

INFOTRIXS INTERNSHIP

Task1:

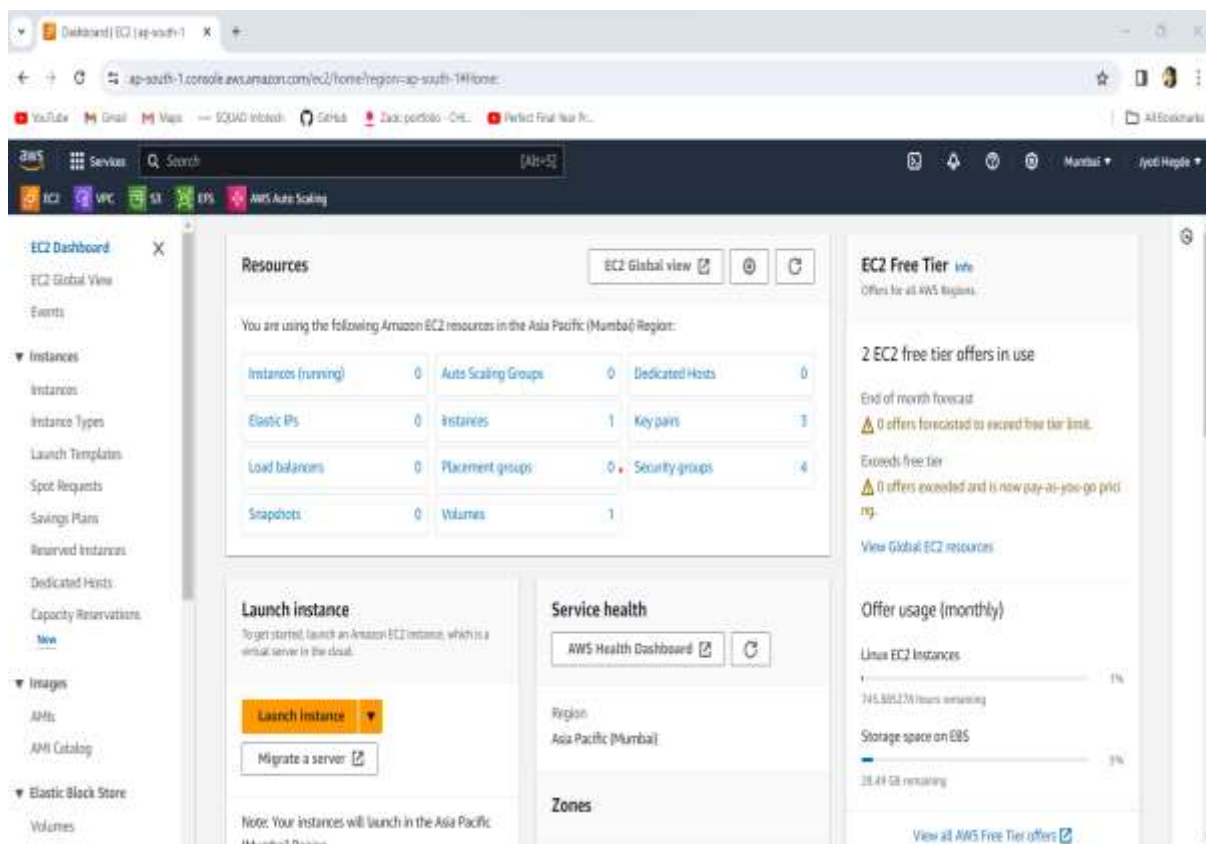
- A. For monolith: 1 EC2 instance create, deploy WordPress and MYSQL on the same instance.
- B. For microservice: 2 EC2 instance create, 1 for WordPress and 1 for MYSQL on the different instance.
- C. Create a welcome page in WordPress that will be the homepage.

Pre-requisites:

- Configure the necessary security group for the instances.
- EC2 instance type: t2 micro, AMI: Ubuntu

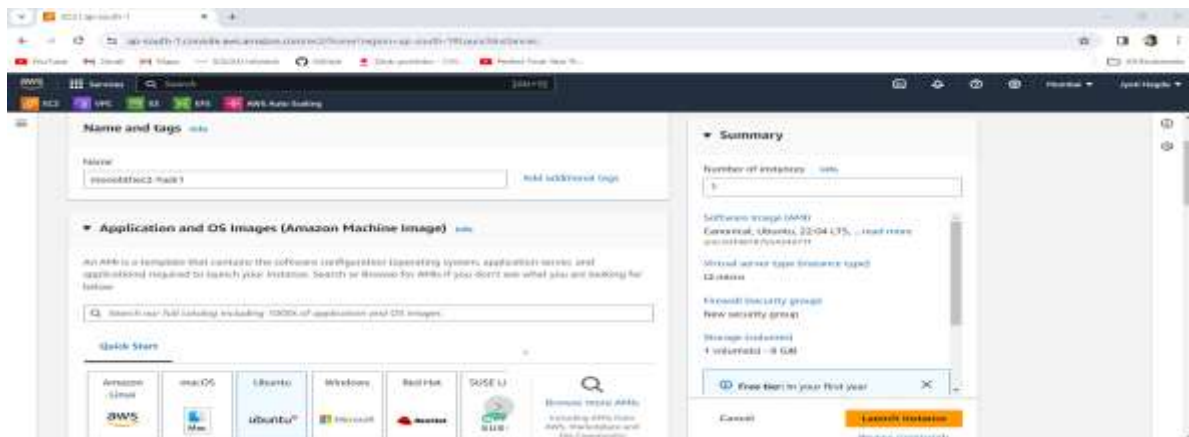
❖ Lab Performed on Monolith: [A]

Step1: Login to AWS free tier account. Search for EC2 instance in the dashboard.

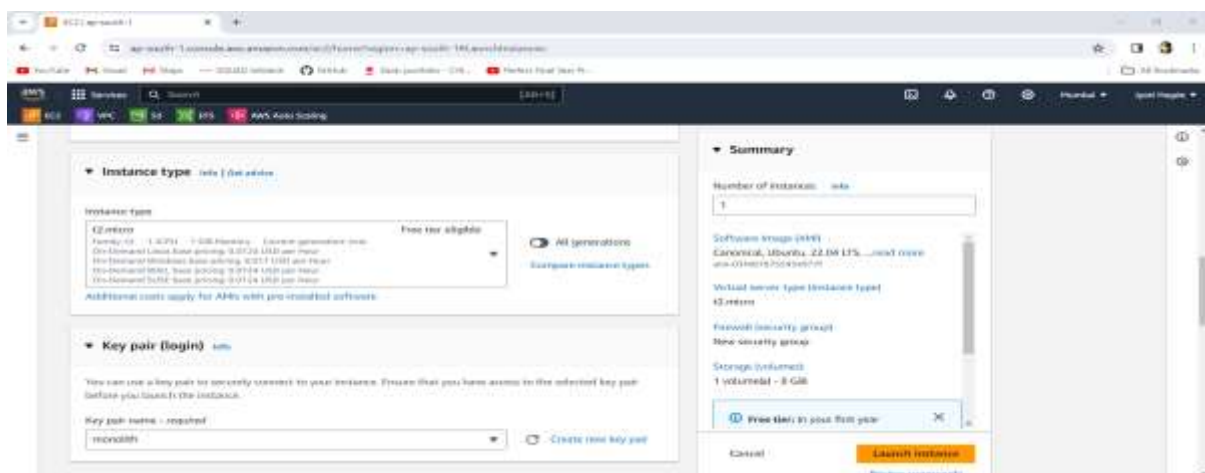


Step 2: Create one EC2 instance.

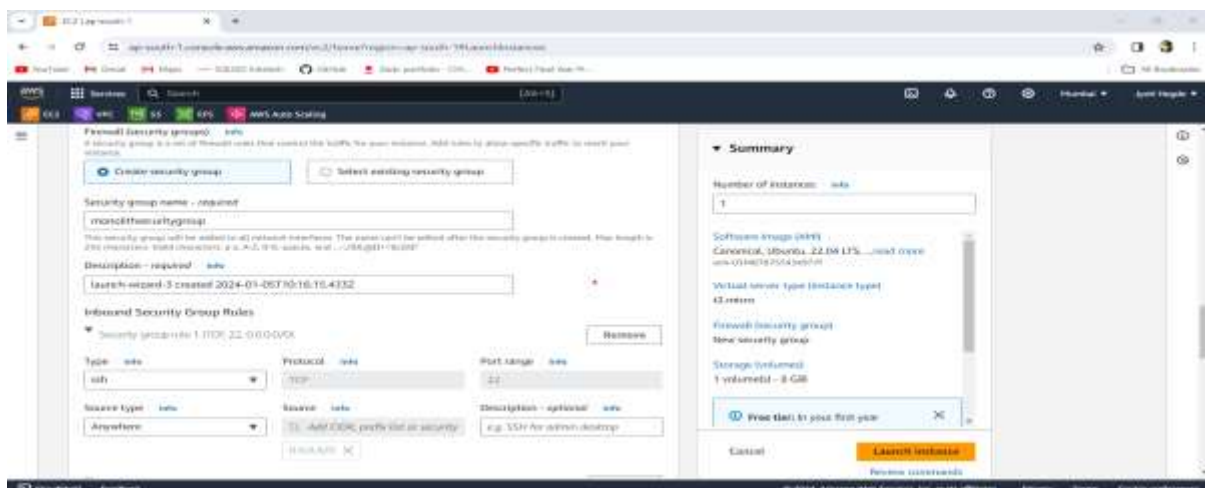
- Name: monolithec2-task1
- AMI: Ubuntu.

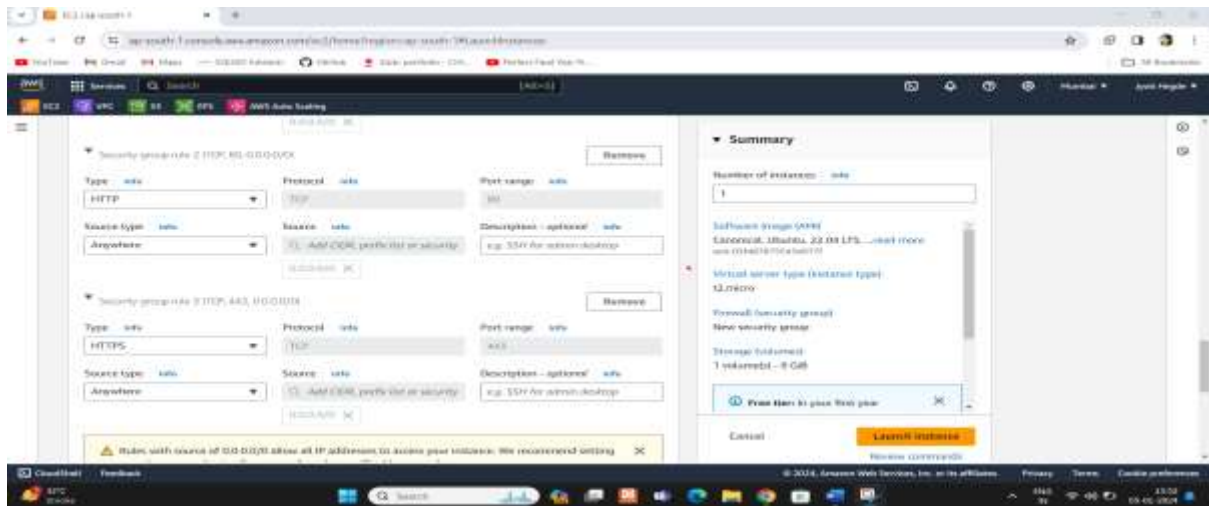


- Instance Type: t2.micro
- Key pair: monolith-key (new created)

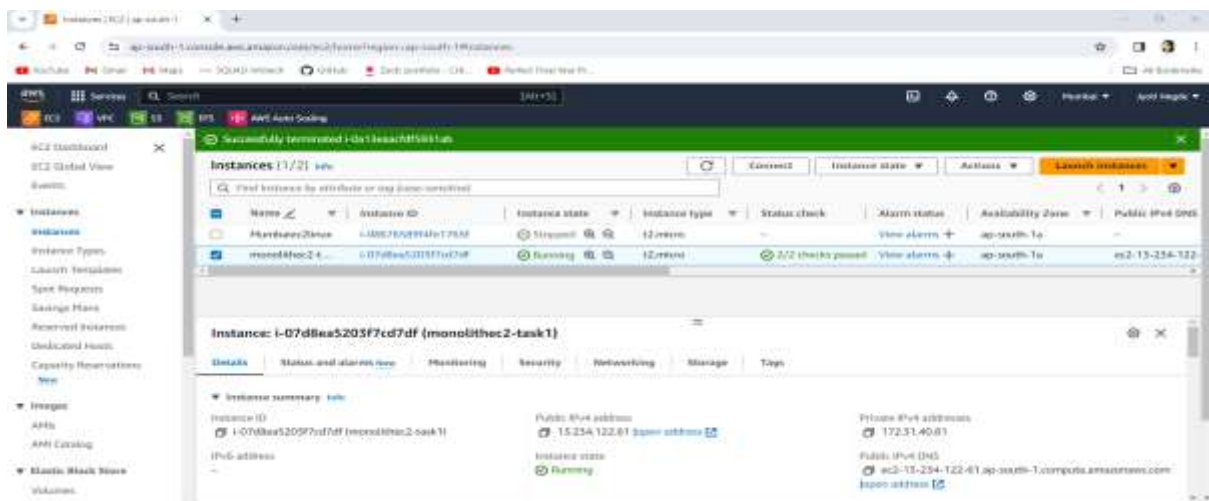


Step 3: allowed inbound traffic for SSH, HTTP/HTTPS (port 80 and 443) for WordPress access and port 3306 for MySQL access.

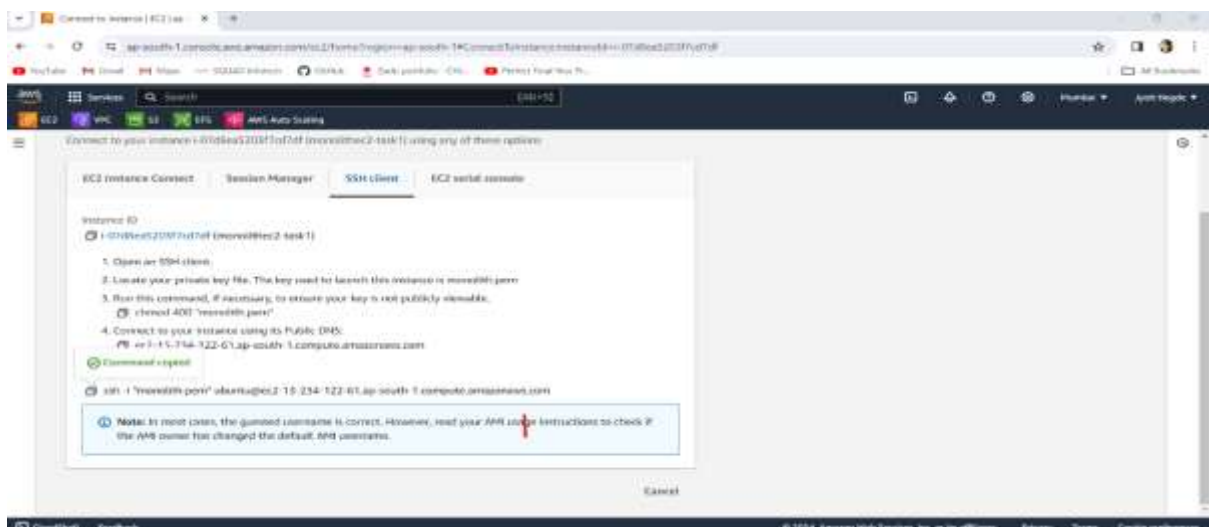




Step 4: The EC2 Instance launched Successfully->Click on connect to further process.



Step 5: To connect the Ubuntu, we need to copy the ssh.



Step 6: Open terminal, go to the download folder (where our security key is downloaded)

➔ **cd downloads**

paste the ssh path (copied above) on terminal to connect our Ubuntu server with our EC2 instance.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sachi> cd downloads
PS C:\Users\sachi\downloads> ssh -i "monolith.pem" ubuntu@ec2-13-234-122-61.ap-south-1.compute.amazonaws.com
```

Step 7: Installed apache2 server on Ubuntu.

➔ **sudo apt install apache2**

```
ubuntu@ip-172-31-40-81:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support
  ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 2139 kB of archives.
After this operation, 8518 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2 [92.8 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.2 [11.3 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.2 [9170 B]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1build1 [140 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.7 [1346 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.7 [165 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.7 [88.8 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mailcap all 3.70+nmu1ubuntu1 [23.8 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mime-support all 3.86 [3696 B]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.7 [97.8 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bzip2 amd64 1.0.8-5build1 [34.8 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ssl-cert all 1.1.2 [17.4 kB]
Fetched 2139 kB in 0s (31.2 MB/s)
```

Step 8: Now copy the public IP of the EC2 and paste on the new tab

➔ <http://<public-ip>>

The Apache2 Default page open means our wordpress is up and running successfully.



Step 9: Installed php runtime and php mysql connector.

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

```
ubuntu@ip-172-31-40-81:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.1 php-common php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php8.1 php php-common php-mysql php8.1 php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 5265 kB of archives.
After this operation, 21.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:92ubuntu1 [12.4 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-common amd64 8.1.2-1ubuntu2.14 [1127 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-opcache amd64 8.1.2-1ubuntu2.14 [365 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-readline amd64 8.1.2-1ubuntu2.14 [13.6 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-cli amd64 8.1.2-1ubuntu2.14 [1834 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapache2-mod-php8.1 amd64 8.1.2-1ubuntu2.14 [1766 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapache2-mod-php all 2:8.1+92ubuntu1 [2898 B]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1 all 8.1.2-1ubuntu2.14 [9158 B]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php all 2:8.1+92ubuntu1 [2756 B]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-mysql amd64 8.1.2-1ubuntu2.14 [130 kB]
```

Step 10: Intalled MySql Server.

➔ **Sudo apt install mysql-server**

```
ubuntu@ip-172-31-40-81:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7 libcgi-bin libcgi-perl libcgi9lobl libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23
  libtime-date-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0
  mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinysa
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7 libcgi-bin libcgi-perl libcgi9lobl libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23
  libtime-date-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server
  mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 0 not upgraded.
Need to get 29.6 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7212 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.35-0ubuntu0.22.04.1 [2682 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-8.0 amd64 8.0.35-0ubuntu0.22.04.1 [22.7 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-pthreads-2.1-7 amd64 2.1.12-stable-1build1 [7642 B]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libmecab2 amd64 0.996-14build9 [199 kB]
```

Step 11: Login to MySql Server

➔ `sudo mysql -u root`

```
ubuntu@ip-172-31-40-81:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, 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>
```

Step 12: Change authentication plugin to mysql_native_password (Choose strong Password)

```
mysql> ALTER USER 'root'@localhost IDENTIFIED WITH mysql_native_password BY 'jyoti123';
Query OK, 0 rows affected (0.01 sec)
```

Step 13: Create new database user for wordpress

```
mysql> CREATE USER 'jyoti'@localhost IDENTIFIED BY 'jyoti123';
Query OK, 0 rows affected (0.02 sec)
```

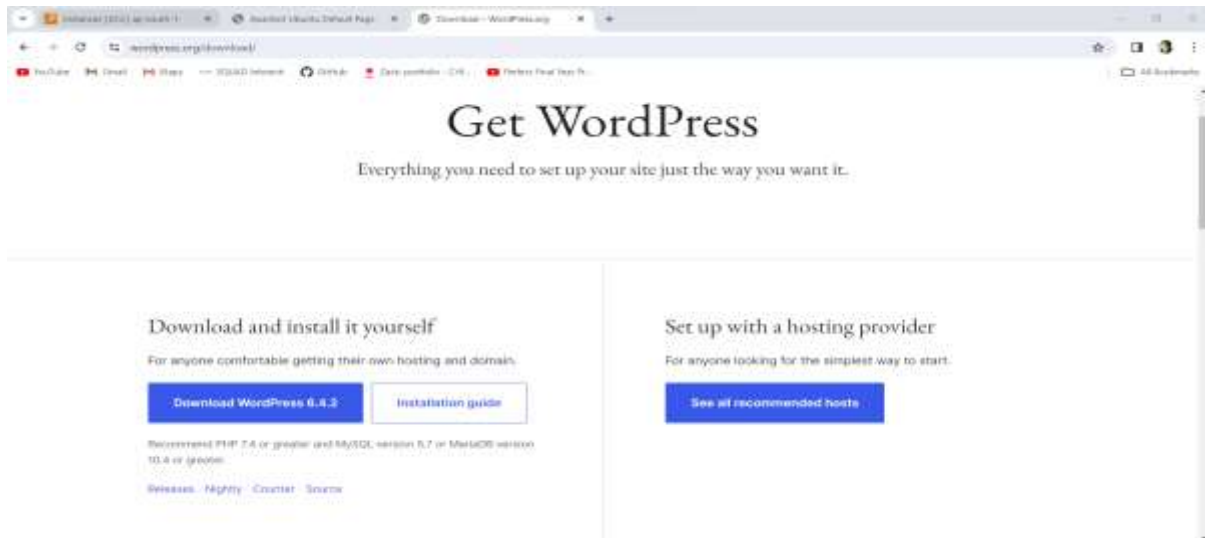
Step 14: Create a database for wordpress

```
mysql> CREATE DATABASE jyotiwordpress;
Query OK, 1 row affected (0.01 sec)
```

Step 15: Grant all Privileges on the database 'wordpress' to the newly created user.

```
mysql> GRANT ALL ON *.* TO 'jyoti'@localhost;
Query OK, 0 rows affected (0.01 sec)
```

Step 16: For download wordpress ->go to the wordpress.org website on Google, copy the download link.



Step 17: on terminal type command as:

- ➔ `cd /tmp`
- ➔ `wget https://wordpress.org/latest.tar.gz`

```
ubuntu@ip-172-31-40-81:~$ cd /tmp
ubuntu@ip-172-31-40-81:/tmp$ wget https://wordpress.org/latest.tar.gz
--2024-01-05 11:14:04-- https://wordpress.org/latest.tar.gz
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: 24479697 (23M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz           100%[=====>] 23.34M  6.38MB/s  in 5.3s

2024-01-05 11:14:10 (4.44 MB/s) - 'latest.tar.gz' saved [24479697/24479697]

ubuntu@ip-172-31-40-81:/tmp$ ls
latest.tar.gz                                systemd-private-9d7cb8f4ec13443789df345d38ec8724-systemd-logind.service-nby8nZ
snap-private-tmp                             systemd-private-9d7cb8f4ec13443789df345d38ec8724-systemd-resolved.service-xmrgCt
systemd-private-9d7cb8f4ec13443789df345d38ec8724-apache2.service-ohd8bX  tmp.H1z891hC9
systemd-private-9d7cb8f4ec13443789df345d38ec8724-chromy.service-GyT6Td
ubuntu@ip-172-31-40-81:/tmp$
```

Step 18: Extract the zip file.

```
ubuntu@ip-172-31-40-81:/tmp$ tar -xvf latest.tar.gz
wordpress/
wordpress/xmlrpc.php
wordpress/wp-blog-header.php
wordpress/readme.html
wordpress/wp-signup.php
wordpress/index.php
wordpress/wp-cron.php
wordpress/wp-config-sample.php
wordpress/wp-login.php
wordpress/wp-settings.php
wordpress/license.txt
```

Step 19: Now WordPress is successfully installed.

➔ ls

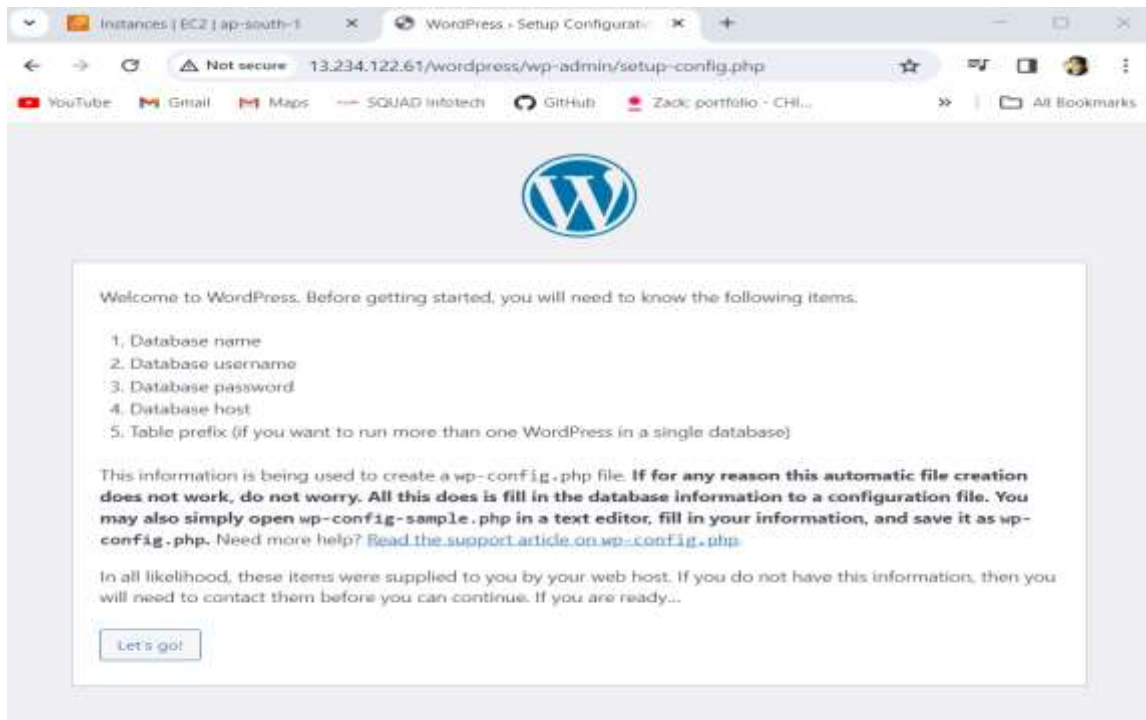
```
ubuntu@ip-172-31-40-81:/tmp$ ls
latest.tar.gz                                systemd-private-9d7cb8f4ec13443789df345d30ec8724-systemd-logind.service-nDyAnZ
snap-private-tmp                             systemd-private-9d7cb8f4ec13443789df345d30ec8724-systemd-resolved.service-xargCt
systemd-private-9d7cb8f4ec13443789df345d30ec8724-apache2.service-oh8r6N    tmp.Malz891hCM
systemd-private-9d7cb8f4ec13443789df345d30ec8724-chrony.service-GyT8Td      wordpress
```

Step 20: wordpress file contain index.html

```
ubuntu@ip-172-31-40-81:/tmp$ sudo mv wordpress/ /var/www/html/
ubuntu@ip-172-31-40-81:/tmp$ cd /var/www/html/
ubuntu@ip-172-31-40-81:/var/www/html$ ls
index.html  wordpress
```

Step 21: Now copy the public IP of the EC2 and paste on the new tab as below,

➔ <http://<public-ip-of-ec2>/wordpress>



Step 22: go to next step and fill all details – database name, username & password.

Click on submit.



The image shows the WordPress installation screen for database configuration. At the top is the WordPress logo. Below it, a message says: "Below you should enter your database connection details. If you are not sure about these, contact your host." The form contains five fields: "Database Name" with the value "jyotiwordpress", "Username" with "jyoti_user", "Password" with "Sachin17@" and a "Hide" button, "Database Host" with "localhost", and "Table Prefix" with "wp_". Each field has a descriptive text below it. At the bottom left is a "Submit" button.

Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name
The name of the database you want to use with WordPress.

Username
Your database username.

Password [Hide](#)
Your database password.

Database Host
You should be able to get this info from your web host, if localhost does not work.

Table Prefix
If you want to run multiple WordPress installations in a single database, change this.

[Submit](#)

Step 23: wp-config.php error occur to fix it copy the code below.



The image shows the WordPress installation screen after an error with wp-config.php. At the top is the WordPress logo. Below it, a message says: "Unable to write to wp-config.php file. You can create the wp-config.php file manually and paste the following text into it." This is followed by "Configuration rules for wp-config.php:" and a code block containing PHP comments. At the bottom, a message says: "After you've done that, click 'Run the installation'." and there is a "Run the installation" button.

Unable to write to wp-config.php file.
You can create the wp-config.php file manually and paste the following text into it.

Configuration rules for wp-config.php:

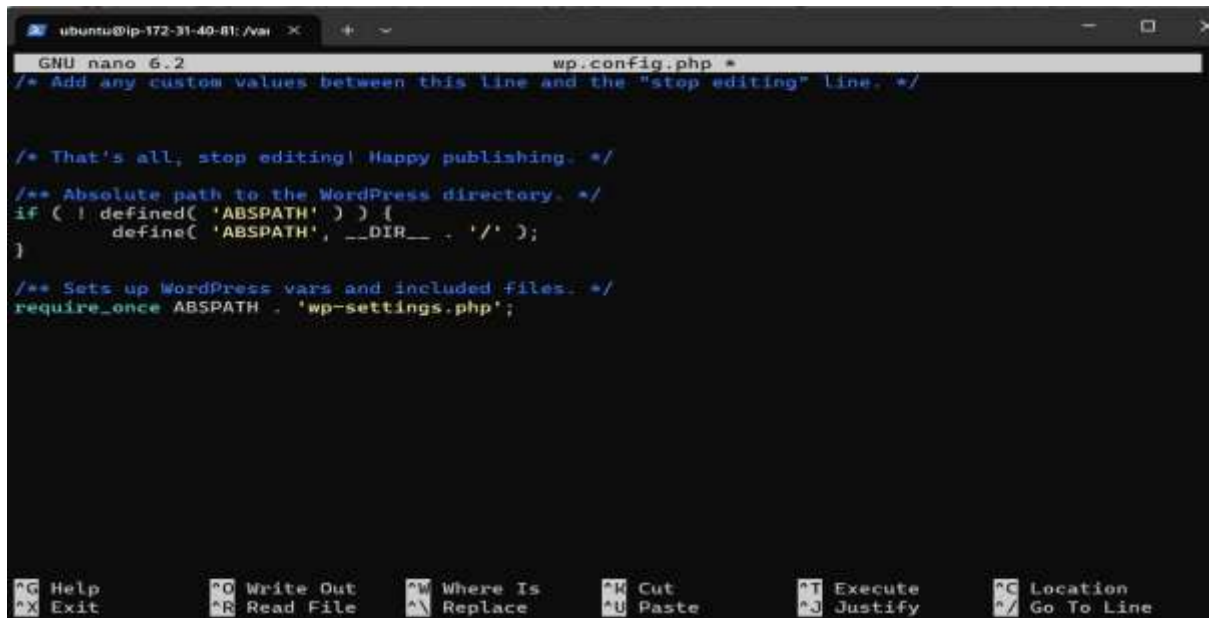
```
<?php
/**
 * The base configuration for WordPress.
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 */
```

After you've done that, click "Run the installation".

[Run the installation](#)

Step 24: on the terminal, inside wordpress folder-> create nano wp.config.php file and paste above copied code.

```
ubuntu@ip-172-31-40-81:/var/www/html$ cd wordpress
ubuntu@ip-172-31-40-81:/var/www/html/wordpress$ nano wp.config.php
```



```
GNU nano 6.2 wp.config.php
/* Add any custom values between this line and the "stop editing" line. */

/* That's all, stop editing! Happy publishing. */

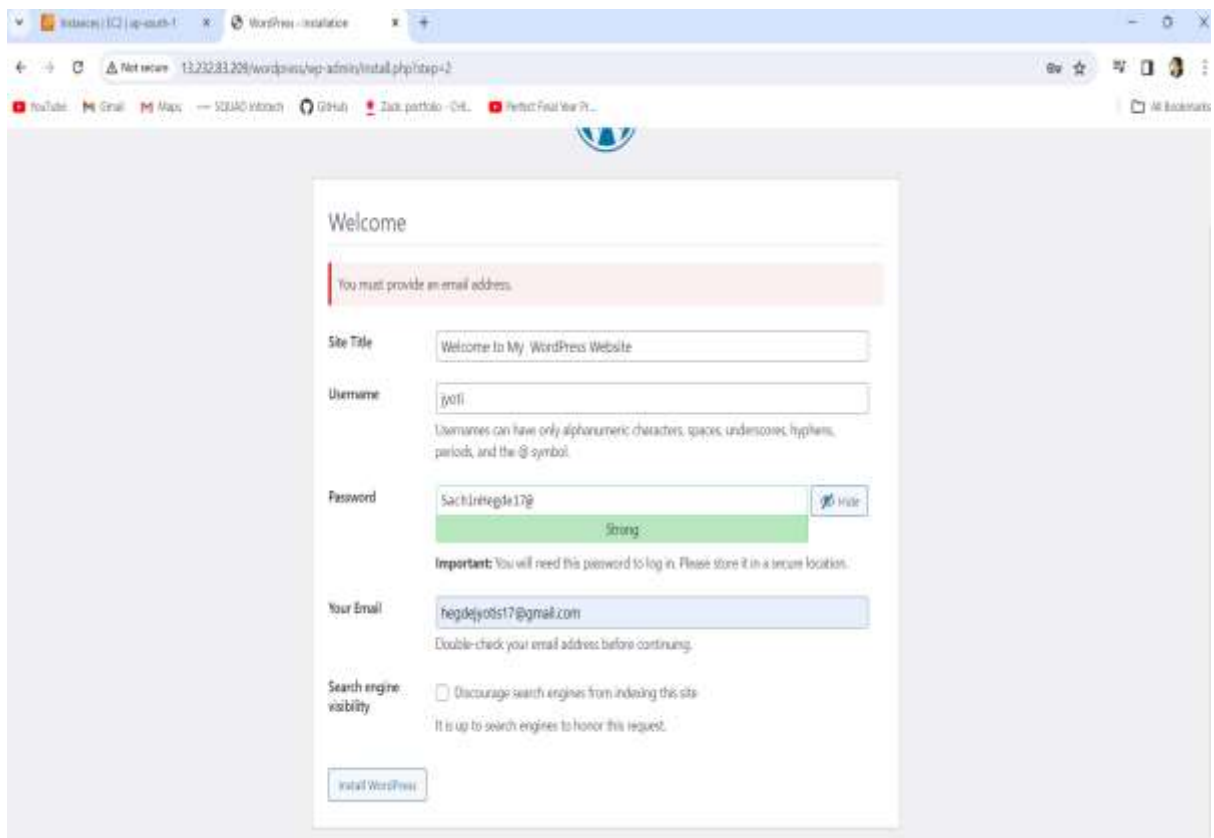
/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';
```

Step 25: Click on Run the installation button.



Step 26: Fill all the details to host/deploy your webpage.



The screenshot shows a web browser window with the URL `13.232.83.208/wordpress/wp-admin/install.php?step=2`. The page is titled "Welcome" and contains a form for installing WordPress. A red error message at the top states "You must provide an email address." The form fields are as follows:

- Site Title:** "Welcome to My WordPress Website"
- Username:** "jyoti". A note below states: "Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol."
- Password:** "Sachin@egde17@". A green bar indicates the password is "Strong". A note below states: "Important: You will need this password to log in. Please store it in a secure location."
- Your Email:** "regdejyoti17@gmail.com". A note below states: "Double-check your email address before continuing."
- Search engine visibility:** A checkbox labeled "Discourage search engines from indexing this site" is unchecked. A note below states: "It is up to search engines to honor this request."

An "Install WordPress" button is located at the bottom of the form.

Step 27: Now install wordpress and Login with username & password.



The screenshot shows the WordPress logo at the top. Below it is a white box with the following content:

Success!

WordPress has been installed. Thank you, and enjoy!

Username: jyoti

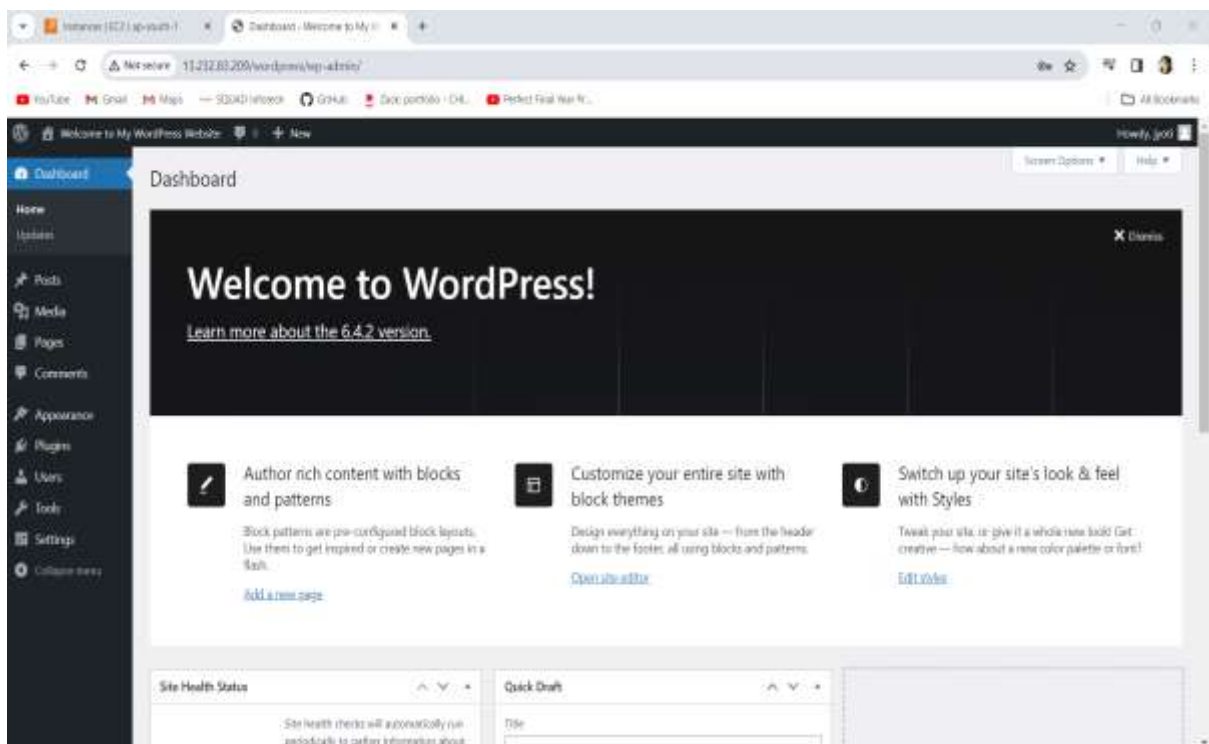
Password: Your chosen password.

[Log In](#)

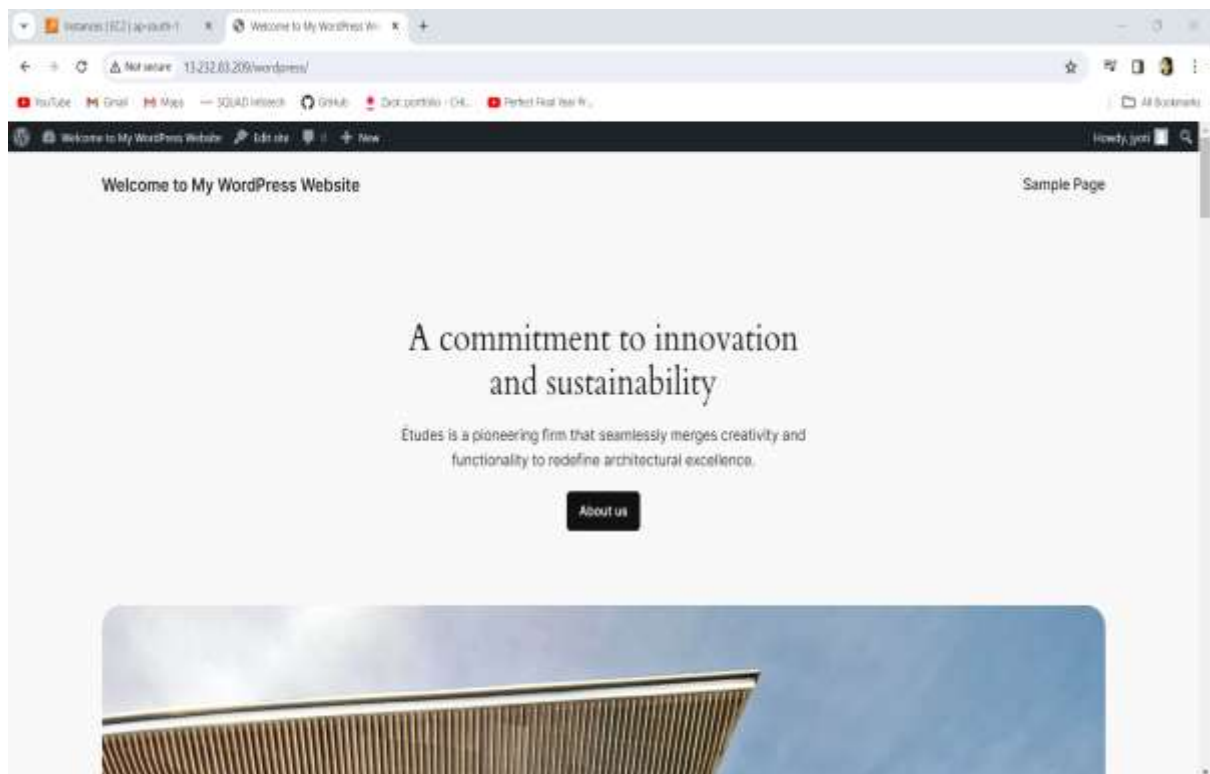


The image shows the WordPress login page. At the top center is the WordPress logo. Below it is a white box containing the login form. The form has two input fields: 'Username or Email Address' with the text 'jyoti' entered, and 'Password' with masked characters. Below the password field is a checkbox labeled 'Remember Me' and a blue 'Log In' button. Under the login box, there is a link 'Lost your password?' and a link 'Go to Welcome to My WordPress Website'.

Step 28: See your first webpage is open successfully on the wordpress.



Welcome to My WordPress Website



Conclusion: successfully performed my 1st task for monolith. In that 1 EC2 instance created, deployed WordPress and MySQL on the same instance and created welcome page as “Welcome to My WordPress Website” as homepage.