

Experiment No:7

Aim: Create a scenario in wordpress for Social Marketing. Search engine and Sharing Tools.

Practical Requirement:

AWS (EC2)

Bitvise SSH Server (s/w)

Introduction:

Why wordpress so popular?

So why is WordPress this popular? Let's briefly look into some of the factors that have led to the immense success of the platform.

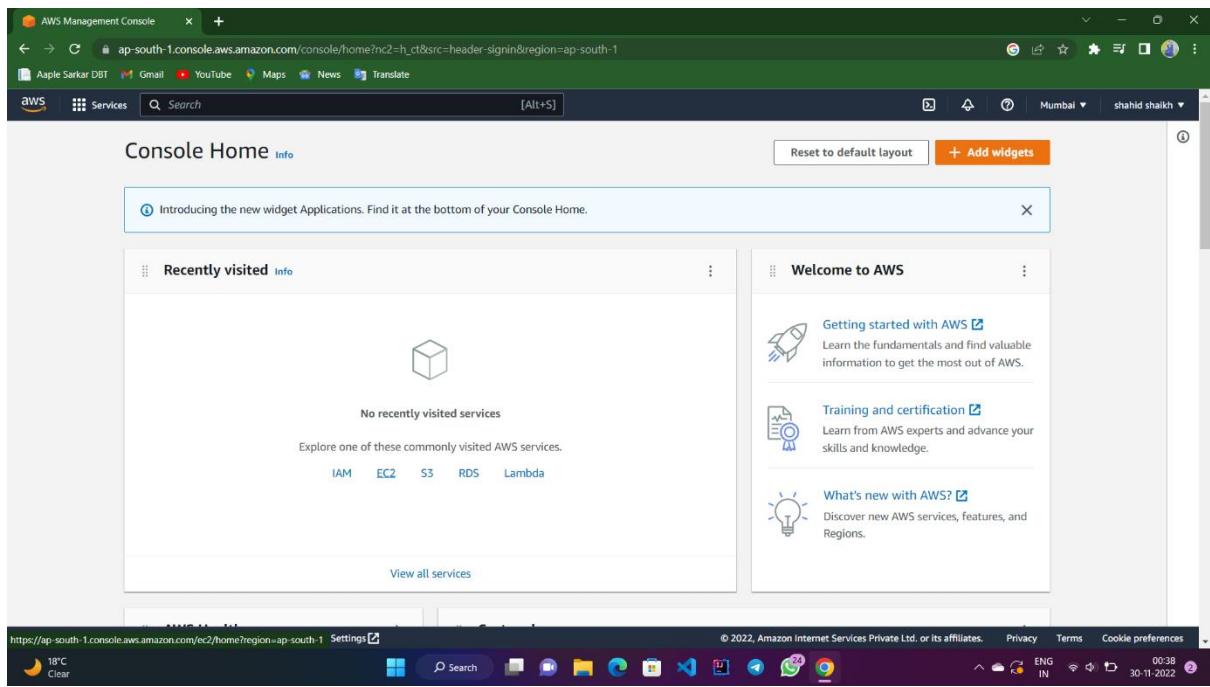
WordPress sites are Responsive

WordPress platform is inherently responsive and you do not have to stay awake worrying about your sites being able to fit across multiple devices. This benefit also adds to your site being ranked higher in Google's SEO score!

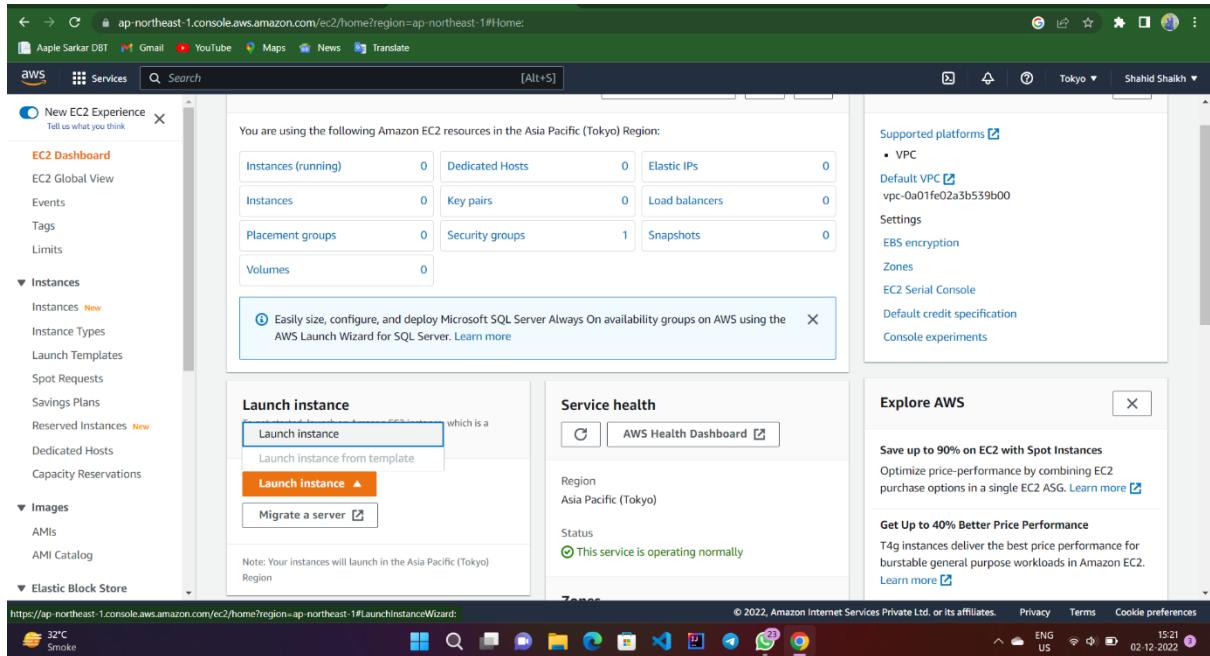
WordPress is SEO ready

WordPress is built using well-structured, clean and consistent code. This makes your blog/site easily indexable by Google and other search engines thereby making your site rank higher. In addition, you can decide which pages rank higher or alternatively use SEO plugins like the popular Yoast plugin which enhances your site's ranking on Google.

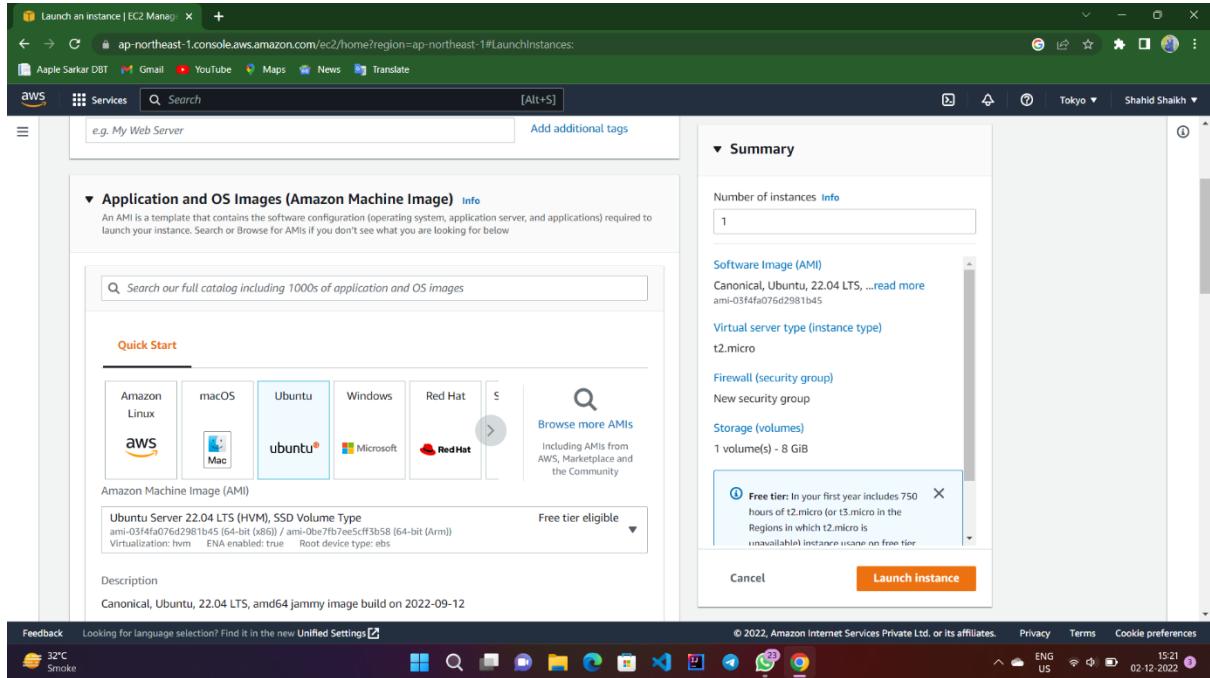
Step 1: Open the AWS Management Console and click on the EC2 service of it.



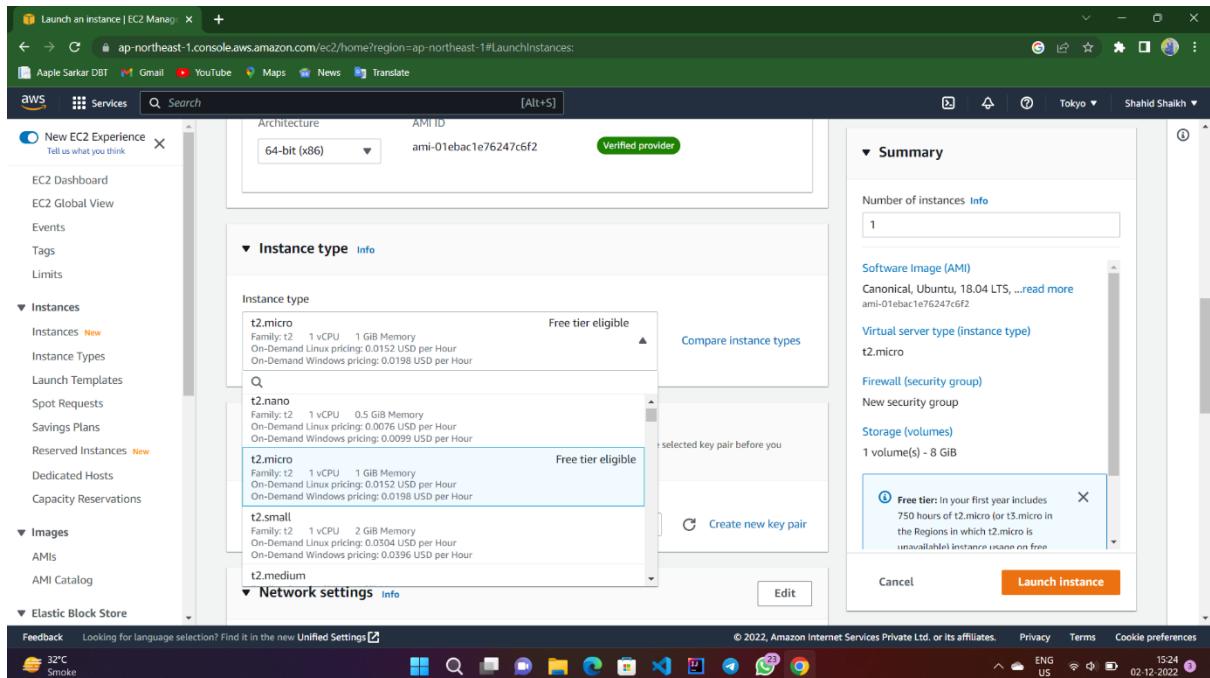
Step 2: Click on the Launch instance button to create a new instance.



Step 3: select the ubuntu server 18.04 LTS (HVM) to install ubuntu on instance.



4: choose the default instance type only available there that is free trial one.



Step 5: Do not change any setting click over the add storage further.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

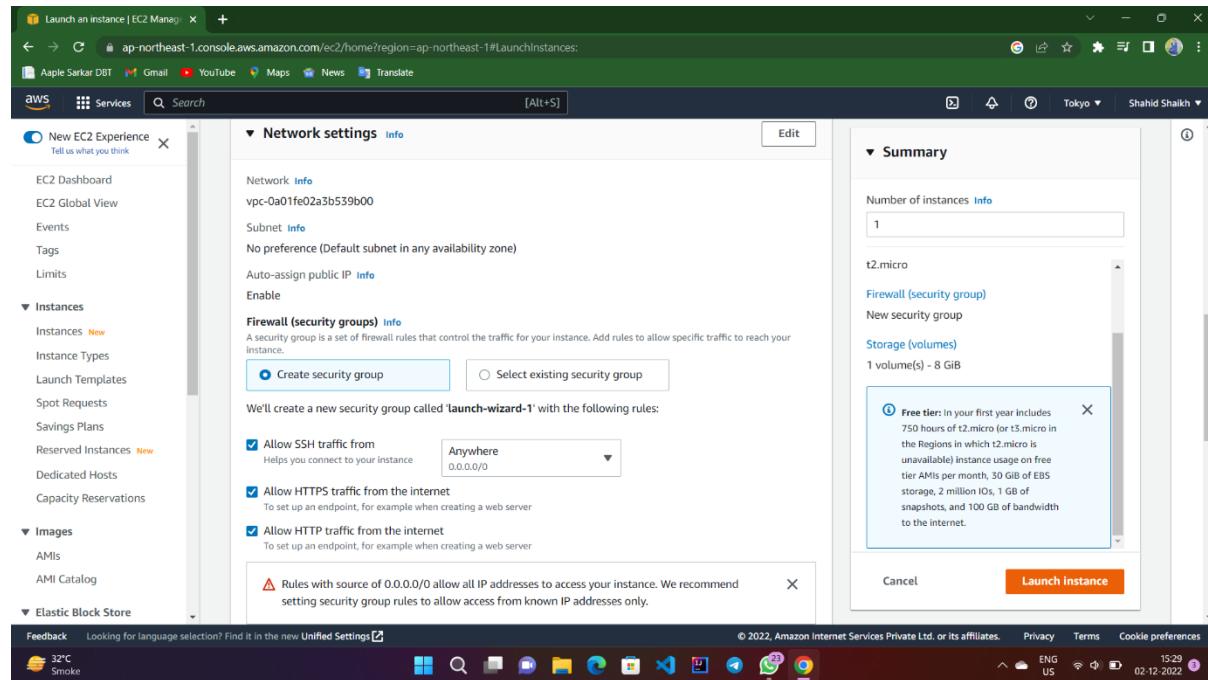
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1
Purchasing option	<input type="checkbox"/> Request Spot Instances
Network	Launch into EC2-Classic <input type="button" value="Create new VPC"/>
Availability Zone	No preference
IAM role	None
Shutdown behavior	Stop <input type="checkbox"/>
Enable termination protection	<input type="checkbox"/> Protect against accidental termination
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small>

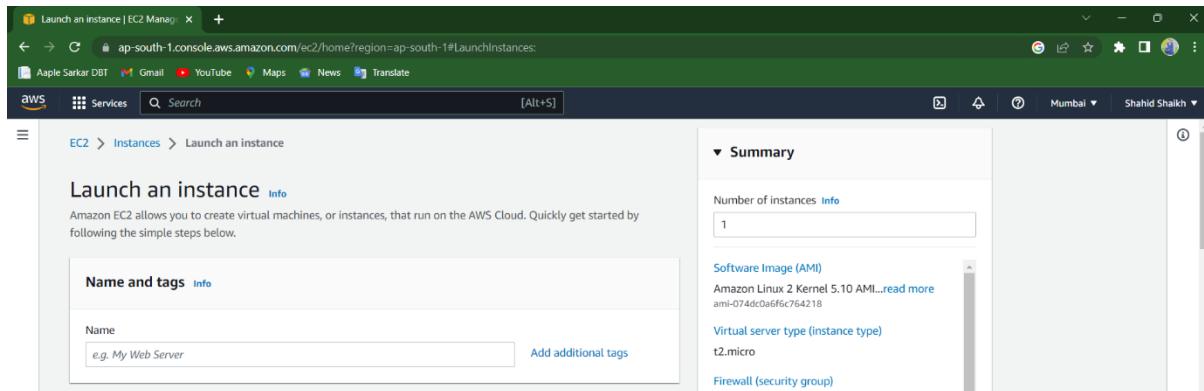
Advanced Details

Cancel Previous **Review and Launch** Next: Add Storage

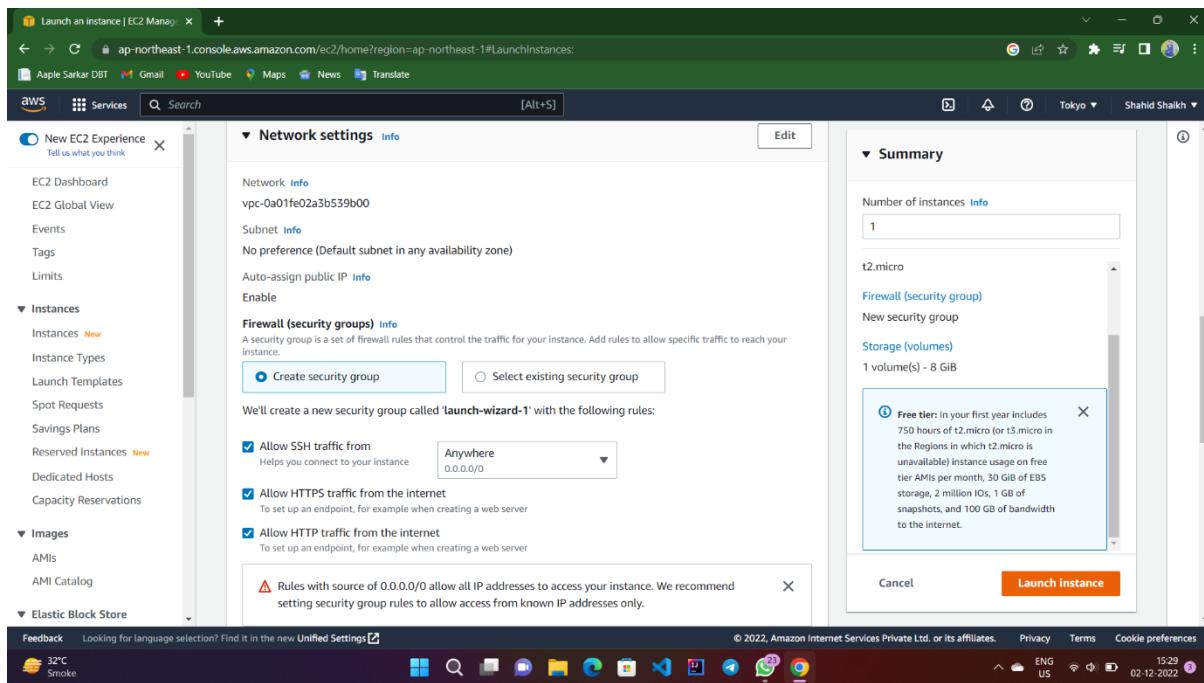
Step 6: At this step if you want then you can change the size of the storage at size(GB) section. But for the hadoop installation the 8 GB space is more than sufficient so don't make change on it and click on next: Add Tags:



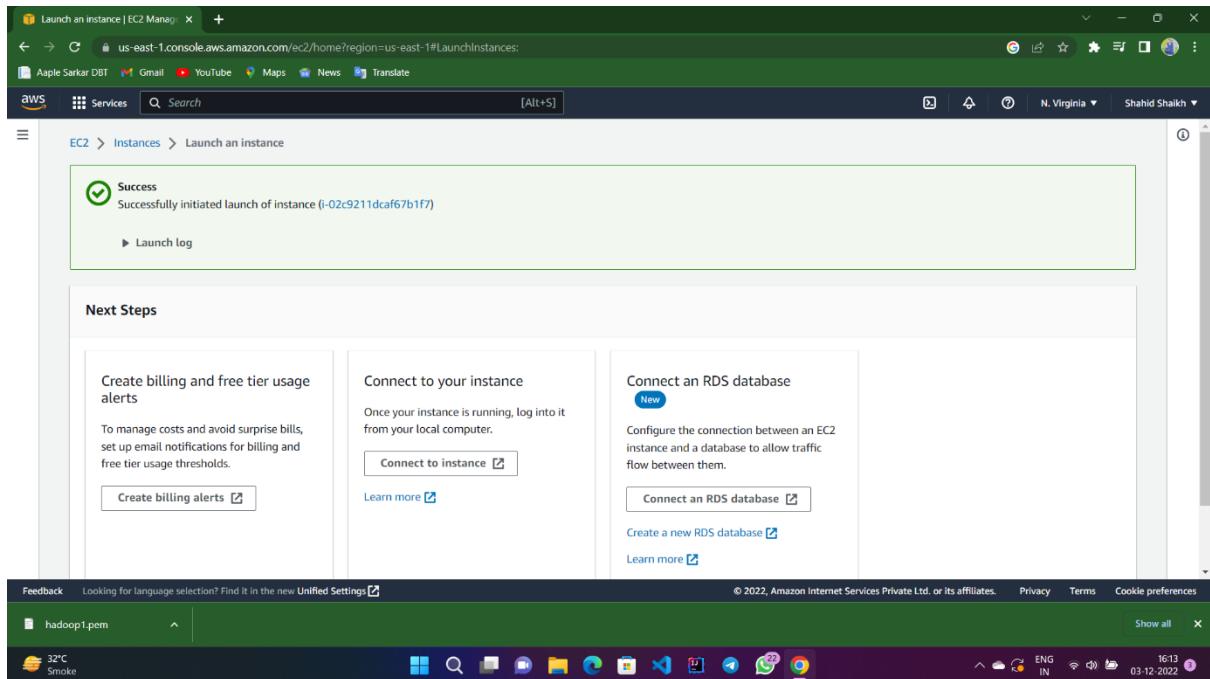
Step 7: click on the next configure security group.



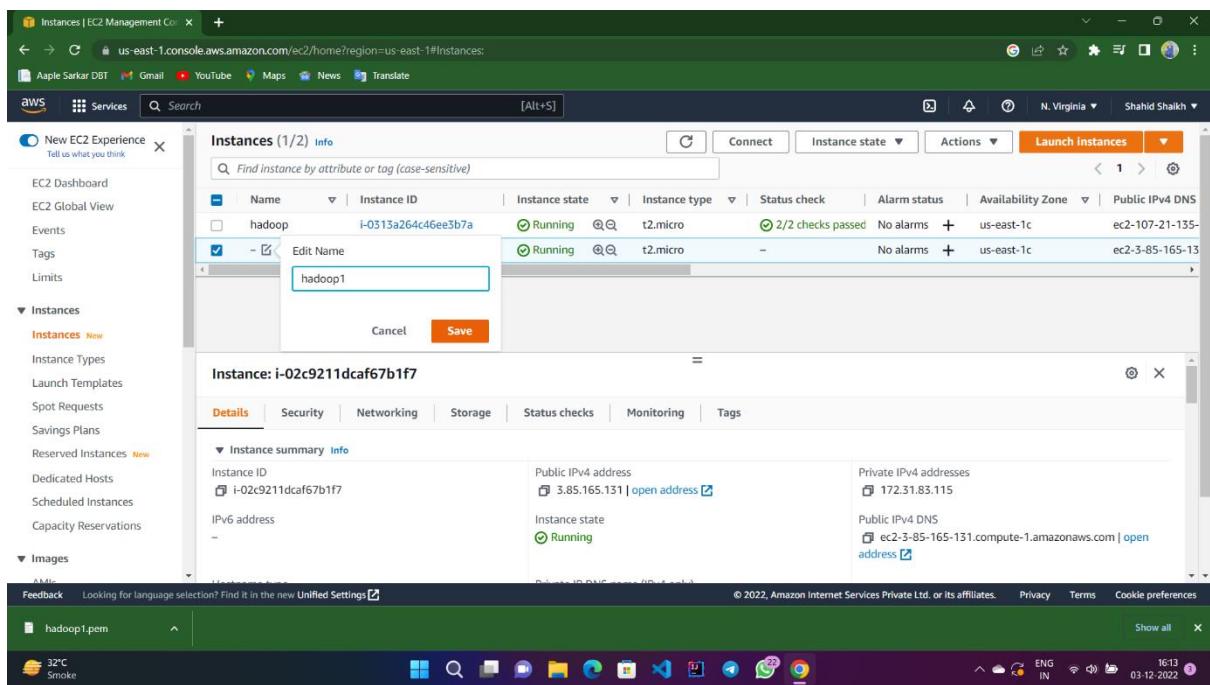
Step 8: At configre security group section add the rules for website accesing.
For hadoop we required the SSH, SMTP, ALL traffic, SMTP, ALL UDP, ALL TCP, HTTPS, HTTP, RDP, MySQL/Aurora.



Step 9: Then at review and launch page click on Launch button.



Step 10: Then create a new key pair. Ex: hadoop1 & and then download it.



Step 11: Then at launch status click on the view instance button.

The screenshot shows the AWS EC2 'Launch an instance' page. At the top, there is a green success message box stating 'Successfully initiated launch of instance (i-02c9211dcf67b1f7)'. Below this, there is a link to 'Launch log'. Under the heading 'Next Steps', there are three cards: 'Create billing and free tier usage alerts', 'Connect to your instance', and 'Connect an RDS database'. Each card has a 'Create' or 'Learn more' button. The bottom of the screen shows the Windows taskbar with various icons and the system tray indicating the date and time.

Step 12: Then at this page your instance will be created and give the name to it to identify better be in future for Ex: hadoop1.

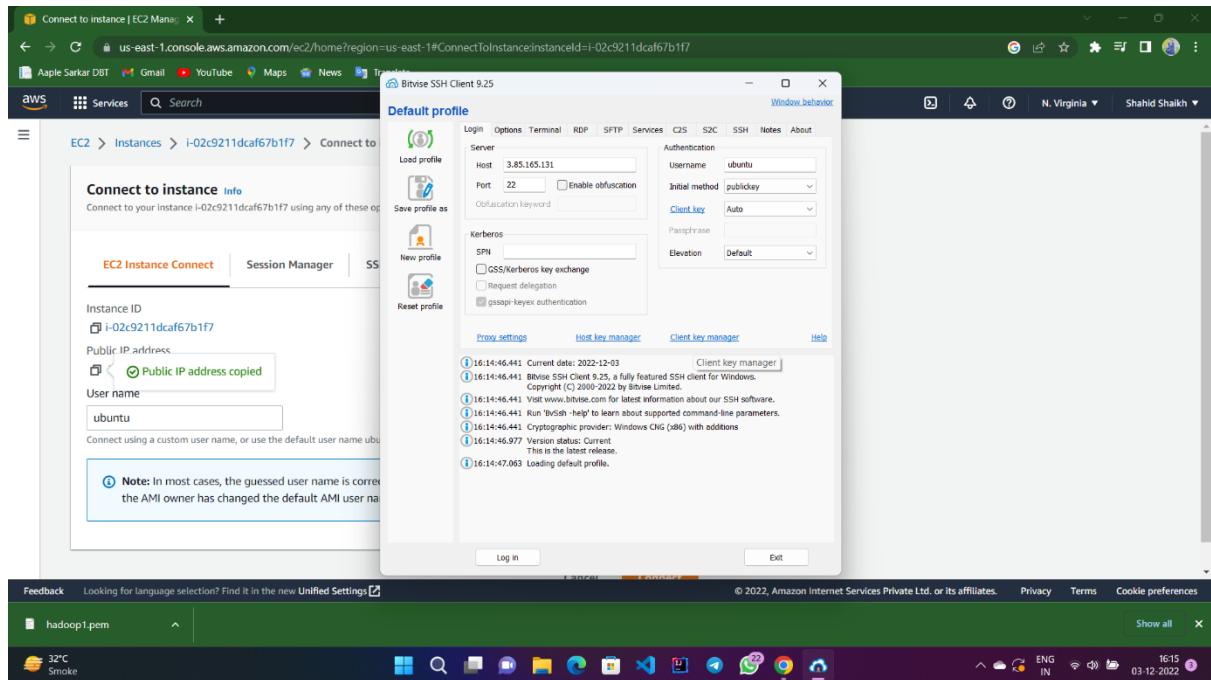
The screenshot shows the AWS EC2 'Instances' page. On the left, there is a sidebar with navigation options like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', and 'Limits'. Under 'Instances', there are sections for 'Instances' (with a 'New' button), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', and 'Images'. The main area displays a table of instances. One instance is selected, named 'hadoop1' with the ID 'i-02c9211dcf67b1f7'. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. Below the table, a detailed view for 'Instance: i-02c9211dcf67b1f7' is shown, with tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The 'Details' tab is active, showing information such as Instance ID, Public IPv4 address (3.85.165.131), Private IPv4 addresses (172.31.83.115), and Instance state (Running). The bottom of the screen shows the Windows taskbar with various icons and the system tray indicating the date and time.

Step 13: Then click on the connect button option at top to see your instance public IP and instance information.

Step 14: Then open then Bitvise SSH client 8.45 and copy the public IP address of your instance from the EC2 instance connect section ex: 3.45.245.231 ex and paste it on the Bitvise Host text area column as shows in 55.

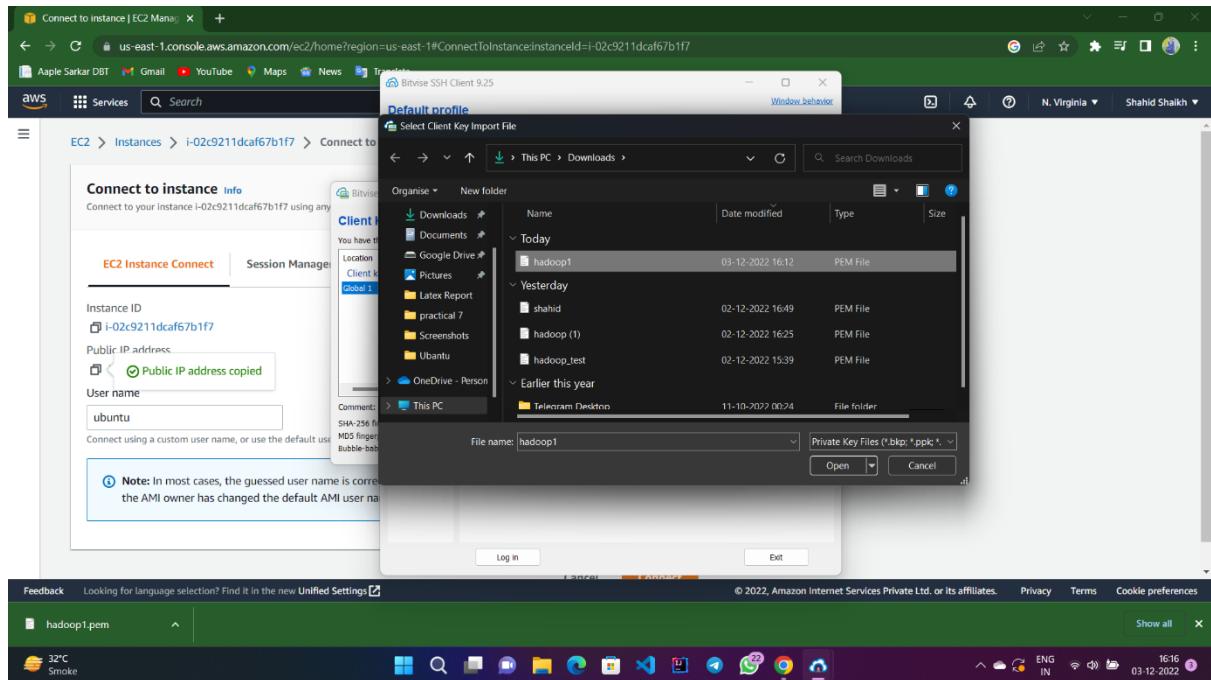
**Then fill the username text field as ubuntu, at initial method option select the public key option and at client key select the client key as auto only."

Step 15: Then click on the client key manager link.

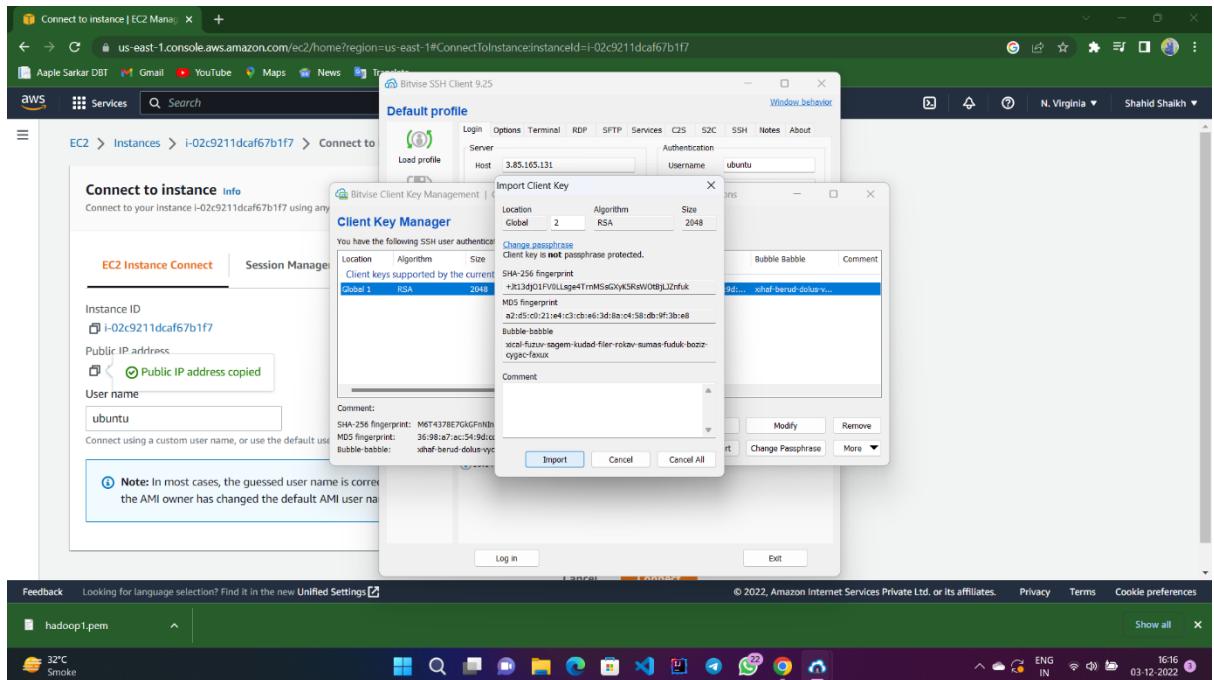


Step 16: Open the .pem file that you have created at the step number 10.
ex:hadoop1.pem

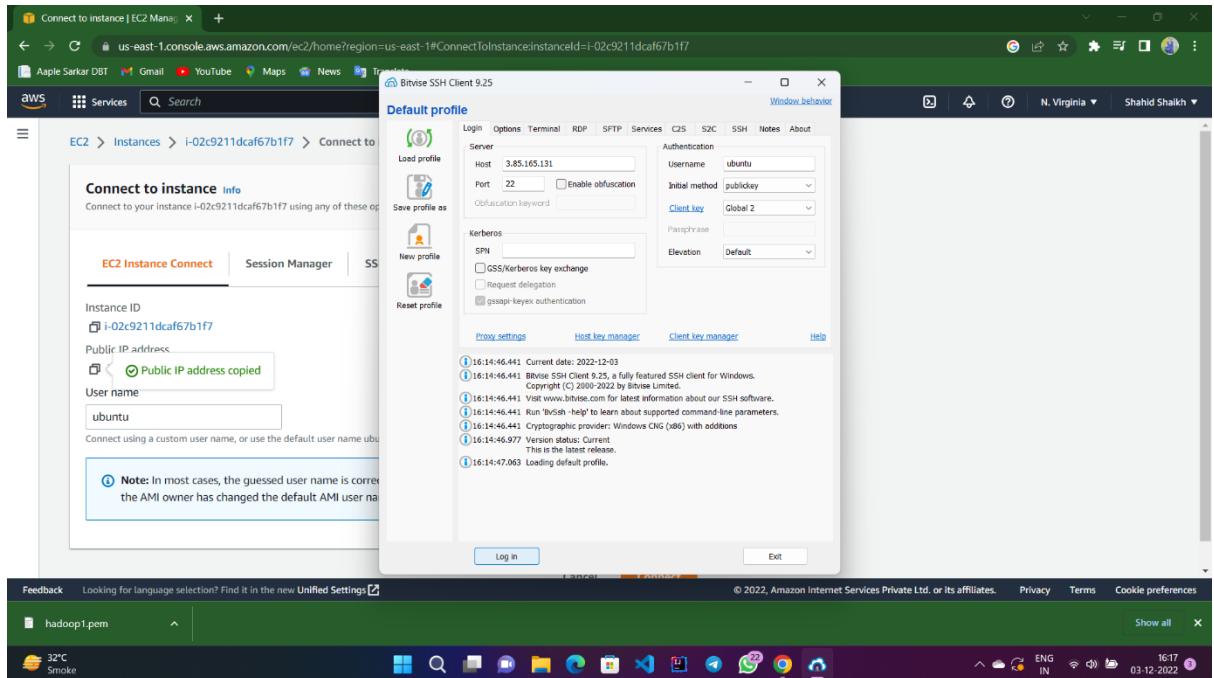
"If you not find your .pem file at the saved location the click on the ALL files searching option as all files or simply put the name of your .pem file from saved location and open it."



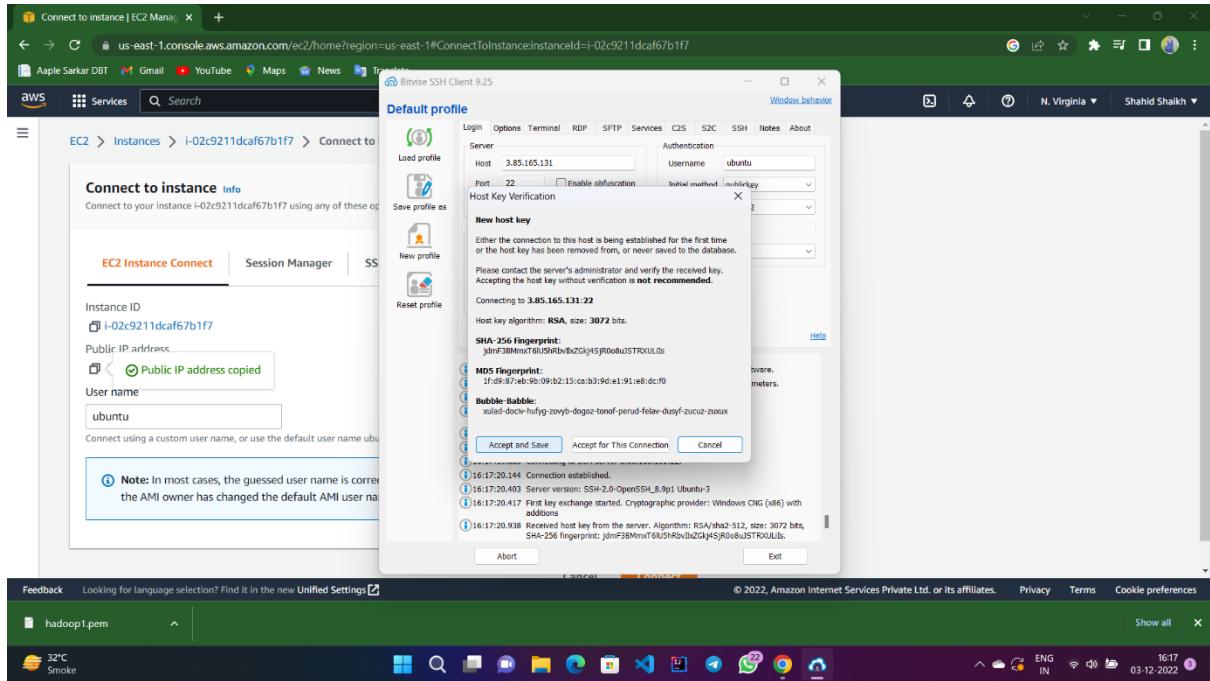
Step 17: Then at next notice click on the import button option futher and it will assign the number to public key ex: Global 2



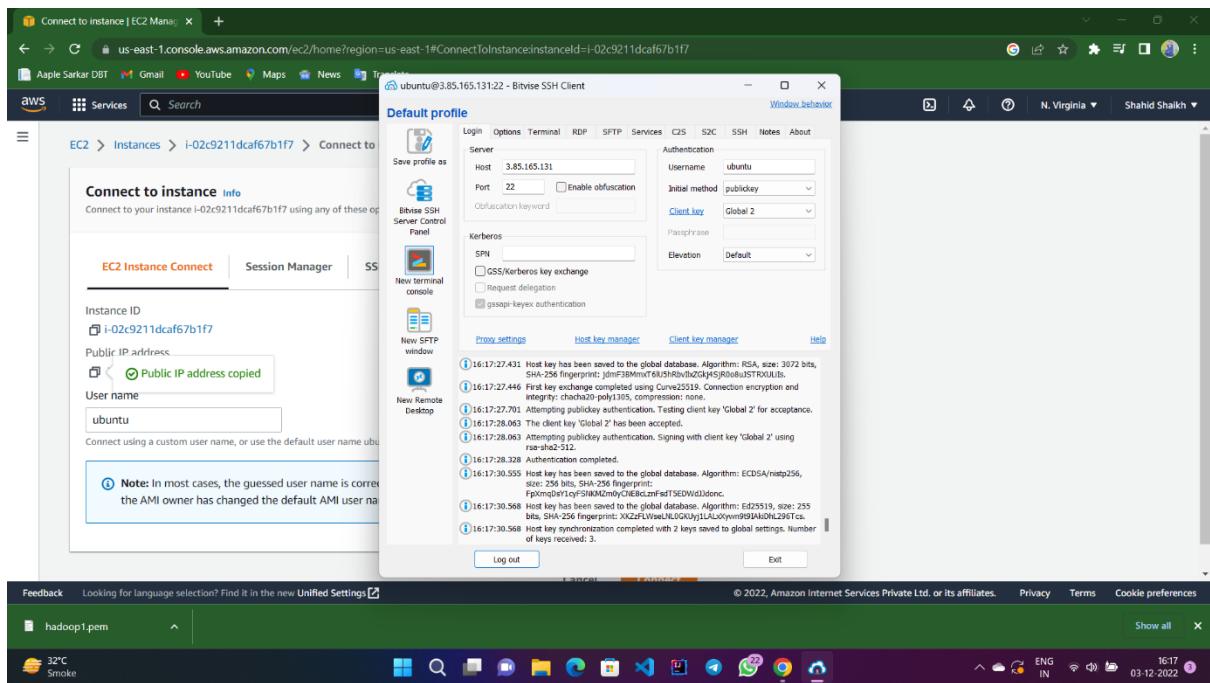
Step 18: Now At this step select the client key as Global 2 and the click on the LOG in Button.



Step 19: After click on the login button at the Host key verification window click on the Accept and Save button.



Step 20: Click on the new terminal window option to open the ubuntu OS.



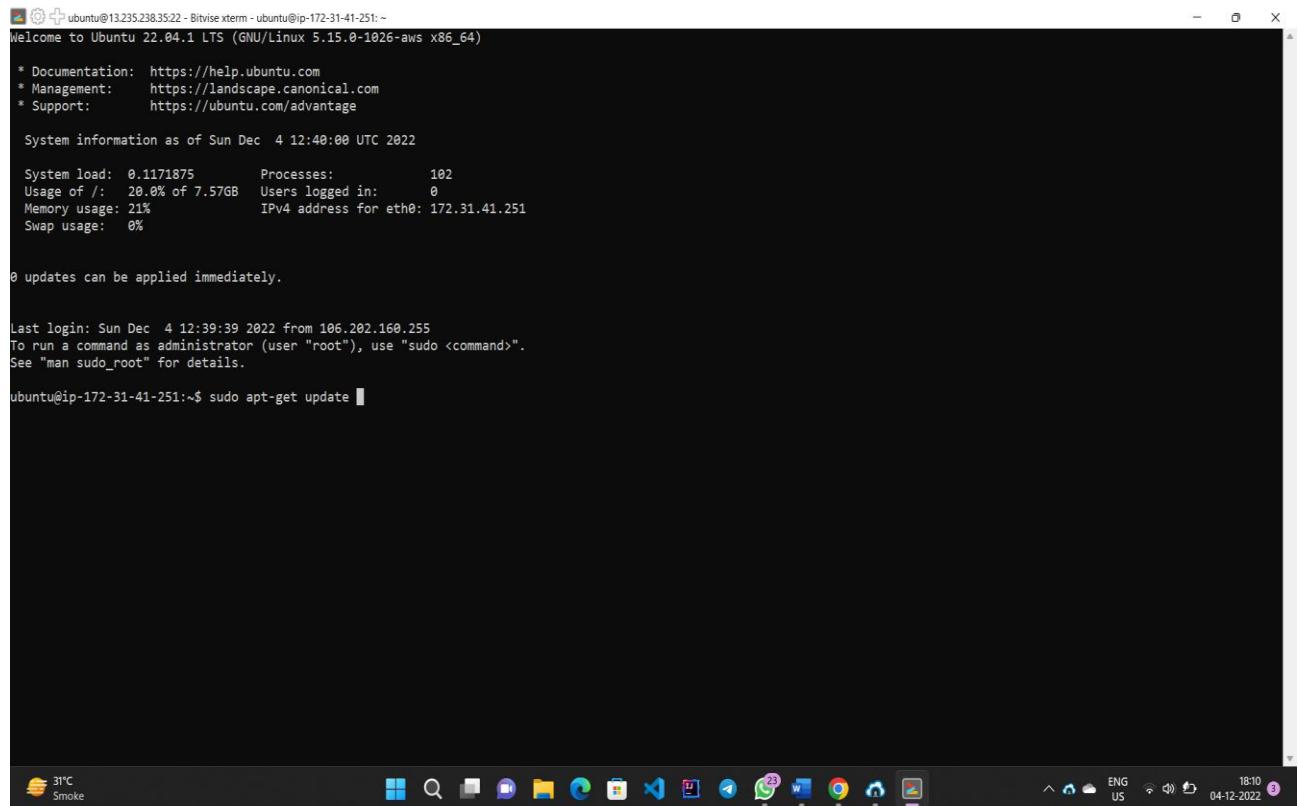
Install WordPress on Ubuntu 18.04

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

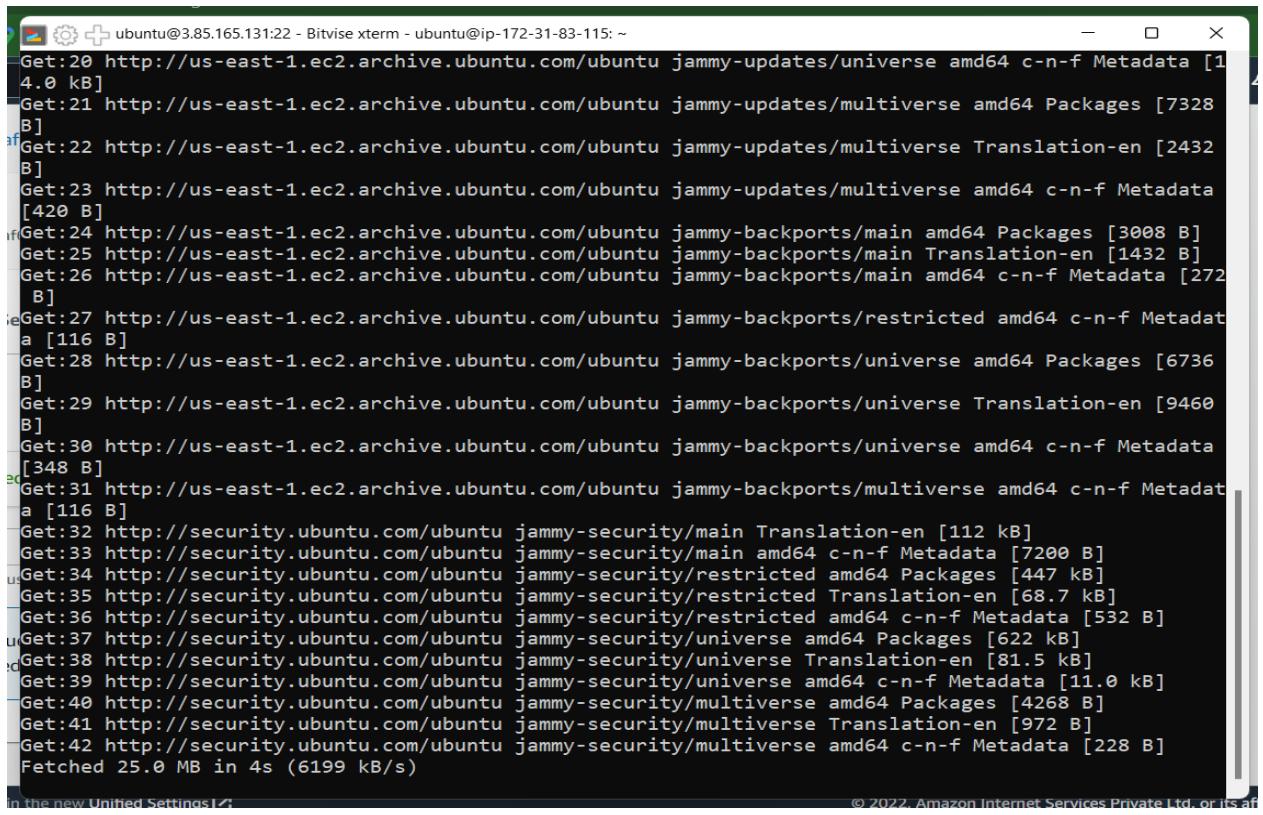
Step 21: update System:

Use command:

```
Sudo apt-get update
```



```
ubuntu@13.235.238.35:22 - Bitvise xterm - ubuntu@ip-172-31-41-251:~  
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
System information as of Sun Dec 4 12:40:00 UTC 2022  
  
System load: 0.1171875 Processes: 102  
Usage of /: 20.0% of 7.57GB Users logged in: 0  
Memory usage: 21% IPv4 address for eth0: 172.31.41.251  
Swap usage: 0%  
  
0 updates can be applied immediately.  
  
Last login: Sun Dec 4 12:39:39 2022 from 106.202.160.255  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-172-31-41-251:~$ sudo apt-get update
```



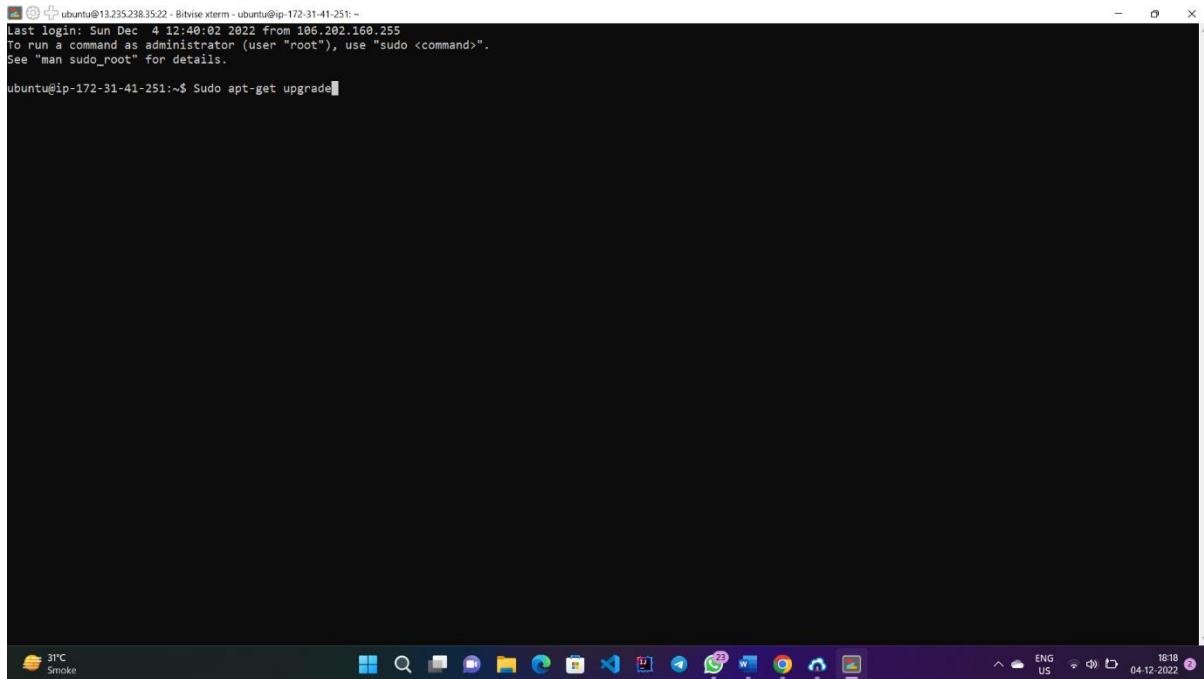
```
ubuntu@3.85.165.131:22 - Bitvise xterm - ubuntu@ip-172-31-83-115: ~
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [1
4.0 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [7328
B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [2432
B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata
[420 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3008 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [1432 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [272
B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadat
a [116 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [6736
B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [9460
B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata
[348 B]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadat
a [116 B]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [112 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [7200 B]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [447 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [68.7 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [532 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [622 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [81.5 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [11.0 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4268 B]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [972 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [228 B]
Fetched 25.0 MB in 4s (6199 kB/s)

© 2022, Amazon Internet Services Private Ltd. or its af
```

Step 22: Upgrade system:

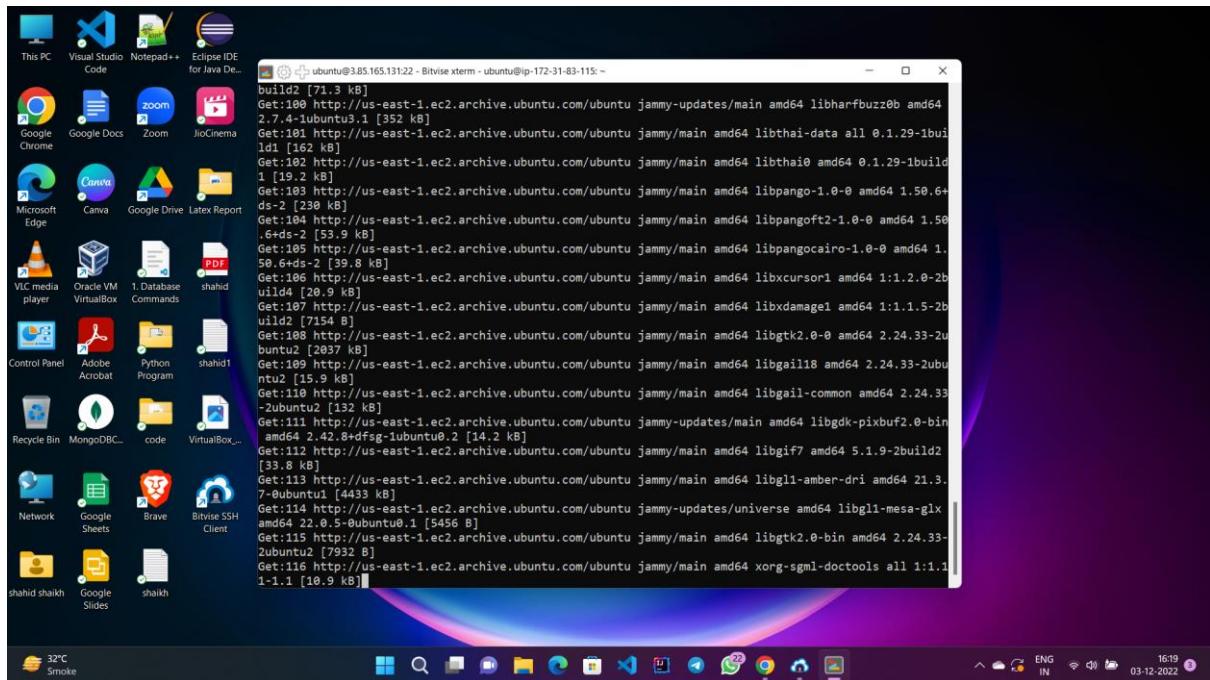
To upgrade system use following command:

Sudo apt-get upgrade



```
ubuntu@13.235.238.35:22 - Bitvise xterm - ubuntu@ip-172-31-41-251: ~
Last login: Sun Dec  4 12:40:02 2022 from 106.202.168.255
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

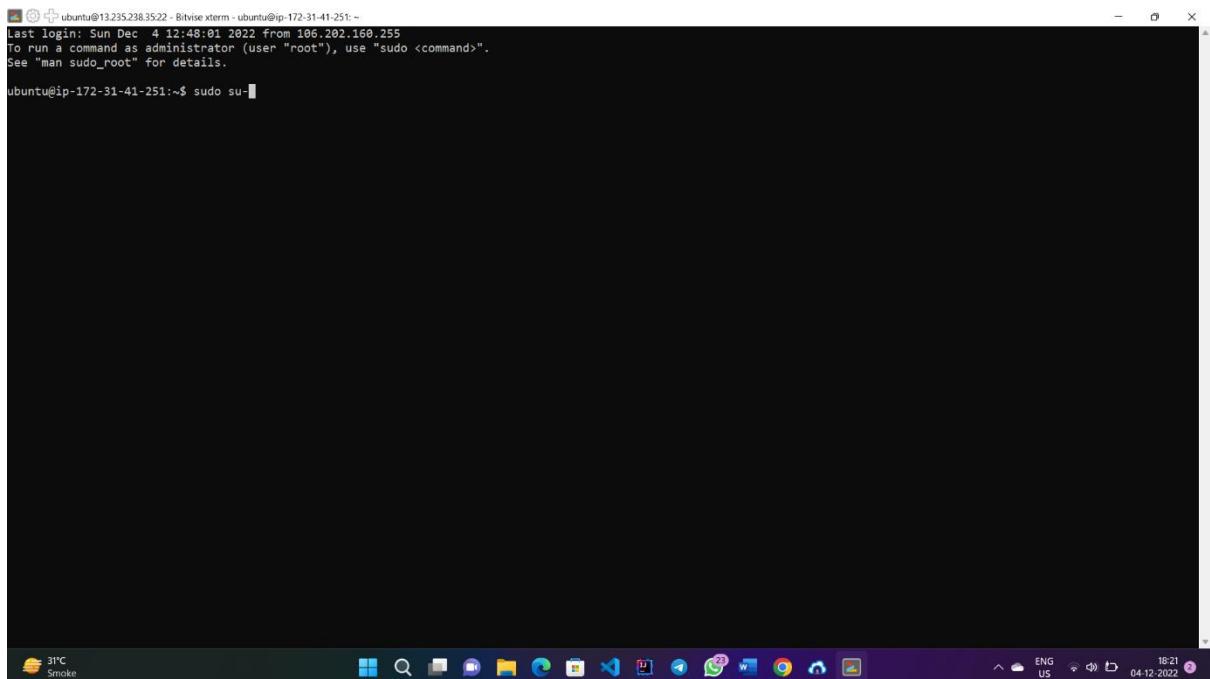
ubuntu@ip-172-31-41-251:~$ Sudo apt-get upgrade
```



Step 23: Install Apache

To install apache in system you need you have to switch to the root user so for that use the command:

`sudo su-`



Now for installation use command:

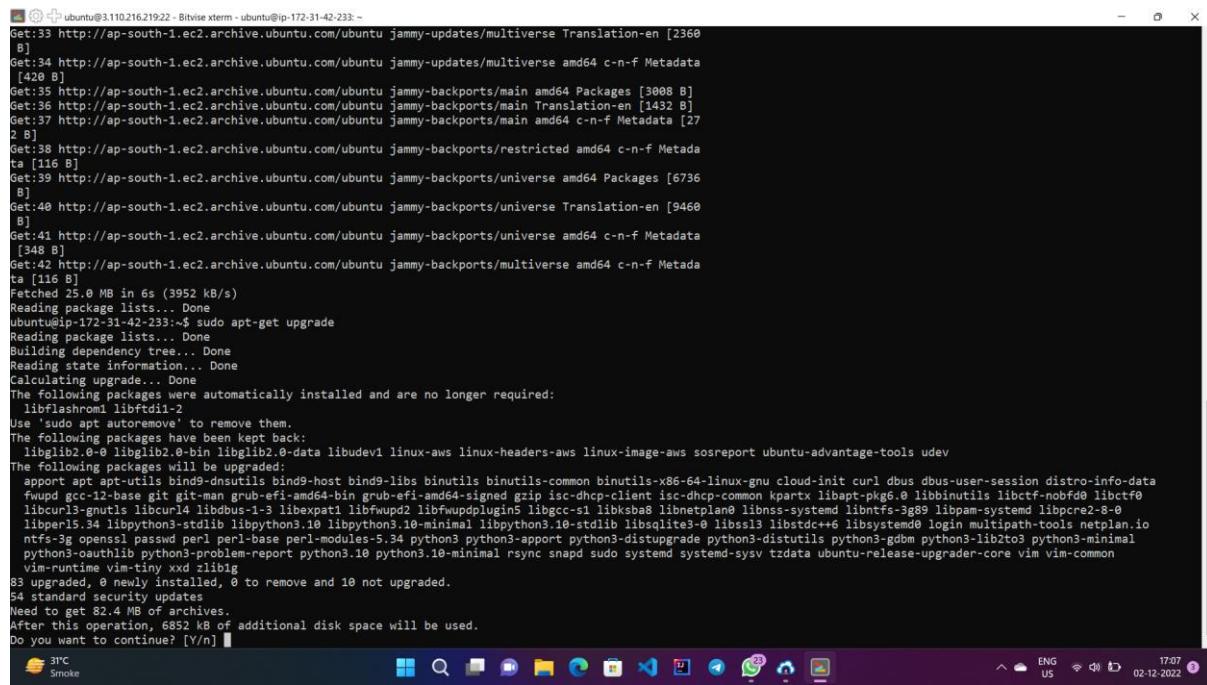
Install Apache

Let's jump right in and install Apache first. To do this, execute the following command.

Sudo apt-get install apache2



```
ubuntu@ip-172-31-41-251:~$ sudo apt-get install apache2
```



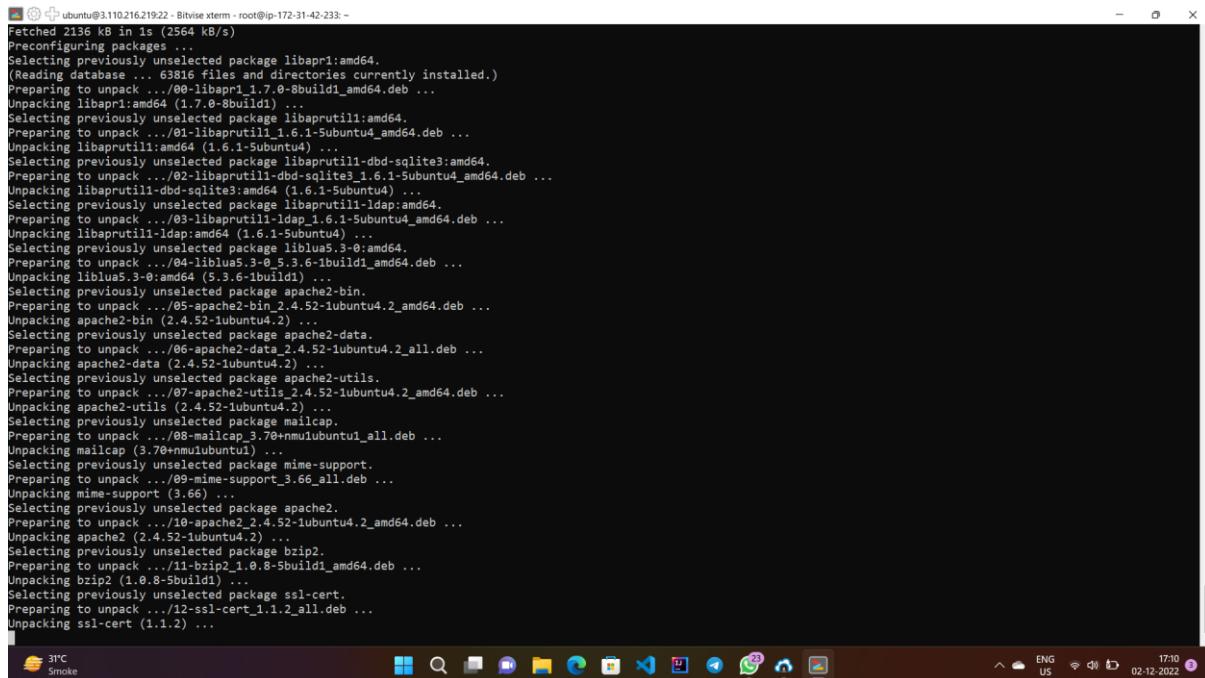
```
ubuntu@ip-172-31-42-233:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libflashrom libfdisk-2
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  libglib2.0-0 libglib2.0-0-data libudev1 linux-aws linux-headers-aws linux-image-aws sosreport ubuntu-advantage-tools udev
The following packages will be upgraded:
  apt apt-utils bind9-dnsutils bind9-host bind9-libs binutils binutils-common binutils-x86_64-linux-gnu cloud-init curl dbus dbus-user-session distro-info-data
  fwupd gcc-12-base git git-man grub-efi-amd64-bin grub-efi-amd64-signed gzip isc-dhcp-client isc-dhcp-common kpartx libapt-pkg6.0 libbinutils libctf-nobfd0 libctf0
  libcurl3-gnutls libcurl4 libdbus-1-3 libexpat1 libfwupd2 libfwupdplugin5 libgcc-s1 libksba8 libnetplan0 libnss-systemd libntfs-3g89 libpam-systemd libpcre2-8-0
  libperl5.34 libpython3-stdlib libpython3.10 libpython3.10-minimal libpython3.10-stdlib libsqllite3-0 libssl3 libstdc++6 libsystemd logind multipath-tools netplan.io
  ntfs-3g openssl passwd perl perl-base perl-modules-5.34 python3 python3-apport python3-distupgrade python3-dbus python3-gdbm python3-libzto3 python3-minimal
  python3-oauthlib python3-problem-report python3.10 python3.10-minimal rsync snapd sudo systemd systemd-sysv tzdata ubuntu-release-upgrader-core vim vim-common
  vim-runtime vim-tiny xxd zlib1g
83 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
54 standard security updates
Need to get 82.4 MB of archives.
After this operation, 6852 kB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Step 24:To confirm that Apache is installed on your system, execute the following command.

```
systemctl status apache2
```



```
ubuntu@ip-172-31-41-251:~$ systemctl status apache2
```



```
Fetched 2136 kB in 1s (2564 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libapr1:amd64.
(Reading database ... 63816 files and directories currently installed.)
Preparing to unpack .../00-libapr1_1.7.0-0build1_amd64.deb ...
Unpacking libapr1:amd64 (1.7.0-0build1) ...
Selecting previously unselected package libaprutil1:amd64.
Preparing to unpack .../01-libaprutil1_1.6.1-Subuntu4_amd64.deb ...
Unpacking libaprutil1:amd64 (1.6.1-Subuntu4) ...
Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.
Preparing to unpack .../02-libaprutil1-dbd-sqlite3_1.6.1-Subuntu4_amd64.deb ...
Unpacking libaprutil1-dbd-sqlite3:amd64 (1.6.1-Subuntu4) ...
Selecting previously unselected package libaprutil1-ldap:amd64.
Preparing to unpack .../03-libaprutil1-ldap_1.6.1-5ubuntu4_amd64.deb ...
Unpacking libaprutil1-ldap:amd64 (1.6.1-5ubuntu4) ...
Selecting previously unselected package liblbu5_3.0-5.3.6-1build1_amd64.deb ...
Unpacking liblbu5_3.0-5.3.6-1build1_amd64 (5.3.6-1build1) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../05-apache2-bin_2.4.52-1ubuntu4.2_amd64.deb ...
Unpacking apache2-bin (2.4.52-1ubuntu4.2) ...
Selecting previously unselected package apache2-data.
Preparing to unpack .../06-apache2-data_2.4.52-1ubuntu4.2_all.deb ...
Unpacking apache2-data (2.4.52-1ubuntu4.2) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../07-apache2-utils_2.4.52-1ubuntu4.2_amd64.deb ...
Unpacking apache2-utils (2.4.52-1ubuntu4.2) ...
Selecting previously unselected package mailcap.
Preparing to unpack .../08-mailcap_3.78+mmulubuntu1_all.deb ...
Unpacking mailcap (3.78+mmulubuntu1) ...
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_2.4.52-1ubuntu4.2_amd64.deb ...
Unpacking apache2 (2.4.52-1ubuntu4.2) ...
Selecting previously unselected package bzip2.
Preparing to unpack .../11-bzip2_1.0.8-5build1_amd64.deb ...
Unpacking bzip2 (1.0.8-5build1) ...
Selecting previously unselected package ssl-cert.
Preparing to unpack .../12-ssl-cert_1.1.2_all.deb ...
Unpacking ssl-cert (1.1.2) ...
```

Step 25: To verify further, open your browser and go to your server's IP Address

[https:// 3.1.10.216.19](https://3.1.10.216.19)


```

ubuntu@3.110.216.219:22 - Bitvise xterm - root@ip-172-31-42-233: ~
Dec 02 11:40:49 ip-172-31-42-233 systemd[1]: Started The Apache HTTP Server.
root@ip-172-31-42-233:~# apt mariadb-server mariadb-client
E: Invalid operation mariadb-server
root@ip-172-31-42-233:~# apt mariadb-server mariadb-client
E: Invalid operation mariadb-server
root@ip-172-31-42-233:~# apt install mariadb-server mariadb-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom libftdi1-2
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  galera-4 libcgifast-perl libclone-perl libconfig-inifiles-perl libdaxctl1 libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl
  libfcgi0dbi libhtml-parser-perl libhtml-tagset-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmariadb3
  libmysqclient21 libndt16 libpmem1 libsnappy1v5 libtimestream-perl liburi-perl liburing2 mariadb-client-10.6 mariadb-client-core-10.6 mariadb-common mariadb-server-10.6
  mariadb-server-core-10.6 mysql-common socat
Suggested packages:
  libltdm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx mariadb-test
The following NEW packages will be installed:
  galera-4 libcgifast-perl libclone-perl libconfig-inifiles-perl libdaxctl1 libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-bin libfcgi-perl
  libfcgi0dbi libhtml-parser-perl libhtml-tagset-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmariadb3
  libmysqclient21 libndt16 libpmem1 libsnappy1v5 libtimestream-perl liburi-perl liburing2 mariadb-client mariadb-client-core-10.6 mariadb-common
  mariadb-server mariadb-server-10.6 mariadb-server-core-10.6 mysql-common socat
0 upgraded, 36 newly installed, 0 to remove and 10 not upgraded.
Need to get 18.5 MB of archives.
After this operation, 165 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/universe amd64 mariadb-common all 1:10.6.11-0ubuntu0.22.04.1 [16.6 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 galera-4 amd64 26.4.9-1build1 [720 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libdbi-perl amd64 1.643-3build3 [741 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libconfig-inifiles-perl all 3.000003-1 [40.5 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libmariadb3 amd64 1:10.6.11-0ubuntu0.22.04.1 [174 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 mariadb-client-core-10.6 amd64 1:10.6.11-0ubuntu0.22.04.1 [920 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 mariadb-client-10.6 amd64 1:10.6.11-0ubuntu0.22.04.1 [1533 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libdbaxctl1 amd64 72.1-1 [19.8 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libndt16 amd64 72.1-1 [57.7 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libpmem1 amd64 1.11.1-3build1 [81.4 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libsnappy1v5 amd64 1.1.8-1build3 [17.5 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liburing2 amd64 2.1-2build1 [10.3 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 mariadb-server-core-10.6 amd64 1:10.6.11-0ubuntu0.22.04.1 [7682 kB]
26% [14 mariadb-server-core-10.6 0 B/7682 kB 0%]

```

Step 27: Let's now secure our MariaDB database engine and disallow remote root login..

\$ mysql_secure_installation

```
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-42-233:~# mysql_secure_installation
```

The first step will prompt you to change the root password to login to the database. You can opt to change it or skip if you are convinced that you have a strong password. To skip changing type n. For safety's sake, you will be prompted to remove anonymous users. Type Y

```
ubuntu@3.110.216.219:~ - Bitvise xterm - root@ip-172-31-42-233: ~
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-42-233:~# mysql_secure_installation

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):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into MariaDB root user without the proper authorisation.

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

Switch to unix_socket authentication [Y/n] n
... skipping.

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

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB 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? [Y/n] y
... Success!

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

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

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

```
Remove anonymous users? [Y/n] y
... Success!

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

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

Next, remove the test database.

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

Finally, reload the database to effect the changes.

```
Reloading the privilege tables will ensure that all changes made so far  
will take effect immediately.  
Reload privilege tables now? [Y/n] y
```

Step 28: Install PHP

Lastly, we will install PHP as the last component of the LAMP stack.

Sudo apt install php php-mysql

```
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-172-31-42-233:~# sudo apt install php php-mysql
```

```
root@ip-172-31-42-233:~# sudo apt install php php-mysql  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
libflashrom1 libftdi1-2  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
libapache2-mod-php8.1 php-common php8.1 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-php8.1 php php-common php-mysql php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline  
0 upgraded, 10 newly installed, 0 to remove and 10 not upgraded.  
Need to get 5258 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-lubuntu2.9 [1126 kB]  
Fetched 13.5 MB in 0s (2.7 MB/s)
```

Step 29: Nowe move to /var/www/html folder, use command:

Command: cd /etc/www/html

```
[root@ip-172-31-42-233:~# cd /var/www/html  
root@ip-172-31-42-233:/var/www/html#
```

Step 30: By executing is command following content will display:

Command: ls

```
[root@ip-172-31-42-233:~# cd /var/www/html  
root@ip-172-31-42-233:/var/www/html# ls  
index.html  
root@ip-172-31-42-233:/var/www/html# ]
```

Step 31: Add the info.php file in the step 29 directory, for this use following command:

Command: sudo nano info.php

```
[root@ip-172-31-42-233:/var/www/html# ls  
index.html  
root@ip-172-31-42-233:/var/www/html# sudo nano info.php
```

Step 32: Add the following in the info.php file

```
</php
```

```
phpinfo();
```

```
?>
```

```
ubuntu@3.110.216.219:~$ nano 6.2
GNU nano 6.2
info.php *  

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

The screenshot shows a terminal window titled "info.php *". It contains the PHP code `<?php phpinfo(); ?>`. The terminal is running on an Ubuntu system, as indicated by the prompt "ubuntu@3.110.216.219:~\$". The bottom of the window shows the nano editor's command bar with various keyboard shortcuts.

Step 33: Open your browser and append/info.php to the server's URL.

3.110.216.219/info.php

Where 3.110.216.219 is your instance public address(step 14)

The screenshot shows a web browser window with the title "PHP 8.1.2-1ubuntu2.9 - phpinfo0". The address bar shows the URL "3.110.216.219/info.php". The page content is the PHP info output, which includes detailed information about the PHP configuration, including build details, extensions, and API versions. The output is presented in a table format with many rows of configuration parameters and their values.

System	Value
Build Date	Oct 19 2022 14:58:09
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2/php.ini
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-gc-cache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendars.ini, /etc/php/8.1/apache2/conf.d/20-crypt.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-freetype.ini, /etc/php/8.1/apache2/conf.d/20-gd.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-snmp.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	API20210902.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
Registered Stream Filters	zlib*, string.rot13, string.toupper, string.tolower, convert*, consumed, dechunk, convert.iconv*

Step 34: Create WordPress Database

Now it's time to log in to our MariaDB database as root and create a database for accommodating our WordPress data.

```
$ mysql -u root -p
```

```
ubuntu@ip-172-31-42-233:~$ mysql -u root -p
```

```
ubuntu@ip-172-31-42-233:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.6.11-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 35: Create a database for our WordPress installation.

```
CREATE DATABASE wordpress_db;
```

```
ubuntu@ip-172-31-42-233:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.6.11-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)]> CREATE DATABASE wordpress_db;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]>
```

Step 36 Next, create a database user for our WordPress setup.

**CREATE USER 'shaikh1'@'localhost'
IDENTIFIED BY 'password'**

```
ubuntu@ip-172-31-42-233:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.6.11-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)]> CREATE DATABASE wordpress_db;
Query OK, 1 row affected (0.000 sec)

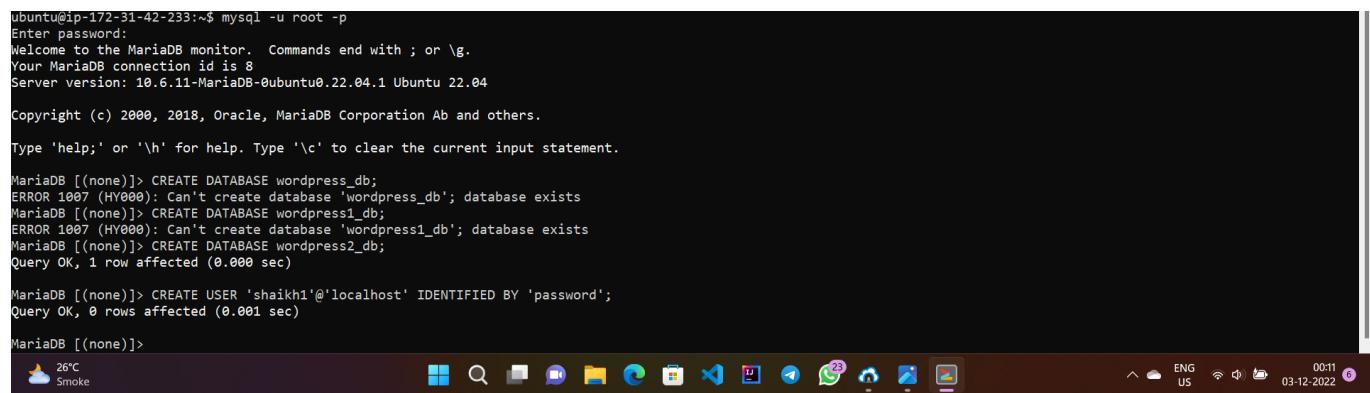
MariaDB [(none)]>
```



Step 37: Grant privileges to the user

Next, grant the user permissions to access the database

```
GRANT ALL ON wordpress_dh. To 'shaikh1'@'localhost'  
IDENTIFIED BY 'password';
```



```
Ubuntu@ip-172-31-42-233:~$ mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 8  
Server version: 10.6.11-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)]> CREATE DATABASE wordpress_db;  
ERROR 1007 (HY000): Can't create database 'wordpress_db'; database exists  
MariaDB [(none)]> CREATE DATABASE wordpress1_db;  
ERROR 1007 (HY000): Can't create database 'wordpress1_db'; database exists  
MariaDB [(none)]> CREATE DATABASE wordpress2_db;  
Query OK, 1 row affected (0.000 sec)  
  
MariaDB [(none)]> CREATE USER 'shaikh1'@'localhost' IDENTIFIED BY 'password';  
Query OK, 0 rows affected (0.001 sec)  
  
MariaDB [(none)]>
```

Step 38: Great, now you can exit the database.

FLUSH PRIVILEGES:

Exit:

```
*** System restart required ***
Last login: Fri Dec  2 18:37:19 2022 from 182.68.14.180
ubuntu@ip-172-31-42-233:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.6.11-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)]> CREATE DATABASE wordpress_db;
ERROR 1007 (HY000): Can't create database 'wordpress_db'; database exists
MariaDB [(none)]> CREATE DATABASE wordpress1_db;
ERROR 1007 (HY000): Can't create database 'wordpress1_db'; database exists
MariaDB [(none)]> CREATE DATABASE wordpress2_db;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'shaikh1'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> GRANT ALL ON wordpress_db.* TO 'shaikh1'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> Exit;
Bye
ubuntu@ip-172-31-42-233:~$ █
  26°
Smoke
  ENG
  US
  00:15
  03-12-2022
```

Step 39: Install WordPress CMS

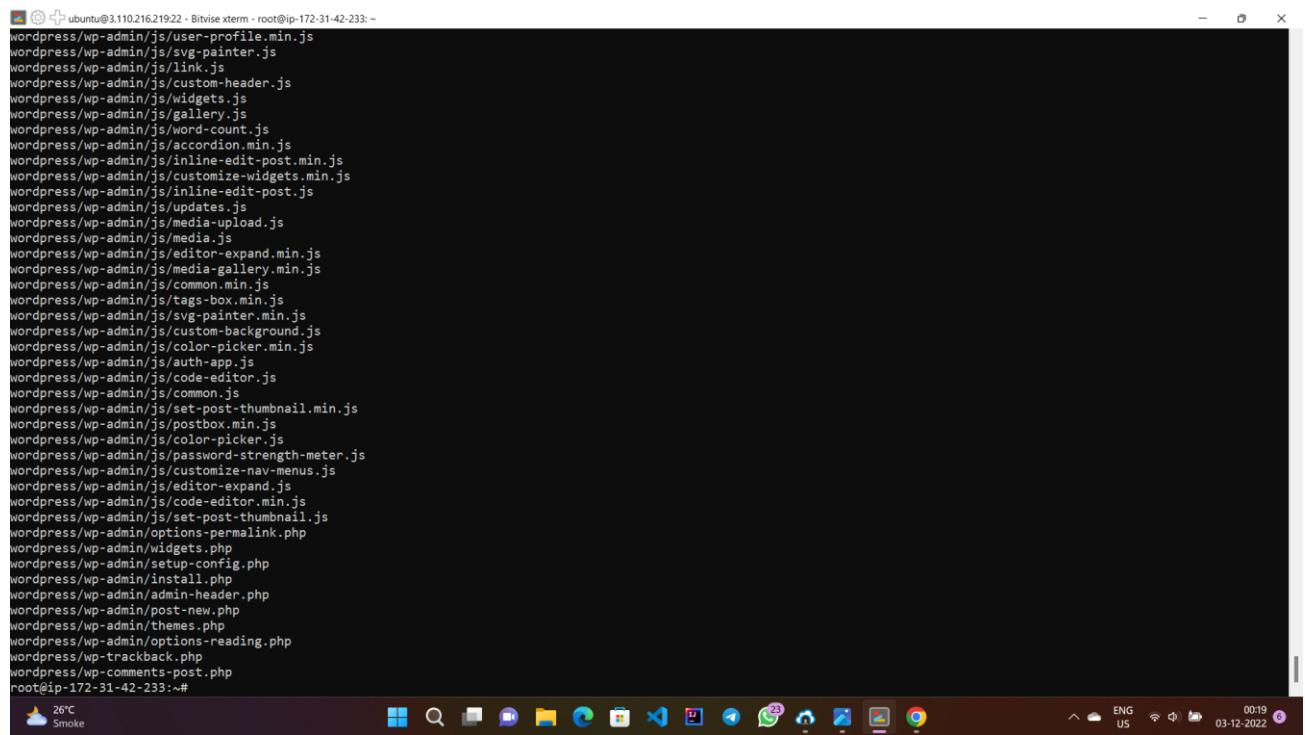
Go to your temp directory and download the latest WordPress File:

```
ubuntu@ip-172-31-42-233:~$ sudo su -
root@ip-172-31-42-233:~# https://wordpress.org/latest.tar.gz
-bash: https://wordpress.org/latest.tar.gz: No such file or directory
root@ip-172-31-42-233:~# wget https://wordpress.org/latest.tar.gz
```

Step 40: Next, Uncompress the tarball which will generate a folder called "wordpress".

tar-xvf latest.tar.gz

```
root@ip-172-31-42-233:~# tar -xvf latest.tar.gz
```



The screenshot shows a terminal window on a Linux desktop environment. The command `tar -xvf latest.tar.gz` has been run, and the output lists numerous JavaScript files extracted from the WordPress admin theme. The files include user-profile.min.js, svg-painter.min.js, link.js, custom-header.js, widgets.js, gallery.js, word-count.js, accordion.min.js, inline-edit-post.min.js, customize-widgets.min.js, inline-edit-post.js, updates.js, media-upload.js, media.js, editor-expand.min.js, media-gallery.min.js, common.min.js, tags-box.min.js, svg-painter.min.js, custom-background.js, color-picker.min.js, auth-app.js, code-editor.js, common.js, set-post-thumbnail.min.js, postbox.min.js, color-picker.js, password-strength-meter.js, customize-nav-menus.js, editor-expand.js, code-editor.min.js, set-post-thumbnail.js, options-permalink.php, widgets.php, setup-config.php, install.php, admin-header.php, post-new.php, themes.php, options-reading.php, trackback.php, comments-post.php. The terminal window is titled "ubuntu@3.110.216.219:22 - Bitvise xterm - root@ip-172-31-42-233:~". The desktop taskbar at the bottom shows various application icons, and the system tray indicates it's 00:19 on 03-12-2022.

Step 41: Copy the wordpress folder to /var/www/html/path.

```
cp-R wordpress /var/www/html/
```

```
root@ip-172-31-42-233:~# cp -R wordpress /var/www/html  
root@ip-172-31-42-233:~#
```

Step 42: Run the command below to change ownership of “wordpress” directory.

Chown – www-data:www-data /var/www/html/wordpress/

```
root@ip-172-31-42-233:~# chown -R www-data:www-data /var/www/html/wordpress/  
root@ip-172-31-42-233:~#
```

Step 43: change File permissions of the WordPress folder.

chmod -R755 /var/www/html/wordpress/

```
root@ip-172-31-42-233:~# chmod -R 755 /var/www/html/wordpress/  
root@ip-172-31-42-233:~#
```

Step 44: Create 'uploads' directory.

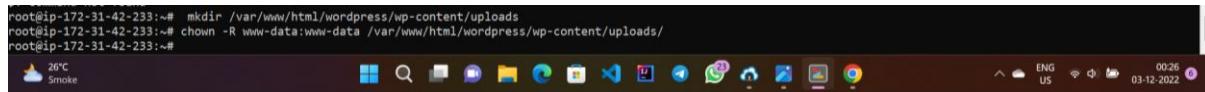
```
mkdir /var/www/html/wordpress/wp-content/uploads
```



```
root@ip-172-31-42-233:~# cp -R wordpress /var/www/html
root@ip-172-31-42-233:~# chown -R www-data:www-data /var/www/html/wordpress/
root@ip-172-31-42-233:~# chmod -R 755 /var/www/html/wordpress/
root@ip-172-31-42-233:~# $ mkdir /var/www/html/wordpress/wp-content/uploads
26°C Smoke 00:23 03-12-2022
```

Step 45: Finally, change permissions of 'uploads' directory.

```
chown - www-data:www-data /var/www/html/wordpress/wp-
content/uploads/
```

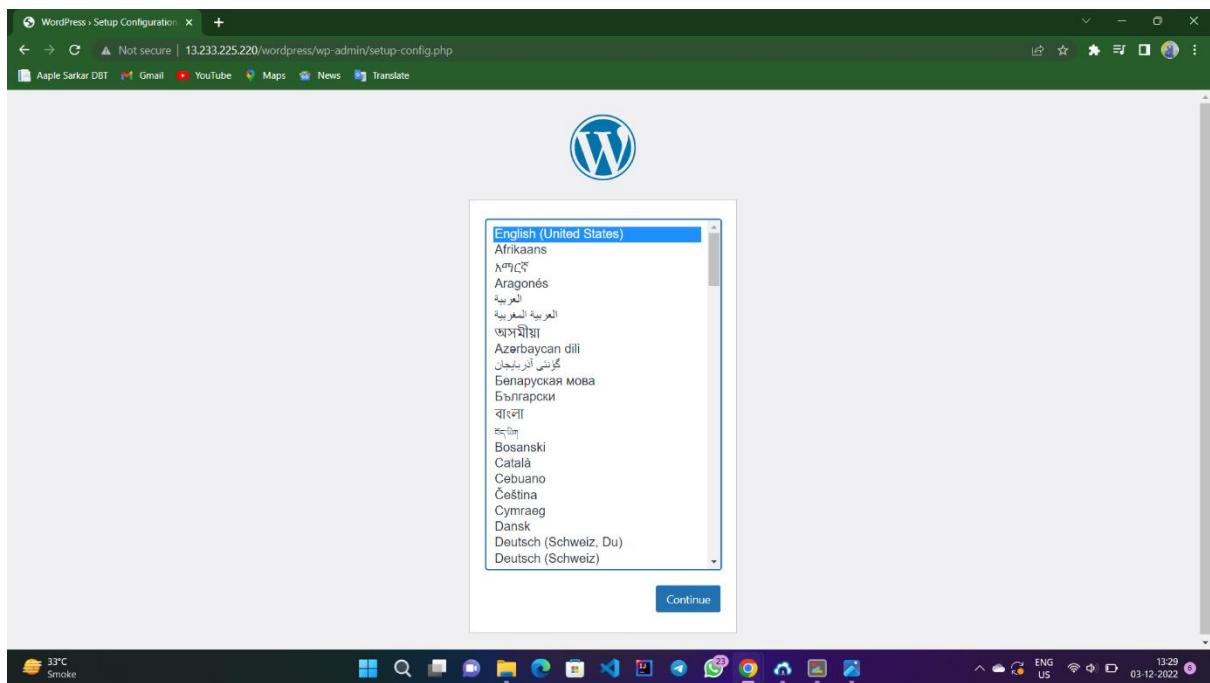


```
root@ip-172-31-42-233:~# mkdir /var/www/html/wordpress/wp-content/uploads
root@ip-172-31-42-233:~# chown -R www-data:www-data /var/www/html/wordpress/wp-content/uploads/
root@ip-172-31-42-233:~# 26°C Smoke 00:26 03-12-2022
```

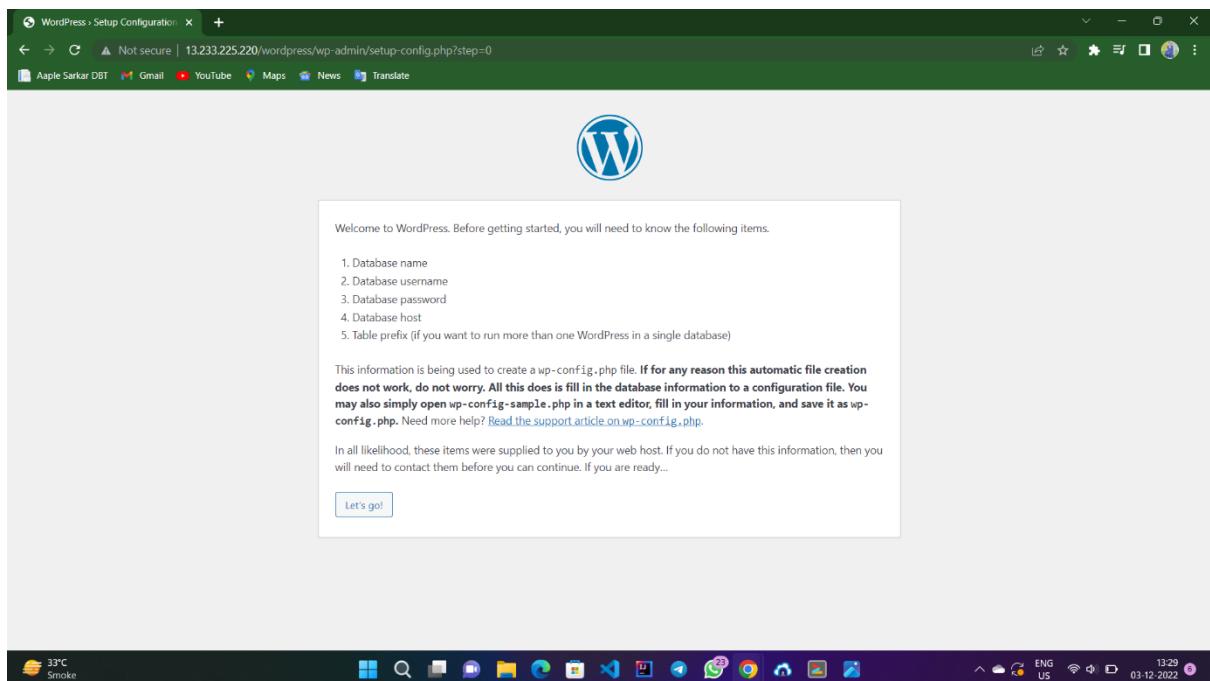
Step 46: Open your browser and go to the server's URL. In my case it's

3.110.216.219/wordpress

*use your instance public ip **



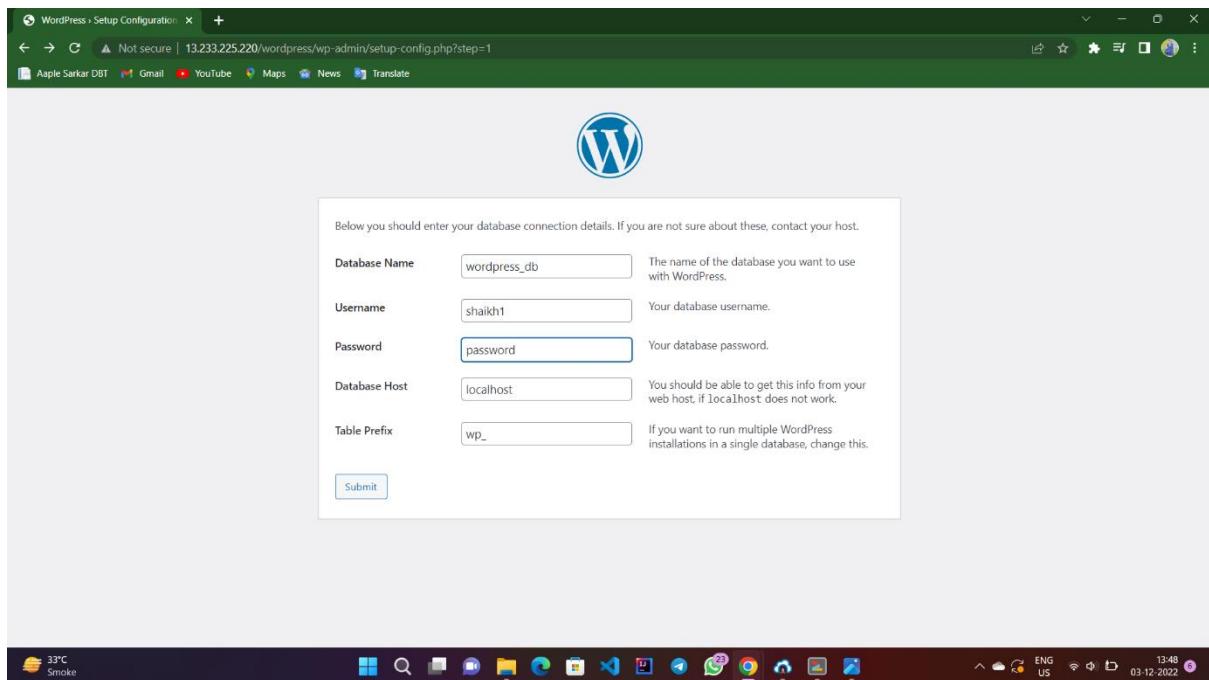
Step 47: You'll be presented with a WordPress wizard and a list of credentials required to successfully set it up.



Step 48: Fill out the form as shown with the credentials specified when creating the WordPress database in the MariaDB database.

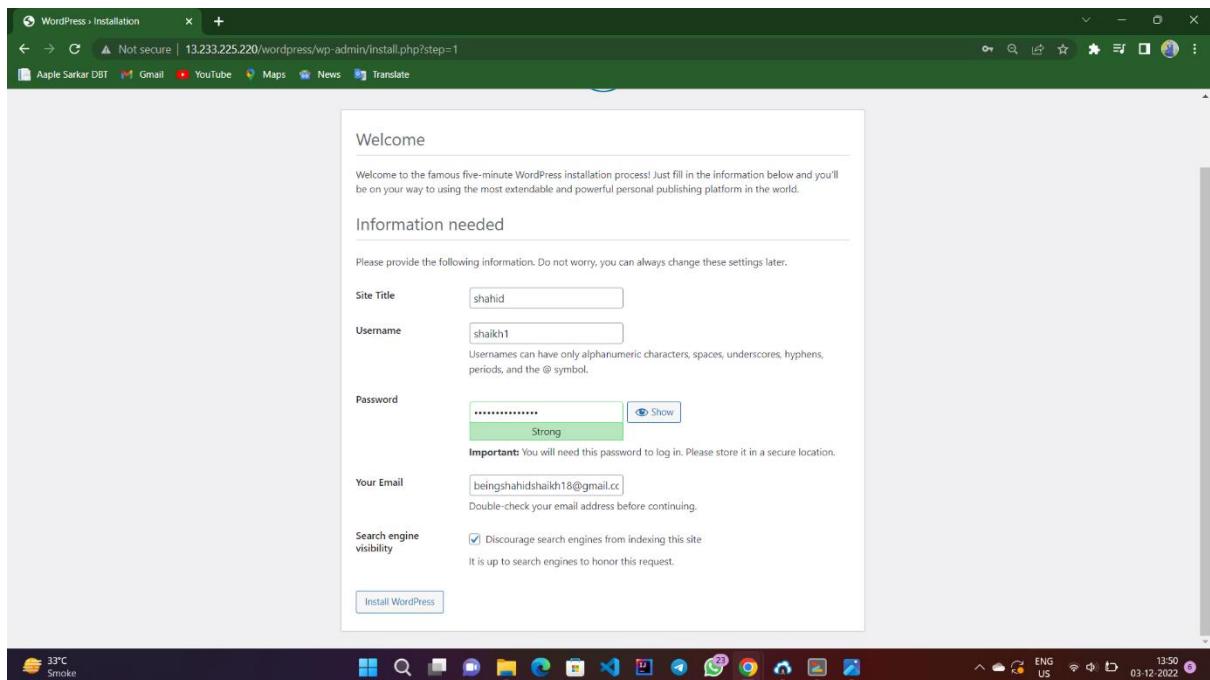
Leave out the database host and table prefix and Hit "Submit" button.

****** Use the all the information (Step 35 to step 37) and fill the form properly******



Step 49: If all the details are correct, you will be given the green light to proceed. Run the installation.

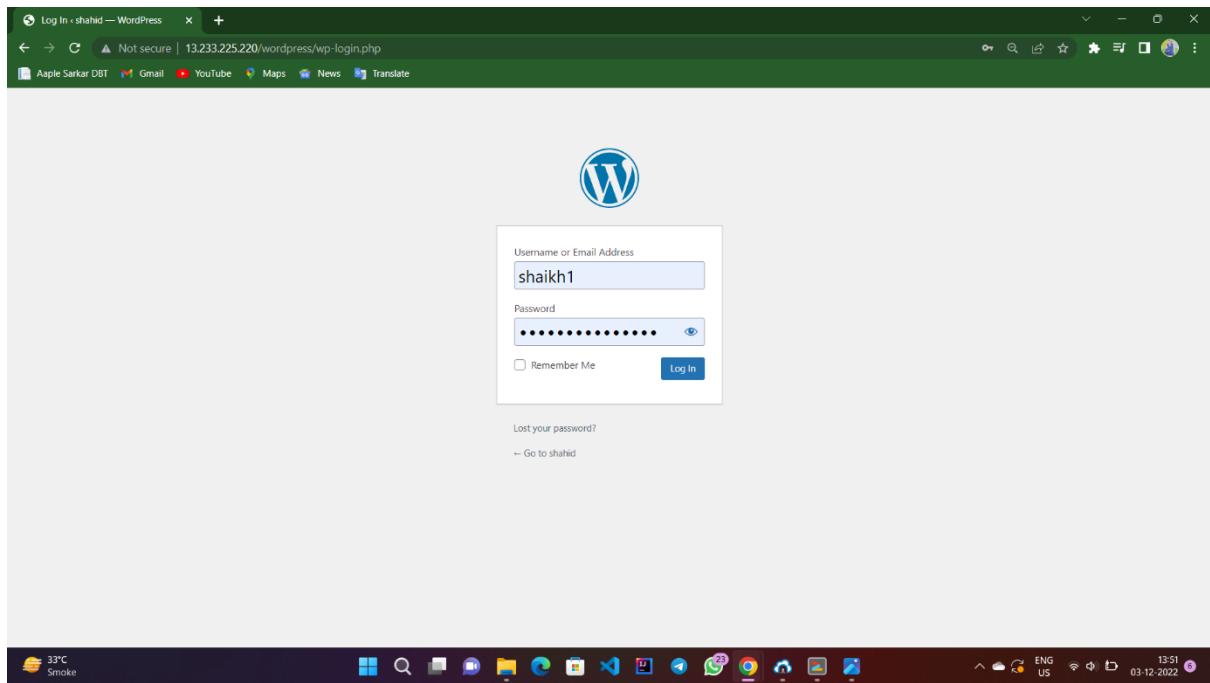
Step 50: Fill out the additional details required such as site title, Username, and Password and save them somewhere safe lest you forget. Ensure to use a strong password.

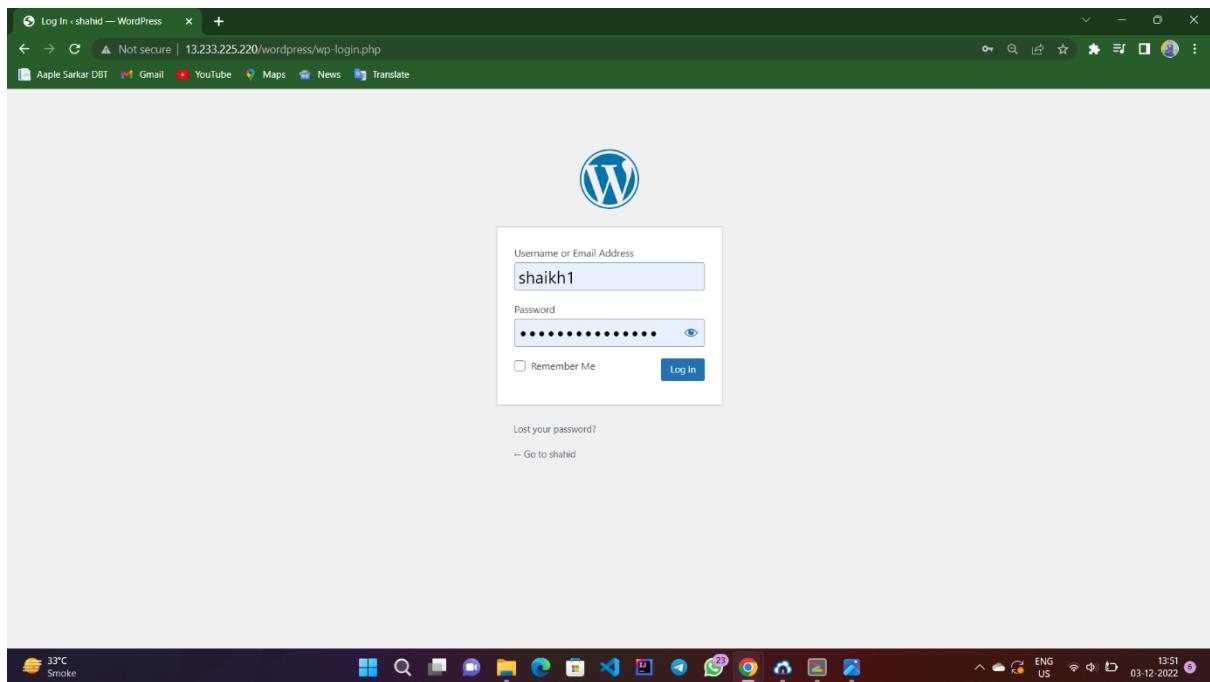


Step 51: Scroll down and Hit 'Install WordPress'.

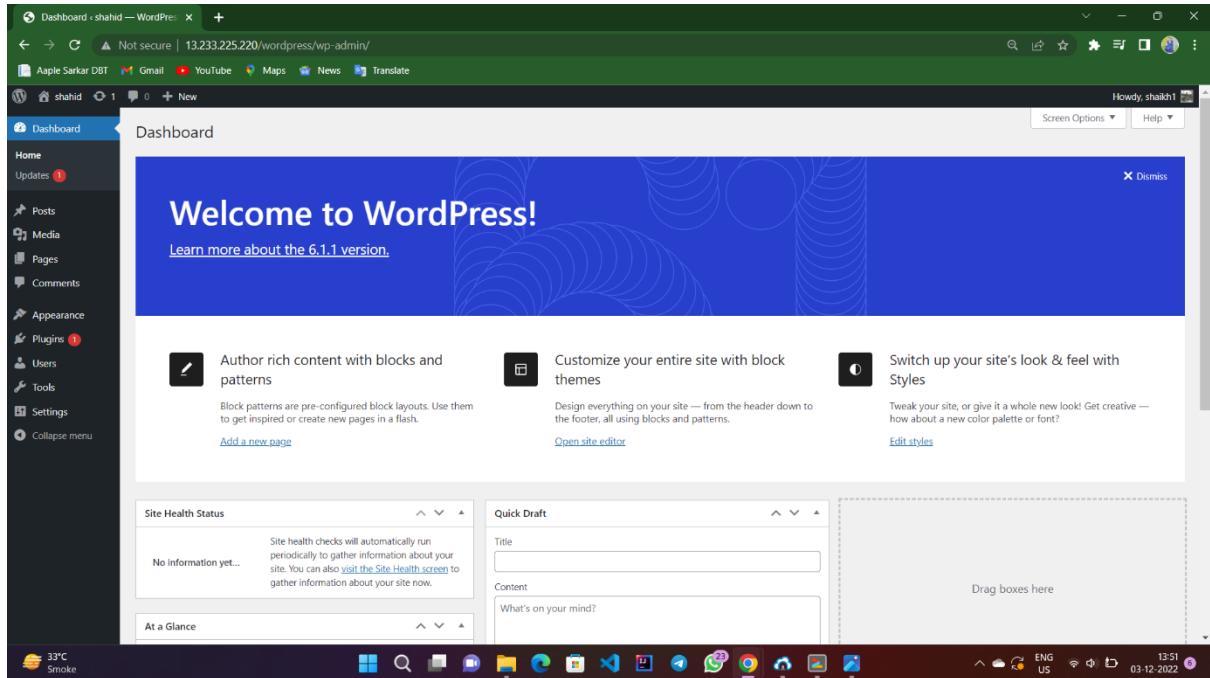
If all went well, then you will get a 'Success' notification as shown.

Step 52: Click on the 'Login' button to get to access the Login page of your fresh WordPress installation.

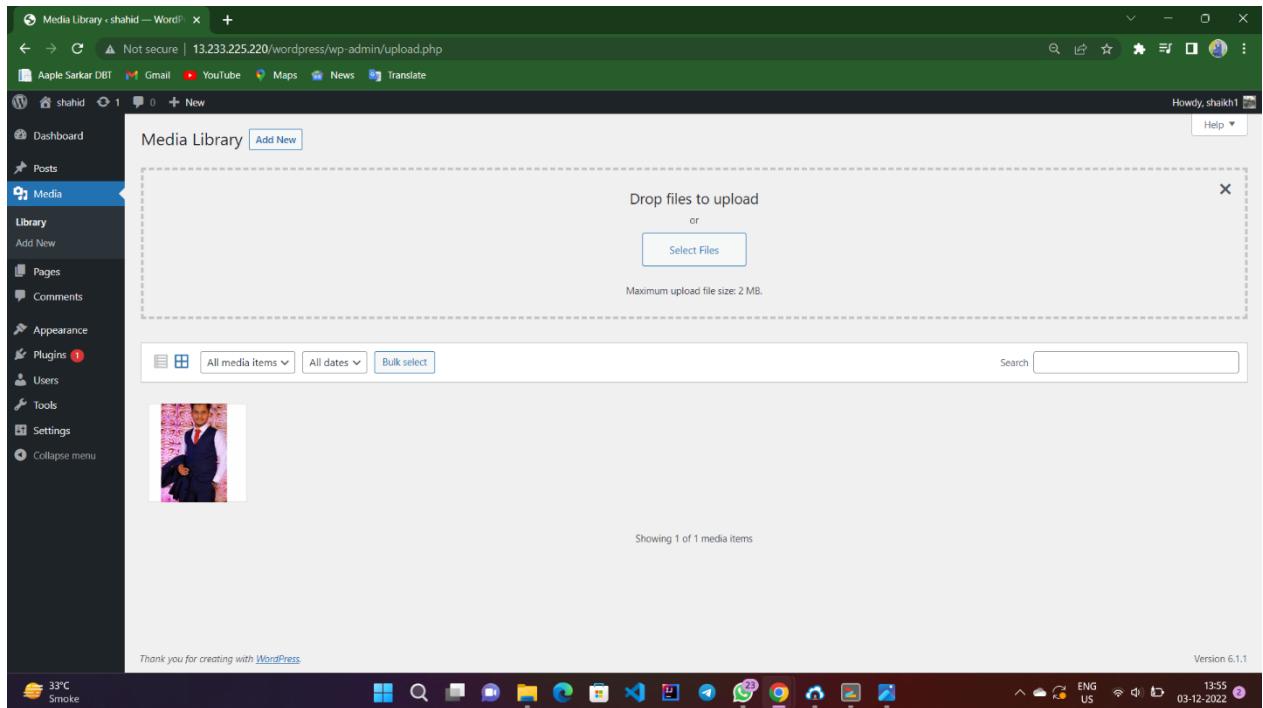




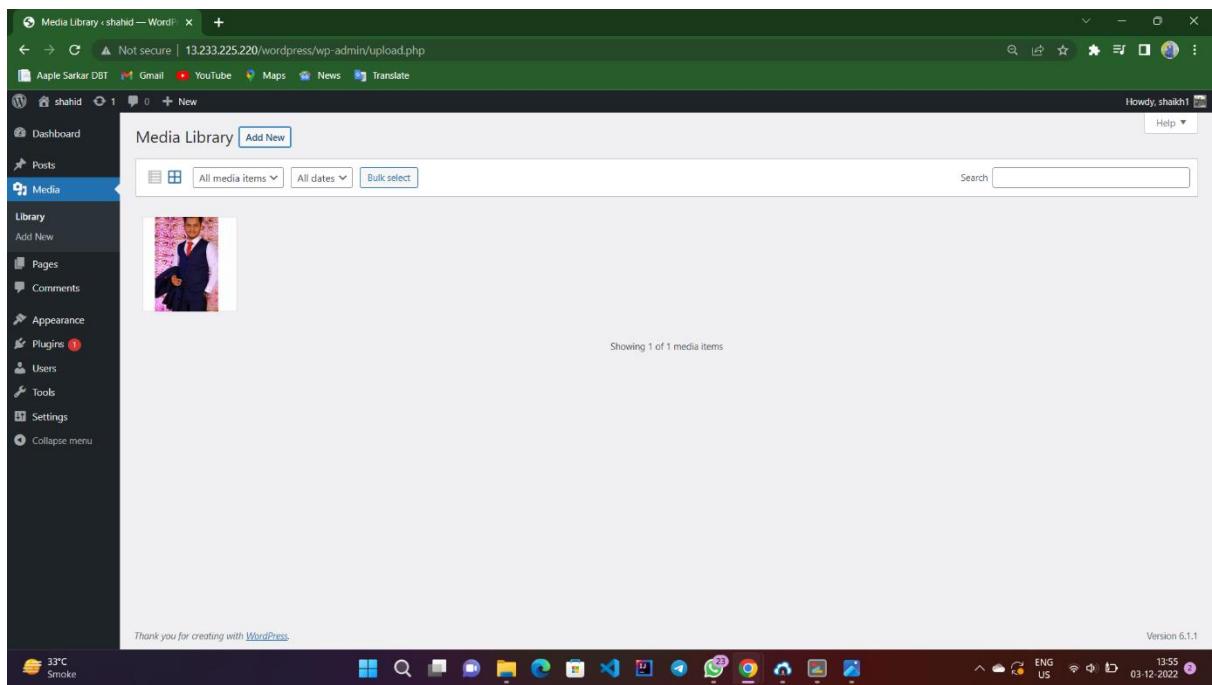
Step 53: Provide your login credentials and hit 'Login'.



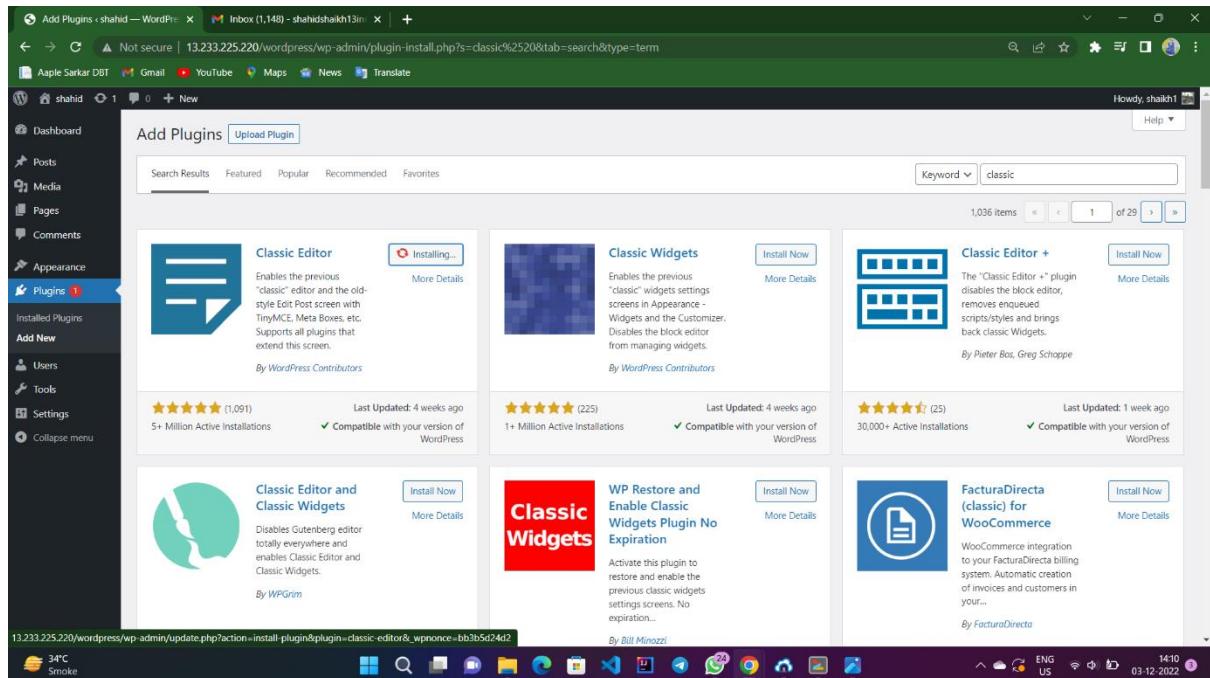
Step 54: Select media option from the left side and a file that you want to save:



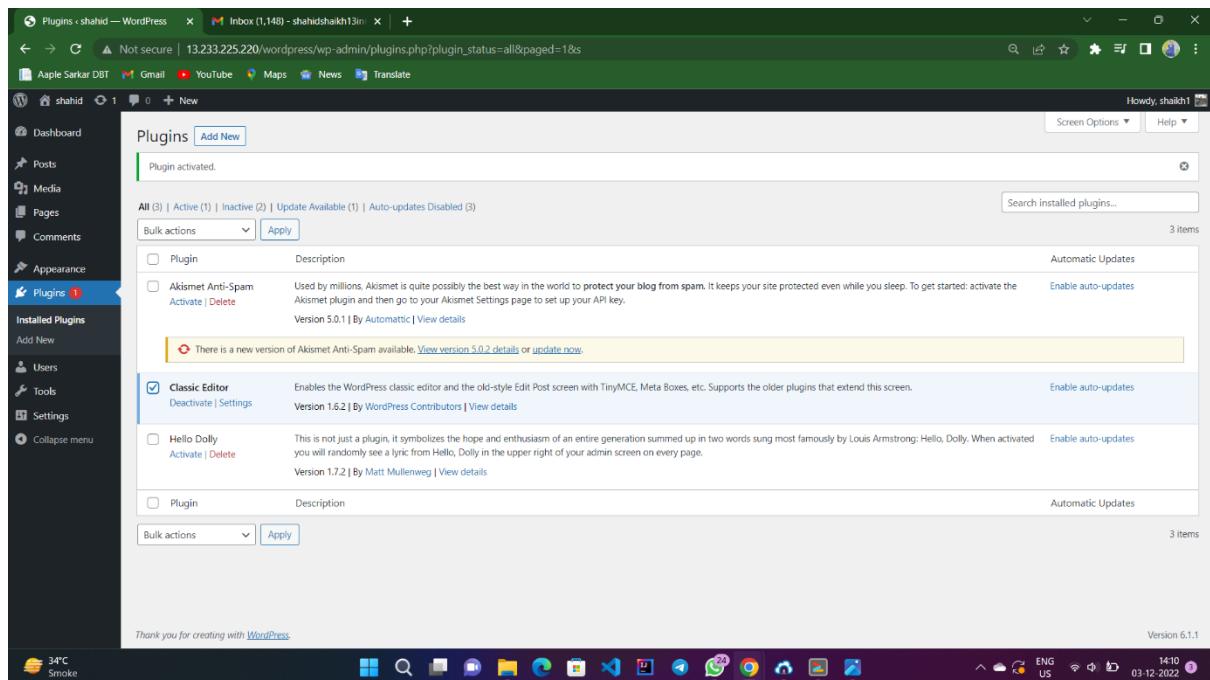
Now you can see that the image is stores on it.



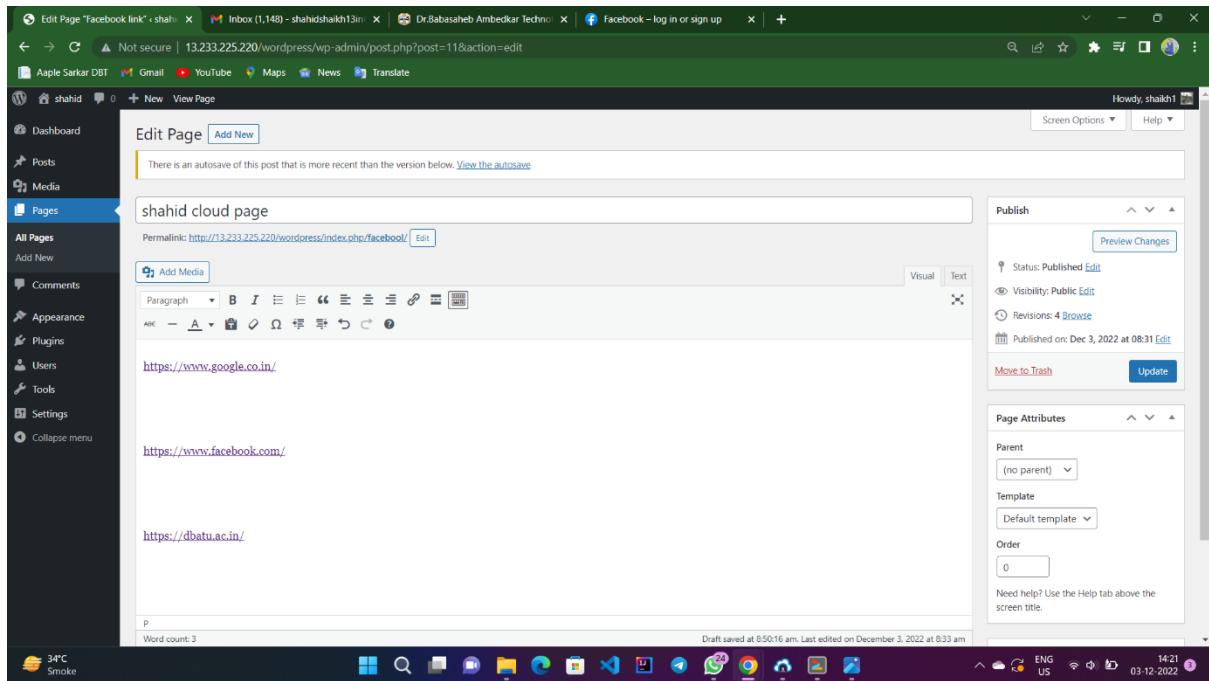
Step 55: Add plugins, for this use the plugin option from left side.



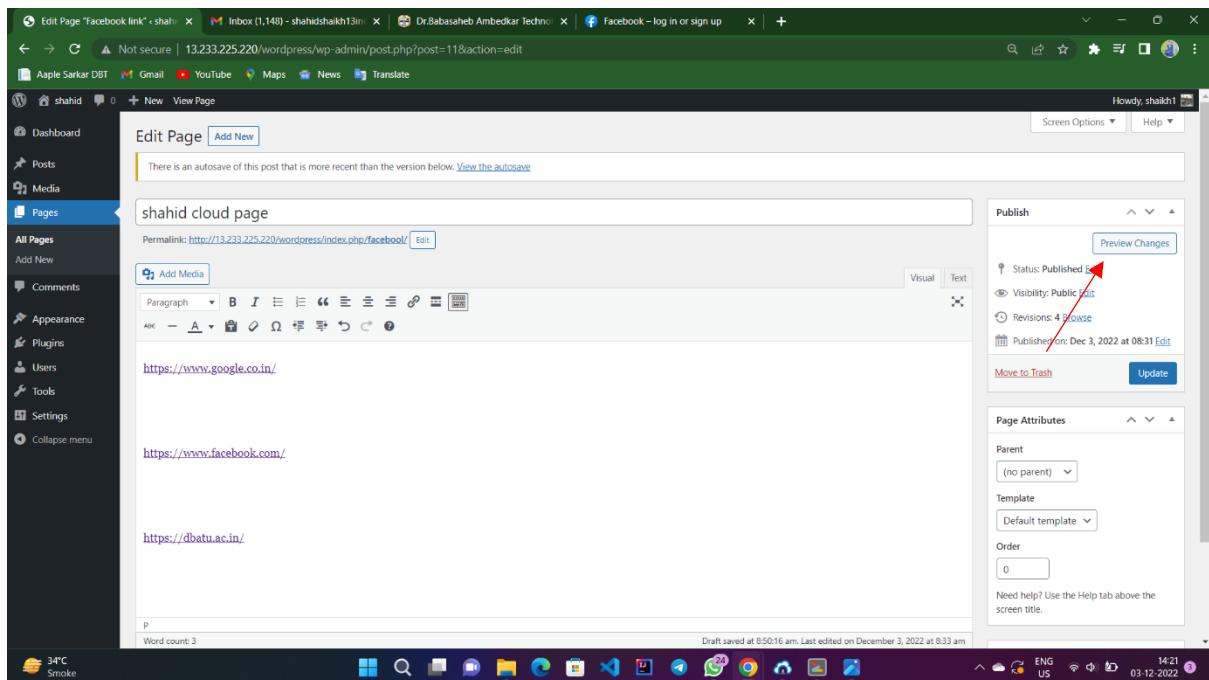
The plugin is now added.



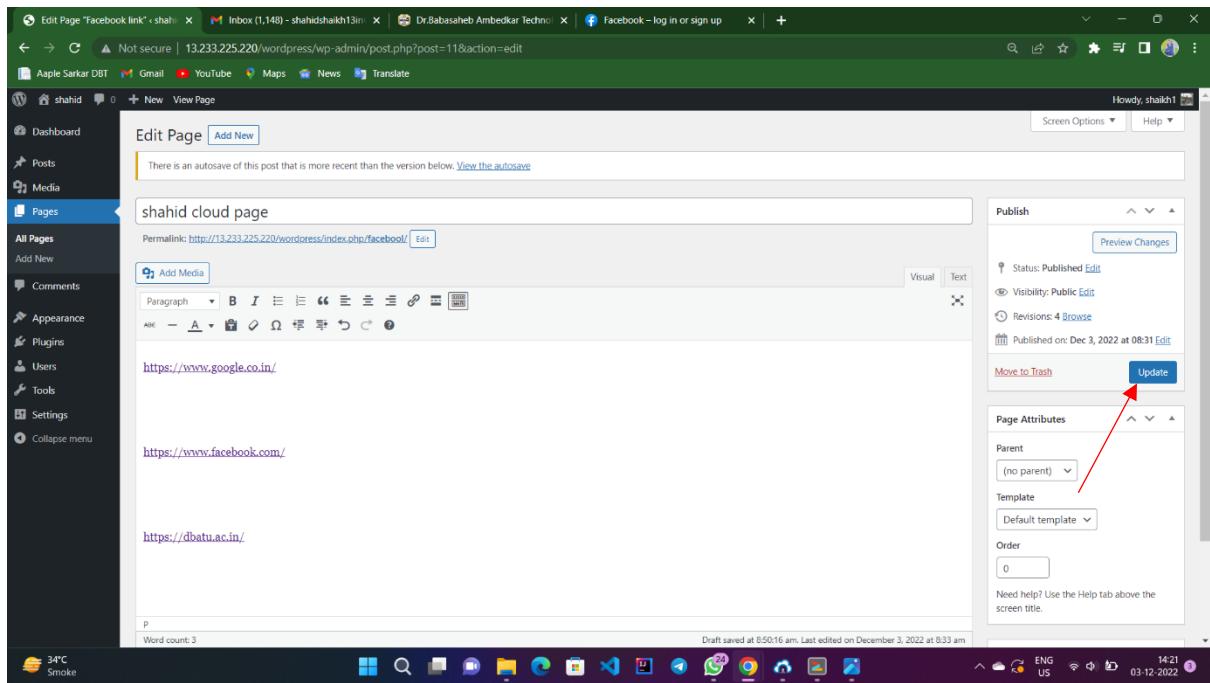
Step 56: To create social media account reference, use page option from left side and click on it and then create a hyperlink of the social media account that you want to use it.



To create this page use the publish button and then click on it.

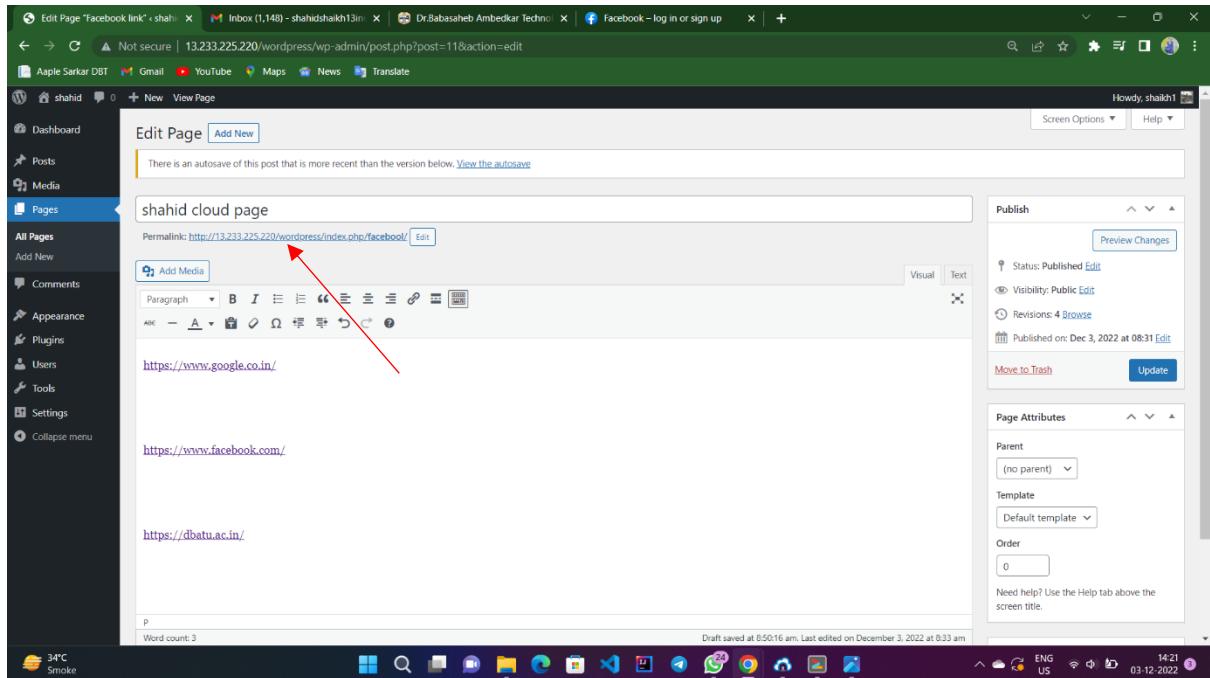


Then update the page if it requires, for this use the update button.

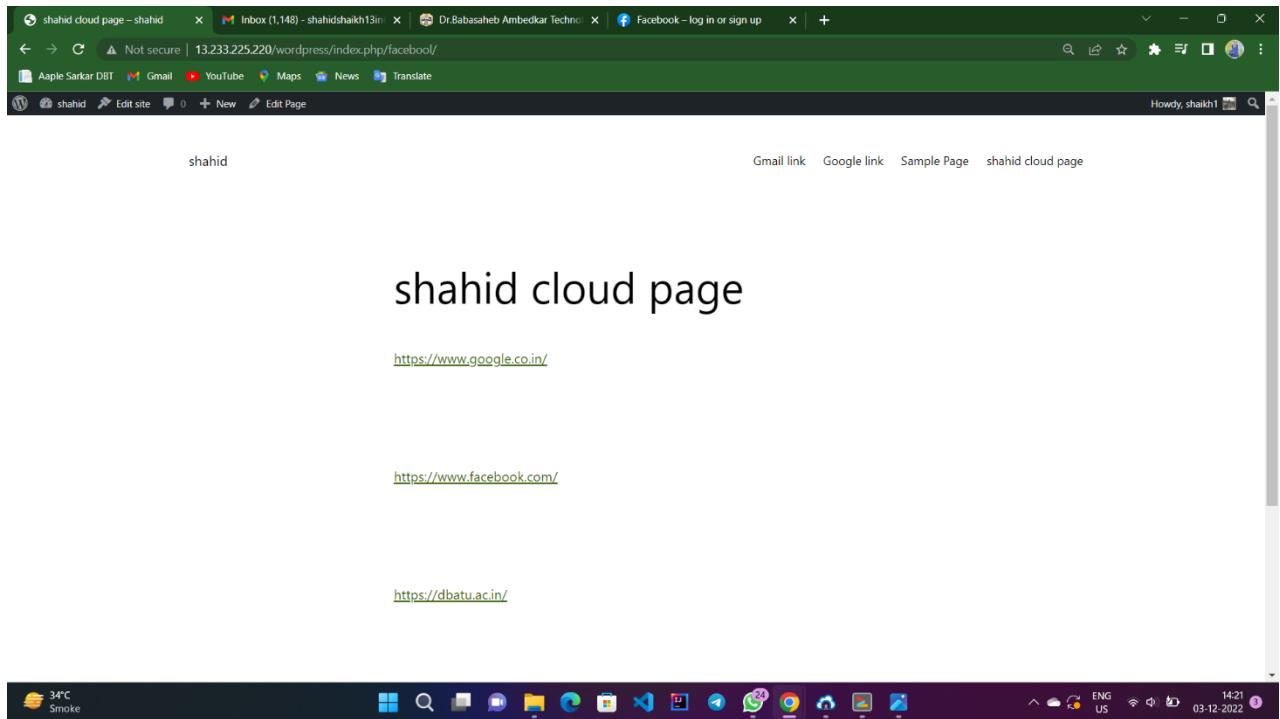


Step 57: Now you can access this page by using the ' permalink' option provided at step 57..

****see the ip address that is shown by using arrow,paste it on chrome.****



Step 58: Now you can access use own wordpress page.



Congratulations for having come this far. You can now proceed to discover the various features, plugins, and themes and proceed setting up your first blog/website!

Reference Websites:

- 1. <https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lamp-on-ubuntu-18-04>**
- 2. <https://www.journaldev.com/24954/install-wordpress-on-ubuntu>**

Conclusion: we are able to create a scenario in wordpress for Social Marketing, Search engine and Sharing Tools.