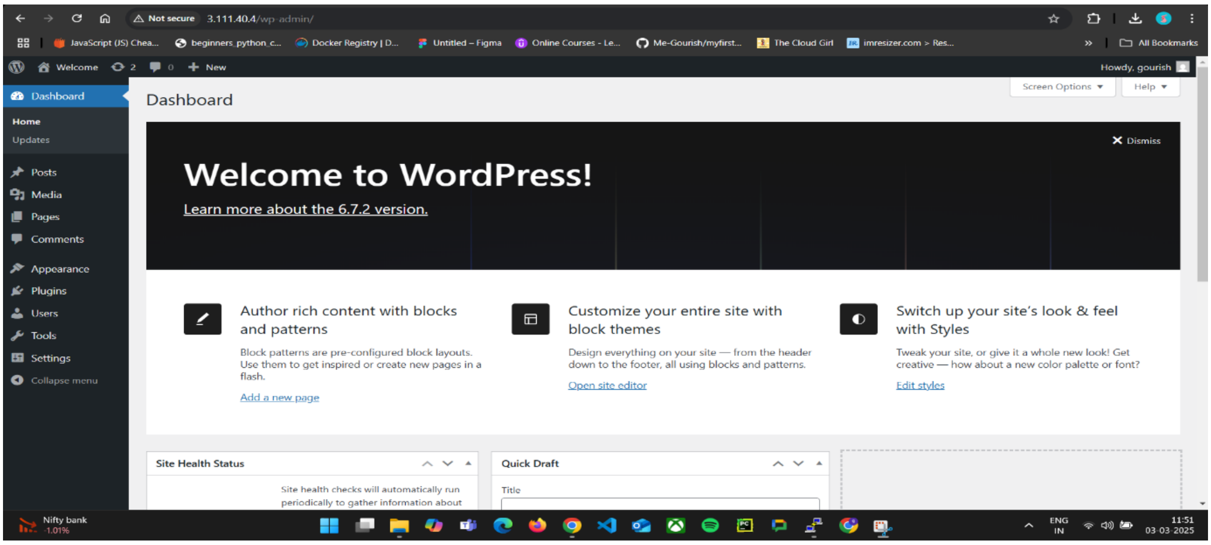
The above image shows changing configuration in wordpress database.

- Update database details:
- define('DB_NAME', 'wordpress');
- define('DB_USER', 'wpuser');
- define('DB_PASSWORD', 'password');
- define('DB_HOST', 'your-mysql-ec2-public-ip');

3. Restart Apache and Access WordPress:
   - sudo systemctl restart apache2:

C. Security Goroup and configuration:

Instance	Port	Protocol	Source



Dashboard

Welcome to WordPress!

Learn more about the 6.7.2 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Open site editor

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

Site Health Status

Site health checks will automatically run periodically to gather information about

Quick Draft

Title

Wordpress	**80, 443**	**TCP**	**0.0.0.0/0**
Wordpress	**22**	**TCP**	**My_IP**
MYSQL	**3306**	**TCP**	**Private ip of wordpress instance**

D.Official Documentation Links

- AWS EC2 User Guide
- Ubuntu Installation Guide
- Apache Documentation
- MySQL Documentation
- WordPress Installation

Conclusion:

This report outlines the deployment of WordPress and MySQL in both monolithic and microservices architectures. Following the steps ensures a secure and efficient setup on AWS EC2 instances. The security groups and configuration files ensure that best practices are followed.