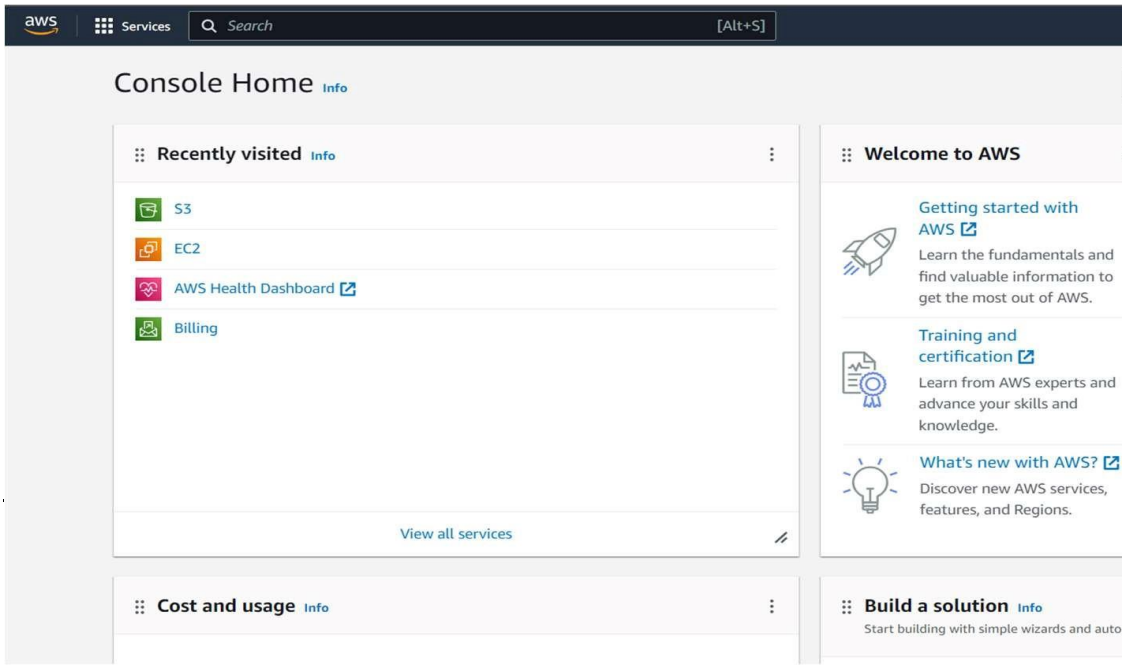
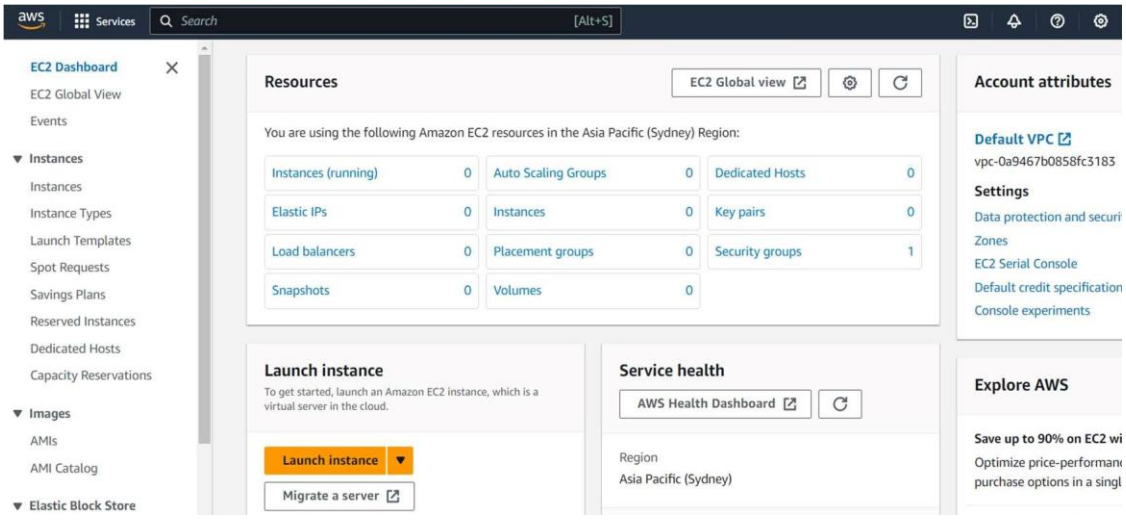
PRACTICAL - 7

Aim : Create a scenario in wordpress for Social marketing, Search Engine and sharing tools.

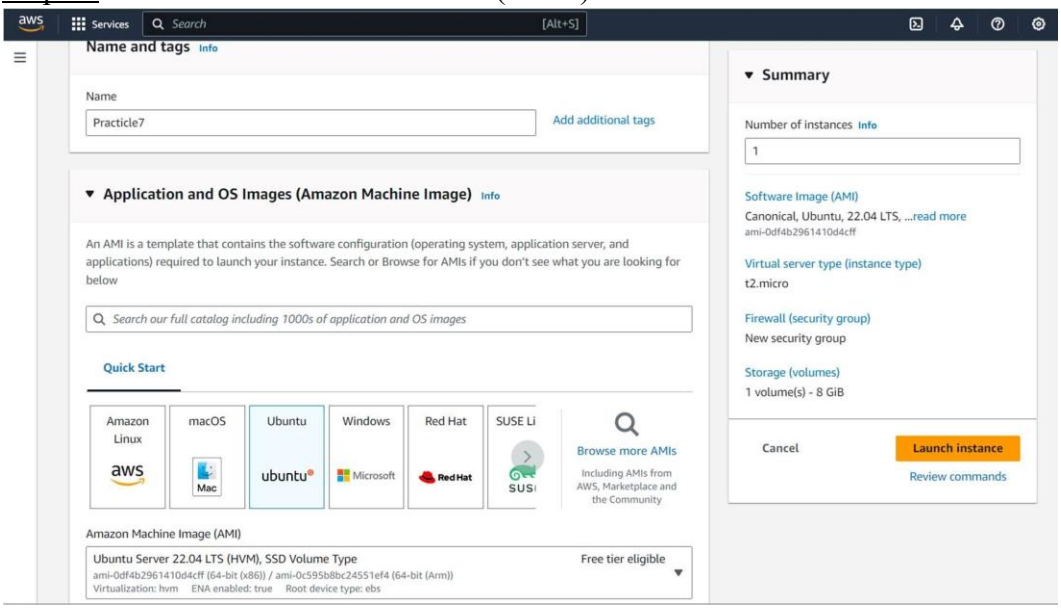
Step 01: Open the AWS Management Console and click on EC2 service of it.



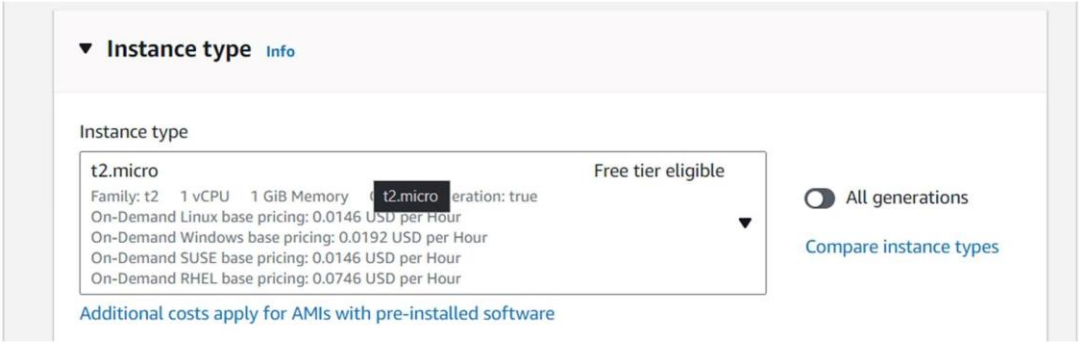
Step 02: Click on the launch instance buΣon to create a new instance.



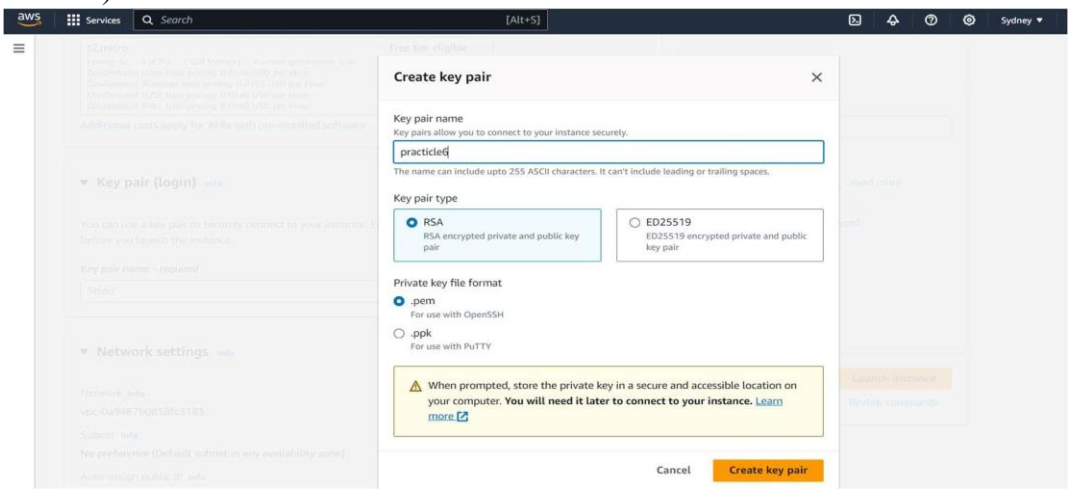
Step 03: Select the Ubuntu 22.04 LTS (HVM) to install ubuntu on instance.



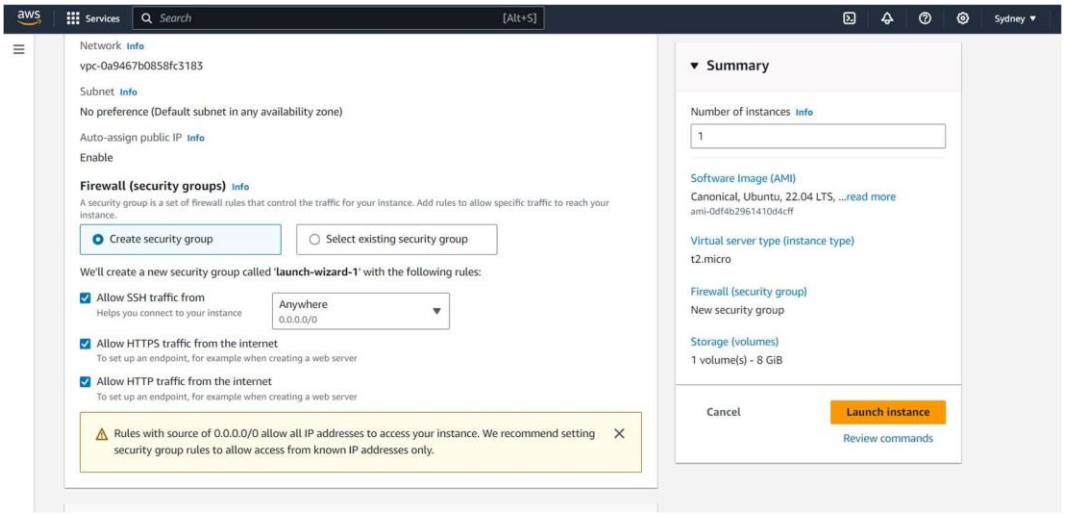
Step 04: select instance type as free tier eligible only.



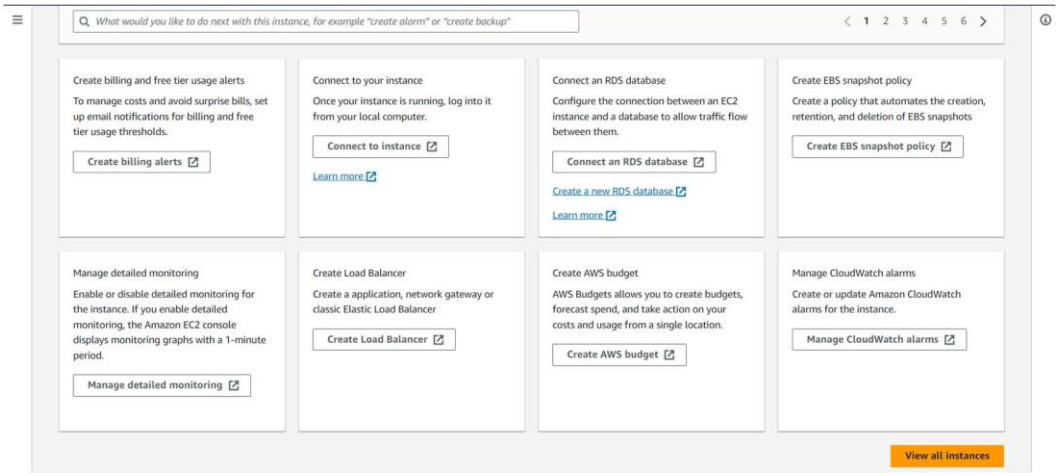
Step 05: Create new key pair as shown below and it will download the .pem file(keep that file).



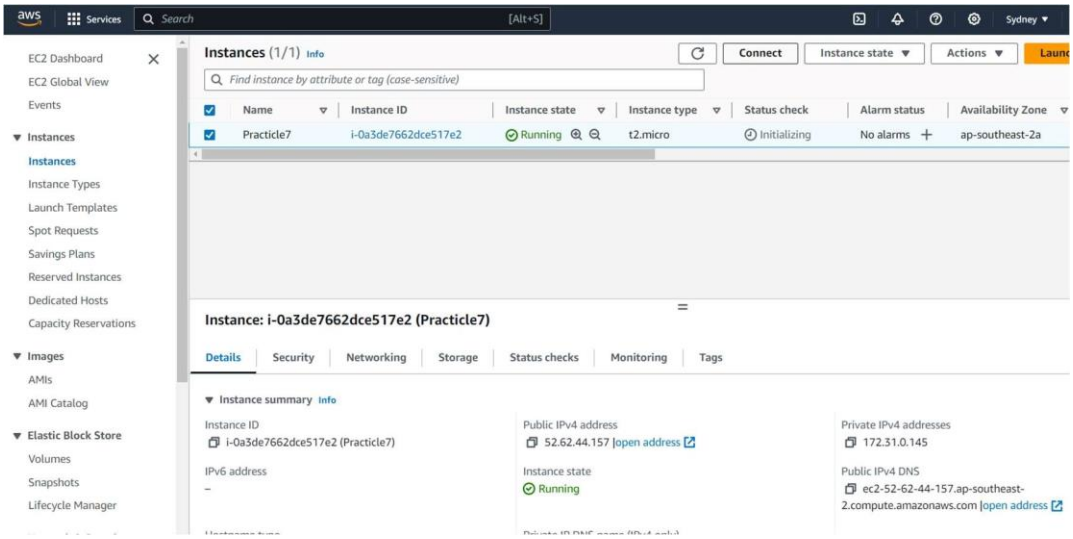
Step 06: In security group allow SSH, HTTPS and HTTP as shown and click on Launch instance.



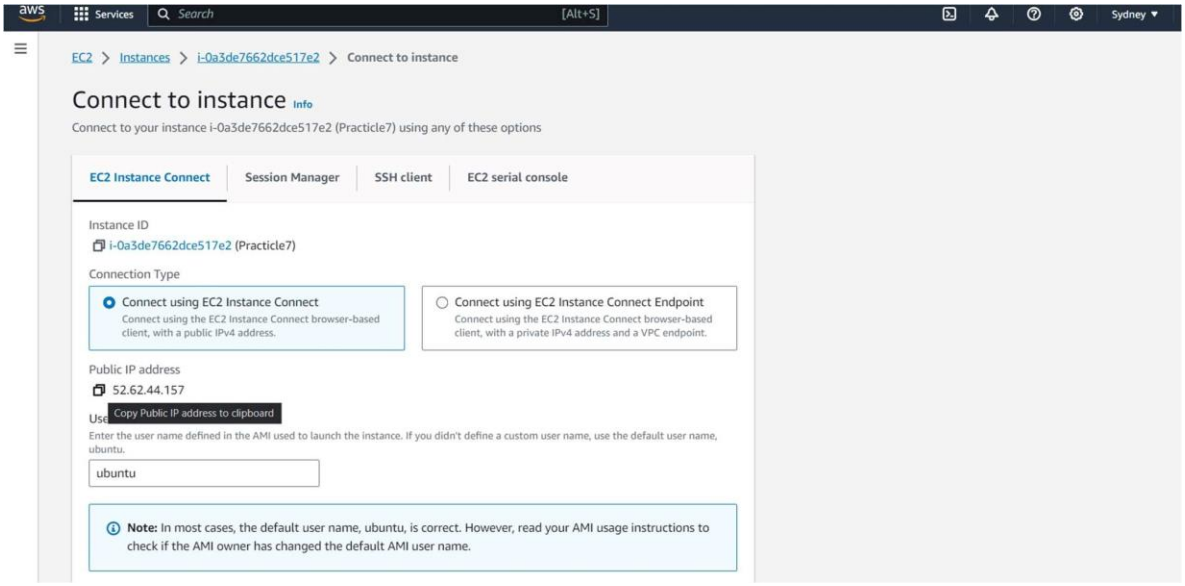
Step 07: Click on view all instances.



Step 08: Select created instance (i.e. practicle7) and click on connect to connect instance.



Step 09: Copy the Public IP address.

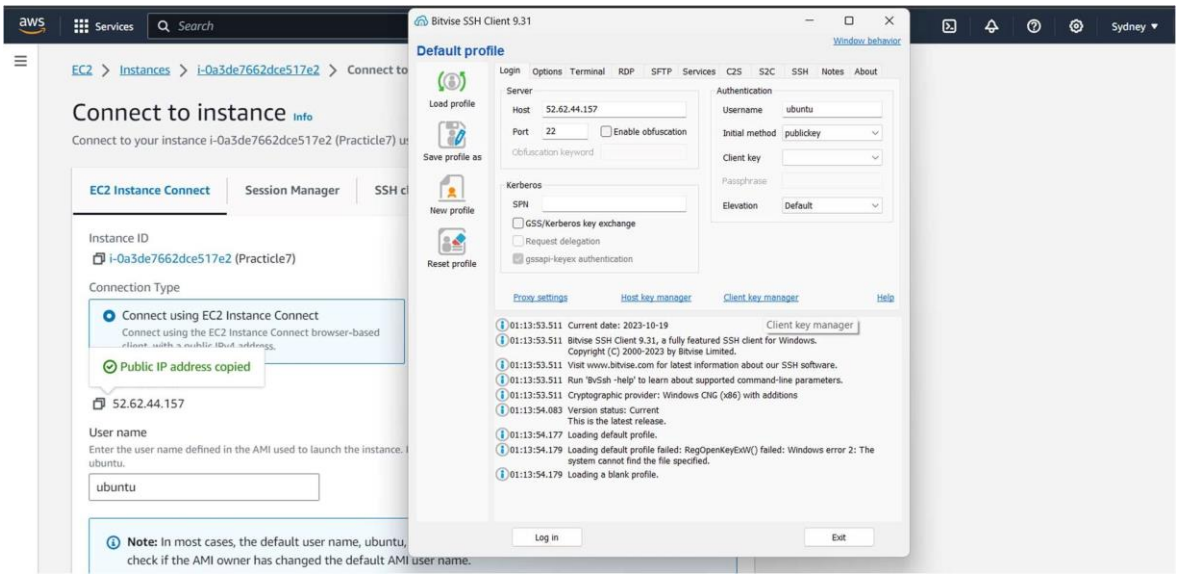


Step 10: Open Bitvise SSH Client soŌfware and paste that copied IP address in Host field. (port- 22)

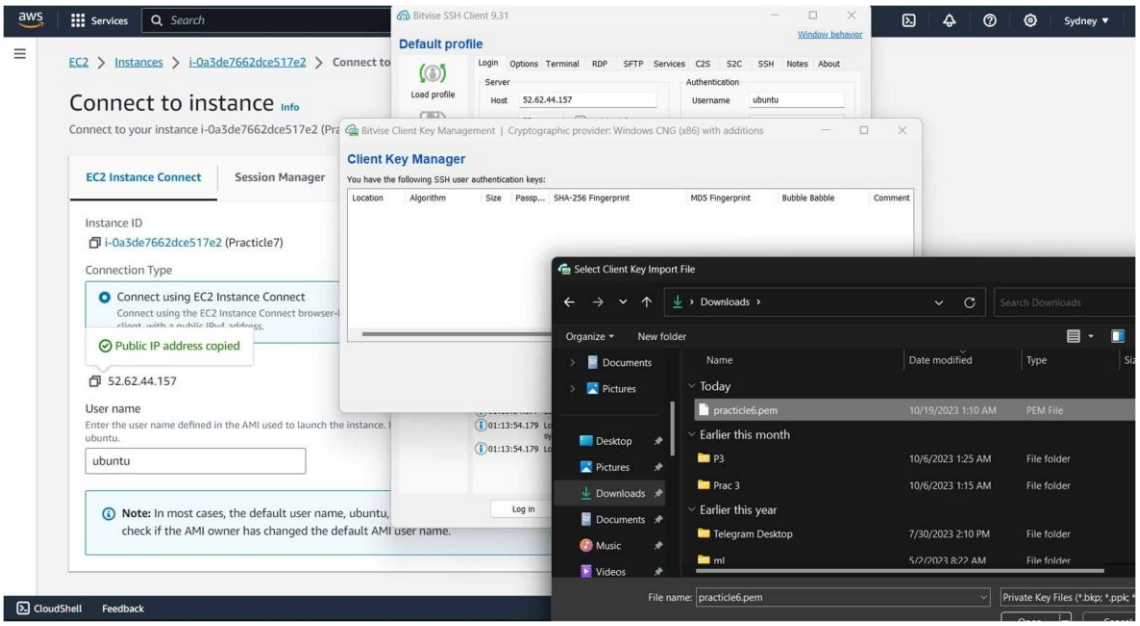
In Authenticaion field username - ubuntu

initial method - publickey
Client key - auto

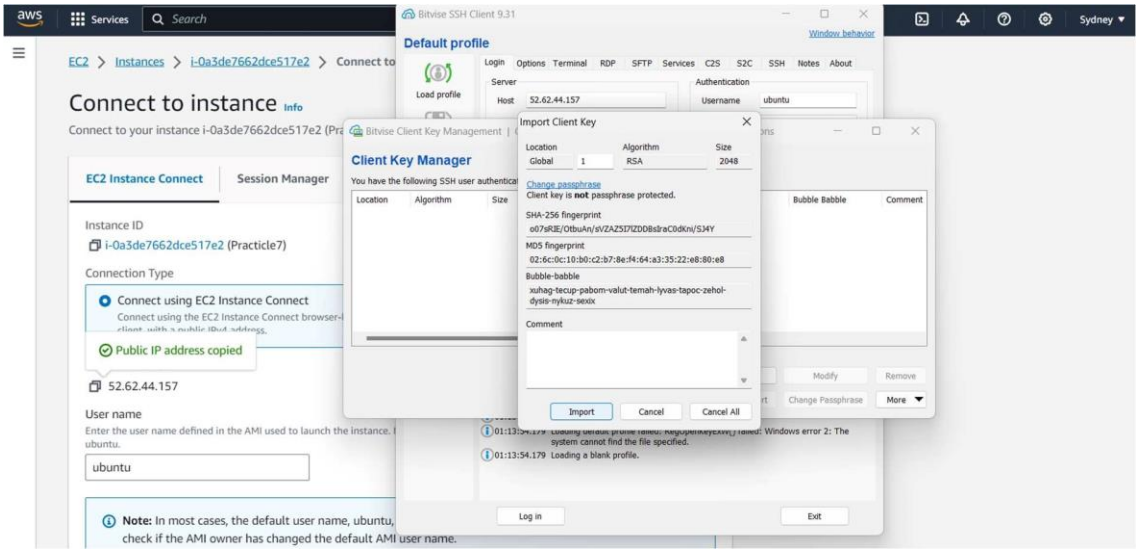
Elevation - default.
And then click on Client key manager.



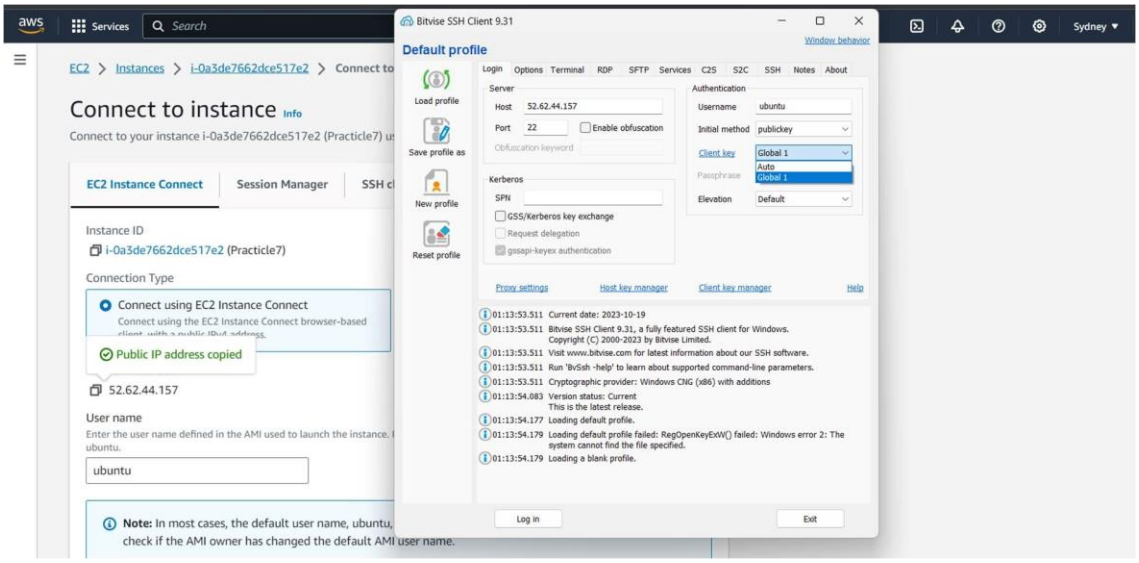
Step 11: Click on import and select .pem file that we have downloaded in step 05 and click on open.



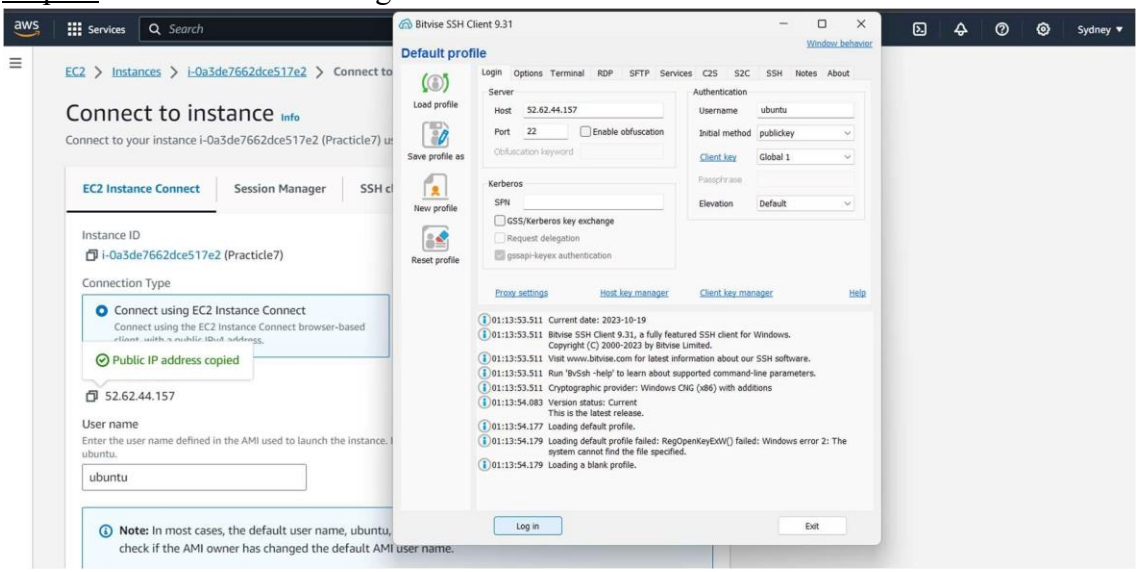
Step 12: Click on import.



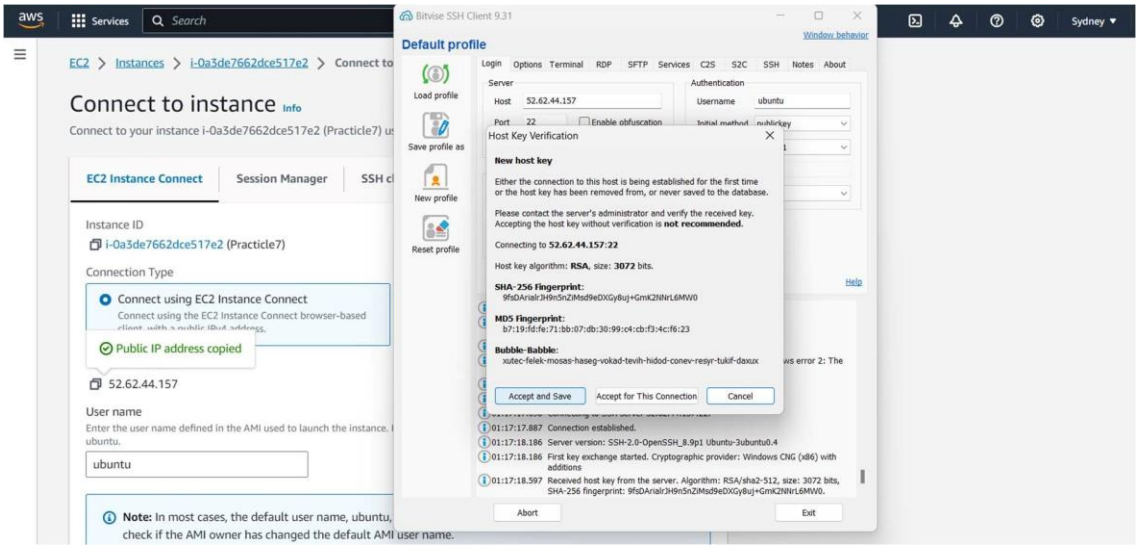
Step 13: Now select Client key as Global 1 as shown.



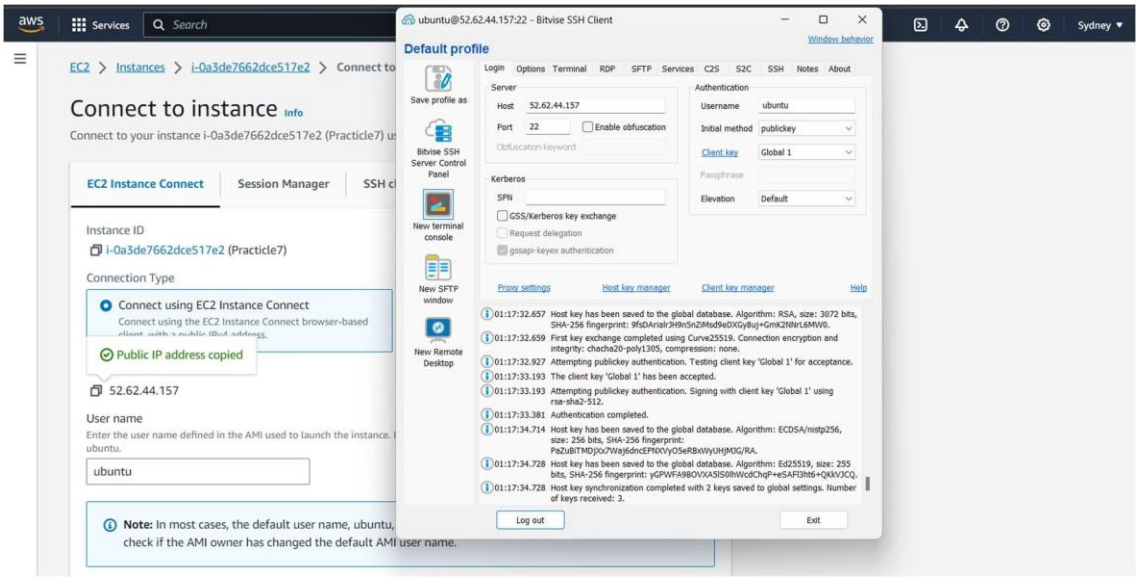
Step 14: And then click on log in.



Step 15: Click on Accept and Save.



Step 16: Click on New terminal control It will enter to ubuntu terminal.

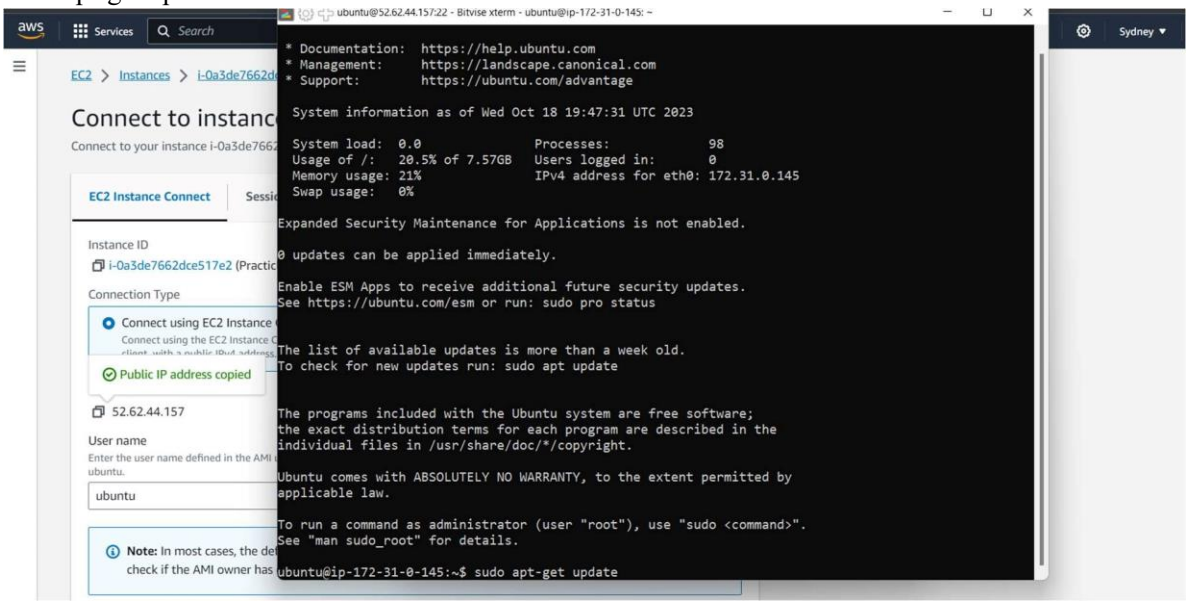


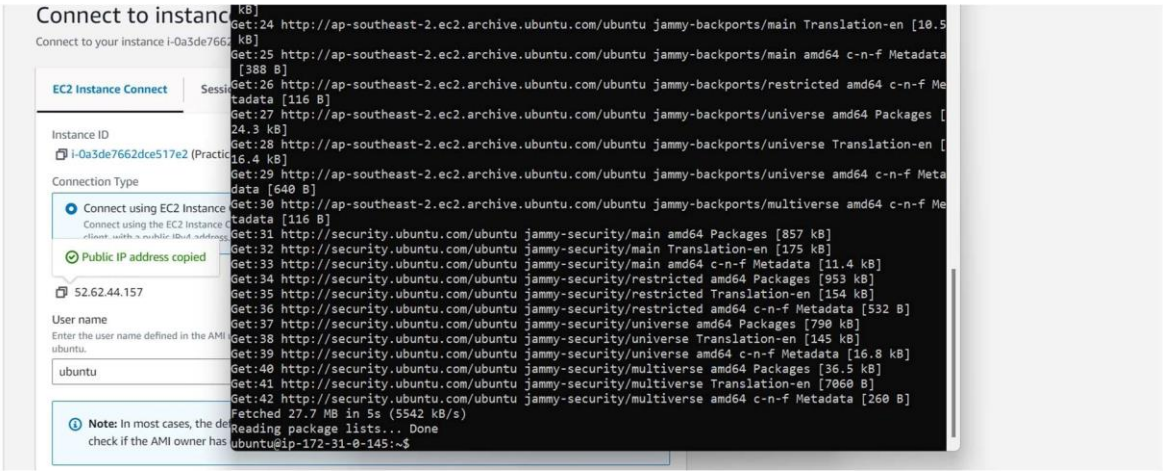
Install Wordpress on ubuntu 22.04

Before we begin, let's update and upgrade the ubuntu system. Login as root user to your system and update the system to update the repositories.

Step 17: To update system use following

sudo apt-get update





Step 18: To upgrade system use following command.
sudo apt-get upgrade

```
Reading package lists... Done
ubuntu@ip-172-31-0-145:~$ sudo apt-get upgrade
```

```
Get:22 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcurl3-gnutls
amd64 7.81.0-1ubuntu1.14 [284 kB]
Get:23 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsgutils2-2 am
d64 1.46-1ubuntu0.22.04.1 [99.0 kB]
Get:24 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 sg3-utils amd64
1.46-1ubuntu0.22.04.1 [841 kB]
Get:25 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 sg3-utils-udev a
ll 1.46-1ubuntu0.22.04.1 [5904 B]
Get:26 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 cloud-init all 2
3.3.1-0ubuntu1~22.04.1 [541 kB]
Fetched 25.5 MB in 1s (47.3 MB/s)
Preconfiguring packages ...
(Reading database ... 64726 files and directories currently installed.)
Preparing to unpack .../libc6_2.35-0ubuntu3.4_amd64.deb ...
Unpacking libc6:amd64 (2.35-0ubuntu3.4) over (2.35-0ubuntu3.3) ...
Setting up libc6:amd64 (2.35-0ubuntu3.4) ...
```

Step 19: Install Apache
To install apache in system you need to switch to root user so ,for use following command sudo su -

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-0-145:~$ sudo su -
```

Step 20: Now for installation of apache use following command
sudo apt-get install apache2

```
ubuntu@ip-172-31-0-145:~$ sudo su -
root@ip-172-31-0-145:~# sudo apt-get install apache2
```

```
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../03-libaprutil1-ldap_1.6.1-5ubuntu4.22.04.2_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1.6.1-5ubuntu4.22.04.2) ...
Selecting previously unselected package liblua5.3-0:amd64.
Preparing to unpack .../04-liblua5.3-0_5.3.6-1build1_amd64.deb ...
Unpacking liblua5.3-0:amd64 (5.3.6-1build1) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../05-apache2-bin_2.4.52-1ubuntu4.6_amd64.deb ...
Unpacking apache2-bin (2.4.52-1ubuntu4.6) ...
Selecting previously unselected package apache2-data.
Preparing to unpack .../06-apache2-data_2.4.52-1ubuntu4.6_all.deb ...
Unpacking apache2-data (2.4.52-1ubuntu4.6) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../07-apache2-utils_2.4.52-1ubuntu4.6_amd64.deb ...
Unpacking apache2-utils (2.4.52-1ubuntu4.6) ...
Selecting previously unselected package mailcap.
Preparing to unpack .../08-mailcap_3.70+nmu1ubuntu1_all.deb ...
Unpacking mailcap (3.70+nmu1ubuntu1) ...
Selecting previously unselected package mime-support.
Preparing to unpack .../09-mime-support_3.66_all.deb ...
Unpacking mime-support (3.66) ...
Selecting previously unselected package apache2.
Preparing to unpack .../10-apache2_2.4.52-1ubuntu4.6_amd64.deb ...
Unpacking apache2 (2.4.52-1ubuntu4.6) ...
```

Step 21: To confirm that Apache is installed on your system use following command.

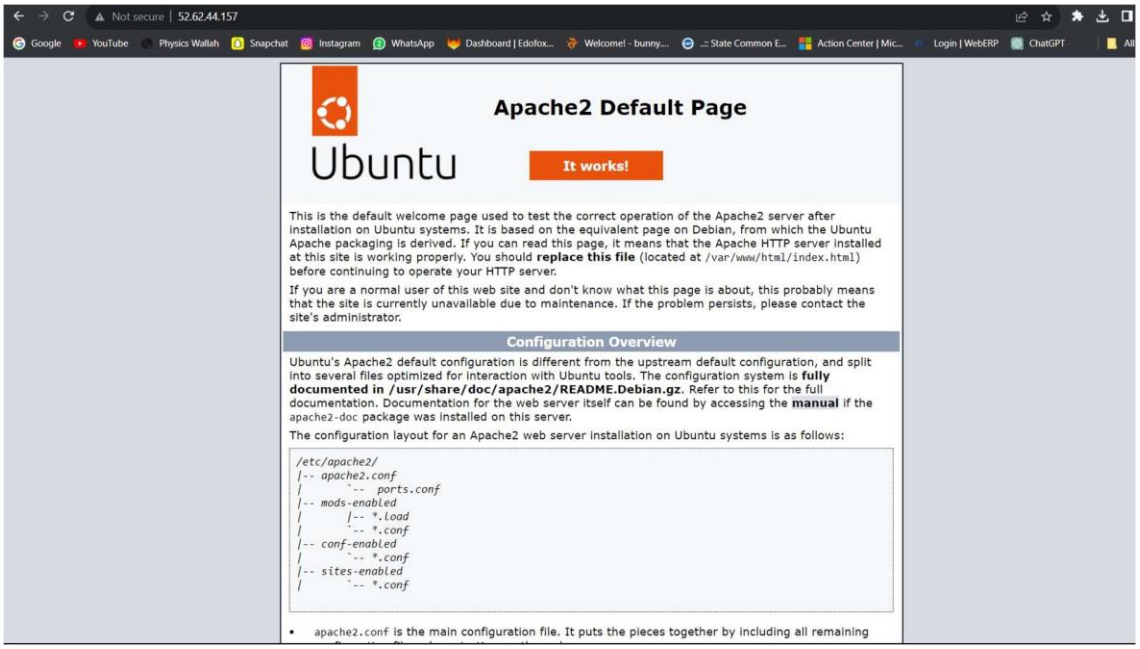
Systemctl status apache2

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-0-145:~# systemctl status apache2
```

```
root@ip-172-31-0-145:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-10-18 19:52:48 UTC; 56s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 8526 (apache2)
      Tasks: 55 (limit: 1121)
     Memory: 5.0M
        CPU: 34ms
     CGroup: /system.slice/apache2.service
             ├─8526 /usr/sbin/apache2 -k start
             ├─8528 /usr/sbin/apache2 -k start
             └─8529 /usr/sbin/apache2 -k start

Oct 18 19:52:48 ip-172-31-0-145 systemd[1]: Starting The Apache HTTP Server...
Oct 18 19:52:48 ip-172-31-0-145 systemd[1]: Started The Apache HTTP Server.
root@ip-172-31-0-145:~#
```

Step 22: To verify further, open your browser and paste Public IP address (from step 09).



Step 23: Install MySQL

Next, we are going to install the MariaDB database engine to hold our WordPress files. MariaDB is an open-source fork of MySQL and most of the hosting companies use it instead of MySQL. To install use following command. apt install mariadb-server mariadb-client

```
root@ip-172-31-0-145:~# apt install mariadb-server mariadb-client
```

```
After this operation, 165 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.
8 [7212 B]
Get:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 mariadb-commo
n all 1:10.6.12-0ubuntu0.22.04.1 [16.4 kB]
Get:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 galera-4 amd64 26.4.9
-1build1 [720 kB]
Get:4 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libdbi-perl amd64 1.643-3
build3 [741 kB]
Get:5 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libconfig-inifiles-perl a
ll 3.000003-1 [40.5 kB]
Get:6 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libmariadb3 a
md64 1:10.6.12-0ubuntu0.22.04.1 [173 kB]
Get:7 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 mariadb-clien
t-core-10.6 amd64 1:10.6.12-0ubuntu0.22.04.1 [976 kB]
Get:8 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 mariadb-clien
t-10.6 amd64 1:10.6.12-0ubuntu0.22.04.1 [1545 kB]
Get:9 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libdaxctl1 amd64 72.1-1 [
19.8 kB]
23% [10 libndctl6 0 B/57.7 kB 0%]
```

Step 24: Now Let's secure our MariaDB database engine and disallow remote root login. For this use following command mysql_secure_installaϴon

```
root@ip-172-31-0-145:~# mysql_secure_installation
```

To skip this click enter.

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
```

Type n and enter

Then here ,this prompt is for you to change the root password to login to the database. You can opt to change it or skip if you are convinced. To skip type n

```
You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n]
You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n]
```

For safety sake, you will be prompted to remove anonymous users. Type y

```
Remove anonymous users? [Y/n] y
```

Next, disallow remote root login to prevent hackers from accessing your database. However, for testing purposes, you may want o allow log in remotely if you are configuring a virtual server. Type y.

```
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
```

Next, remove the test database. Type y.

```
Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!
```

Finally, reload privilege tables to reload database. Type y.

```
Reload privilege tables now? [Y/n] y
```

```
Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@ip-172-31-0-145:~#
```

Step 25: Install PHP

Lastly, We have to install PHP as the last component of the LAMP stack.

To install this use following command.

sudo apt install php php-mysql

```
Thanks for using MariaDB!
root@ip-172-31-0-145:~# sudo apt install php php-mysql
```

```
Do you want to continue? [Y/n] y
Get:1 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:92ubuntu
1 [12.4 kB]
Get:2 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-common amd
64 8.1.2-1ubuntu2.14 [1127 kB]
Get:3 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-opcache am
d64 8.1.2-1ubuntu2.14 [365 kB]
Get:4 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-readline a
md64 8.1.2-1ubuntu2.14 [13.6 kB]
Get:5 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-cli amd64
8.1.2-1ubuntu2.14 [1834 kB]
```

Step 26: Now move to /var/www/html folder, use following command cd /var/www/html

```
root@ip-172-31-0-145:~# cd /var/www/html
root@ip-172-31-0-145:/var/www/html#
```

Step 27: By, executing ls command following content will

displayed ls

```
root@ip-172-31-0-145:~# cd /var/www/html
root@ip-172-31-0-145:/var/www/html# ls
index.html
root@ip-172-31-0-145:/var/www/html#
```

Step 28: Add the info.php file in the step 26 directory, for this use following command sudo nano info.php

```
index.html
root@ip-172-31-0-145:/var/www/html# sudo nano info.php
```

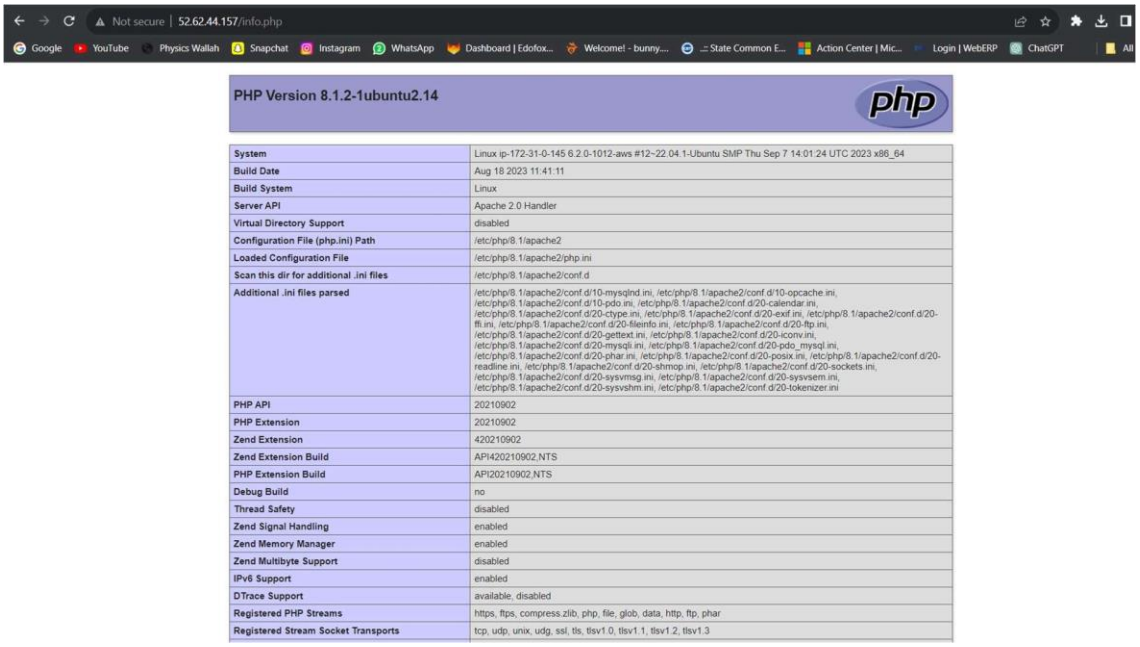
Step 29: Add the following in the info.php file and save that file.

```
<?php
Phpinfo();
?>
```

Step 30: Now open browser and append /info.php to the server's URL

Where 52.62.44.157 is your instance public IP address.

52.62.44.157/info.php



System	Linux ip-172-31-0-145 6.2.0-1012-aws #12~22.04.1-Ubuntu SMP Thu Sep 7 14:01:24 UTC 2023 x86_64
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-mysqli.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3

Step 31: Create WordPress Database.

Now it's ϴme to log in to our MariaDB database as root and create a database for accommodating our WordPress data. For this use following command. mysql -u root -p

```
root@ip-172-31-0-145:/var/www/html# mysql -u root -p
```

It will ask for password click enter only.

```
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

Step 32: Now Create database for our WordPress installation.

CREATE DATABASE wordpress_db;

```
MariaDB [(none)]> CREATE DATABASE wordpress_db;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]>
```

Step 33: Next, create database user for our WordPress setup

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

```
MariaDB [(none)]> CREATE USER 'msbecl'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]>
```

Step 34: Grant privileges to the user

Next, grant the user permissions to access the database.

GRANT ALL ON wordpress_db.* TO 'msbecl'@'localhost' IDENTIFIED BY 'password'l;

```
MariaDB [(none)]> GRANT ALL ON wordpress_db.* TO 'msbecl'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

Step 35: Great, now you can exit the database. For this use following command.

FLUSH PRIVILEGES;

EXIT;

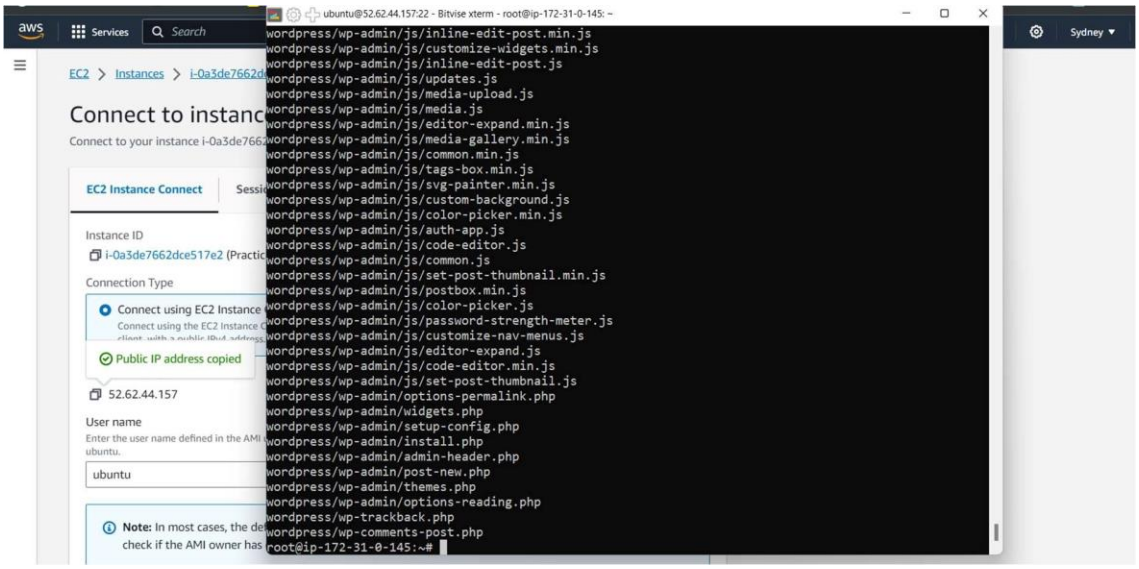
Step 36: Install WordPress CMS.

Go to your temp directory and download the latest WordPress File. For this use following command. sudo su –

then use following aŌer switching root user

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

Step 37: Then, Uncompress the tarball which will generate a folder called "wordpress". For this use following command. tar -xvf latest.tar.gz



Step 38: Copy the wordpress folder to /var/www/html/ path, for this use following command.
cp -R wordpress /var/www/html/

```
root@ip-172-31-0-145:~# cp -R wordpress /var/www/html/
root@ip-172-31-0-145:~#
```

Step 39: Run the following command to change ownership of 'wordpress' directory.
chown -R www-data:www-data /var/www/html/wordpress/

```
root@ip-172-31-0-145:~# cp -R wordpress /var/www/html/
root@ip-172-31-0-145:~# chown -R www-data:www-data /var/www/html/wordpress/
root@ip-172-31-0-145:~#
```

Step 40: Change file permissions of the WordPress folder. For this use below command.
chmod -R 755 /var/www/html/wordpres/

```
root@ip-172-31-0-145:~# chown -R www-data:www-data /var/www/html/wordpress/
root@ip-172-31-0-145:~# chmod -R 755 /var/www/html/wordpress/
root@ip-172-31-0-145:~#
```

Step 41: Create 'uploads' directory. For this use below command.
mkdir /var/www/html/worpress/wp-content/uploads

```
root@ip-172-31-0-145:~# mkdir /var/www/html/wordpress/wp-content/uploads
```

Step 42: Finally, change the permissions of 'uploads' directory.
Chown -R www-data:www-data /var/www/html/wordpress/wp-content/uploads/

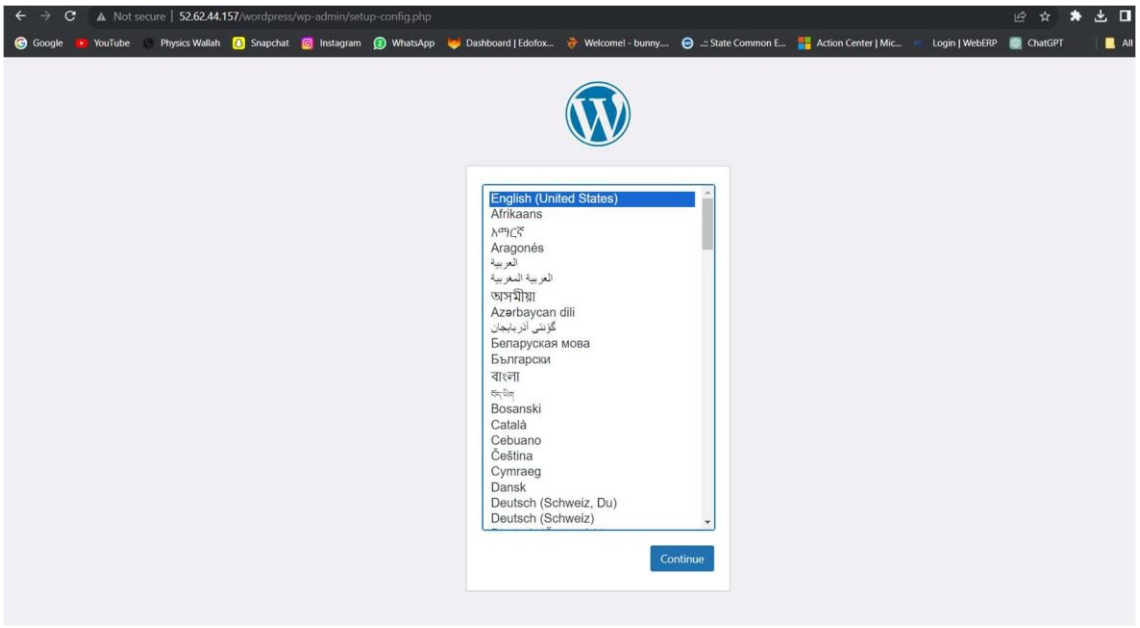
Now, change file permissions of uploads directory chmod -R
755 /var/www/html/wordpress/wp-content/uploads/

```
root@ip-172-31-0-145:~# chown -R www-data:www-data /var/www/html/wordpress/wp-content/uploads/
root@ip-172-31-0-145:~# chmod -R 755 /var/www/html/wordpress/wp-content/uploads/
root@ip-172-31-0-145:~#
```

Step 43: Now, Open browser and go to server's URL In my case it is

52.62.44.157/wordpress (Public IP address of your instance/wordpress)

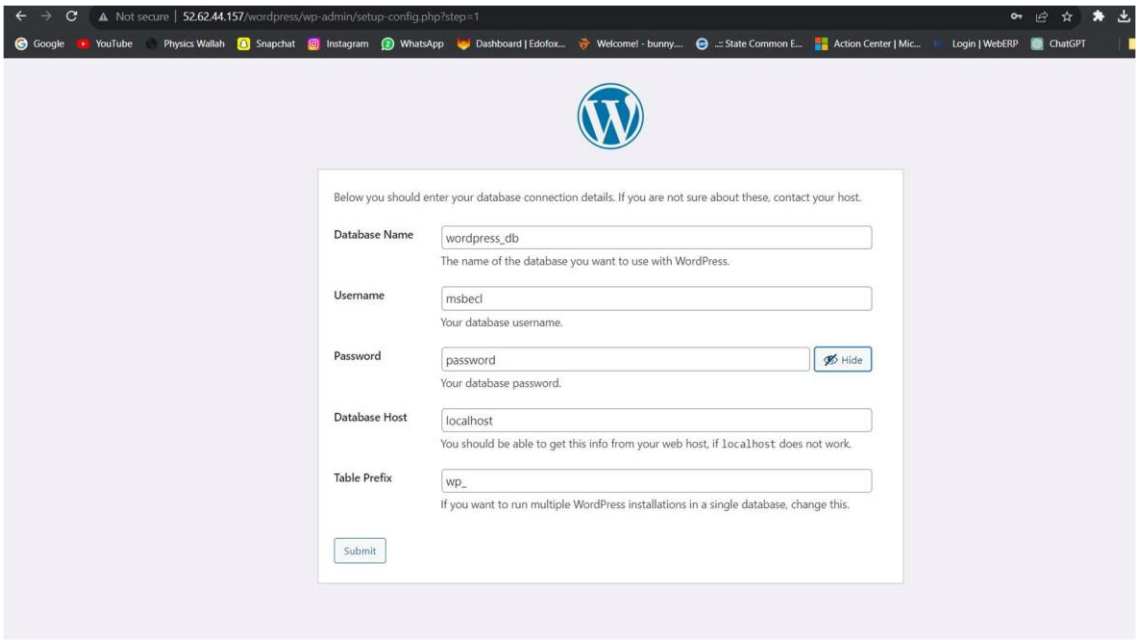
Click on conϴnue



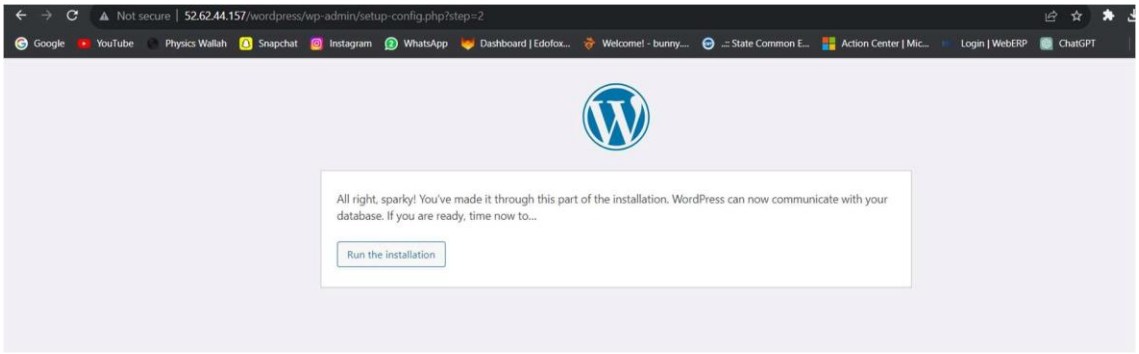
Step 44: You'll be presented with a WordPress wizard and a list of credentials required to successfully set it up.

Click on Let's go.

Step 45: Fill out the form as shown with the credentials specified whwn creating the WordPress database in the MariaDB database.(use all informaϴon from step 32 to step 34) Leave out database host and table prefix an click on submit.

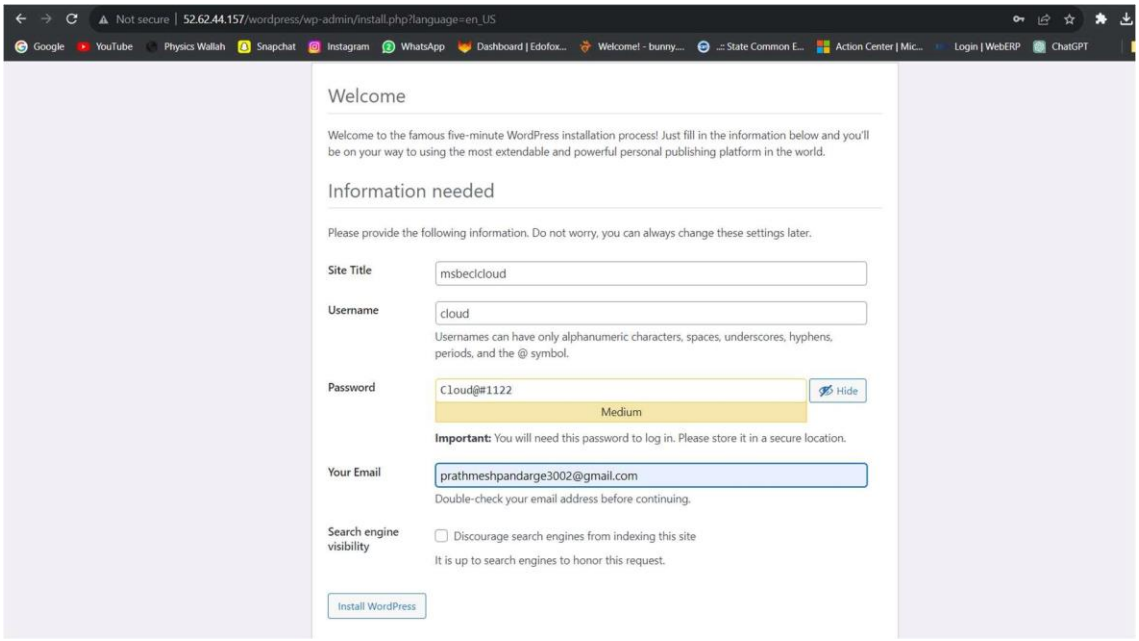


Step 46: If all the details are correct, you will be redirecting like these. Click on Run the installaϴon



Step 47: Fill out additional details required such as site title, username and password as shown. Save them somewhere safe for future purpose. Ensure to use strong password.

And then Click on install WordPress.



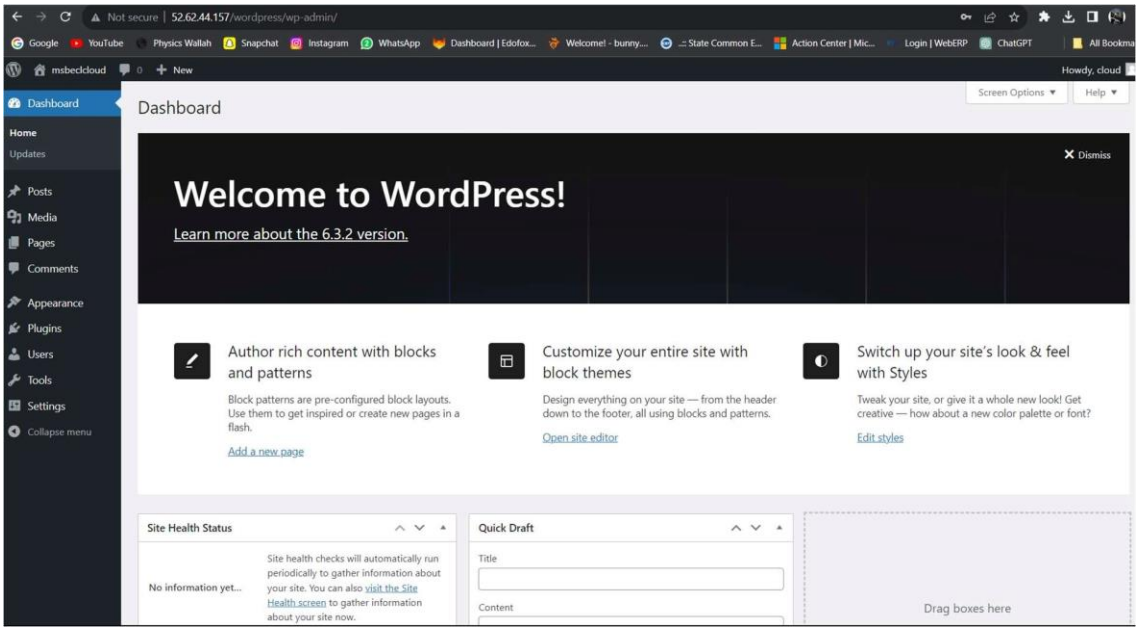
Step 48: If all went well, then you will get “Success!” notification as shown.

Then click on Log in.

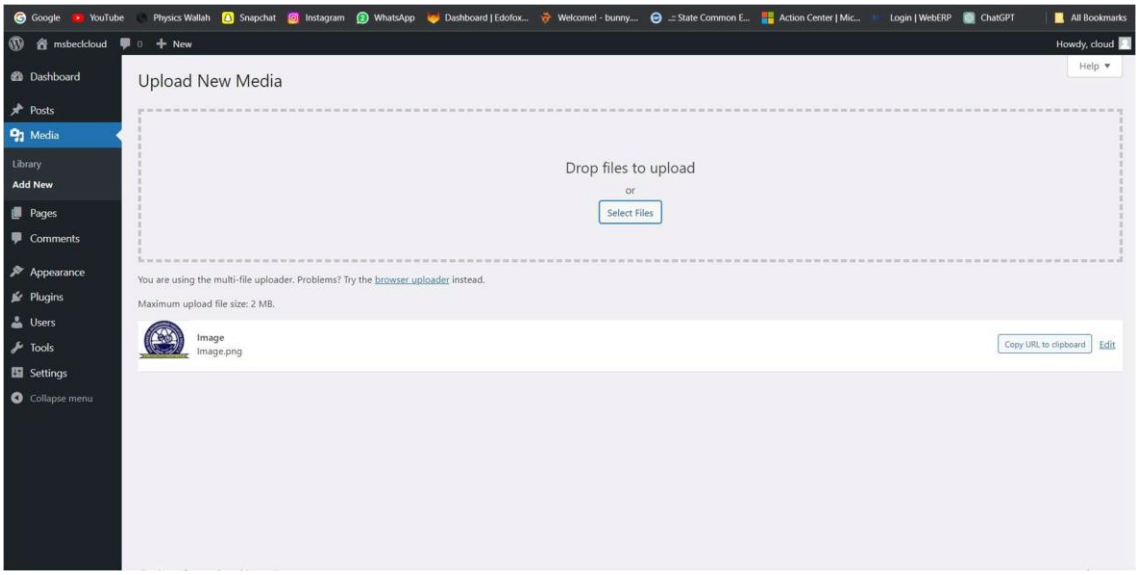
Step 49: Input all the credentials from step 45. Username as cloud and use your password and click on Log in.

Step 50: It will enter to the WordPress Dashboard

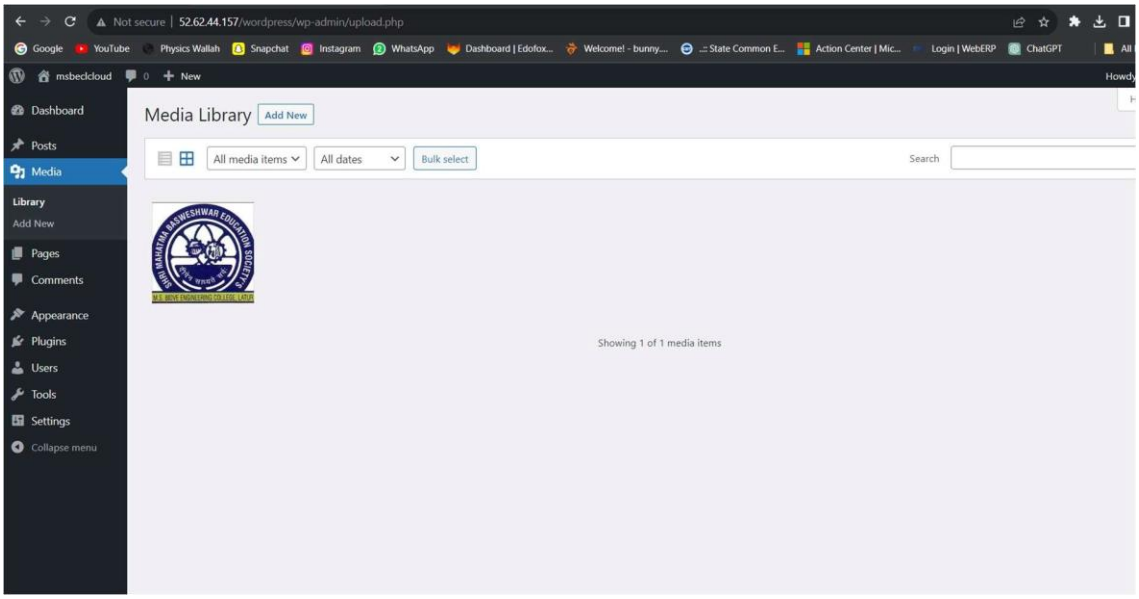
Now, Click on media tab option which is on left side.



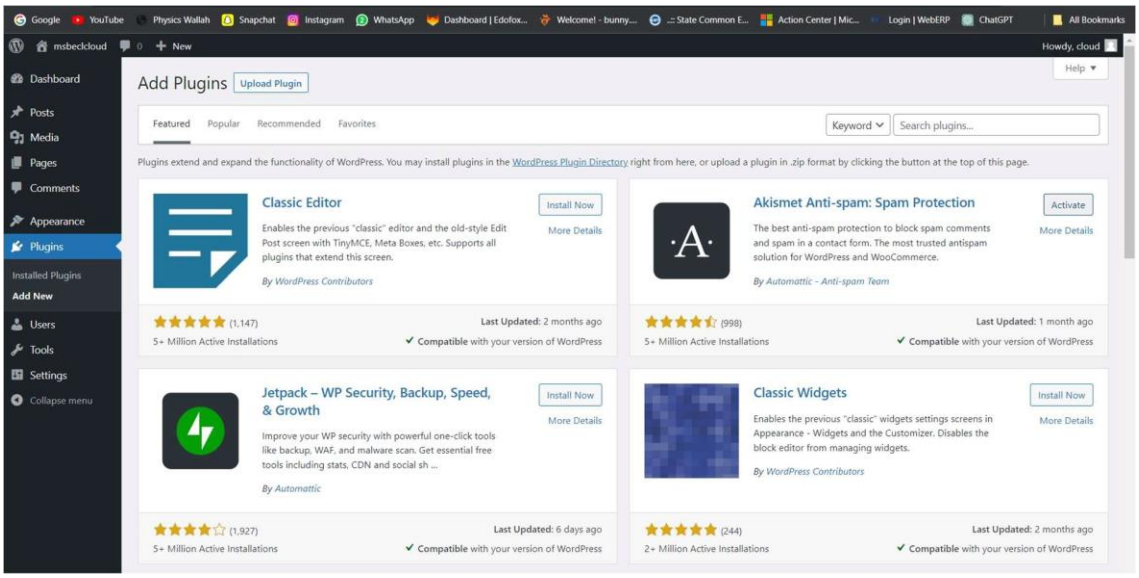
Step 51: Upload some files to save by clicking on select files.



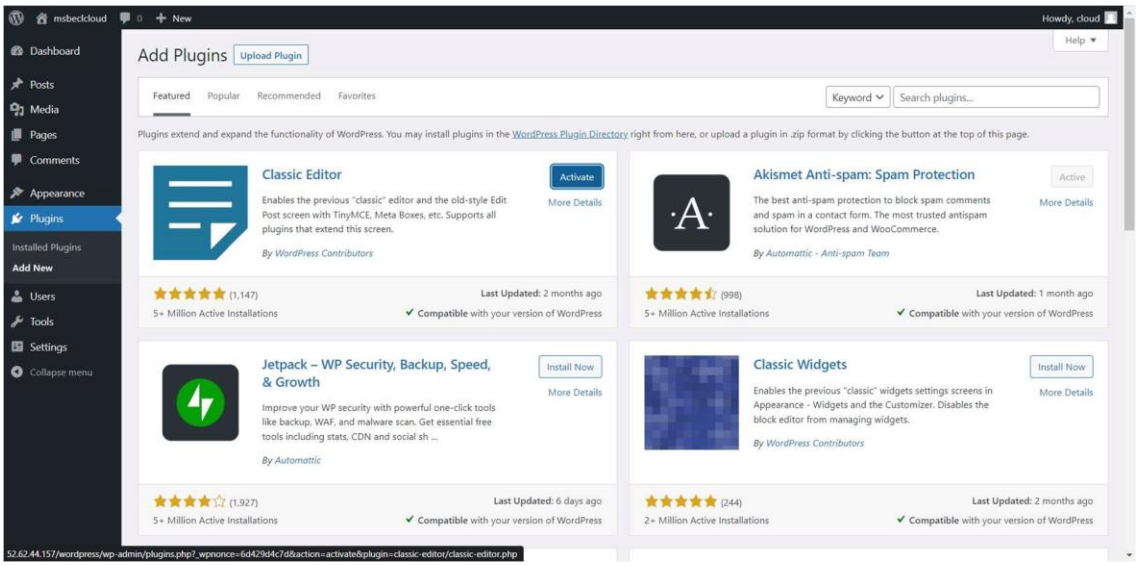
Now you can see that image successfully stores in it.



Step 52: Now, Go to Plugins to add/install Classic Editor then click on install now

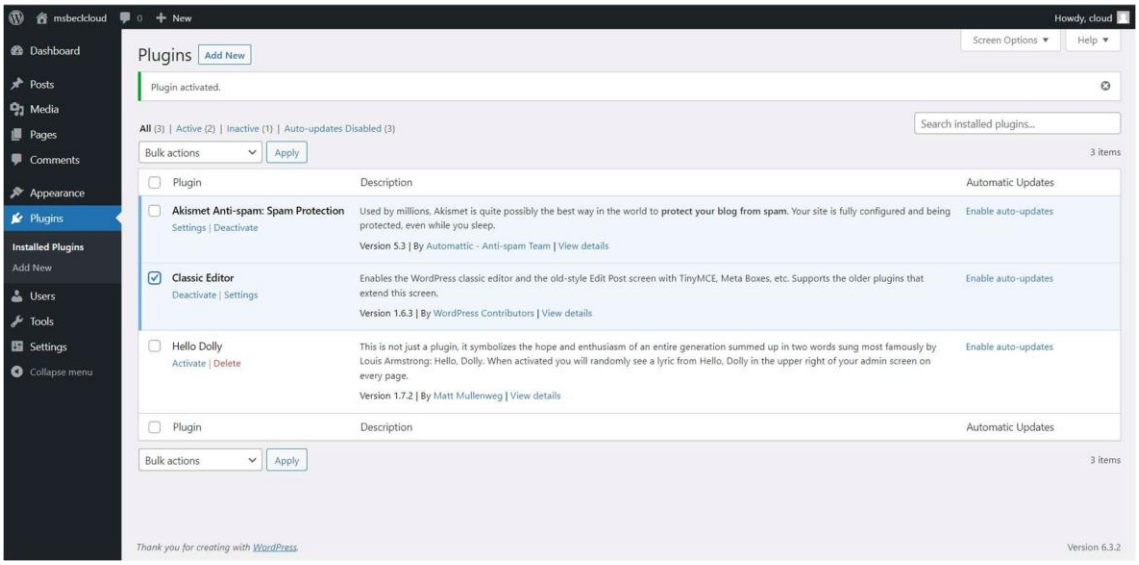


Step 53: After Installing click on Activate. It will activate the plugin we have installed.



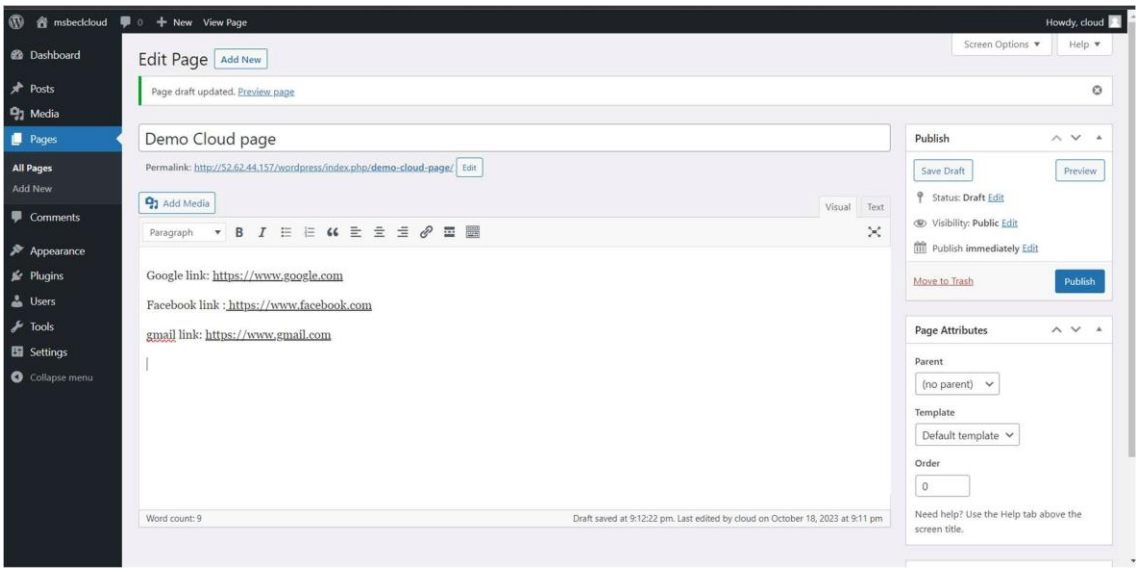
Step 54: Now select Classic Editor and click on apply.

Then plugin is now added successfully.

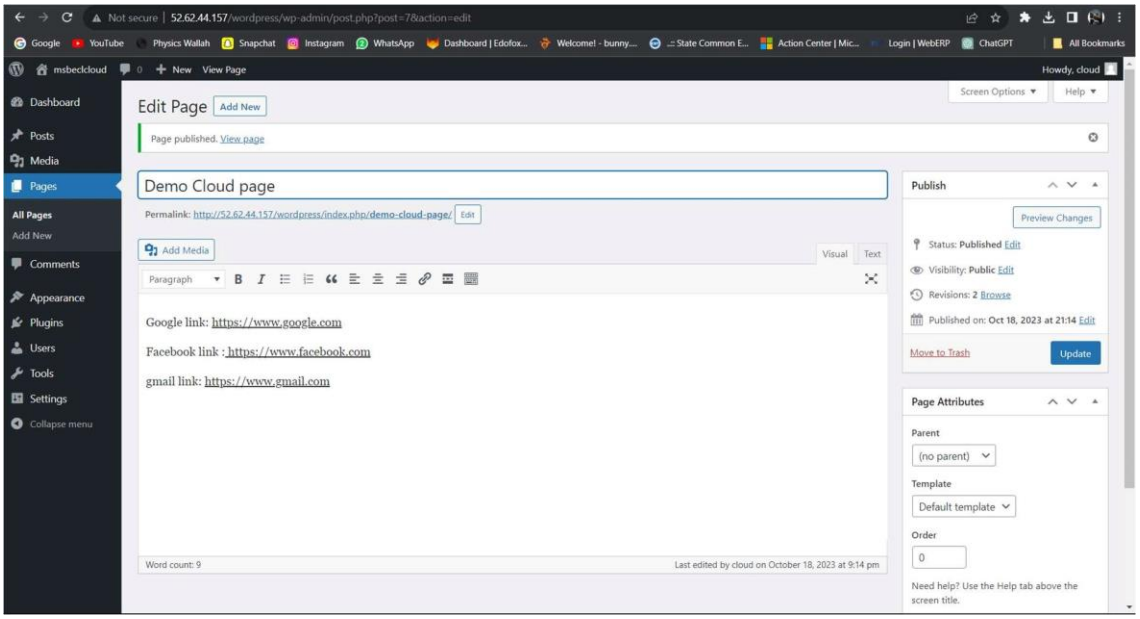


Step 55: To create Social media account reference, use pages opϴon from left side and click on it. Then create a hyperlink of the social media accound that you want to use it.

And then to create this page click on publish.

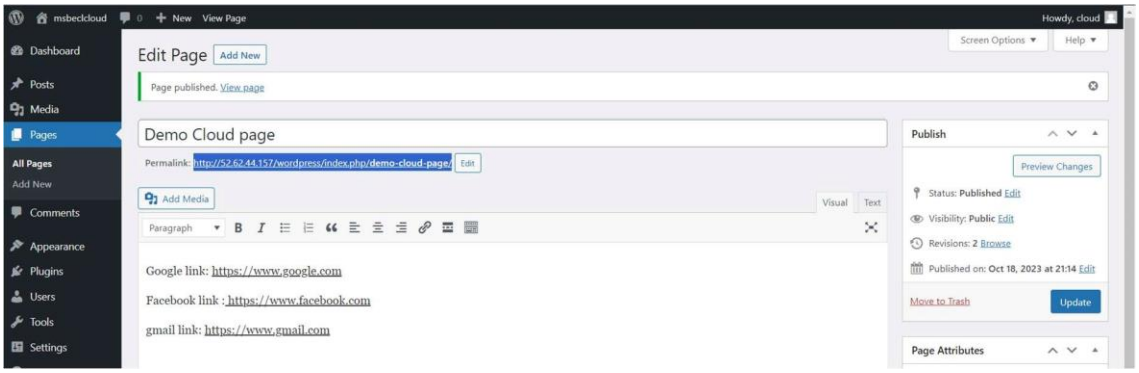


Step 56: To update pages (if requires) the click on update.

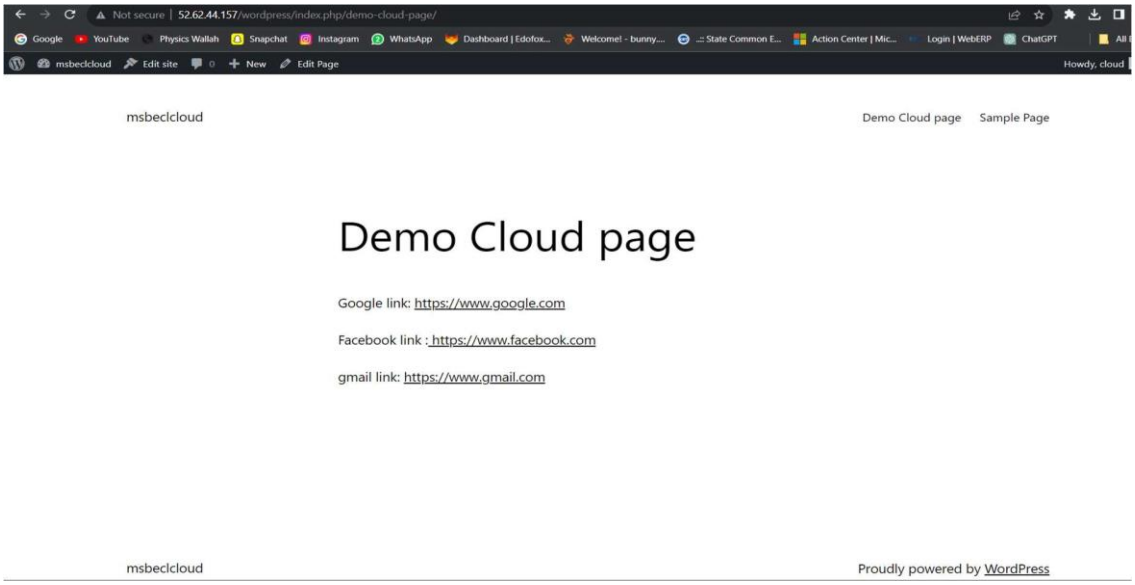


Step 57: Now you can access this page by using permalink, for this

Copy the Permalink as shown and paste to the browser.



Step 58: Now you can access your own WordPress page.



Now You can proceed to discover the various features, plugins, and themes and proceed for setng up your first blog/website.

Conclusion : Hence in this practical We have successfully created a scenario in WordPress for social marketing, Search engine and Sharing tools.