- ## WEB STACK IMPLEMENTATION (LEMP STACK)

After successfully implementing LAMP stack, In this project we will be implementing LEMP stack but with an alternative Web Server – NGINX, which is also very popular and widely used by many websites in the Internet. As we have done in project 1, we'll be creating a new instance for project 2 and ssh into the instance.

- ## INSTALLING THE NGINX WEB SERVER

We will need to install NGINX in other to display web pages to site visitors and NGINX is a high-performance web server. Since this is our first time using apt [ADVANCED PACKAGE TOOL] for this project, we will start by updating the server's package index. Following that, we'll use apt install to get Nginx installed:

1. `sudo apt update`

```
sudo apt install nginx
```

After entering the command, enter Y to confirm that you want to install Nginx. Once the installation is finished, the Nginx web server will be active and running on your Ubuntu 20.04 server.

2. To verify that nginx was successfully installed and is running as a service in Ubuntu, run `sudo systemctl status nginx`

If it is green and running, then everything was done correctly. We've launched NGINX in the clouds. In other to start receiving traffic, we need to open TCP Port 80 which is the default port used by web browsers to access web pages. As we know, we have TCP port 22 open by default on our EC2 machine to access it via SSH, so we need to add a rule to EC2 configuration to open inbound connection through port 80. Once done the server will be running and we can access it locally in our Ubuntu shell by running this command `curl http://localhost:80 Or curl` http://127.0.0.1:80

Now let's open our browser and try accessing our web server with the following url http://<Public-IP-Address>:80 you can also access the page using your DNS name as well.

```
Preparing to unpack .../10-libgd3_2.3.0-2ubuntu2_amd64.deb ...
Unpacking libgd3:amd64 (2.3.0-2ubuntu2) ...
Selecting previously unselected package nginx-common.
Preparing to unpack .../11-nginx-common_1.18.0-6ubuntu14.3_all.deb ...
Unpacking nginx-common (1.18.0-6ubuntu14.3) ...
Selecting previously unselected package libnginx-mod-http-geoip2.
Preparing to unpack .../12-libnginx-mod-http-geoip2_1.18.0-6ubuntu14.3_amd64.deb ...
Unpacking libnginx-mod-http-geoip2 (1.18.0-6ubuntu14.3) ...
Selecting previously unselected package libnginx-mod-http-image-filter.
Preparing to unpack .../13-libnginx-mod-http-image-filter_1.18.0-6ubuntu14.3_amd64.deb ...
Unpacking libnginx-mod-http-image-filter (1.18.0-6ubuntu14.3) ...
Selecting previously unselected package libnginx-mod-http-xslt-filter.
Preparing to unpack .../14-libnginx-mod-http-xslt-filter_1.18.0-6ubuntu14.3_amd64.deb ...
Unpacking libnginx-mod-http-xslt-filter (1.18.0-6ubuntu14.3) ...
Selecting previously unselected package libnginx-mod-mail.
Preparing to unpack .../15-libnginx-mod-mail_1.18.0-6ubuntu14.3_amd64.deb ...
Unpacking libnginx-mod-mail (1.18.0-6ubuntu14.3) ...
Selecting previously unselected package libnginx-mod-stream.
Preparing to unpack .../16-libnginx-mod-stream_1.18.0-6ubuntu14.3_amd64.deb ...
Unpacking libnginx-mod-stream (1.18.0-6ubuntu14.3) ...
Selecting previously unselected package libnginx-mod-stream-geoip2.
Preparing to unpack .../17-libnginx-mod-stream-geoip2_1.18.0-6ubuntu14.3_amd64.deb ...
Unpacking libnginx-mod-stream-geoip2 (1.18.0-6ubuntu14.3) ...
Selecting previously unselected package nginx-core.
Preparing to unpack .../18-nginx-core_1.18.0-6ubuntu14.3_amd64.deb ...
Unpacking nginx-core (1.18.0-6ubuntu14.3) ...
Selecting previously unselected package nginx.
Preparing to unpack .../19-nginx_1.18.0-6ubuntu14.3_amd64.deb ...
Unpacking nginx (1.18.0-6ubuntu14.3) ...
Setting up libxpm4:amd64 (1:3.5.12-1ubuntu0.22.04.1) ...
Setting up libdeflate0:amd64 (1.10-2) ...
Setting up nginx-common (1.18.0-6ubuntu14.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /lib/systemd/system/nginx.service.
Setting up libjbig0:amd64 (2.1-3.1ubuntu0.22.04.1) ...
Setting up libnginx-mod-http-xslt-filter (1.18.0-6ubuntu14.3) ...
Setting up fonts-dejavu-core (2.37-2build1) ...
Setting up libjpeg-turbo8:amd64 (2.1.2-0ubuntu1) ...
Setting up libwebp7:amd64 (1.2.2-2) ...
Setting up libnginx-mod-http-geoip2 (1.18.0-6ubuntu14.3) ...
Setting up libjpeg8:amd64 (8c-2ubuntu10) ...
Setting up libnginx-mod-mail (1.18.0-6ubuntu14.3) ...
Setting up fontconfig-config (2.13.1-4.2ubuntu5) ...
Setting up libnginx-mod-stream (1.18.0-6ubuntu14.3) ...
Setting up libtiff5:amd64 (4.3.0-6ubuntu0.4) ...
Setting up libfontconfig1:amd64 (2.13.1-4.2ubuntu5) ...
Setting up libnginx-mod-stream-geoip2 (1.18.0-6ubuntu14.3) ...
Setting up libgd3:amd64 (2.3.0-2ubuntu2) ...
Setting up libnginx-mod-http-image-filter (1.18.0-6ubuntu14.3) ...
Setting up nginx-core (1.18.0-6ubuntu14.3) ...
 * Upgrading binary nginx                                                 [ OK ]
Setting up nginx (1.18.0-6ubuntu14.3) ...
Processing triggers for ufw (0.36.1-4build1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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.
ubuntu@ip-172-31-30-20:~$
```

```
ubuntu@ip-172-31-30-20:~$ 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:>
     Active: active (running) since Sun 2023-03-12 14:51:01 UTC; 52s ago
       Docs: man:nginx(8)
    Process: 1751 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proce>
    Process: 1752 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (c>
   Main PID: 1845 (nginx)
      Tasks: 2 (limit: 1143)
     Memory: 4.6M
        CPU: 23ms
     CGroup: /system.slice/nginx.service
             ├─1845 "nginx: master process /usr/sbin/nginx -g daemon on; master>
             └─1848 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" ""

Mar 12 14:51:01 ip-172-31-30-20 systemd[1]: Starting A high performance web ser>
Mar 12 14:51:01 ip-172-31-30-20 systemd[1]: Started A high performance web serv>
ubuntu@ip-172-31-30-20:~$
```

```
ubuntu@ip-172-31-30-20:~$ 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>
ubuntu@ip-172-31-30-20:~$
```

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**

NGINX web server is up and running now we will need to install a Database Management System (DBMS) to be able to store and manage data for your site in a relational database so this is where MYSQL will come into play.

1. Again, use ‘apt’ to acquire and install this software

```
sudo apt install mysql-server
```

confirm installation by typing Y, and then ENTER.

2. log in to the MySQL console using this command

```
sudo mysql
```

This will connect to the MySQL server as the administrative database user **root**, which is inferred by the use of sudo when running this command.

3. It’s recommended that we run a security script that comes pre-installed with MySQL. This script will remove some insecure default settings and lock down access to your database system. Before running the script we will set a password for the root user, using mysql_native_password as default authentication method. We’re defining this user’s password as PassWord.1

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'PassWord.1';
```

4. Exit the MYSQL shell with

```
mysql> exit
```

5. Start the interactive script by running

```
sudo mysql_secure_installation
```

This will ask if you want to configure the VALIDATE PASSWORD PLUGIN. Answer Y for yes, or anything else to continue without enabling. If you answer “yes”, you’ll be asked to select a level of password validation. Keep in mind that if you enter 2 for the strongest level, you will receive errors when attempting to set any password which does not contain numbers, upper and lowercase letters, and special characters, or which is based on common dictionary words e.g PassWord.1

Regardless of whether you chose to set up the VALIDATE PASSWORD PLUGIN, your server will next ask you to select and confirm a password for the MySQL root user. This is not to be confused with the system root. The database root user is an administrative user with full privileges over the database system. Even though the default authentication method for the MySQL root user dispenses the use of a password, even when one is set, you should define a strong password here as an additional safety measure. We’ll talk about this in a moment.

If you enabled password validation, you’ll be shown the password strength for the root password you just entered and your server will ask if you want to continue with that password. If you are happy with your current password, enter Y for “yes” at the prompt. For the rest of the questions, press Y and hit the ENTER key at each prompt.

6. When you’re finished, test if you’re able to log in to the MySQL console with this command `sudo mysql -p`

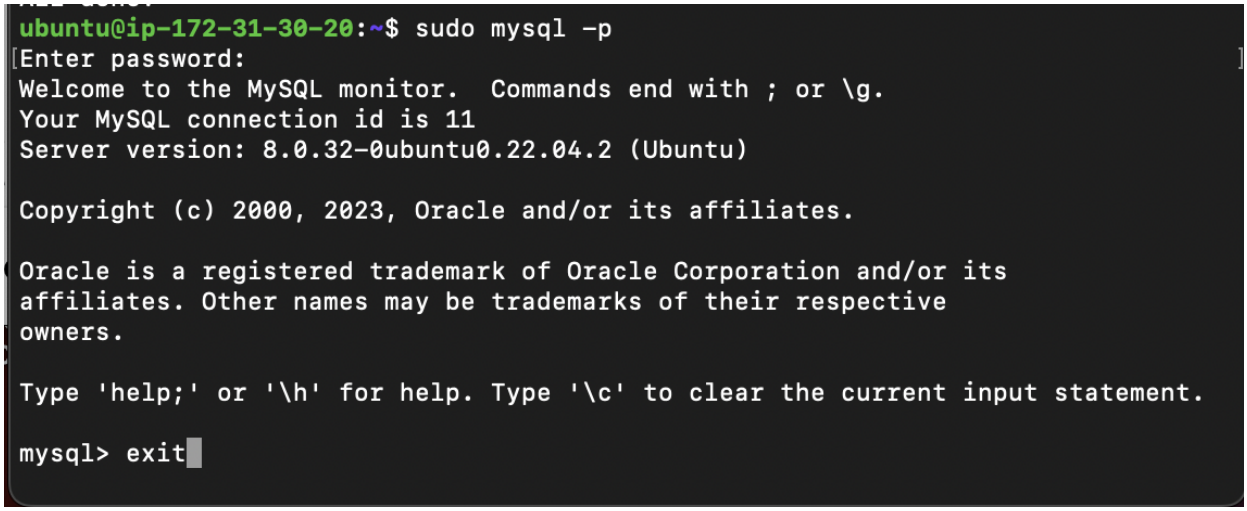
The -p flag in this command, which will prompt you for the password used after changing the root user password in this case PassWord.1

7. To exit the MySQL console `mysql> exit`

MySQL server is now installed and secured.

```
done!
update-alternatives: using /var/lib/mecab/dic/ipadic to provide /var/lib/mecab/dic/debian (mecab-dictionary) in auto mode
Setting up mysql-server-core-8.0 (8.0.32-0ubuntu0.22.04.2) ...
Setting up mecab-ipadic-utf8 (2.7.0-20070801+main-3) ...
Compiling IPA dictionary for Mecab.  This takes long time...
reading /usr/share/mecab/dic/ipadic/unk.def ... 40
emitting double-array: 100% |###########################################|
/usr/share/mecab/dic/ipadic/model.def is not found. skipped.
reading /usr/share/mecab/dic/ipadic/Suffix.csv ... 1393
reading /usr/share/mecab/dic/ipadic/Noun.others.csv ... 151
reading /usr/share/mecab/dic/ipadic/Interjection.csv ... 252
reading /usr/share/mecab/dic/ipadic/Others.csv ... 2
reading /usr/share/mecab/dic/ipadic/Noun.place.csv ... 72999
reading /usr/share/mecab/dic/ipadic/Symbol.csv ... 208
reading /usr/share/mecab/dic/ipadic/Adnominal.csv ... 135
reading /usr/share/mecab/dic/ipadic/Noun.adjv.csv ... 3328
reading /usr/share/mecab/dic/ipadic/Noun.proper.csv ... 27328
reading /usr/share/mecab/dic/ipadic/Noun.org.csv ... 16668
reading /usr/share/mecab/dic/ipadic/Verb.csv ... 130750
reading /usr/share/mecab/dic/ipadic/Noun.verbal.csv ... 12146
reading /usr/share/mecab/dic/ipadic/Auxil.csv ... 199
reading /usr/share/mecab/dic/ipadic/Postp-col.csv ... 91
reading /usr/share/mecab/dic/ipadic/Noun.number.csv ... 42
reading /usr/share/mecab/dic/ipadic/Noun.demonst.csv ... 120
reading /usr/share/mecab/dic/ipadic/Noun.adverbal.csv ... 795
reading /usr/share/mecab/dic/ipadic/Adj.csv ... 27210
reading /usr/share/mecab/dic/ipadic/Filler.csv ... 19
reading /usr/share/mecab/dic/ipadic/Noun.csv ... 60477
reading /usr/share/mecab/dic/ipadic/Noun.name.csv ... 34202
reading /usr/share/mecab/dic/ipadic/Postp.csv ... 146
reading /usr/share/mecab/dic/ipadic/Conjunction.csv ... 171
reading /usr/share/mecab/dic/ipadic/Adverb.csv ... 3032
reading /usr/share/mecab/dic/ipadic/Prefix.csv ... 221
reading /usr/share/mecab/dic/ipadic/Noun.nai.csv ... 42
emitting double-array: 100% |###########################################|
reading /usr/share/mecab/dic/ipadic/matrix.def ... 1316x1316
emitting matrix      : 100% |###########################################|

done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/lib/mecab/dic/debian (mecab-dictionary) in auto mode
Setting up libhtml-parser-perl:amd64 (3.76-1build2) ...
Setting up libhttp-message-perl (6.36-1) ...
Setting up mysql-server-8.0 (8.0.32-0ubuntu0.22.04.2) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 2479
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up libcgi-pm-perl (4.54-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up mysql-server (8.0.32-0ubuntu0.22.04.2) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

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.
ubuntu@ip-172-31-30-20:~$
```

```
ubuntu@ip-172-31-30-20:~$ 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: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary                  file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 2
Using existing password for root.

Estimated strength of the password: 100
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n

 ... skipping.
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL 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? (Press y|Y for Yes, any other key for No) : 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? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-172-31-30-20:~$ sudo mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
```



- **INSTALLING PHP**

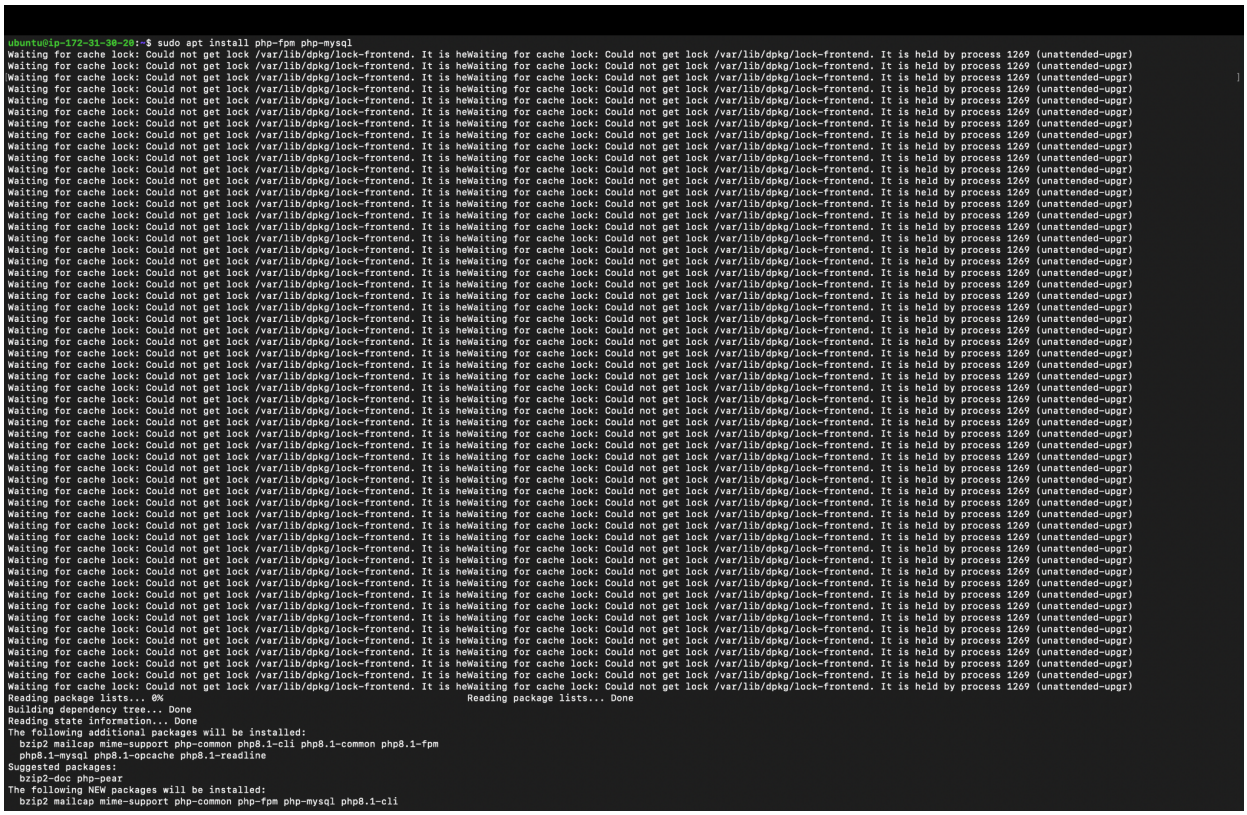
Nginx has been installed to serve your content and MySQL installed to store and manage your data. Now we can install PHP to process code and generate dynamic content for the web server.

While Apache embeds the PHP interpreter in each request, Nginx requires an external program to handle PHP processing and act as a bridge between the PHP interpreter itself and the web server. This allows for a better overall performance in most PHP-based websites, but it requires additional configuration. You'll need to install php-fpm, which stands for "PHP fastCGI process manager", and tell Nginx to pass PHP requests to this software for processing. Additionally, you'll need php-mysql, a PHP module that allows PHP to communicate with MySQL-based databases. Core PHP packages will automatically be installed as dependencies.

To install these 2 packages at once run `sudo apt install php-fpm php-mysql`

When prompted, type Y and press ENTER to confirm installation.

Now PHP components has been installed.



- **CONFIGURING NGINX TO USE PHP PROCESSOR**

When using the Nginx web server, we can create server blocks (similar to virtual hosts in Apache) to encapsulate configuration details and host more than one domain on a single server. we will use piusprojectLEMP as domain name.

On Ubuntu 20.04, Nginx has one server block enabled by default and is configured to serve documents out of a directory at /var/www/html. While this works well for a single site, it can become difficult to manage if you are hosting multiple sites. Instead of modifying /var/www/html, we'll create a directory structure within /var/www for the your_domain website, leaving /var/www/html in place as the default directory to be served if a client request does not match any other sites.

1. Create the root web directory for **your_domain**

```
sudo mkdir /var/www/projectLEMP
```

2. Change ownership of the directory with the $USER, which is your current system user. `sudo chown -R $USER:$USER /var/www/piusprojectLEMP`

3. open a new configuration file in Nginx's sites-available directory using your preferred command-line editor. Here, we'll use nano `sudo nano /etc/nginx/sites-available/piusprojectLEMP`
4. This will create a new blank file. Paste in the following configuration

```
#/etc/nginx/sites-available/piusprojectLEMP



server {

    listen 80;

    server_name piusprojectLEMP www.piusprojectLEMP;

    root /var/www/piusprojectLEMP;



    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;
```

```
    }
```

```
}
```

When you're done editing, save and close the file. If you're using nano, you can do so by typing CTRL+X and then y and ENTER to confirm.

5. Activate your configuration by linking to the config file from Nginx's sites-enabled directory: `sudo ln -s /etc/nginx/sites-available/piusprojectLEMP /etc/nginx/sites-enabled/`
6. This will tell Nginx to use the configuration next time it is reloaded. You can test your configuration for syntax errors `sudo nginx -t`

   If any errors are reported, go back to your configuration file to review its contents before continuing. If you see the following message then you are good to go.

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

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

7. Now disable default Nginx host that is currently configured to listen on port 80, for this run `sudo unlink /etc/nginx/sites-enabled/default`
8. Reload Nginx to apply the changes `sudo systemctl reload nginx`
9. The website is now active, but the web root /var/www/piusprojectLEMP is still empty. Create an index.html file in that location so that we can test that your new server block works as expected. `sudo echo 'Hello piusLEMP 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/piusprojectLEMP/index.html`

Go to your browser and use your IP or DNS name, if you get what you pasted after "echo" then you did everything correctly.

```
[ubuntu@ip-172-31-30-20:~$ sudo mkdir /var/www/piusprojectLEMP
 ubuntu@ip-172-31-30-20:~$ sudo chown -R $USER:$USER /var/www/piusprojectLEMP
 ubuntu@ip-172-31-30-20:~$ sudo nano /etc/nginx/sites-available/piusprojectLEMP
 ubuntu@ip-172-31-30-20:~$ sudo ln -s /etc/nginx/sites-available/projectLEMP /etc/nginx/sites-enabled/
 ubuntu@ip-172-31-30-20:~$ sudo ln -s /etc/nginx/sites-available/piusprojectLEMP /etc/nginx/sites-enabled/
[ubuntu@ip-172-31-30-20:~$
```

```
[ubuntu@ip-172-31-30-20:~$ sudo nginx -t
 nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
 nginx: configuration file /etc/nginx/nginx.conf test is successful
 ubuntu@ip-172-31-30-20:~$
```

```
[ubuntu@ip-172-31-30-20:~$ sudo echo 'Hello piusLEMP from hostname' $(curl -s htt]
 p://169.254.169.254/latest/meta-data/public-hostname) 'with public IP' $(curl -s
  http://169.254.169.254/latest/meta-data/public-ipv4) > /var/www/piusprojectLEMP
 /index.html
 ubuntu@ip-172-31-30-20:~$
```

Hello piusLEMP from hostname ec2-52-24-144-213.us-west-2.compute.amazonaws.com with public IP 52.24.144.213

- **TESTING PHP WITH NGINX**

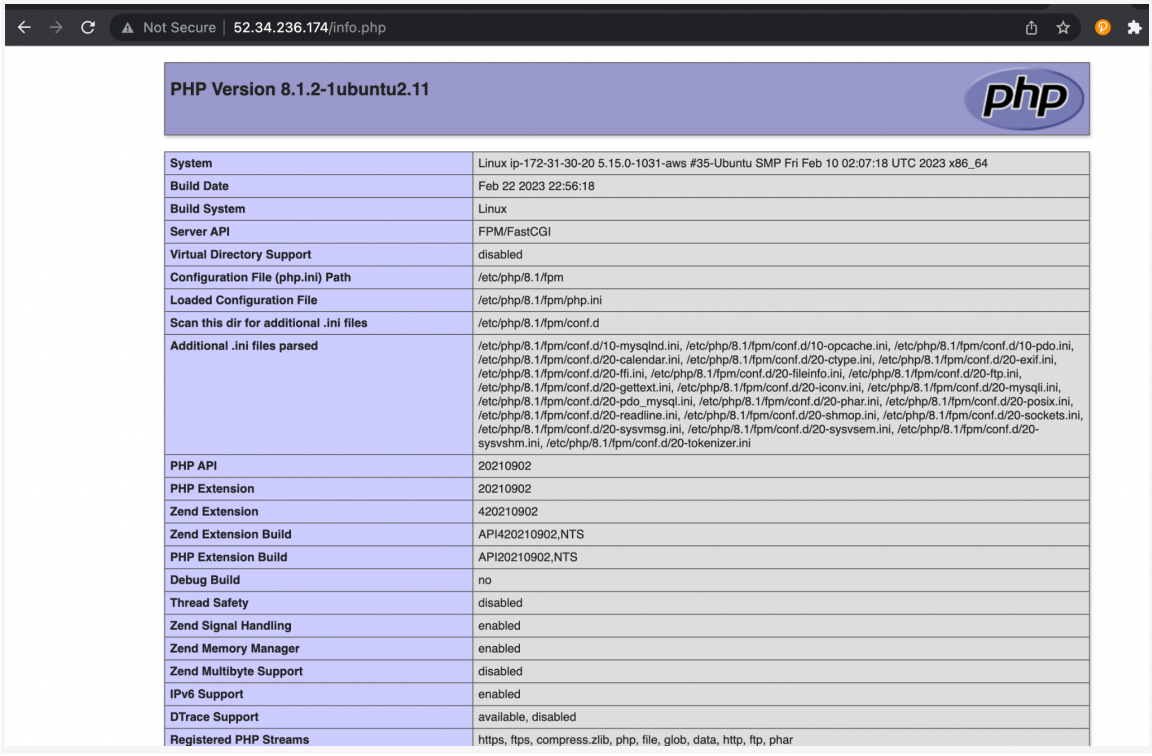
Lets validate that Nginx can correctly hand .php files off to your PHP processor. Lets create a test PHP file in document root.

1. Open a new file called info.php within document root in your text editor: `sudo nano /var/www/piusprojectLEMP/info.php`
2. Type or paste the following lines into the new file. This is valid PHP code that will return information about your server:

   ```
   <?php

   phpinfo();
   ```

   You can now access this page in your web browser by visiting the domain name or public IP address you've set up in your Nginx configuration file, followed by /info.php. `http://`server_domain_or_IP`/info.php`

   After checking the relevant information about your PHP server through that page, it's best to remove the file you created as it contains sensitive information about your PHP environment and your Ubuntu server. You can use rm to remove that file: `sudo rm /var/www/piusprojectLEMP/info.php`



System	Linux ip-172-31-30-20 5.15.0-1031-aws #35-Ubuntu SMP Fri Feb 10 02:07:18 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

- **RETRIEVING DATA FROM MYSQL DATABASE WITH PHP**

Lets create a test database (DB) with "To do list" and configure access to it, so the Nginx website would be able to query data from the DB and display it. We'll create a new user with the mysql_native_password authentication method in order to be able to connect to the MySQL database from PHP.
We will create a database named example_database and a user named example_user, but it can replaced with different values/names.

1. Connect to the MYSQL consol using the root user account
   `sudo mysql -p`
2. To create a new database, run the following command from MySQL console
   `mysql> CREATE DATABASE `example_database`;`
3. create a new user and grant him full privileges on the database you have just created. The following command creates a new user named example_user, using mysql_native_password as default authentication method. We're defining this user's password as PassWord.1, but you should replace this value with a secure password of your own choosing. `mysql>  CREATE USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'PassWord.1';`
4. Now we need to give this user permission over the example_database database:
   `mysql> GRANT ALL ON example_database.* TO 'example_user'@'%';`

   This will give the **example_user** user full privileges over the **example_database** database, while preventing this user from creating or modifying other databases on your server

5. Now exit the MySQL shell with: `mysql> exit`
6. Lets test if the new user has the proper permissions by logging in to the MySQL console again, this time using the custom user credentials: `mysql -u example_user -p`
7. After logging in to the MySQL console, confirm that you have access to the example_database database: `mysql> SHOW DATABASES;`

This will give you the following output:

```
Output
+--------------------+
| Database           |
+--------------------+
| example_database   |
| information_schema |
+--------------------+
2 rows in set (0.000 sec
```

8. Next, we'll create a test table named **todo_list**. From the MySQL console, run the following statement

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

9. Insert a few rows of content in the test table. You might want to repeat the next command a few times, using different VALUES: `mysql> INSERT INTO example_database.todo_list (content) VALUES ("My first important item");`

   ```
   mysql> INSERT INTO example_database.todo_list (content) VALUES ("Pius is a KIng");
   ```

10. To confirm that the data was successfully saved to your table, run: `mysql> SELECT * FROM example_database.todo_list;`

11. After confirming that you have valid data in your test table, you can exit the MySQL console: `mysql> exit`

12. Now you can create a PHP script that will connect to MySQL and query for your content. Create a new PHP file in your custom web root directory using your preferred editor. We'll use vi for that: `nano /var/www/piusprojectLEMP/todo_list.php`
13. The following PHP script connects to the MySQL database and queries for the content of the todo_list table, displays the results in a list. If there is a problem with the database connection, it will throw an exception.
    Copy this content into your todo_list.php script:

```
<?php
$user = "example_user";
$password = "PassWord.1";
$database = "example_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();
}
```

    Save and close the file when you are done editing.

    You can now access this page in your web browser by visiting the domain name or public IP address configured for your website, followed by /todo_list.php:

    `http://<Public_domain_or_IP>/todo_list.php`

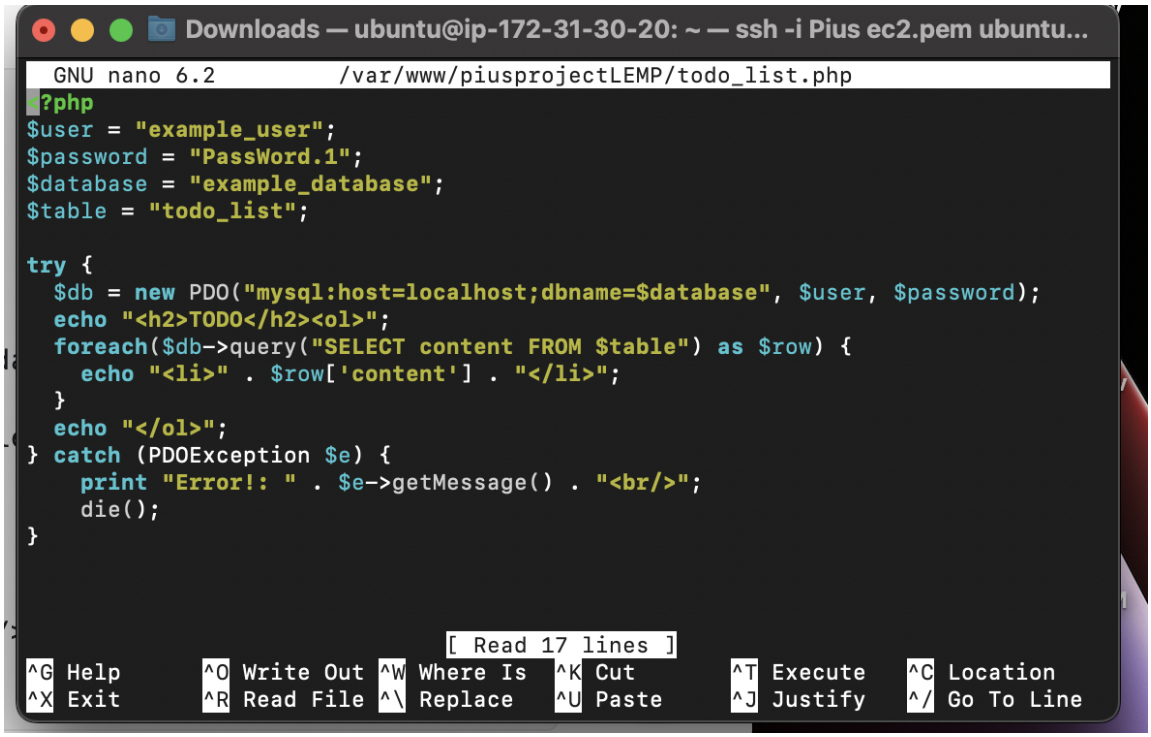
    If you see your to do list content that means your PHP environment is ready to connect and interact with your MySQL server.

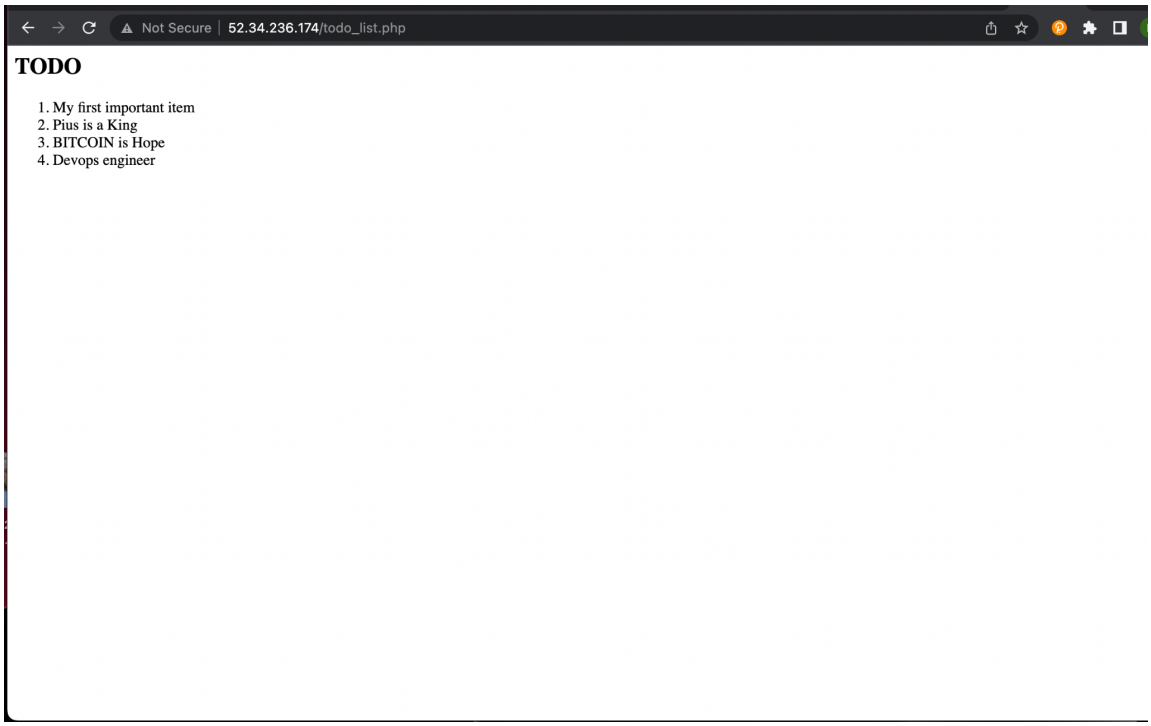
```
ubuntu@ip-172-31-30-20:~$ sudo mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

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

mysql> CREATE USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'password';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> CREATE USER 'example_user'@'%' IDENTIFIED WITH mysql_native_password BY 'PassWord.1';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL ON example_database.* TO 'example_user'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
ubuntu@ip-172-31-30-20:~$ mysql -u example_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.32-0ubuntu0.22.04.2 (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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| example_database   |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0.01 sec)
```

```
mysql> CREATE TABLE example_database.todo_list (
    ->  item_id INT AUTO_INCREMENT,
    -> content VARCHAR(255),
    ->  PRIMARY KEY(item_id)
    ->  );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("My first important item");
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM example_database.todo_list;
+---------+-------------------------+
| item_id | content                 |
+---------+-------------------------+
|       1 | My first important item |
+---------+-------------------------+
1 row in set (0.00 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("Pius is a King");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("BITCOIN is Hope");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("Devops engineer");
Query OK, 1 row affected (0.01 sec)

mysql>
```

```
mysql> SELECT * FROM example_database.todo_list;
+----------+-------------------------+
| item_id | content                  |
+----------+-------------------------+
|       1 | My first important item |
|       2 | Pius is a King          |
|       3 | BITCOIN is Hope         |
|       4 | Devops engineer         |
+----------+-------------------------+
4 rows in set (0.00 sec)
```


Downloads — ubuntu@ip-172-31-30-20: ~ — ssh -i Pius ec2.pem ubuntu...
GNU nano 6.2 /var/www/piusprojectLEMP/todo_list.php
<?php
$user = "example_user";
$password = "PassWord.1";
$database = "example_database";
$table = "todo_list";
try {
$db = new PDO("mysql:host=localhost;dbname=$database", $user, $password);
echo "<h2>TODO</h2>
";
foreach($db->query("SELECT content FROM $table") as $row) {
echo "
" . $row['content'] . "
";
}
echo "
";
} catch (PDOException $e) {
print "Error!: " . $e->getMessage() . "
";
die();
}
[Read 17 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line


Not Secure | 52.34.236.174/todo_list.php
TODO
1. My first important item
2. Pius is a King
3. BITCOIN is Hope
4. Devops engineer