**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG TP. HỒ CHÍ MINH**



**TÀI LIỆU CÀI ĐẶT**

**CHUYỀN NGÀNH CÔNG NGHỆ PHẦN MỀM**

**HỆ CHÍNH QUI**

**NIÊN KHÓA 2020 - 2025**

**Đề tài : CÀI ĐẶT CMS, E-COMMERCE , REVERSE PROXY , CI/CD.**

**Giáo viên hướng dẫn: Th.S Lê Hà Thanh**

**Sinh viên thực hiện: Nguyễn Thanh Hữu.**

**Lớp: D20CQCNPM02-N**

**Mã sv: N20DCCN108**

TP.HCM - THÁNG 06 NĂM 2024

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Catalog

**[I. CÀI ĐẶT CMS 3](#_Toc14597)**

[1. Cài đặt image cho mysql và wordpress, reverse proxy nginx. 3](#_Toc29661)

[2. Chạy trình duyệt web wordpress. 6](#_Toc14384)

**[II.CÀI ĐẶT E-COMMERCE. 9](#_Toc12500)**

[1. Cài đặt Plugin Plugin. 9](#_Toc18813)

**[III.CÀI ĐẶT CICD VỚI GIT ACTION. 10](#_Toc15731)**

[1. Tạo 1 repository trên github. 10](#_Toc5812)

[2. Tạo dự án test với Reactjs Vite. 11](#_Toc24247)

[3. Tạo file node.js.yml triển khai CICD lên git. 12](#_Toc29129)

[4. Chạy demo test. 12](#_Toc1010)

1. **CÀI ĐẶT CMS**
2. **Cài đặt image cho mysql và wordpress, reverse proxy nginx.**

Trong file docker-compose.yml ,cấu hình cho image mysql và wordpress.

version: "3.9"

services:

  db:

    image: mysql:5.7

    volumes:

      - ./db\_data:/var/lib/mysql

    restart: always

    environment:

      MYSQL\_ROOT\_PASSWORD: somewordpress  # Replace with a strong password

      MYSQL\_DATABASE: wordpress

      MYSQL\_USER: wordpress

      MYSQL\_PASSWORD: wordpress  # Replace with a strong password

    networks:

      - app-network

  wordpress:

    image: wordpress:latest

    volumes:

      - ./wordpress:/var/www/html

      - ./plugins:/var/www/html/wp-content/plugins

    ports:

      - "8000:80"

    restart: always

    environment:

      WORDPRESS\_DB\_HOST: db:3306

      WORDPRESS\_DB\_USER: wordpress

      WORDPRESS\_DB\_PASSWORD: wordpress

      WORDPRESS\_DB\_NAME: wordpress

    networks:

      - app-network

  nginx:

    build: ./nginx-conf

    restart: unless-stopped

    ports:

      - "80:80"

    volumes:

      - ./wordpress:/var/www/html

      - ./nginx-conf:/etc/nginx/conf.d

    networks:

      - app-network

volumes:

  db\_data:

networks:

  app-network:

*Hình 1.1 Mã nguồn built docker image.*

Cấu hình config Nginx:

server {

    listen 80;  # Adjust if using a different port

    server\_name localhost;  # Replace with your domain name (optional)

    location / {

        root /var/www/html;

        index index.php index.html index.htm;

        # Pass PHP requests to PHP-FPM (adjust based on your image)

        try\_files $uri /index.php?$query\_string;

    }

    location ~ \.php$ {

        fastcgi\_pass wordpress:8000;  # Adjust port if different

        fastcgi\_index index.php;

        include /etc/nginx/fastcgi\_params;

    }

}

*Hình 1.2 Mã nguồn built config nginx.conf .*

Cấu hình image cho nginx:

FROM nginx:stable-alpine

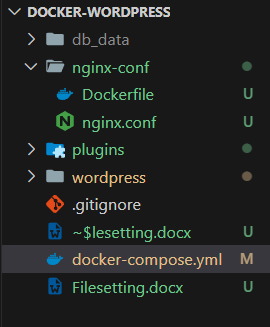
COPY nginx.conf /etc/nginx/conf.d/default.conf

EXPOSE 80

*Hình 1.3 Mã nguồn built image nginx .*

Tiếp theo chạy lệnh **docker-compose up -d** trên terminer để chạy các image hoặc có thể nhấn chuột phải vào file docker compose và chọn **compose up**.

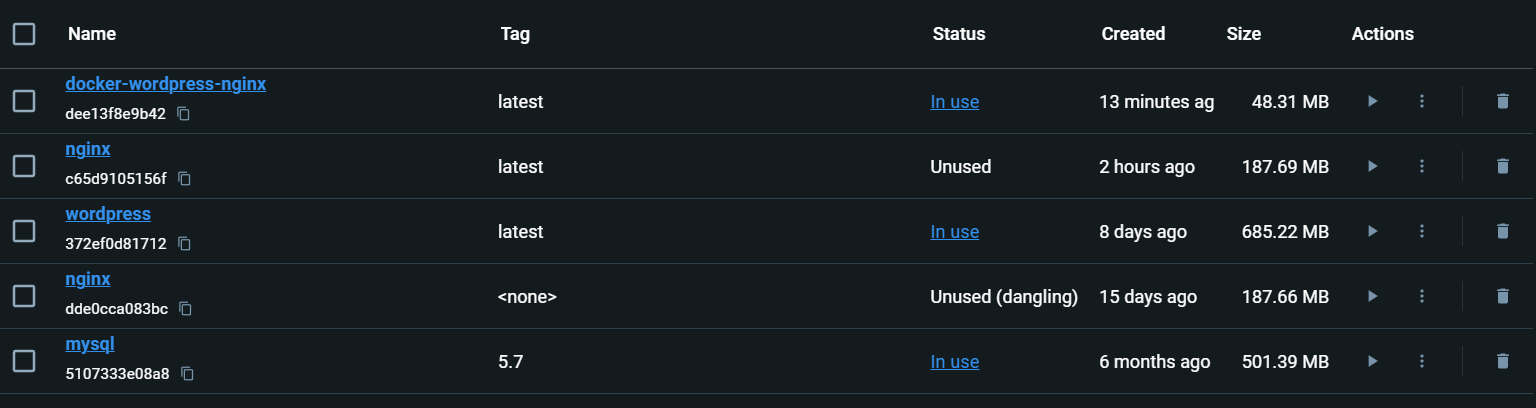
Sau khi chạy thành công , các forder mới trong file sẽ như sau:



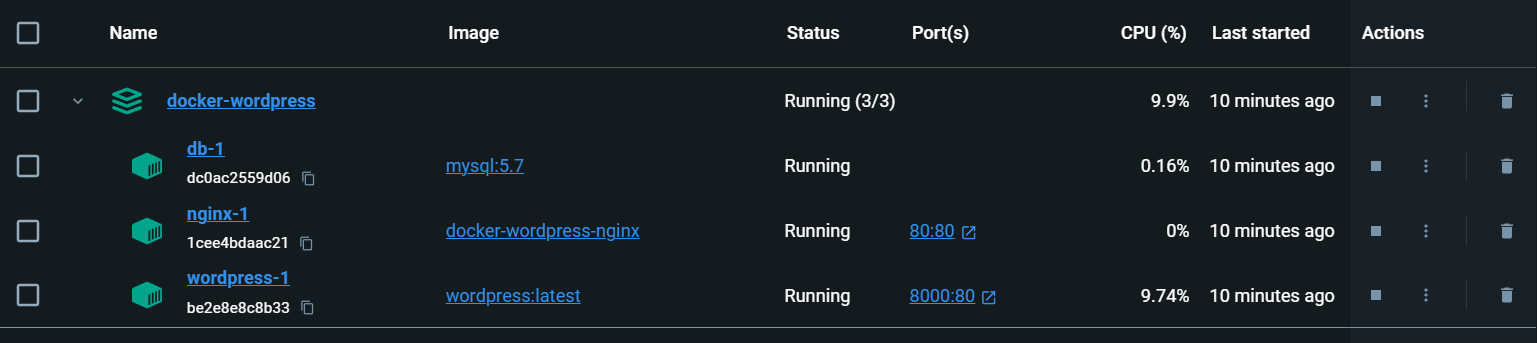
*Hình 1.4 Cấu trúc thư mục forder.*

Docker sẽ sinh ra các volumes để lưu trữ data mysql. Đồng thời sinh ra forder wordpress để lưu trữ mã nguồn của wordpress.

Trong docker container sẽ như sau:



*Hình 1.5 Các image mysql và worpress.*

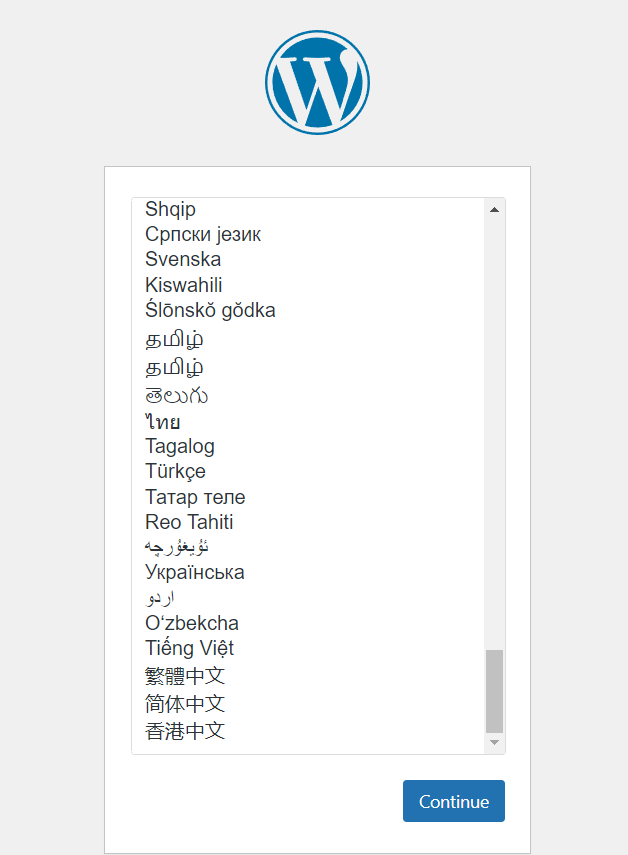


*Hình 1.6 Các container của mysql và wordpress chạy cùng netwwork.*

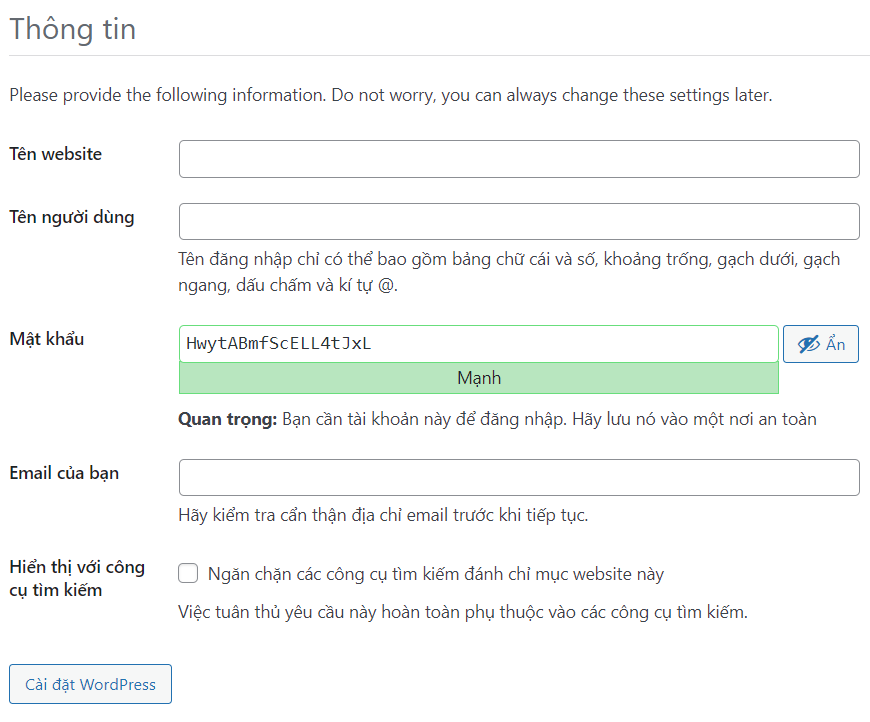
1. **Chạy trình duyệt web wordpress.**

Chạy ***<http://localhost:8000/>*** *, màn hình trên trình duyệt sẽ như sau.*

Chọn Tiếng Việt-> nhập các thông tin cần thiết theo yêu cầu ->

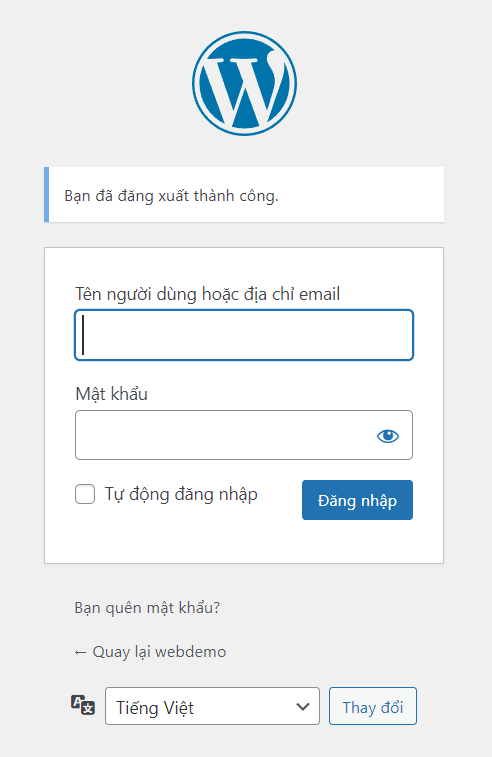


*Hình 1.7 Màn hình hiển thị đầu tiên khi chạy .*



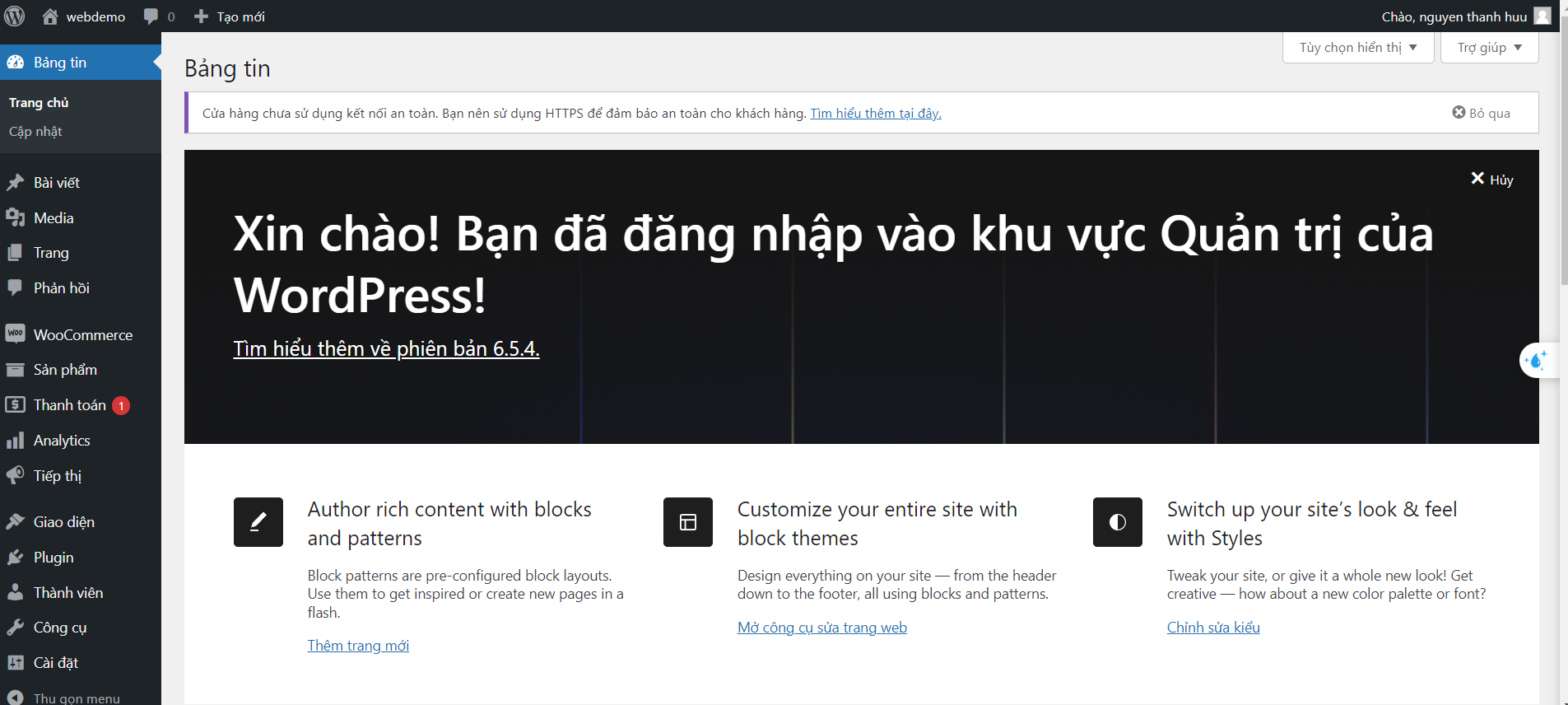
*Hình 1.8 Màn hình hiển thị yêu cầu nhập thông tin người dùng.*

Nhập username và password đã tạo trước đó :



*Hình 1.9 Màn hình hiển thị nhập tên người dùng (email) password.*

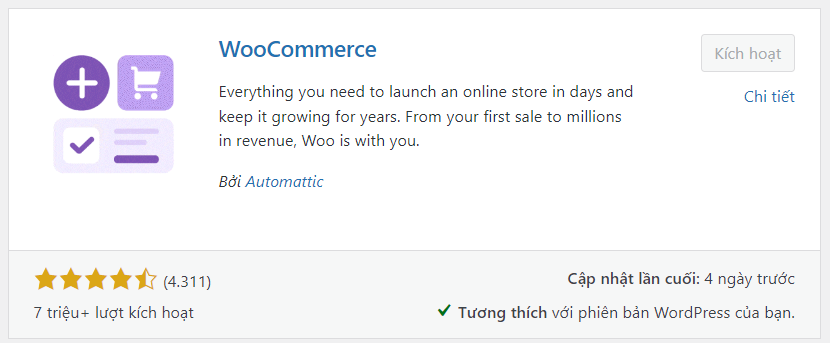
Màn hình giao diện chính của wordpress khi đăng nhập thành công:



*Hình 1.10 Màn hình hiển thị giao diện chính..*

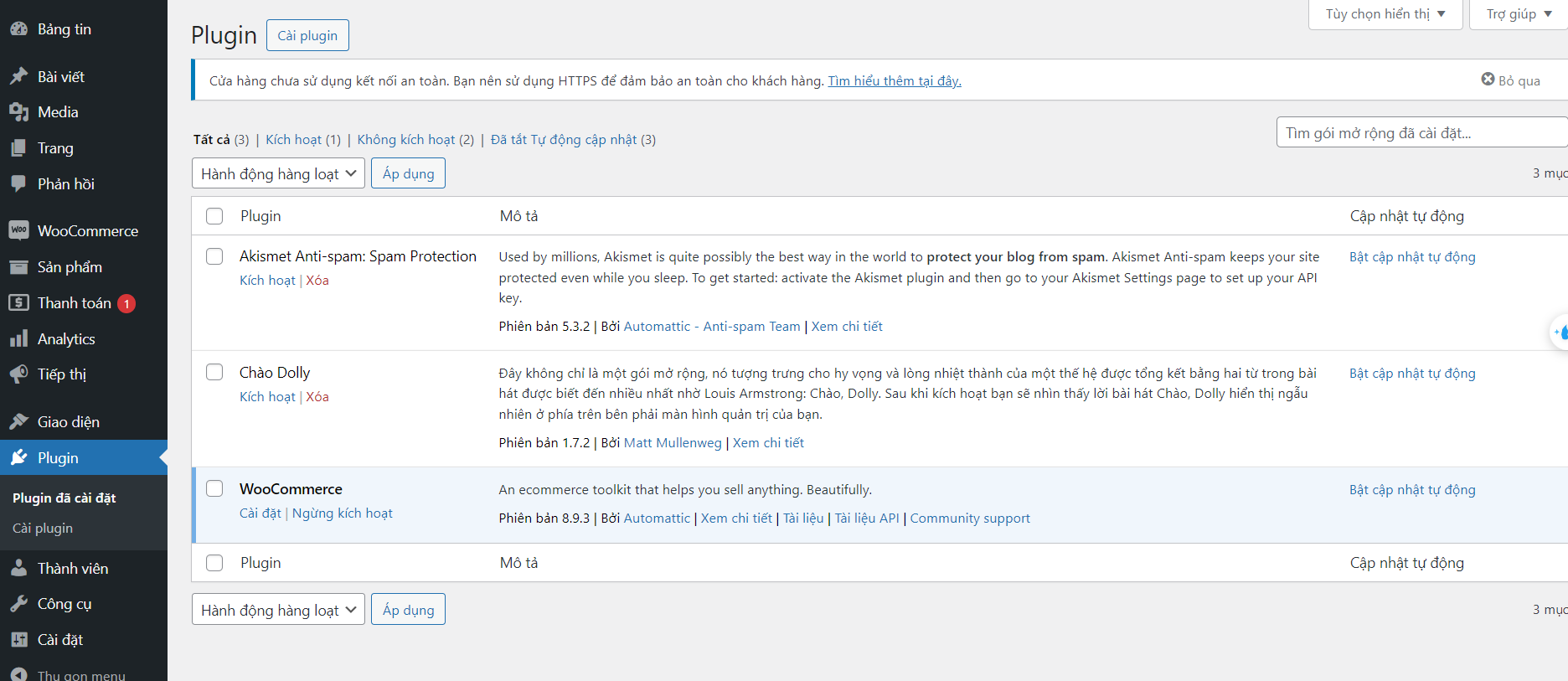
1. **CÀI ĐẶT E-COMMERCE.**
2. **Cài đặt Plugin Plugin.**

Cài đặt plugin woocommerce: Chọn thanh Plugin-> nhập Pluginvào ô tìm kiếm-> nhấn cài đặt-> kích hoạt.

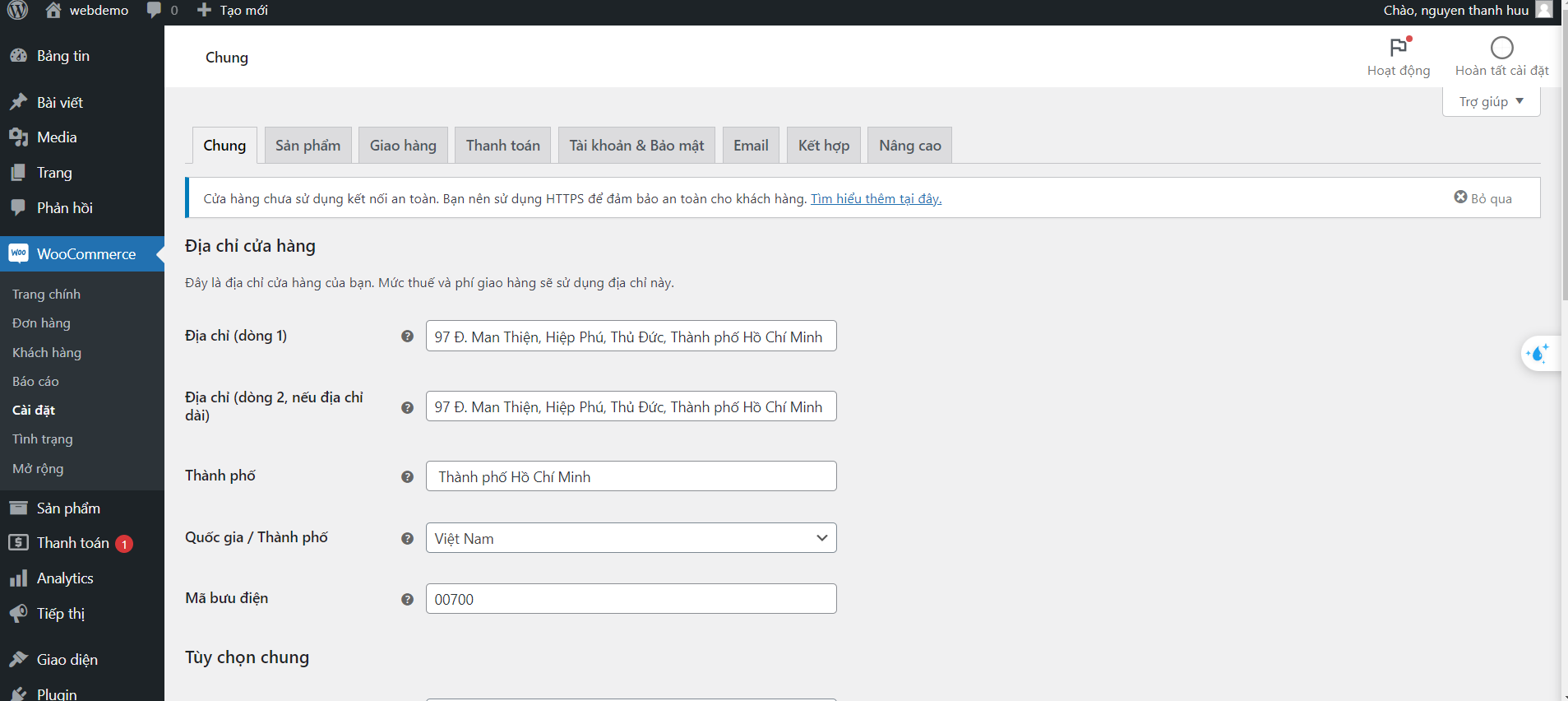


*Hình 2.1 Màn hình hiển thành phần* woocommerce *.*

*Cấu hình cho plugin* woocommerce: *Chọn cài đặt của plugin* woocommerce-> Thêm các thông tin yêu cầu-> Lưu thay đổi ->



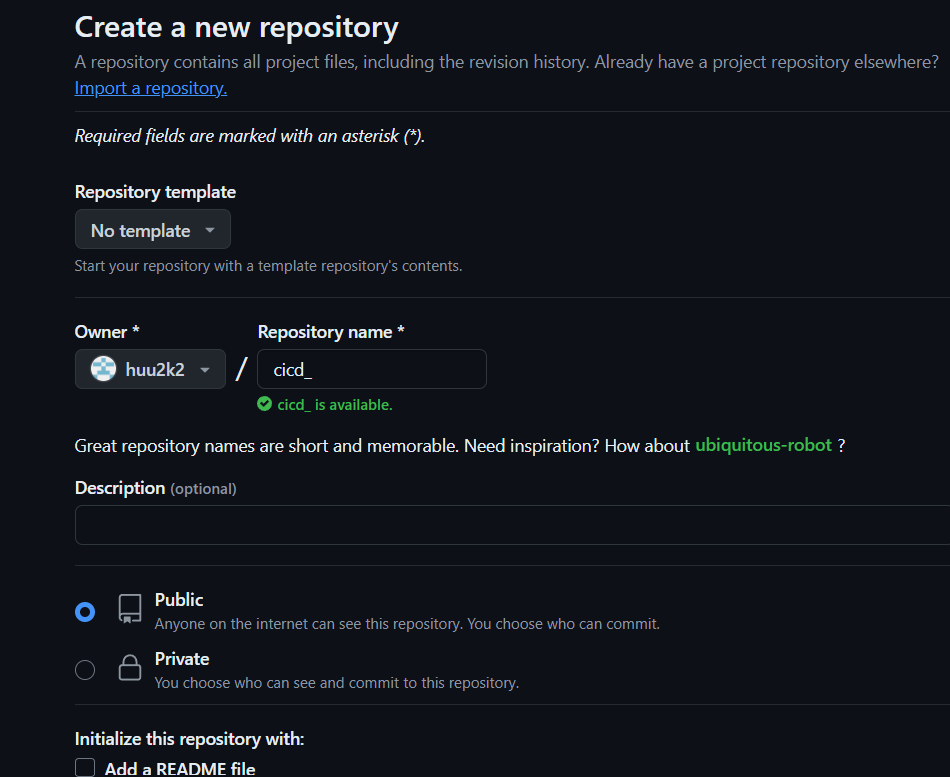
*Hình 2.2 Màn hình hiển thành plugin* woocommerce *.*



*Hình 2.3 Màn hình nhập thông tin cho* woocommerce .

1. **CÀI ĐẶT CICD VỚI GIT ACTION.**
2. **Tạo 1 repository trên github.**

Vào github cá nhân tạo 1 repository mới. Và cài đặt theo hướng dẫn.



*Hình 2.1 Màn hình tạo repository mới.*

1. **Tạo dự án test với Reactjs Vite.**

Tạo 1 dự án Reactjs với Vite bằng các câu lệnh sau:



*Hình 2.2 Câu lệnh tạo dự án Reactjs.*

Tiếp theo ta cần push dự án này lên repository ta vừa tạo bằng các lệnh sau .

***- git init***

***- git add .***

***- git commit -m ‘first deploy’***

***- git push -u origini main***

Chú ý : Hãy tham khảo về các lệnh git cơ bản.

1. **Tạo file node.js.yml triển khai CICD lên git.**

Trong forder chính tạo forder .github/workflows, tạo thêm file node.js.yml để chưa mã nguồn sau:

name: Node.js CI

on:

  push:

    branches: ["main"]

  pull\_request:

    branches: ["main"]

permissions:

  contents: read

  pages: write

jobs:

  build:

    runs-on: ubuntu-latest

    strategy:

      matrix:

        node-version: [20.x]

    steps:

      - name: Checkout repository

        uses: actions/checkout@v4

      - name: Set up Node.js ${{ matrix.node-version }}

        uses: actions/setup-node@v3

        with:

          node-version: ${{ matrix.node-version }}

          cache: "npm"

      - name: Install dependencies

        run: npm ci

      - name: Build project

        run: npm run build

      - name: List dist directory contents

        run: ls -la dist # Debug step to confirm dist directory exists and is populated

      - name: List dist run test

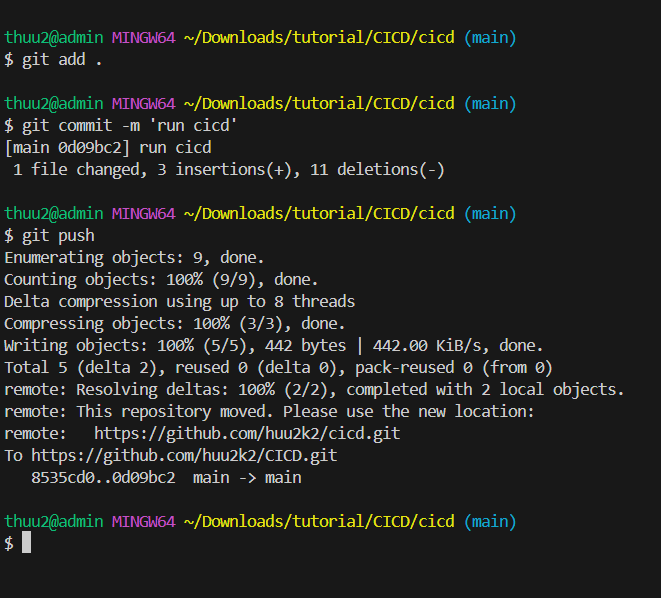
        run: npm run test

*Hình 2.3 Config cicd với môi trường Node.js.*

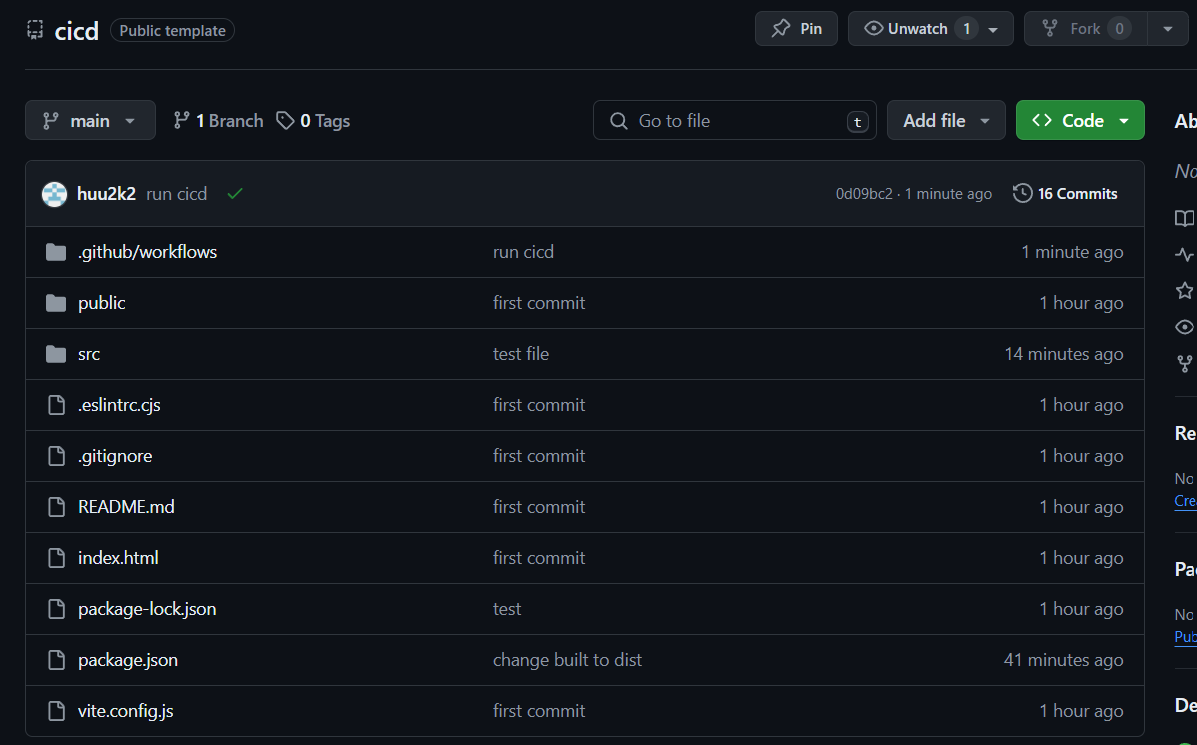
1. **Chạy demo test.**

Để chạy được với file cấu hình trên thì chúng ta sẽ phải cài thêm 1 số thư viện như : ***npm i jest .***

Tiếp theo ta sẽ thêm 1 số code mới để chạy giao diện. Và chạy các lệnh để push code mới lên .



*Hình 2.4 Các lệnh deploy dự án.*

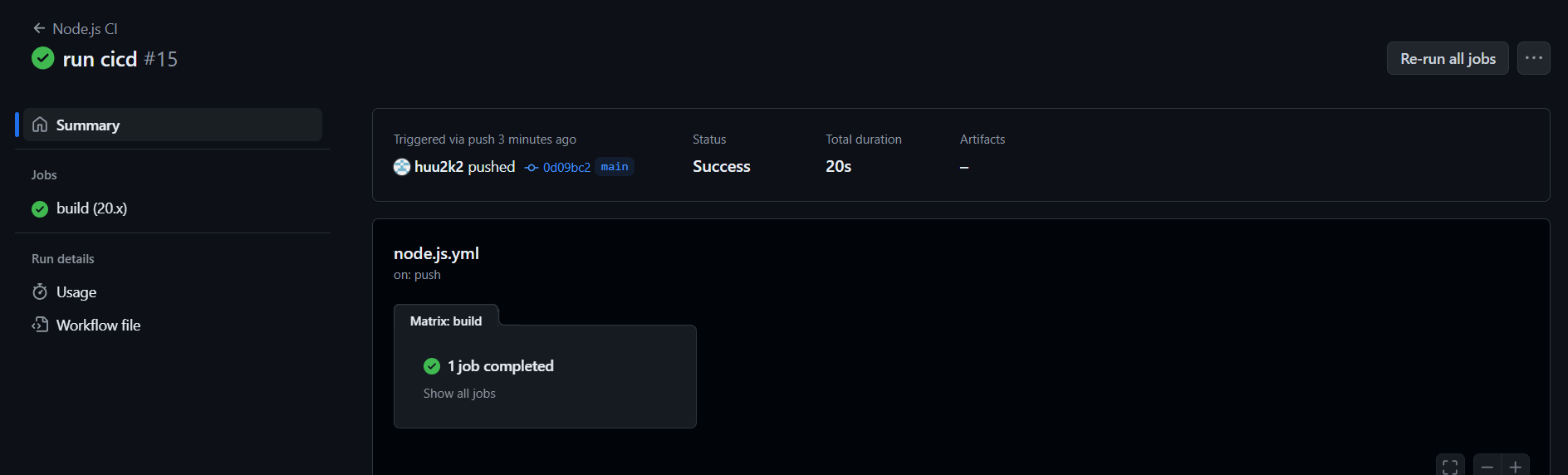


*Hình 2.5 Màn hình trên github.*

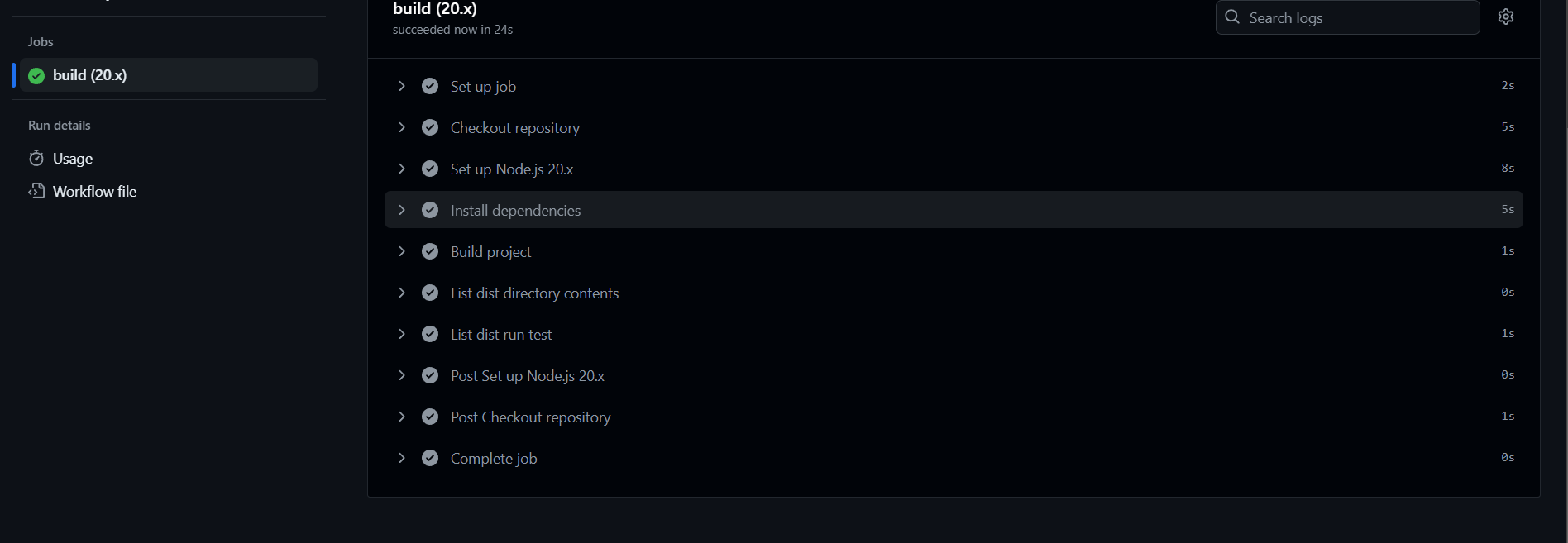
Ta vào Action -> click **run cicd :**



*Hình 2.6 Màn hình Actions nhiều workflows*



*Hình 2.7 Màn hình workflows* ***run cicd***



*Hình 2.7 Màn hình task chạy và kiểm tra*