PROJECT1

DEVOPS ENGR. 21/04/2023.

- LAMP STACK IMPLEMENTATION
- ➢ **Step 1 — Installing Apache and Updating the Firewall**
- ➢ To get the instance ID(EC2) connected to the ubuntu OS terminal(Install Apache using Ubuntu's package manager)
- ➢ ssh -i "Hill_EC2.pem" ubuntu@ec2-13-48-28-237.eu-north-1.compute.amazonaws.com.
  To update the firewall on the Apache web server

```
ubuntu@ip-172-31-11-37:~$ sudo apt update
Hit:1 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:5 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1031 kB]
Get:6 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [902 kB]
Fetched 2270 kB in 1s (2427 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
32 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Step 2 — Installing MySQL

I need to install a Database Management System (DBMS) to be able to store and manage data for your site in a relational database. MySQL.

ubuntu@ip-172-31-11-37:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server is already the newest version (8.0.32-0ubuntu0.22.04.2).
0 upgraded, 0 newly installed, 0 to remove and 18 not upgraded.
ubuntu@ip-172-31-11-37:~$ sudo mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
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>
```

It's recommended that you 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 you 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';
```

Exit the MySQL shell with:

```
mysql> exit
```

Start the interactive script by running:

```
$ sudo mysql_secure_installation.
```

Note: Enabling this feature is something of a judgment call. If enabled, passwords which don't match the specified criteria will be rejected by MySQL with an error. It is safe to leave validation disabled, but you should always use strong, unique passwords for database credentials.

Answer `Y` for yes, or anything else to continue without enabling.

```
VALIDATE PASSWORD PLUGIN 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 plugin?

Press y|Y for Yes, any other key for No: always say no as it will still prompt you to add
password before you login into my sql server.
```

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

```
$ sudo mysql -p
```

Notice the `-p` flag in this command, which will prompt you for the password used after changing the **root** user password. The password i used was We!come@$140

To exit the MySQL console, type:

```
mysql> exit
```

Notice that you need to provide a password to connect as the **root** user.

For increased security, it's best to have dedicated user accounts with less expansive privileges set up for every database, especially if you plan on having multiple databases hosted on your server.

Note: At the time of this writing, the native MySQL PHP library `mysqlnd` doesn't support `caching_sha2_authentication`, the default authentication method for MySQL 8. For that reason, when creating database users for PHP applications on MySQL 8, you'll need to make sure they're configured to use `mysql_native_password` instead.

> Your MySQL server is now installed and secured. Next, we will install PHP, the final component in the LAMP stack.

Step 3 — Installing PHP

You have Apache installed to serve your content and MySQL installed to store and manage your data. PHP is the component of our setup that will process code to display dynamic content to the end user. In addition to the `php` package, you'll need `php-mysql`, a PHP module that allows PHP to communicate with MySQL-based databases. You'll also need `libapache2-mod-php` to enable Apache to handle PHP files. Core PHP packages will automatically be installed as dependencies.

To install these 3 packages at once, run:

> sudo apt install php libapache2-mod-php php-mysql
> The you click or/enter button.
>
> Once the installation is finished, you can run the following command to confirm your PHP version:
> php -v
> PHP 8.1.2-1ubuntu2.11 (cli) (built: Feb 22 2023 22:56:18) (NTS)
> Copyright (c) The PHP Group
> Zend Engine v4.1.2, Copyright (c) Zend Technologies
> with Zend OPcache v8.1.2-1ubuntu2.11, Copyright (c), by Zend Technologies.

At this point, your LAMP stack is completely installed and fully operational.

- **L**inux (Ubuntu)
- **A**pache HTTP Server
- **M**ySQL
- **P**HP

Step 4 — Creating a Virtual Host for your Website using Apache

In this project, you will set up a domain called `projectlamp`, but you can replace this with any domain of your choice.

Apache on Ubuntu 20.04 has one server block enabled by default that is configured to serve documents from the **/var/www/html** directory.
We will leave this configuration as is and will add our own directory next next to the default one.

Create the directory for `projectlamp` using **'mkdir'** command as follows:

sudo mkdir /var/www/projectlamp

Next, assign ownership of the directory with your current system user:

sudo chown -R $USER:$USER /var/www/projectlamp

Then, create and open a new configuration file in Apache's `sites-available` directory using your preferred command-line editor. Here, we'll be using `vi` or `vim` (They are the same by the way):

sudo vi /etc/apache2/sites-available/projectlamp.conf

This will create a new blank file. Paste in the following bare-bones configuration by hitting on `i` on the keyboard to enter the insert mode, and paste the text:

```
<VirtualHost *:80>
    ServerName projectlamp
    ServerAlias www.projectlamp
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/projectlamp
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

To save and close the file, simply follow the steps below:

1. Hit the `esc` button on the keyboard

2. Type `:`
3. Type `wq`. **w** for `write` and **q** for `quit`
4. Hit `ENTER` to save the file

You can use the **`ls`** command to show the new file in the **sites-available** directory

sudo ls /etc/apache2/sites-available
000-default.conf default-ssl.conf projectlamp.conf
With this VirtualHost configuration, we're telling Apache to serve `projectlamp` using **/var/www/projectlampl** as its web root directory. If you would like to test Apache without a domain name, you can remove or comment out the options ServerName and ServerAlias by adding a **#** character in the beginning of each option's lines. Adding the **#** character there will tell the program to skip processing the instructions on those lines.

You can now use ***a2ensite*** command to enable the new virtual host:

sudo a2ensite projectlamp
You might want to disable the default website that comes installed with Apache. This is required if you're not using a custom domain name, because in this case Apache's default configuration would overwrite your virtual host. To disable Apache's default website use ***a2dissite*** command , type:

sudo a2dissite 000-default
To make sure your configuration file doesn't contain syntax errors, run:

sudo apache2ctl configtest
Syntax OK
Finally, reload Apache so these changes take effect:

sudo systemctl reload apache2
Your new website is now active, but the web root **/var/www/projectlamp** is still empty. Create an index.html file in that location so that we can test that the virtual host works as expected:

sudo echo 'Hello LAMP 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/projectlamp/index.html
Now go to your browser and try to open your website URL using IP address:

http://<Public-IP-Address>:80

If you see the text from ***'echo'*** command you wrote to index.html file, then it means your Apache virtual host is working as expected.
In the output you will see your server's public hostname (DNS name) and public IP address.
You can also access your website in your browser by public DNS name, not only by IP – try it out, the result must be the same (port is optional)

http://<Public-DNS-Name>:80
You can leave this file in place as a temporary landing page for your application until you set up an `index.php` file to replace it. Once you do that, remember to remove or rename the `index.html` file from your document root, as it would take precedence over an `index.php` file by default.

Step 5 — Enable PHP on the website

With the default **DirectoryIndex** settings on Apache, a file named `index.html` will always take precedence over an `index.php` file. This is useful for setting up maintenance pages in PHP applications, by creating a temporary `index.html` file containing an informative message to visitors. Because this page will take precedence over the `index.php` page, it will then become the landing page for the application. Once maintenance is over, the `index.html` is renamed or removed from the document root, bringing back the regular application page.

In case you want to change this behavior, you'll need to edit the **/etc/apache2/mods-enabled/dir.conf** file and change the order in which the **index.php** file is listed within the **DirectoryIndex** directive:

sudo vim /etc/apache2/mods-enabled/dir.conf

Then to clear every thing i(insert), esc, :(shift:), %d then insert again
<IfModule mod_dir.c>
    #Change this:
    #DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
    #To this:
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
After saving and closing the file, you will need to reload Apache so the changes take effect:

sudo systemctl reload apache2
Finally, we will create a PHP script to test that PHP is correctly installed and configured on your server.

Now that you have a custom location to host your website's files and folders, we'll create a PHP test script to confirm that Apache is able to handle and process requests for PHP files.

Create a new file named `index.php` inside your custom web root folder:

Sudo vim /var/www/projectlamp/index.php

Insert, esc, :%d which is just to clear things or just insert :q
This will open a blank file. Add the following text, which is valid PHP code, inside the file:

<?php
phpinfo();
:wq(save apache2 file)
When you are finished, save and close the file, refresh the page and you will see a page similar to this:

PHP Version 8.1.2-1ubuntu2.11

System	Linux ip-172-31-11-37 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	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
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-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.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-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.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	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled

Configuration

apache2handler

Apache Version	Apache/2.4.52 (Ubuntu)
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	projectlamp:0
User/Group	www-data(33)/33
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/etc/apache2
Loaded Modules	core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_authn_core mod_authn_file mod_authz_core mod_authz_host mod_authz_user mod_autoindex mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php mod_reqtimeout mod_setenvif mod_status

Directive	Local Value	Master Value
engine	On	On
last_modified	Off	Off
xbithack	Off	Off

This page provides information about your server from the perspective of PHP. It is useful for debugging and to ensure that your settings are being applied correctly.

If you can see this page in your browser, then your PHP installation is working as expected.

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 do so:

sudo rm /var/www/projectlamp/index.php
Then reload apache
sudo systemctl reload apache2

PROJECT2:

Step 1 – Installing the Nginx Web Server

Step 1 – Installing the Nginx Web Server

In order to display web pages to our site visitors, we are going to employ Nginx, a high-performance web server. We'll use the **`apt`** package manager to install this package.

Since this is our first time using `apt` for this session, start off by updating your server's package index. Following that, you can use `apt install` to get Nginx installed:

$ sudo apt update
$ sudo apt install nginx

sudo systemctl stop apache2

GIT PROJECT:
First of all install visual code editor and git
Then i choose to be using visual code editor terminal to configure and do all that needed.

Note that that there are difference between the GIT and GITHUB.
Git is a distributed version control and Github is the front end .

There are other front ends like github,gitlab,gitbuckect and in AWS we have code commit as another front end.
Also microsoft word is a front end as well as it's a word processor.

So, you create a folder from desktop and name it whatever you want in html or php, like i called my own ugwucohort1.html or php or txt etc.

So, when you open terminal
You either see th output PS C:\Users\Hillary> or PS C:\Users\Hillary\.Vscode>
When you see PS C:\Users\Hillary\.Vscode> you can go back to the normal terminal like PS C:\Users\Hillary> by running cd ..
PS C:\Users\Hillary> Then cd to our created folder
PS C:\Users\Hillary> cd .\desktop\ugwucohort1\
PS C:\Users\Hillary\desktop\ugwucohort1> git config ..global user.name "hillaru"
PS C:\Users\Hillary\desktop\ugwucohort1> git config ..global user.email "hillaru.ugwuanyi"
PS C:\Users\Hillary\desktop\ugwucohort1> git init
Initialized empty Git repository in C:/Users/Hillary/Desktop/ugwucohort1/.git/
At this juncture click on the explorer icon top left side of the terminal, then click on the created folder name like ugwucohort1 and it should turn green in colour.
Then write something in our html file like "this is my first my git class".
Run git init again and it will tell empty …..
Run git status to know exactly where we are in the
Run status
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ugwucohort1

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Hillary\desktop\ugwucohort1>
PS C:\Users\Hillary\desktop\ugwucohort1>
Untracked file(s) are the file that is existing and had been saved but has no interaction with it on git(ie have not done any commit).
Also we need to add git on the file and commit.
PS C:\Users\Hillary\desktop\ugwucohort1> git add
PS C:\Users\Hillary\desktop\ugwucohort1> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file: ugwucohort1

PS C:\Users\Hillary\desktop\ugwucohort1> git add .\ugwucohort1
PS C:\Users\Hillary\desktop\ugwucohort1> git commit -m "A new file ugwucohort1 created"
[master (root-commit) 45f6b96] A new file ugwucohort1 created
 1 file changed, 1 insertion(+)
 create mode 100644 ugwucohort1
PS C:\Users\Hillary\desktop\ugwucohort1> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\Hillary\desktop\ugwucohort1>
PS C:\Users\Hillary\desktop\ugwucohort1> git status
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified: ugwucohort1

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Hillary\desktop\ugwucohort1> git add .\ugwuchort1
fatal: pathspec '.\ugwuchort1' did not match any files
PS C:\Users\Hillary\desktop\ugwucohort1> git add .\ugwuchort.html
fatal: pathspec '.\ugwuchort.html' did not match any files
PS C:\Users\Hillary\desktop\ugwucohort1> git add .\ugwucohort1 (when i just want to add just a single change not all the changes already had.
PS C:\Users\Hillary\desktop\ugwucohort1> git commit -m "i added a new line"
[master 6dbf7fb] i added a new line
 1 file changed, 3 insertions(+), 1 deletion(-)
PS C:\Users\Hillary\desktop\ugwucohort1>.git log (shows all the commit that we have committed).
git remote add origin https://github.com/hillarug/hillarucohort1.git
git remote set url origin https://github.com/hillarug/hillarucohort1.git

Note all the commit we have been doing is on master branch.

So to create a new branch:

PS C:\Users\Hillary\desktop\ugwucohort1> git checkout -b prod
Switched to a new branch 'prod'
PS C:\Users\Hillary\desktop\ugwucohort1>
So, what happened here is that it created a new branch for me and the branch is called prod and it has switched me to the new brach called Prod.

PS C:\Users\Hillary\desktop\ugwucohort1> git status
On branch prod
nothing to commit, working tree clean
PS C:\Users\Hillary\desktop\ugwucohort1>
What this is telling me is that am now a prod branch not master branch and i have nothing to commit.
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified: ugwucohort1

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Hillary\desktop\ugwucohort1> git add .
PS C:\Users\Hillary\desktop\ugwucohort1> git commit -m "added master to line 5"
[master a7bdf25] added master to line 5
 1 file changed, 2 insertions(+), 2 deletions(-)
PS C:\Users\Hillary\desktop\ugwucohort1>
no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Hillary\desktop\ugwucohort1> git add .\ugwucohort1
PS C:\Users\Hillary\desktop\ugwucohort1> git commit -m "added ugwu"
[prod 38b6648] added ugwu
 1 file changed, 7 insertions(+), 2 deletions(-)
PS C:\Users\Hillary\desktop\ugwucohort1> git checkout master
Switched to branch 'master'
PS C:\Users\Hillary\desktop\ugwucohort1> git merge prod
Auto-merging ugwucohort1
CONFLICT (content): Merge conflict in ugwucohort1
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\Hillary\desktop\ugwucohort1>
PS C:\Users\Hillary\desktop\ugwucohort1> git log (This tells us all the commit we have committed with the commit ID)
commit 92c58ef46394b23510931372d06a53183e7c3bb3 (HEAD -> master)
Merge: a61bc79 38b6648
Author: hillaru <hillaru.ugwuanyi@gmail.com>
Date: Sat May 20 19:46:08 2023 +0100

        merge prod

commit 38b664852c29776ff9b8aab72881a8910f213988 (prod)
Author: hillaru <hillaru.ugwuanyi@gmail.com>
Date: Sat May 20 19:37:30 2023 +0100

The use esc(shift:):wq

Repository name: cohortgit/hillarucohort1
Owner: hillarug
Email: hillaru.ugwuanyi@gmail.com(Normal password)

Make sure you have a github account:

PS C:\Users\Hillary\desktop\ugwucohort1>
PS C:\Users\Hillary\desktop\ugwucohort1> git remote add origin
https://github.com/hillarug/cohortgit
PS C:\Users\Hillary\desktop\ugwucohort1>
PS C:\Users\Hillary\Desktop\ugwucohort1> git push -u origin master
info: please complete authentication in your browser…

Git pull will bring to your local repository from your online(URL) github account that you saved the edited file or added the new write up.

Then you will now authenticate your browser to continue

Also remember how to fork all the projects in the for instance DARY OLUFUNMILAYO githubaccount that you want to copy from

To bring to your local repository what is in the URL(Internet).
Just run git pull and enter.
Git clone https://github.com/hillarug/ci-pipeline-for-tooling-web.git.
You will find it in your desktop if it was desktop that you created it.
open integrated terminal that you .
: