

Configure the LAMP environment and deploy the sample PHP code

Step 1: Create an AWS Account

1. Visit [AWS Sign Up](#) and create an account.
2. Provide your email, password, and payment details (required for verification).
3. Log in to the AWS Management Console.

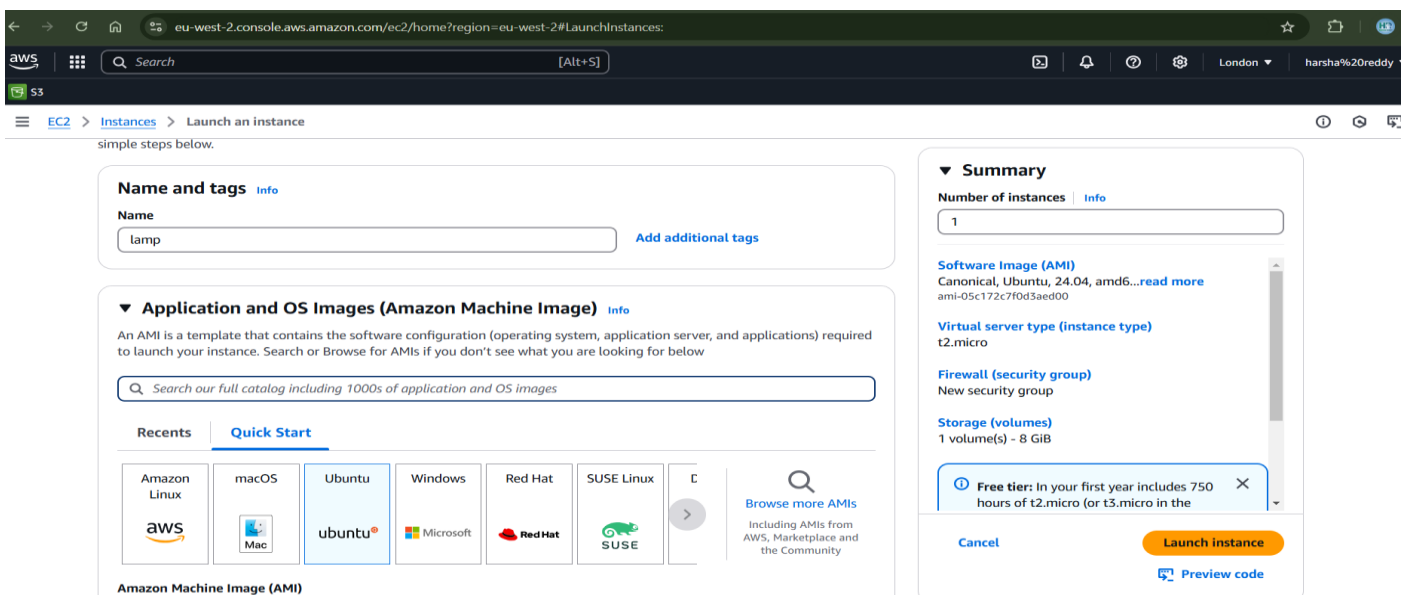
Step 2: Launch an EC2 Instance

1. Navigate to the **EC2 Dashboard**.
2. Click **Launch Instance**.
3. Configure the instance:
 - **Name:** Give it a name (e.g., LAMP-Demo).
 - **AMI:** Choose **Amazon Linux 2 AMI** (or Ubuntu, if preferred).
 - **Instance Type:** Select **t2. Micro** (eligible for Free Tier).
 - **Key Pair:** Create a new key pair or use an existing one.
 - **Network Settings:**
 - Allow HTTP (port 80) and HTTPS (port 443) in addition to SSH (port 22).
 - **Storage:** Use the default storage size (8GB) or increase if needed.
4. Click **Launch Instance** and wait for it to initialise.

Step 3: Connect to the Instance

1. From the EC2 dashboard, select the instance and click **Connect**.
2. Use the SSH command provided or a tool like git bash etc ...

```
ssh -i "your-key.pem" ec2-user@your-public-ip
```



eu-west-2.console.aws.amazon.com/ec2/home?region=eu-west-2#LaunchInstances:

Search [Alt+S]

EC2 > Instances > Launch an instance

Network settings

Network | Info

vpc-0009f0bcd77050a3c

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

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

Common security groups | Info

Select security groups

security-1 sg-08e82fd9df0812818 X
VPC: vpc-0009f0bcd77050a3c

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Configure storage

Advanced

Summary

Number of instances | Info

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-05c172c7f0d3aed00

Virtual server type (instance type)

t2.micro

Firewall (security group)

security-1

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the

Cancel [Launch instance](#) [Preview code](#)


Update System Packages

Once connected, update your system packages: `sudo apt update`

Step 4: Install LAMP Stack

- **Install Apache Web Server**
 - `sudo apt install apache2`
 - After installation, verify that Apache is running by accessing your EC2 instance's public IP address in a web browser. You should see the default Apache welcome page.

Not secure 18.171.192.119



Ubuntu

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.Load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

```

root@ip-172-31-20-161:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-12-18 07:06:37 UTC; 47s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 2082 (apache2)
    Tasks: 55 (limit: 1130)
   Memory: 5.4M (peak: 5.7M)
      CPU: 34ms
   CGroup: /system.slice/apache2.service
           └─2082 /usr/sbin/apache2 -k start
             └─2084 /usr/sbin/apache2 -k start
               └─2085 /usr/sbin/apache2 -k start

Dec 18 07:06:37 ip-172-31-20-161 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 18 07:06:37 ip-172-31-20-161 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-20-161:~# |

```

Step5: Install MySQL

5.1 Installation command

To install MySQL: Sudo apt install MySQL-server

5.2 Create Password for MySQL root user:

Log in to MySQL and run the following command:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '<desird password>';
```

5.3 Creating a Test Database and User

Log in to MySQL and Create a test database and user:

```

sudo mysql -u root -p
CREATE DATABASE testdb;
CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON testdb.* TO 'testuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;

```

```

Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
root@ip-172-31-26-9:~# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2018-10-09 13:01:55 UTC; 23s ago
 Main PID: 20532 (mysqld)
    Tasks: 27 (limit: 1152)
   CGroup: /system.slice/mysql.service
           └─20532 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

```

```

root@ip-172-31-32-186:~# sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, 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>

```

```

root@ip-172-31-26-9:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.23-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

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> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'mysql12345';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
root@ip-172-31-26-9:~#

```

Step 6: Install PHP

6.1 installation command: install PHP and necessary modules:

```
sudo apt install php libapache2-mod-php php-mysql -y
```

6.2 Verifying PHP installation

Check the PHP version to ensure it was installed successfully: `php -v`

```

root@ip-172-31-32-186:~# php -v
PHP 8.3.6 (cli) (built: Dec  2 2024 12:36:18) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
root@ip-172-31-32-186:~#

```

Step 7: Configure Apache2 for PHP

7.1 Creating a test PHP File

- Create a test PHP file to Verify PHP integration with Apache2:
 - The default path of Apache is: “`/var/www/html/`”
 - Change `index.html` to `index.php`
- Add the following content to the file:

```

<?php
phpinfo();
?>


```

7.2 Restarting the Apache

- Restart the Apache Server to apply the Changes: `sudo systemctl restart apache`
- Now, Copy the Public IP and paste it into the browser

echo "

PHP Version 8.3.6



System	Linux ip-172-31-20-161 6.8.0-1018-aws #20-Ubuntu SMP Thu Oct 10 18:14:42 UTC 2024 x86_64
Build Date	Dec 2 2024 12:36:18
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqld.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/15-xml.ini, /etc/php/8.3/apache2/conf.d/20-bz2.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-curl.ini, /etc/php/8.3/apache2/conf.d/20-dom.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gd.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-mbstring.ini, /etc/php/8.3/apache2/conf.d/20-mcrypt.ini, /etc/php/8.3/apache2/conf.d/20-mysqli.ini, /etc/php/8.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-simplexml.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini, /etc/php/8.3/apache2/conf.d/20-xmlreader.ini, /etc/php/8.3/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.3/apache2/conf.d/20-xsl.ini, /etc/php/8.3/apache2/conf.d/20-zip.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831.NTS
PHP Extension Build	API20230831.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled

Step 8: Deploy the PHP Application on your LAMP Stack

8.1 Creating a simple PHP script:

- Create a PHP File to test the Connection to your MySQL database:
- The path of the PHP file is: nano /var/www/html/index.php

Simple PHP Code

```
GNU nano 7.2 index.php *
!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>PHP with HTML & CSS</title>
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f9;
    color: #333;
    text-align: center;
  }
  header {
    background-color: #007bff;
    color: white;
    padding: 20px 0;
  }
  main {
    margin: 20px auto;
    padding: 20px;
    max-width: 800px;
    background: white;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  }
  footer {
    margin-top: 20px;
    background-color: #333;
    color: white;
    padding: 10px 0;
  }
  .success {
    color: green;
  }
  .error {
    color: red;
  }
</style>
</head>
<body>
<header>
<h1>Welcome to My PHP Application</h1>
</header>
<main>
<h2>Server Information</h2>
<p><strong>Current Server Time:</strong>
<?php
  echo date('Y-m-d H:i:s');
?>
</p>
```

```
GNU nano 7.2
  footer {
    margin-top: 20px;
    background-color: #333;
    color: white;
    padding: 10px 0;
  }
  .success {
    color: green;
  }
  .error {
    color: red;
  }
</style>
</head>
<body>
<header>
<h1>Welcome to My PHP Application</h1>
</header>
<main>
<h2>Server Information</h2>
<p><strong>Current Server Time:</strong>
<?php
  echo date('Y-m-d H:i:s');
?>
</p>

<h2>Database Connection</h2>
<?php
// Database configuration
$host = "localhost";
$username = "testuser"; // Replace with your MySQL username
$password = "12345678"; // Replace with your MySQL password
$dbname = "testdb"; // Replace with your database name

// Try connecting to the database
try {
  $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  echo "<p class='success'>Successfully connected to the database!</p>";
} catch (PDOException $e) {
  echo "<p class='error'>Database connection failed: " . $e->getMessage() . "</p>";
}

// Close the connection
$conn = null;
?>
</main>
<footer>
<p>&copy; <?php echo date('Y'); ?> My PHP Application</p>
</footer>
</body>
</html>
```

FINAL RESULT IS:

Welcome to My PHP Application

Server Information

Current Server Time: 2024-12-18 11:01:52

Database Connection

Successfully connected to the database!

© 2024 My PHP Application