**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

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

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN 2**

---o0o---

**BÁO CÁO GIỮA KỲ & CUỐI KỲ**

**TÀI LIỆU HƯỚNG DẪN THIẾT LẬP VÀ TRIỂN KHAI PHẦN MỀM**

**Môn học:** Chuyên đề Hệ thống thông tin

**Giảng viên hướng dẫn:** Lê Hà Thanh

**Lớp:** D20CQCNPM01-N

**Sinh viên thực hiện:**

Văn Tố Hữu N20DCCN026

*TP.HCM, ngày 20 tháng 06 năm 2024*

**MỤC LỤC**

[**I. GIỚI THIỆU TÀI LIỆU 15**](#_Toc169763364)

[**II. TỔNG QUAN WEBSITE 15**](#_Toc169763365)

[**III. YÊU CẦU HỆ THỐNG 15**](#_Toc169763366)

[**IV. HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG 16**](#_Toc169763367)

[**4.1. Clone repository về máy tính cá nhân và cấu hình 16**](#_Toc169763368)

[**4.2. Triển khai cài đặt microservice với Kubernetes 17**](#_Toc169763375)

[**4.3. Triển khai cài đặt CMS (Reactjs) 21**](#_Toc169763391)

[**4.4. Triển khai cài đặt giao diện web movie (Reactjs) 23**](#_Toc169763394)

[**4.5. Triển khai CI/CD 25**](#_Toc169763397)

[**V. TÀI LIỆU THAM KHẢO 38**](#_Toc169763450)

# GIỚI THIỆU TÀI LIỆU

* Tài liệu này dùng để:
  + Triển khai cài đặt website sử dụng kỹ thuật Kubernetes cho mircroservice.
  + Triển khai cài đặt CMS (Movie cms) và cài đặt Frontend (Movie web).
  + Triển khai cài đặt Backend.
  + Triển khai sử dụng CI/CD bằng Github Webhook và Jenkins.
* Phạm vi: tài liệu này hướng dẫn triển khai ở môi trường phát triển (Development Environment) và môi trường kiểm thử (Testing Environment).

# TỔNG QUAN WEBSITE

* Các công nghệ sử dụng:
  + OS-level virtualization: Docker Desktop
  + Reverse proxy server: Ingress NGINX
  + Content Management System (CMS): Reactjs
  + Database: SQL Server

# YÊU CẦU HỆ THỐNG

* Yêu cầu phần cứng:
  + Hệ điều hành: Windows 10/11 64-bit.
  + Dung lượng RAM: tối thiểu 4GB.
  + Dung lượng ổ cứng: tối thiểu 10GB.
  + Kích hoạt ảo hóa phần cứng (hardware virtualization) trong BIOS.
  + Bật được tính năng WSL 2 trên Windows.
* Yêu cầu phần mềm:
  + Cài đặt sẵn Docker Desktop, Git.
  + Cài đặt sẵn IDE hoặc TextEditor như Visual Studio Code, Notepad++, …
  + Phần mềm quản lý mã nguồn như Git, …

# HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG

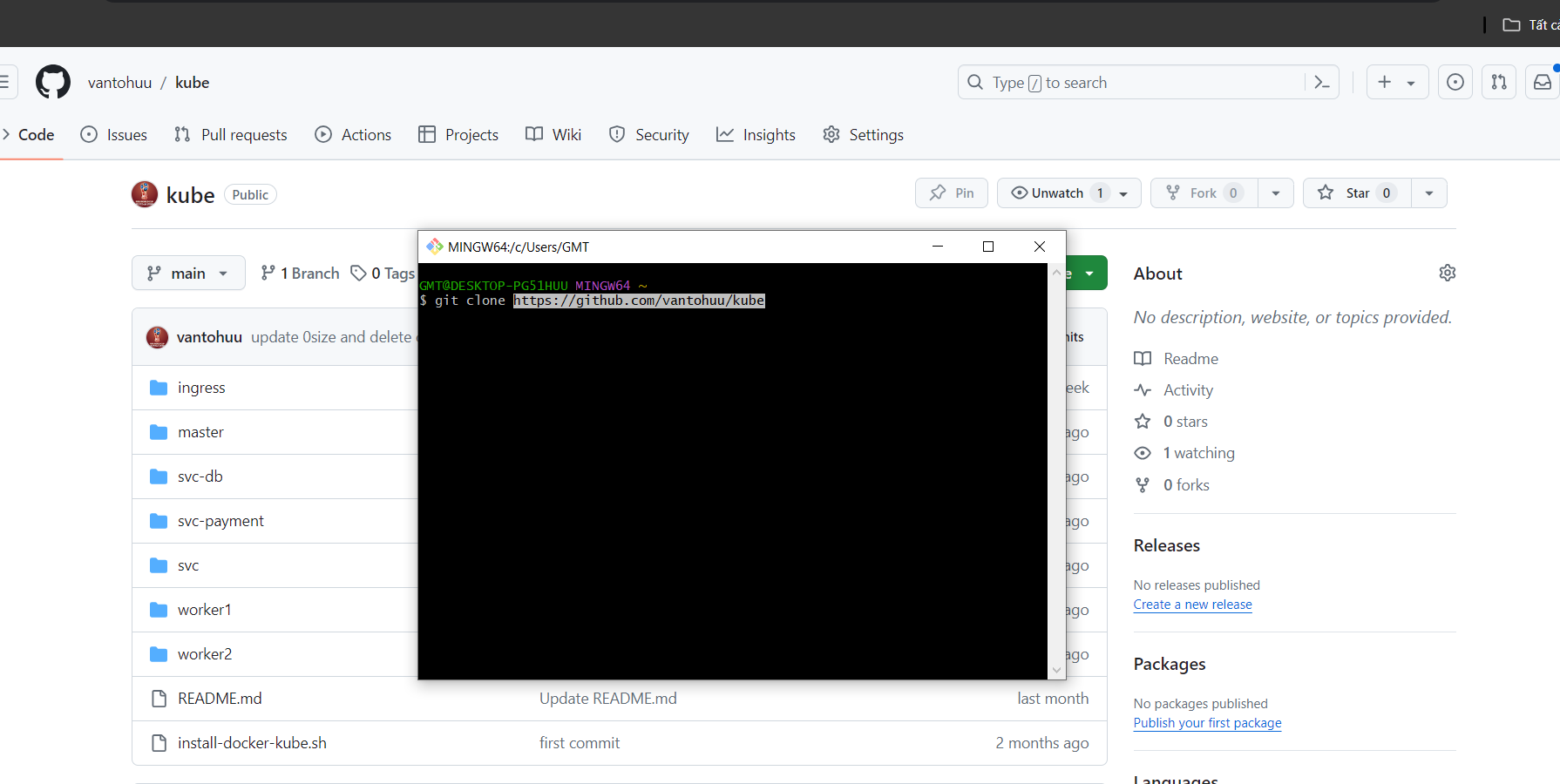
1. **Clone repository về máy tính cá nhân và cấu hình**

**Source code gồm 5 phần**

* Main service | Link: https://github.com/vantohuu/SOA-Backend
* Payment sevice | Link: https://github.com/vantohuu/payment-service.git
* Cms Movie | Link: https://github.com/lucvan02/web-xem-phim-admin
* Web Movie | Link: https://github.com/lucvan02/web-xem-phim
* Deploy kubernetes | Link: https://github.com/vantohuu/kube

**Tạo một thư mục trống**

**Thực hiện lệnh: git clone <link source> để clone remote repository về máy tính**

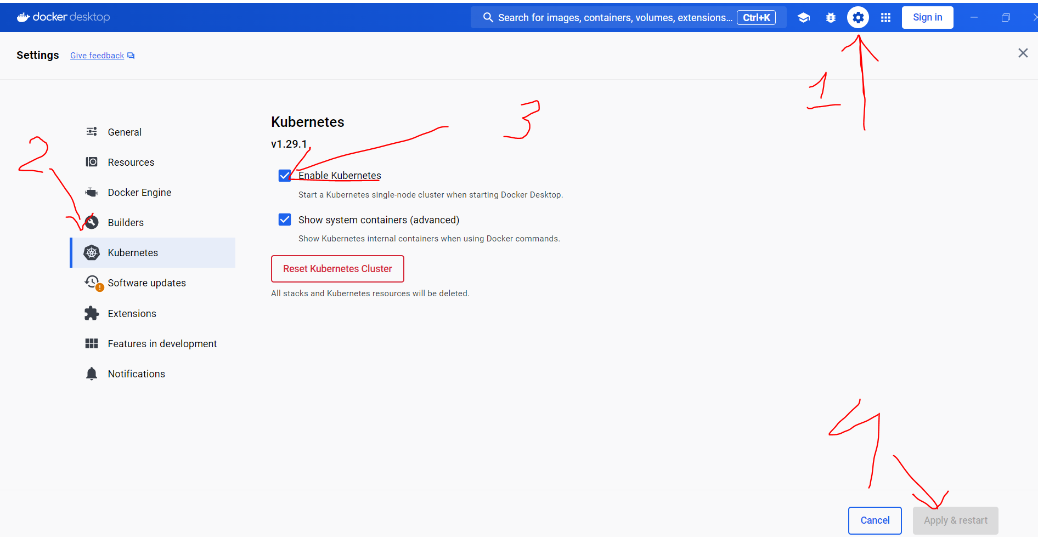


Hình 1. Giao diện phần mềm mẫu lệnh clone

1. **Triển khai cài đặt microservice với Kubernetes**

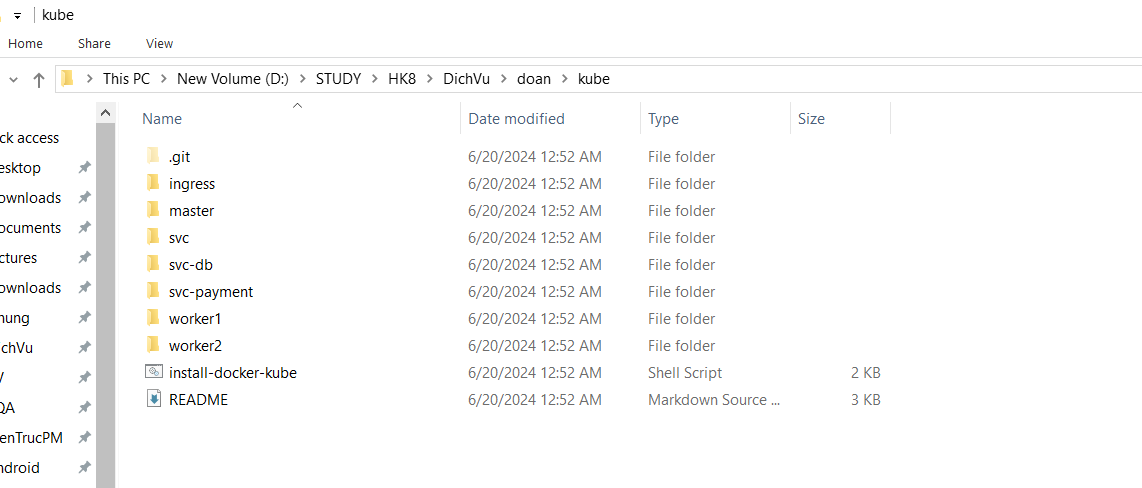
**Các bước triển khai**

**B1: Bật kubenetes trong docker**



**B2: Clone code về máy, trỏ vào thư mục kube**

Link: <https://github.com/vantohuu/kube>

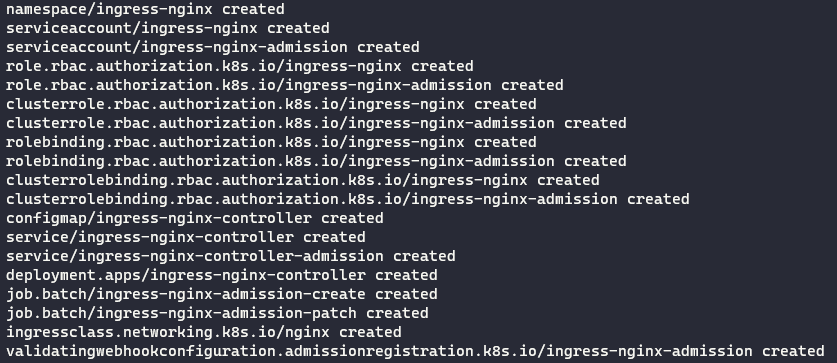


**B3: Triển khai ingress nginx controller (Reverse proxy)**

Tạo controller trong ingress-nginx

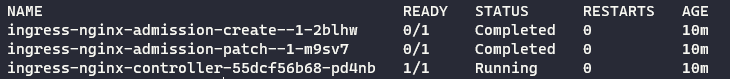
Thực hiện lệnh

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.3.0/deploy/static/provider/cloud/deploy.yaml



Kiểm tra Ingress controller pod đang chạy

kubectl get pods --namespace ingress-nginx

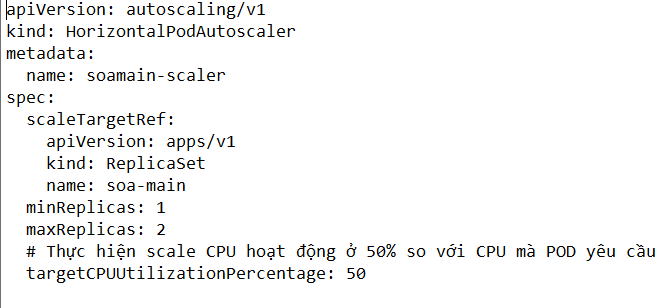
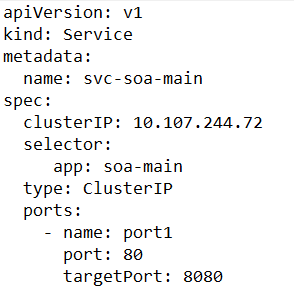
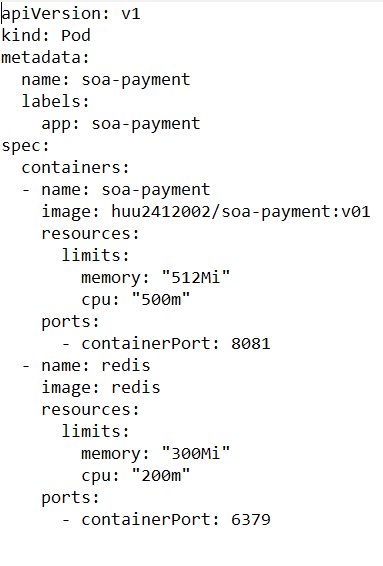
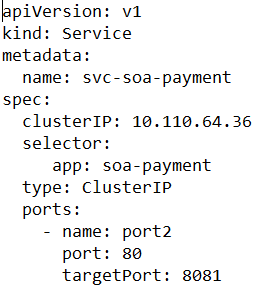
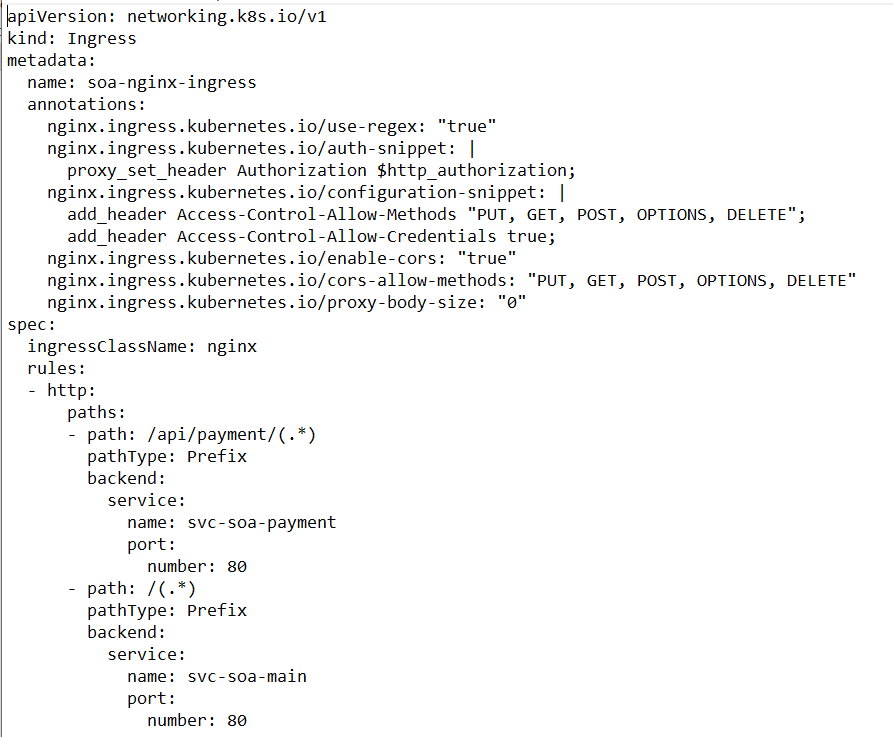


Kiểm tra Nginx Ingrss controller đã được gán public IP hay chưa

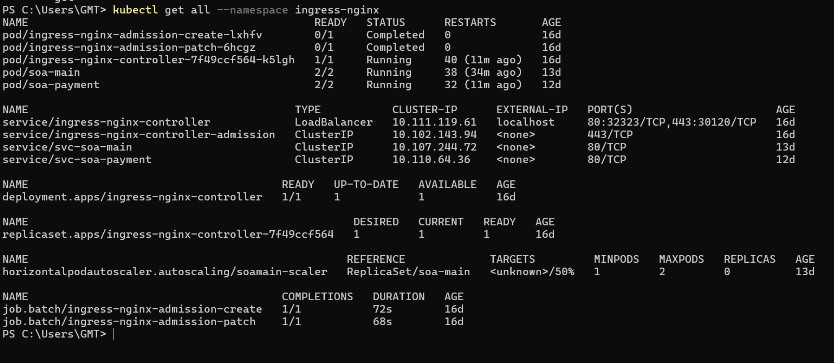
kubectl get service ingress-nginx-controller --namespace=ingress-nginx

kubernetes ingress - loadbalancer

**B4: Triển khai các service lên namspace ingress-nginx vừa triển khai xong**

* **Triển khai soa service**
  + Vào thư mục svc bằng cách: cd svc
  + Triển khai pod
    - kubectl apply -f 3.pods.yaml --namespace ingress-nginx (nếu muốn xóa cái vừa triển khai thì thay apply thành delete)
    - 
  + Triển khai autoscale cho pod
    - kubectl apply -f 3.pods.soa.hpa.yaml --namespace ingress-nginx (nếu muốn xóa cái vừa triển khai thì thay apply thành delete)
    - 
  + Triển khai service
    - kubectl apply -f soamain.svc.yaml --namespace ingress-nginx (nếu muốn xóa cái vừa triển khai thì thay apply thành delete)
    - 
* **Triển khai payment service**
  + Vào thư mục svc bằng cách: cd ../svc-payment
  + Triển khai pod
    - kubectl apply -f 3.pods.yaml --namespace ingress-nginx (nếu muốn xóa cái vừa triển khai thì thay apply thành delete)
    - 
  + Triển khai service
    - kubectl apply -f soapayment.svc.yaml --namespace ingress-nginx (nếu muốn xóa cái vừa triển khai thì thay apply thành delete)
    - 
* **Đưa các services lên ingress**
  + Vao thư mục svc bằng cách: cd ../ingress
    - kubectl apply -f nginx.ingress.yaml --namespace ingress-nginx (nếu muốn xóa cái vừa triển khai thì thay apply thành delete)
    - 

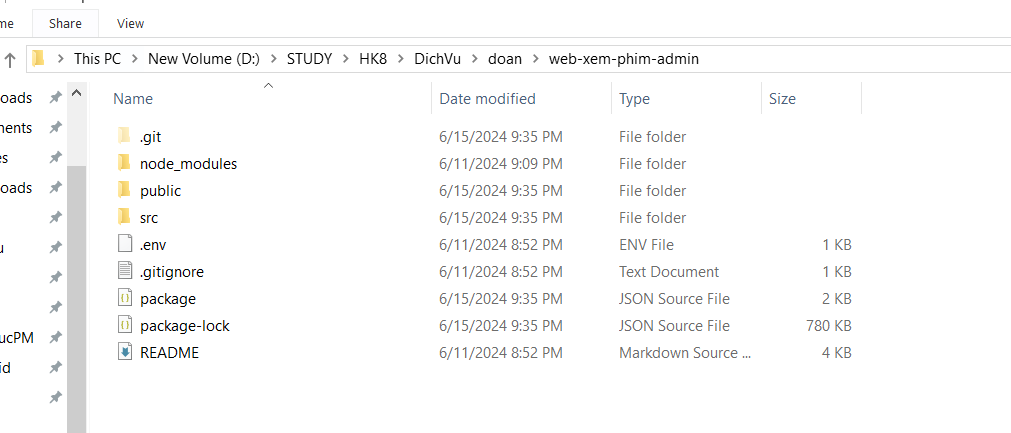
**Kiểm tra những cái vừa triển khai: kubectl gel all --namespace ingress-nginx**



1. **Triển khai cài đặt CMS (Reactjs)**

**Clone code về máy, trỏ vào thư mục web-xem-phim-admin**

Link: : https://github.com/lucvan02/web-xem-phim-admin

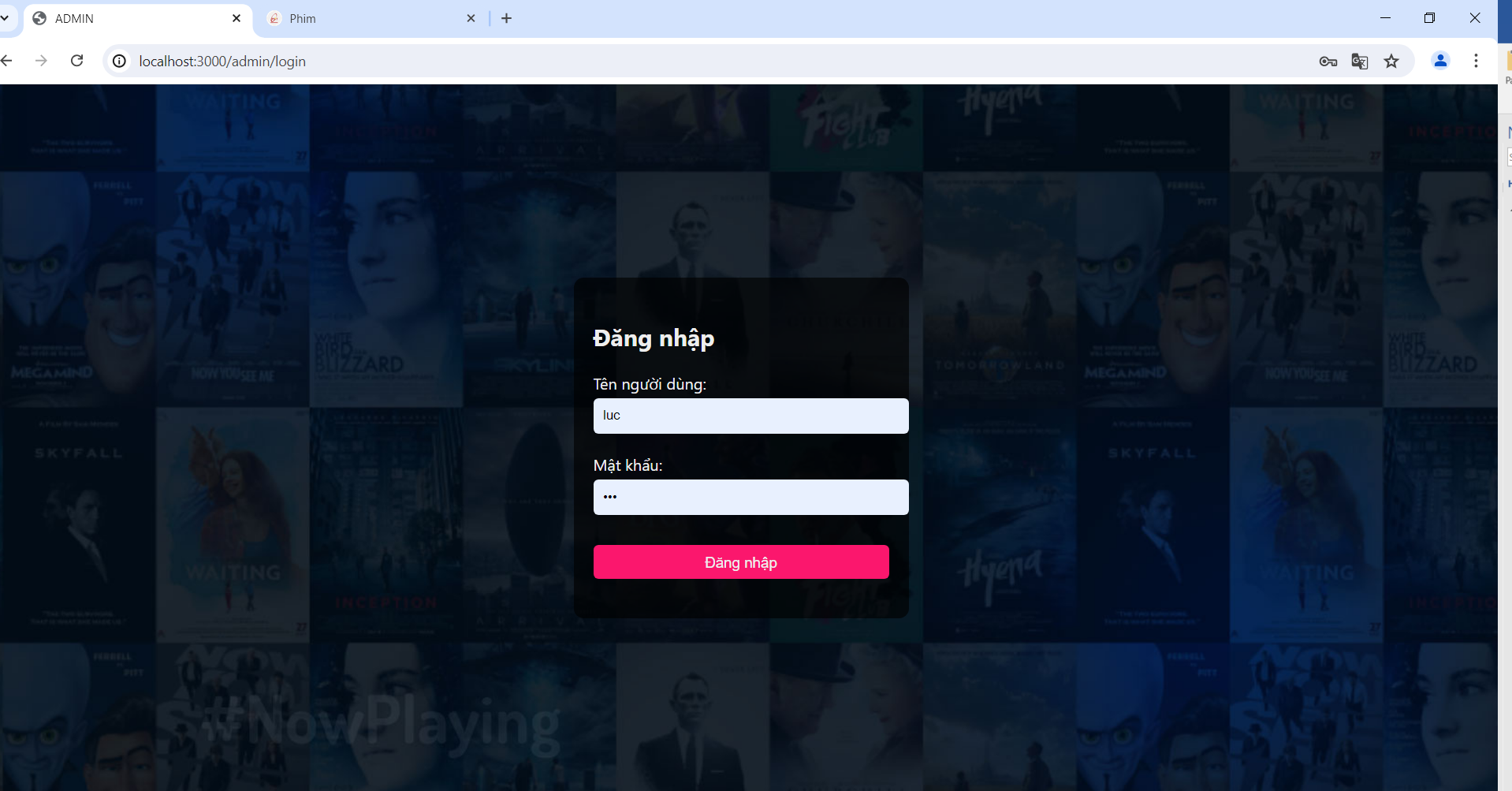


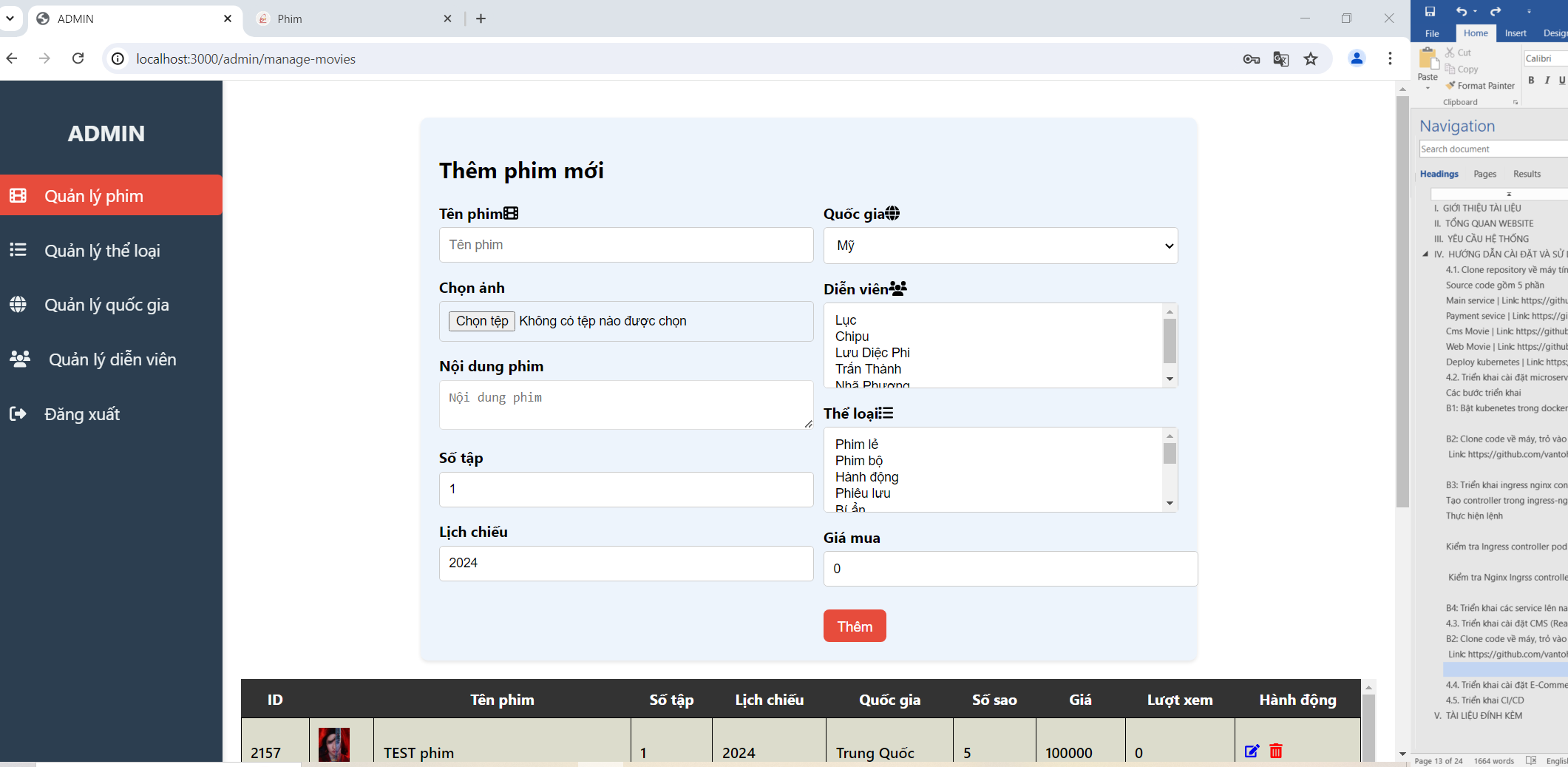
* Khởi động Visual code chọn project

+ Thực hiện lệnh npm install để cài đặt các thư viện

+ Thực hiện lệnh npm start chạy chương trình

+ Đăng nhập test: username: luc, pass: 123

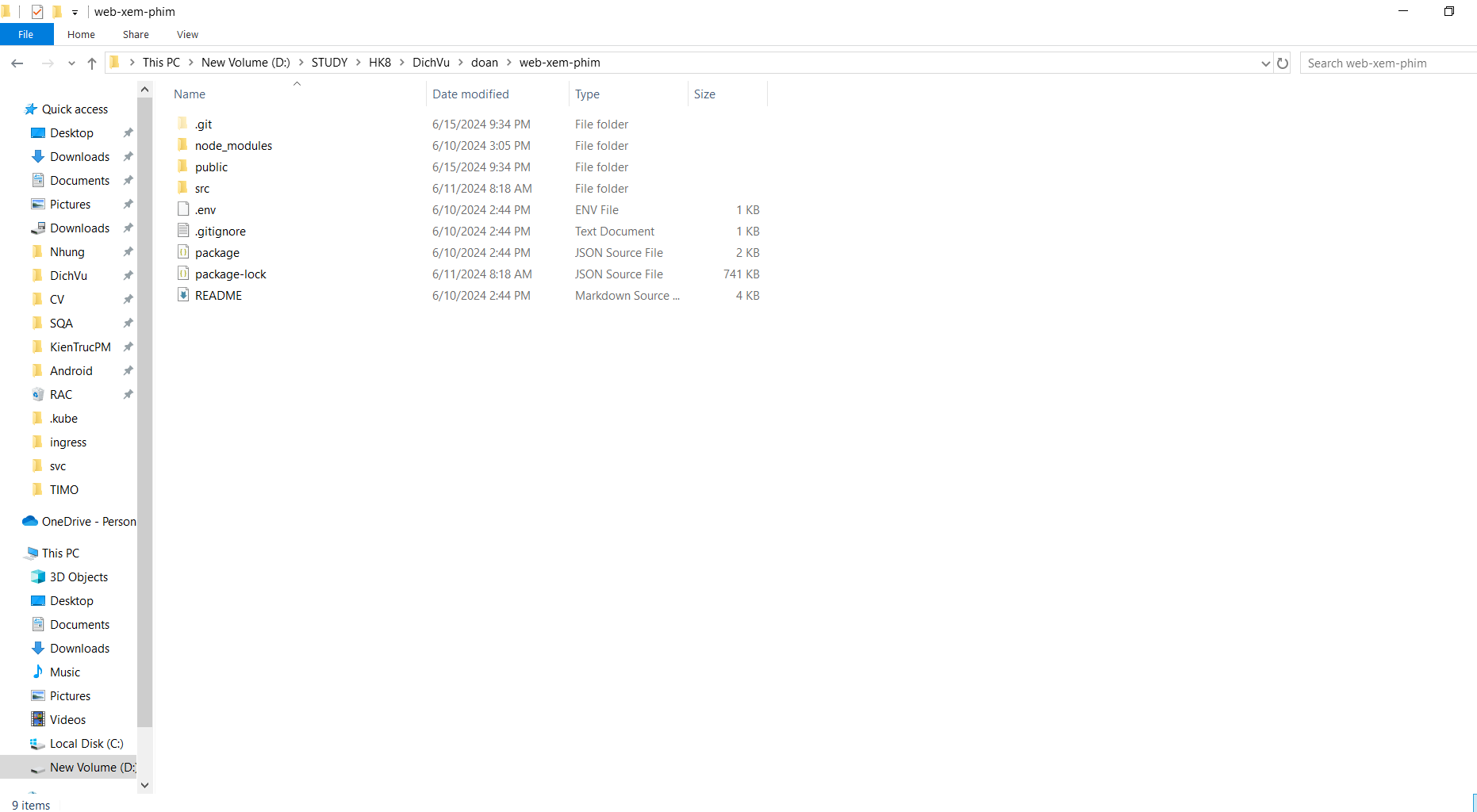




1. **Triển khai cài đặt giao diện web movie (Reactjs)**

**Clone code về máy, trỏ vào thư mục web-xem-phim**

Link: : https://github.com/lucvan02/web-xem-phim

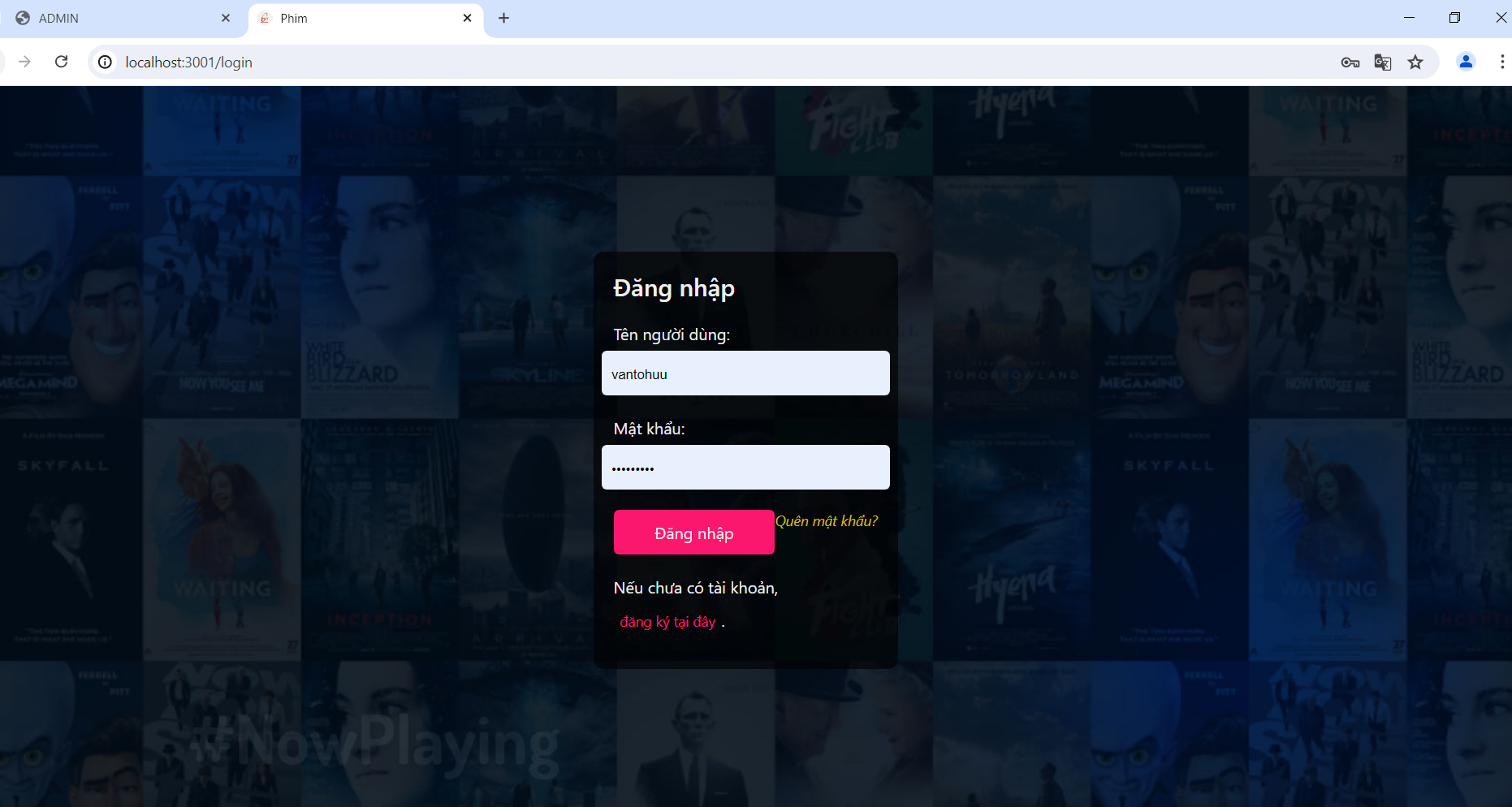


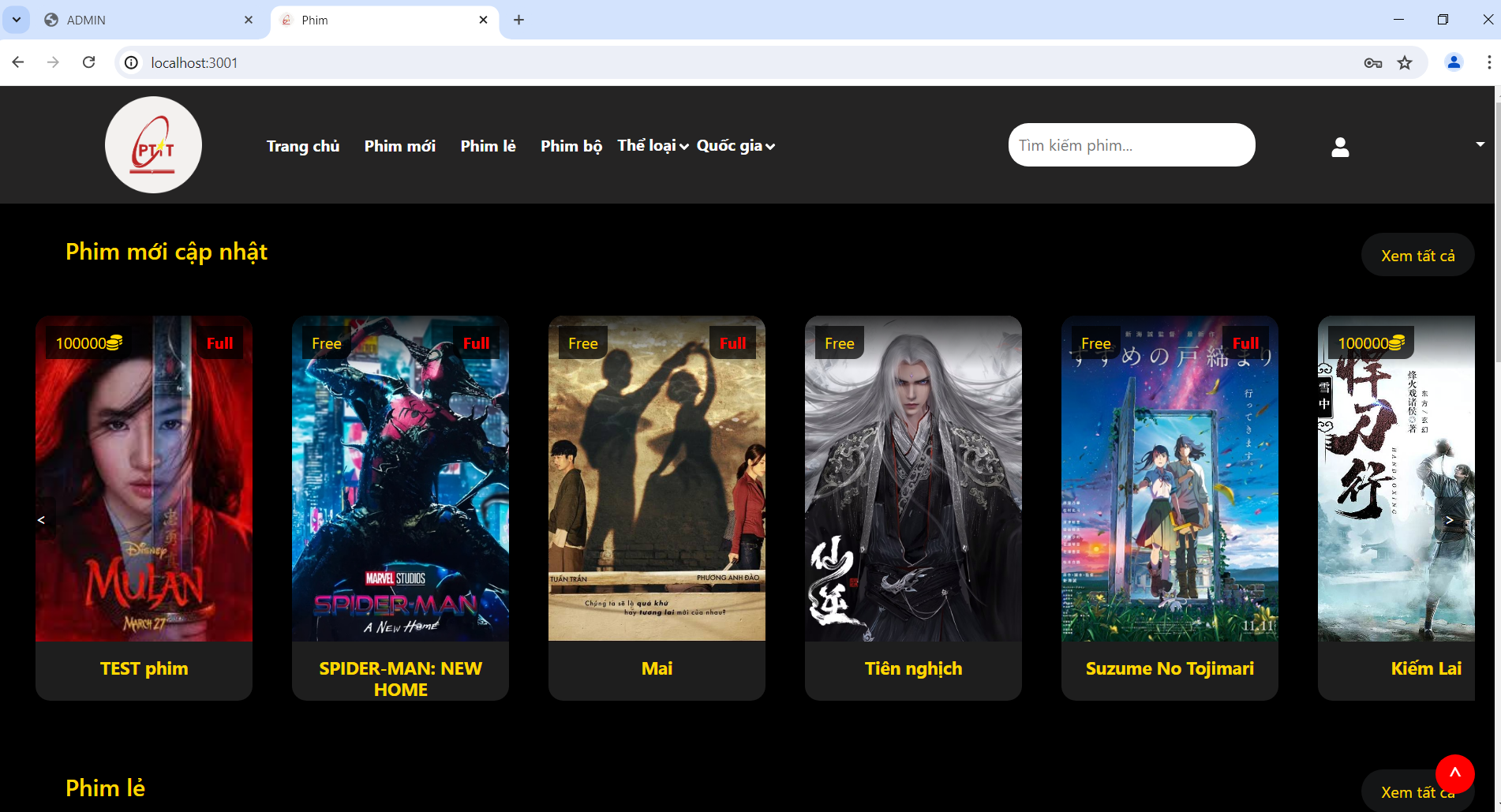
* Khởi động Visual code chọn project

+ Thực hiện lệnh npm install để cài đặt các thư viện

+ Thực hiện lệnh npm start chạy chương trình

+ Đăng nhập test: username: vantohuu, pass: 123456789





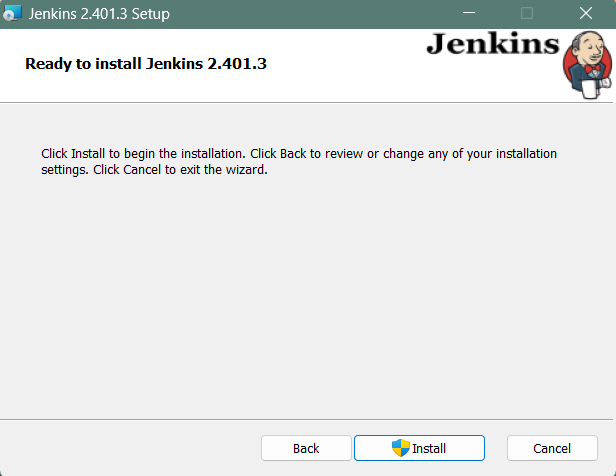
1. **Triển khai CI/CD**

**Cài đặt Jenkins (Windows)**

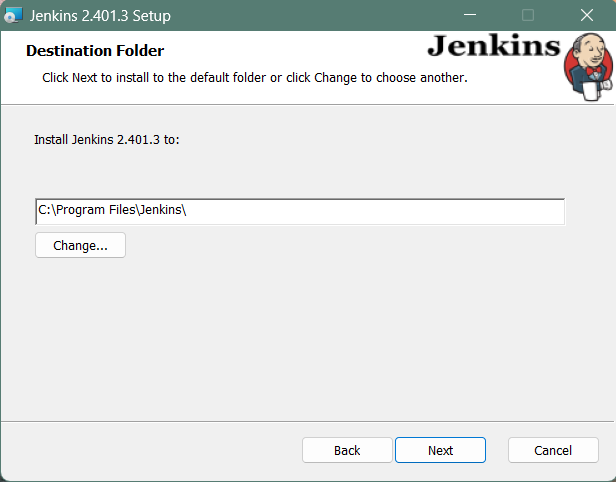
[**https://www.jenkins.io/doc/book/installing/windows**](https://www.jenkins.io/doc/book/installing/windows)

Tiến hành download và cài đặt theo hướng dẫn

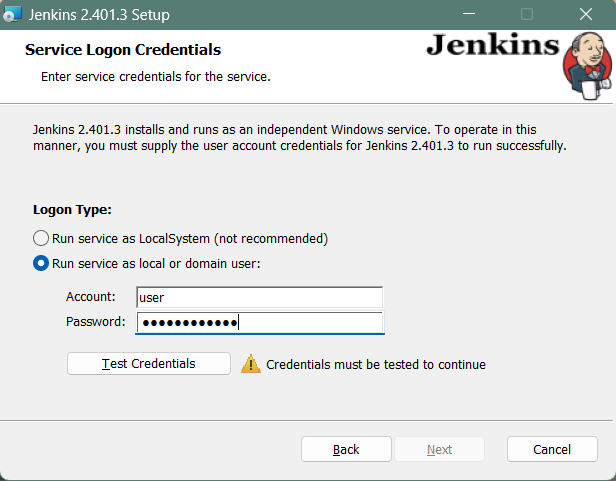
B1: Setup Wizard



B2: Chọn đường dẫn cài đặt



B3: Setup Service logon credentials

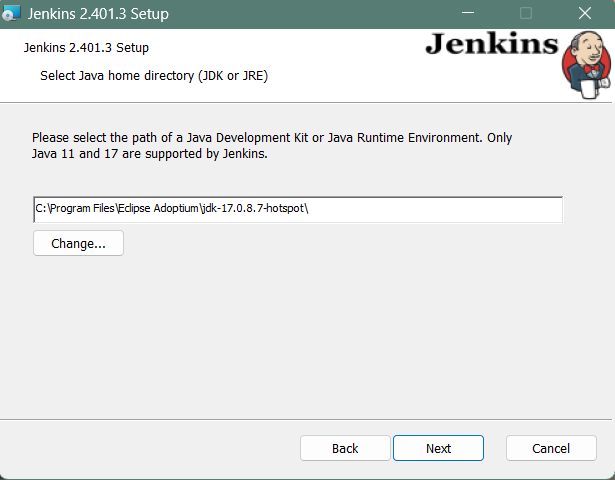


B4: Chọn Port cho Jenkins

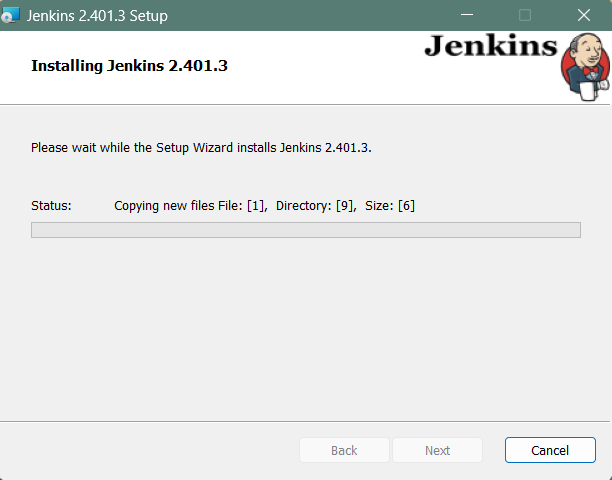


Ở đây mình sử dụng port là 8082

B6: Chọn đường dẫn Java home



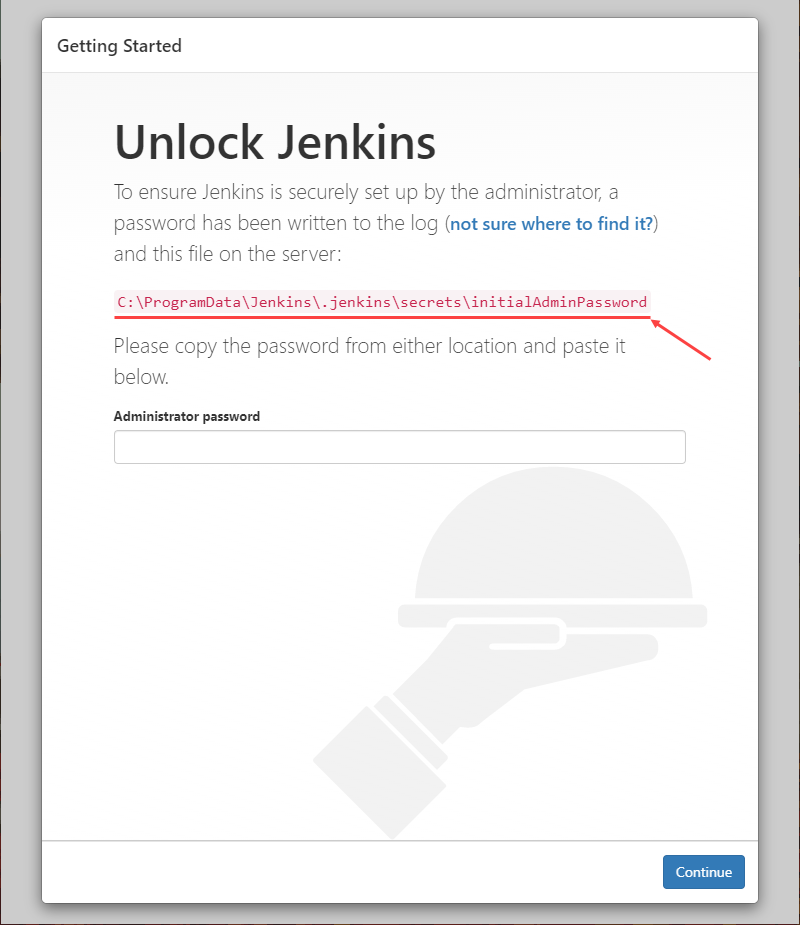
B6: Install Jenkins



**Khởi chạy Jenkins**

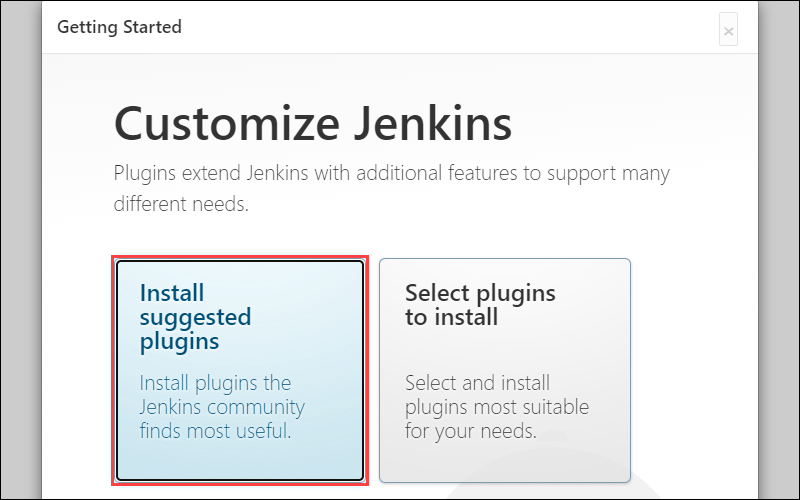
Truy cập <http://localhost:8082> để chạy Jenkins sau khi cài đặt xong

Ở lần chạy đầu tiên ta cần tiến hành unlock Jenkins

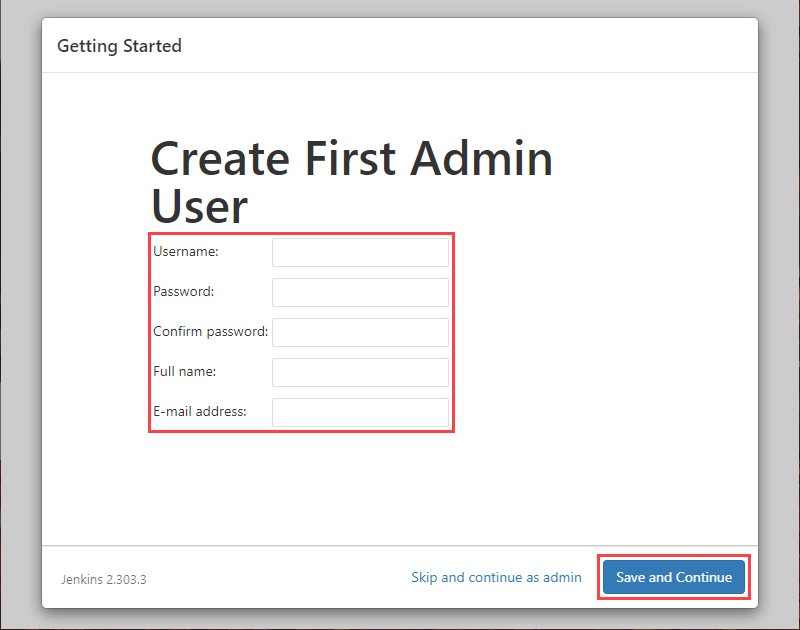


Vào đường dẫn như đường đẫn trong máy của bạn như trong ảnh, để tiến hành lấy password, sau đó dán vào ô Administrator password và nhấn Continute để thực hiên bước kế tiếp.

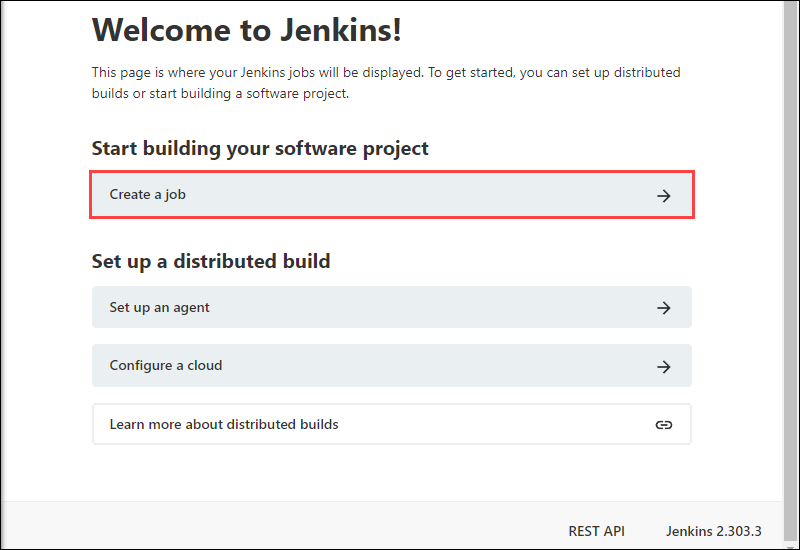
Chọn Install Suggested plugins hoặc select plugins to install



Khởi tạo username và password

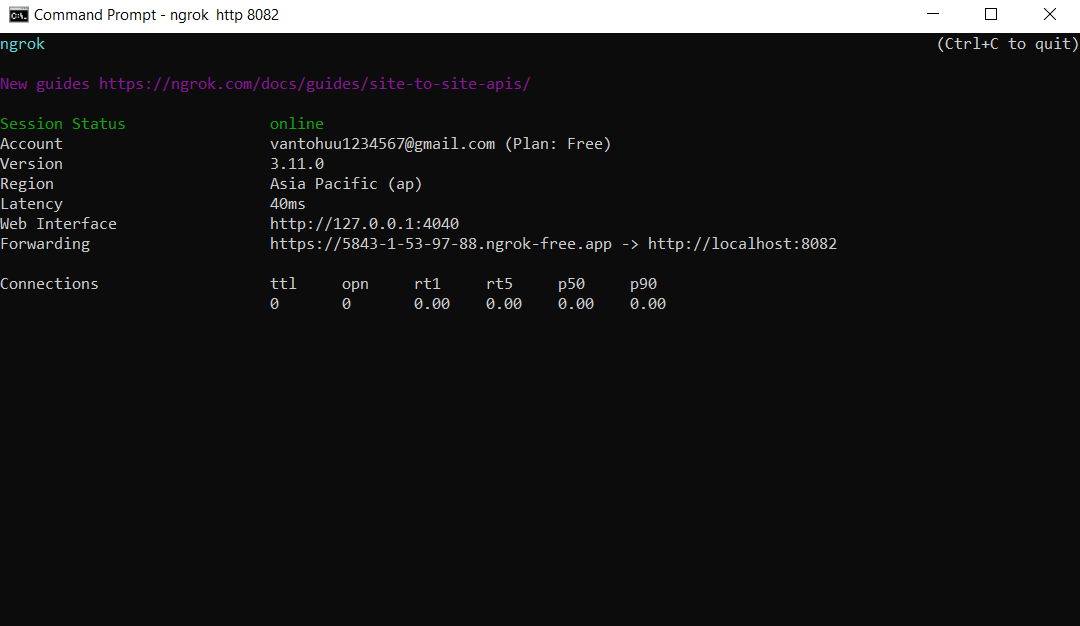


Khỏi tạo thành công



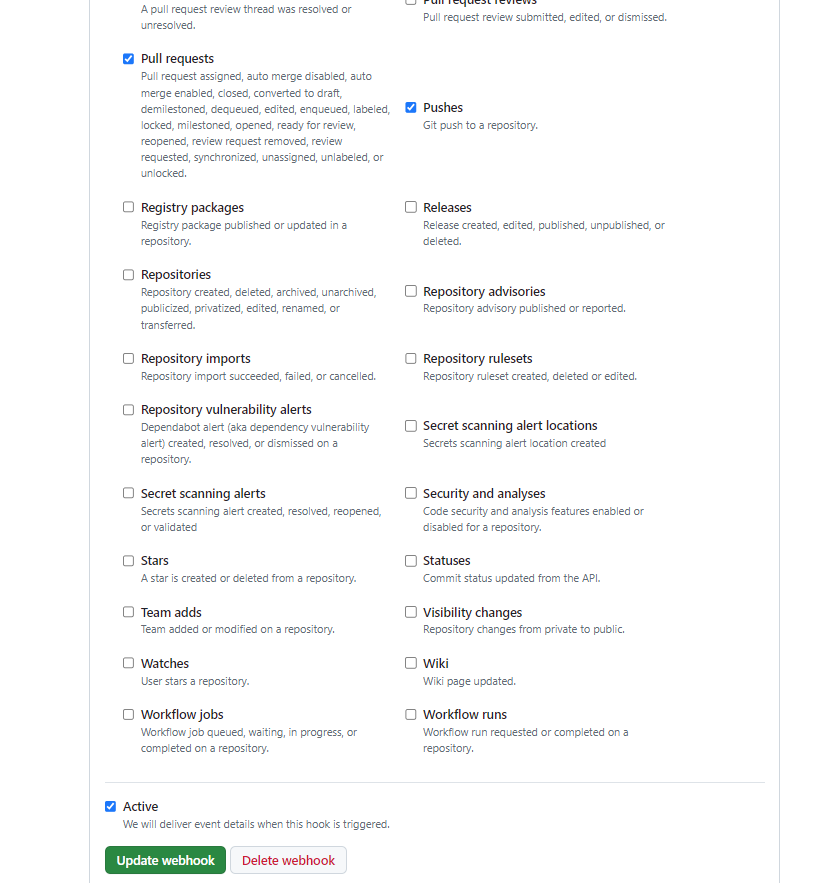
**Tiến hành cấu hình github webhook**

Đưa localhost:8082 ra bên ngoài internet bằng ngrox

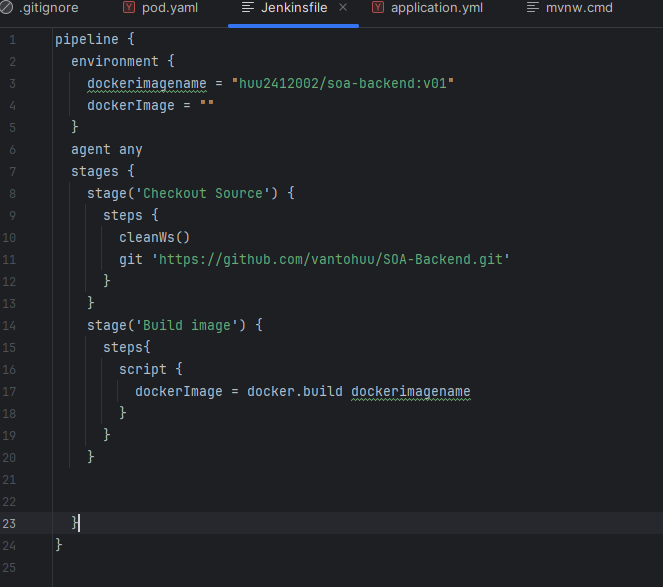
Cài đặt ngrok, sau đó thự hiện authencation, sau đó thực hiên lệnh ngrok http 8082

Vào webhook github tạo một webhook

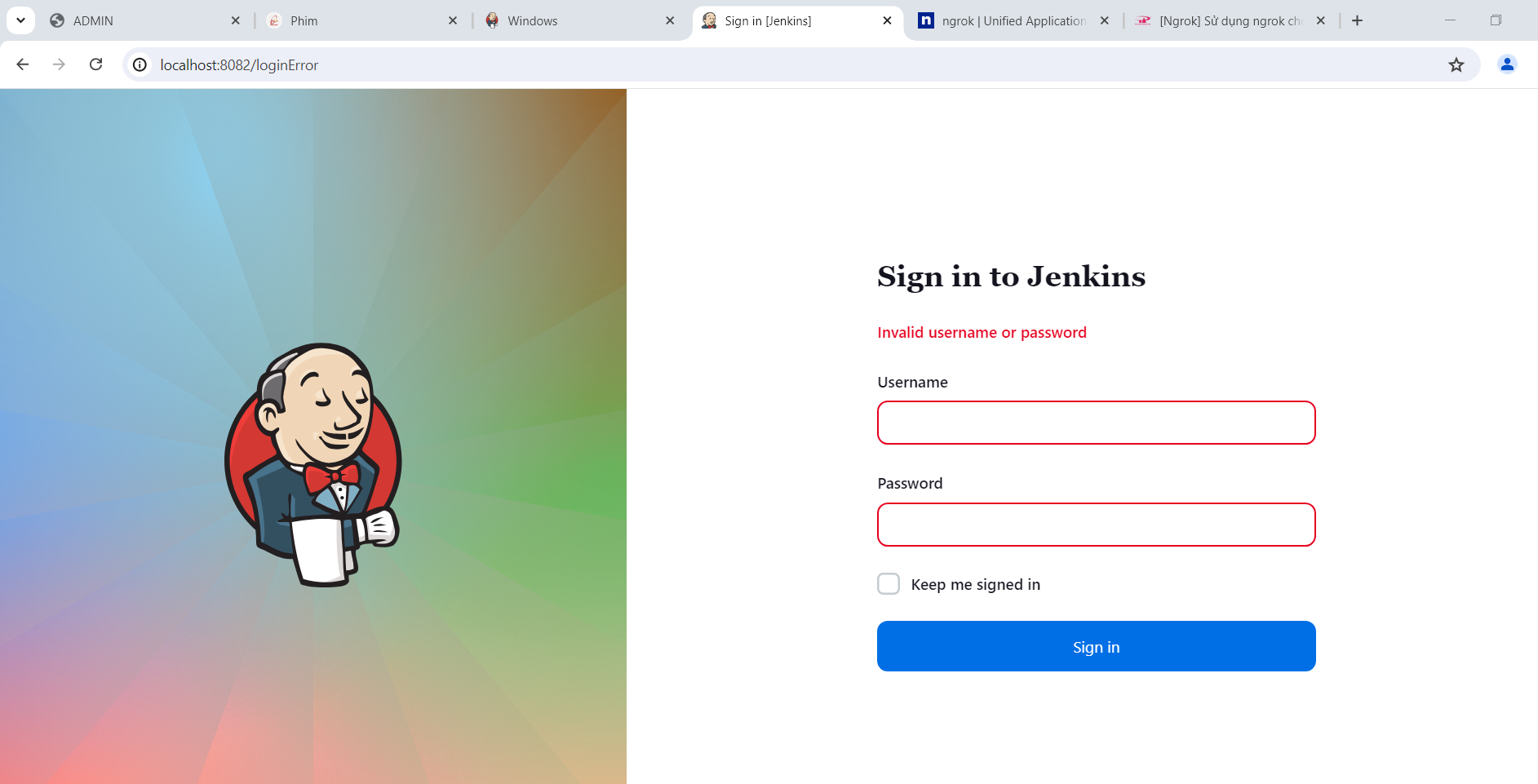




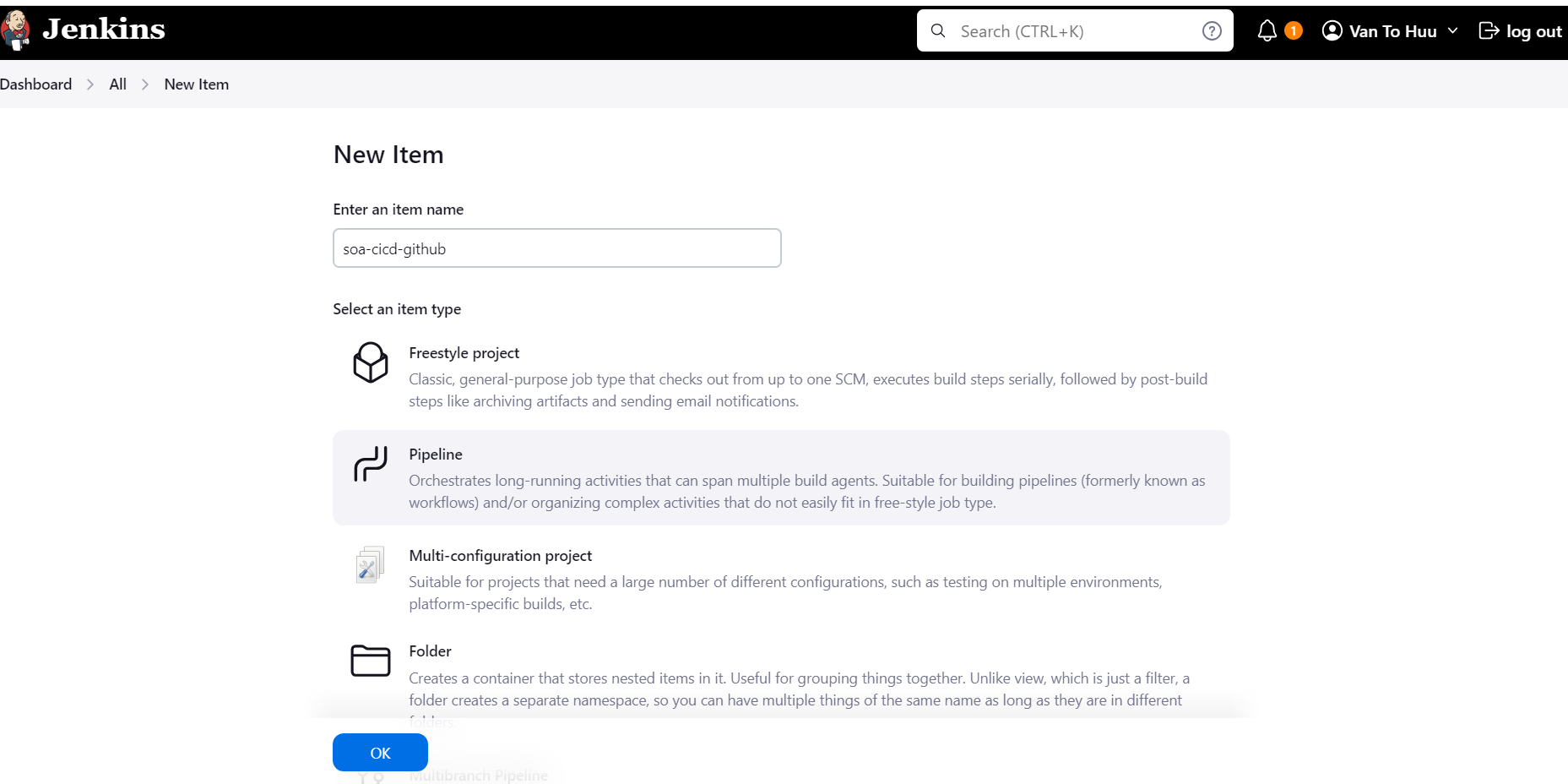
Tạo pipeline trên Jenkins



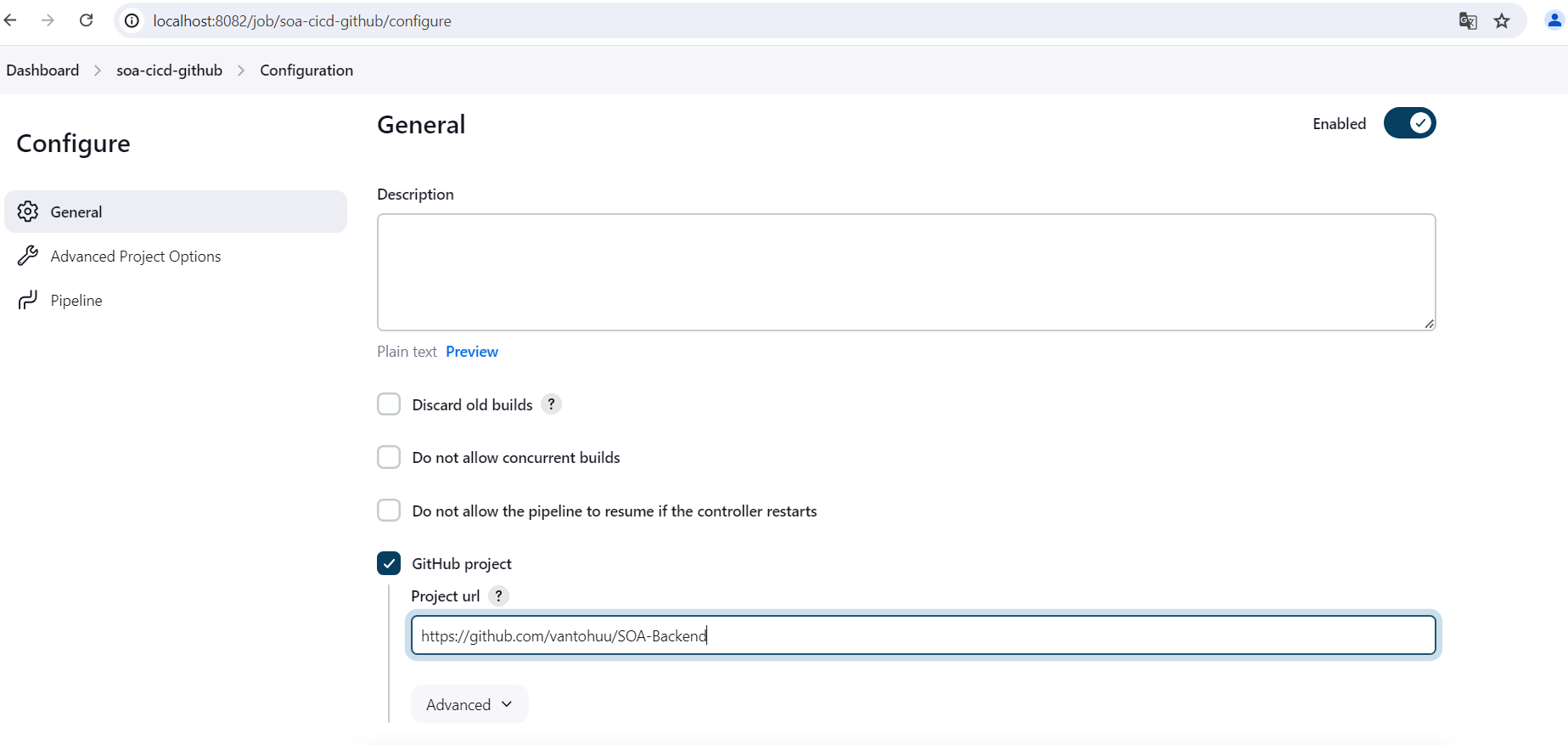
Đăng nhập



Chọn pipeline và gõ tên



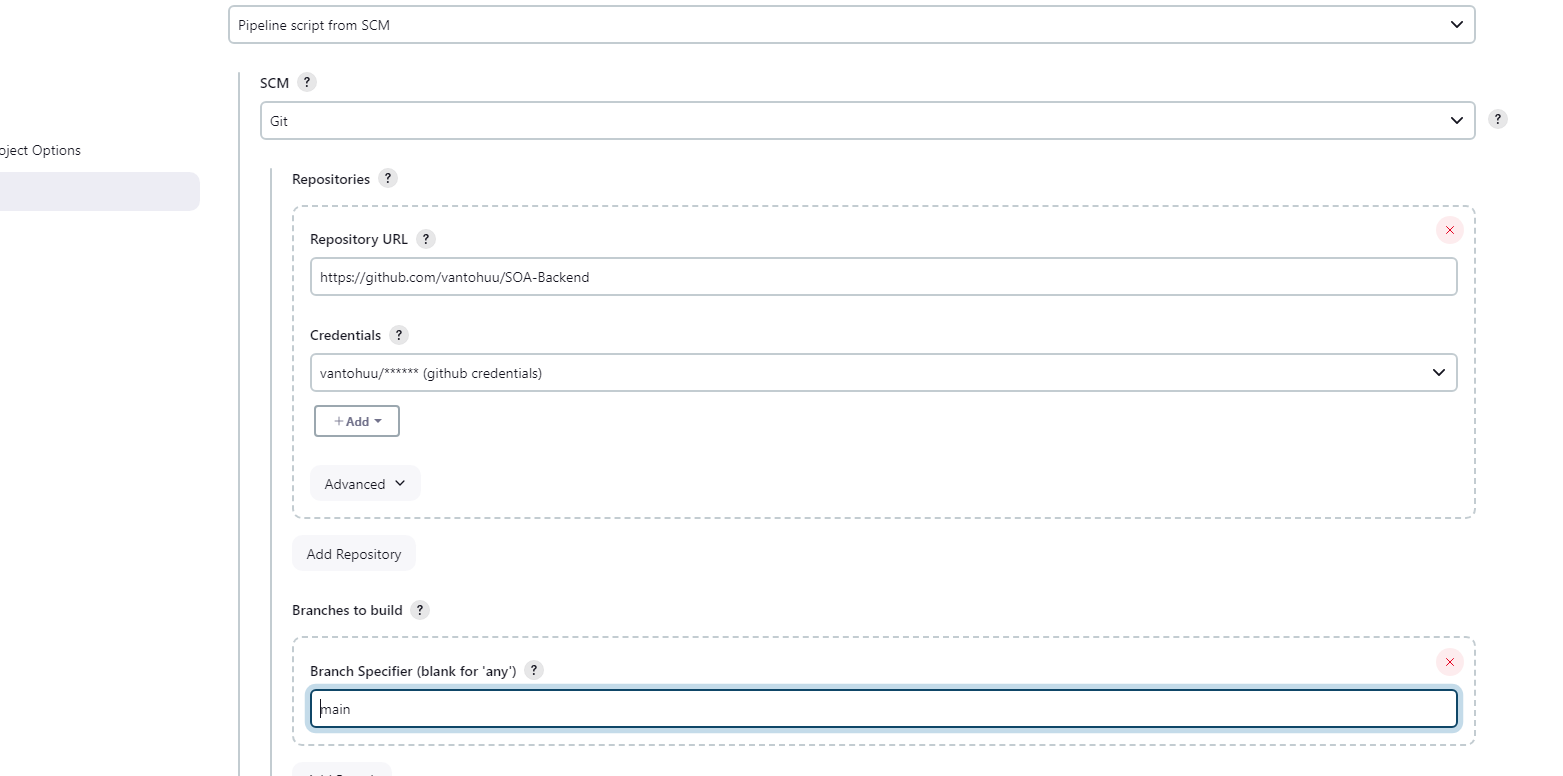
(tùy chọn) Chọn mô tả là github project và dán đường dẫn kink git hub vào

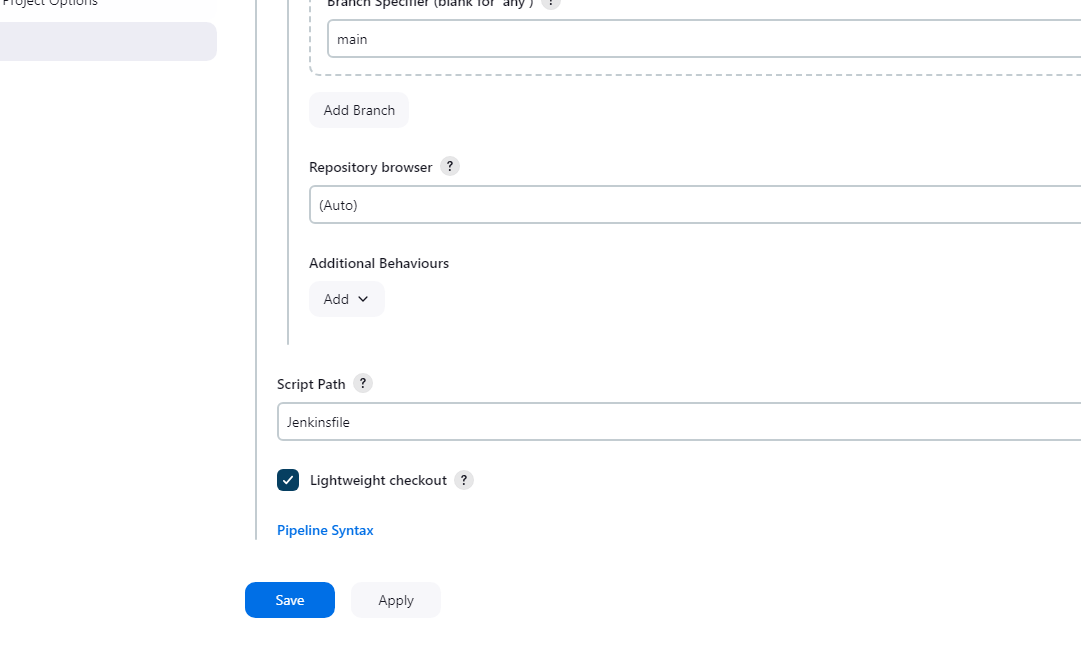


Tick chọn Git hook trigger for GIT Scm polling để nhận thông tin từ github webhook



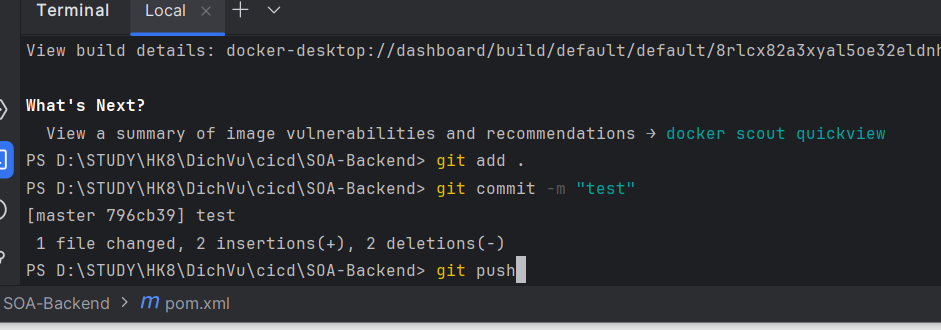
Tùy chọn pipeline trong SCM để chạy code jenkinsfile từ github và cấu hình các thông tin cho chúng



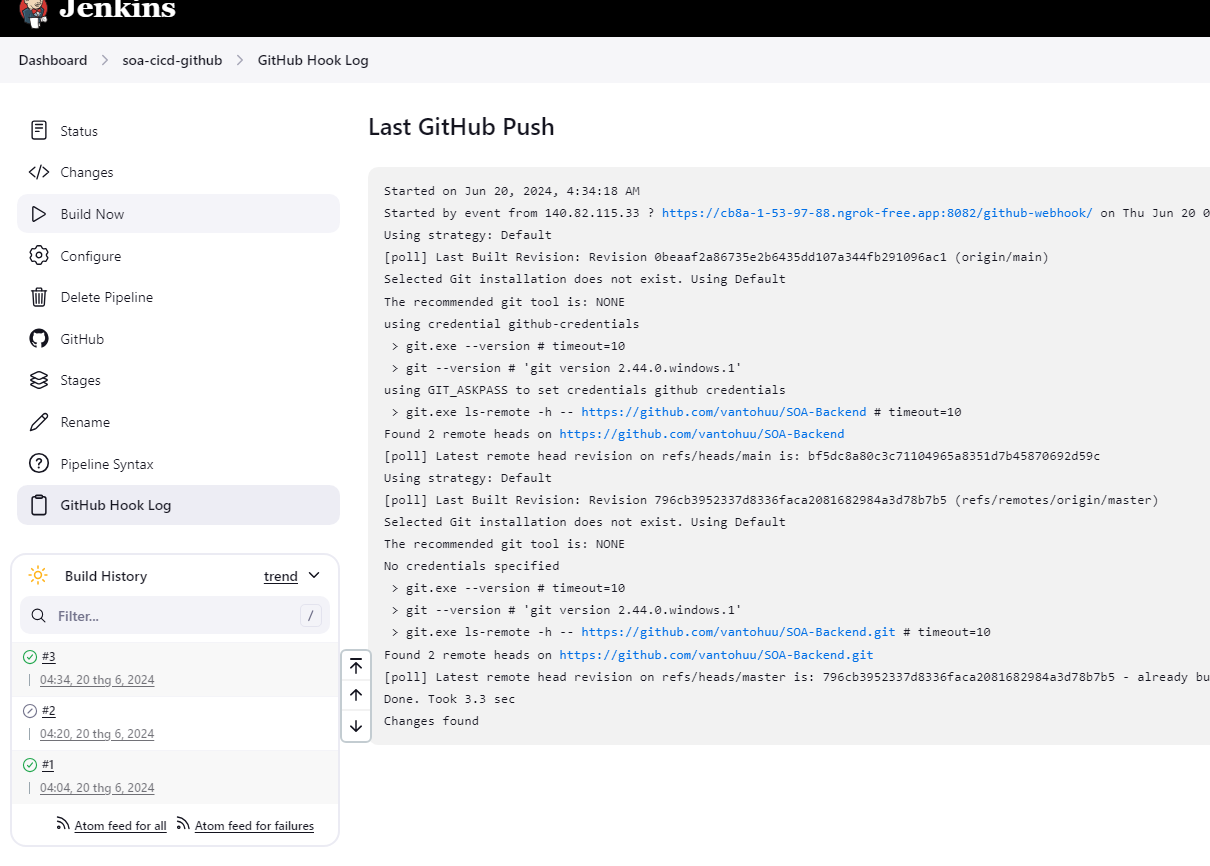
Nhấn save để lưu

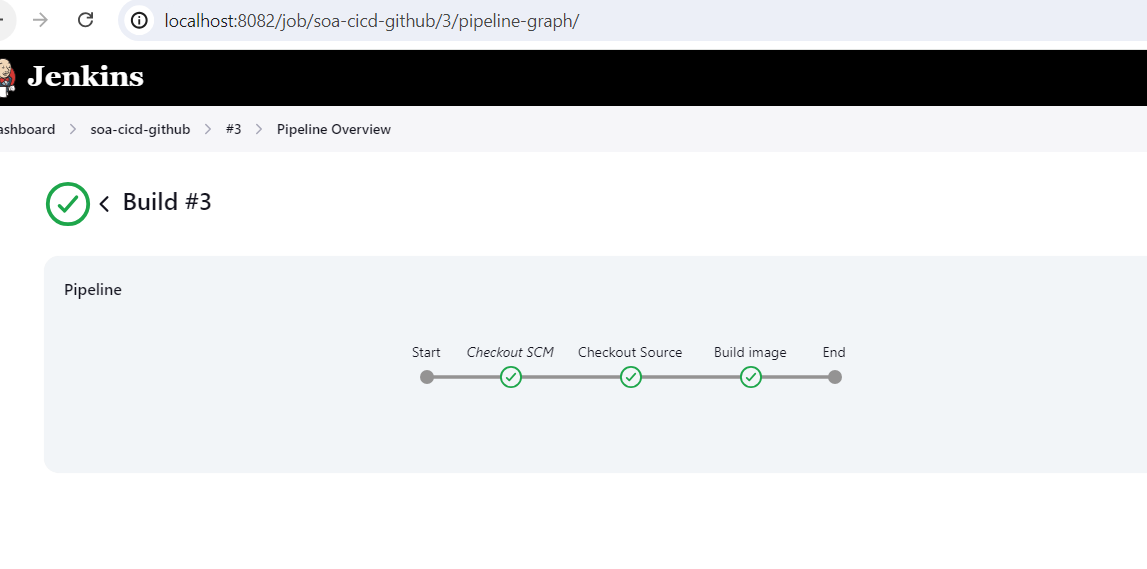
**Thực hiện cicd**

Tiến hành push code lên git



Jenkins nhận thay đổi và tự động build





# TÀI LIỆU THAM KHẢO

<https://www.jenkins.io/>

<https://ngrok.com/>

<https://www.red-gate.com/simple-talk/devops/containers-and-virtualization/deploying-a-dockerized-application-to-the-kubernetes-cluster-using-jenkins/>