# MAG-PBL2-WEB STACK IMPLEMENTATION (LEMP STACK)

# Download and install Git Bash

# Launch Git Bash and run following command:

# ssh -i "PBL2.pem" ubuntu@ec2-3-135-232-204.us-east-2.compute.amazonaws.com

#### A screenshot of a computer screen Description automatically generated with medium confidence

#### Step 1 – Installing the Nginx Web Server:

**sudo apt update**

A picture containing text, screenshot, black and white, pattern

Description automatically generated

**sudo apt install nginx**

A screen shot of a computer

Description automatically generated with medium confidence

To verify that nginx was successfully installed and is running as a service in Ubuntu, run:

**sudo systemctl status nginx**

**A picture containing text, screenshot, font

Description automatically generated**

Add a rule to EC2 configuration to open inbound connection through port 80:

A screenshot of a computer

Description automatically generated with medium confidence

To check how we can access it locally in our Ubuntu shell, run:

**curl http://localhost:80**

A picture containing text, screenshot, font

Description automatically generated

To test how Nginx server can respond to requests from the Internet.

Run: http:// 3.135.232.204:80

A close-up of a message

Description automatically generated with low confidence

# STEP 2 — INSTALLING MYSQL

**sudo apt install mysql-server**

A screenshot of a computer screen

Description automatically generated with medium confidence

 Log in to the MySQL console by typing:

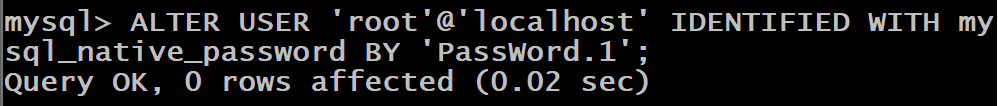
**sudo mysql**

**A screenshot of a computer screen

Description automatically generated with medium confidence**

 Before running the script 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';**



Exit the MySQL shell with:

**exit**

A picture containing text, font, screenshot, black

Description automatically generated

Start the interactive script by running:

**sudo mysql\_secure\_installation**

**A screenshot of a computer screen

Description automatically generated with medium confidence**

When finished, test if you’re able to log in to the MySQL console by typing:

**sudo mysql -p**

**A screenshot of a computer screen

Description automatically generated with medium confidence**

Enter password and To exit the MySQL console, type:

**Exit**

**A picture containing text, font, graphics, screenshot

Description automatically generated**

# STEP 3 – INSTALLING PHP

To install php-fpm and php-mysql packages at once, run:

**sudo apt install php-fpm php-mysql**

**A screenshot of a computer screen

Description automatically generated with medium confidence**

# STEP 4 — CONFIGURING NGINX TO USE PHP PROCESSOR

Create the root web directory for **your\_domain** as follows:

**sudo mkdir /var/www/projectLEMP**

****

Next, assign ownership of the directory with the $USER environment variable, which will reference your current system user:

**sudo chown -R $USER:$USER /var/www/projectLEMP**



Then, 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/projectLEMP**

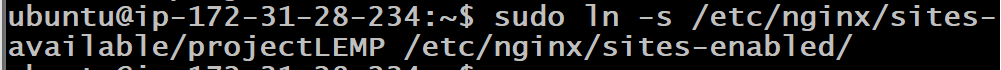
**A picture containing text, screenshot, font, software

Description automatically generated**

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

Activate configuration by linking to the config file from Nginx’s sites-enabled directory:

**sudo ln -s /etc/nginx/sites-available/projectLEMP /etc/nginx/sites-enabled/**

****

Test configuration for syntax errors by typing:

**sudo nginx -t**

**A picture containing text, font, screenshot

Description automatically generated**

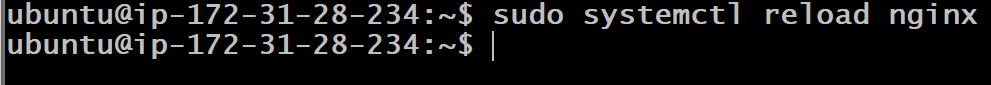
To disable default Nginx host that is currently configured to listen on port 80, for this run:

**sudo unlink /etc/nginx/sites-enabled/default**

****

Reload Nginx to apply the changes:

**sudo systemctl reload nginx**

****

The new website is now active, but the web root /var/www/projectLEMP 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 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**

**A screen shot of a computer

Description automatically generated with low confidence**

Now go to your browser and try to open your website URL using IP address:

**A picture containing text, screenshot, font, line

Description automatically generated**

# STEP 5 – TESTING PHP WITH NGINX

 Create a test PHP file in your document root. Open a new file called info.php within your document root in your text editor:

sudo nano /var/www/projectLEMP/info.php

info.php

**sudo nano /var/www/projectLEMP/**