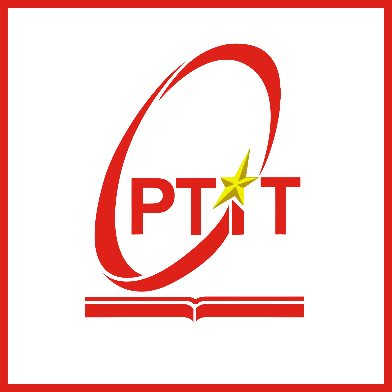
**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

- - - 🙞 🕮 🙜 - - -



**BÁO CÁO CUỐI KỲ**

**CHUYÊN ĐỀ HỆ THỐNG THÔNG TIN**

**“LẬP TÀI LIỆU TRIỂN KHAI PHẦN MỀM VÀ CI/CD”**

**Người hướng dẫn** :Ths. Lê Hà Thanh

**Lớp** : D20CQCNHT01-N

**Sinh viên** :N20DCCN085 – Lương Thúy Vy

**TP. HCM, 6/2024**

# 

**MỤC LỤC**

[I. Cài đặt website sử dụng kĩ thuật Reverse Proxy 1](#_Toc169471965)

[1. Cài đặt CMS: Wordpress(blogs) 1](#_Toc169471966)

[2. Cài đặt E-commerce: WooCommerce 3](#_Toc169471967)

[a. WooCommerce là gì? 3](#_Toc169471968)

[b. Cài đặt WooCommerce. 4](#_Toc169471969)

[II. Thực hành triển khai website trên theo cách thức CI/CD. 6](#_Toc169471970)

[1. CI/CD là gì? 6](#_Toc169471971)

[2. Đưa code lên Github 7](#_Toc169471972)

[3. GitHub Action là gì? 8](#_Toc169471973)

[4. Thiết lập CI/ CD 11](#_Toc169471974)

[5. Kết quả demo 13](#_Toc169471975)

[6. Khởi chạy sản phẩm trên public IP 13](#_Toc169471976)

# Cài đặt website sử dụng kĩ thuật Reverse Proxy

Trước khi cài đặt WordPress, cần tải về hai tệp tin chính, đó là tệp tin WampServer và tệp tin WordPress.

## 1. Cài đặt CMS: Wordpress(blogs)

**a. Cài đặt WAMP**

WAMP là tên viết tắt cho phần mềm Windows, Apache, MySQL, PHP. Nó có nguồn gốc từ LAMP là viết tắt của Linux, Apache, MySQL và PHP. Trong khi LAMP được sử dụng trên máy chủ Linux thì WAMP được sử dụng trên Windows server. Vì WordPress thường không được cài đặt trên Windows server, WAMP đã trở nên phổ biến trong giới developer như một phương pháp cài đặt WordPress trên máy tính cá nhân của họ.

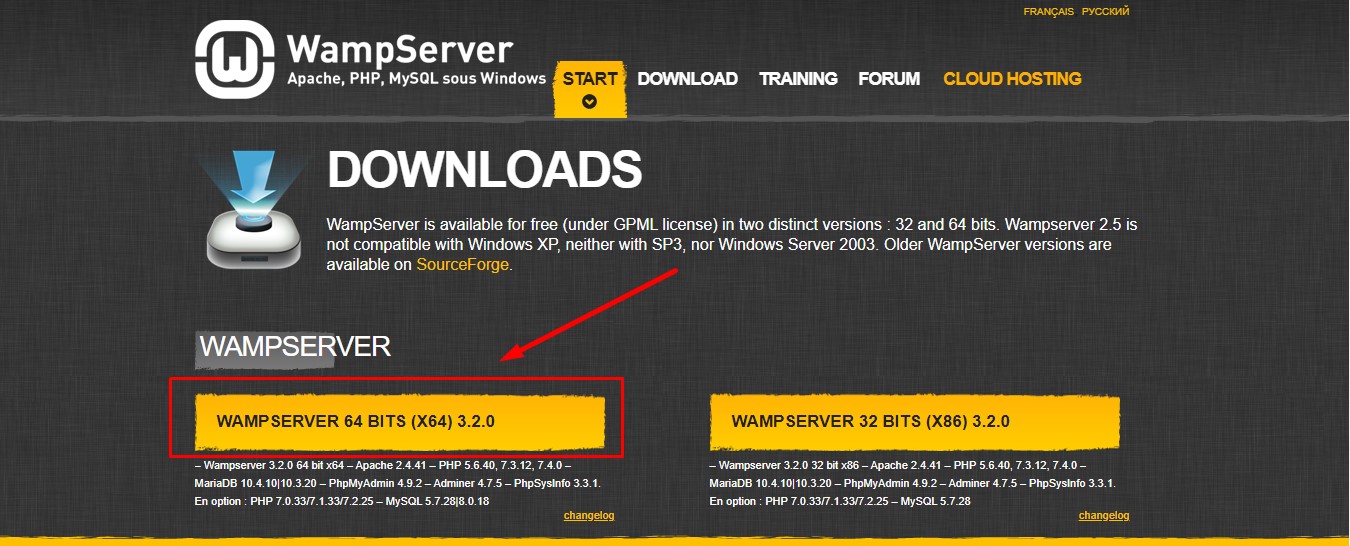


Chữ “A” trong WAMP là viết tắt của Apache. Apache là phần mềm máy chủ được sử dụng để phục vụ các trang web. Bất cứ khi nào ai đó nhập URL trang web WordPress của bạn, Apache là phần mềm “phục vụ” trang web WordPress của bạn.

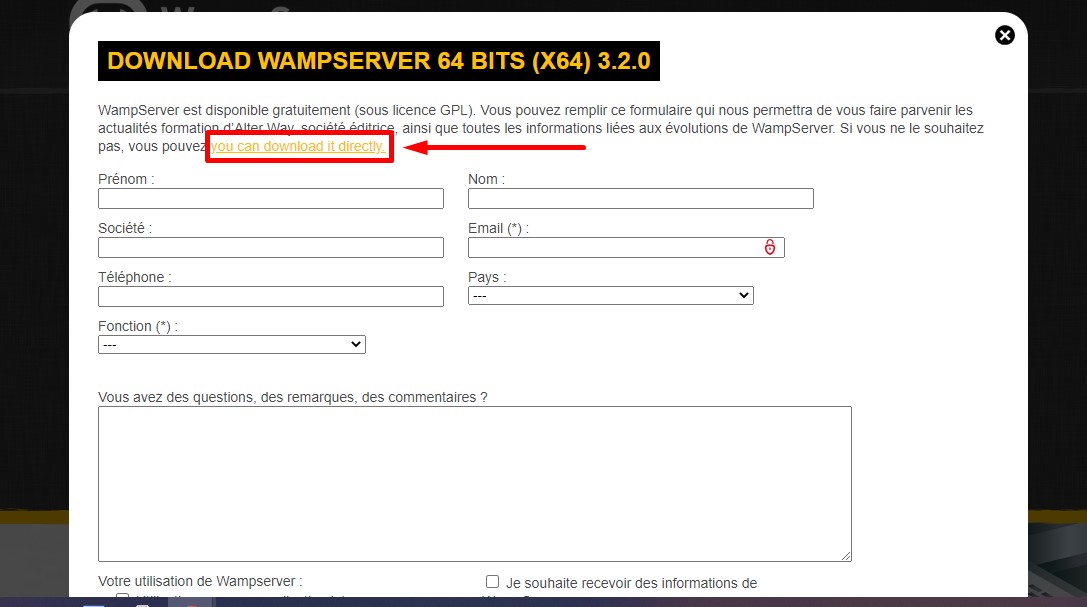
- Chữ “M” trong WAMP là viết tắt của MySQL. MySQL là một hệ quản trị cơ sở dữ liệu. Công việc trong stack là lưu trữ tất cả nội dung trang web của bạn, hồ sơ người dùng, nhận xét, v.v.

- Chữ “P” trong WAMP là viết tắt của PHP. PHP là ngôn ngữ lập trình mà WordPress được viết ra. Nó cũng là mảnh ghép toàn bộ phần mềm lại với nhau. Nó chạy như một quy trình trong Apache và giao tiếp với cơ sở dữ liệu MySQL để xây dựng các trang web của bạn.

* **Bước 1:**Để tải WAMPServer bạn hãy hãy truy cập trang chủ [tại đây](https://www.wampserver.com/en/#download-wrapper)



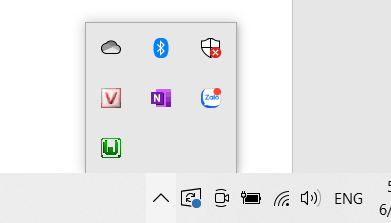
**Bước 2:** Vào file vừa tải về, chạy trình cài đặt (mở file có đuôi .exe) và làm theo hướng dẫn để cài đặt WAMP trên máy tính của bạn.



Khi cài đặt xong, hãy bấm đúp chuột vào biểu tượng của WAMP trên màn hình Desktop để khởi động. Sau đó, nhấn vào mũi tên ở góc phải màn hình và bạn sẽ thấy biểu tượng WAMP nằm ở thanh tác vụ ẩn. Màu của biểu tượng tương ứng với trạng thái của máy chủ:

* **Màu đỏ:** Máy chủ WAMP tạm thời bị vô hiệu hóa hoặc dịch vụ Apache, MySQL gặp trở ngại không hoạt động được.
* **Màu cam:** Dịch vụ đang gặp một vài vấn đề.
* **Màu xanh:** Máy chủ đã hoạt động và sẵn sàng sử dụng.

Nếu biểu tượng WampSever màu xanh, điều đó có nghĩa là việc cài đặt WampServer đã thành công, giống như hình ảnh xuất hiện dưới đây.



**b. Cài đặt Word Press**

**Bước 1:** Tải phiên bản mới nhất của WordPress từ trang web chính thức (<https://wordpress.org/latest.zip>.)

****

**Bước 2:** Sau khi cài đặt Wordpress, thực hiện giải nén và tiến hành sao chép thư mục WordPress đến Ổ đĩa **Local Disk (C:)** > chọn thư mục wamp64 > và paste vào thư mục www.

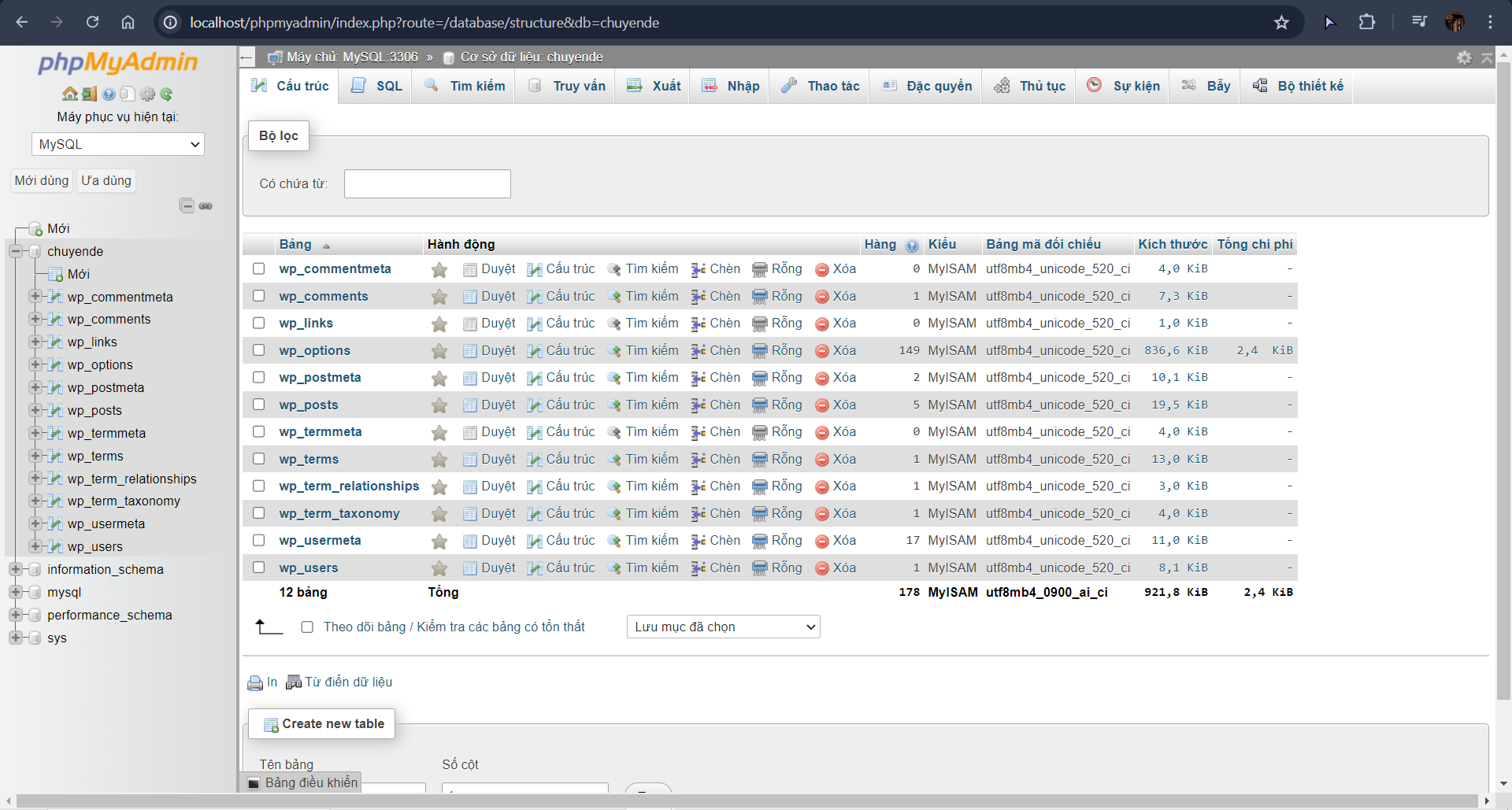
Bước tiếp theo là set up web server cho Wamp và tạo database cho WordPress.

**c. Tạo cơ sở dữ liệu MySQL**

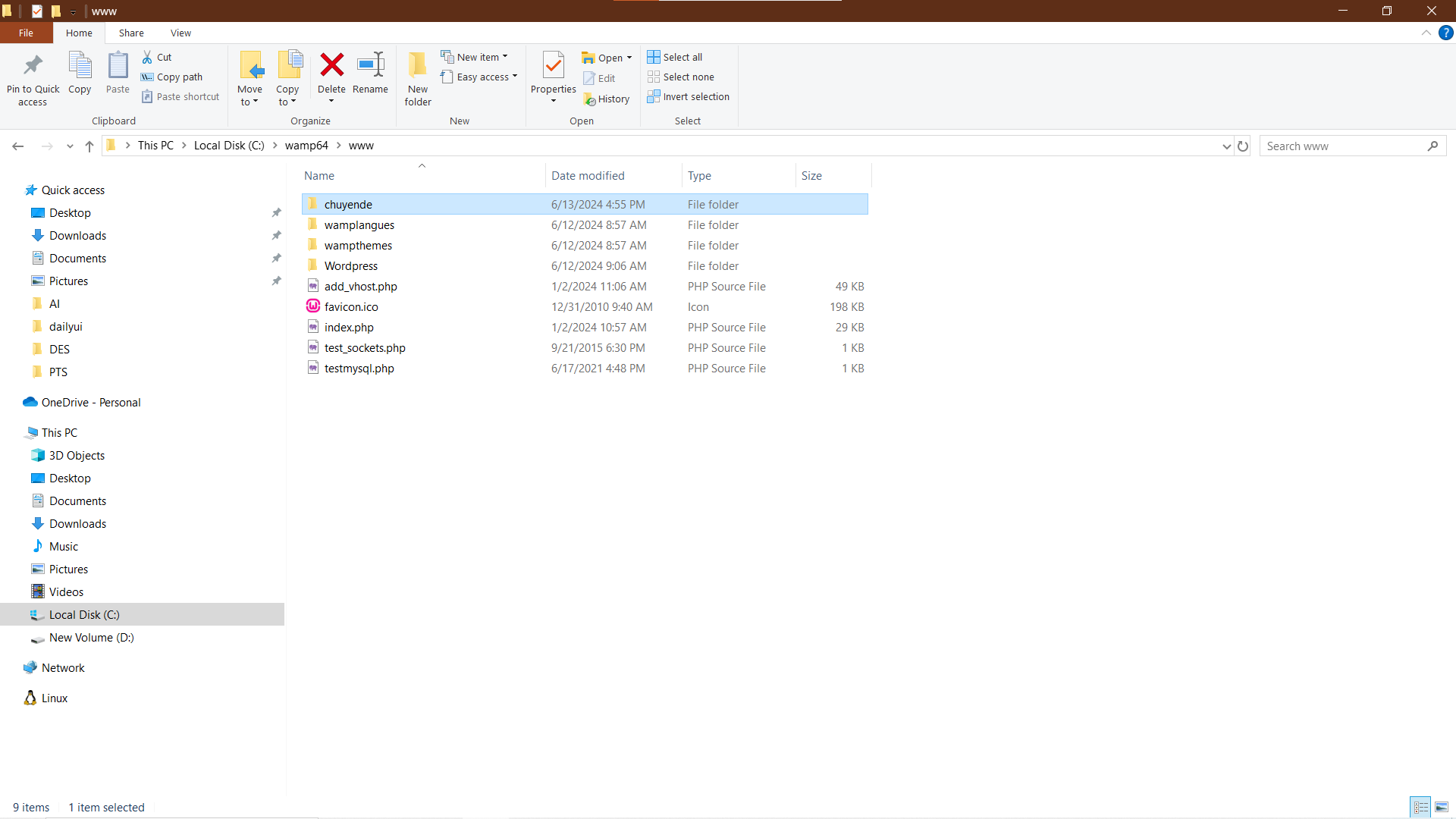
**Bước 1:** Truy cập vào **localhost/phpmyadmin/** để tạo database.

**Bước 2:** Đăng nhập với tên người dùng là root và mật khẩu để trống (mặc định).

**Bước 3:** Tạo một cơ sở dữ liệu mới cho WordPress, ví dụ: chuyende, bằng cách nhấp vào menu **Databases.** Sau đó ở phần Create database, bạn nhập tên database cần tạo vào ô Database name, phần Collation bạn hãy chọn là **utf8\_unicode\_ci**   
Như trong hình đã tạo database của wordpress:

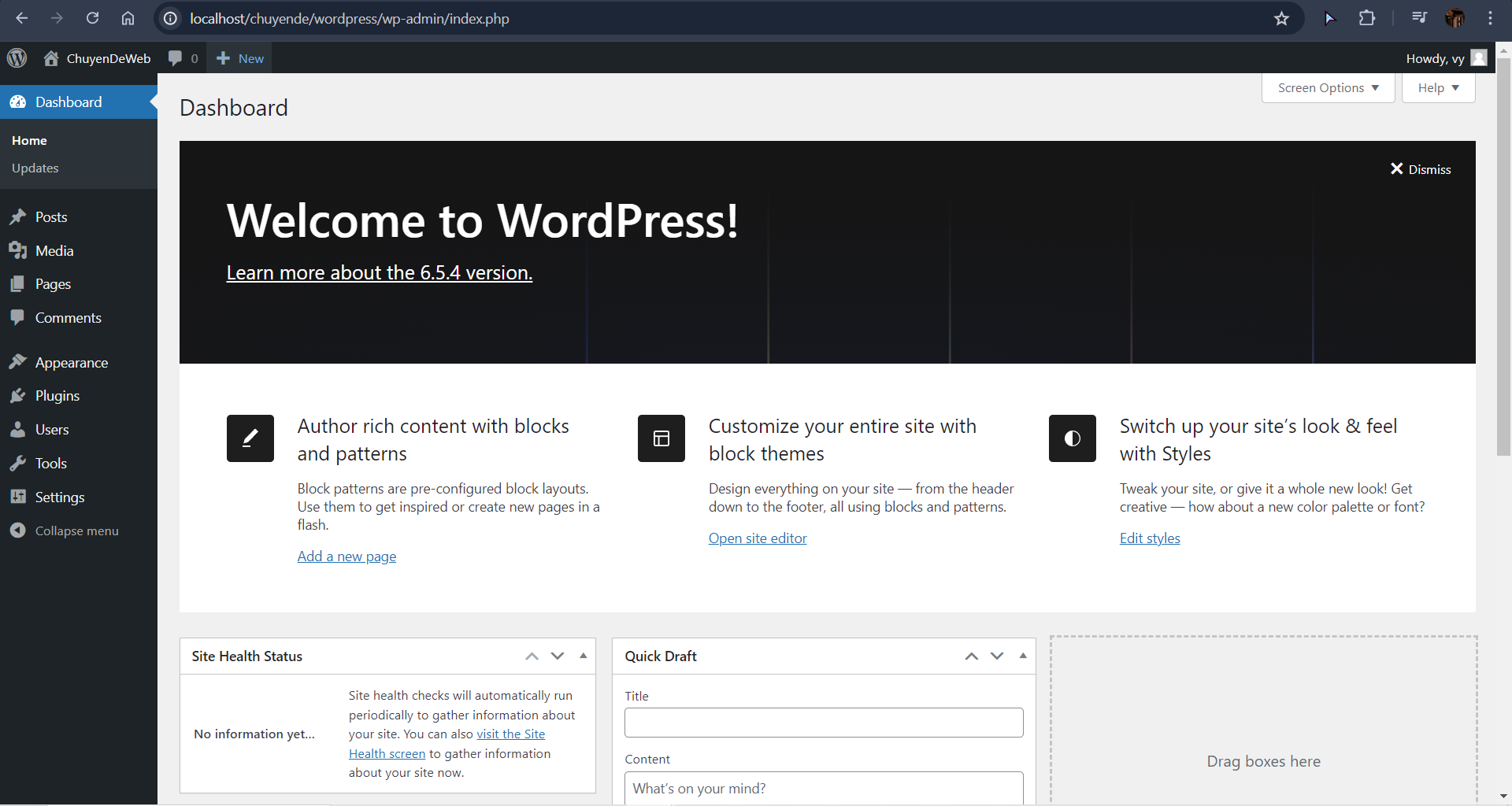


**Bước 4:** Vào file chứa file Wordpress lúc nãy vừa copy, đổi tên thư mục WordPress thành tên database *chuyende* đã tạo trong phpMyAdmin.



**Bước 5:** Khi đã hoàn thành, mở trình duyệt và nhập localhost/chuyende (tên thư mục WordPress). Tiến hành thiếp lập một số thông tin cơ bản và đăng nhập.

Cài đặt thành công Wordpress trên Localhost bằng WampServer sẽ đưa về kết quả như bên dưới:



## Cài đặt E-commerce: WooCommerce

### a. WooCommerce là gì?

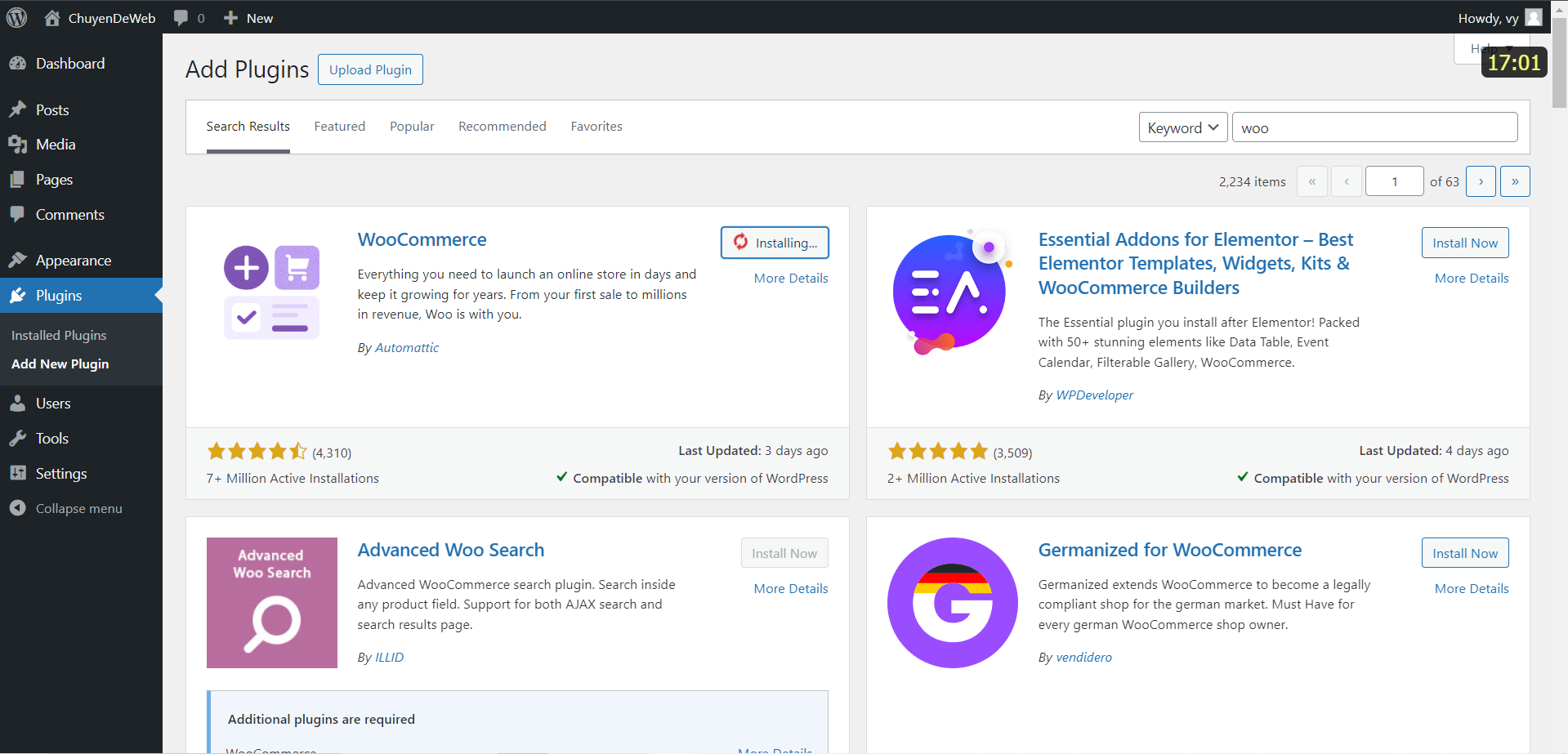
**WooCommerce là một plugin mã nguồn mở trên WordPress hỗ trợ thiết kế website thương mại điện tử hay web bán hàng**. Tác vụ của nó là biến website trở thành một trang thương mại điện tử nhanh chóng, hiệu quả. Vậy chức năng của **WooCommerce là gì**? **WooCommerce** có toàn bộ các chức năng mà một web bán hàng cần có như: giỏ đơn hàng, số lượng hàng hóa, thanh toán, khuyến mãi, đăng ký khách hàng,…

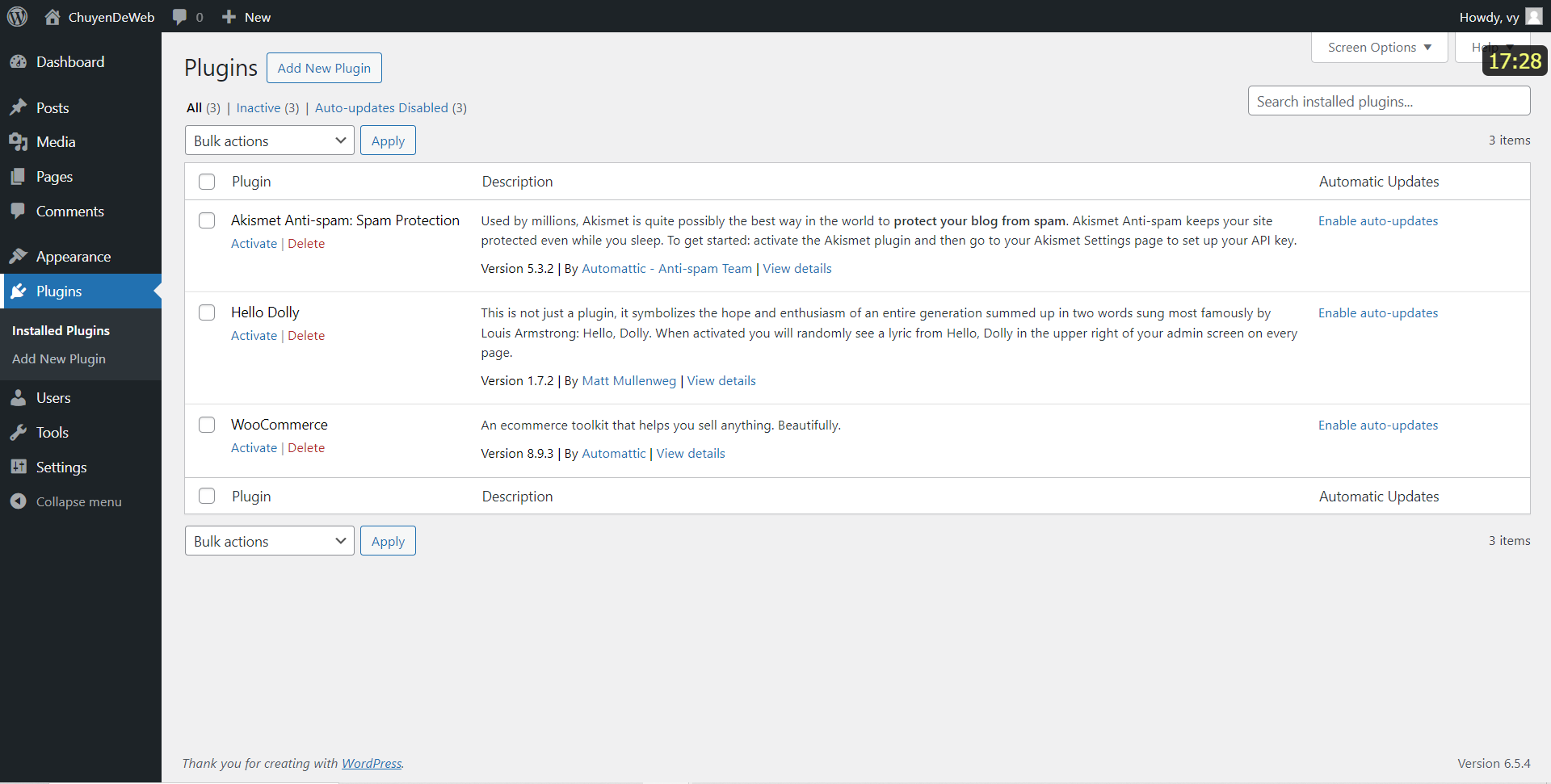
### b. Cài đặt WooCommerce.



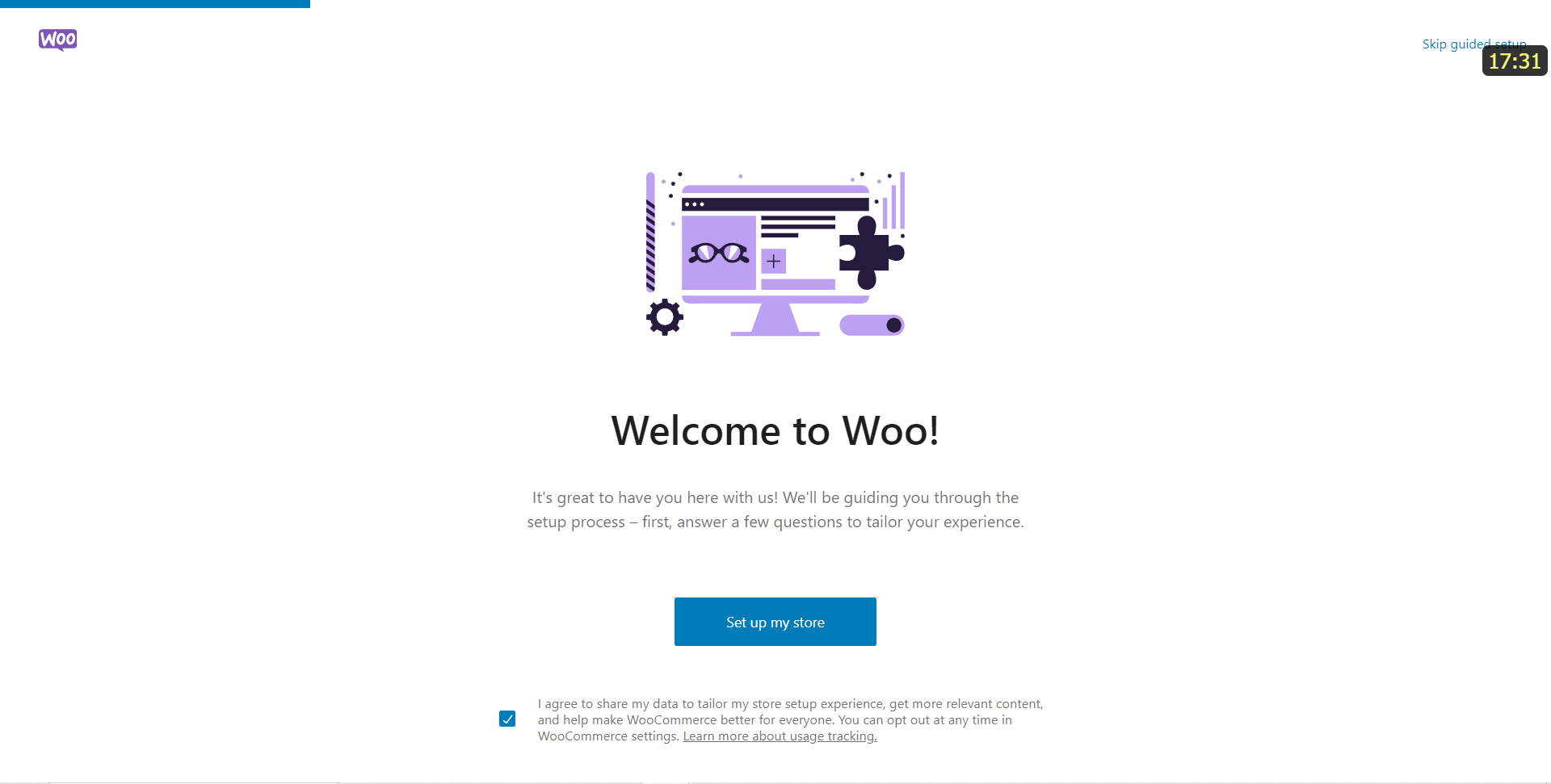
**Bước 1:** Truy cập vào **giao diện WordPress admin** bằng username và password theo đường link có dạng: <http://localhost/chuyende/wordpress/wp-admin>

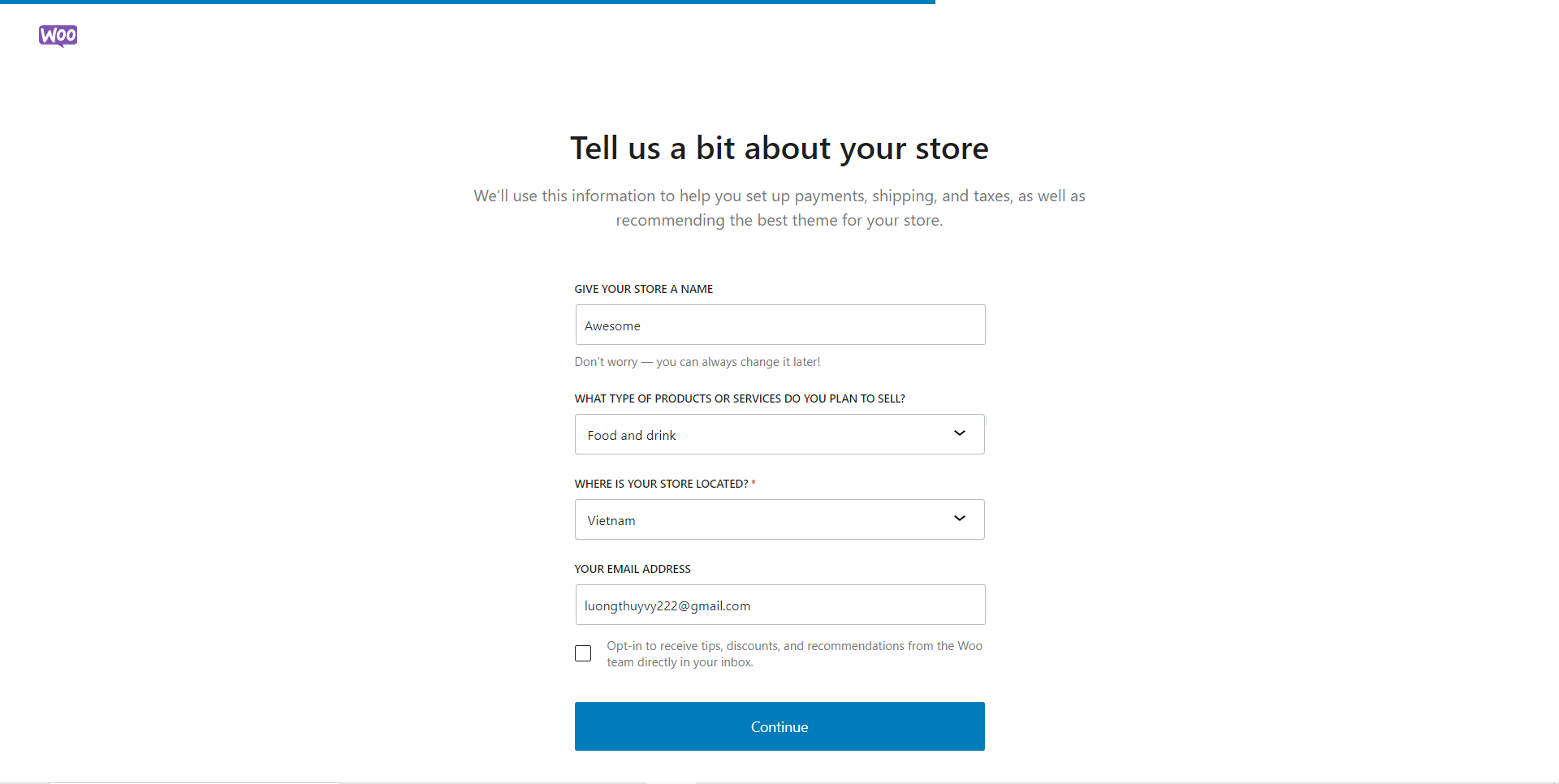
**Bước 2:** Click chọn mục **Plugins** (gói mở rộng) ở thanh menu bên trái màn hình, sau đó nhấn nút Add new, tìm kiếm và tải về **Plugin WooCommerce.**



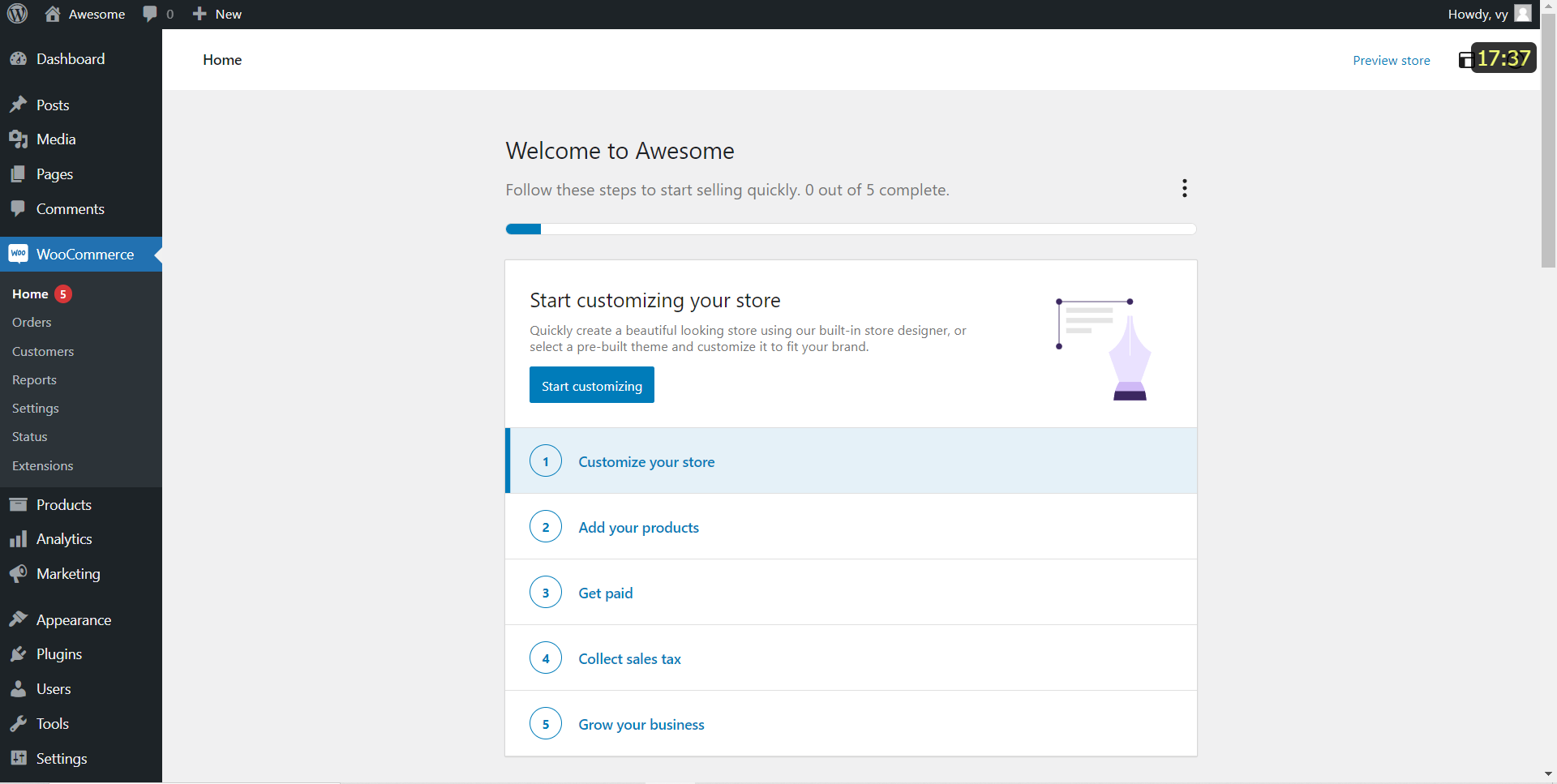


**Bước 3:** Sau khi cài đặt xong, bấm Activate và làm theo các hướng dẫn.





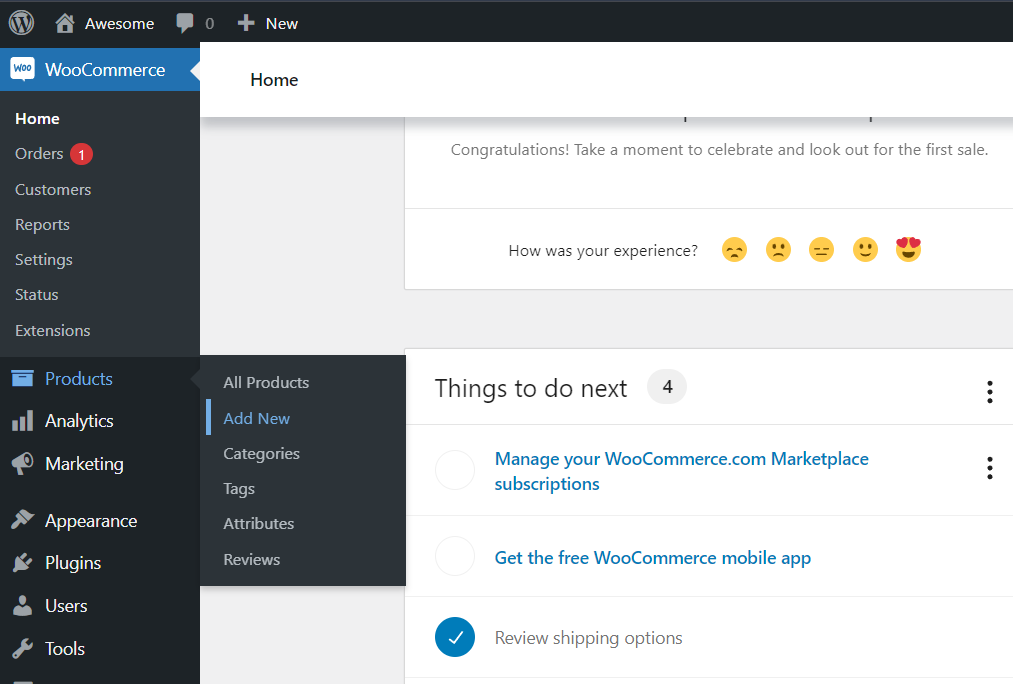
Hệ thống lần lượt hiển thị các mục cơ bản phải hoàn thành trước khi sử dụng **WooCommerce**:



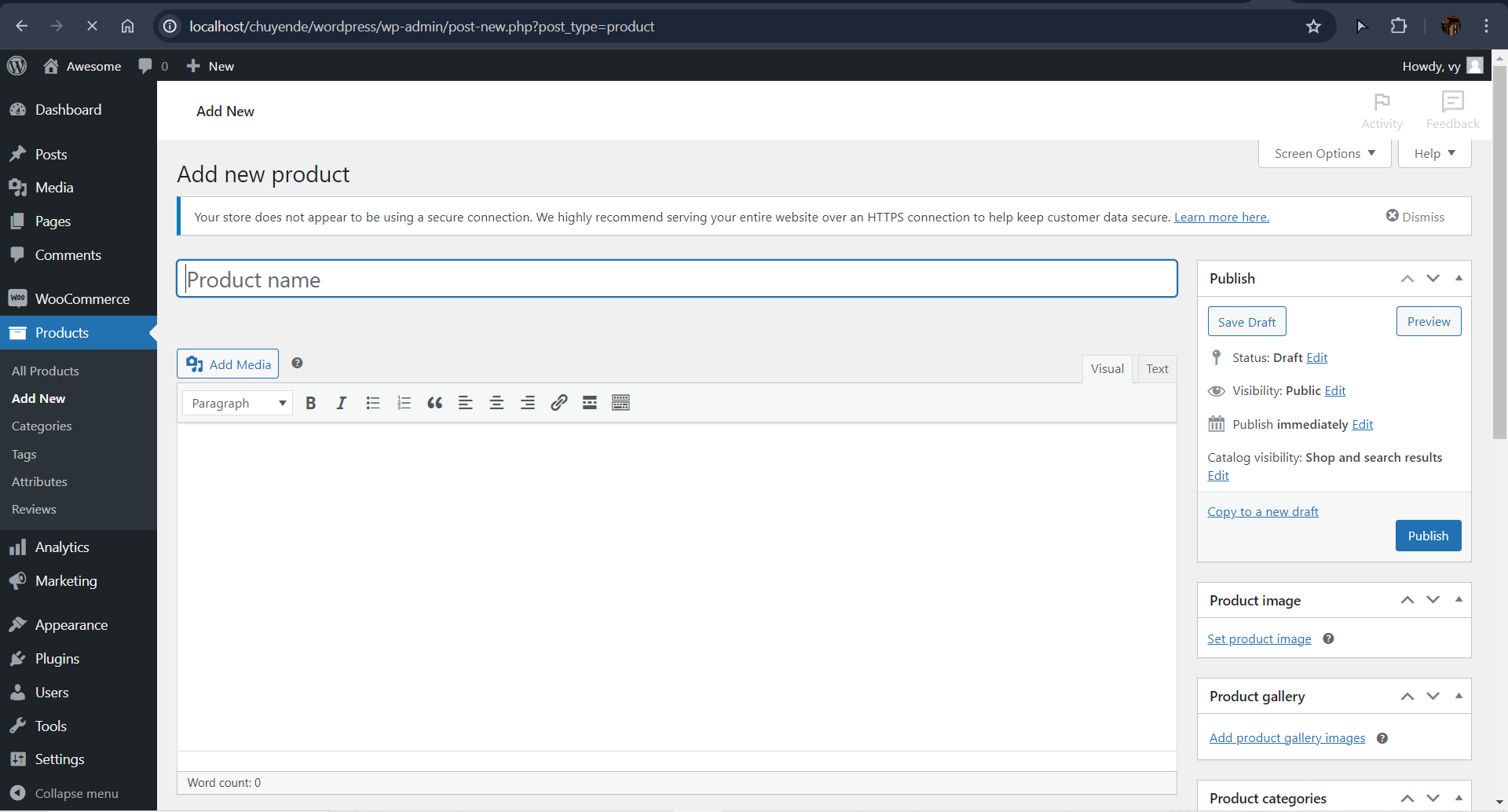
**Bước 4:** Thêm và quản lý sản phẩm mới trong WooCommerce

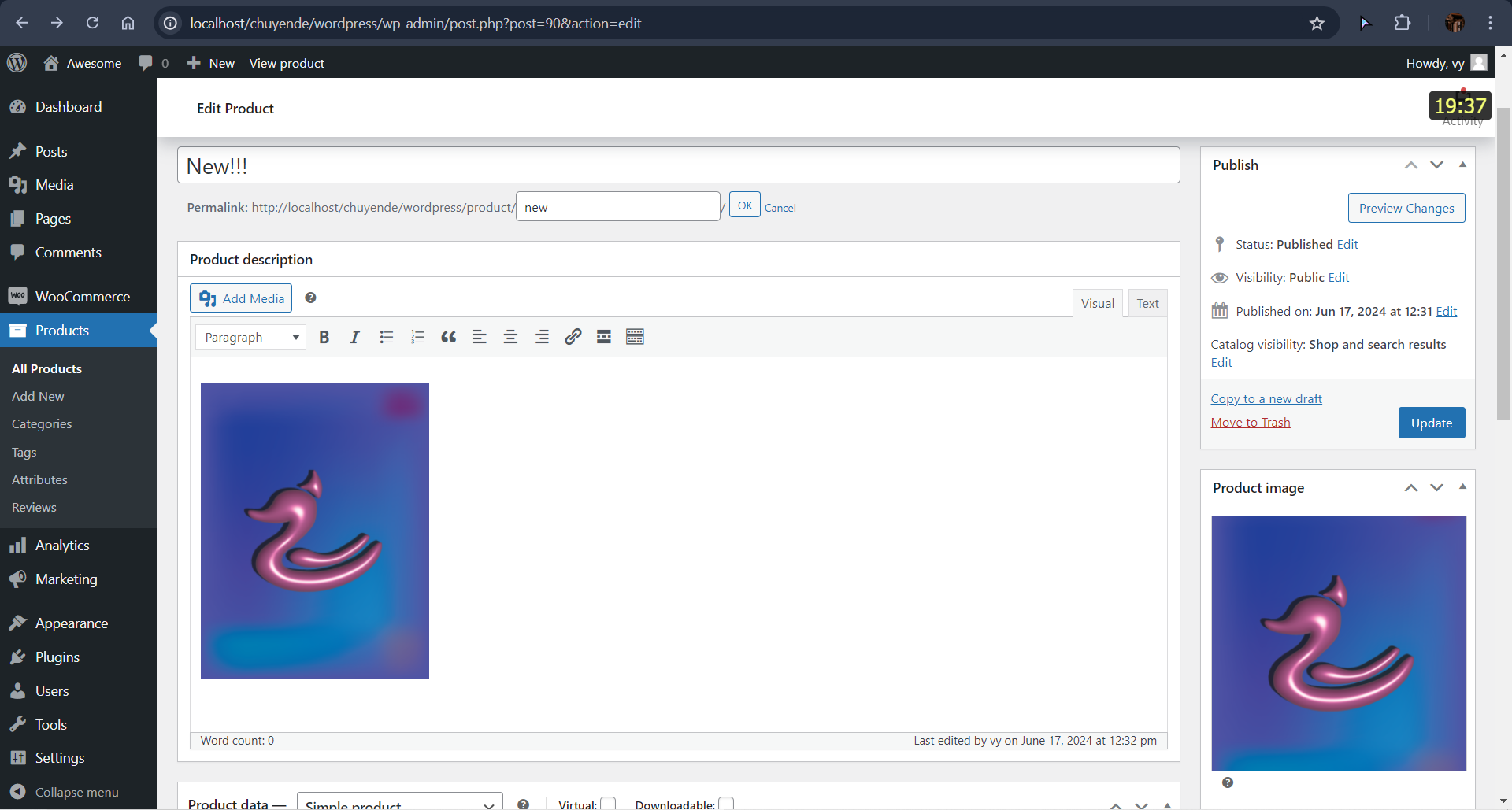
Sau khi cài đặt WooCommerce, Bạn sẽ tìm thấy hai tab mới trên menu của bảng điều khiển. Chúng được gọi là **WooCommerce** (khu vực cài đặt chung của Bạn) và **Sản phẩm** (khu vực cài đặt sản phẩm của Bạn).

Để bắt đầu thiết lập cửa hàng trực tuyến của Bạn, Bạn có thể theo liên kết ở cuối trình hướng dẫn thiết lập hoặc đi đến **Sản phẩm> Thêm mới**. Dù bằng cách nào, điều này sẽ cho phép Bạn thêm sản phẩm đầu tiên của mình, sử dụng trình chỉnh sửa WordPress quen thuộc:

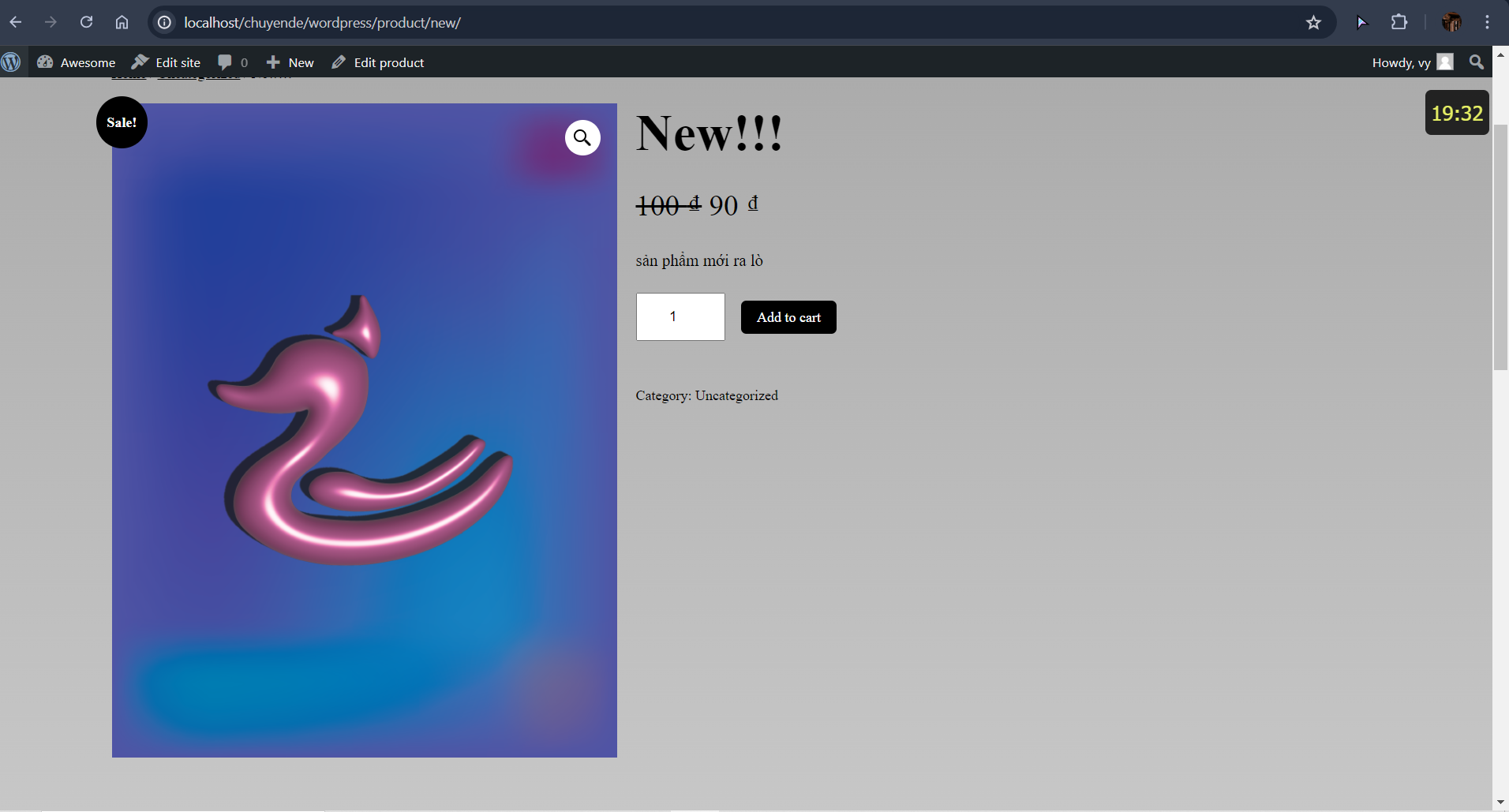


**Bước 5:** Điền các thông tin về sản phẩm

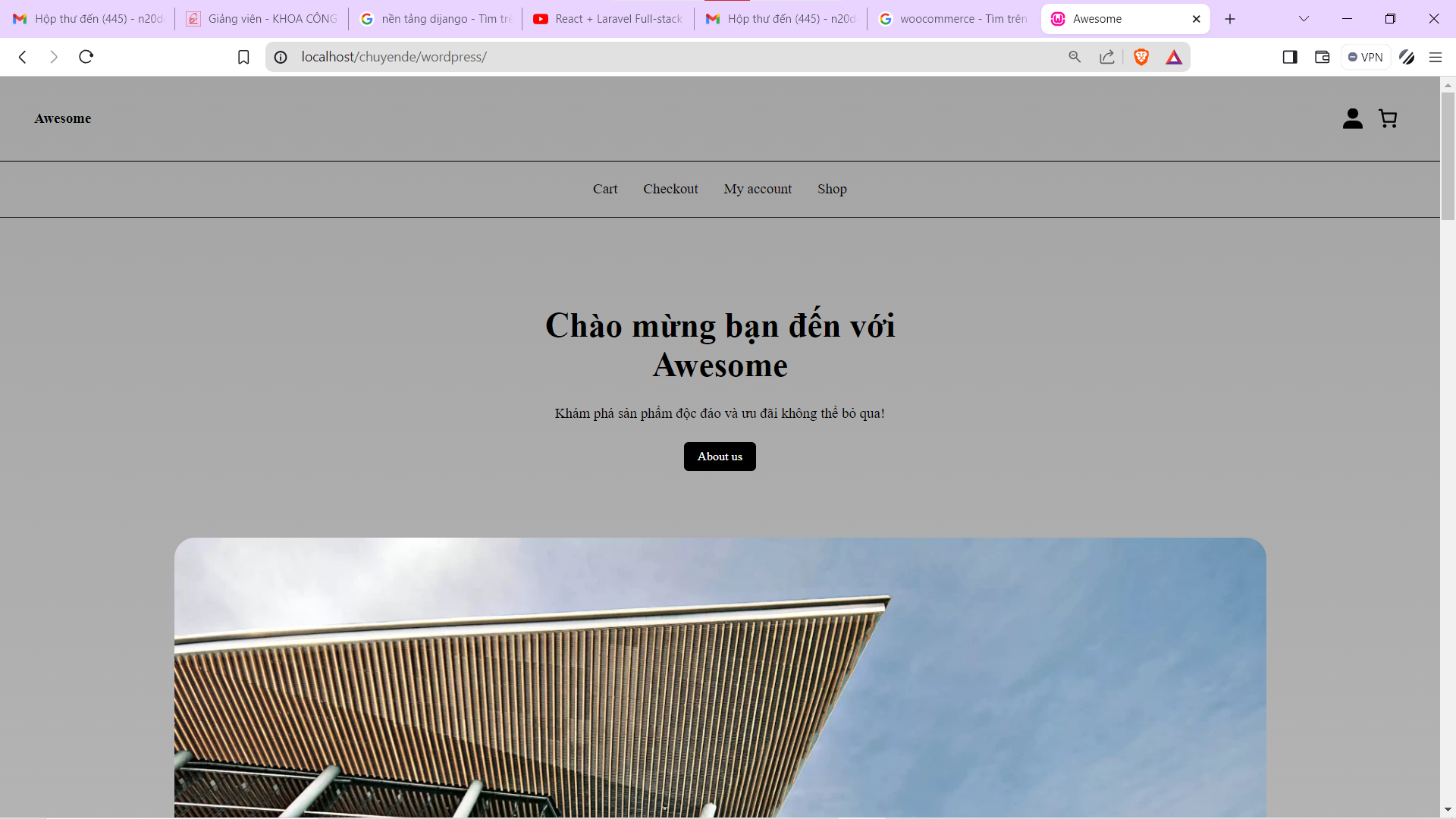




**Bước 6:** Sản phẩm sau khi thêm thành công sẽ hiển thị trên website như sau:



Sau khi cài đặt thành công website sẽ có giao diện như sau:



# Thực hành triển khai website trên theo cách thức CI/CD.

## Reverse Proxy

Reverse proxy là một loại proxy server trung gian giữa một máy chủ và các client gửi tới các yêu cầu. Nó kiểm soát yêu cầu của các client, nếu hợp lệ, sẽ luân chuyển đến các server thích ứng. Trái ngược với một forward proxy, là một trung gian cho phép các client liên hệ với nó liên lạc với bất kỳ máy chủ ảo nào, reverse proxy là một trung gian cho các máy chủ liên hệ với nó được liên lạc bởi bất kỳ client nào. Ưu điểm lớn nhất của việc sử dụng reverse proxy là khả năng quản lý tập trung. Nó giúp kiếm soát mọi request do clieny gửi lên các server được bảo vệ.

Proxy server là một máy chủ trung gian hoặc trung gian chuyển tiếp các yêu cầu nội dung từ nhiều máy khách đến các máy chủ khác nhau trên Internet. Reverse proxy server là một proxy server thường nằm sau tưởng lửa trong một mạng riêng và điều hướng các yêu cầu của client đến máy chủ thích hợp. Các reverse proxy thường được triển khai để giúp tăng cường bảo mật, hiện suất và độ tin cậy.

## CI/CD là gì?

CI/CD được viết tắt của Continuous Integration / Continuous Deployment. Đó là quá trình tự động hoá việc kiểm tra ( test ) mã nguồn, xây dựng ( build ) và triển khai ( deploy ) ứng dụng của mình lên các môi trường như Dev, Staging hoặc Production.  
**CI (Continuous Integration):** là một quá trình tự động hóa trong phát triển phần mềm, cho phép các thành viên trong team kiểm tra và hợp nhất mã nguồn của họ vào kho chung một cách tự động và định kỳ. Mỗi lần được hợp nhất thì được xây dựng 1 cách tự động để phát hiện lỗi phát sinh 1 cách nhanh nhất có thể.  
Quy trình của CI được diễn ra như sau:

* **Tích hợp mã nguồn:** Các lập trình viên commit và push code của mình lên repository.
* **Kích hoạt CI:** CI server giám sát repository và kiểm tra liên tục sự xem có sự thay đổi nào hay không. Khi phát ra sự thay đổi, CI server bắt đầu quá trình CI.
* **Build code**: Với code mới nhất trên repository, CI server sẽ build mã nguồn này thành một phần mềm hoàn chỉnh.
* **Chạy các test case:** Sau khi quá trình build hoàn tất, công cụ CI server bắt đầu chạy kiển tra đơn vị (unit test), kiểm tra tích hợp (integration test), kiểm tra giao diện (UI test) hoặc các kiểm tra khác tùy thuộc vào yêu cầu của dự án.
* **Đưa ra thông báo:** Nếu có lỗi trong quá trình kiểm tra, CI server sẽ đưa ra thông báo lỗi cho các lập trình viên biết. Các thông báo này thường được gửi qua email, chat...
* **Hợp nhất mã nguồn:** Nếu không có lỗi, CI server sẽ tự động hợp nhất mã nguồn của các lập trình viên vào một phiên bản mới nhất của phần mềm. Phiên bản mới này được lưu trữ trong kho chung của dự án và có thể được triển khai vào các môi trường khác nhau của dự án.
* **CD ( Continuos Deployment ):** Là quá trình triển khai tự động sau khi quá trình CI được diễn ra thành công. Quá trình CD sẽ được kích hoạt để triển khai ứng dụng của chúng ta vào các môi trường của dự án. Các gói phần mềm đã build thành công được triển khai bằng cách sử dụng các công cụ tự động hoặc được triển khai thủ công bằng tay. Điều này đảm bảo rằng phần mềm được cài đặt đúng cách và hoạt động tốt trên các môi trường

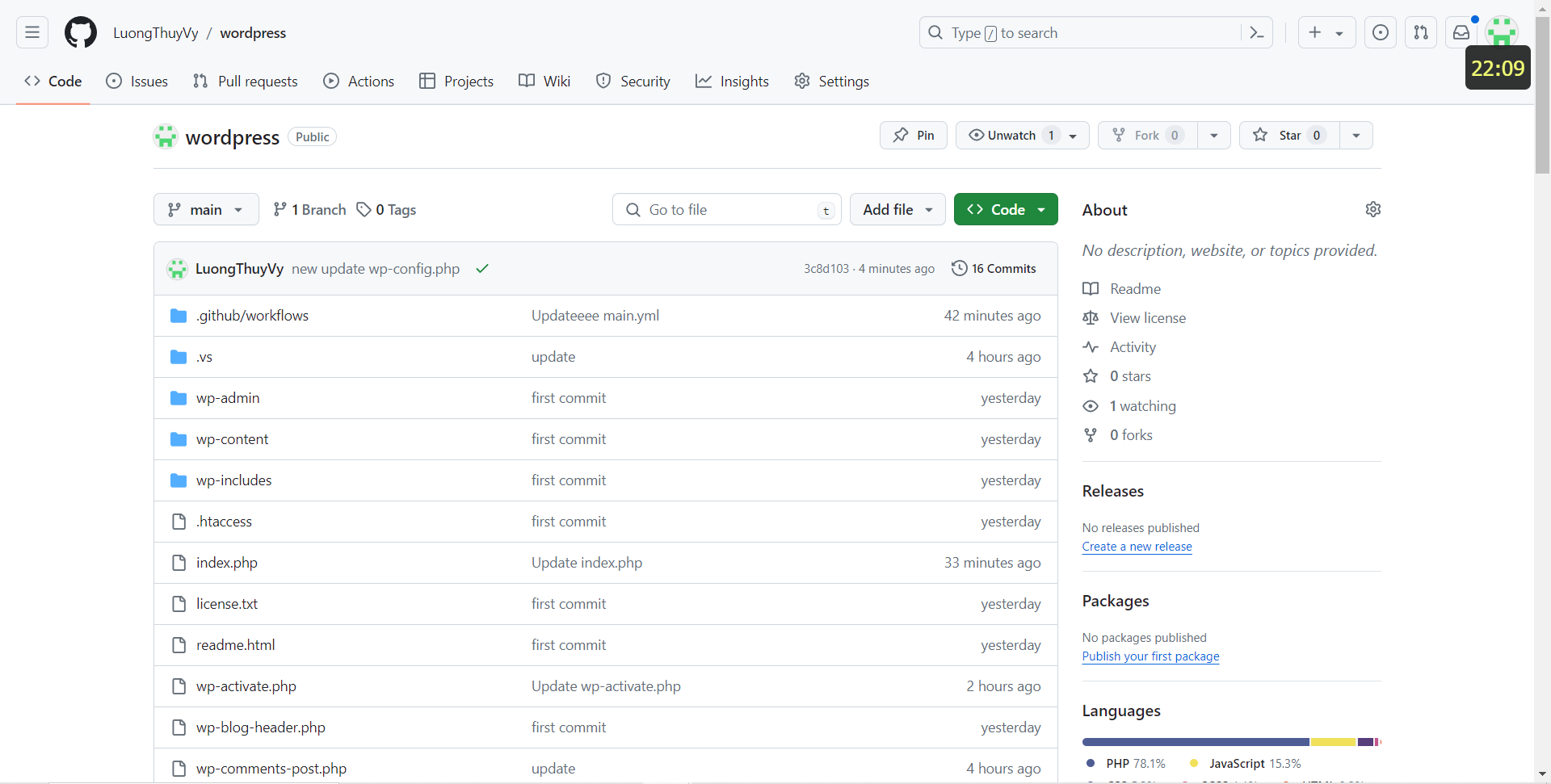
## Đưa code lên Github

Bước 1: Tạo tài khoản Github

Bước 2: Tạo một repository mới trên Github

Bước 3: Clone repo về máy

Bước 4: Thêm/sửa/xóa file/thư mục trên repo vừa clone về, sau đó push lên Github



## GitHub Action là gì?

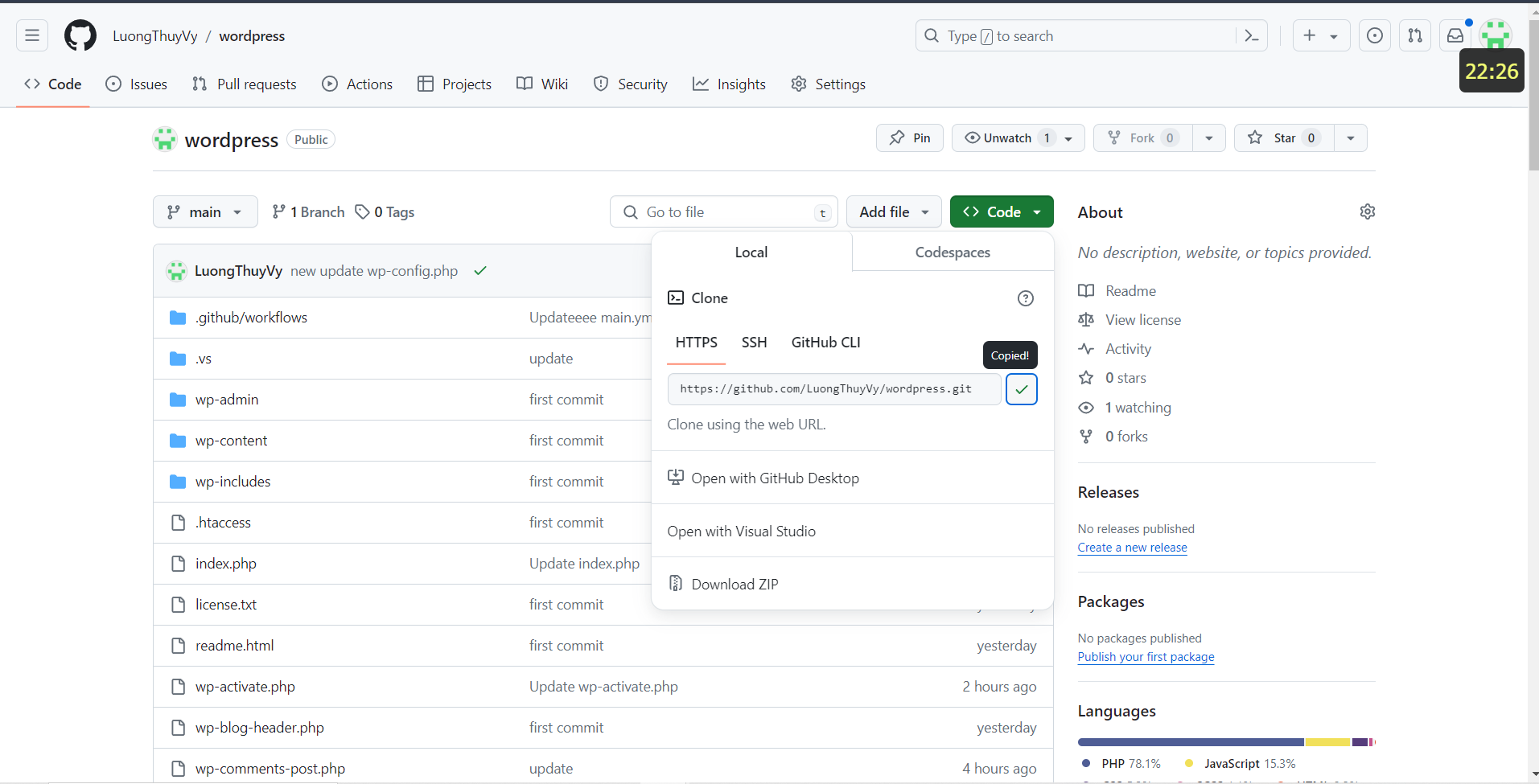
* GitHub Actions là 1 nền tảng miễn phí do GitHub cung cấp để giúp chúng ta tự động hoá quá trình CI/CD, cho phép người dùng định nghĩa các workflow tự động hoá các hoạt động trong phát triển phần mềm.
* Mỗi workflow trong GitHub Actions là một tập hợp các hành động (actions) được định nghĩa trong file YAML. Các actions này có thể là các lệnh cụ thể như: build ứng dụng, test, triển khai ứng dụng. Bạn có thể sử dụng các actions được cung cấp sẵn bởi GitHub, các actions mà cộng đồng lập trình viên tạo ra hoặc tạo các hành động tùy chỉnh để phù hợp với nhu cầu của dự án của mình.

Các bước thực hiện:

1. **Clone project về máy.**

Vào project [*https://github.com/LuongThuyVy/wordpress*](https://github.com/LuongThuyVy/wordpress.git%20) trên github và copy ulr (như hình bên dưới)

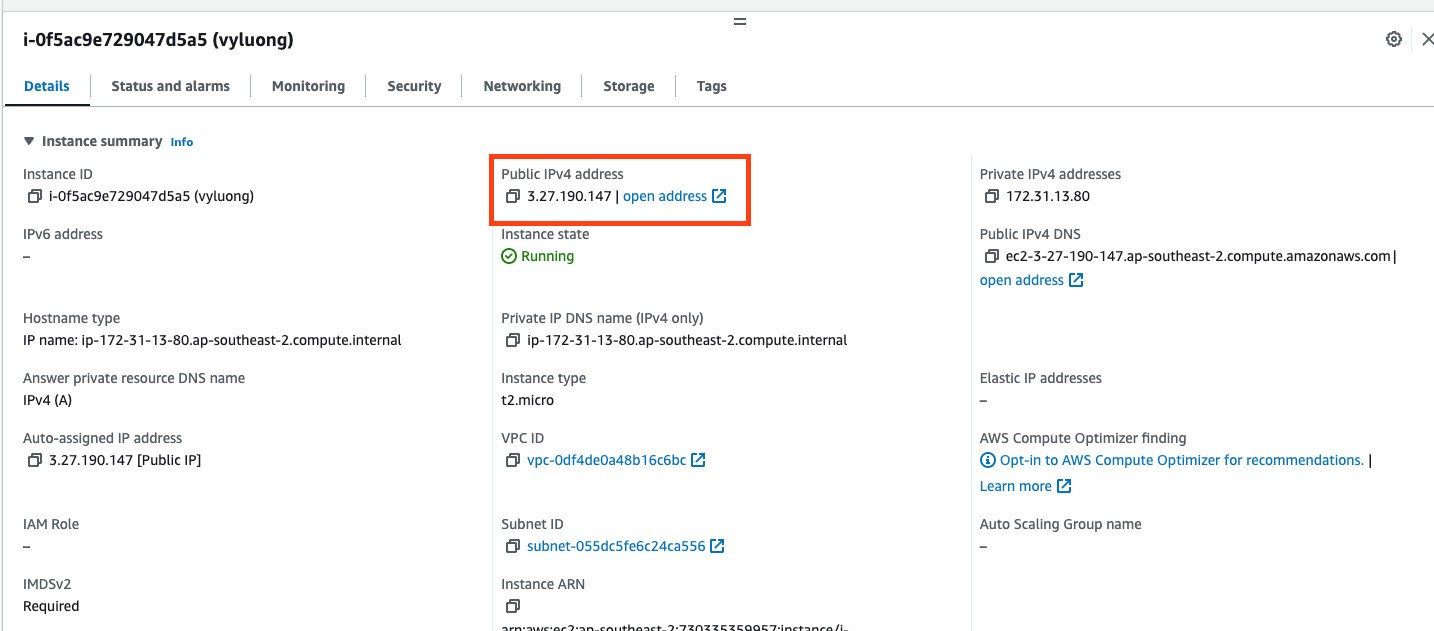
Clone project về máy dùng lệnh: *git clone https://github.com/LuongThuyVy/wordpress.git*



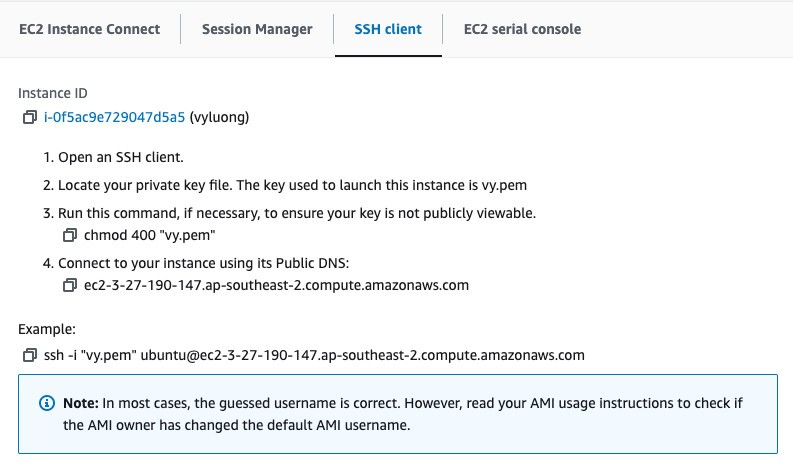
1. **Cấu hình** 
   * Tạo instance của EC2 trên AWS (dùng để triển khai dự án), sau khi tạo thành công sẽ đưa về như sau:

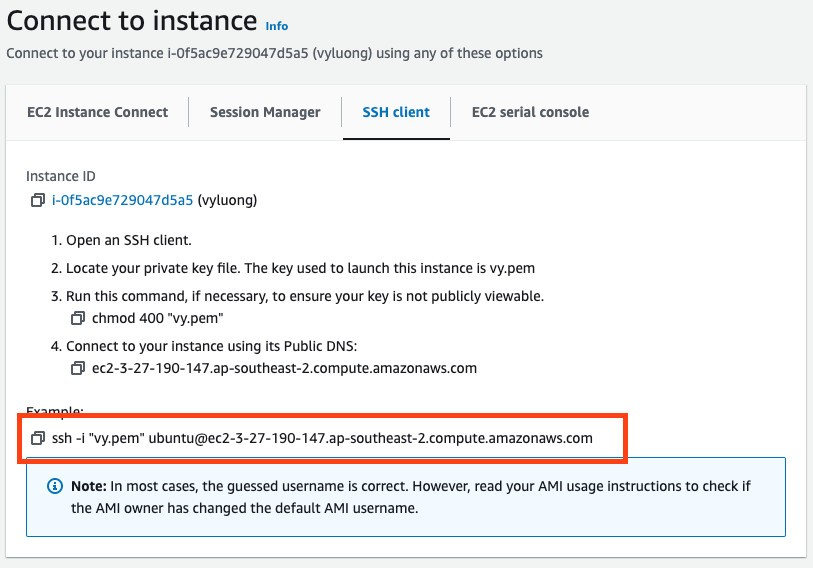
****

* + Xem thông tin chi tiết (dùng để sau này truy cập bằng IP pubic)

****

* + Cách kết nối thông qua SSH (hoặc được gọi là Secure Shell, là một giao thức điều khiển từ xa cho phép người dùng kiểm soát và chỉnh sửa server từ xa qua Internet.)

****

****

1. **Triển khai**

**Bước 1: Cài đặt các Dependencies**

>sudo apt update

>sudo apt install apache2 \

ghostscript \

libapache2-mod-php \

mysql-server \

php \

php-bcmath \

php-curl \

php-imagick \

php-intl \

php-json \

php-mbstring \

php-mysql \

php-xml \

php-zip

1. **Cấu hình Apache cho WordPress**

**Bước 1:** Tạo folder đích: sudo mkdir /src/wordpress

**Bước 2:** Cấp quyền cho thư mục

>sudo shmod -R 777 /src/wordpress

Tạo file conf cho wordpress

>sudo Nano /etc/apache2/sites-available/wordpress.conf

<VirtualHost \*:80>

DocumentRoot /src/wordpress

<Directory /src/wordpress>

Options FollowSymLinks

AllowOverride Limit Options FileInfo

DirectoryIndex index.php

Require all granted

</Directory>

<Directory /src/wordpress/wp-content>

Options FollowSymLinks

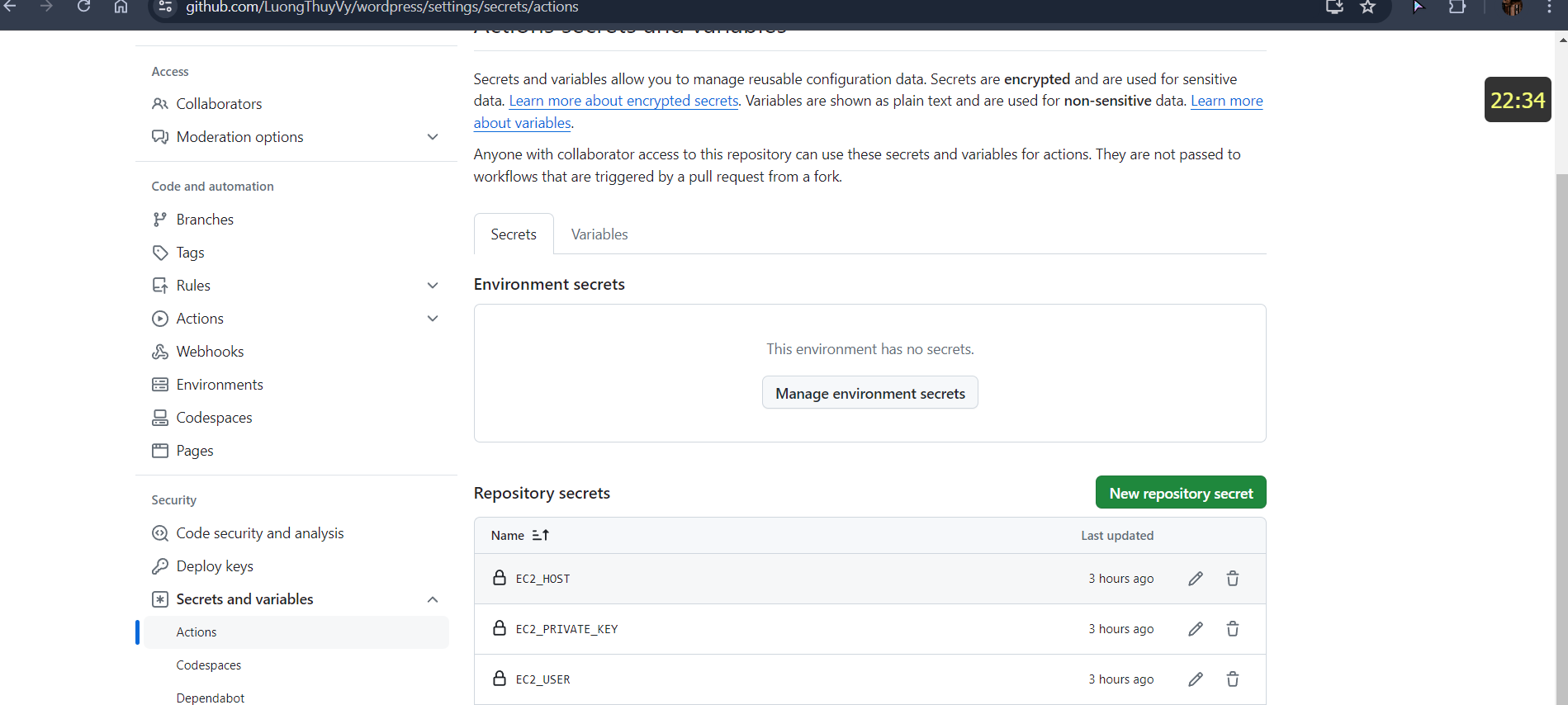
Require all granted

</Directory>

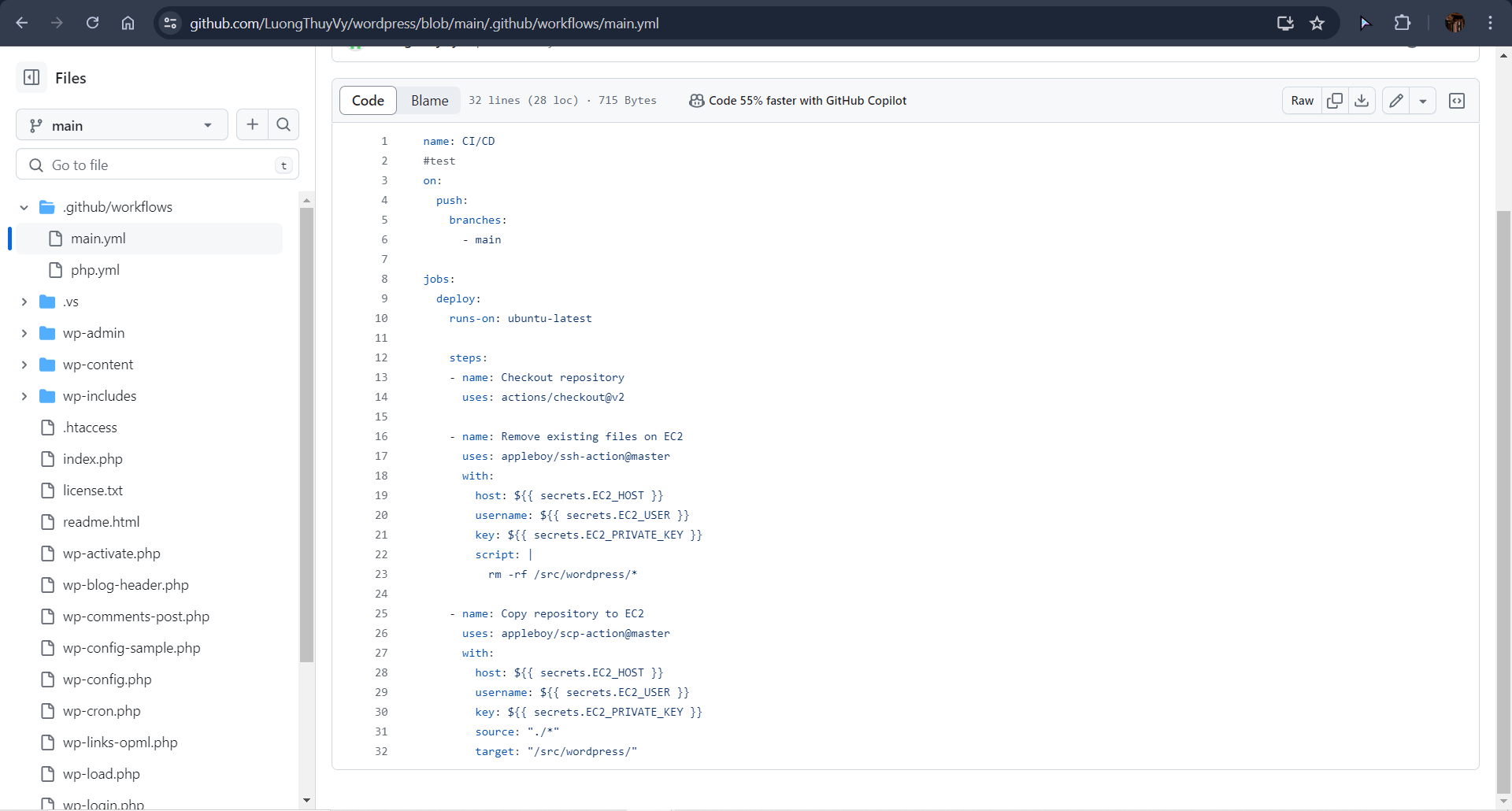
</VirtualHost>

## Thiết lập CI/ CD

Các giá trị enviroment được lưu trữ trong **Github Security** với các giá trị như sau:



* Trong thư mục *./.github/workflows/main.yml*. Project github định dạng workflow như sau:

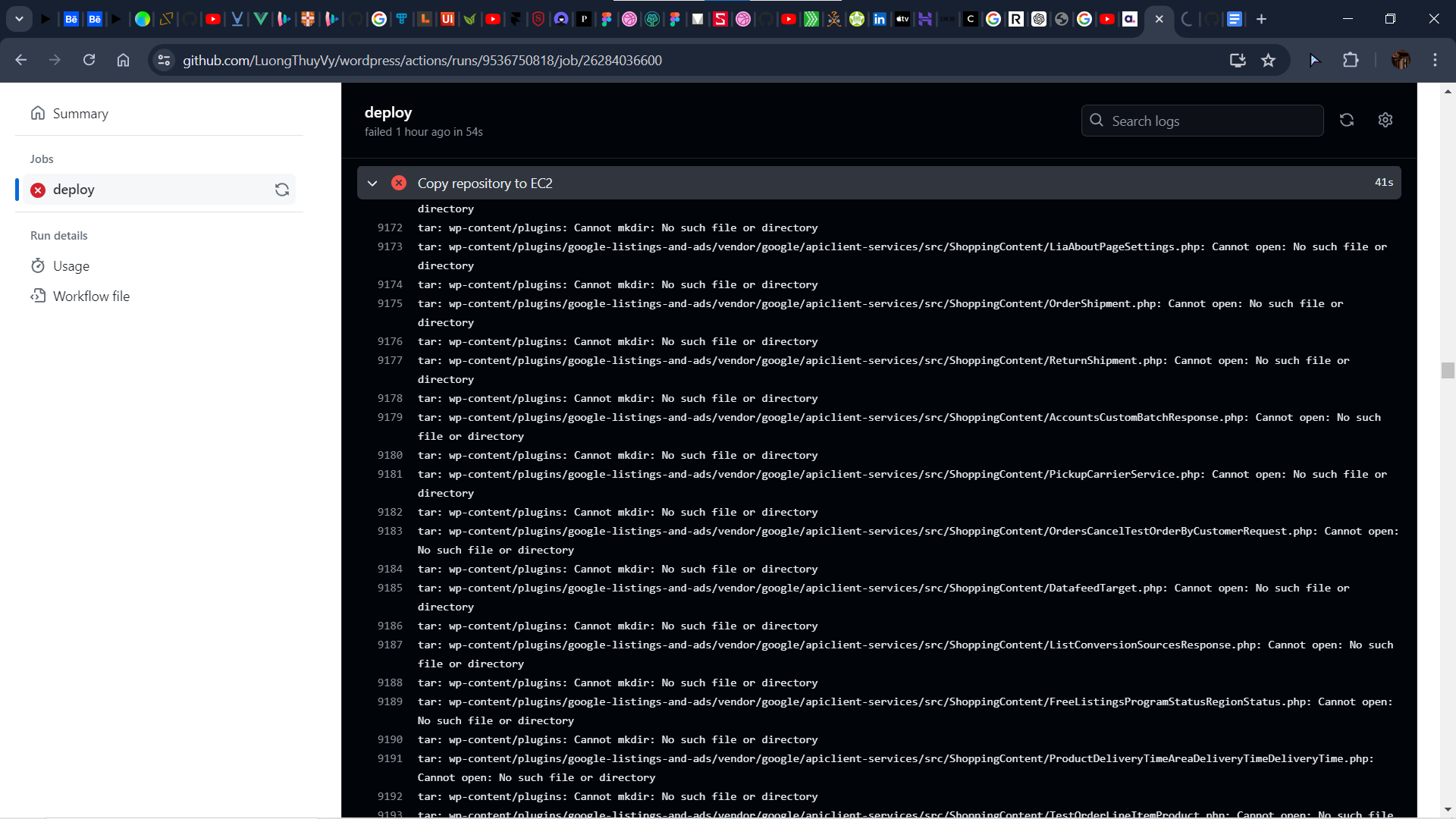


Trong đó, cấu hình bao gồm:

* + Name: tên của workflow
  + On: sự kiện kích hoạt workflows
  + Jobs: định nghĩa các công việc cần thực hiện trong workflow
  + Build: tên của jobs
  + Run-on: môi trường chạy job ubuntu-lastest
  + Steps: định nghĩa các bước cần thực hiện trong job
  + Name: tên của bước hiện tại
  + Uses: sử dụng action để lấy mã nguồn từ respository về môi trường làm việc
  + Run: các lệnh shell để cài đặt ác depencies (phụ thuộc) hoặc lệnh để kiểm tra mã nguồn

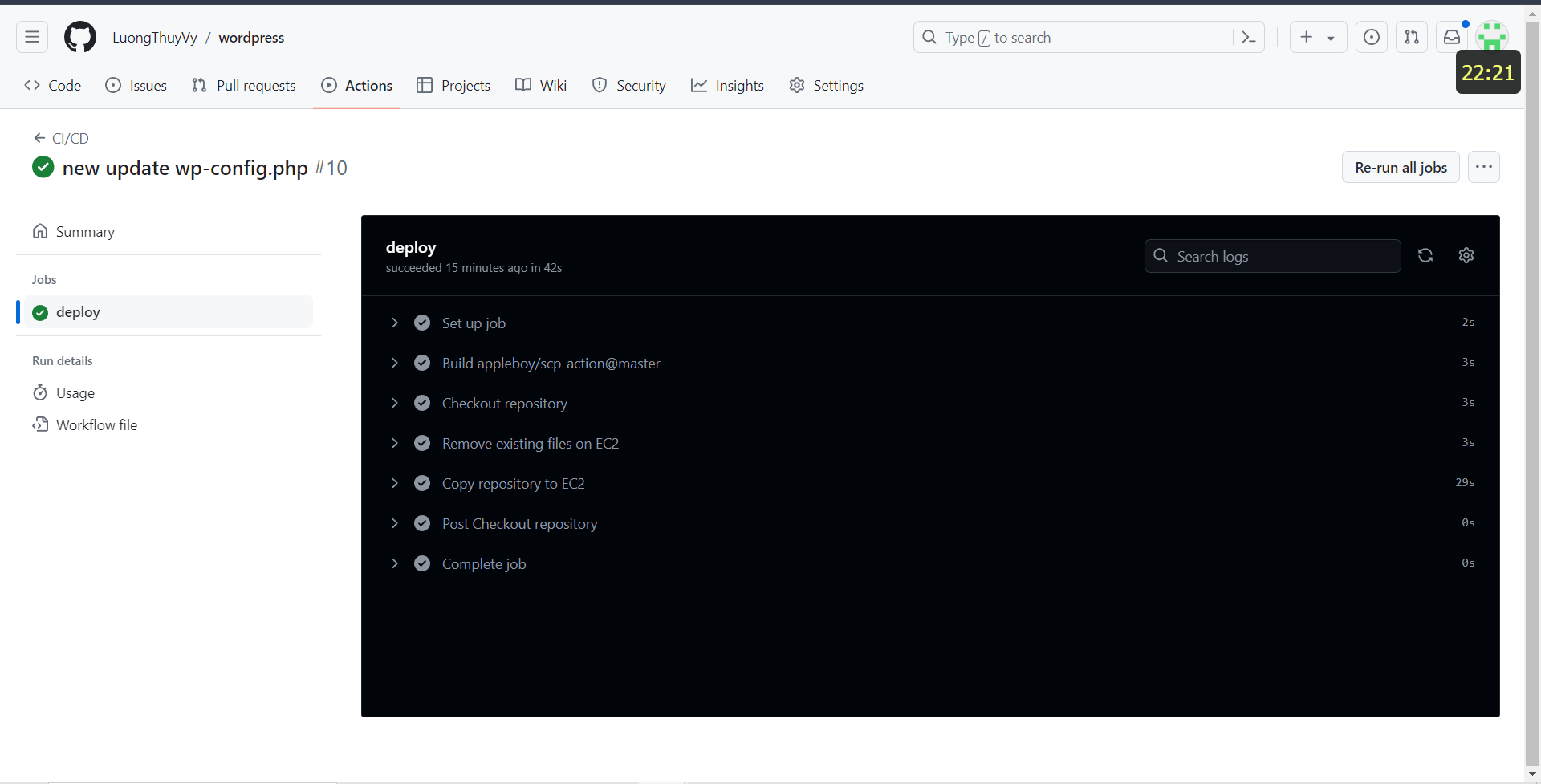
Sau khi config thành công, mỗi lần push hoặc pull request sẽ tự động chạy workflows để kiểm tra.

Nếu không thành công vượt qua các bài test sẽ hiển thị như sau:



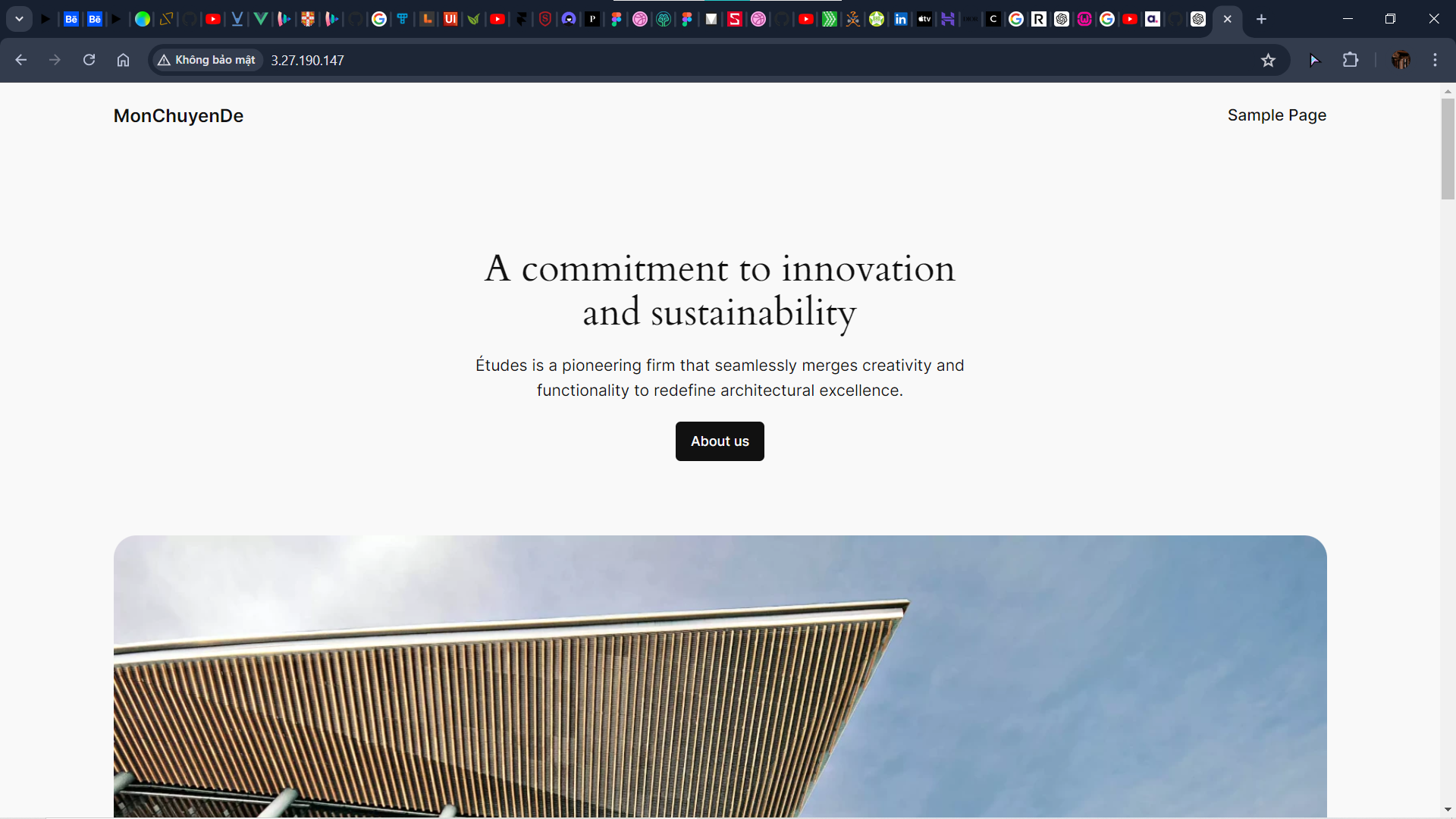
## Kết quả demo

Nếu thành công vượt qua các bài test thì mục Actions trên repository sẽ hiển thị như sau:



## Khởi chạy sản phẩm trên public IP

* Giờ đây ta có thể truy cập website trên <http://3.27.190.147/> và thực hiện thiết lập **Wordpress** và **Ecommerce** (như đã thiết lập trước đó):
* *(Note: Các bước thiết lập tương tự như phần thiết lập ở môi trường* ***development****)*

**