

# Deploying PHP application on AWS EC2 instance.

---

## Introduction

This project shows how to deploy a PHP web application on an Amazon EC2 server using the LAMP stack (Linux, Apache, MariaDB, PHP). It explains the full process, including launching an EC2 instance, connecting to it with SSH, installing the required software, setting up the web server, and hosting a signup form application.

The setup includes:

Amazon EC2 instance (Amazon Linux)

Apache (httpd) as the web server

MariaDB as the database

PHP for backend processing

## Requirements

Before starting, ensure you have:

An AWS account with access to launch EC2 instances

A key pair (.pem file) to connect to your EC2 instance

Basic knowledge of Linux commands and SSH

Security group configured to allow:

Port 22 (SSH) → for connecting to the server

Port 80 (HTTP) → for accessing the web application

Port 3306 (MySQL/MariaDB) → if connecting to the database remotely

Installed tools on your local machine:

SSH client (e.g., Git Bash)

## Steps for Deployment

---

Step 1: Launch EC2 instance and Establishing a secure connection to your EC2 instance

1. Launch instance

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, and the main content area displays the following:

**Instances (1/1) Info**

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
php-application	i-03f1b661aa4c36654	Running	t3.micro	Initializing	View alarms +	us-east-1d	ec2-34-207-

## 2. Copy the SSH command

**EC2 Instance Connect** | **Session Manager** | **SSH client** | **EC2 serial console**

**Instance ID**  
 i-03f1b661aa4c36654 (php-application)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is key.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 "key.pem"  
4. Connect to your instance using its Public DNS:  
 ec2-34-207-245-173.compute-1.amazonaws.com

✓ Command copied

ssh -i "key.pem" ec2-user@ec2-34-207-245-173.compute-1.amazonaws.com

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI userna

### 3 Paste command in Git bash

```
HP@LAPTOP-HFSFVC80 MINGW64 /d/dhanashri_workspace/ssh key
$ ssh -i "key.pem" ec2-user@ec2-34-207-245-173.compute-1.amazonaws.com
The authenticity of host 'ec2-34-207-245-173.compute-1.amazonaws.com' (34.207.245.173) using ED25519 key fingerprint is SHA256:92DGVTfhpEcc9F9fEcfaRAAsr45YpGMcbMNUBfun54Fw. This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-207-245-173.compute-1.amazonaws.com' (ED25519)
,      #
~\_\_ #####_          Amazon Linux 2023
~~ \_\#\#\#\_\_
~~   \#\#\#|_
~~     \|/_ __  https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '-'>
~~~\_
~~ ._. /_ / \
~~ /m /'_ /
```

## Step 2: Automating LAMP Stack Setup on AWS EC2

## 1. Create a LAMP.sh file

```
[ec2-user@ip-172-31-16-220:~]$ sudo bash LAMP.sh
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:00:01 ago on Sat Sep  6 19:40:09 2025.
Dependencies resolved.
```

## 2. Insert the code for installing apache, mysql and php

```
sudo yum update
sudo yum install httpd mariadb-server php -y
sudo systemctl start httpd mariadb php-fpm
sudo systemctl enable httpd mariadb php-fpm
sudo echo"<h1> Welcome to my php Application </h1>">
/var/www/html/index.html
```

```
ec2-user@ip-172-31-16-220:~
sudo yum update
sudo yum install httpd mariadb-server php -y
sudo systemctl start httpd mariadb php-fpm
sudo systemctl enable httpd mariadb php-fpm
sudo echo"<h1> Welcome to my php Application </h1>"> /var/www/html/index.html
~
```

## 3. Check the status of apache, mysql and php

```
ec2-user@ip-172-31-16-220:~
[ec2-user@ip-172-31-16-220 ~]$ sudo systemctl status httpd mariadb php-fpm
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Sat 2025-09-06 19:40:27 UTC; 50s ago
```

## Step 3: Change the Directory to Default Directory

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

```
ec2-user@ip-172-31-16-220:~/var/www/html
[ec2-user@ip-172-31-16-220 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-16-220 html]$ ls
index.html
```

## Step 4: Building the Signup Page

### 1. Create a Signup.html file.

```
sudo vim signup.html
```

```
ec2-user@ip-172-31-16-220:~/var/www/html
[ec2-user@ip-172-31-16-220 html]$ sudo vim signup.html|
```

### 2. Code of Signup.html

```
ec2-user@ip-172-31-16-220:/var/www/html
<!DOCTYPE html>
<html>
<head>
  <title>Signup Form</title>
</head>
<body>
  <h2>Signup Form</h2>
  <form action="submit.php" method="post">

    <label for="name">Name:</label><br>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">Email:</label><br>
    <input type="email" id="email" name="email" required><br><br>

    <label for="website">Website:</label><br>
    <input type="url" id="website" name="website"><br><br>

    <label for="comment">Comment:</label><br>
    <textarea id="comment" name="comment" rows="4" cols="50"></textarea><br><br>

    <label>Gender:</label><br>
    <input type="radio" id="female" name="gender" value="female" required>
    <label for="female">Female</label><br>

    <input type="radio" id="male" name="gender" value="male">
    <label for="male">Male</label><br>

    <input type="radio" id="other" name="gender" value="other">
    <label for="other">Other</label><br><br>

    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

## Step 5: Configure the Database (MariaDB)

1. Generate the username and password.

```
sudo mysql
alter user root@localhost identified by 'root';
```

2. Login to Mysql (mariadb105-server)

```
sudo mysql -u root -p
```

```
[ec2-user@ip-172-31-16-220:~]$ sudo mysql  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 3  
Server version: 10.5.29-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2018, oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> alter user root@localhost identified by 'root';  
Query OK, 0 rows affected (0.001 sec)  
  
MariaDB [(none)]> exit;  
Bye  
[ec2-user@ip-172-31-16-220:~]$ sudo mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 4  
Server version: 10.5.29-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2018, oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> |
```

### 3. Create Database

```
#create database  
create database FCT;  
#to see all databases  
show databases;  
#to use that database  
use FCT;
```

```
MariaDB [(none)]> create database FCT;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| FCT      |
| information_schema |
| mysql    |
| performance_schema |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]> use FCT;
Database changed
MariaDB [FCT]> |
```

4. Create table accoring to the signup form.

```
#create table
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    website VARCHAR(255),
    gender ENUM('male', 'female', 'other') NOT NULL,
    comment TEXT
);
```

```
MariaDB [FCT]> CREATE TABLE users (
    ->     id INT PRIMARY KEY AUTO_INCREMENT,
    ->     name VARCHAR(50) NOT NULL,
    ->     email VARCHAR(100) NOT NULL UNIQUE,
    ->     website VARCHAR(255),
    ->     gender ENUM('male', 'female', 'other') NOT NULL,
    ->     comment TEXT
    -> );
Query OK, 0 rows affected (0.012 sec)
```

5. Describe the table

```
desc users;
```

```
MariaDB [FCT]> desc users;
+-----+-----+-----+-----+-----+
| Field | Type            | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+
| id    | int(11)         | NO   | PRI  | NULL    | auto_increment |
| name  | varchar(50)     | NO   |      | NULL    |               |
| email | varchar(100)    | NO   | UNI  | NULL    |               |
| website | varchar(255) | YES  |      | NULL    |               |
| gender | enum('male','female','other') | NO   |      | NULL    |               |
| comment | text           | YES  |      | NULL    |               |
+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)
```

## Step 6: Connect Form to Database with submit.php

1. Create the file submit.php

```
sudo vim submit.php
```

```
ec2-user@ip-172-31-16-220:~$ sudo vim submit.php
```

2. Code of submit.php

```
ec2-user@ip-172-31-16-220:/var/www/html
<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);

// Get form data safely
$name    = $_POST['name'] ?? '';
$email   = $_POST['email'] ?? '';
$website = $_POST['website'] ?? '';
$comment = $_POST['comment'] ?? '';
$gender  = $_POST['gender'] ?? '';

// Database connection
$servername = "localhost";
$username   = "root";
$password   = "root";
$dbname     = "FCT";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
...
```

## Step 7: Install PHP-MySQL Connector

```
sudo yum install php8.4-mysqlnd.x86_64
```

```
ec2-user@ip-172-31-16-220:/var/www/html
[ec2-user@ip-172-31-16-220 html]$ sudo yum install php8.4-mysqlnd.x86_64
Last metadata expiration check: 0:26:05 ago on Sat Sep  6 19:40:09 2025.
Dependencies resolved.
=====
 Package           Architecture      Version
=====
Installing:
 php8.4-mysqlnd          x86_64        8.4.10-1.amzn2023.0

Transaction Summary
=====
Install 1 Package

Total download size: 156 k
```

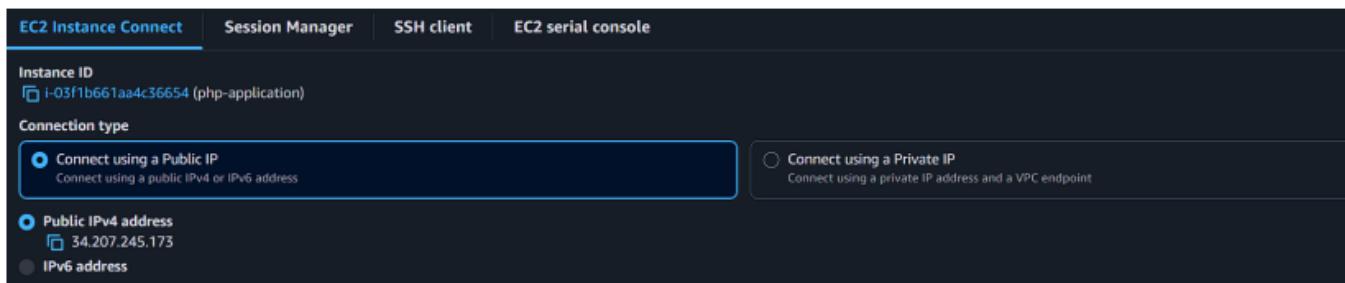
## Step 8: Restart the services

```
sudo systemctl restart httpd mariadb php-fpm
```

```
[ec2-user@ip-172-31-16-220 html]$ sudo systemctl restart httpd mariadb php-fpm
[ec2-user@ip-172-31-16-220 html]$ |
```

## Step 9: Deploy and Test the Signup Form

1. Copy the public IP and Paste the public IP in any browser



2. Signup.html

The screenshot shows a web browser window titled "Signup Form". The URL in the address bar is 174.129.44.147/signup.html. The page content is a "Signup Form" with the following fields:

- Name: Shubham Pawar
- Email: sp7250007@gmail.com
- Website: http://174.129.44.147/signup
- Comment: Hii
- Gender: Male (radio button selected)

3. Submit.php

The screenshot shows a web browser window titled "Form Submission Result". The URL in the address bar is 174.129.44.147/submit.php. The page displays a success message: "New record created successfully!" followed by a summary of the submitted information:

**Submitted Information:**

- Name: Shubham Pawar
- Email: sp7250007@gmail.com
- Website: http://174.129.44.147/signup.html
- Comment: Hii
- Gender: male

## Summary

---

In this project, we deployed a PHP-based signup form on an AWS EC2 instance using the LAMP stack (Linux, Apache, MariaDB, PHP). We launched and set up an EC2 instance with Amazon Linux, then used a custom lamp.sh script to install the LAMP stack automatically. After that, we built a signup.html form and connected it to a MariaDB database with submit.php, which saved user input into the database. To enable communication between PHP and MariaDB, we added the PHP-MySQL connector. Finally, we tested the form in a browser and confirmed that the data was stored successfully.