

# Develop and Deploy Php Application

## (LAB-M03-01)

Version Control	
Document	Develop and deploy Php application
Owner	Ahmad Majeed Zahoory
Version	3.1
Last Change	22 <sup>nd</sup> May 2024
Description of Change	Task steps updated

**Lab duration:** 30 minutes

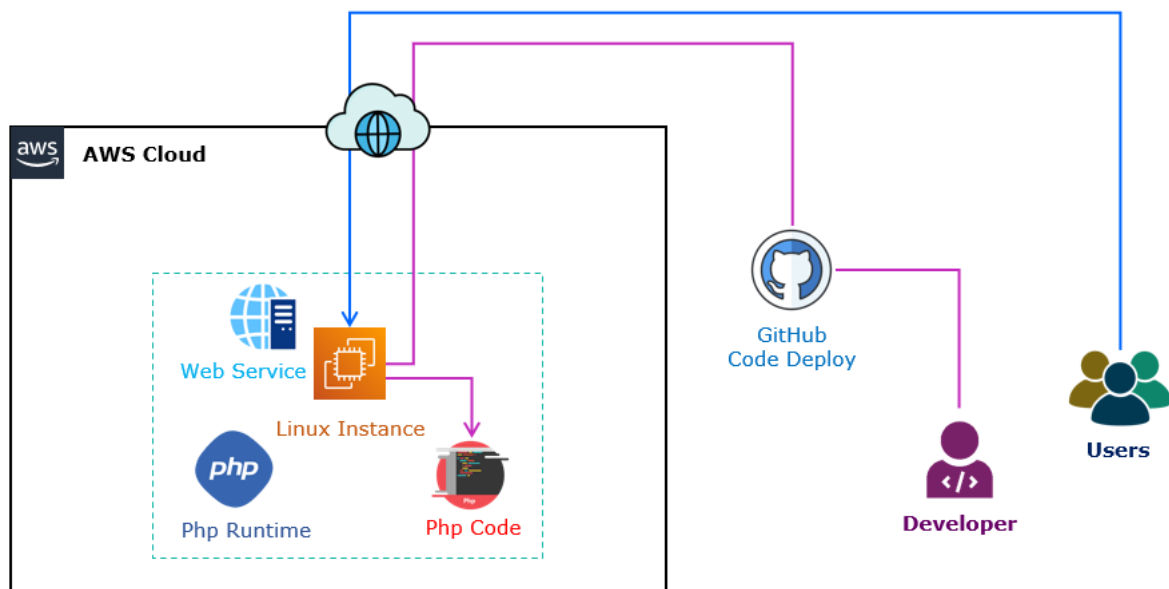
### Lab scenario

You're preparing to deploy a web application in AWS. As a development group, your team has decided to use the Php application to deploy in the Linux environment in AWS.

### Objectives

After you complete this lab, you will be able to:

- Develop the Php code.
- Create the Linux virtual machine.
- Build the Run-time environment.
- Deploy the Php code.
- Access Web application server.



## Task 1: Develop the Php Code

In this task, you will develop the Php code to display the server IP address.

### Step 1: Develop the Code to Display the Server IP Address

1. Unzip the **LAB-03-01-Code.zip** (Php code).

**Note:** **lab-03-01-code.zip** code is available with the **Lab manual**.

- a. Open the **index.php** in the **Notepad**.
  - i. **Add** the **code** after **TO DO** to **Display** the **Server Private IP Address**.

#### Info:

- a. You can also use the below code to **display the Server Private IP Address**.

```
<p class="ui header center"><font color="white">Server IP Address <?=$_SERVER['SERVER_ADDR'] ?></p>
```

- b. Add the above code below to **<! TODO >** in the index.php.

```
banner -->
<section id="banner">
  <h2>AWS Developer LAB</h2>
  <! TO DO >
  <p class="ui header center"><font color="white">Server IP Address <?=$_SERVER['SERVER_ADDR'] ?></p>
  <ul class="actions">
    <li><a href="#" class="button special big">Get Started</a></li>
  </ul>
</section>
```

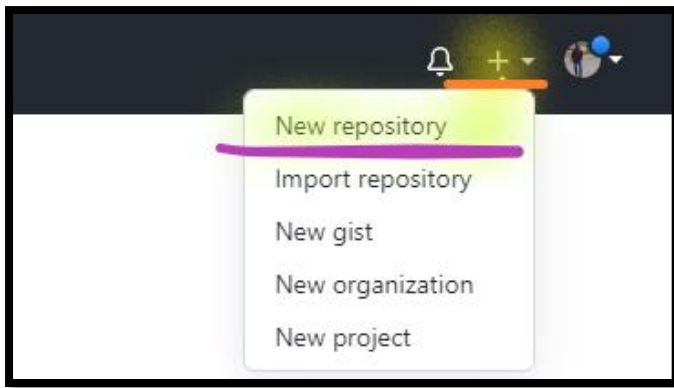
- a) Select the **File**.
- b) Select **Save**.

### Step 2: Upload the Code in GitHub

2. Open your **GitHub account**.

**Note:** If you **don't have GitHub account**, follow the **Create GitHub Account.txt**, available with the Lab manual.

- a. Select the **+** sign.



- i. From the **Create a new repository** page:
  - a) **Repository name:** Write **lab-03-01**.
  - b) Select **Public**.

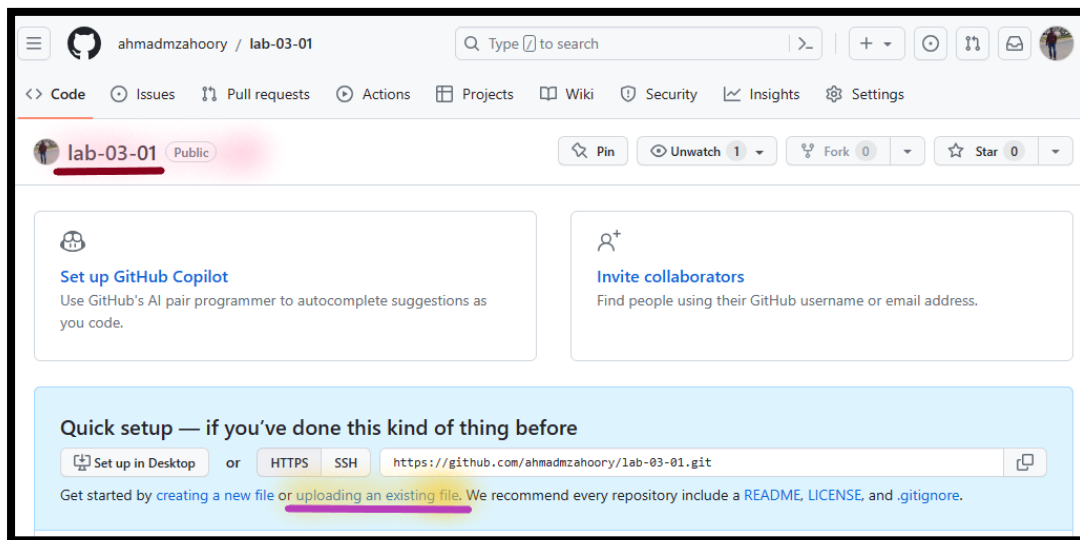
A screenshot of the 'Create a new repository' page on GitHub. The form includes fields for 'Owner' (set to 'ahmadmzahoory') and 'Repository name' (set to 'lab-03-01'). A green checkmark indicates 'lab-03-01 is available'. Below the name field, there is a suggestion for 'animated-octo-succotash'. The 'Description' field is optional and empty. At the bottom, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option is highlighted with an orange underline.

- c) Select the **Create repository**.

**Note:** You can see the **lab-03-01** repository page.

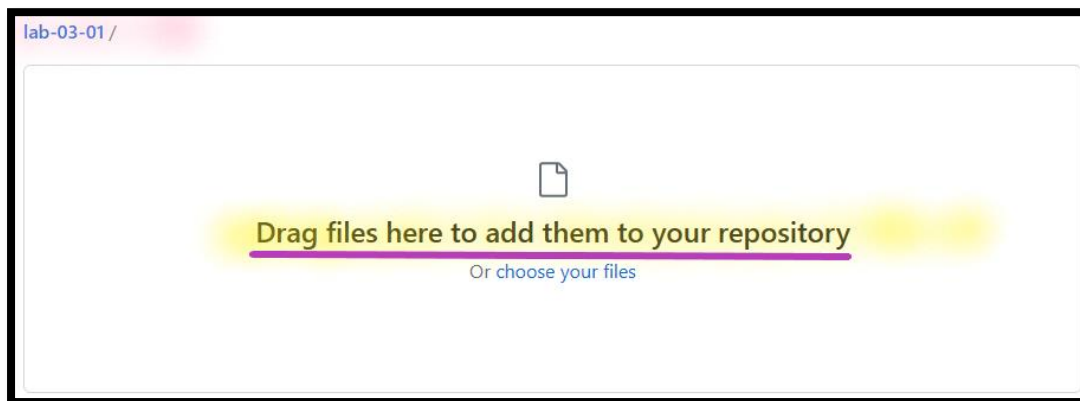
b. From the **lab-03-01 repository**:

i. Select the **Uploading an existing file**.

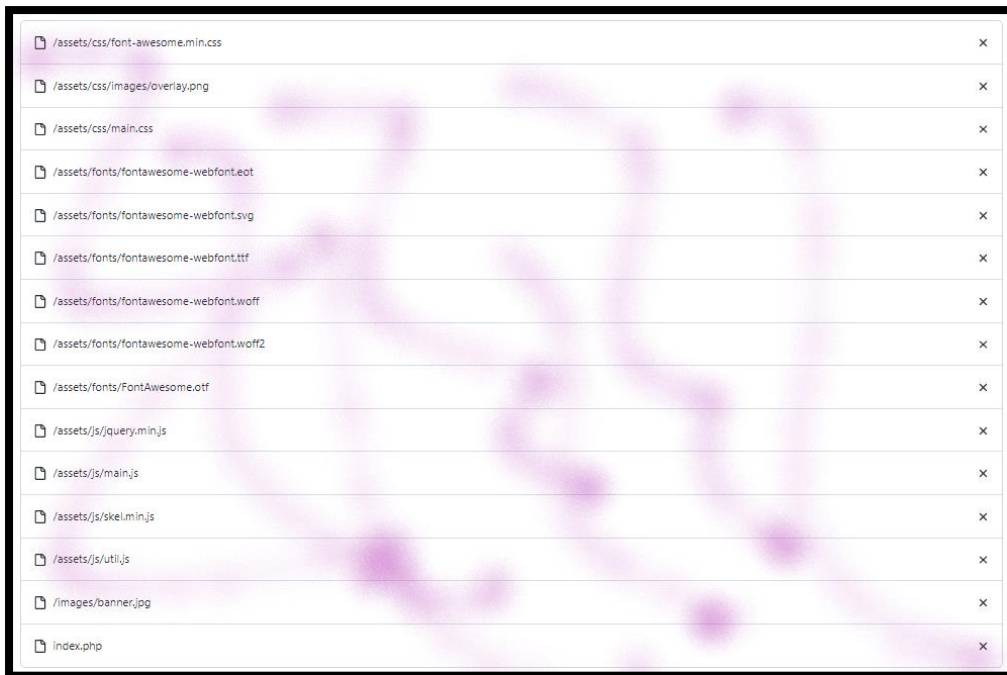


ii. **Drag and drop** the **Code** in the **GitHub Repository**.

**Note:** You need to **Upload** the **folder** and **files**, **not zip file**.



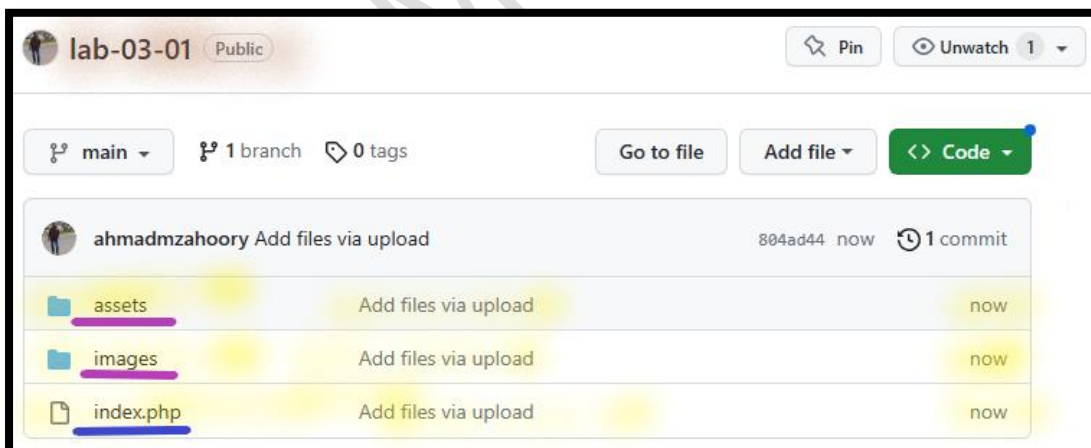
**Note:** You can see the **files to be uploaded**.



iii. Select **Commit changes**.

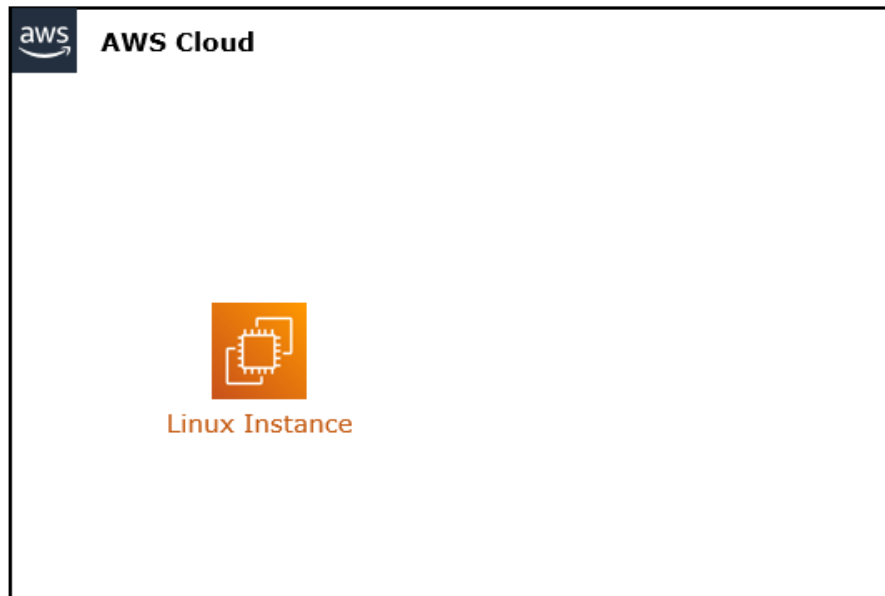
**Note:** After code **uploaded successfully**, you can see them in repository.

**Note:** You can see the **assets** & **images folder** and **index.php** file.



## Task 2: Create Linux Server

In this task, you will launch an Amazon EC2 instance using the management console to deploy the Php code.



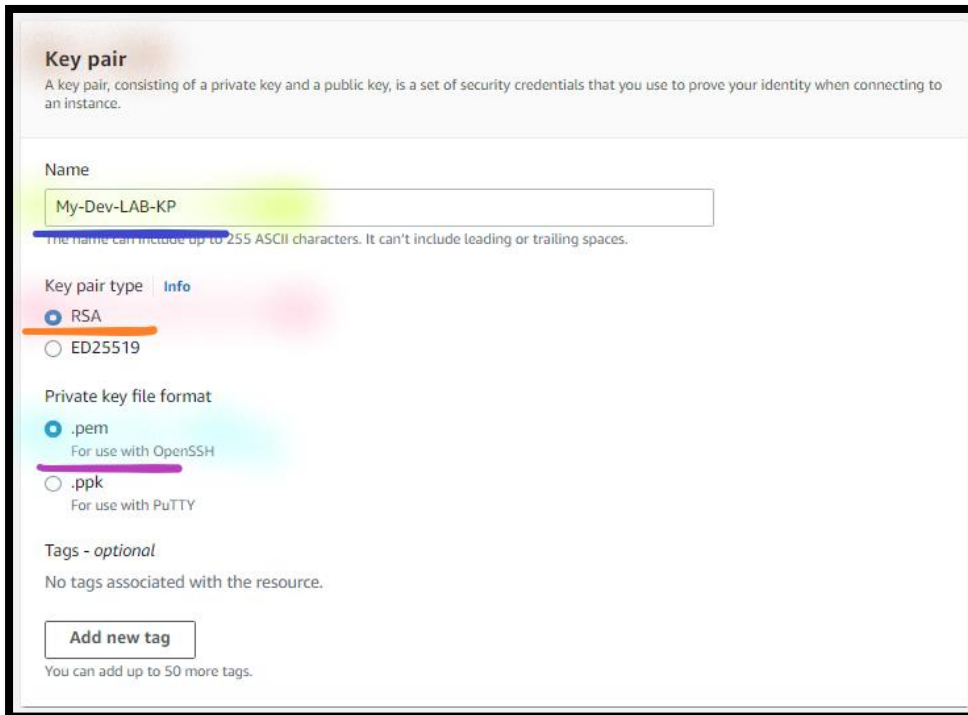
### Step 1: Create Key Pair

3. In the **AWS Management Console**, on the **Services** menu, search and select **EC2**.
4. Choose the **YOUR ALLOCATED REGION**, region list to the right of your account information on the navigation bar.



5. Select **Key pairs**.
  - a. Select **Create key pair**.
    - i. **Name**: Write **My-Dev-LAB-KP**.
    - ii. **Private key file format**: Select **pem**.

**Note:** Leave other details as default.



**Key pair**

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

My-Dev-LAB-KP

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)

☒ RSA

☐ ED25519

Private key file format

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

Tags - optional

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

- iii. Select **Create key pair**.

**Note:** **My-Dev-LAB-KP.pem** downloaded in your **Local Laptop/Desktop**.

## Step 2: Launch Linux Instance

6. **From** the **EC2** console.
7. Select **Instances**.
8. Select **Launch Instances**.
  - a. In the **Name and tags** section:
    - i. **Name:** Write **Linux Web Server**.



**Name and tags** [Info](#)

Name

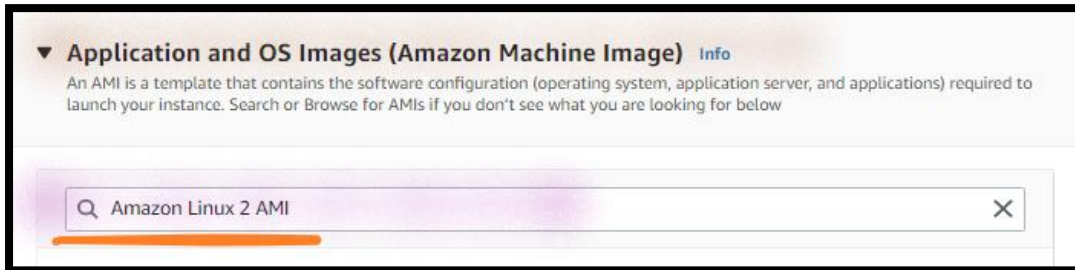
Linux Web Server

[Add additional tags](#)

b. In the **Application and OS Images** section:

i. In the **Search box**:

a) Type **Amazon Linux 2 AMI**.

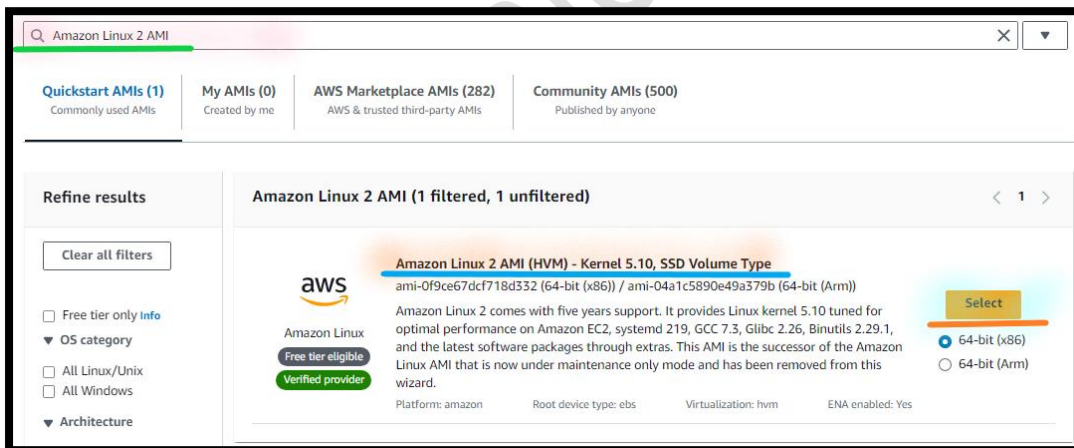


b) Press **Enter** key.

**Note:** You can see the **Choose an Amazon Machine Image** page.

c) **From** the **Choose an Amazon Machine Image** page:

1) Select **Amazon Linux 2 AMI**.



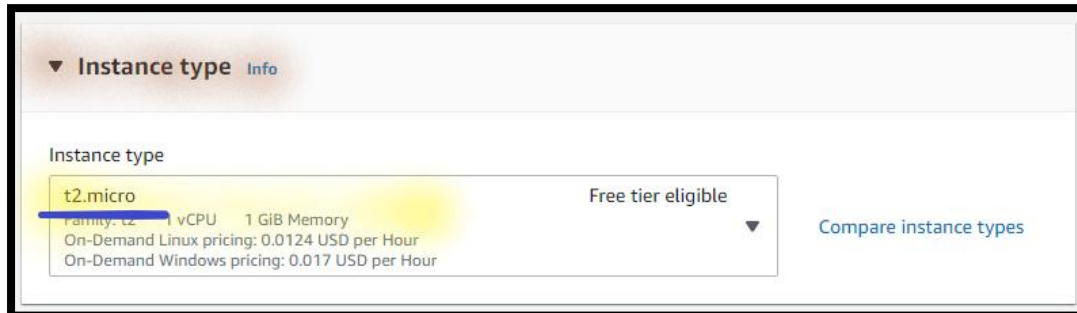
**Note:** You can see the **Launch an Instance** page.



c. In the **Instance Type** section:

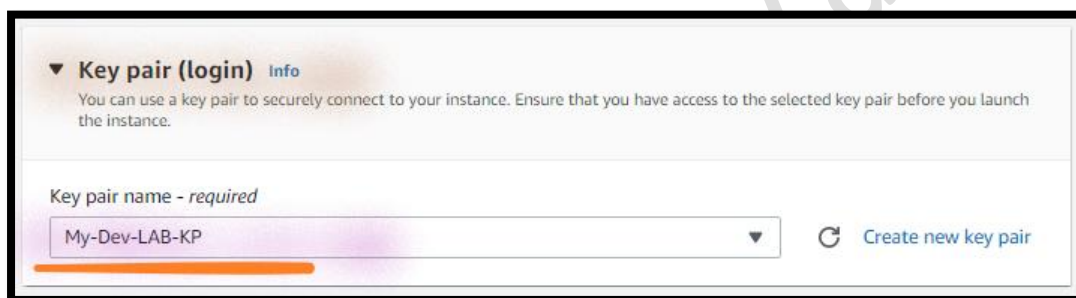
i. **Instance type:** Dropdown and in the **Search box:**

a) Type and select **t2.micro**.



d. In the **Key pair (login)** section:

i. **Key pair name:** Dropdown and select **My-Dev-LAB-KP**.



e. In the **Network setting** section:

**Note:** You can see "**Allow SSH traffic**" is already **enabled** from "**Anywhere**".

i. **Firewall:** Select **Create security group**.

a) **Allow HTTP traffic from the internet:** **Enable** the **Checkmark**.

**Note:** Leave the other details as default.

▼ Network settings Info Edit

Network Info  
vpc-07861f65386e98997

Subnet Info  
No preference (Default subnet in any availability zone)

Auto-assign public IP Info  
Enable

Firewall (security groups) Info  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

☒ Allow SSH traffic from  
Helps you connect to your instance. Anywhere  
0.0.0.0/0

☐ Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

f. In the **Summary** section:

i. Select **Launch Instances**.

▼ Summary

Number of instances Info  
1

Software Image (AMI)  
Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD  
Volume Type  
ami-09d3b3274b6c5d4aa

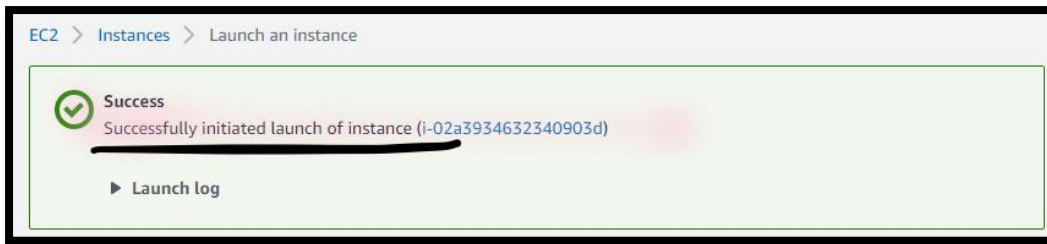
Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

Cancel Launch Instance

**Note:** Wait, till you can see the message "Successfully initiated launch of instance".

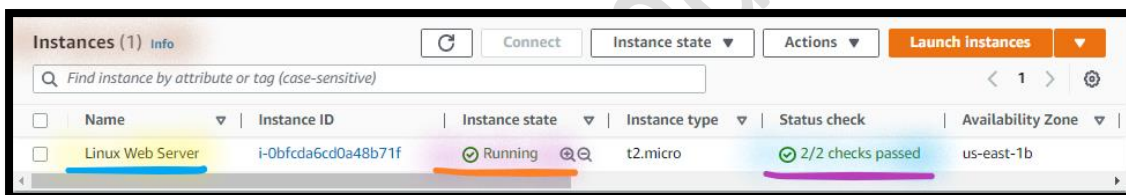


g. Select **View all instances**

**Note:** **Wait**, till you can see the **Linux Web Server** Instance **State** is **Running**.

**Note:** **Wait**, till you can see the **Linux Web Server** Instance **Status check** is **2/2 check passed**.

**Note:** **Refresh** your **screen** unless you can see the **2/2 check passed**.



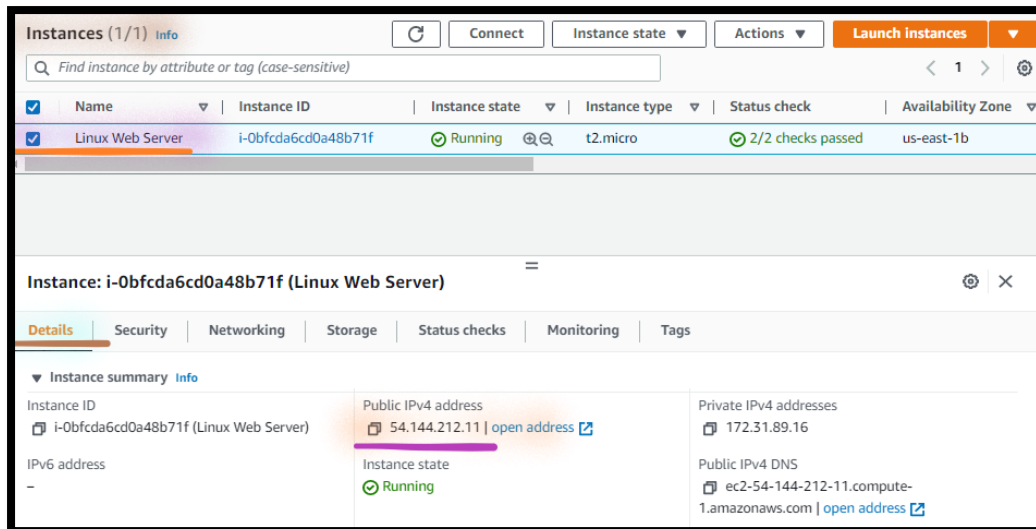
### Task 3: Connect to Linux Web Server

In this task, you will log into the Linux web server.

#### Step 1: Copy the IP Address of Linux Web Server

9. **From** the **EC2** console.
10. Select the **Linux Web Server**.
  - a. Select the **Details**.

**Note:** **Copy** the **Public IP address** of **Linux Web Server** in the **Notepad**.



## Step 2: Connect to Linux Web Server Instance

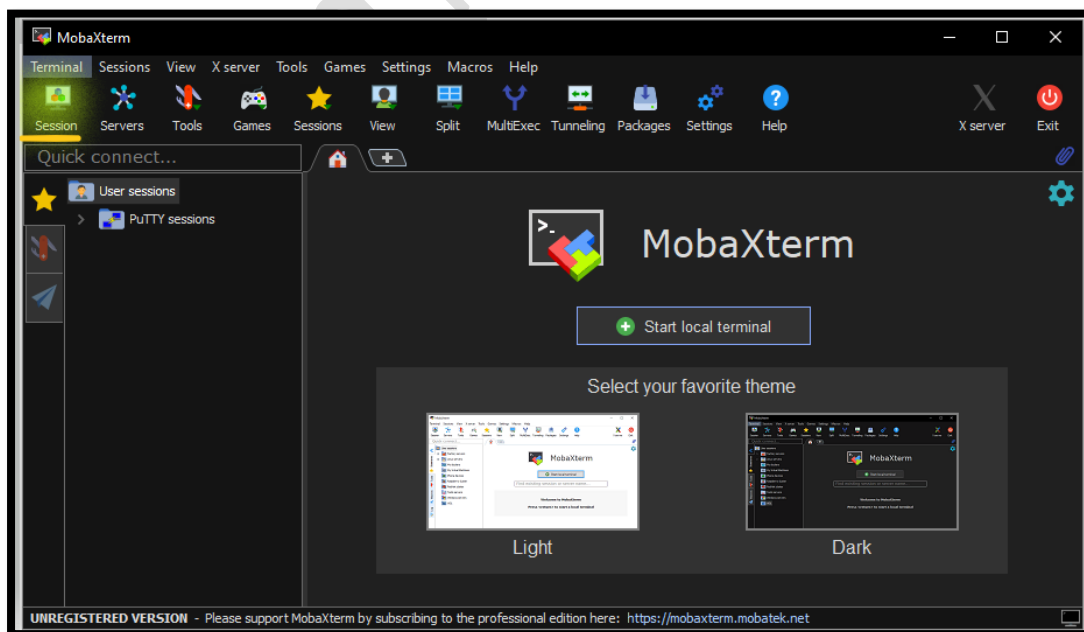
11. From the **Local Desktop/ Laptop** (Windows Desktop), **Download** the **MobaXterm** (**Portable edition**).

<https://mobaxterm.mobatek.net/download-home-edition.html>

12. From the **Local Desktop/ Laptop** (Windows Desktop), **Open** the **MobaXterm**.

13. From the **MobaXterm**.

- a. Select **Session**.



b. Select **SSH**.

i. Select **Advanced SSH settings**.

a) **Remote host**: Write **Public IP address** of the **Linux Web Server**.

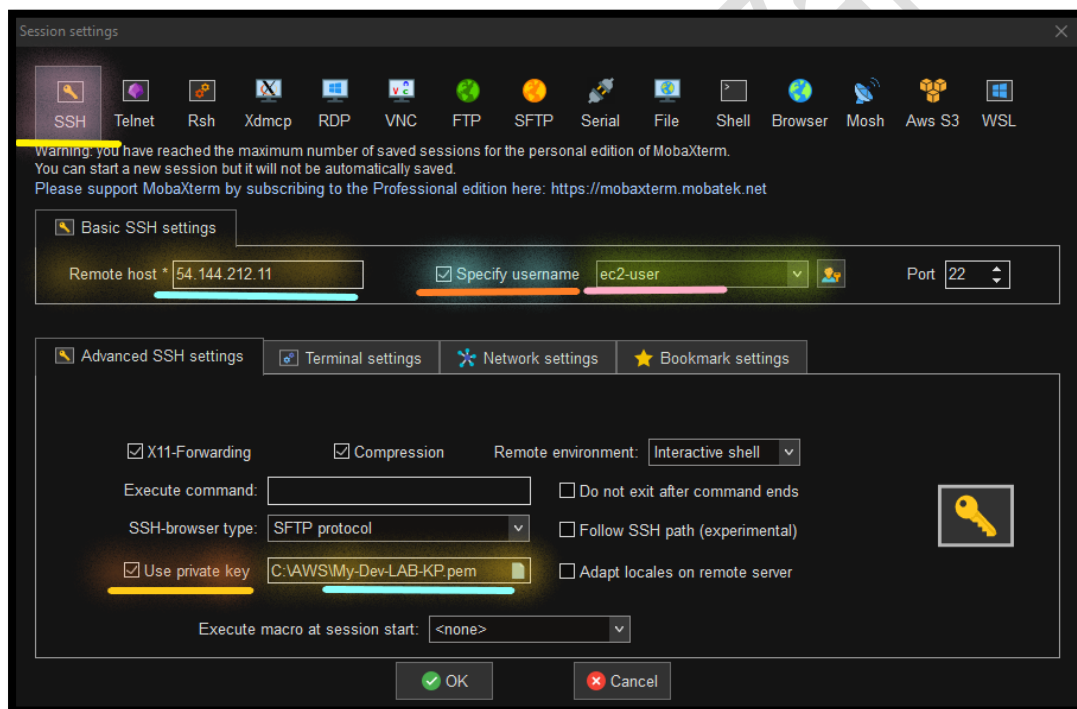
b) **Specify username**: **Enable** the **Checkmark**.

c) **Specify username**: Write **ec2-user**.

d) **Use Private key**: **Enable** the **Checkmark**.

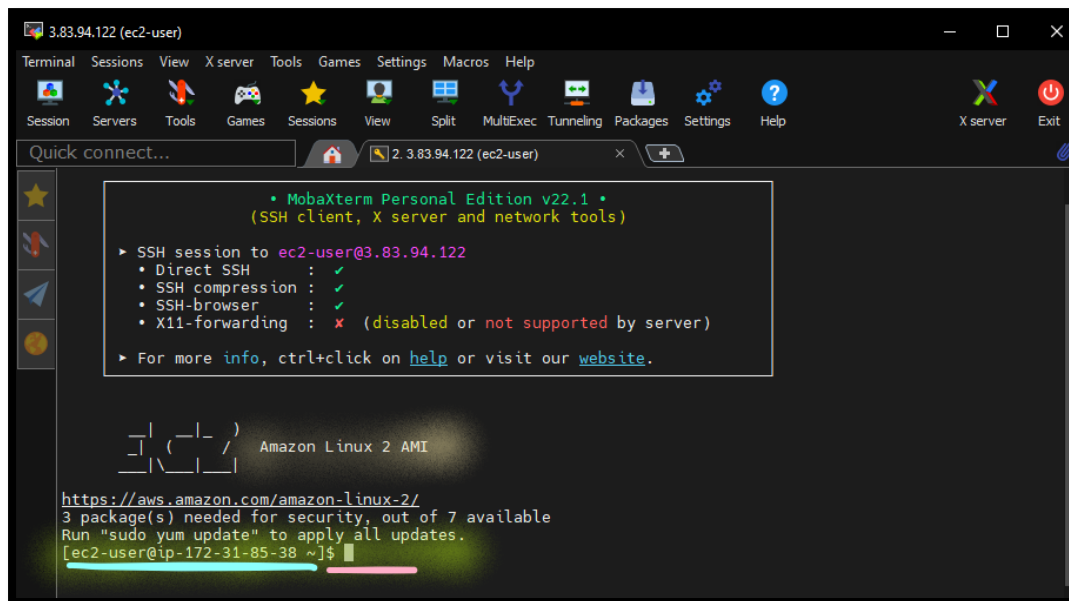
e) **Use Private key**: Click on the **Search box**:

1) **Navigate** and **select** the **My-Dev-LAB-KP.pem**.



f) Select **Ok**.

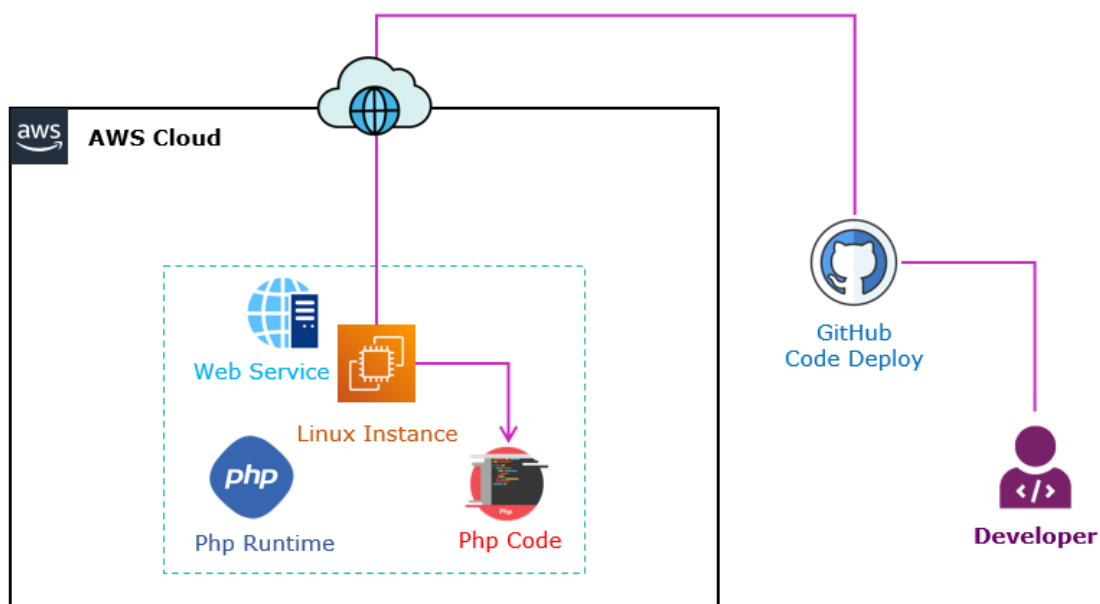
**Note:** You can see the **Linux Console**.



**Note:** Go to the next task, But **Don't close** the **Linux terminal**.

## Task 4: Deploy the Php App Code

In this task, you will install the web service and Php run-time environment to deploy the php code.



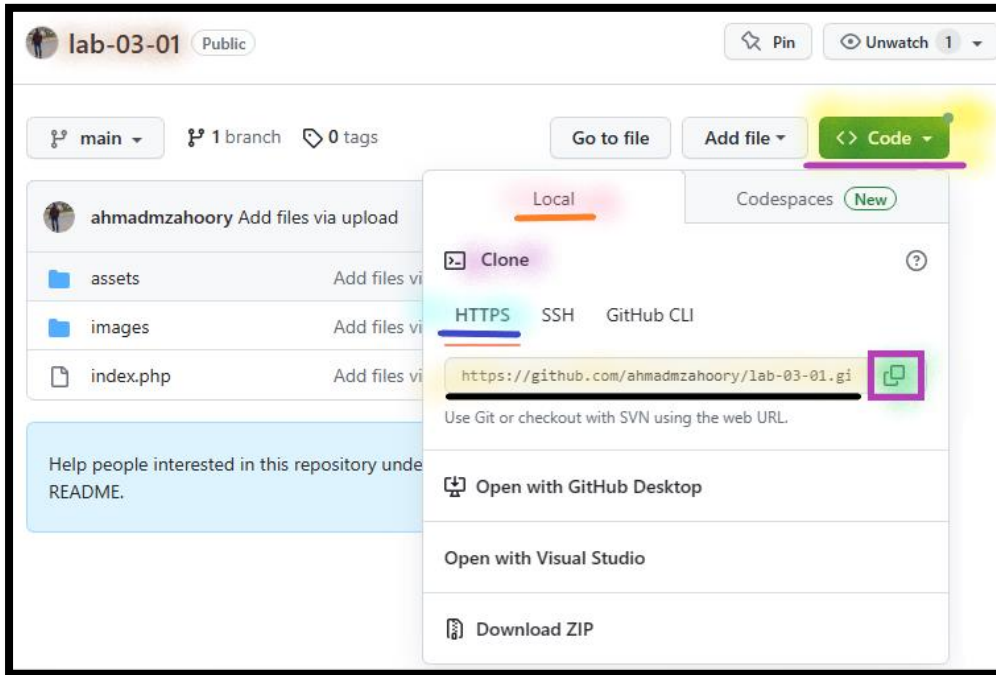
## Step 1: Clone the GitHub Repository

14. Open the **lab-03-01** GitHub repository.

a. Select the **Code**.

i. Select the **HTTPS**.

a) Copy the **Clone URL** in **Notepad**.



## Step 2: Deploy the Php Code

15. Return to the **Linux Web Server**.

16. From the **Linux terminal**:

a. **Execute** the **below command** to **install** the **apache**:

```
sudo yum install -y httpd
```

b. **Execute** the **below command** to **install** the **php**:

```
sudo yum install -y php
```

- c. **Execute** the **below command** to **install** the **git**:

```
sudo yum install -y git
```

- d. **Execute** the **below command** to **change** to the **/var/www/html**:

```
cd /var/www/html/
```

- e. **Execute** the **below command** to **clone** the **lab-03-01 GitHub repository**:

```
sudo git clone CLONE-WEB-URL
```

**Note:** Replace the **CLONE-WEB-URL** with the **Lab-03-01 Github Repository URL** you have copied in the previous step.

- f. **Execute** the **below command** to **list** the **file & folders**:

```
ls -l
```

**Note:** You can see the **lab-03-01 folder**.

- g. **Execute** the **below command** to **change** the **lab-03-01 folder**:

```
cd lab-03-01
```

- h. **Execute** the **below command** to **move** the **contents** to **/var/www/html folder**:

```
sudo mv -v /var/www/html/lab-03-01/* /var/www/html/
```



- i. **Execute** the **below command** to **change** the **parent directory**:

```
cd ..
```

- j. **Execute** the **below command** to **list** the **file & folders**:

```
ls -l
```

**Note:** You can see the **Php web app code**.

- k. **Execute** the **below command** to **start** the **apache service**:

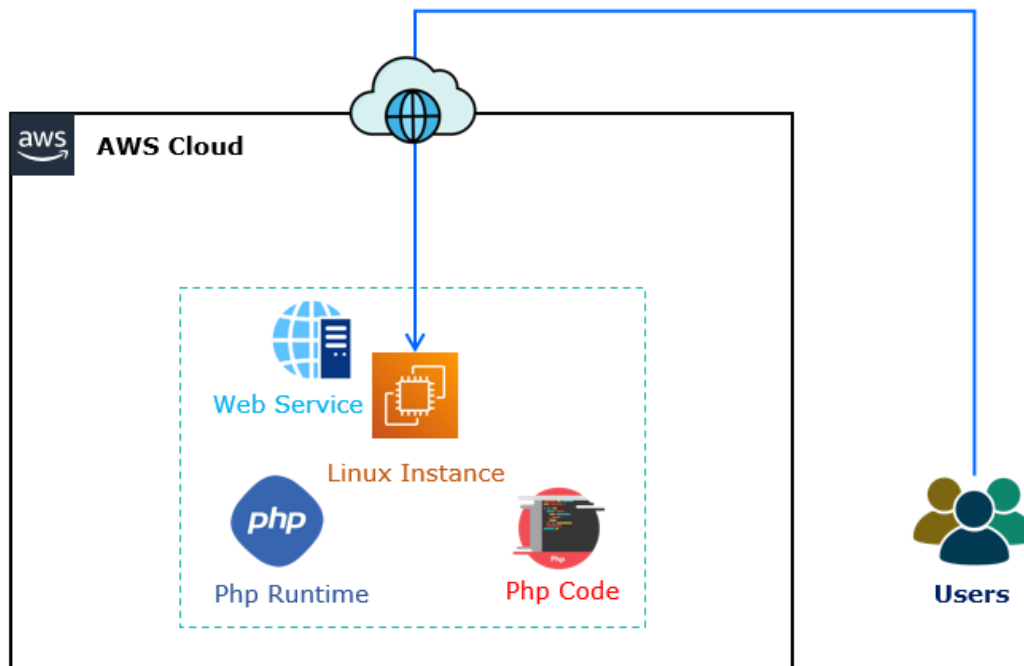
```
sudo service httpd start
```

- l. **Execute** the **below command** to **exit** the **linux terminal**:

```
exit
```

## Task 5: Access the Web Server

In this task, you will access the web server.

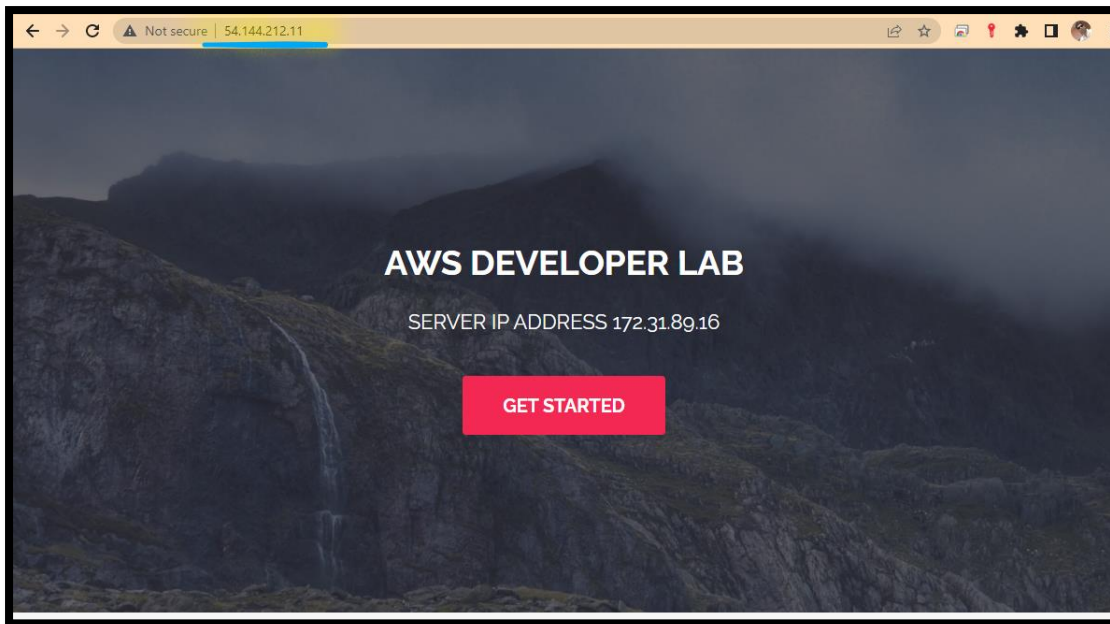


### Step 1: Access the Php App Server

17. From your **Local Desktop/ Laptop**, open the **Browser**, write **Public IP Address** of the **Linux web server**, to access the **website**.

**Note:** You can see the Php Application **web page**.

**Note:** Php Application **web page display** the Php Application Server (Linux virtual machine) **Private IP address**.



## Task 6: Clean up the Environment

### Step 1: Terminate the EC2 Instances

18. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
19. Choose the **YOUR ALLOCATED REGION**, region list to the right of your account information on the navigation bar.
20. Select **Instances**.
21. Select **Linux Web Server**.
  - a. Select the **Instance state**.
    - i. Select **Terminate instance**.
      - a) Select **Terminate**.