Table of Contents

[1 Cài đặt docker: 2](#__RefHeading___Toc341_1384329262)

[2 Cài đặt Java: 2](#__RefHeading___Toc343_1384329262)

[3 Cài đặt Nodejs, npm, yarn 3](#__RefHeading___Toc345_1384329262)

[4 Cài đặt .Net 3](#__RefHeading___Toc347_1384329262)

[5 Cài đặt Jenkin: 3](#__RefHeading___Toc349_1384329262)

[6 Backend: 8](#__RefHeading___Toc351_1384329262)

[7 Frontend: 12](#__RefHeading___Toc353_1384329262)

## Cài đặt docker:

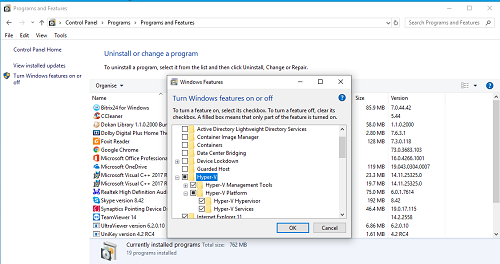
Yêu cầu: hyper V/ WSL2

Link chứa file cài đặt:

https://www.docker.com/

Bật hyper V

Control Panel → Programs → Turn Windows features on or off

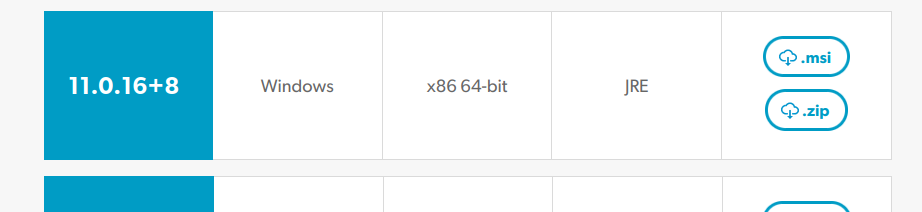


Tải và chạy file.

## Cài đặt Java:

Link chứa file cài đặt:

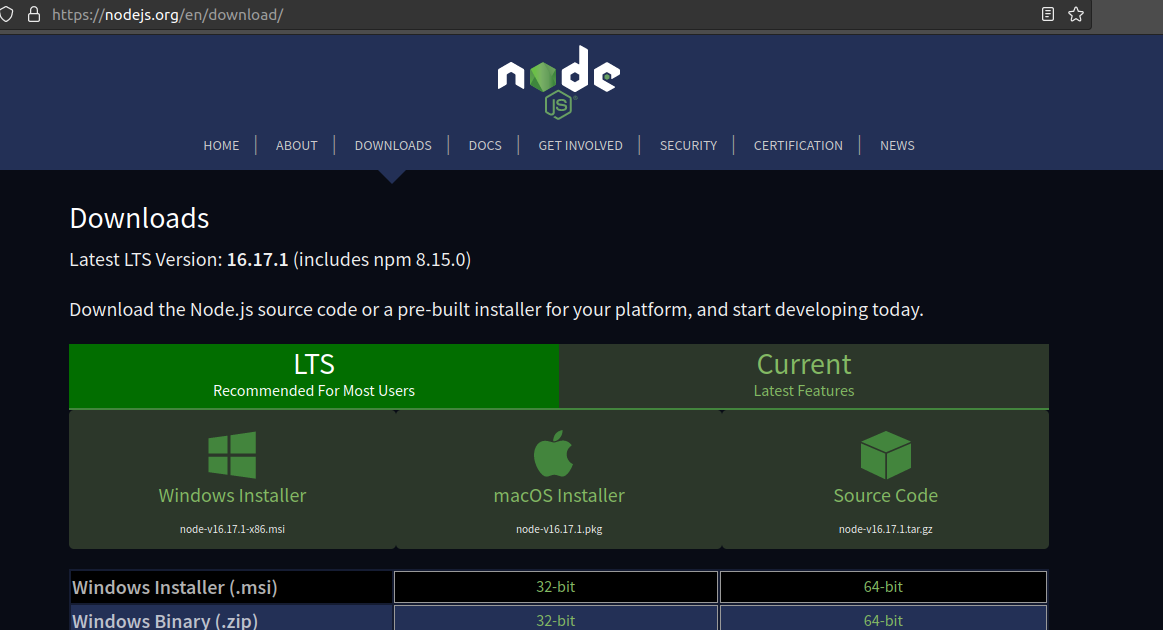
<https://www.openlogic.com/openjdk-downloads>

®

Tải và chạy file .msi.

## Cài đặt Nodejs, npm, yarn

Link chứa file cài đặt: https://nodejs.org/en/download/

Tải và chạy file .msi.

Mở terminal run command ‘npm install --global yarn’

## Cài đặt .Net

Link chứa file cài đặt

<http://www.microsoft.com/net/download>

Tải file và cài đặt file .msi.

## Cài đặt Jenkin:

Yêu cầu: java 11

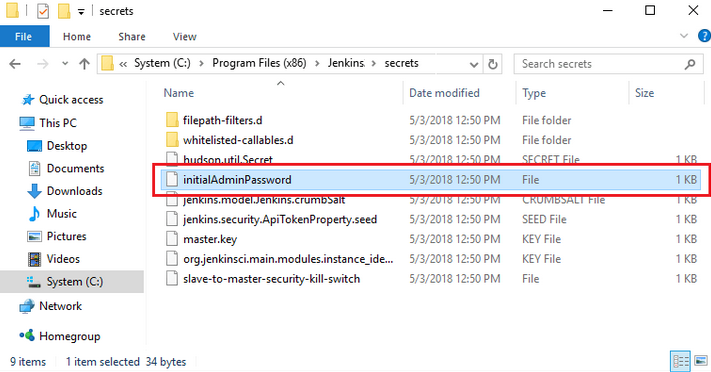
Link chứa file cài đặt:

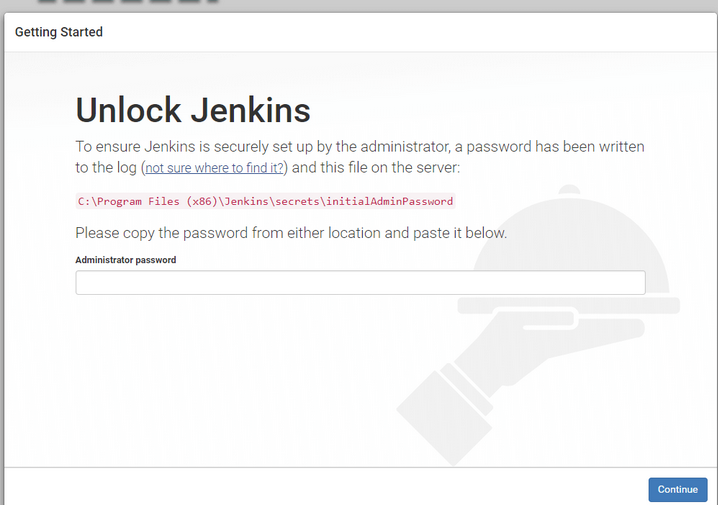
<https://www.jenkins.io/download/>

Tải và chạy file.

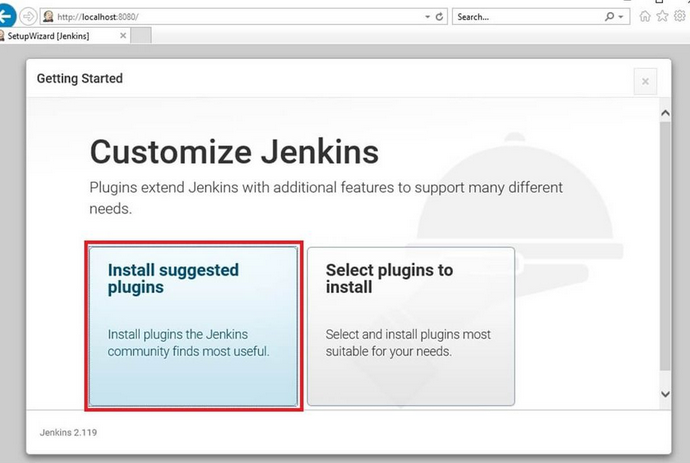
Sau khi cài đặt thành công:

Lấy init password: C:\Program Files (x86)\Jenkins\initialAdminPassword

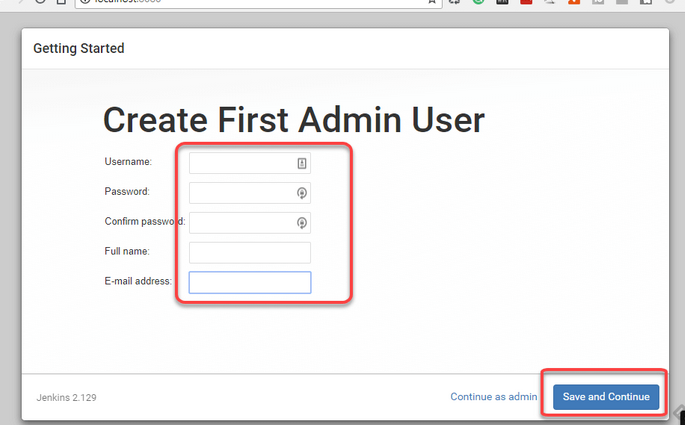
Mở trình duyệt nhập URL: localhost:8080

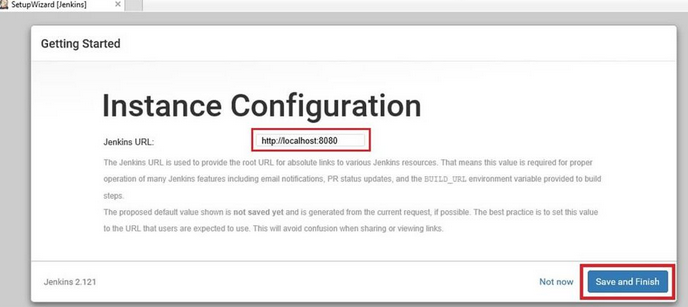
Nhập password vừa lấy được.

Chọn install suggested plugin:

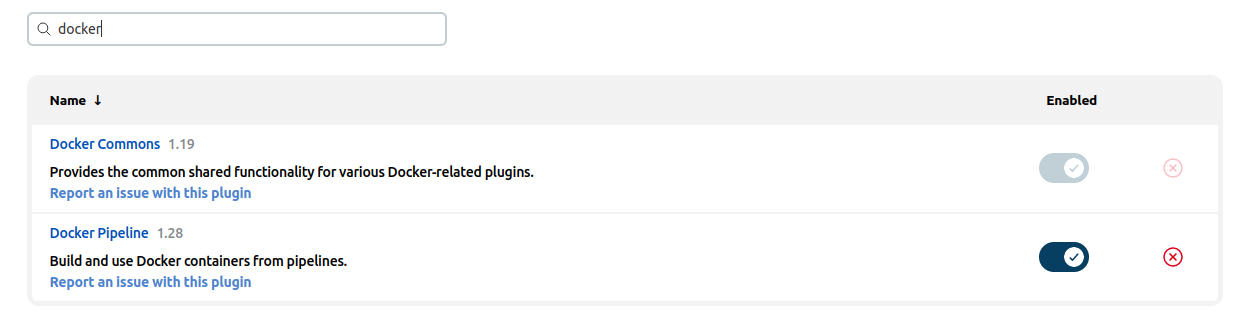
Sau khi cài đặt hoàn tất chon Save and finish

Tạo admin user:

Chọn save and continue → save and finish

Truy cập URL: <http://localhost:8080/pluginManager/available>

Tìm kiếm: Docker Pipeline và cài đặt “install without restart”

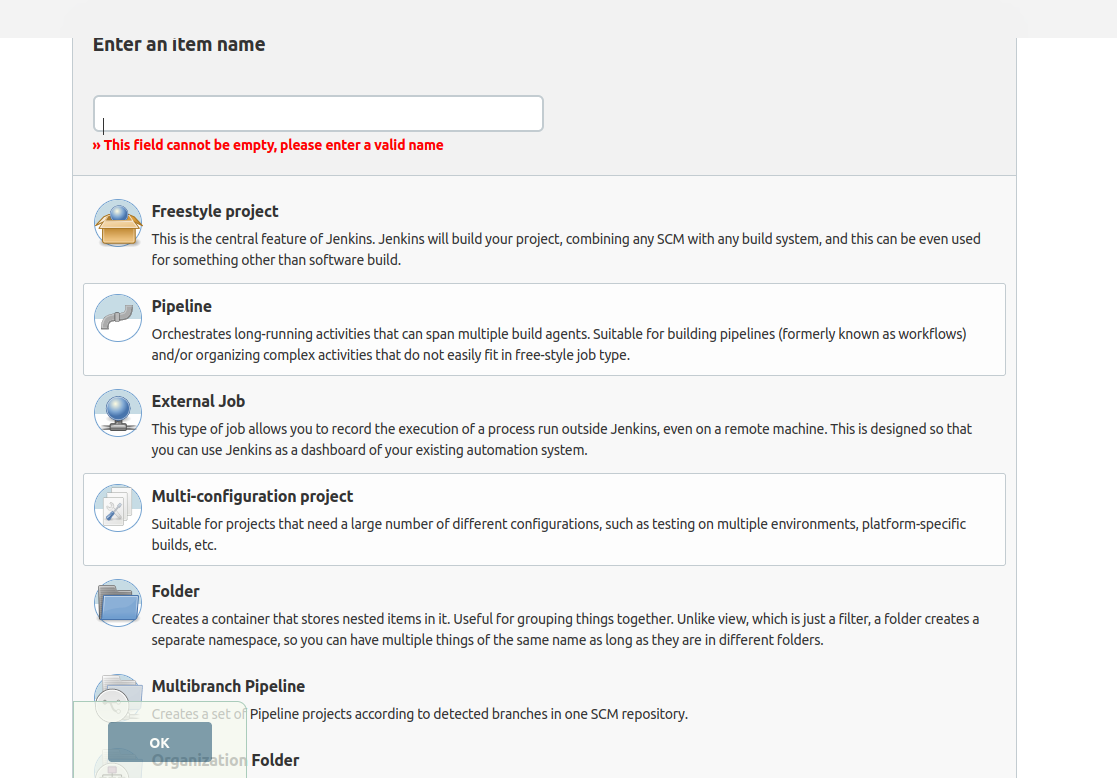
Cài đặt Msbuild

Truy cập URL: <http://localhost:8080/pluginManager/available>

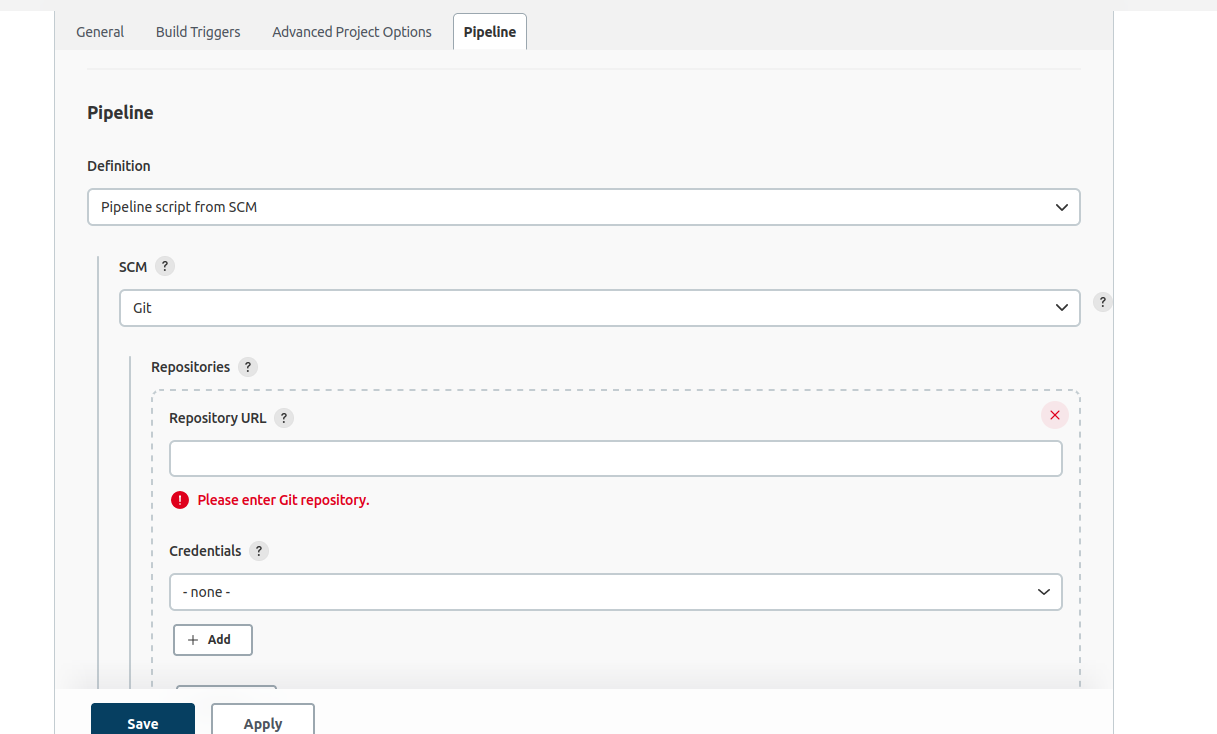
## Backend:

Truy cập URL: <http://localhost:8080/view/all/newJob>

Nhập tên project chọn Pipeline

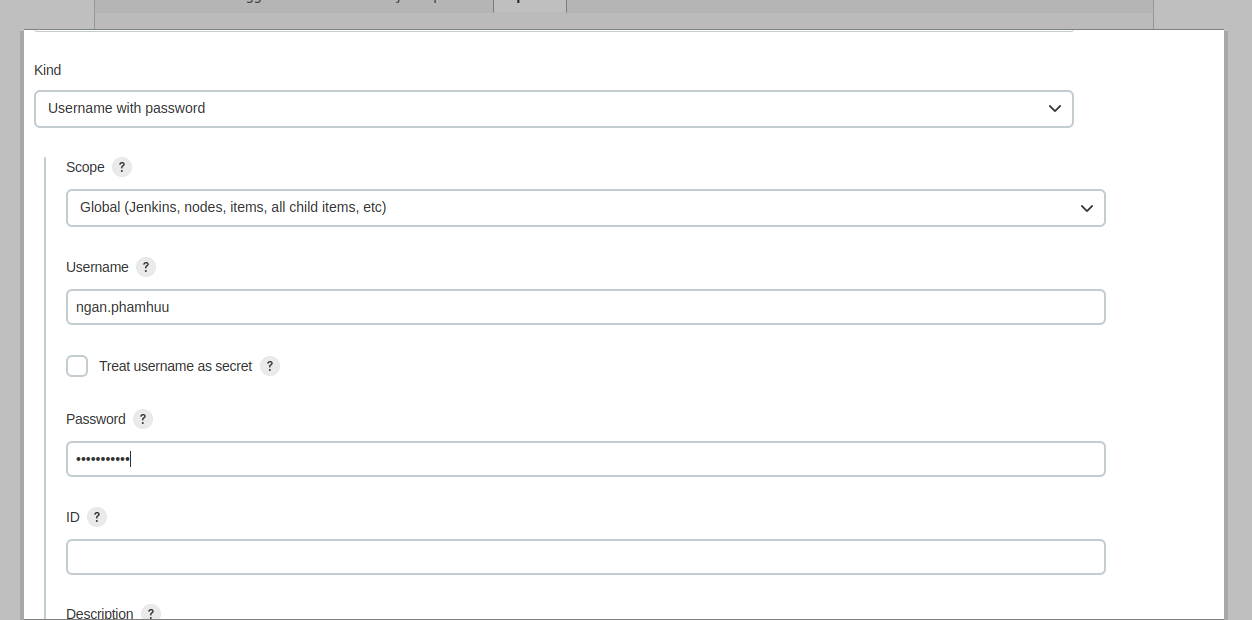


Chọn pipeline

  
Nhập username password truy cập git credential → add

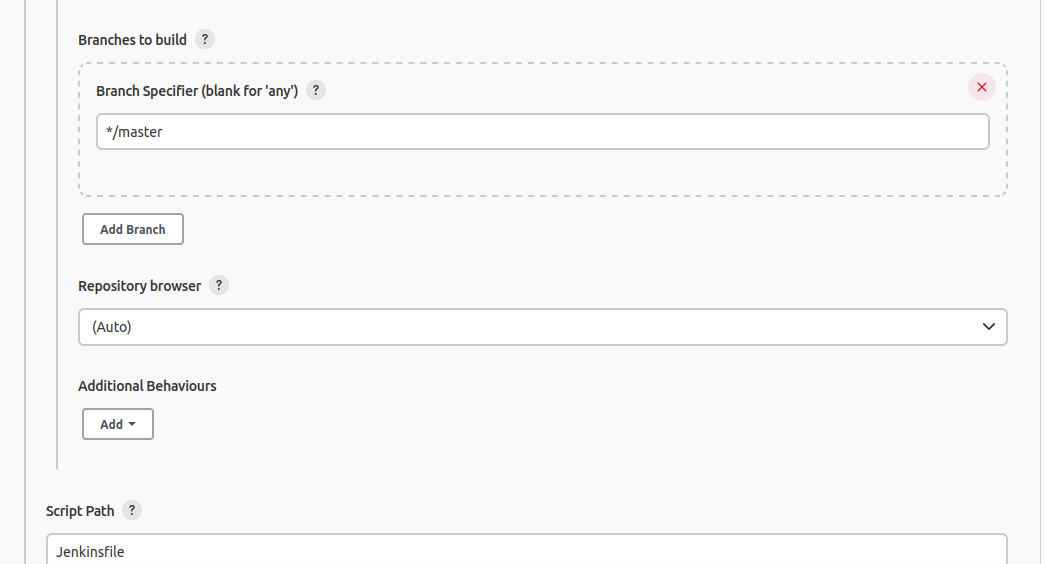
Repository: https://gitlab.com/thangph9/MBF-GQKN-BE.git

Nhập xong chọn add



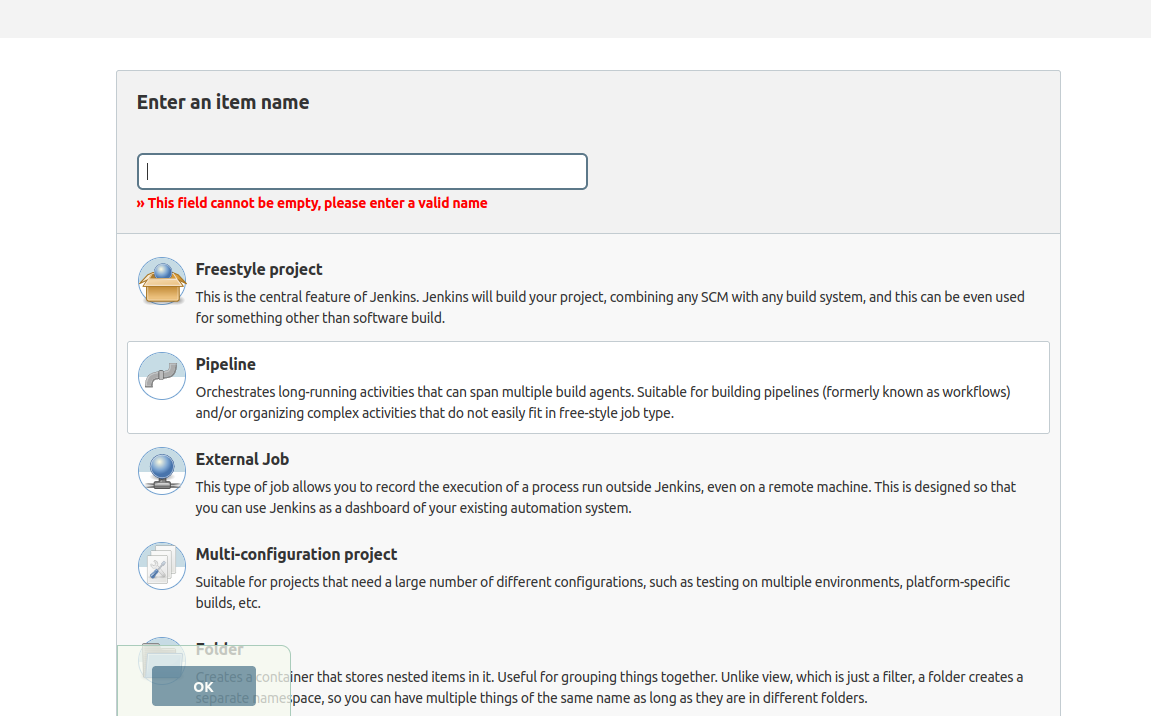
Chọn save

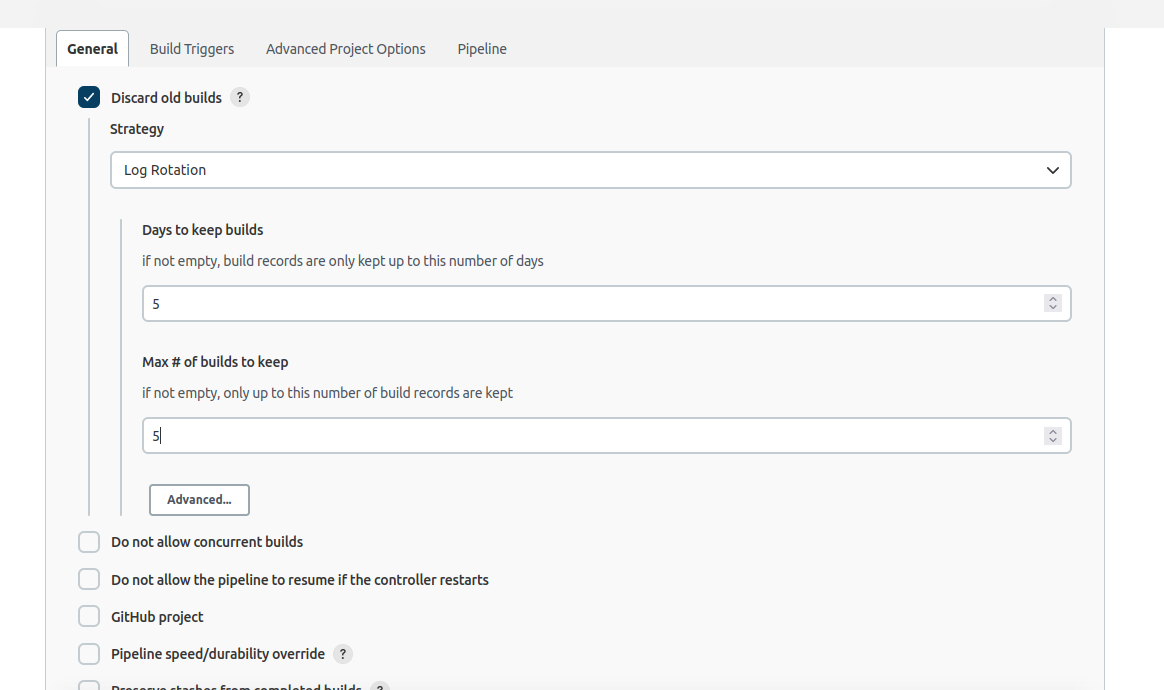
Chọn branch to build \*/main

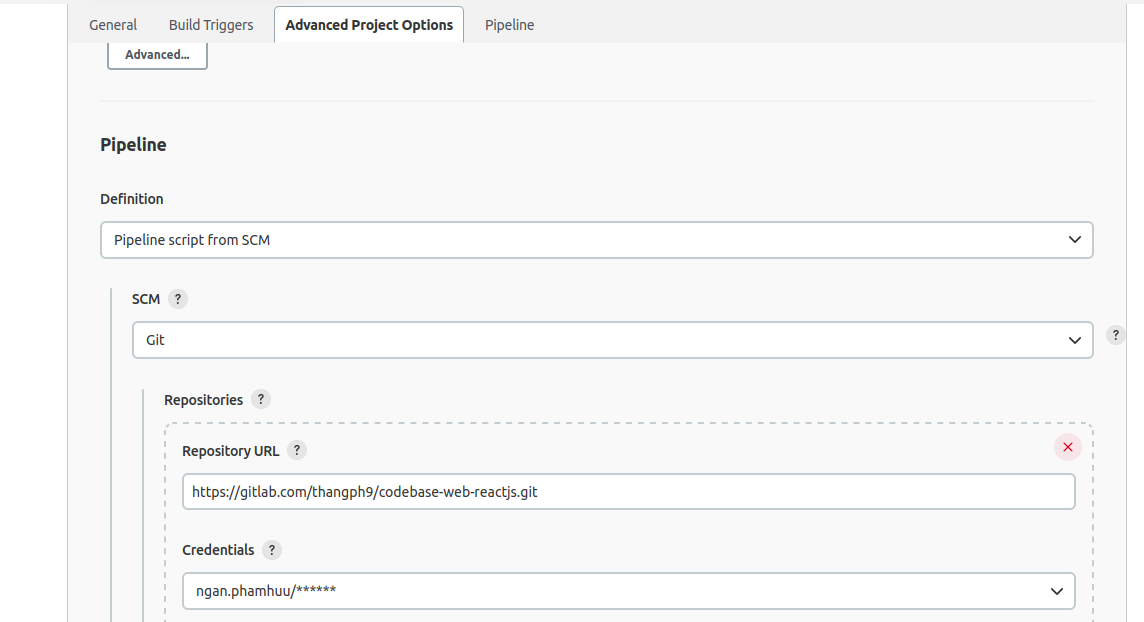
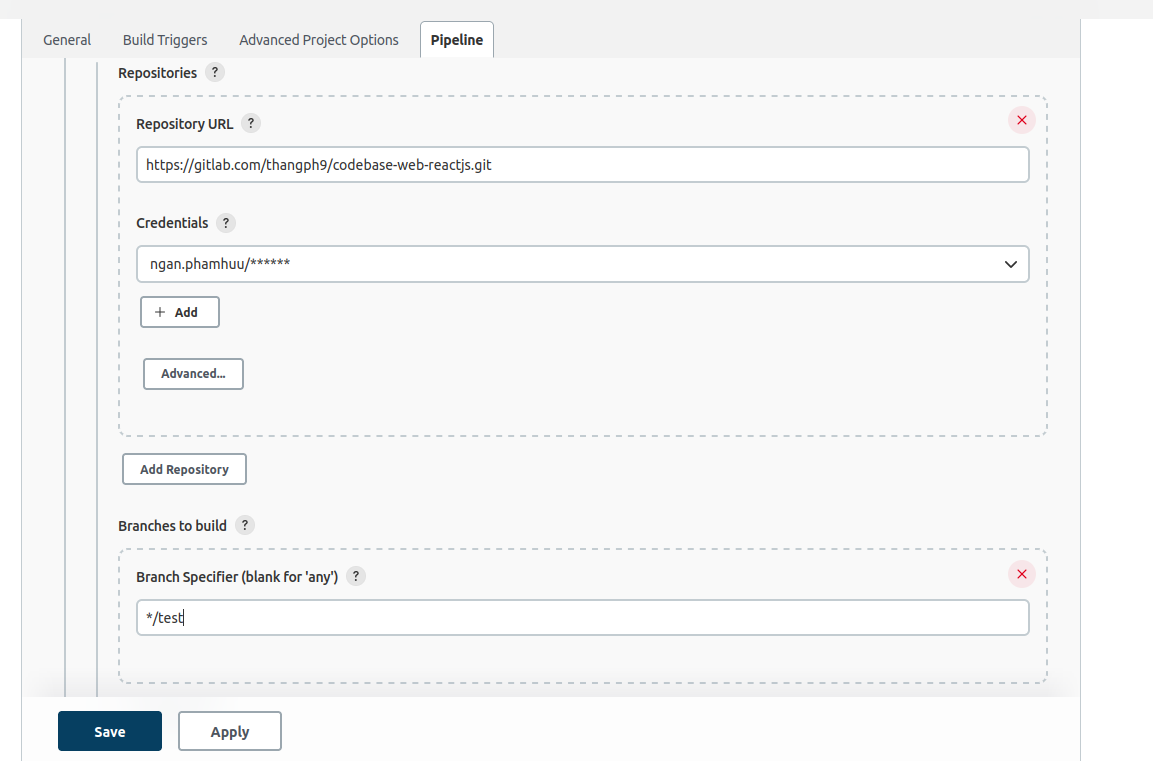


## Frontend:

Truy cập URL: localhost:8080/view/all/newJob

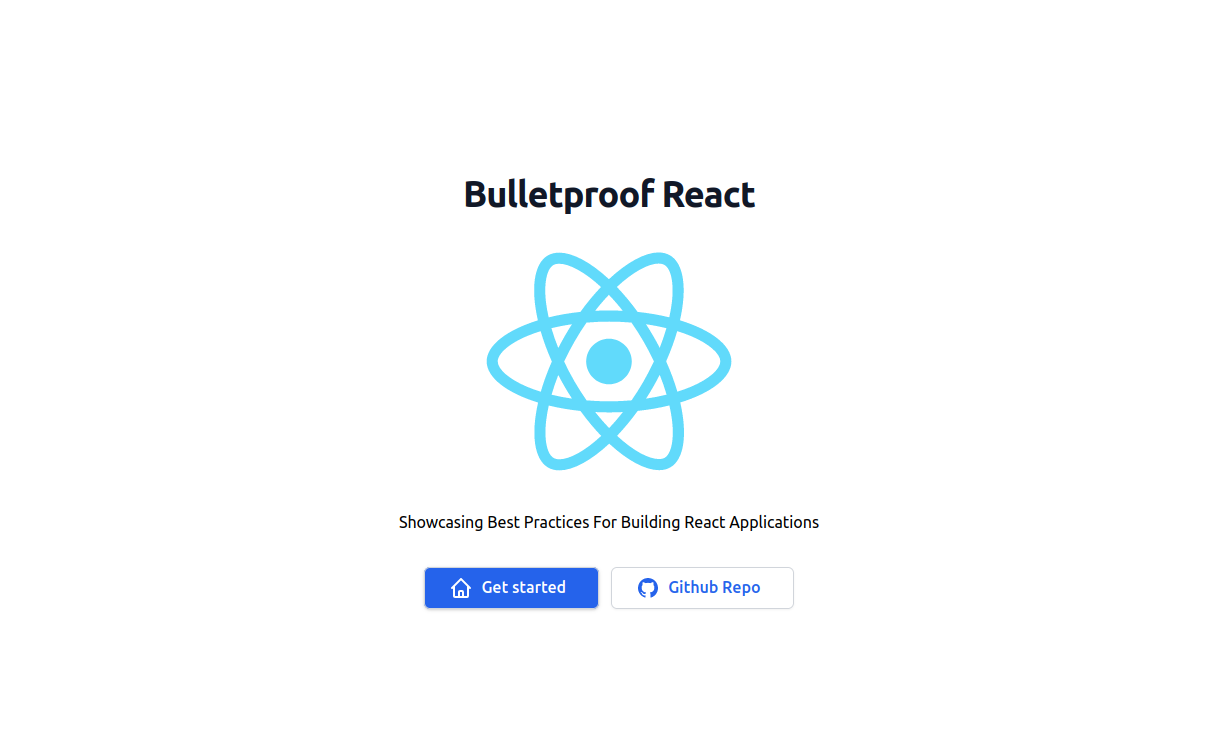
  
Nhập tên project chọn pipeline



Kiểm tra kết quả:

Nhập URL: localhost:8443

File cấu hình

### Dockerfile

### STAGE 1:BUILD ###

FROM node:14-alpine AS dev

WORKDIR /app

#

# COPY package.json /app/package.json

# COPY package-lock.json /app/package-lock.json

# Same as npm install

RUN npm cache clean --force

RUN npm install -D typescript@4.6.4

COPY . /app

ENV HUSKY\_SKIP\_INSTALL=1

RUN yarn install

ENV CI=false

ENV PORT=8443

FROM dev AS build

RUN CI=false yarn build

### STAGE 2:RUN ###

FROM nginx:latest AS nginx

# Copy config nginx

COPY --from=build /app/.nginx.conf /etc/nginx/conf.d/default.conf

WORKDIR /usr/share/nginx/html

# Remove default nginx static assets

RUN rm -rf ./\*

# Copy static assets from builder stage

COPY --from=build /app/build .

# Containers run nginx with global directives and daemon off

ENTRYPOINT ["nginx", "-g", "daemon off;"]

### docker-compose.yml

version: '3.5'

networks:

default:

driver: bridge

services:

fe:

build: .

image: fe:latest

container\_name: fe\_react

ports:

- 8443:8443

### Jenkinsfile

#!/usr/bin/env groovy

node {

stage('checkout') {

checkout scm

}

stage('check docker') {

sh "docker --version"

}

stage('down') {

sh "docker-compose down -v --remove-orphans"

}

stage('deploy') {

sh "docker-compose up --build -d"

}

}