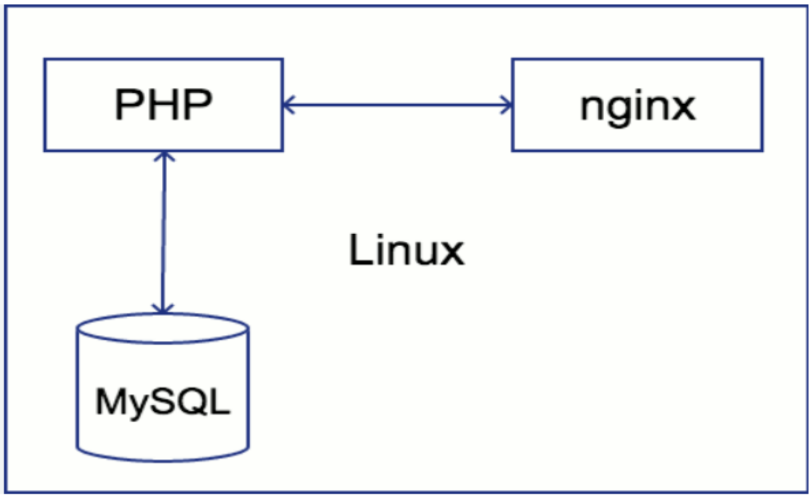
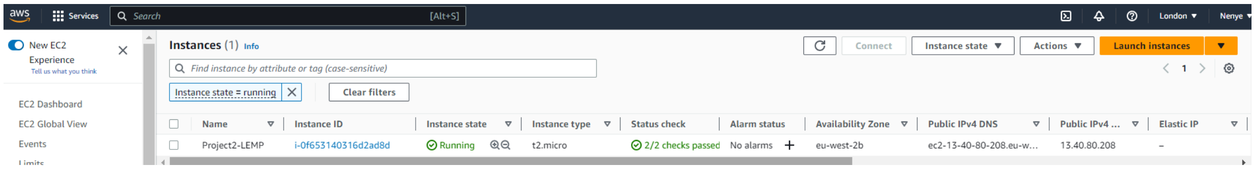
PROJECT 2: LEMP STACK IMPLEMENTATION

LEMP is an open-source web application stack that combines Linux as the Operating system, Nginx (pronounced as engine-x) as the web server, MySQL as the database, and PHP scripting language for the development of web applications. Nginx serves as the LEMP stack's web server, listening for HTTP requests and routing them to the relevant PHP scripts. The user receives a response from the PHP script via Nginx after it has been generated while the website's data is stored and managed using MySQL. An illustration of a LEMP stack is shown in the diagram below.



Launch an EC2 Instance

A step-by-step process of how to do this has been documented in Project 1. My instance is running and will now need to connect to an ssh client.



Instances (1) Info

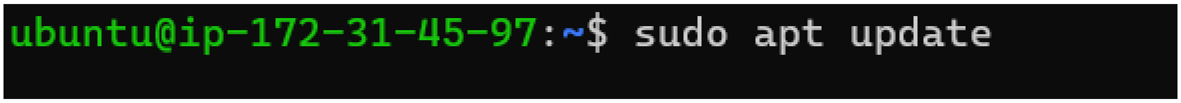
Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Project2-LEMP	i-0f653140316d2ad8d	Running	t2.micro	2/2 checks passed	No alarms	eu-west-2b	ec2-13-40-80-208.eu-w...	13.40.80.208	–

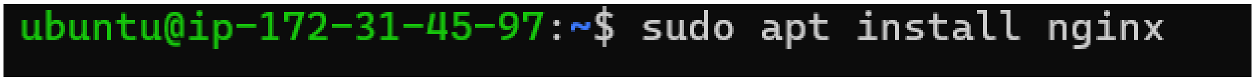
Installing Nginx

Nginx is a general TCP proxy server, which also functions as an HTTP and reverse proxy server. It is a high-performance web server developed to facilitate the increasing needs of the modern web.

To commence the installation of the Nginx, I started off by updating the Ubuntu package followed by the apt install command and then typed y when prompted. Alternatively, you can run the command as sudo apt install nginx -y.

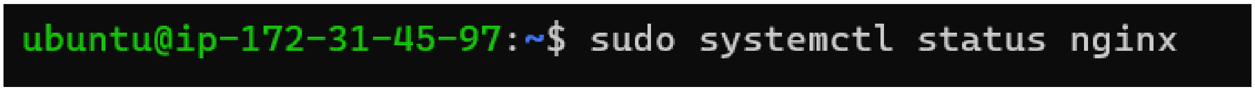


```
ubuntu@ip-172-31-45-97:~$ sudo apt update
```



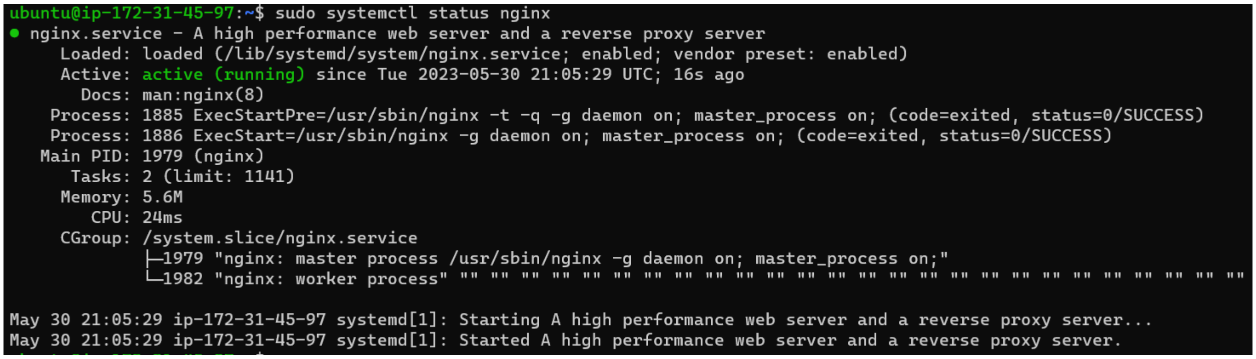
```
ubuntu@ip-172-31-45-97:~$ sudo apt install nginx
```

To refresh the service to confirm that this is running, I used



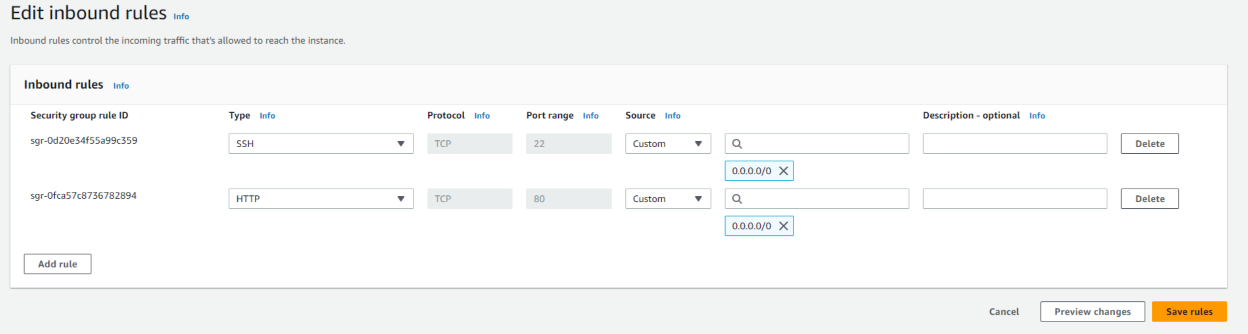
```
ubuntu@ip-172-31-45-97:~$ sudo systemctl status nginx
```

The output should appear as running which shows that the Nginx service is active as below.



```
ubuntu@ip-172-31-45-97:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-05-30 21:05:29 UTC; 16s ago
       Docs: man:nginx(8)
    Process: 1885 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 1886 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 1979 (nginx)
      Tasks: 2 (limit: 1141)
     Memory: 5.6M
        CPU: 24ms
     CGroup: /system.slice/nginx.service
             ├─1979 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─1982 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

May 30 21:05:29 ip-172-31-45-97 systemd[1]: Starting A high performance web server and a reverse proxy server...
May 30 21:05:29 ip-172-31-45-97 systemd[1]: Started A high performance web server and a reverse proxy server.
```

The next step now will be to open our TCP Port 80 on our EC2 configuration to be able to receive traffic on our web server as unencrypted web pages are sent and received on this network port by default. Refer to Project 1 for steps to achieve this.



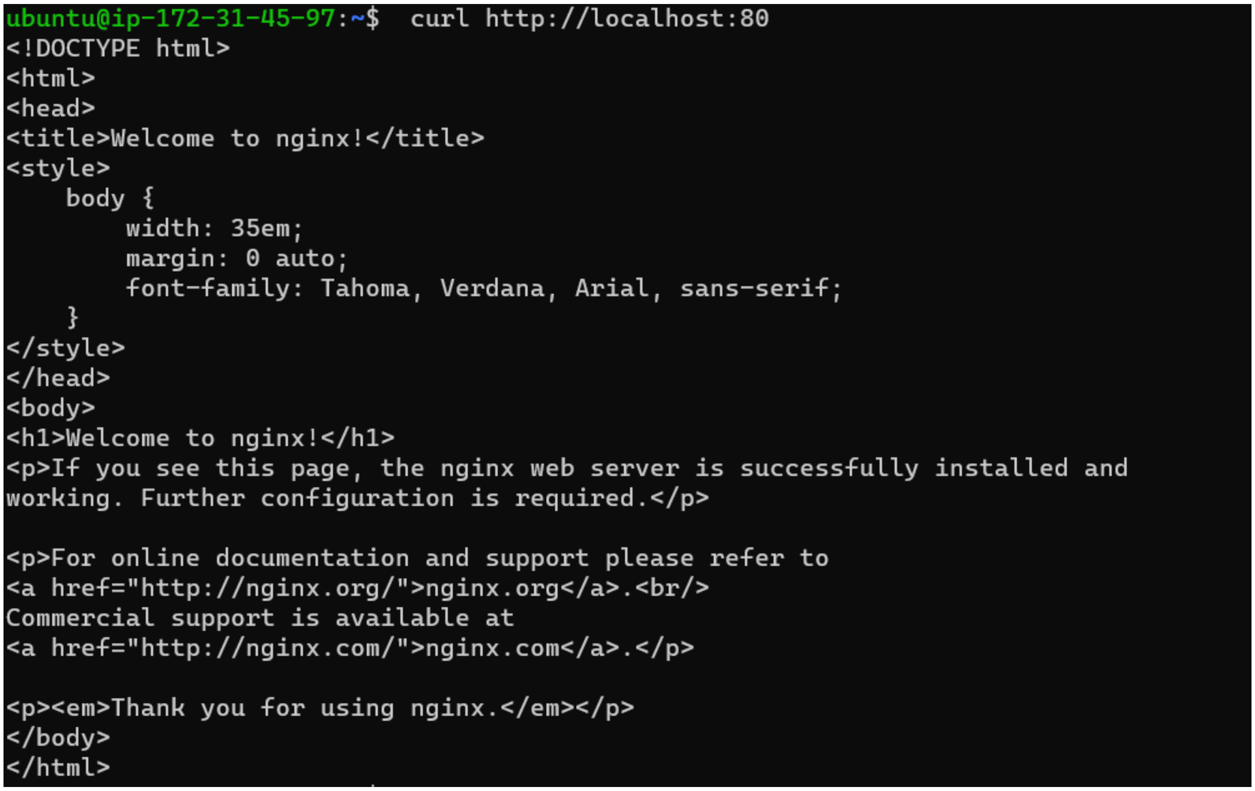
Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

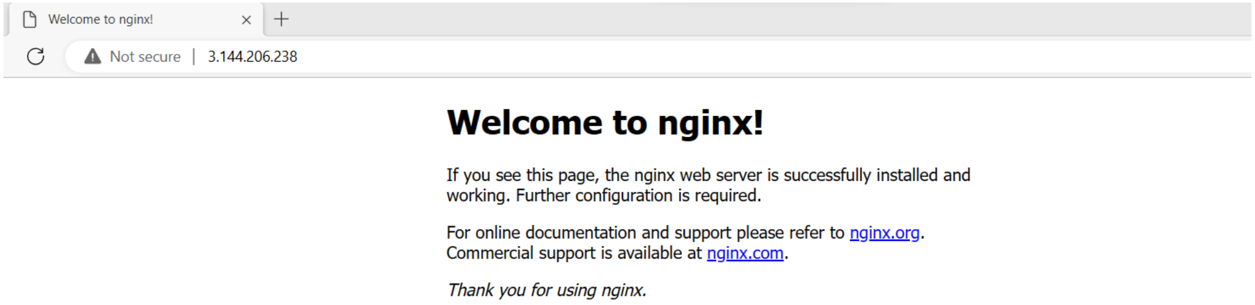
Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-0d20e34f55a99c359	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0fca57c8736782894	HTTP	TCP	80	Custom	0.0.0.0/0		Delete

I will now need to run the curl command to see if I can access the localhost remotely in Ubuntu and then the content should be something around 'Welcome to Nginx which shows the connection is successful.



```
ubuntu@ip-172-31-45-97:~$  curl http://localhost:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

The same content will now be displayed when I use the public IP address to confirm this on our browser e.g. http://3.144.206.238:80. The content of the page should appear similar to what I have on the terminal.



Welcome to nginx!

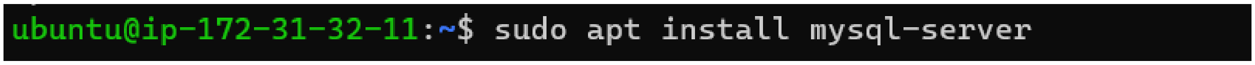
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

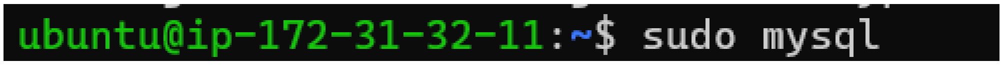
Installing MySQL

An open-source relational database management system is called MySQL. To install this, I ran the sudo apt install command. As always type y for yes when prompted during the installation.



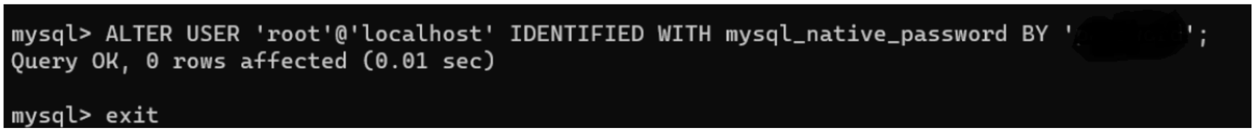
```
ubuntu@ip-172-31-32-11:~$ sudo apt install mysql-server
```

To connect to the MySQL console run



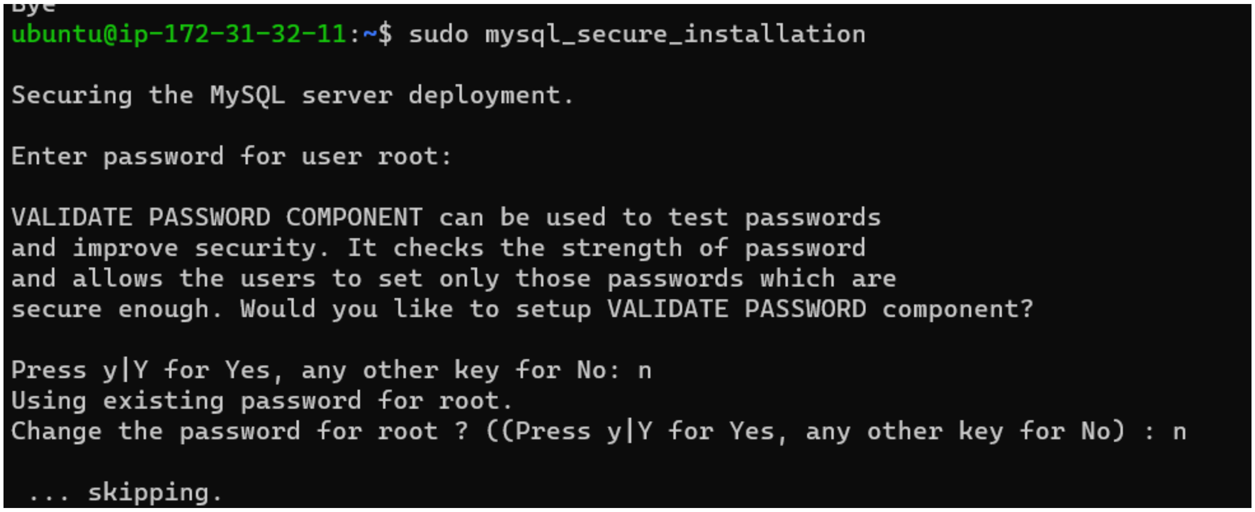
```
ubuntu@ip-172-31-32-11:~$ sudo mysql
```

To guarantee that our database is protected using mysql_native_password as the default authentication mechanism, I will need to set up a password for our root user.



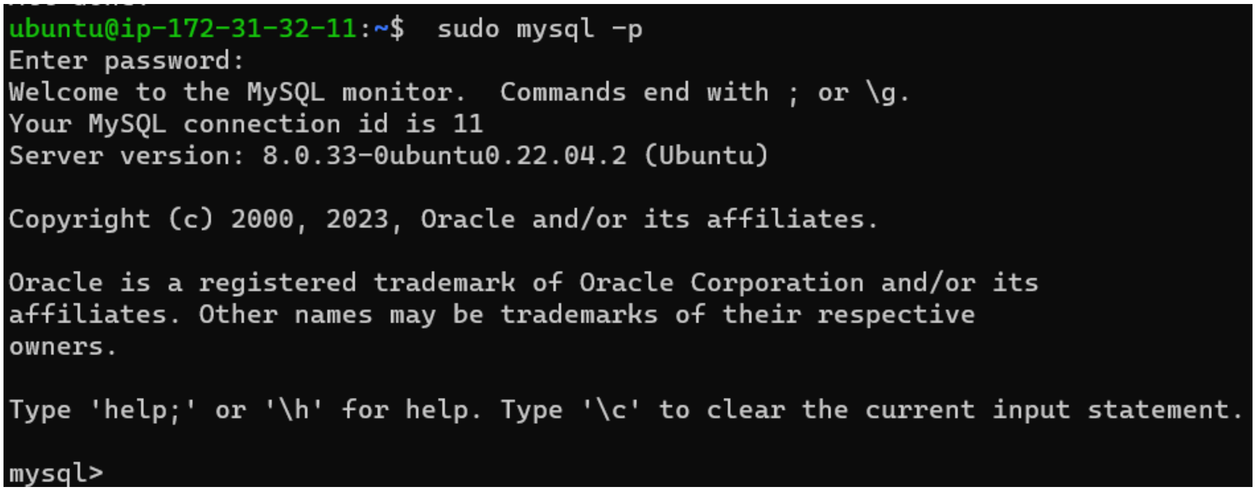
```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '[redacted]';
Query OK, 0 rows affected (0.01 sec)

mysql> exit
```

Now exit the MySQL environment and run the interactive command to validate our password component (type y and n where applicable).



```
Bye
ubuntu@ip-172-31-32-11:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: n
Using existing password for root.
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n

 ... skipping.
```

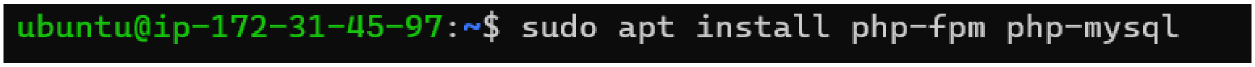
To validate that you are able to log in to the MySQL console type the below command and put in your password for the root user once prompted. This concludes the process for MySQL installation



```
ubuntu@ip-172-31-32-11:~$  sudo mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.33-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

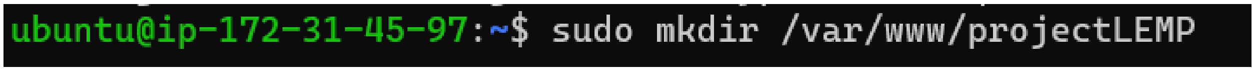
Installing PHP

PHP is a commonly used open-source scripting language mainly used for web development**.** To install this, we will need to run all 2 packages at once and type y to confirm installation when required.

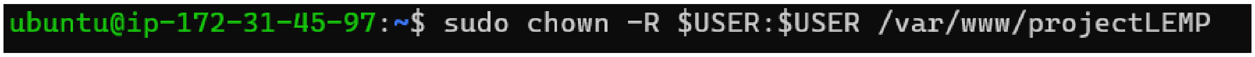


```
ubuntu@ip-172-31-45-97:~$ sudo apt install php-fpm php-mysql
```

Now it's time to set up a domain, I will use projectLEMP as the domain name. I will create a directory and then assign ownership to it with the following commands.

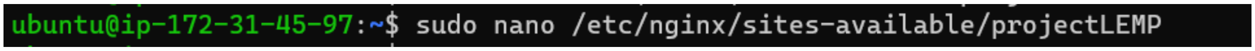


```
ubuntu@ip-172-31-45-97:~$ sudo mkdir /var/www/projectLEMP
```



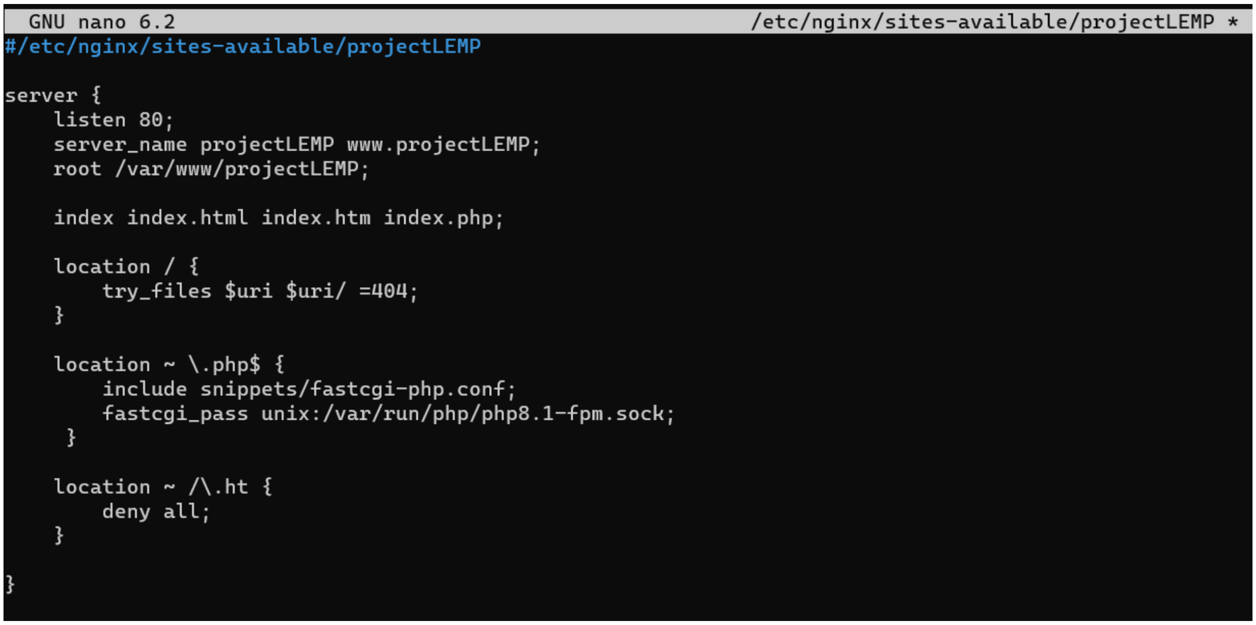
```
ubuntu@ip-172-31-45-97:~$ sudo chown -R $USER:$USER /var/www/projectLEMP
```

I ran the nano command to open a new configuration file in Nginx's sites-available directory.



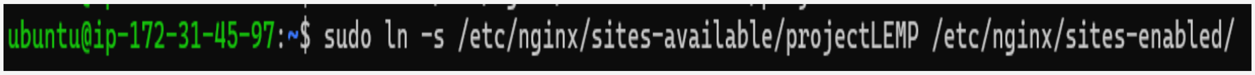
```
ubuntu@ip-172-31-45-97:~$ sudo nano /etc/nginx/sites-available/projectLEMP
```

Following that, I pasted in the following basic configuration and then saved the changes. To save in the nano editor, use CTRL+X and then y and click on ENTER to confirm.



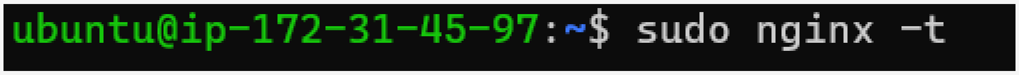
```
  GNU nano 6.2                                        /etc/nginx/sites-available/projectLEMP *
#/etc/nginx/sites-available/projectLEMP

server {
    listen 80;
    server_name projectLEMP www.projectLEMP;
    root /var/www/projectLEMP;

    index index.html index.htm index.php;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
     }

    location ~ /\.ht {
        deny all;
    }

}
```

To activate the above configuration, I will link this to the Nginx config file: sites-enabled.



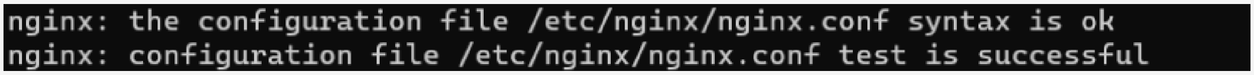
```
ubuntu@ip-172-31-45-97:~$ sudo ln -s /etc/nginx/sites-available/projectLEMP /etc/nginx/sites-enabled/
```

This will tell Nginx to use the configuration the next time it is reloaded. I then tested the configuration for syntax errors by typing:



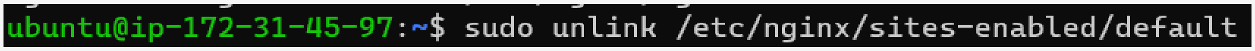
```
ubuntu@ip-172-31-45-97:~$ sudo nginx -t
```

The result should appear as below to show that syntax is ok and the test is successful.



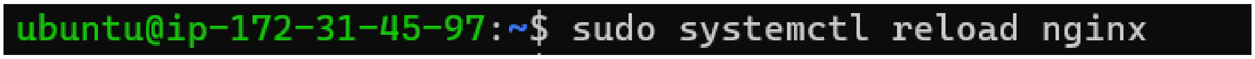
```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

I will need to disable the default Nginx host so that it can listen on Port 80 by running.



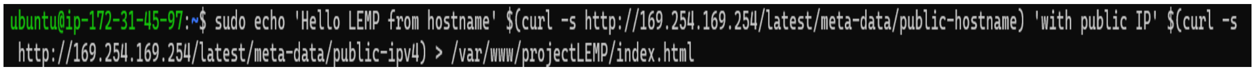
```
ubuntu@ip-172-31-45-97:~$ sudo unlink /etc/nginx/sites-enabled/default
```

And then reload the nginx service with the systemctl reload command.



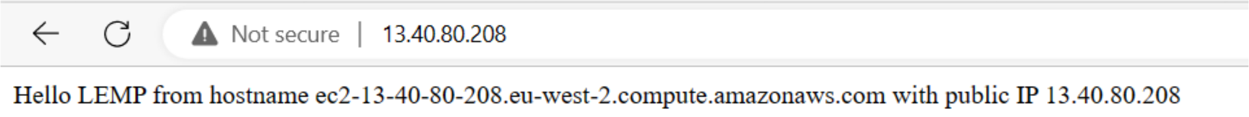
```
ubuntu@ip-172-31-45-97:~$ sudo systemctl reload nginx
```

To create an index.html file to test that the virtual host works as expected run.



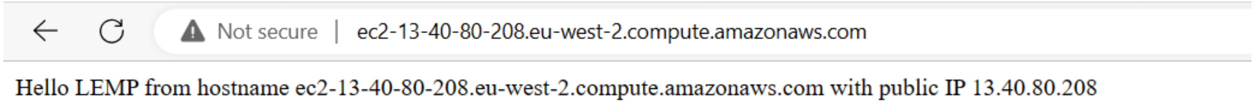
```
ubuntu@ip-172-31-45-97:~$ sudo echo 'Hello LEMP from hostname' $(curl -s http://169.254.169.254/latest/meta-data/public-hostname) 'with public IP' $(curl -s
 http://169.254.169.254/latest/meta-data/public-ipv4) > /var/www/projectLEMP/index.html
```

Using the Public IP address http://13.40.80.208/80 on my browser which displayed the text from the 'echo' command above indicating that my Nginx virtual host is operating as intended.



Hello LEMP from hostname ec2-13-40-80-208.eu-west-2.compute.amazonaws.com with public IP 13.40.80.208

And if I change this using the DNS on my browser the result should be the same.

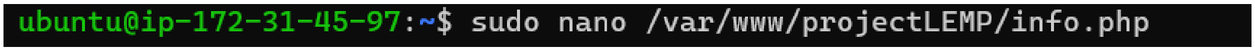


Not secure | ec2-13-40-80-208.eu-west-2.compute.amazonaws.com

Hello LEMP from hostname ec2-13-40-80-208.eu-west-2.compute.amazonaws.com with public IP 13.40.80.208

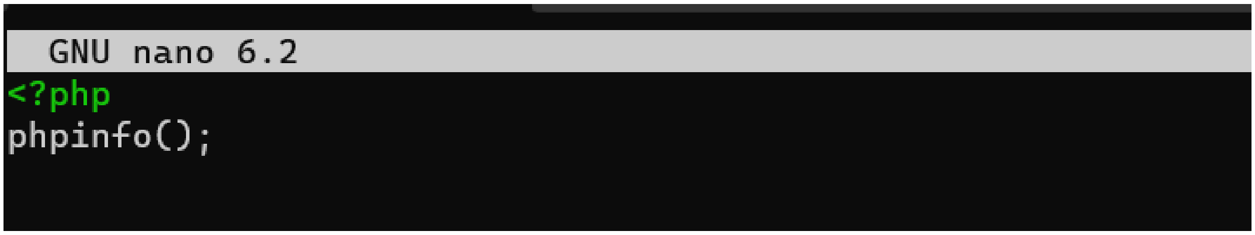
Testing PHP with Nginx

To validate that Nginx is fully operational, I created and opened an info.php file inside the custom web root folder.



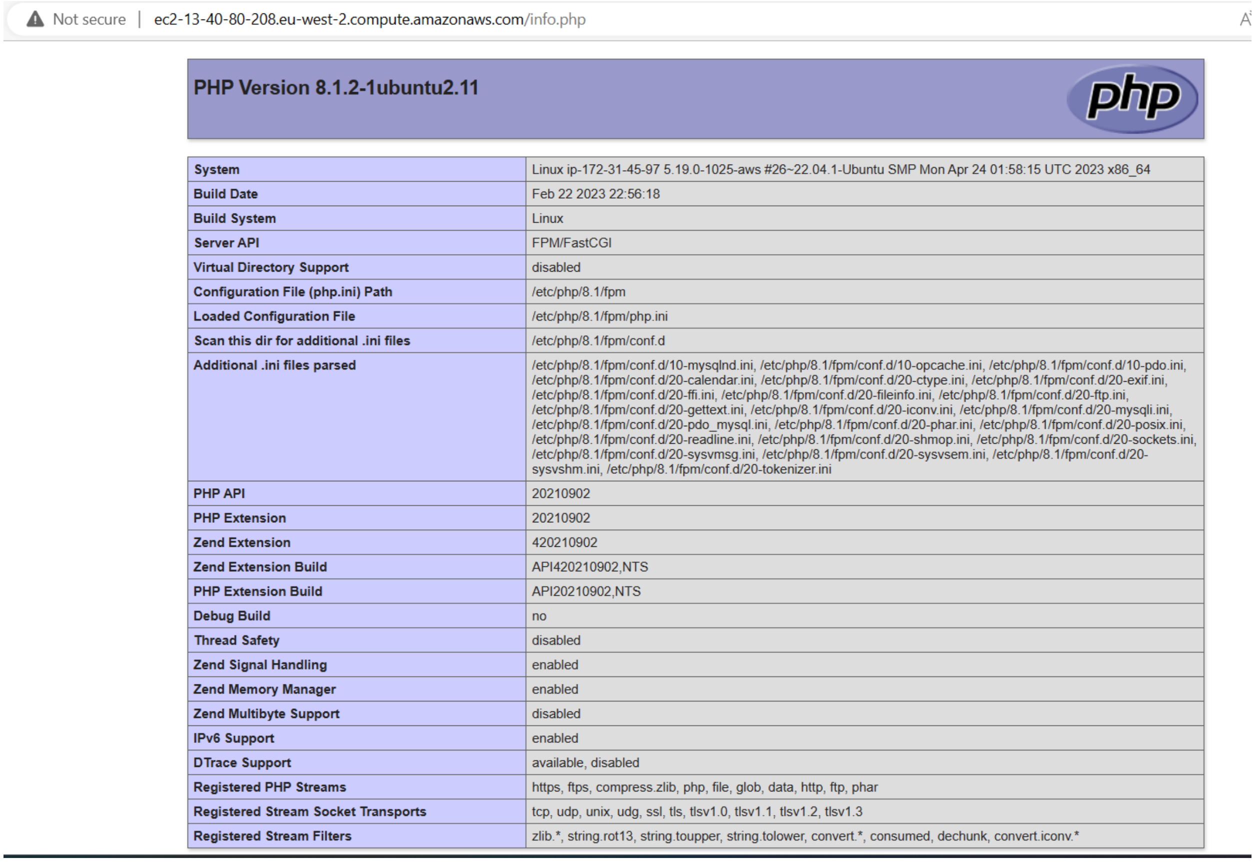
```
ubuntu@ip-172-31-45-97:~$ sudo nano /var/www/projectLEMP/info.php
```

I pasted the below PHP code into the file and saved the changes.



```
  GNU nano 6.2
<?php
phpinfo();
```

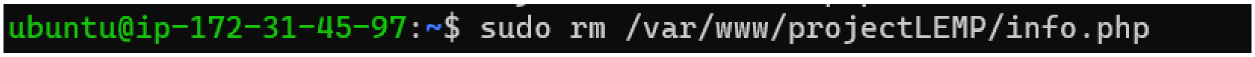
So using my domain name with info.php - http://ec2-3-144-206-238.us-easst-2.compute.amazonaws.com/info.php, will give the below as the expected result which gives one some detailed information about my server.


Not secure | ec2-13-40-80-208.eu-west-2.compute.amazonaws.com/info.php

PHP Version 8.1.2-1ubuntu2.11

System	Linux ip-172-31-45-97 5.19.0-1025-aws #26~22.04.1-Ubuntu SMP Mon Apr 24 01:58:15 UTC 2023 x86_64
Build Date	Feb 22 2023 22:56:18
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/fpm
Loaded Configuration File	/etc/php/8.1/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/fpm/conf.d
Additional .ini files parsed	/etc/php/8.1/fpm/conf.d/10-mysqlnd.ini, /etc/php/8.1/fpm/conf.d/10-opcache.ini, /etc/php/8.1/fpm/conf.d/10-pdo.ini, /etc/php/8.1/fpm/conf.d/20-calendar.ini, /etc/php/8.1/fpm/conf.d/20-ctype.ini, /etc/php/8.1/fpm/conf.d/20-exif.ini, /etc/php/8.1/fpm/conf.d/20-ffi.ini, /etc/php/8.1/fpm/conf.d/20-fileinfo.ini, /etc/php/8.1/fpm/conf.d/20-ftp.ini, /etc/php/8.1/fpm/conf.d/20-gettext.ini, /etc/php/8.1/fpm/conf.d/20-iconv.ini, /etc/php/8.1/fpm/conf.d/20-mysqli.ini, /etc/php/8.1/fpm/conf.d/20-pdo_mysql.ini, /etc/php/8.1/fpm/conf.d/20-phar.ini, /etc/php/8.1/fpm/conf.d/20-posix.ini, /etc/php/8.1/fpm/conf.d/20-readline.ini, /etc/php/8.1/fpm/conf.d/20-shmop.ini, /etc/php/8.1/fpm/conf.d/20-sockets.ini, /etc/php/8.1/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.1/fpm/conf.d/20-sysvsem.ini, /etc/php/8.1/fpm/conf.d/20-sysvshm.ini, /etc/php/8.1/fpm/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
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

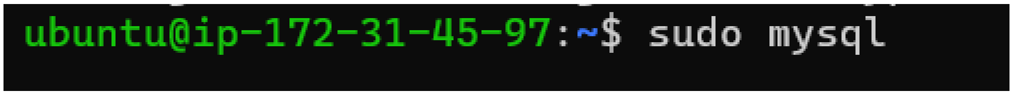
As a best practice, it is important I removed the index.php file I created as it contains sensitive data by running: sudo rm /var/www/your_domain/info.php



```
ubuntu@ip-172-31-45-97:~$ sudo rm /var/www/projectLEMP/info.php
```

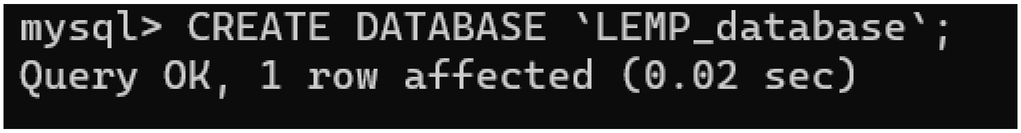
Retrieving data from MySQL

To continue with the MySQL database, I switched over to Mysql.



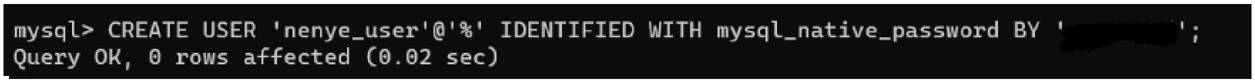
```
ubuntu@ip-172-31-45-97:~$ sudo mysql
```

I then created a database known as LEMP_database since this is a LEMP stack implementation.

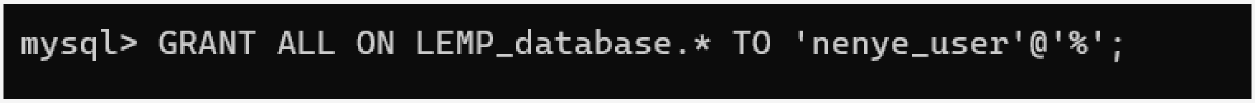


```
mysql> CREATE DATABASE `LEMP_database`;
Query OK, 1 row affected (0.02 sec)
```

Now it is time I created the database user by running the below command and setting up a password for it as well. While granting the user all the intended privileges to the LEMP_database.

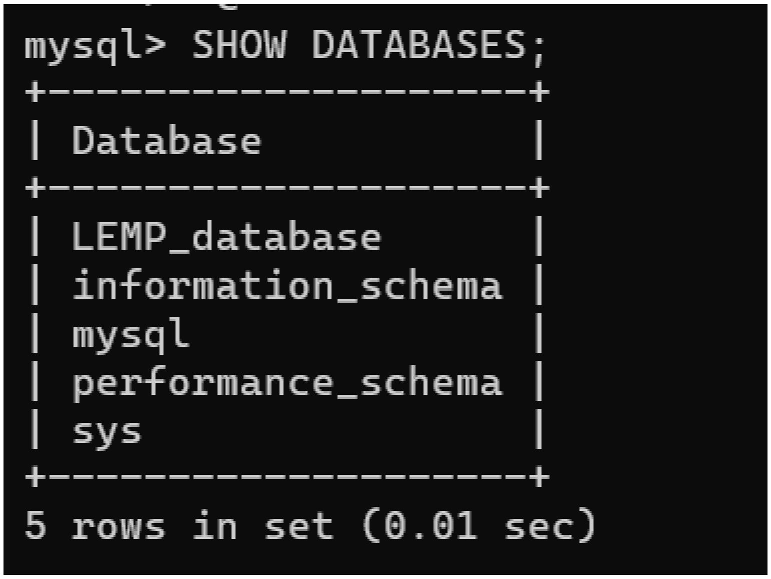


```
mysql> CREATE USER 'nenye_user'@'%' IDENTIFIED WITH mysql_native_password BY '[redacted]';
Query OK, 0 rows affected (0.02 sec)
```



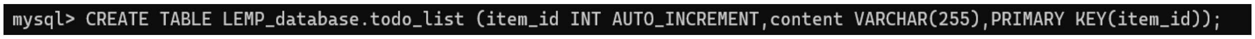
```
mysql> GRANT ALL ON LEMP_database.* TO 'nenye_user'@'%';
```

Now that the database and user have been successfully created, the show databases command will now provide a list of Databases that have been created. The additional ones there appear by default.



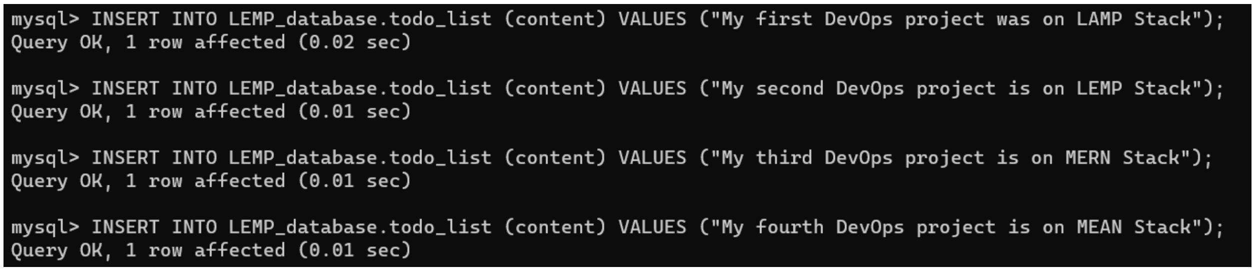
```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| LEMP_database      |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)
```

Also, to create a table known as todo_list for the LEMP_database, I used the CREATE.



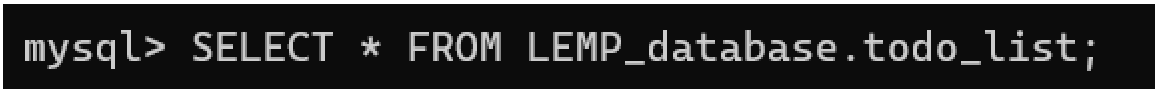
```
mysql> CREATE TABLE LEMP_database.todo_list (item_id INT AUTO_INCREMENT,content VARCHAR(255),PRIMARY KEY(item_id));
```

To add data into the LEMP_database.todo_list the below INSERT command was used to add the various data.

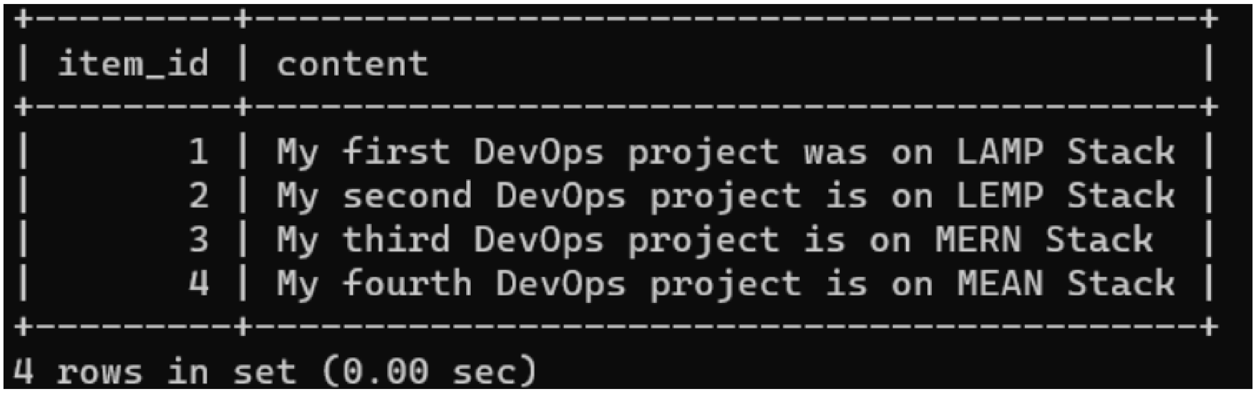


```
mysql> INSERT INTO LEMP_database.todo_list (content) VALUES ("My first DevOps project was on LAMP Stack");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO LEMP_database.todo_list (content) VALUES ("My second DevOps project is on LEMP Stack");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LEMP_database.todo_list (content) VALUES ("My third DevOps project is on MERN Stack");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LEMP_database.todo_list (content) VALUES ("My fourth DevOps project is on MEAN Stack");
Query OK, 1 row affected (0.01 sec)
```

The SELECT command is now used to view all data on the table which should give the item id as well as content in the tabular form.

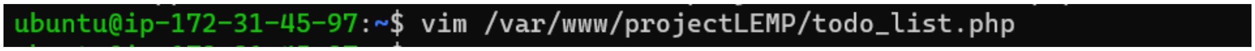


```
mysql> SELECT * FROM LEMP_database.todo_list;
```



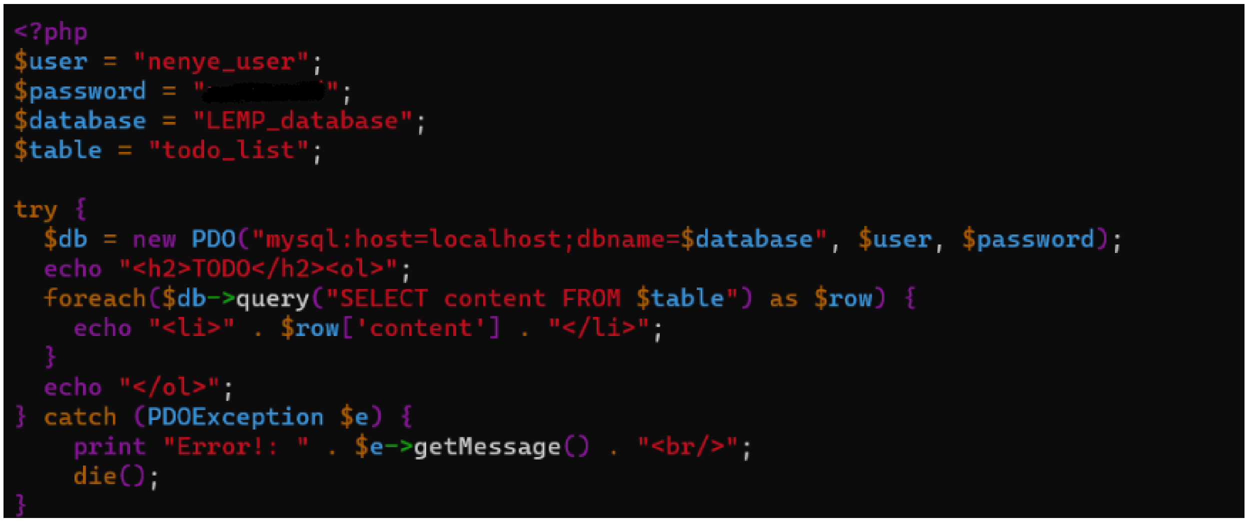
```
+---------+-------------------------------------------+
| item_id | content                                   |
+---------+-------------------------------------------+
|       1 | My first DevOps project was on LAMP Stack |
|       2 | My second DevOps project is on LEMP Stack |
|       3 | My third DevOps project is on MERN Stack  |
|       4 | My fourth DevOps project is on MEAN Stack |
+---------+-------------------------------------------+
4 rows in set (0.00 sec)
```

The final step now will be to populate this to my browser by allowing the PHP script to talk to the LEMP_database. The **vim** or nano command should be able to do this. Let me go with **vim** this time.



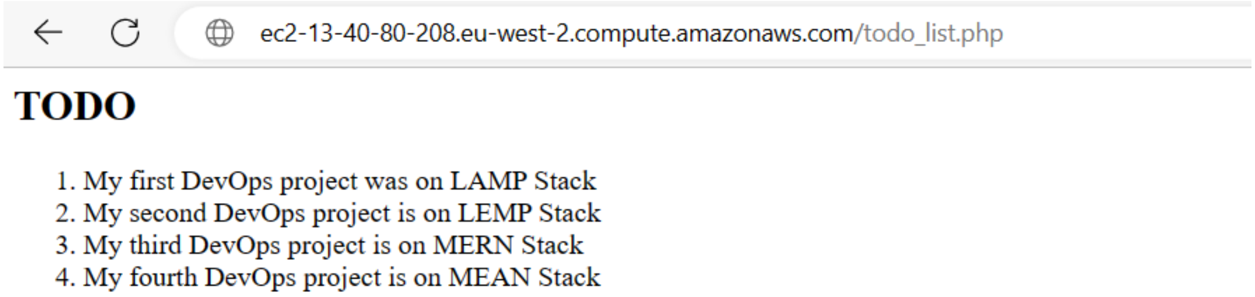
```
ubuntu@ip-172-31-45-97:~$ vim /var/www/projectLEMP/todo_list.php
```

I then copied the content of my database into the todo_list.php script.



```
<?php
$user = "nenye_user";
$password = "[redacted]";
$database = "LEMP_database";
$table = "todo_list";

try {
  $db = new PDO("mysql:host=localhost;dbname=$database", $user, $password);
  echo "<h2>TODO</h2><ol>";
  foreach($db->query("SELECT content FROM $table") as $row) {
    echo "<li>" . $row['content'] . "</li>";
  }
  echo "</ol>";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
```

The PHP script has now connected successfully to the MySQL database to generate the below in my web browser.



ec2-13-40-80-208.eu-west-2.compute.amazonaws.com/todo_list.php

TODO

1. My first DevOps project was on LAMP Stack
2. My second DevOps project is on LEMP Stack
3. My third DevOps project is on MERN Stack
4. My fourth DevOps project is on MEAN Stack

My LEMP Stack environment is now ready and good to go! 💃💃

REFERENCES

Heng, Y.C. (2020) Lemp, Techcoil. Available at: https://www.techcoil.com/glossary/lemp/ (Accessed: 30 May 2023).

Nginx. Available at: https://nginx.org/en/ (Accessed: 31 May 2023).

What is Lemp Stack? (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/what-is-lemp-stack/ (Accessed: 30 May 2023).