Node App Deployment using Docker

1. Project 1 : Docker-build (To create image from Dockerfile)

Requirement:

Dockerfile

**# Use an official Node.js runtime as a base image**

**FROM node:21**

**# Set the working directory in the container**

**WORKDIR /usr/src/app**

**# Copy package.json and package-lock.json to the working directory**

**COPY package\*.json ./**

**# Install application dependencies**

**RUN npm install**

**# Copy the application code to the working directory**

**COPY . .**

**# Expose the port your app will run on**

**EXPOSE 3000**

**# Command to run your application**

**CMD ["node","index.js"]**

Package.json

**{**

**"name": "node-app",**

**"description": "hello jenkins test app",**

**"version": "0.0.1",**

**"private": true,**

**"dependencies": {**

**"express": "3.12.0"**

**},**

**"devDependencies": {**

**"mocha": "10.2.0",**

**"supertest": "6.3.3"**

**}**

**}**

Index.js

**var express = require('express');**

**var app = express();//Respond with "hello world" for requests that hit our root "/"**

**app.get('/', function (req, res) {**

**res.send('changed not ');**

**});//listen to port 3000 by default**

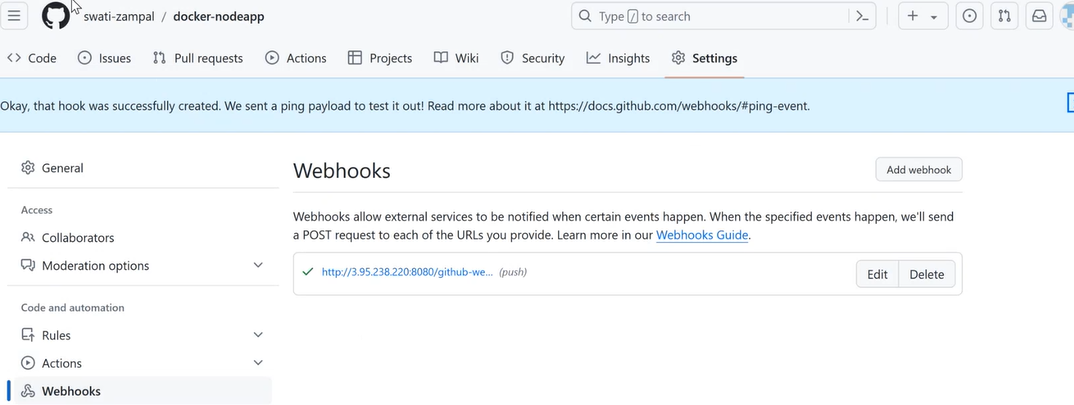
**app.listen(process.env.PORT || 3000);**

**module.exports = app;**

On Jenkins server

* Create a git repo for this project

And set webhook for Jenkins



* Create a directory DockerDemo

mkdir DockerDemo

cd DockerDemo

nano Dockerfile

nano package.json

nano index.js

git init

git config - -global user.name “Swati”

git config - -global user.name “[ss@gmail.com](mailto:ss@gmail.com)”

git pull gitrepolink

git remote add origin gitrepolink

git add .

git commit –m “Commit 1”

* Install docker

sudo apt update

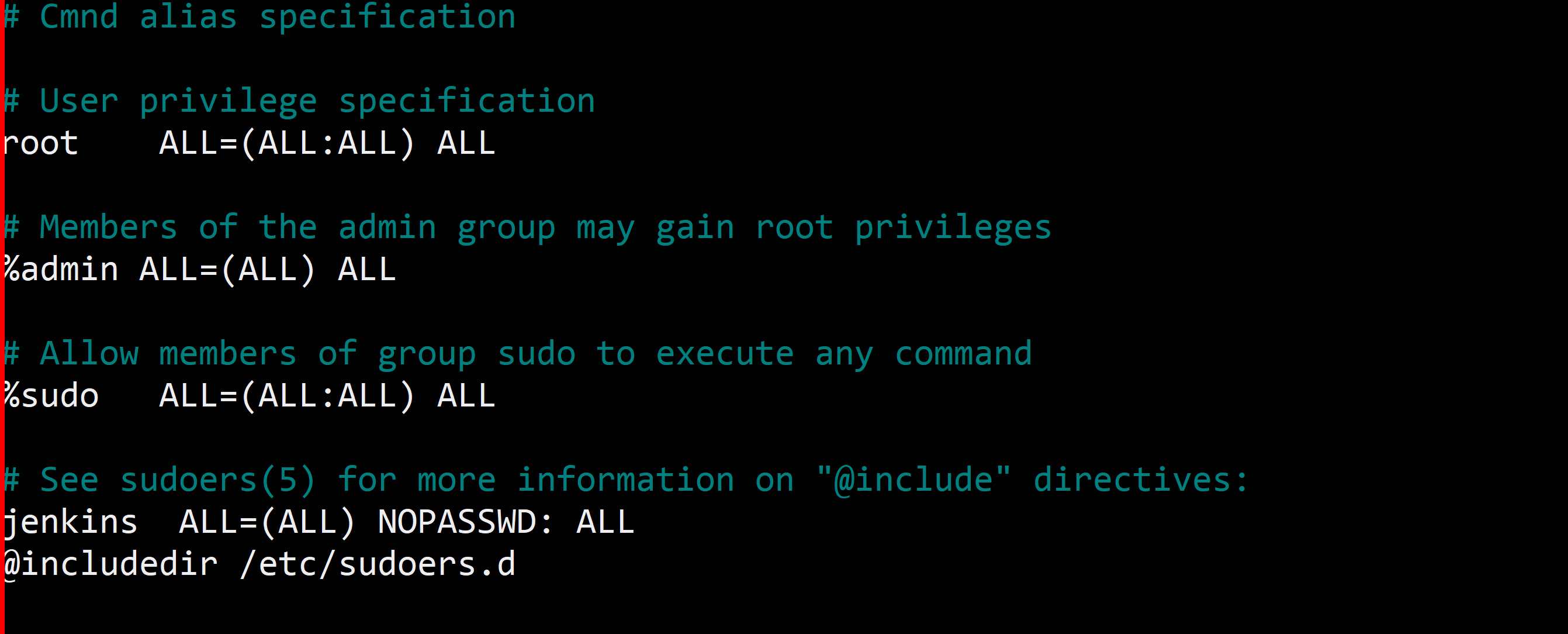
sudo apt install docker.io –y

* Give sudo permissions to Jenkins user

sudo nano /etc/sudoers

##Add below line

jenkins  All=(All) NoPASSWD: ALL

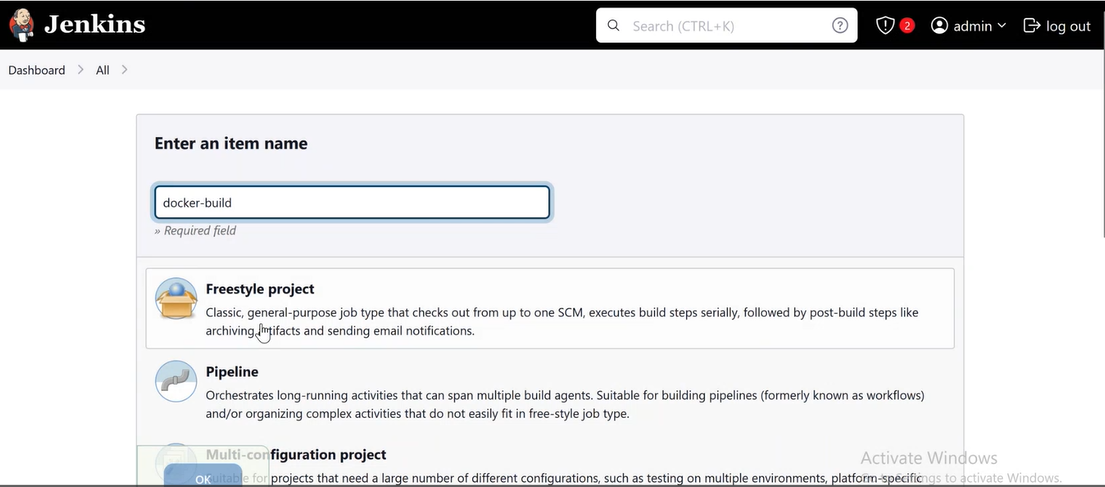


* To give permission to Jenkins user to run docker commands add Jenkins to docker group

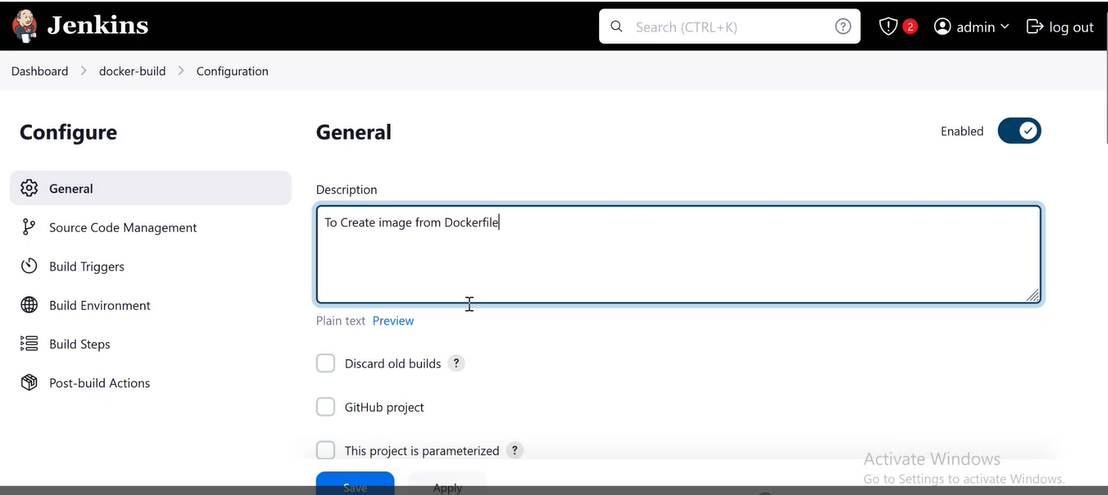
sudo usermod -aG docker jenkins

Hit IP of Jenkins server with port 8080 and Login to Jenkins

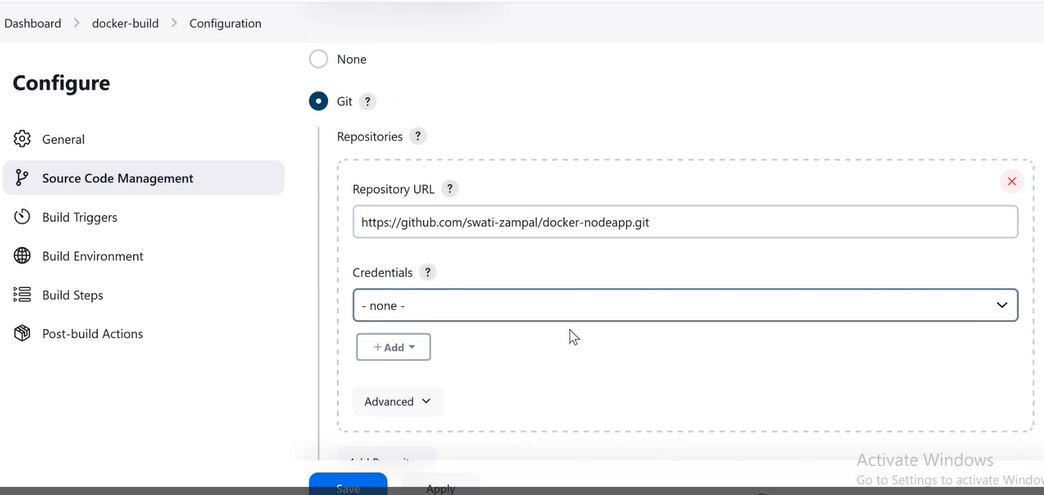
Create a Freestyle project of docker-build



Step 1: General



Step 2: SCM



Step 3: Build Trigger



Step 4: Build Environment

(No change)

Step 5: Build Step

Execute Shell

sudo docker build –t myimg .

sudo docker images



Save it

On Jenkins server

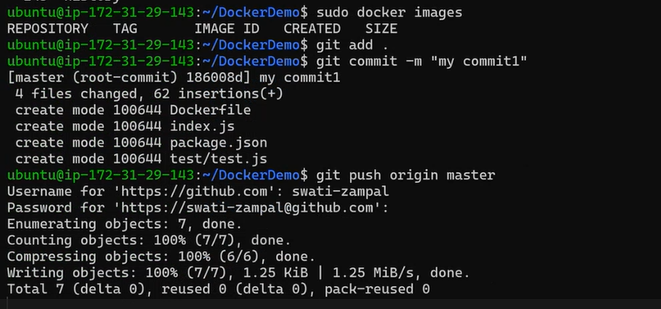
* Check docker images

Sudo docker images

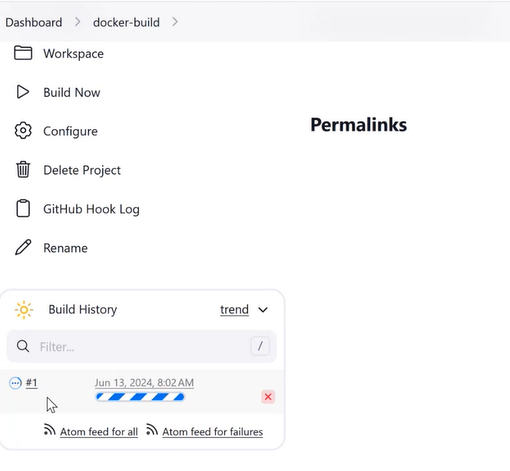
* To push files to git hub

cd DockerDemo

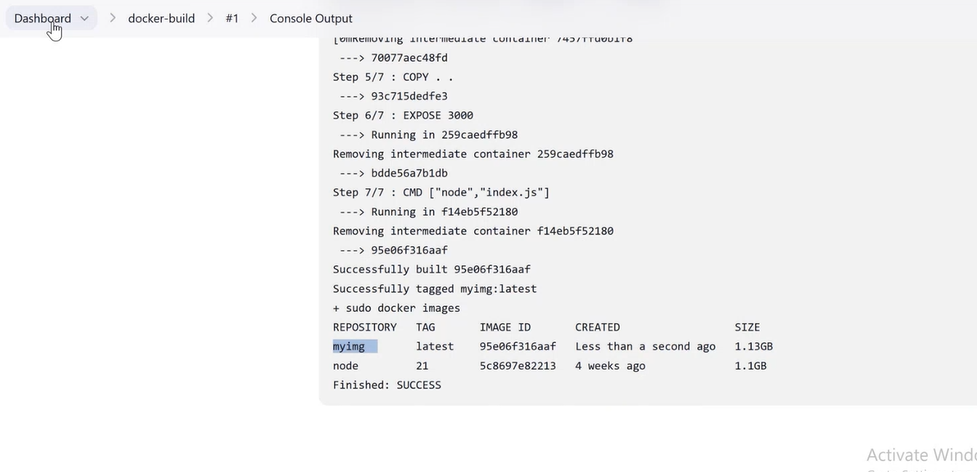
git push origin master



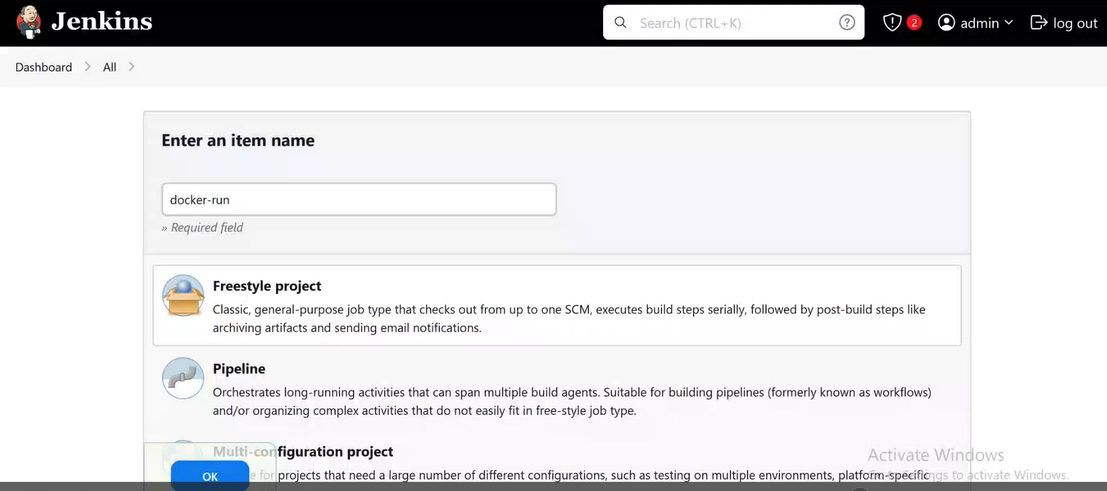
On Browser:



Check console output



Project 2: docker-run



On Jenkins server

cd DockerDemo

mkdir test

sudo nano test/dockertest

**# Wait for the container to be running**

**max\_attempts=30**

**current\_attempt=0**

**while [ $current\_attempt -lt $max\_attempts ]; do**

**container\_status=$(sudo docker ps --format "{{.Names}}: {{.Status}}" | grep 'nodecontainer')**

**if [[ $container\_status == \*"Up"\* ]]; then**

**echo "Containers are running!"**

**break**

**else**

**echo "Waiting for containers to be running..."**

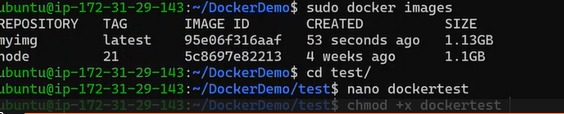
**sleep 30**

**((current\_attempt++))**

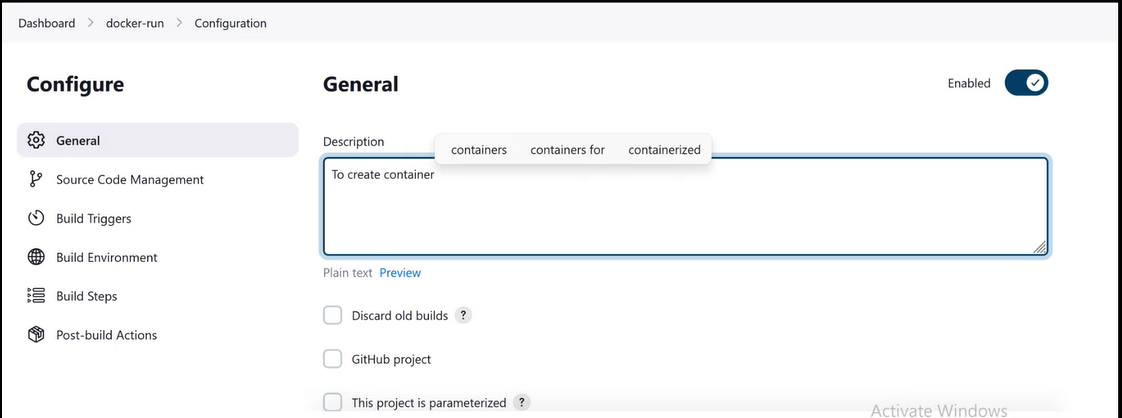
**fi**

**done**

chmod +x test/dockertest



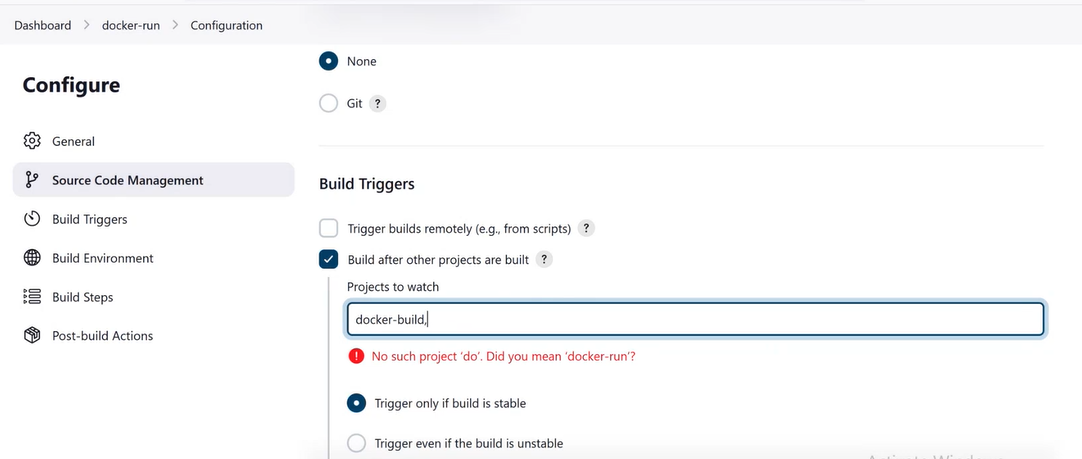
Step 1: General



Step 2: SCM

None

Step 3: Build Trigger



Step 4: Build Environment

(No change)

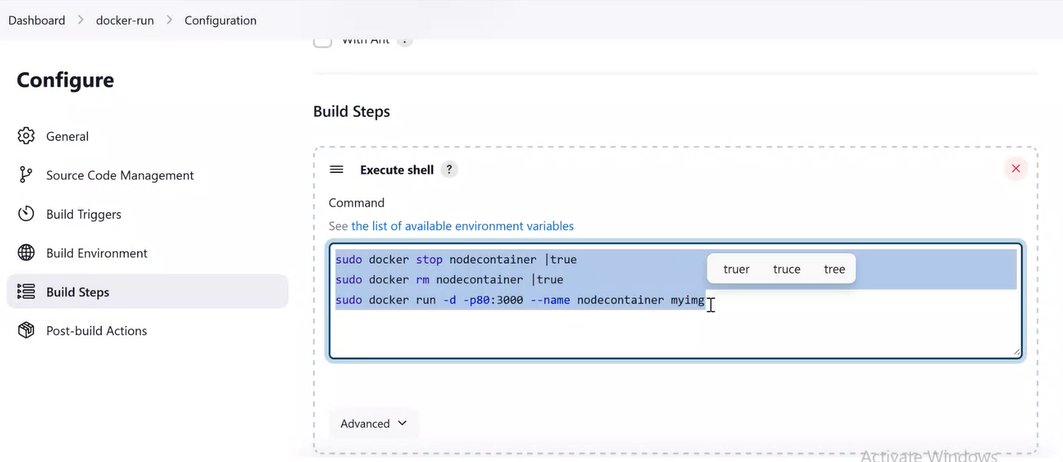
Step 5: Build Step

Execute Shell

sudo docker stop nodecontainer | true

sudo docker rm nodecontainer | true

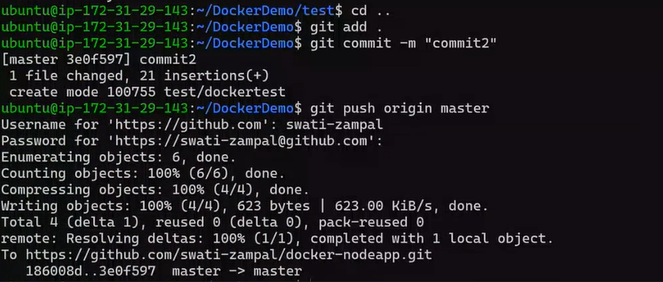
sudo docker run –d –p80:3000 –name nodecontainer myimg



Save it

On Jenkins server

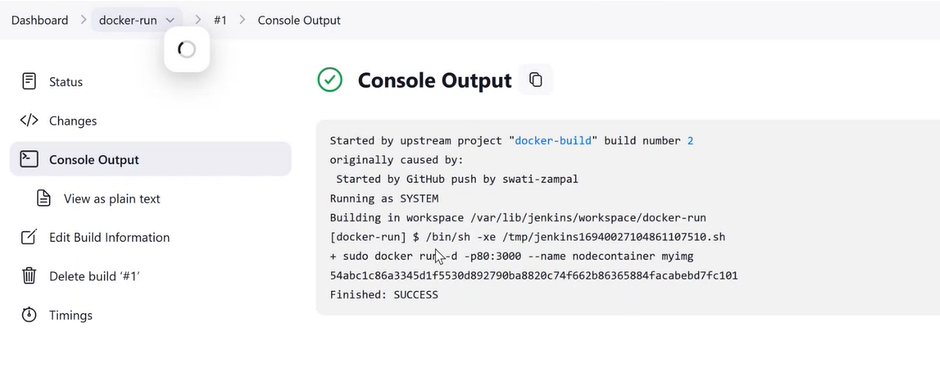
cd DockerDemo



Check on Browser

It builds docker-build project and after that docker-run project.

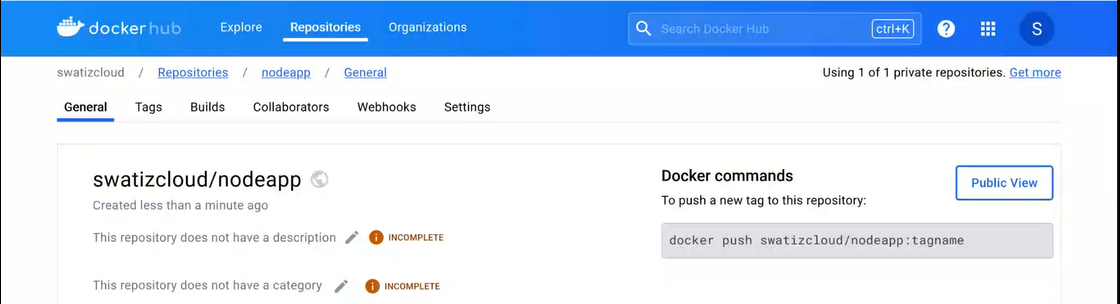
Docker-run build’s console output-



Project 3: docker-deploy

Requirement:

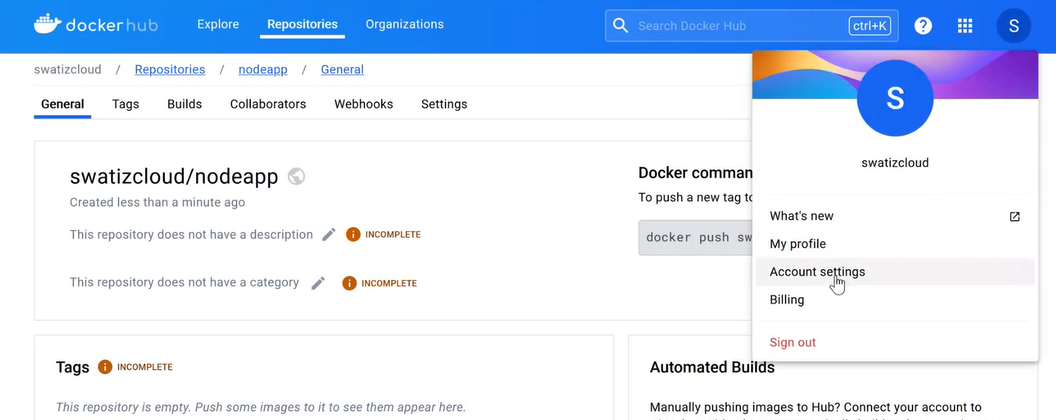
Create a public Repository on DockerHub to push image

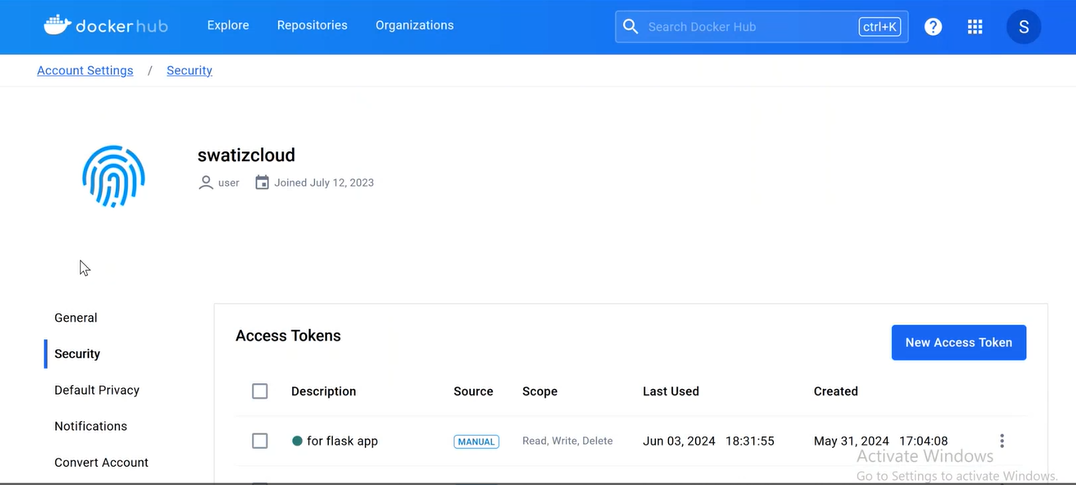


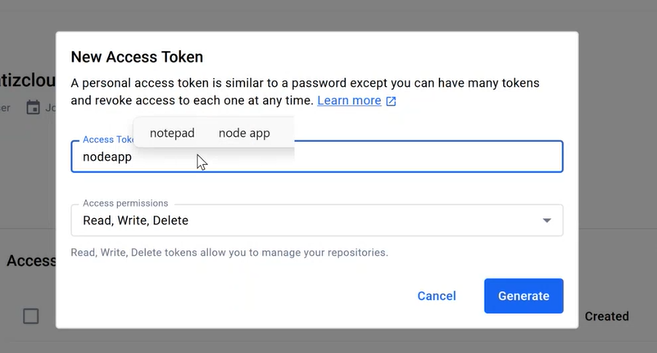
Create token:

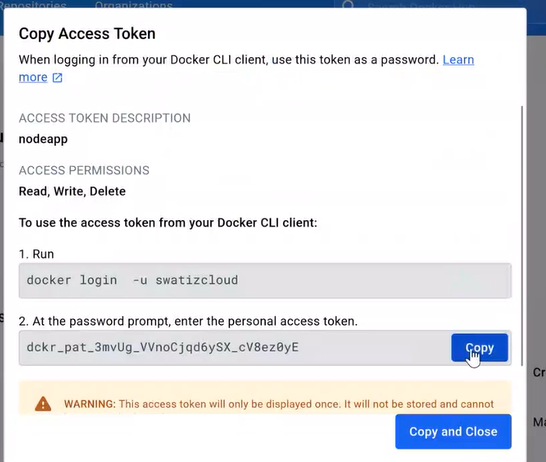
Profile name 🡪 Account Settings 🡪 security 🡪 New Access Token 🡪 Access Token Description 🡪 Generate

(Copy it and save same where)



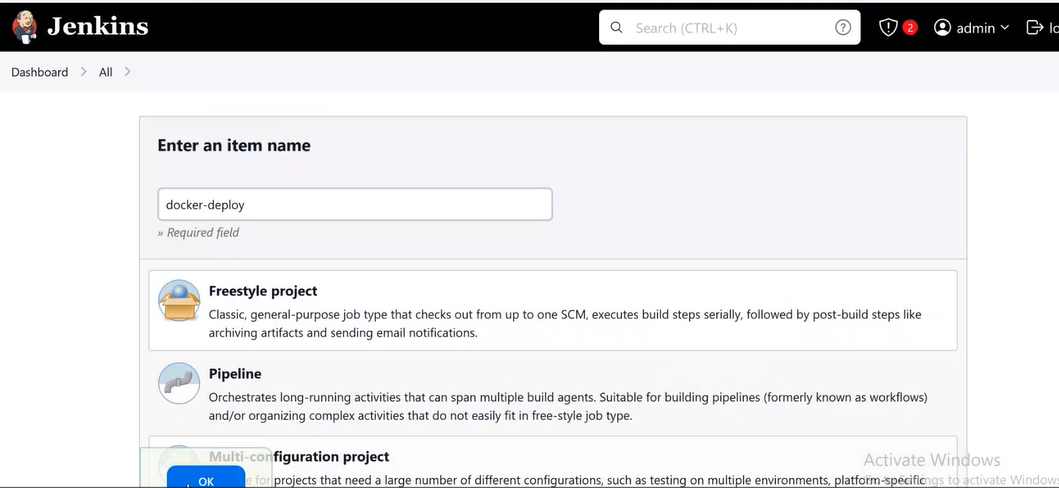




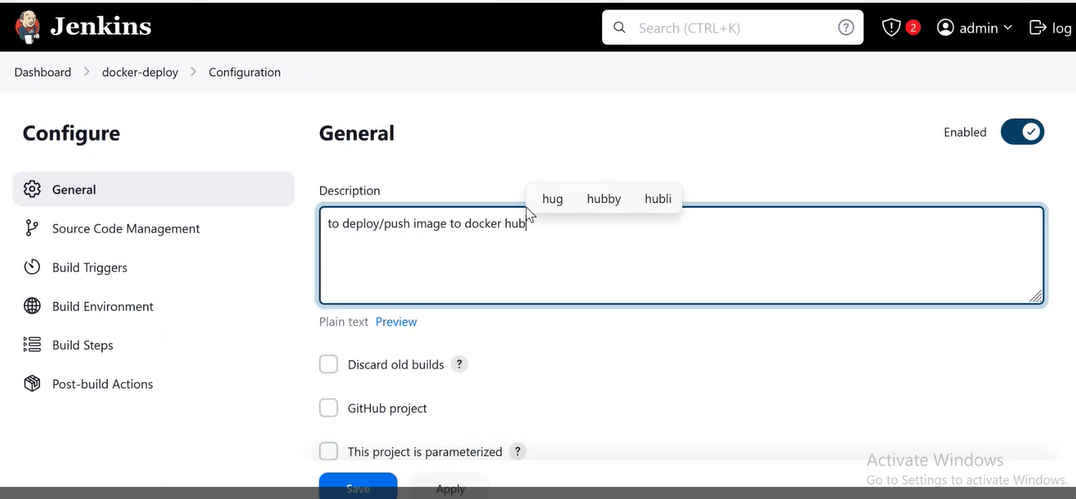


On Browser

Create a freestyle project docker-deploy



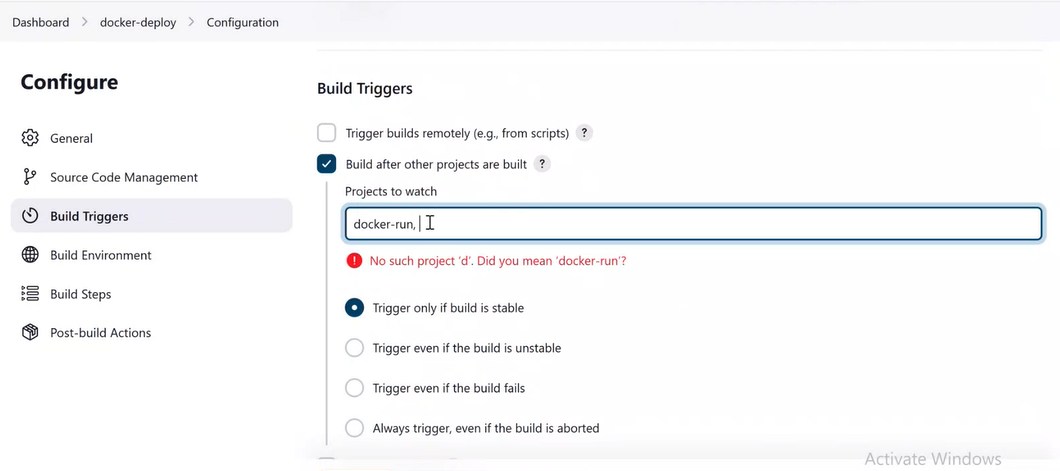
Step 1:



Step 2: SCM

None

Step 3: Build Trigger



Step 4: Build Environment

(No changes)

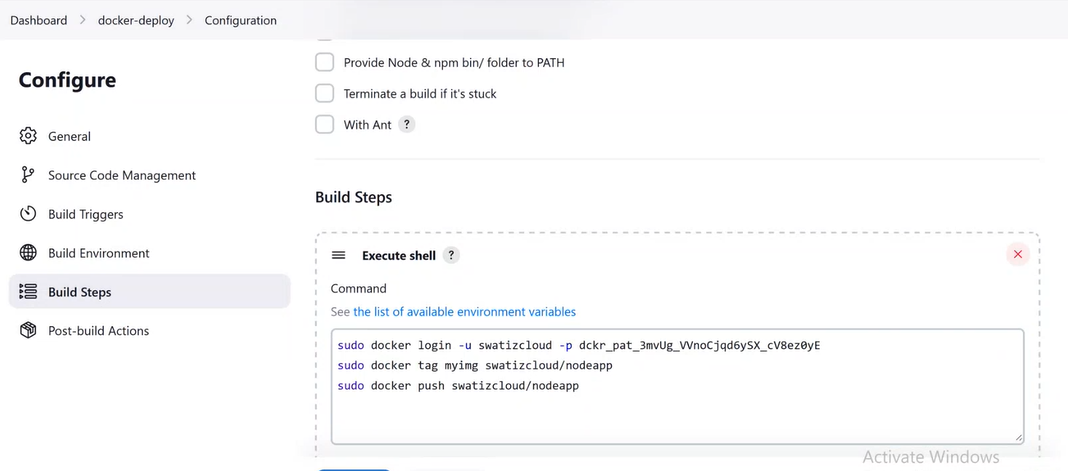
Step 5: Build Step

Execute shell

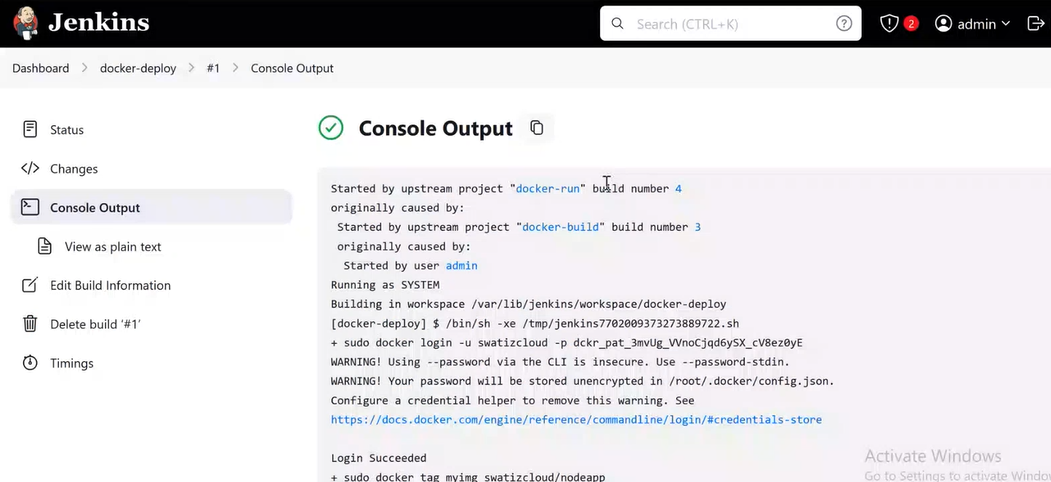
sudo docker login –u dockerhub-username –p dockerhub-token

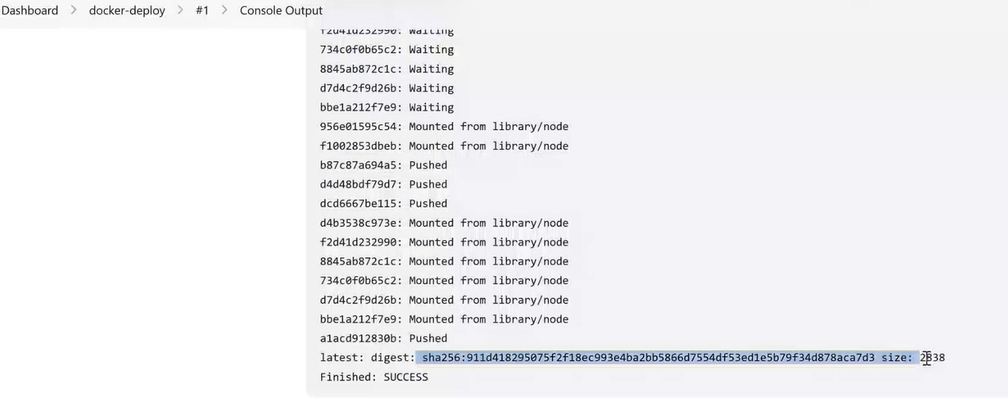
sudo docker tag myimg dockerhub-username/repo-name

sudo docker push dockerhub-username/reponame



Build projects and check





Go to Docker Hub and check image pushed

