**LEMP STACK PROJECT IMPLEMENTATION**

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

**LEMP is an acronym of sets of technologies used to develop a technical software product.**

**Linux**

**NGINX**

**MySQL**

**PHP**

**(P could also stand for Python or Perl )**

**NGINX is pronounced engineX where the acronym “E” originated from**

**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.**

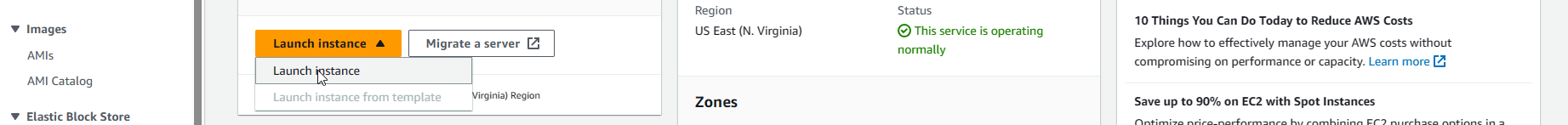
A screenshot of a computer

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 LEMP project such as :**

**Type in the name and additional tag to the project (lemp) .Select 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 with medium confidence

**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 unchanged on the “Create security group”.**

A screenshot of a computer

Description automatically generated

**vi)Type 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”**

A screenshot of a computer

Description automatically generated with medium confidence

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

A screenshot of a computer

Description automatically generated with medium confidence

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

**Then we proceed to launch our instance finally.**

A screenshot of a computer

Description automatically generated with medium confidence

**Instance successfully launched and click to view Instance details with the IP address.**

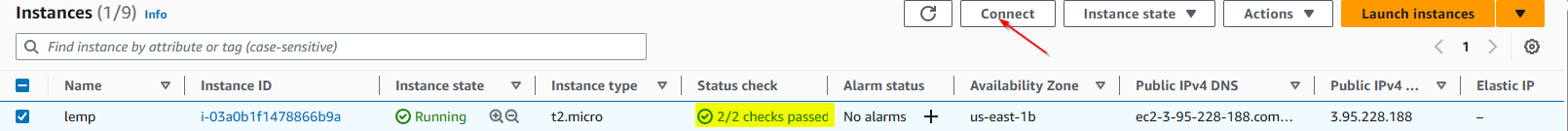
A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

**Click the “Connect” button and copy the ssh client details we would be using on the git bash 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 screenshot of a computer

Description automatically generated with medium confidence

**Type YES to connect.**

A screen shot of a computer

Description automatically generated with medium confidence

**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.**

A screenshot of a computer screen

Description automatically generated with low confidence

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

A screenshot of a computer

Description automatically generated with medium confidence

**Type YES to continue NGINX installation.**



**We have to verify that Nginx is running in our Operating System and press the**

**Ctl + C button to get the ubuntu root.**

A picture containing text, screenshot, font, software

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 screenshot of a computer

Description automatically generated with medium confidence

**Add a new rule.**

A close-up of a computer screen

Description automatically generated with low confidence

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

A screenshot of a computer

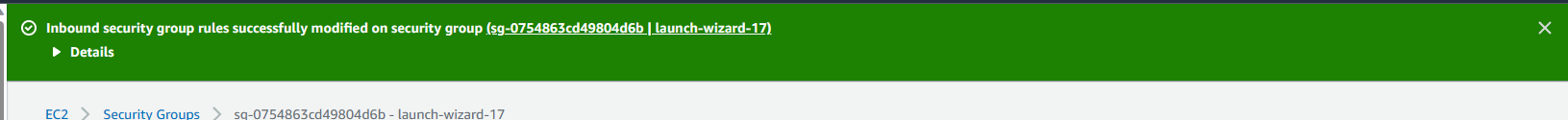
Description automatically generated with medium confidence

**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://3.95.228.188:80**

A picture containing text, font, screenshot

Description automatically generated

**Nginx default page successfully displayed.**

**From the LEMP stack, we have implemented with Linux and now have Nginx 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 press 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

Description automatically generated

**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 black screen with white text

Description automatically generated with low 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 screen

Description automatically generated with medium confidence

**Disallow remote login: No**

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

**Reload Privilege tables: Yes**

A screenshot of a computer error

Description automatically generated with low confidence

**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 is now installed and secured .**

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

**PHP INSTALLATION**

**PHP is the component that would process the codes to display dynamic content to the end user. Nginx requires an external program to handle the PHP processing and this act as a bridge between PHP interpreter and the web server .This enhances the overall performance for most PHP web based site .It is called the PHP fastCGI process manager)**

**Hence ,we would need to install 2 packages namely :**

**1)php-fpm 2) php-mysql .**

A black background with white text

Description automatically generated with low confidence

**Installation continues.**

A screenshot of a computer program

Description automatically generated with medium confidence

**At this point the PHP component is installed but we need to configure the Nginx to use the PHP processor**

**CONFIGURING NGINX TO USE PHP PROCESSOR**

**We need to create server block to encapsulate the Nginx configuration details and it can host more than one domain on a single server .Therefore we would create a direct structure with the /var/www for the domain website .**

**Root web directory created for our domain below.**

A picture containing text, screenshot, font, line

Description automatically generated

**Then proceed to open new nginx in sites-available directory using the vim editor**



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

A screen shot of a computer program

Description automatically generated with low confidence

**Please observe the following functions the location block carries out:**

**listen: Port Nginx listen to port 80**

**root: defines where the document root is served and stored by the website**

**index: defines how Nginx prioritize the index files. Index .html file gets the highest priority than the rest of the indexes.**

**Server-name: The domain name or Ip address the server block is responsible for .**

**Location: Includes try file directives that checks the files that correlates with the matching URI request. If not found, it will return Error 404**

**Location ~\.php$: Handles the PHP processing and ensures it points Nginx to the fastcgi-php.conf files and declares what socket is associated with php-fpm**

**Location ~ /\.ht : deals with .htaccess files that Nginx does not process by denying all directives and ensuring they are not served to users .**

**Now we have to activate our configuration by linking the config files from Nginx site –enabled directory and also test the configuration to know if the syntax are OK**

A black screen with white text

Description automatically generated with low confidence

**Test successful and syntax are okay.**

**Next step is disabling default nginx host that is currently configured to listen to port 80.**

A black screen with white text

Description automatically generated with low confidence

**Proceed to reload Nginx to apply all changes.**



**Hence , Our new website is active but note that the /var/www/projectLemp is empty.**

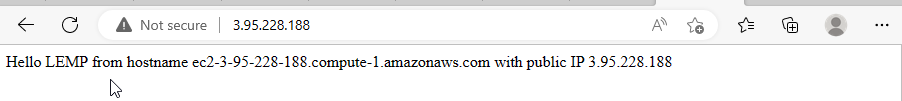
**We should create a file in that location so that we can test that the blocks is working as expected .**

A screen shot of a computer

Description automatically generated with medium confidence

**Now go and launch your browser URL using the IP address.**

[**http://3.95.228.188:80**](http://3.95.228.188:80)



**Our LEMP stack is fully configured successfully.**

**Next would be to create a PHP Script to test that our Nginx can handle .php files within the new configured website.**

**TESTING PHP WITH NGINX**

**Our LEMP stack is set up and completely installed and fully operational. We would test to validate that Nginx can handle .php files off to our PHP processor.**

**Create a new file called info.php and paste the file**



**Paste the simplest valid php code that would return information about your server**

A picture containing screenshot

Description automatically generated

**Let us access this page on our web browser with the endpoint /info.php**

[**http://3.95.228.188/info.php**](http://3.95.228.188/info.php)

A screenshot of a computer

Description automatically generated with low confidence**A webpage containing details information about our server should be successfully displayed .**

**Please note: After checking the relevant information about your php .Its best to remove the file created as it contains sensitive information about your PHP environment and your ubuntu server by the command below**

**sudo rm /var/www/your\_domain/info.php**

**Next ,we would be retrieving data from MySQL database with PHP**

**RETRIEVING DATA FROM MYSQL DATABASE WITH PHP**

**We would be creating a test database with a simple To-do list and configure to access it ,so the Nginx would be able to query data from the database and display it .**

**A new user would be created with the “mysql\_native\_password” in order for it to connect to the MySQL database from PHP**

**Our new user: example\_user**

**Our new database: example\_database**

**We would connect to the root account by the command below**

A screenshot of a computer

Description automatically generated

**Then we create a new database called “example\_database” and password.**

A screen shot of a computer code

Description automatically generated with low confidence

**We grant all permissions to the new user over the new “example\_database” in order to give the new user full privileges.**

**We exit the shell and test if the new user has actually been granted permission to be able to login to the MySQL console again and display the database.**

A black background with green numbers

Description automatically generated with low confidence

A screenshot of a computer screen

Description automatically generated with medium confidence

**Next, we create a test table named “todo\_list” and input 4 different values on each section**

A picture containing text, screenshot, font, line

Description automatically generated

A picture containing text, screenshot, font

Description automatically generated

**We confirm that the data were successfully saved to our table below and exit the table.**

A screenshot of a computer program

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with low confidence

**Next, we create a PHP Script that would connect to the MySQL database for our content in out root directory using vim editor to input the PHP script in the todo\_list.php**

A picture containing text, screenshot, font, line

Description automatically generated

**Save and close file when done editing.**

A picture containing text, screenshot, software, display

Description automatically generated

**We can now access the page in the browser by visiting the IP address configured for the website follow by the endpoint /todo\_list.php**

[**http://3.95.228.188/todo\_list.php**](http://3.95.228.188/todo_list.php)

A screenshot of a computer

Description automatically generated with medium confidence

**Test table have been successfully displayed on the webpage and our PHP environment has connected and was able interact with our MySQL server**

**Congratulations.**