**LAMP STACK PROJECT IMPLEMENTATION**

**The main aim for this project is to explain the DevOps concepts and processes using a LAMP web stack. Some developers use this set of framework and tools to develop a software products .We would be carrying out this project in the AWS platform**

**LAMP is an acronym of sets of technology used to develop a technical software product.**

**Linux**

**Apache**

**MySQL**

**PHP**

**Please note : (P could also stand for Python or Perl )**

**Apache server used is the apache2 version**

**Pre-requisite for the projects is the following.**

1. **Fundamental Knowledge of Installing and downloading software**
2. **Basic Understanding of Linux Commands**
3. **AWS account login with EC2 instance**
4. **Internet connection**

**IMPLEMENTATION STEPS:**

1. **Ensure you login with your details to your AWS console via the** [**https://aws.amazon.com**](https://aws.amazon.com)
2. **Click on the EC2 link to create instances.**

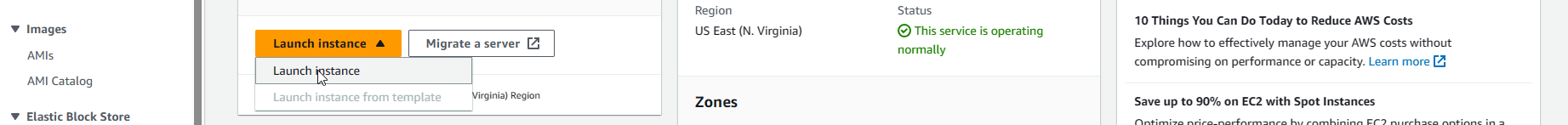
A screenshot of a computer

Description automatically generated

**iii)Click on launch instance dropdown button and select launch instance .**

A screenshot of a computer

Description automatically generated with medium confidence



**iv)Fill in all relevant details to the lamp project such as :**

**Type in the name and additional tag to the project (lamp) .Selected ubuntu from the quick start option .Also note that the Amazon machine image selection varies from user to user**

**Select Ubuntu server 22.04 LTS (HVM),SSD Volume Type (Free Tier )**

A screenshot of a computer

Description automatically generated

**v)The instance type selected in the configuration is the t2 micro -free tier.**

**Click on the “Create new key pair” link.**

**Ensure the Checkbox remains on the “Create security group”.**

A screenshot of a computer

Description automatically generated

**vi)Typed in the key pair name, chose the default key pair type and private key file format (rsa and .pem) and clicked the “Create key pair button”**

Screens screenshot of a computer screen

Description automatically generated with low confidence

**vii)The .pem file was downloaded successfully.**



**viii)I have deliberately chosen default settings to allow SSH traffic from anywhere as well as the storage volume given by AWS. Then proceed to launch our instance finally.**

A screenshot of a computer

Description automatically generated with medium confidence

**ix)Instance successfully launched.**

A screenshot of a computer

Description automatically generated with medium confidence

**x)Select checkboxes to view more details about the instance created.**A screenshot of a computer

Description automatically generated with medium confidence

**The public IP address shown on the screenshot should be copied as we would be using it on the console.**

**Open git bash on visual studio code or whichever console is convenient to use .**

**We are using git bash here with Visual Studio Code**

A picture containing text, screenshot, font

Description automatically generated

**Type YES ,to connect**

A picture containing text, screenshot, font

Description automatically generated

**You have successful connected to the EC2 instance launched on AWS via ssh**

**Type clear to have a clear console and proceed to updating the lists of packages in the package manager**



**Then we run apache2 installation and click yes to complete installation**



**We have to verify that Apache is running in our Operating System.**

A picture containing text, screenshot, font

Description automatically generated

**To proceed by launching the web server in the AWS Cloud, we need to navigate back to the security group on the platform to add a new rule for TCP port 80 which is the default for web browsers .**

**Once done we can access the web page on internet.**

**Click on security button.**

A screenshot of a computer

Description automatically generated with low confidence

**And click the security group link**

A screenshot of a computer

Description automatically generated with medium confidence

**Click on “Edit inbound rules “ in order to add a new rule for port 80**

A picture containing text, screenshot, line, font

Description automatically generated

**Add a new rule** A screenshot of a computer

Description automatically generated with low confidence

**Type in the port range and click “Anywhere ipv4”**

A screenshot of a computer

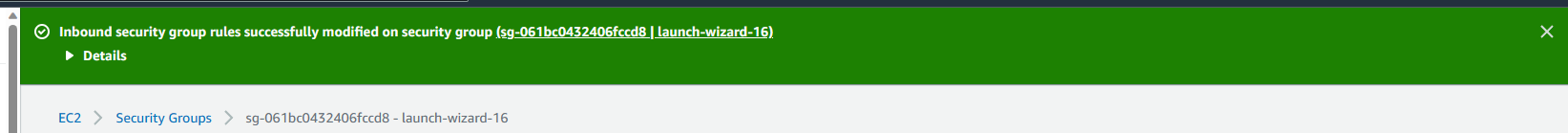
Description automatically generated with medium confidence

**Click the “Save rules” Button**

A screenshot of a computer

Description automatically generated with medium confidence

**Inbound rule successfully modified.**



**Open any browser of your choice and access the URL** [**http://34.201.134.152:80**](http://34.201.134.152:80)

A screenshot of a computer

Description automatically generated with medium confidence

**Apache2 default page successfully displayed.**

**From the LAMP stack, we have implemented with Linux and got Apache ready .**

**Next step would be to get the MySQL installed.**

**MYSQL INSTALLATION**

**Now that our web server is running, we need a relational database uses within the PHP environment hence we install MySQL server**

**Type “Y” and enter.**

A black background with white text

Description automatically generated with low confidence

**When installation is finished, Log in to connect to the MySQL server as the administrator user root so that you can have access to the sudo command.**

A screenshot of a computer program

Description automatically generated with medium confidence

**It is important to set up a password for the user root using mysql\_native\_password as a default authentication method.**

**Please note, Password not revealed for security purpose**

**Exit MySQL**

A screen shot of a computer

Description automatically generated with medium confidence

**Interactive script is started, and all modifications are answered with a Y/N response**

**Root user password was set Validate password: No**

**Change password: No**

**Remove anonymous user: No**

A screenshot of a computer

Description automatically generated

**Disallow remote login: No**

**Remove test data base and access to it: No**

**Reload Privilege tables: Yes.**

A screenshot of a computer

Description automatically generated

**Verify login details to ensure all details were inputted correctly and exiting MySQL**

A screenshot of a computer screen

Description automatically generated with medium confidence

**MySQL server was correctly installed and secured.**

**Next, we proceed to the PHP installation which is the final component of the LAMP STACK**

**PHP INSTALLATION**

**PHP is the component that would process the codes to display dynamic content to the end user. Hence, we would need to install 3 packages namely :**

**1)PHP package 2) libapache2-mod-php 3) php-mysql .**

A black screen with white text

Description automatically generated with low confidence

**Installation continues.**

A screen shot of a computer program

Description automatically generated with low confidence

**After installing, we check the PHP version.**

A screenshot of a computer screen

Description automatically generated with low confidence

**At this point the LAMP STACK implementation is completed and fully operational**

**We need to test our set up with a PHP script and this needs a proper APACHE virtual host to keep your website files and folder .Multiple website can be hosted on a single machine and the users would not notice**

**CREATING AN APACHE VIRTUAL HOST FOR OUR WEBSITE TO USE .**

**Next step, making a directory for the site directory, running below**



**Then proceed to edit a new site directory to input the virtual host information.**



**Put the edited file in an insert mode by typing “i” without quotes and add the config files, press ESC ,save and exit with “ :wq” command**

A picture containing text, screenshot, font

Description automatically generated

**Next check the content of the sites-available directory and you would see 3 configurations files on here .**



**With this configuration files, we would need to DISABLE the 000-default config file and ENABLE the new directory we created using the following command**

A screen shot of a computer

Description automatically generated with medium confidence

**After enabling and disabling done successfully, we would verify that there are no syntax errors with the command below**



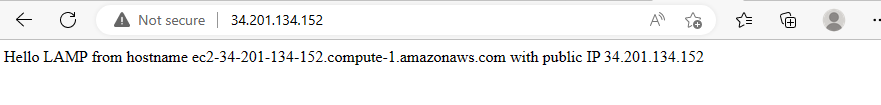
**Then we proceed by reloading the Apache server to make these changes take effects.**



**The new website is now active but the projectlamp has empty file .We create an index.html file in that location so that we can test our virtual host is performing as expected .**

A screenshot of a computer

Description automatically generated with medium confidence

**Proceed to the browser and open the previous website using the ip address** 

**Echo successfully displayed but this is just to test the website.**

**Type “clear” command to clear screen.**

**ENABLE PHP ON THE WEBSITE**

**We would need to set up an index.php file to replace the index.html file from the document root as it needs to override the default settings. This is a very useful maintenance page in PHP application**

A black background with white text

Description automatically generated with low confidence

**Files are edited correctly while index.php and index.html are in that order respectively.**

A screen shot of a computer

Description automatically generated with medium confidence

**Edited successfully and the Apache needs to be reloaded again by the command below.**



**Finally we would create the PHP script to test that PHP is correctly installed and configured on the server .The importance is to be able to handle and process request for PHP files with the command below**

A black screen with white text

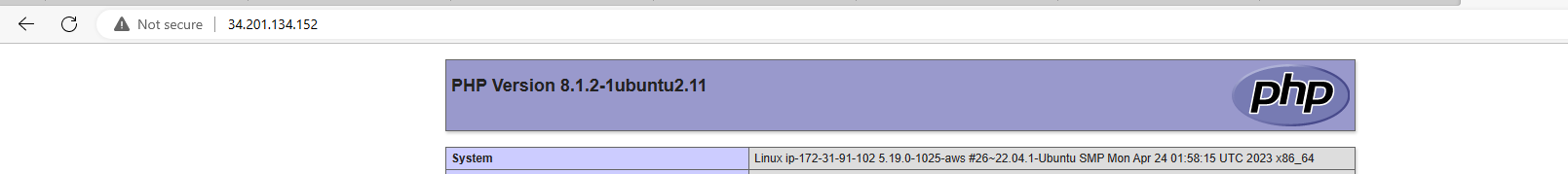
Description automatically generated with low confidence

**Put the edited file in an insert mode by typing “i” without quotes and add the valid PHP code files, press ESC ,save and exit with “ :wq” command**

A picture containing font, graphics, screenshot, design

Description automatically generated

**Refresh the web page and you would see the web page server in a PHP perspective.**



**This is the minimum requirement to set up an AWS instance with LINUX ,APACHE,MYSQL AND PHP for a web project.**

**Please note: Remember to terminate your EC2 instance.**