Semester & Academic Year: 1/ 2021 - 22

Module Name: Fundamentals of Computing

Assessment: CA2 Group Document on Managing Ubuntu on AWS

Class: DCITP/A1/01 CA2 Team No. 2

Team members:

Shushant Shashwat ()

Javier Fok ()

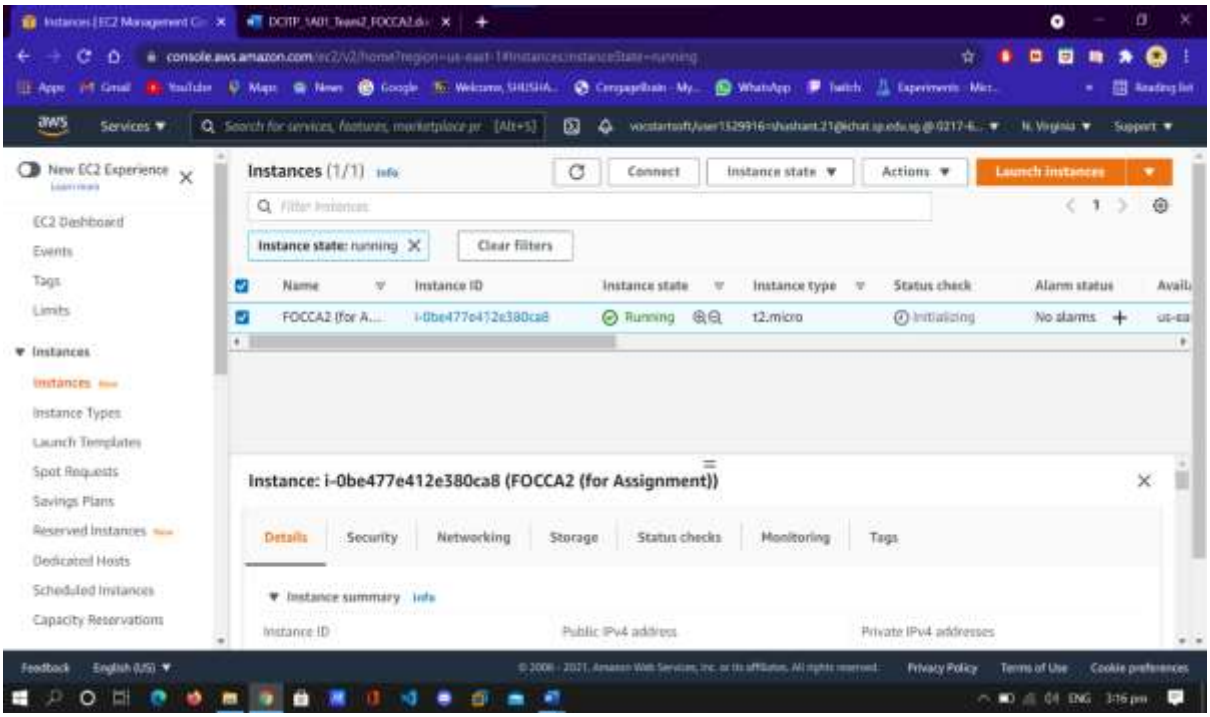
Adeeb ()

Zheng Yao ()

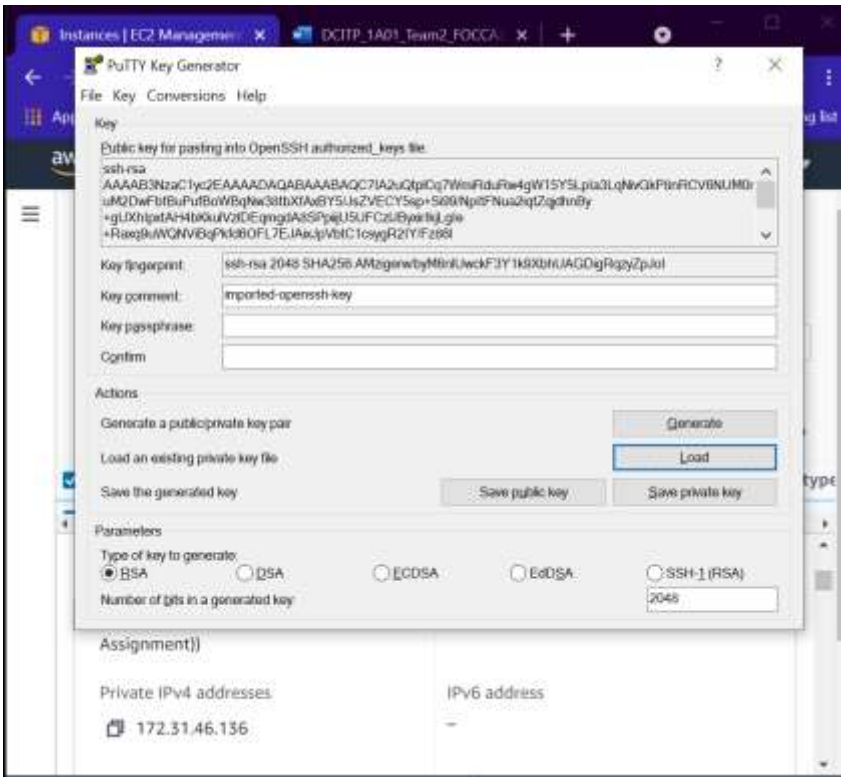
1. Creating a Virtual Machine on AWS Instances and how to access it

In this document, we will be documenting major steps and commands with screenshots on how we install and manage Ubuntu Server and WordPress web site on Ubuntu Linux on AWS.

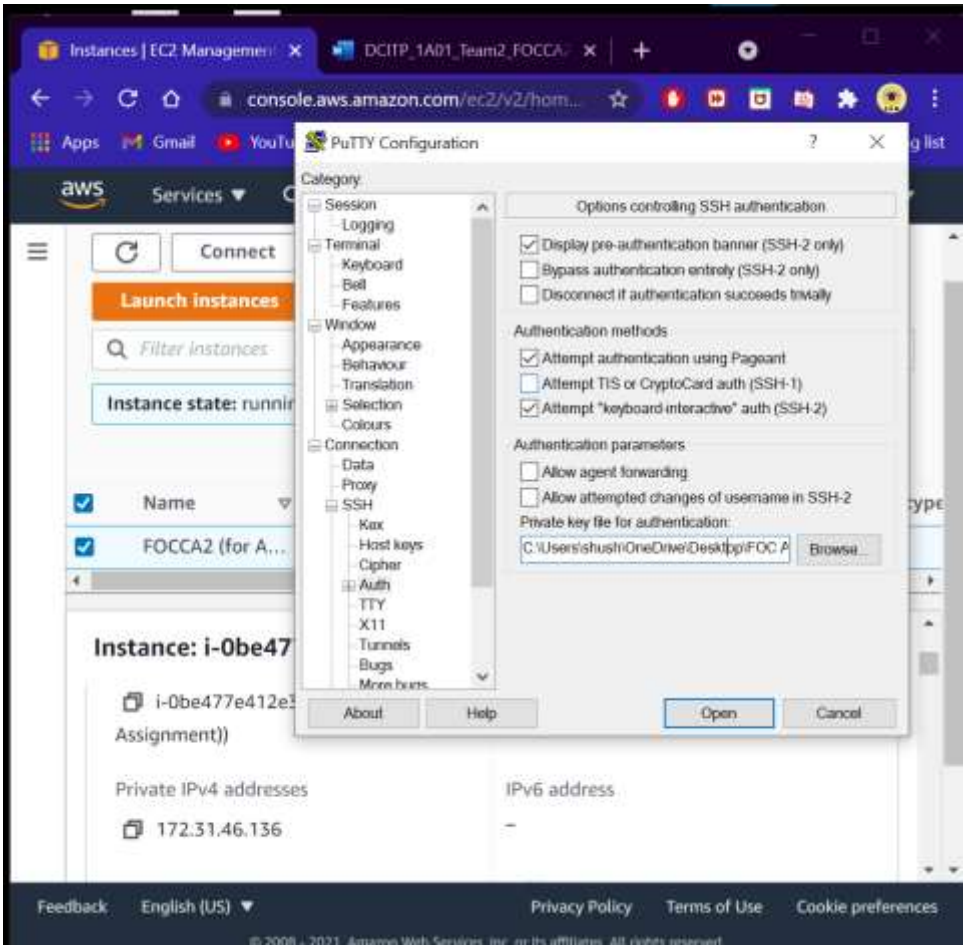
Firstly, we need to launch a virtual machine hosted on the AWS cloud in AWS console. We select Ubuntu Server 18.04 LTS (HVM), SSD Volume Type (64-bit), and create a new keypair to connect to our instance and download the keypair as a .pem file called WPFOC-KEYPAIR. We will name it as FOCCA2 (For Assignment).



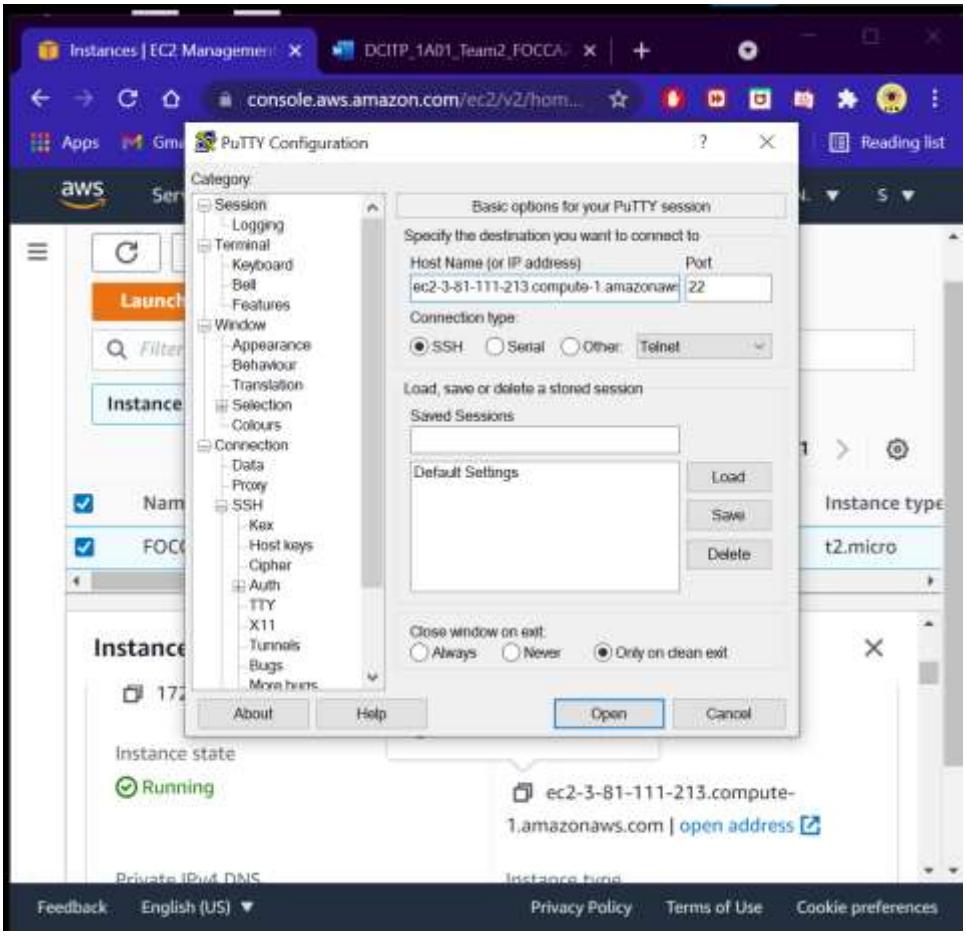
Then we have to open Putty Gen to generate a private key to login to remote instance using putty. In Putty Gen, we will select File then load private key and select the keypair we downloaded.



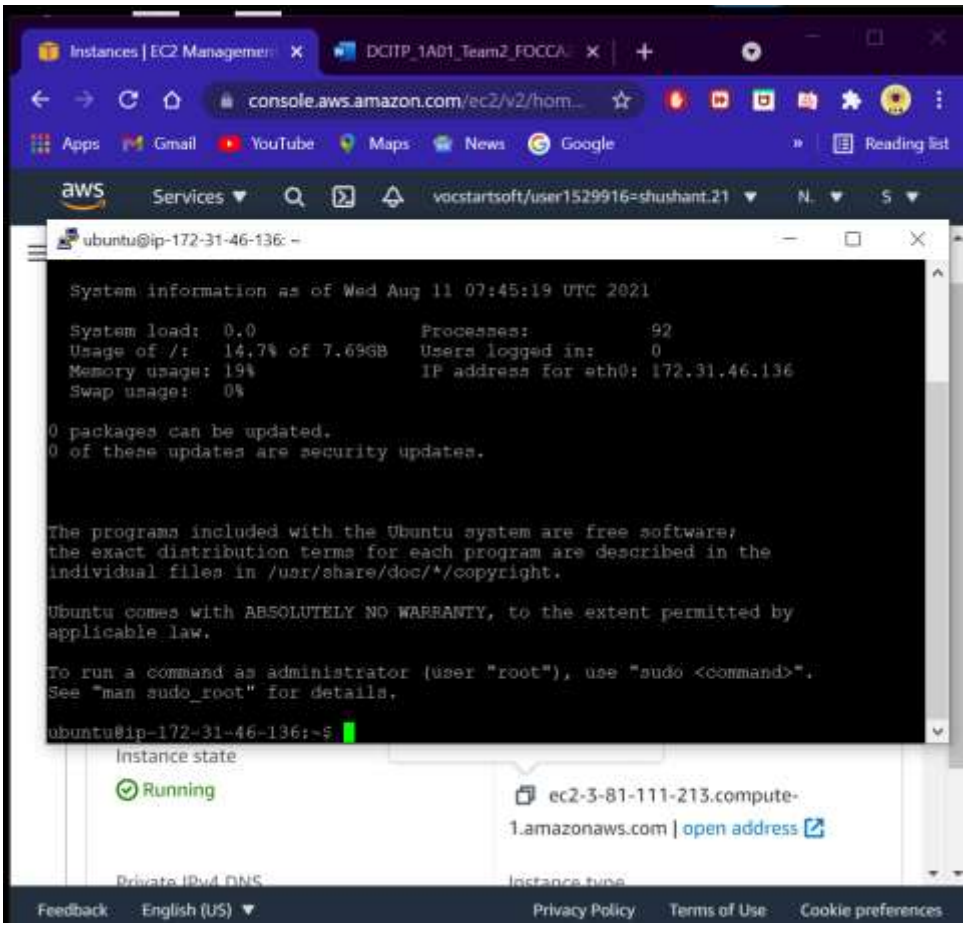
Then we will open Putty itself, go to Connection, SSH, Auth and load our, .ppk file we created in Putty Gen.



We also have to set the hostname field in session with our public DNS name



Finally, we can click open, and it should open! We will login as ‘ubuntu’ to the server.



Now we will set up the Apache Web Server, PHP, and MySQL server on the AWS instance.

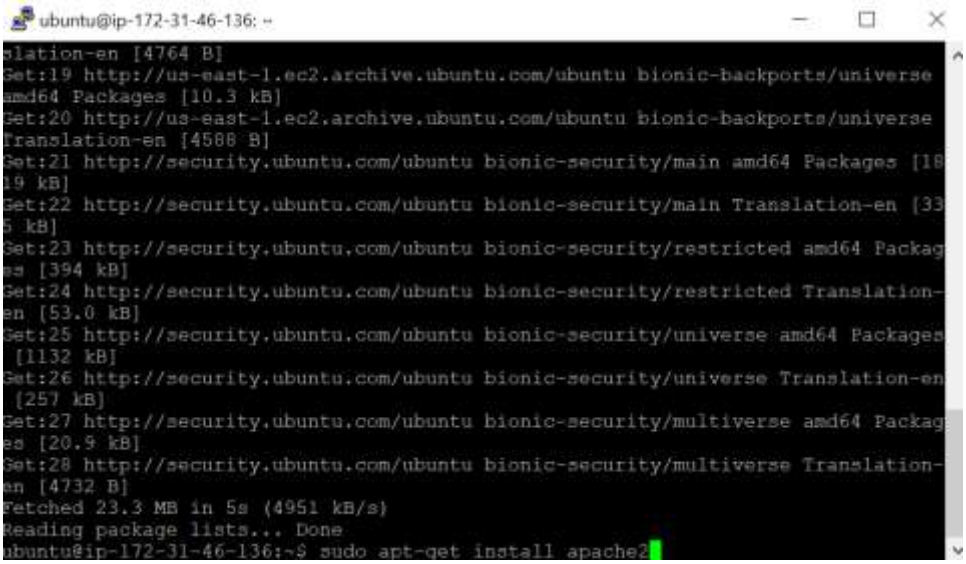
2. Installing Apache

We will install the HTTP server (Web server software that runs web servers) via the command:

sudo apt-get update

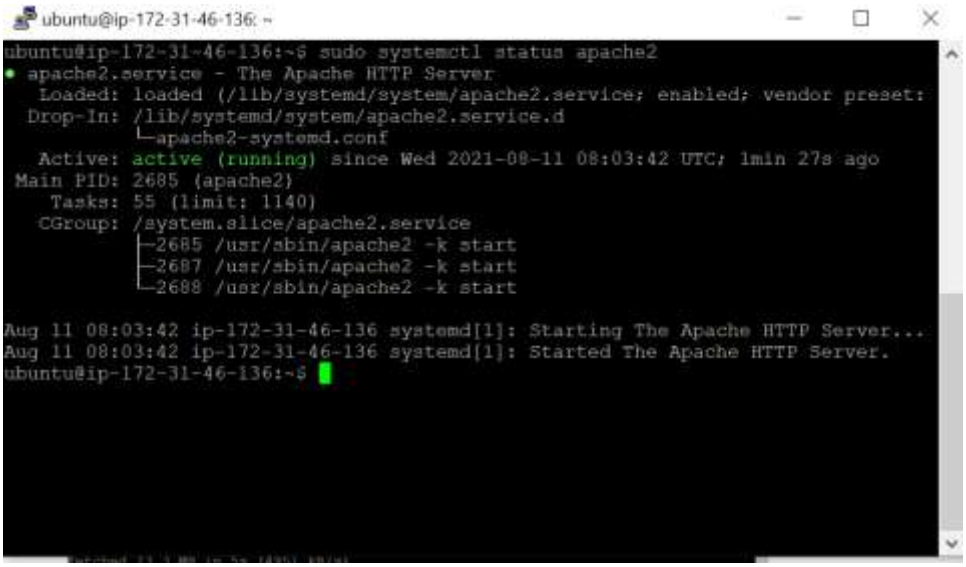
To update our ubuntu to check for any pending updates.

Installing Apache:



sudo apt-get install apache2

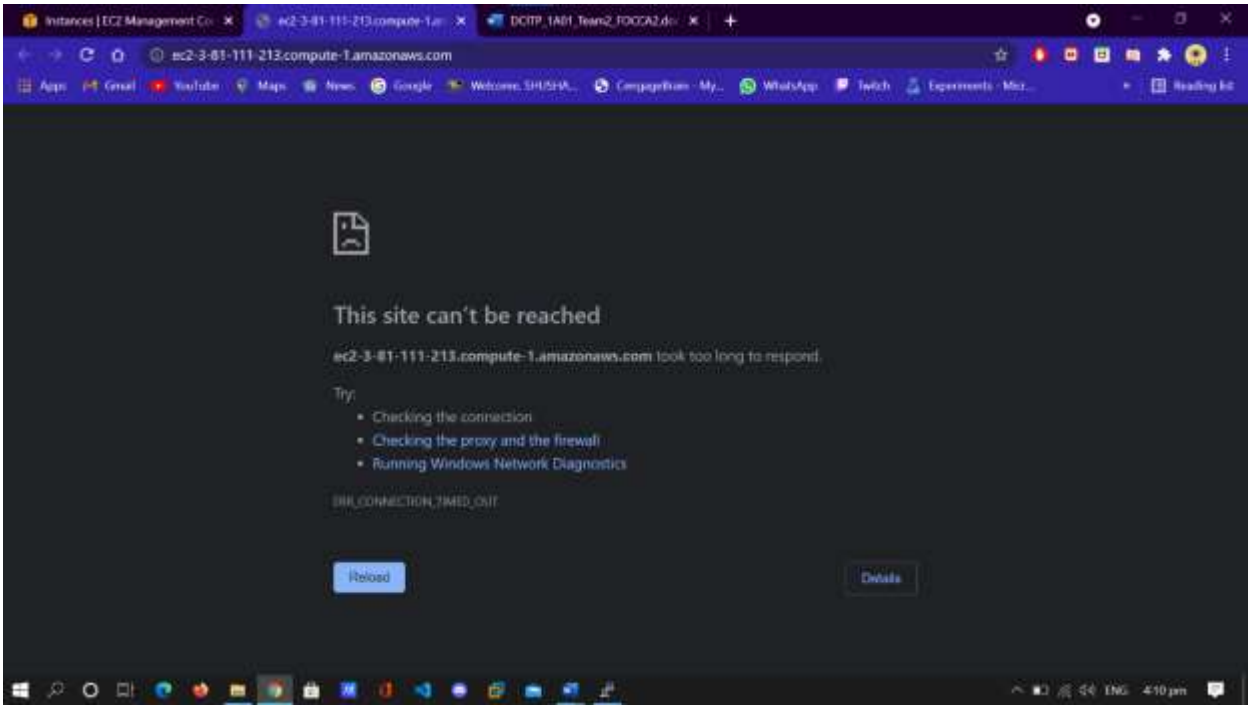
To check if apache2 service is running:



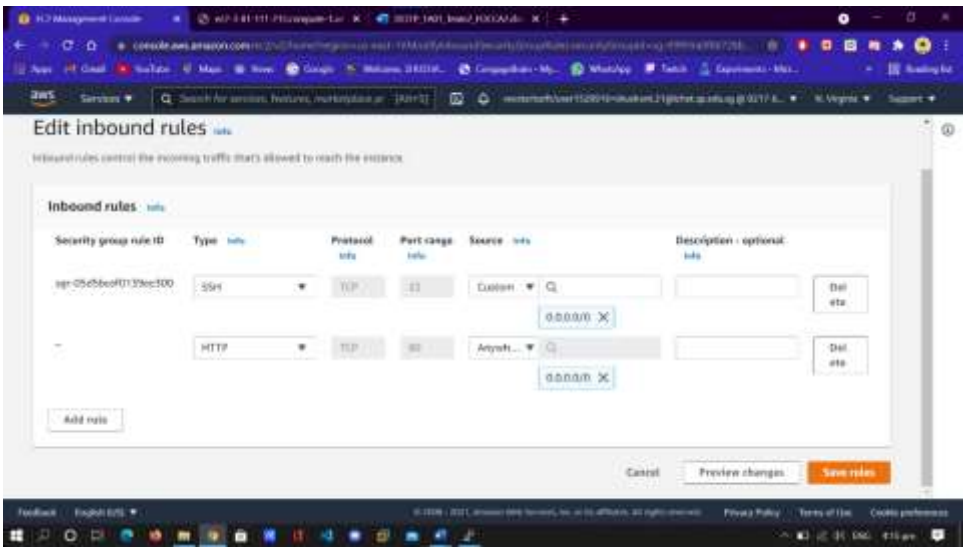
sudo systemctl status apache2

Troubleshooting

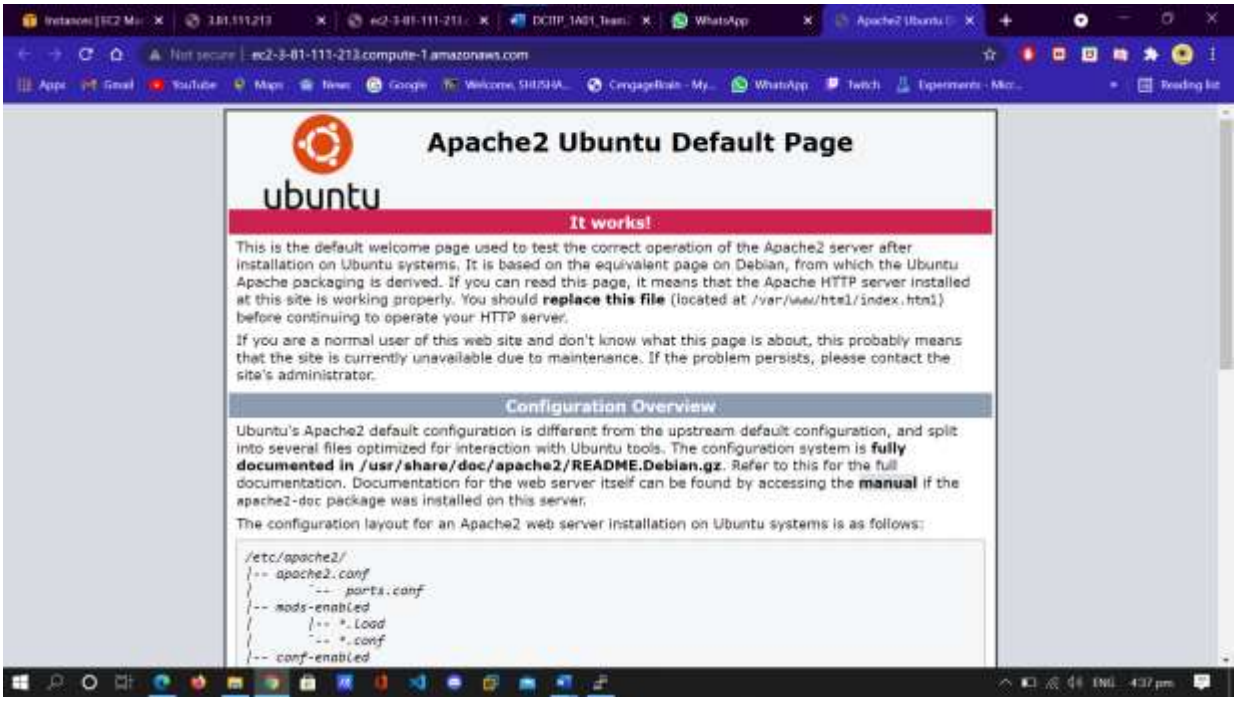
However, we encountered a problem when trying to check out the pages on our public DNS



So, we went to AWS Console to change the security group for the EC2 instance. We added a HTTP port 80 in our inbound rules.



The changes should be saved and then it finally works!



3. Installing PHP

Next, we should also install PHP (Server Scripting Language) which is fully supported and recommended for WordPress.

The command to install PHP and all required extensions:

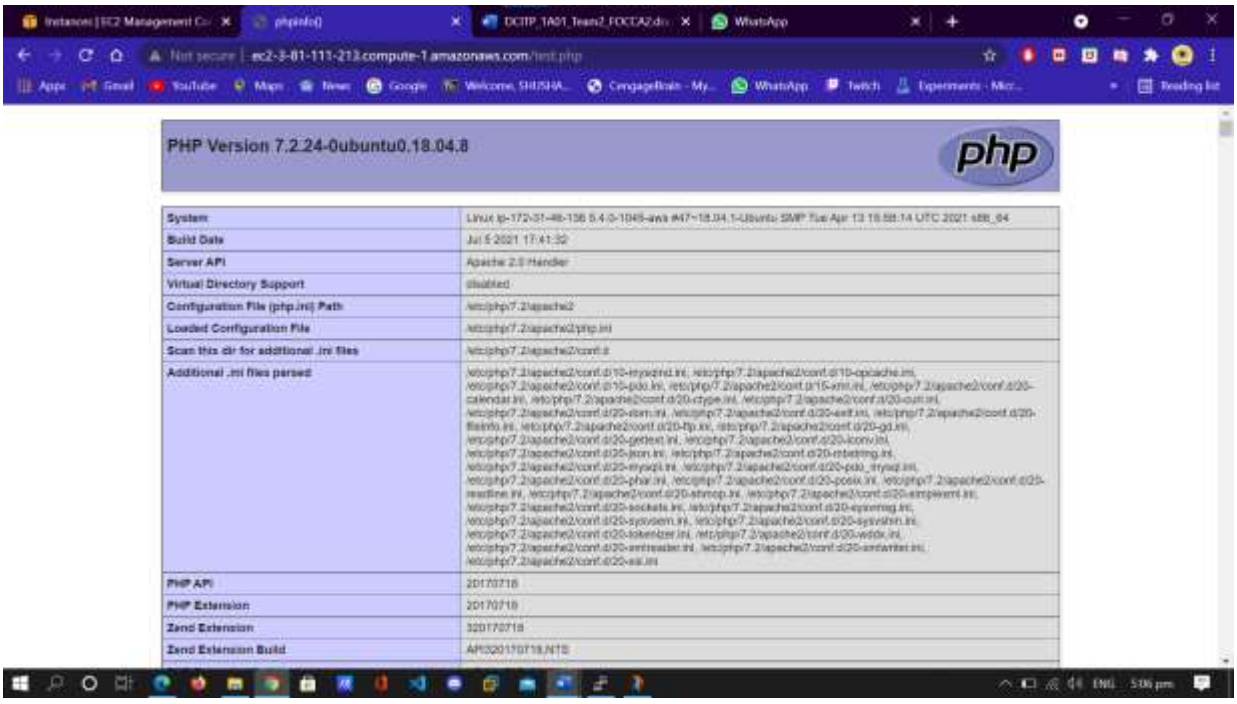
Then we have to restart apache2 so the newly installed PHP extensions will be loaded.

sudo systemctl restart apache2

We will now test if php works by editing and creating a new file called test.php in /var/www/html using sudo nano

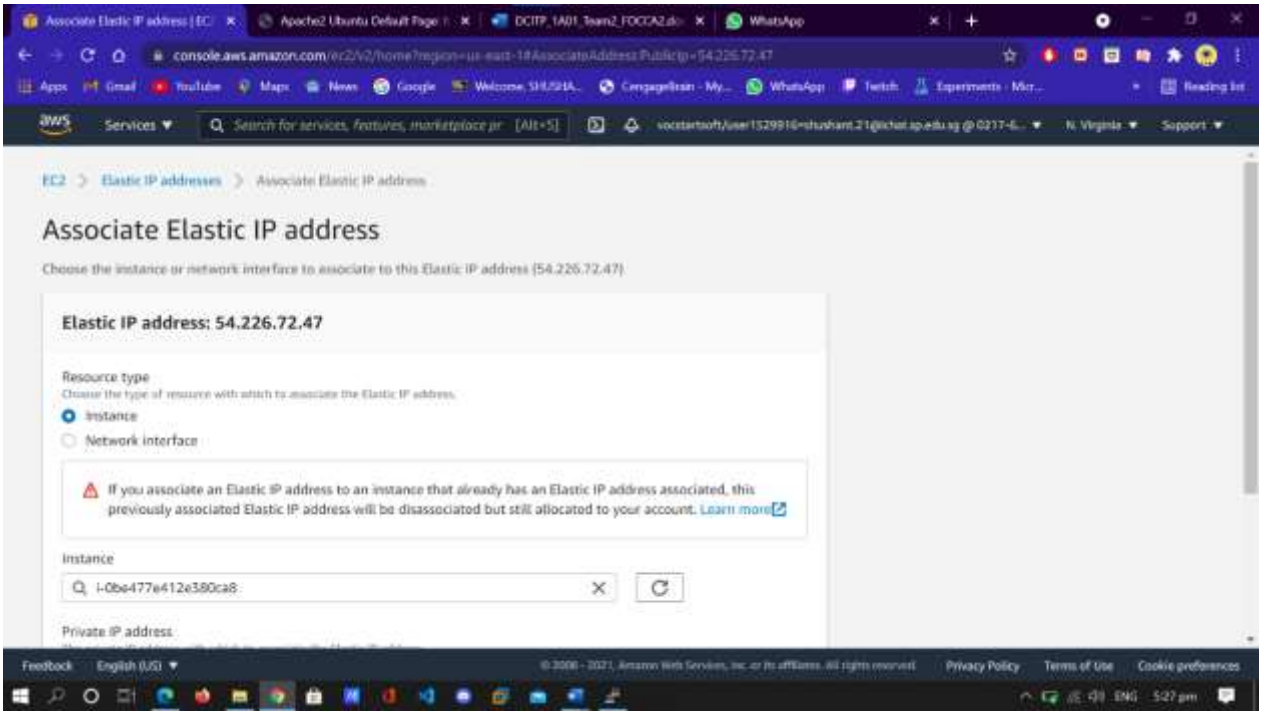
sudo nano /var/www/html/test.php

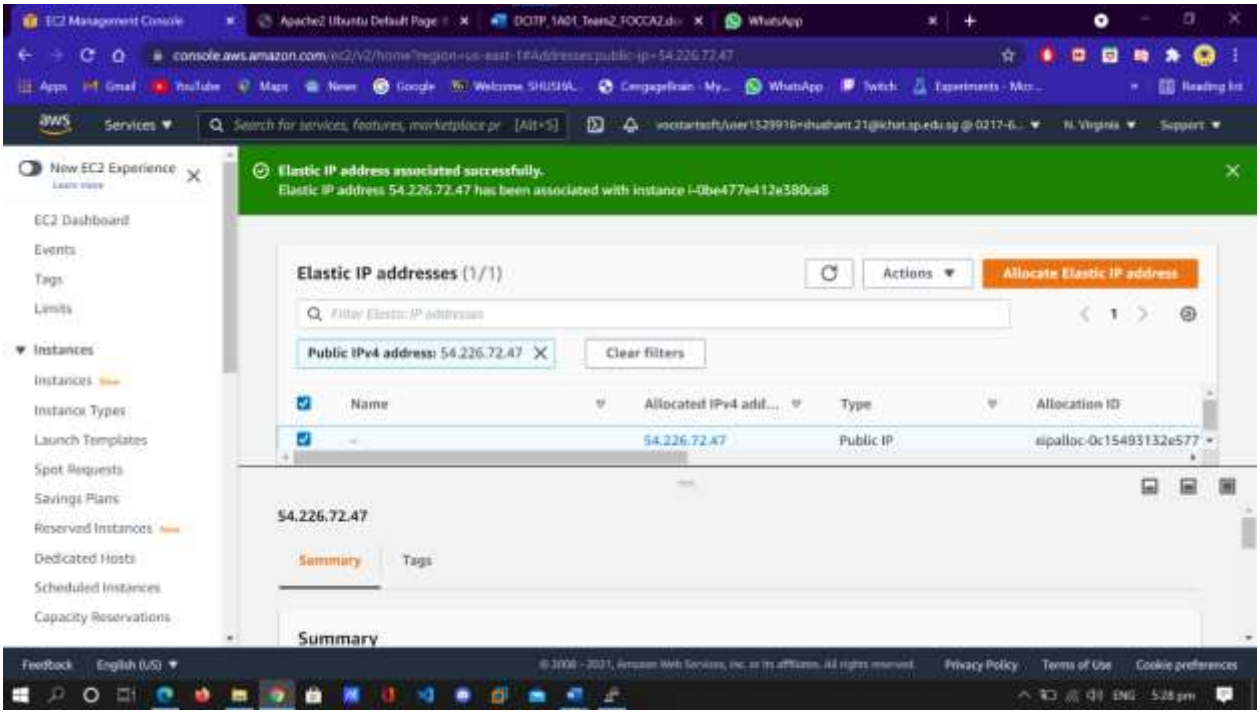
Checking if our web server can correct display content generated by a PHP script.



It works!

Before moving on, we are going to create and get an Elastic IP address for our AWS instance so that we can get a fixed IP address associated with our instance, since the address changes every time.





4. Installing MySQL

MySQL is a database management system, uses a Structured Query Language to manage its data. We will be installing MySQL on Ubuntu using the following command:

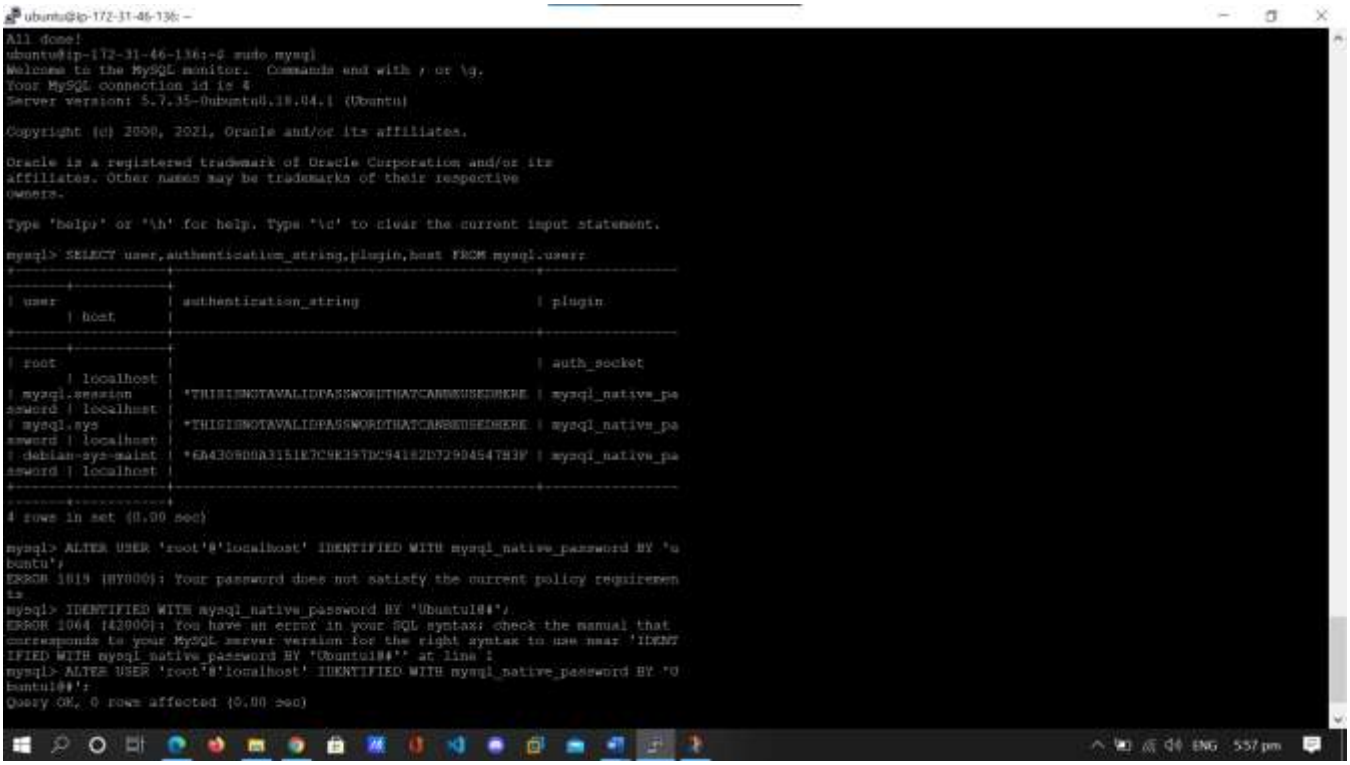
```
ubuntu@ip-172-31-46-136:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
```

Then we will run the mysql with:

sudo mysql

And configure the root account to authenticate with a password, with ALTER USER command.

Then FLUSH PRIVILEGES; to reload grant tables put new changes into effect.



Before moving on to creating users and database for WordPress on MySQL, we need to configure the Apache web server to set the document root for WordPress Web site to /var/www/html/p2123602

Firstly, we need to stop the apache2 service using:

sudo systemctl stop apache2 service

```
Aug 11 09:44:16 ip-172-31-46-136 systemd[1]: Started My
ubuntu@ip-172-31-46-136:~$ sudo systemctl stop apache2
ubuntu@ip-172-31-46-136:~$
```

Then we will make a new directory at /var/www/html called p2123602

```
ubuntu@ip-172-31-46-136:~$ cd /var/www/html
ubuntu@ip-172-31-46-136:/var/www/html$ mkdir p2123602
mkdir: cannot create directory ‘p2123602’: Permission denied
ubuntu@ip-172-31-46-136:/var/www/html$ sudo mkdir p2123602
ubuntu@ip-172-31-46-136:/var/www/html$ ls
index.html  p2123602
ubuntu@ip-172-31-46-136:/var/www/html$
```

After that we shall edit go /etc/apache2 and edit the apache2.conf file and change the directory file.

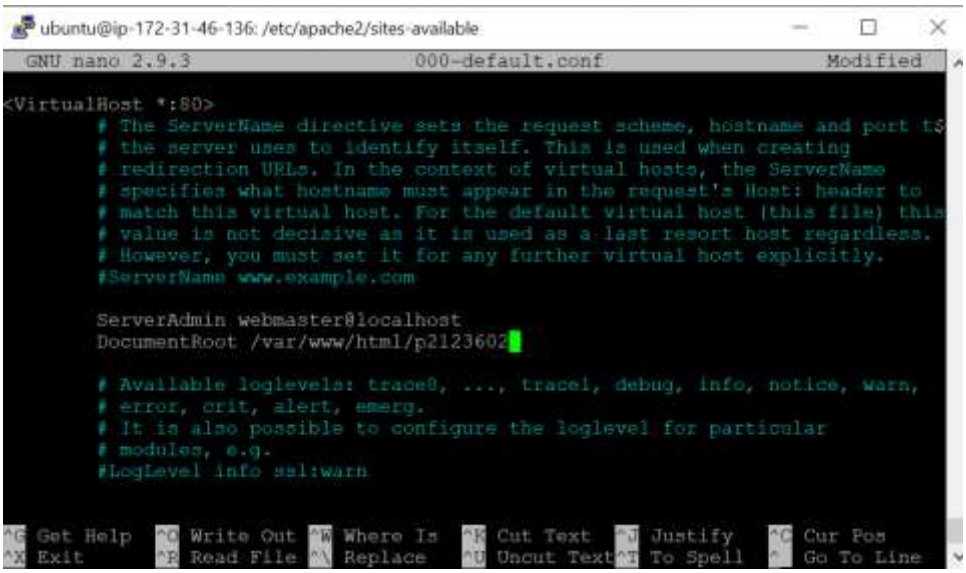
```
ubuntu@ip-172-31-46-136:~$ cd /etc/apache2
ubuntu@ip-172-31-46-136:/etc/apache2$ ls
apache2.conf    conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars       mods-available  ports.conf    sites-enabled
ubuntu@ip-172-31-46-136:/etc/apache2$ sudo nano apache2.conf
```

ubuntu@ip-172-31-46-136: /etc/apache2

```
 GNU nano 2.9.3                    apache2.conf                        Modified

        AllowOverride None
        Require all denied
</Directory>

<Directory /usr/share>
        AllowOverride None
        Require all granted
</Directory>

<Directory /var/www/html/p2123602>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>

#<Directory /srv/>
#       Options Indexes FollowSymLinks
#       AllowOverride None
#       Require all granted

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

We now have to make changes in the 000-default.conf file

```
ubuntu@ip-172-31-46-136:/etc/apache2$ cd /etc/apache2/sites-available
ubuntu@ip-172-31-46-136:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
ubuntu@ip-172-31-46-136:/etc/apache2/sites-available$ sudo nano 000-default.conf
```

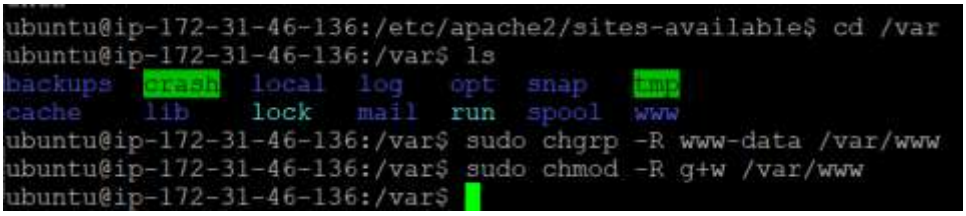


Restart apache2 to take in the changes:

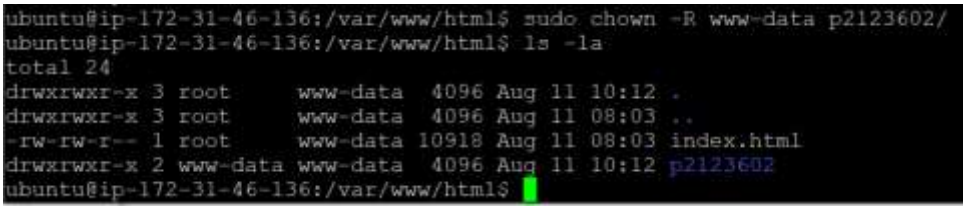
sudo systemctl restart apache2

Now, we have to set proper permissions for www-data system user.

Change the group of /var/www from root to www-data



Change owner of root directory and its files to www-data and check if root directory has www-data as its owner.



Creating a database user in mysql server for PHP to access mysql, server and assign proper rights

We will now enter mysql using the command:

mysql –u root –p

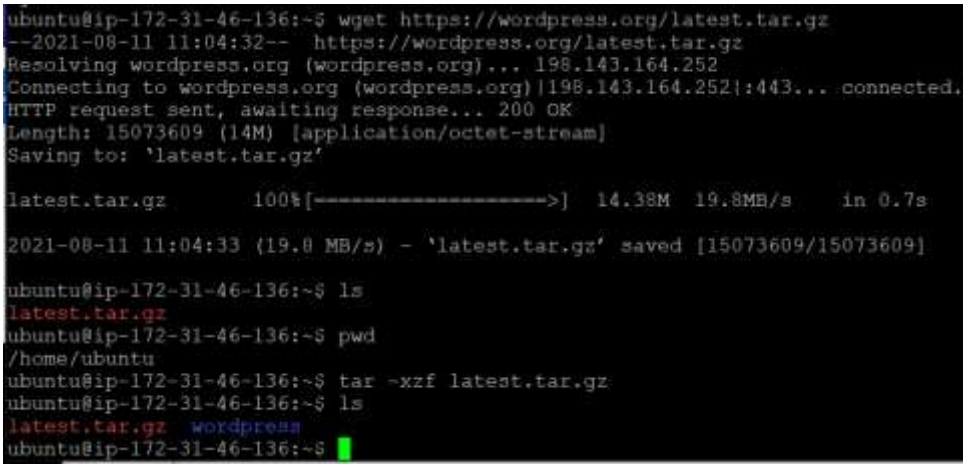
And enter our password. Then, we shall create the user and database in MySQL as shown below.

```
mysql> CREATE USER 'wordpress-user-p2123602'@'localhost' IDENTIFIED BY 'Ubuntu1@
#';
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT user FROM mysql. user;
+--------------------------+
| user                     |
+--------------------------+
| debian-sys-maint         |
| mysql.session            |
| mysql.sys                |
| root                     |
| ubuntu                   |
| wordpress-user-p2123602  |
+--------------------------+
6 rows in set (0.00 sec)

mysql> CREATE DATABASE `wordpress-db-p2123602`;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON `wordpress-db-p2123602`.* TO "wordpress-user-p212
3602"@"localhost";
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

DB_NAME	DB_USER	DB_PASSWORD
Wordpress-db-p2123602	Wordpress-user-p2123602	Ubuntu1@#

5. Installing WordPress

We will now download and unzip the WordPress installation package as shown below:



Where we unzip and unarchive the installation package and the installation folder is unzipped to a folder called WordPress, which located in our home directory.

To create and edit the wp-config.php file

Firstly, we copy the wp-config-sample.php file to another file called wp-config.php. It creates a new configuration file and keeps the original sample file intact as a backup.

```
ubuntu@ip-172-31-46-136:~$ sudo cp wordpress/wp-config-sample.php wordpress/wp-c
onfig.php
ubuntu@ip-172-31-46-136:~$
```

Secondly, we edit the wp-config.php file with nano and enter the values for our installation (such as changing the DB_NAME, DB_USER, DB_PASSWORD)

```
ubuntu@ip-172-31-46-136:~$ sudo nano wordpress/wp-config.php
ubuntu@ip-172-31-46-136:~$
```

ubuntu@ip-172-31-46-136: ~

```
  GNU nano 2.9.3                wordpress/wp-config.php                Modified

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress-db-p2123602' );

/** MySQL database username */
define( 'DB_USER', 'wordpress-user-p2123602' );

/** MySQL database password */
define( 'DB_PASSWORD', 'Ubuntu1@#' );

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

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

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


^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Lastly, we need to copy WordPress to our Apache root directory (/var/www/html/p2123602)

```
ubuntu@ip-172-31-46-136:~$ sudo cp -r wordpress/* /var/www/html/p2123602/
ubuntu@ip-172-31-46-136:~$
```

We are now ready to install WordPress!

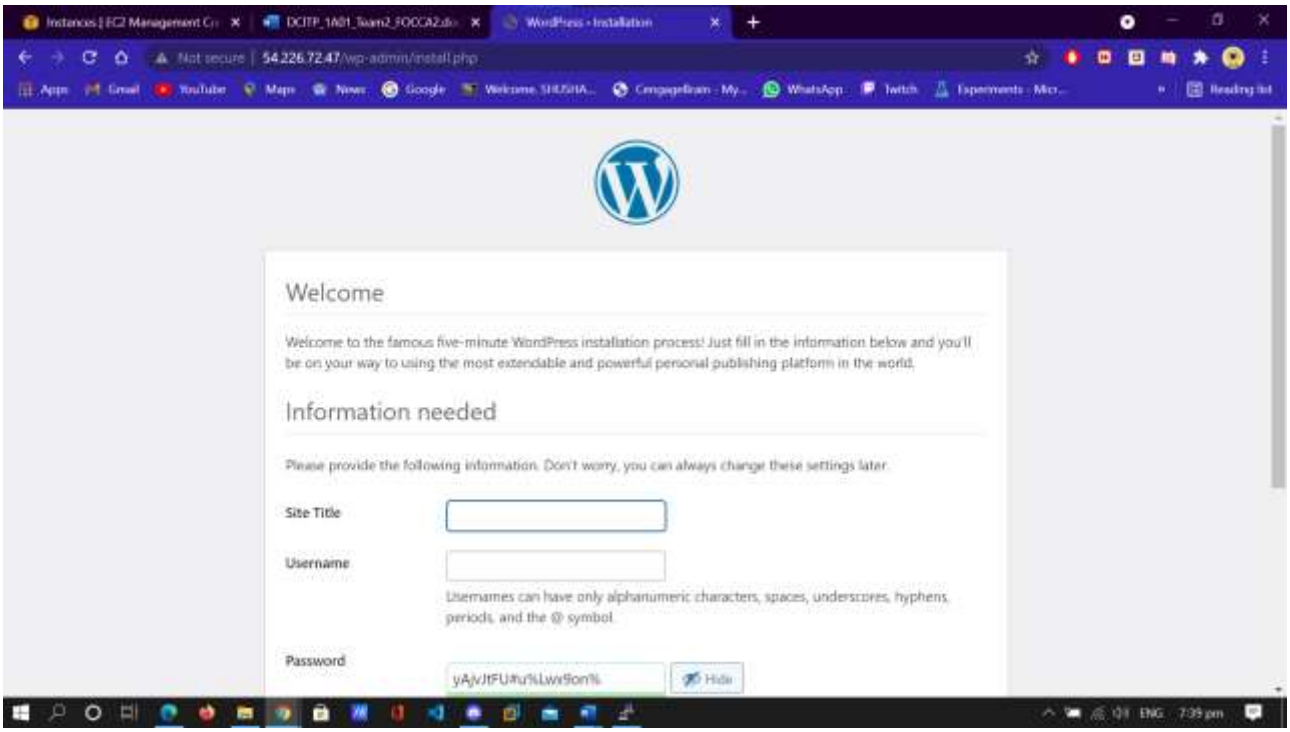
We need to check if Apache and MySQL are running

```
ubuntu@ip-172-31-46-136:~$ sudo systemctl enable apache2 && sudo systemctl enabl
e mysql
Synchronizing state of apache2.service with SysV service script with /lib/system
d/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Synchronizing state of mysql.service with SysV service script with /lib/systemd/
systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mysql
ubuntu@ip-172-31-46-136:~$
```

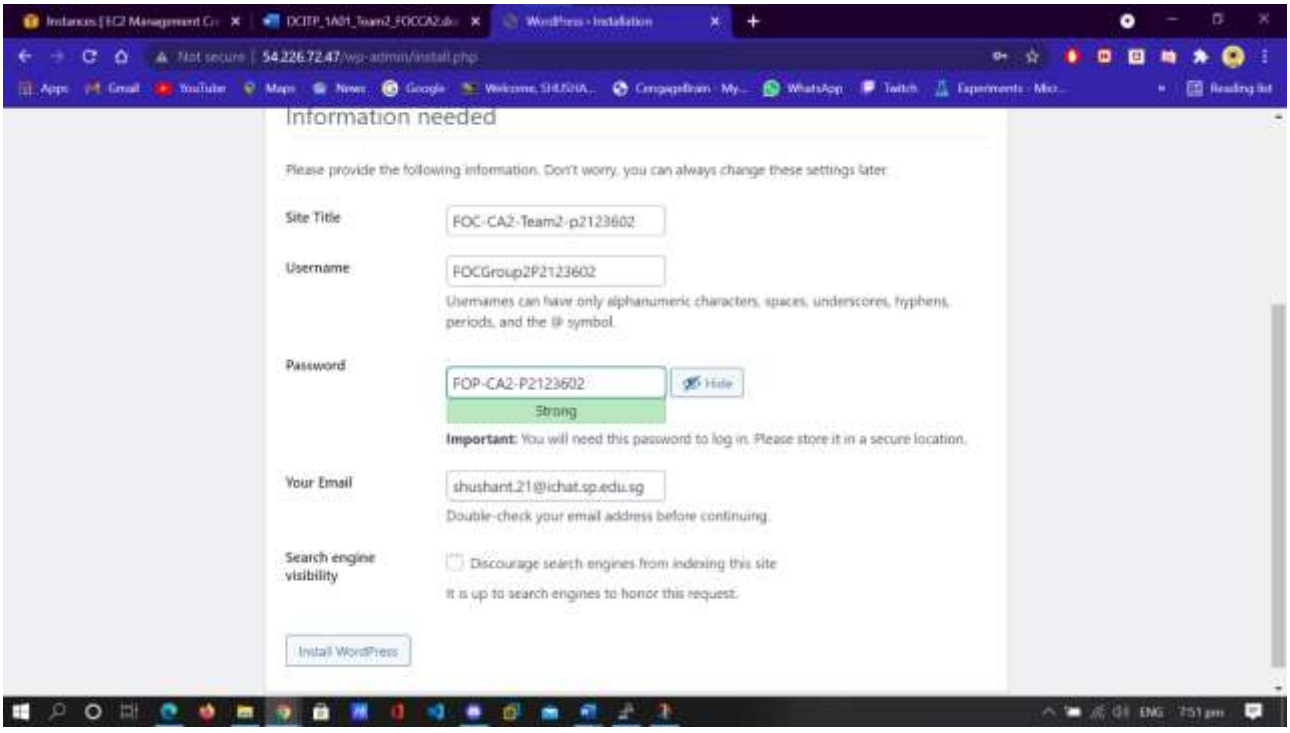
```
ubuntu@ip-172-31-46-136:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Wed 2021-08-11 09:44:16 UTC; 1h 52min ago
 Main PID: 26237 (mysqld)
    Tasks: 29 (limit: 1140)
   CGroup: /system.slice/mysql.service
           └─26237 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pi

Aug 11 09:44:15 ip-172-31-46-136 systemd[1]: Starting MySQL Community Server...
Aug 11 09:44:16 ip-172-31-46-136 systemd[1]: Started MySQL Community Server.
ubuntu@ip-172-31-46-136:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2021-08-11 11:33:03 UTC; 3min 59s ago
 Main PID: 27275 (apache2)
    Tasks: 6 (limit: 1140)
   CGroup: /system.slice/apache2.service
           ├─27275 /usr/sbin/apache2 -k start
           ├─27280 /usr/sbin/apache2 -k start
           ├─27281 /usr/sbin/apache2 -k start
           ├─27282 /usr/sbin/apache2 -k start
           ├─27283 /usr/sbin/apache2 -k start
           └─27284 /usr/sbin/apache2 -k start

Aug 11 11:33:03 ip-172-31-46-136 systemd[1]: Starting The Apache HTTP Server...
Aug 11 11:33:03 ip-172-31-46-136 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-46-136:~$
```

Now that WordPress is downloaded and the server configuration is complete, it is time to finalize the WordPress installation through the web interface.

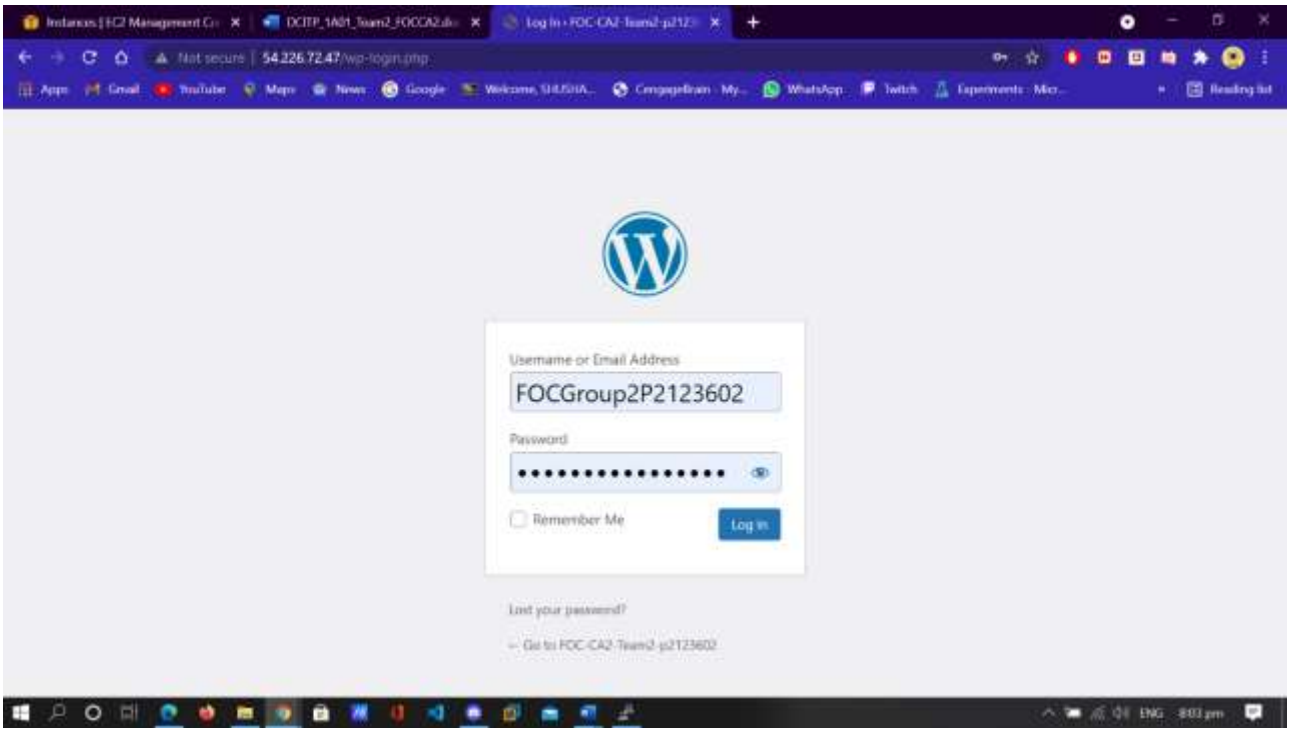
We entered our public IPv4 Address and then the installation screen showed up! Congratulations! We just installed WordPress!

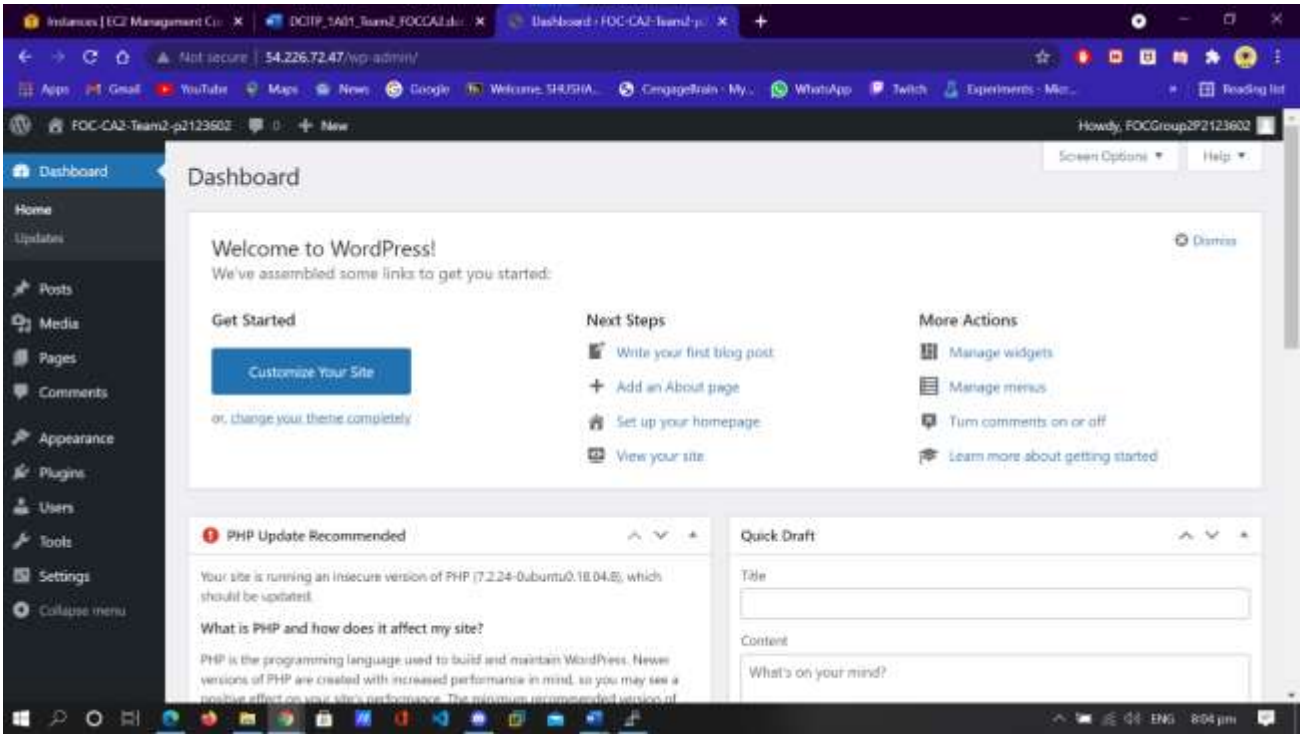


Here we can enter the name for our web site and assign a username and password along with an email address.



Logging into WordPress



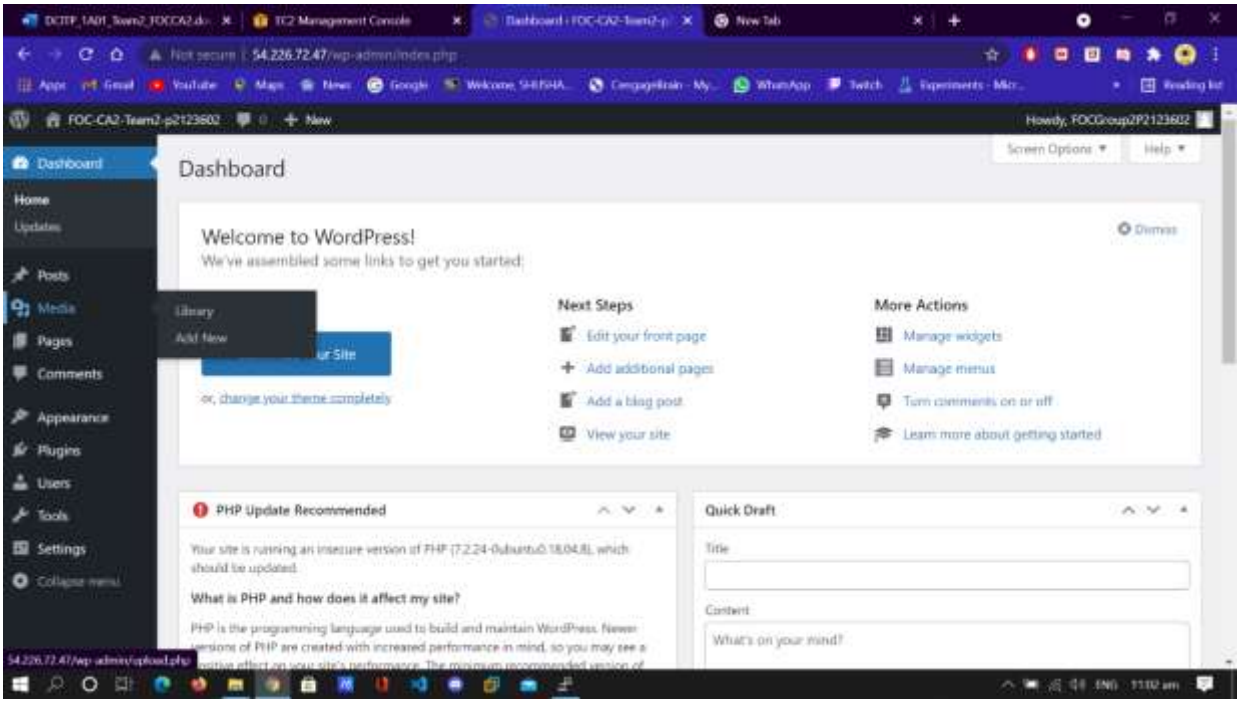


From here we can customize about how we want our webpage to look like and add content.

6. How we add content to our WordPress web site

In this final part, we will be showing how to add content to our website and publishing it. In this example we will be using our practical reflections (pdf files) of Shushant's reflections as an example of how we add content.

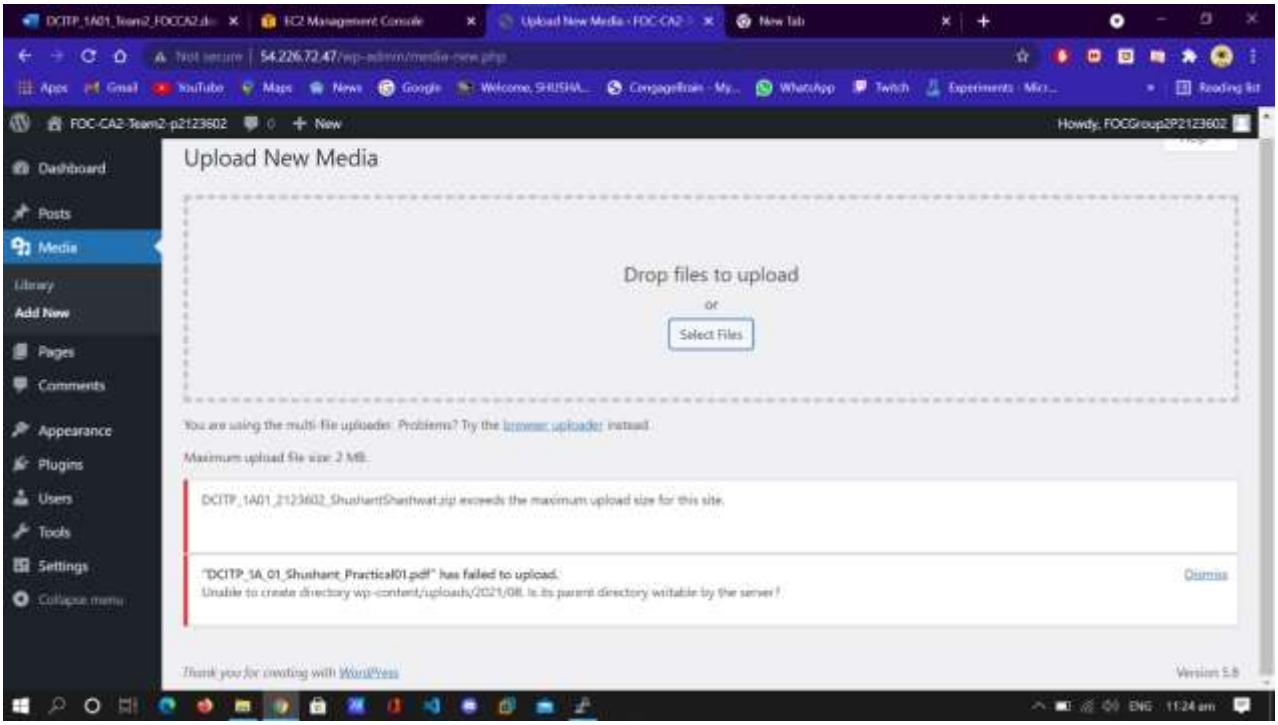
Firstly, we need to add media in order to add our pdf files. On the left side of the site in the menu, click on Media then add new.



After that click on 'Select Files' and add in your desired pdf files. In this case, we will be adding practical reflections for FOC 1,2,3,4,5,6 and 7. (Shushant's practical reflections)

Troubleshooting

However, we faced an issue when uploading our pdf files



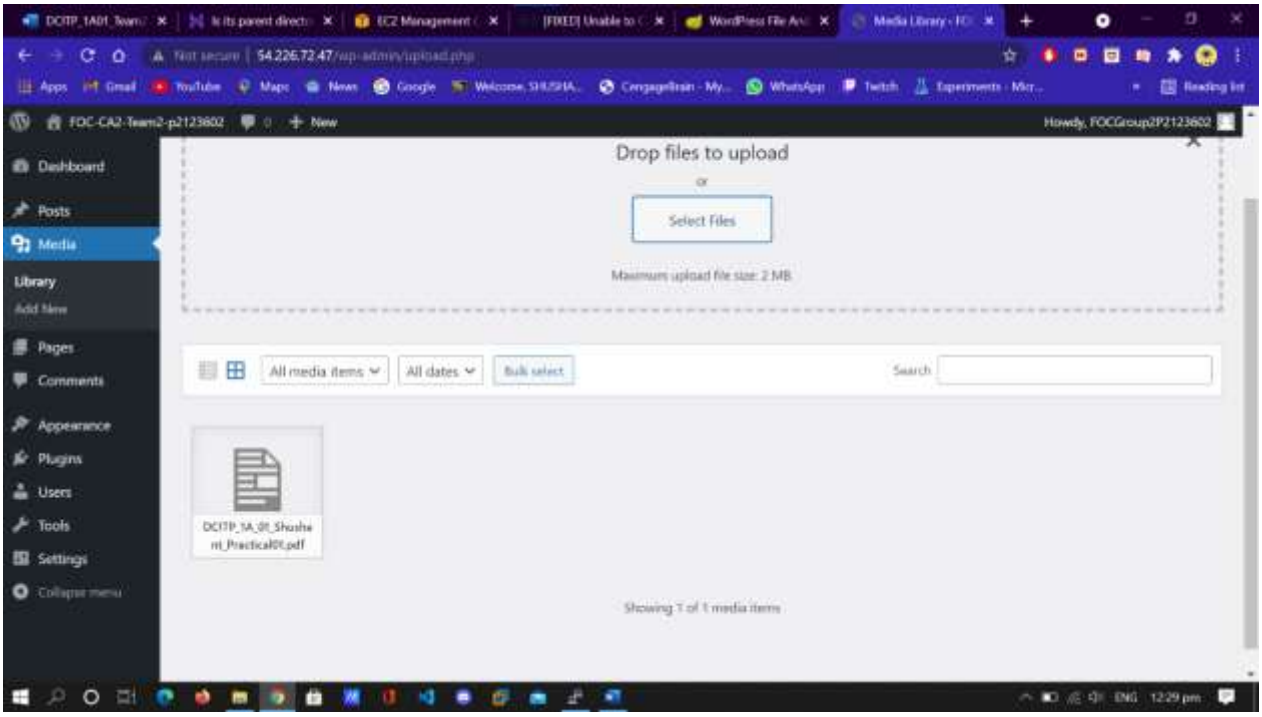
We are unable to add our pdf files as there is a possibility that the www-data does not have access to the uploads folder, hence we need to fix our file permissions.

To fix file permissions

```
ubuntu@ip-172-31-46-136:~$ find /var/www -type d -exec sudo chmod 2775 {} \;
ubuntu@ip-172-31-46-136:~$ find /var/www -type f -exec sudo chmod 0664 {} \;
ubuntu@ip-172-31-46-136:~$ sudo systemctl restart apache2
ubuntu@ip-172-31-46-136:~$
```

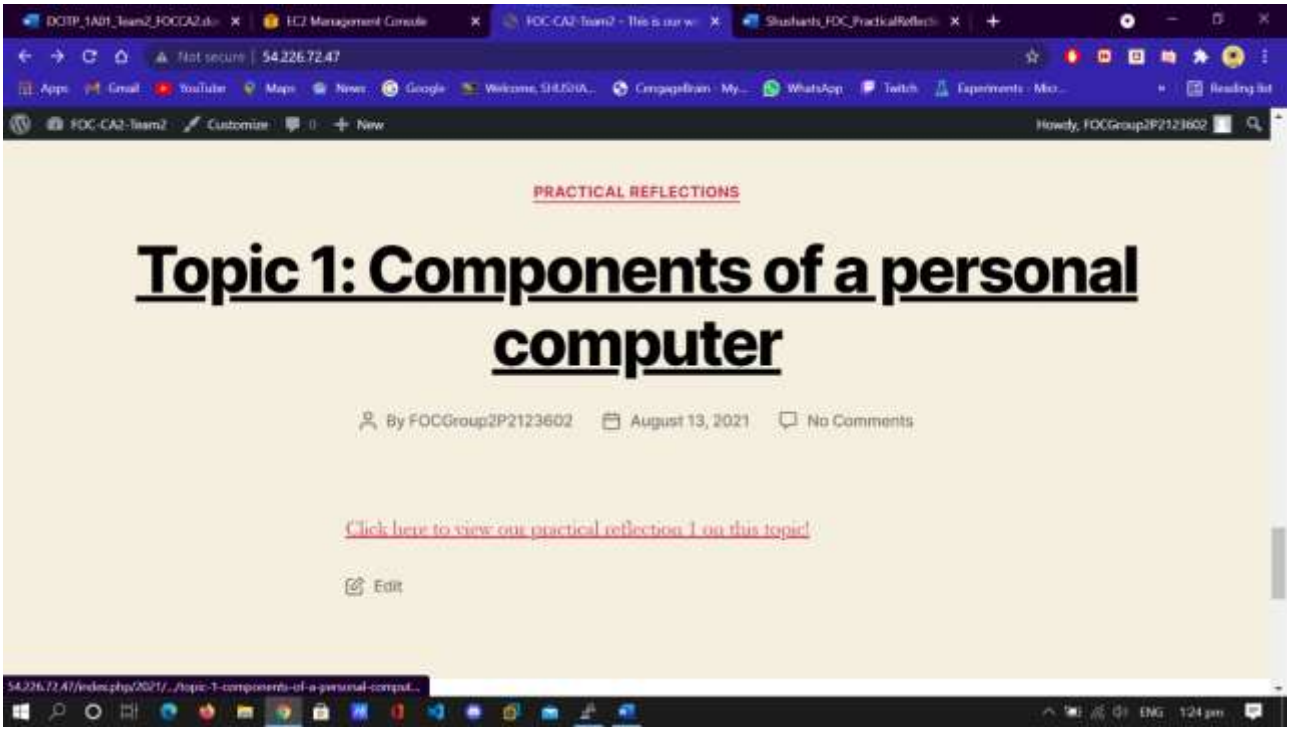
```
ubuntu@ip-172-31-46-136:/var/www/html/p2123602$ sudo chown -R www-data /var/www
ubuntu@ip-172-31-46-136:/var/www/html/p2123602$ sudo nano wp-config.php
ubuntu@ip-172-31-46-136:/var/www/html/p2123602$ sudo chgrp -R www-data /var/www
ubuntu@ip-172-31-46-136:/var/www/html/p2123602$ ls -l /var/www
total 4
drwxrwsr-x 3 www-data www-data 4096 Aug 11 10:12 html
ubuntu@ip-172-31-46-136:/var/www/html/p2123602$
```

And now we can upload our pdf files!



It finally uploads!

Then we go to Posts, add new post and then put our title and link to our pdf



And that is how we manage on how we install and manage Ubuntu Server and WordPress web site on Ubuntu Linux on AWS!