**Certification** **Project II: Problem Statement**

A retail company Abstergo Corp. has recently setup an online shopping portal (website) to sell their products. Due to fierce competition, the company wants a solution that can reduce the time and effort it needs to enhance the functionality of their website on a regular basis. They are looking for an automated way to deploy the new code (for new features) to production website whenever they want.

**Business Requirements**

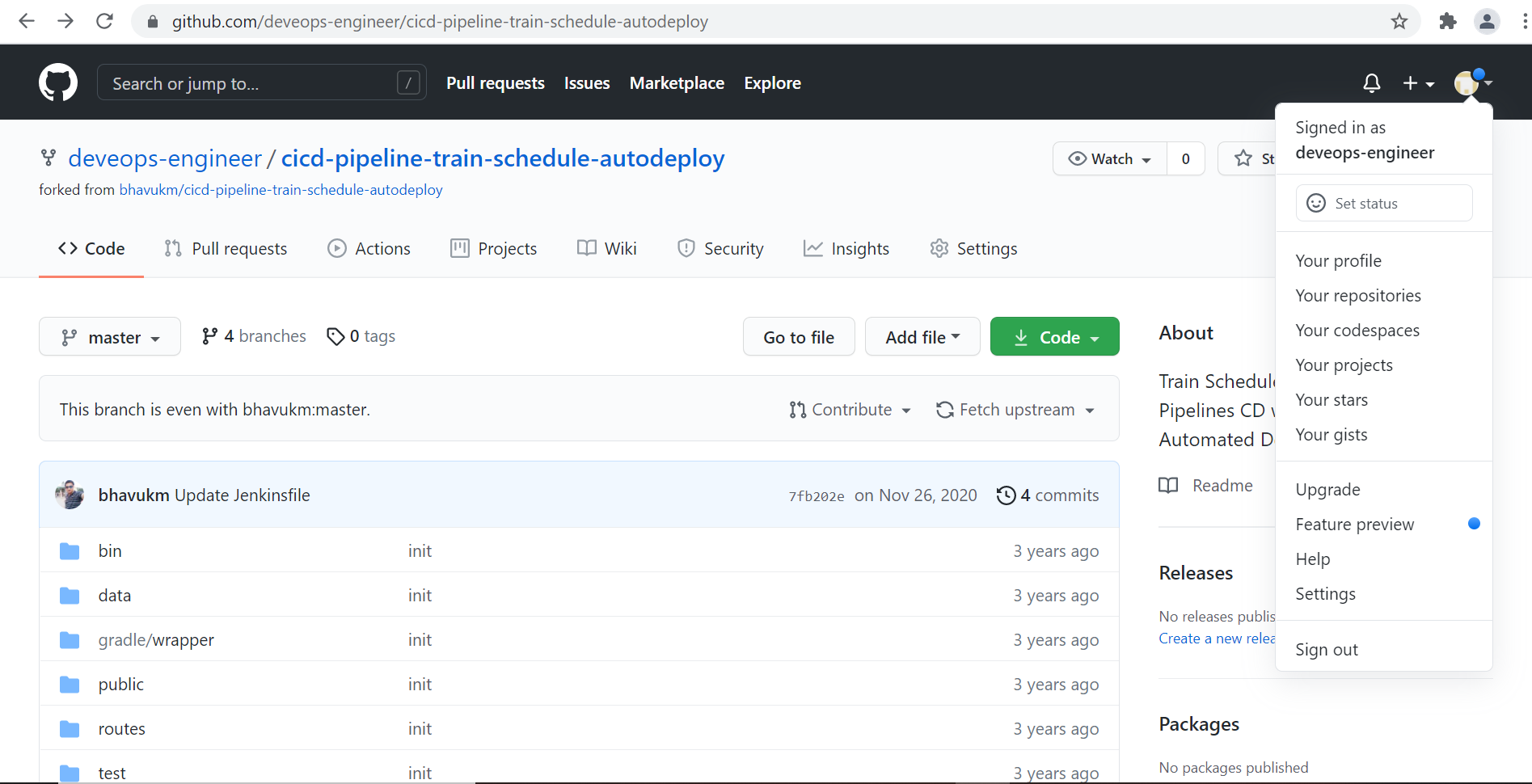
* The team of developers working on new features will merge their code to a GitHub repo.
* As soon as the code reaches GitHub, using a CI (Continuous Integration) pipeline, setup in Jenkins, automated builds will be triggered.
* The automated builds will frequently deploy new features to the production website.
* Every build will prepare a Docker file and push Docker images to docker-hub.
* Every docker image will be deployed (Continuous Deployment) to a kubernetes-cluster.

Fork the given repository to your own account and use it as the application for your pipeline project

GitHub: <https://github.com/bhavukm/cicd-pipeline-train-schedule-autodeploy>

**Solution:**

Above given repository has been forked to <https://github.com/deveops-engineer/cicd-pipeline-train-schedule-autodeploy.git> as screenshot attached below.



Prerequisite: - Installed Git, java, Jenkins, and Kubernetes.

yum install git -y

yum install java-1.8.0-openjdk-devel

alternatives –config java

copy - /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86\_64

vim /etc/profile

edit – export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-

1.amzn2.0.1.x86\_64

Export PATH=$JAVA\_HOME/bin:$PATH

source /etc/profile

echo $JAVA\_HOME

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io.key>

yum install jenkins

systemctl start jenkins

systemctl status jenkins

vim /etc/hosts

edit –

172.31.17.72 master-node

172.31.16.97 worker-node

hostnamectl set-hostname master-node

retenforce 0

( on worker node:

vim /etc/hosts

edit –

172.31.17.72 master-node

172.31.16.97 worker-node

hostnamectl set-hostname worker-node

retenforce 0 )

######### on both master and worker ###########

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=1

repo\_gpgcheck=1

gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg

EOF

yum install -y docker kubelet kubeadm kubectl

systemctl start docker kubelet

systemctl enable docker

systemctl enable kubelet

#########################################

kubeadm init --apiserver-advertise-address=172.31.17.72 --pod-network-cidr=172.168.0.0/16 --ignore-preflight-errors=NumCPU --ignore-preflight-errors=Mem

(run token on worker node)

su -ec2-user

mkdir -p $HOME/kube

sudo cp -I /etc/Kubernetes/admin.config $HOME/kube/config

sudo chown $(id-u)$(id-g)$ HOME/kube/config

kubectl get nodes

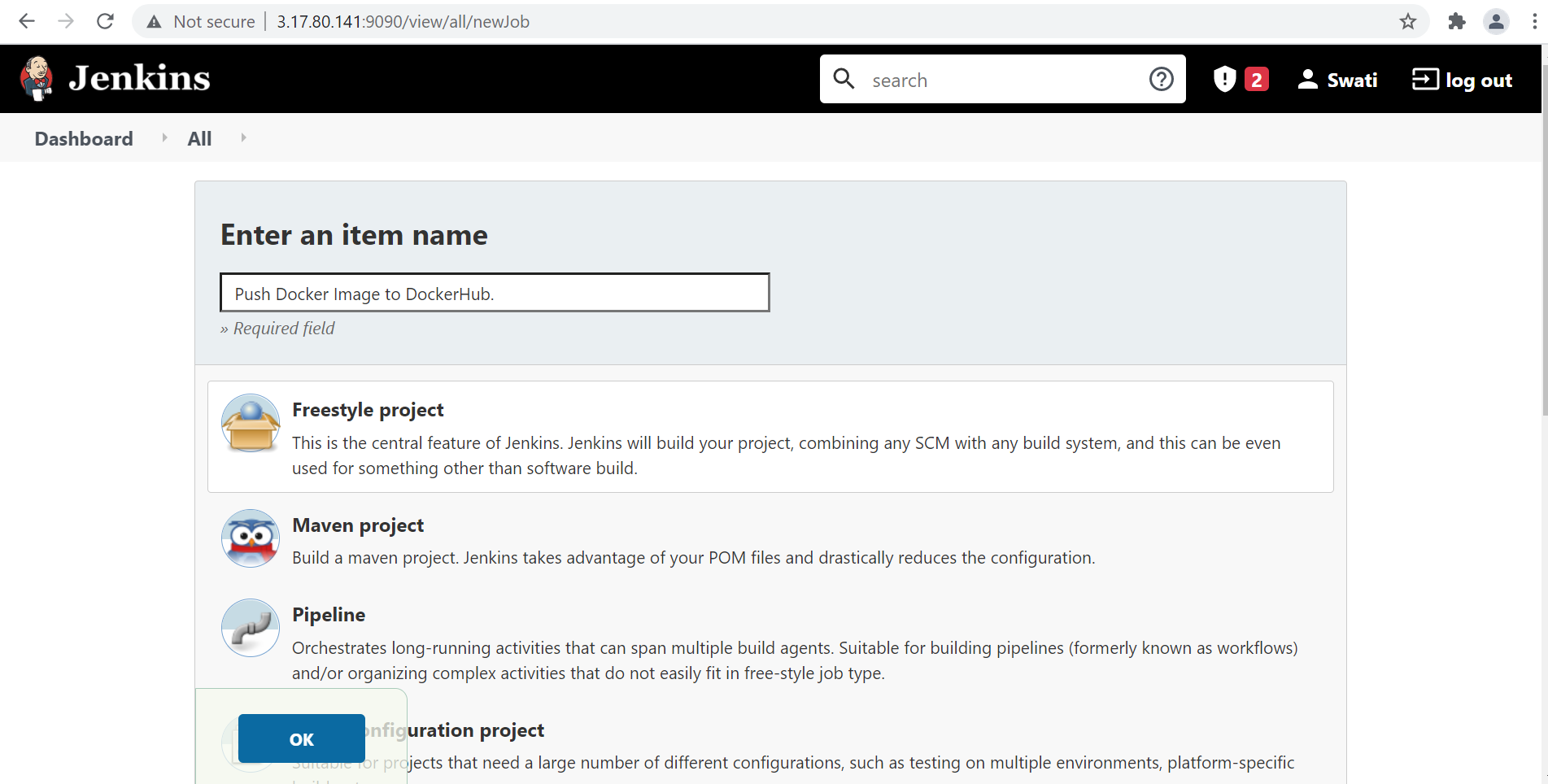
kubectl apply -f <https://docs.projectcalico.org/v3.3/getting-started/kubernetes/installation/hosted/rbac-kdd.yaml>

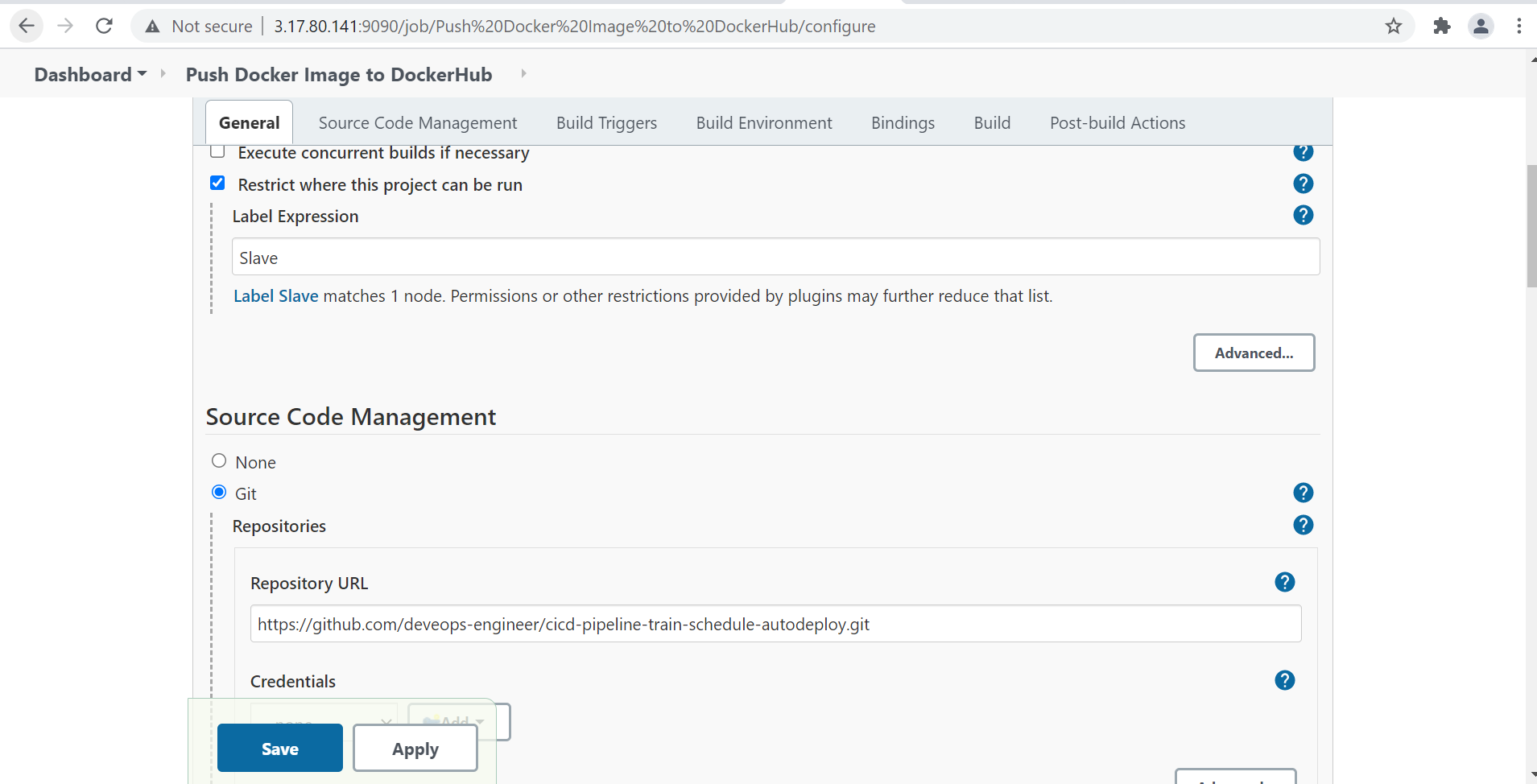
kubectl apply -f <https://docs.projectcalico.org/v3.8/getting-started/kubernetes/installation/hosted/kubernetes-datastore/calico-networking/1.7/calico.yaml>

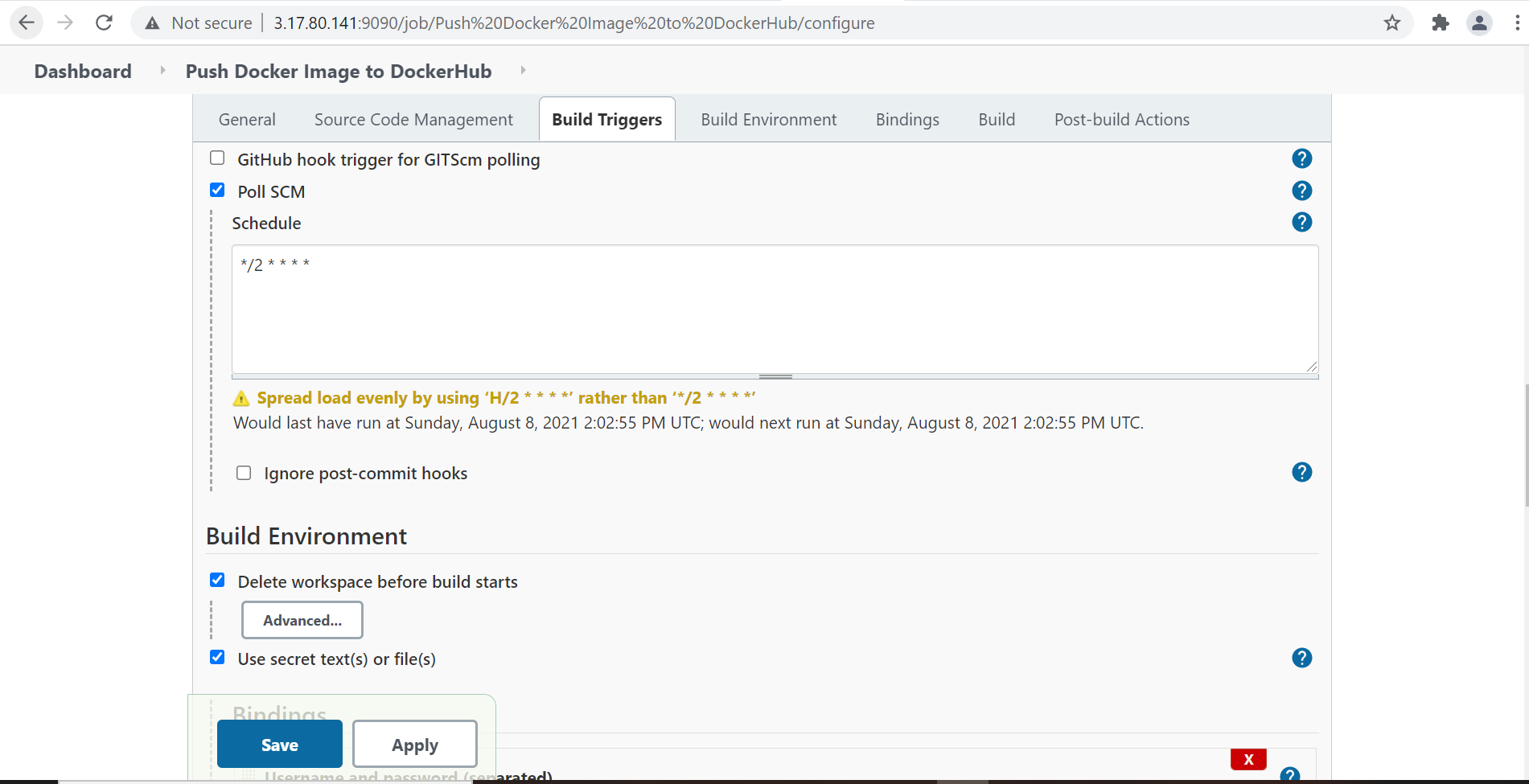
kubectl get nodes

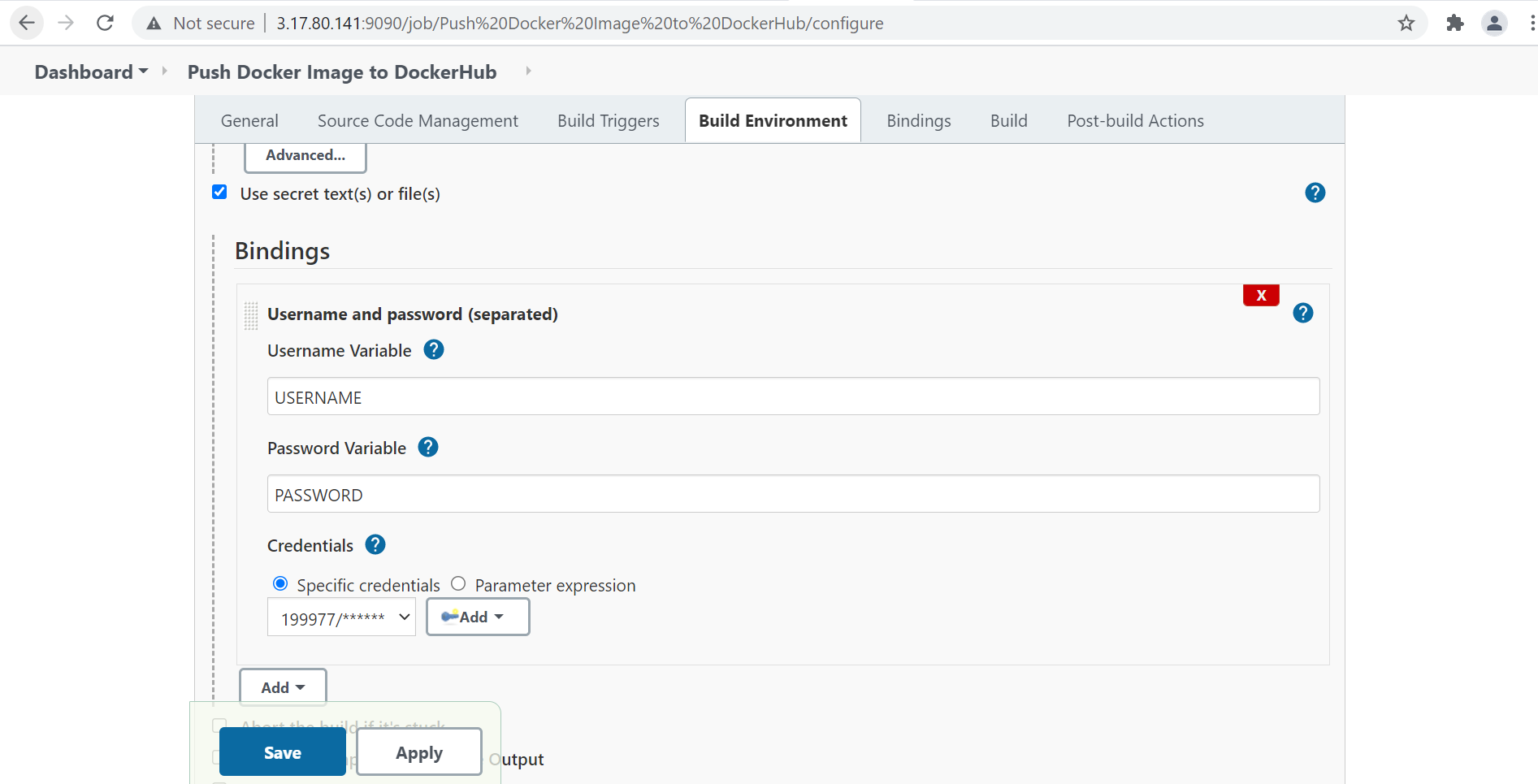
kubectl get pods

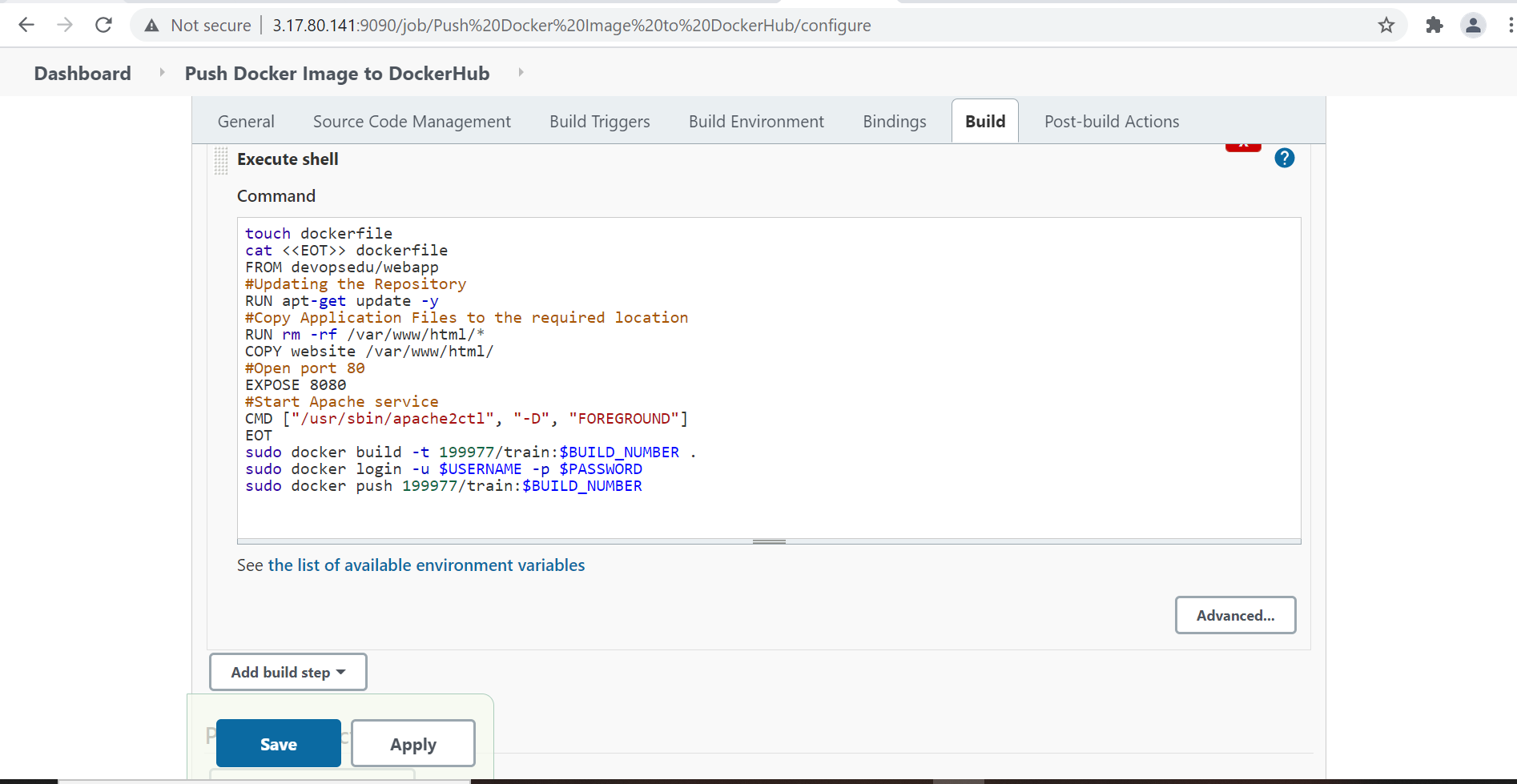
**Job 1** :- Build the code, make docker file and push that docker file to docker hub

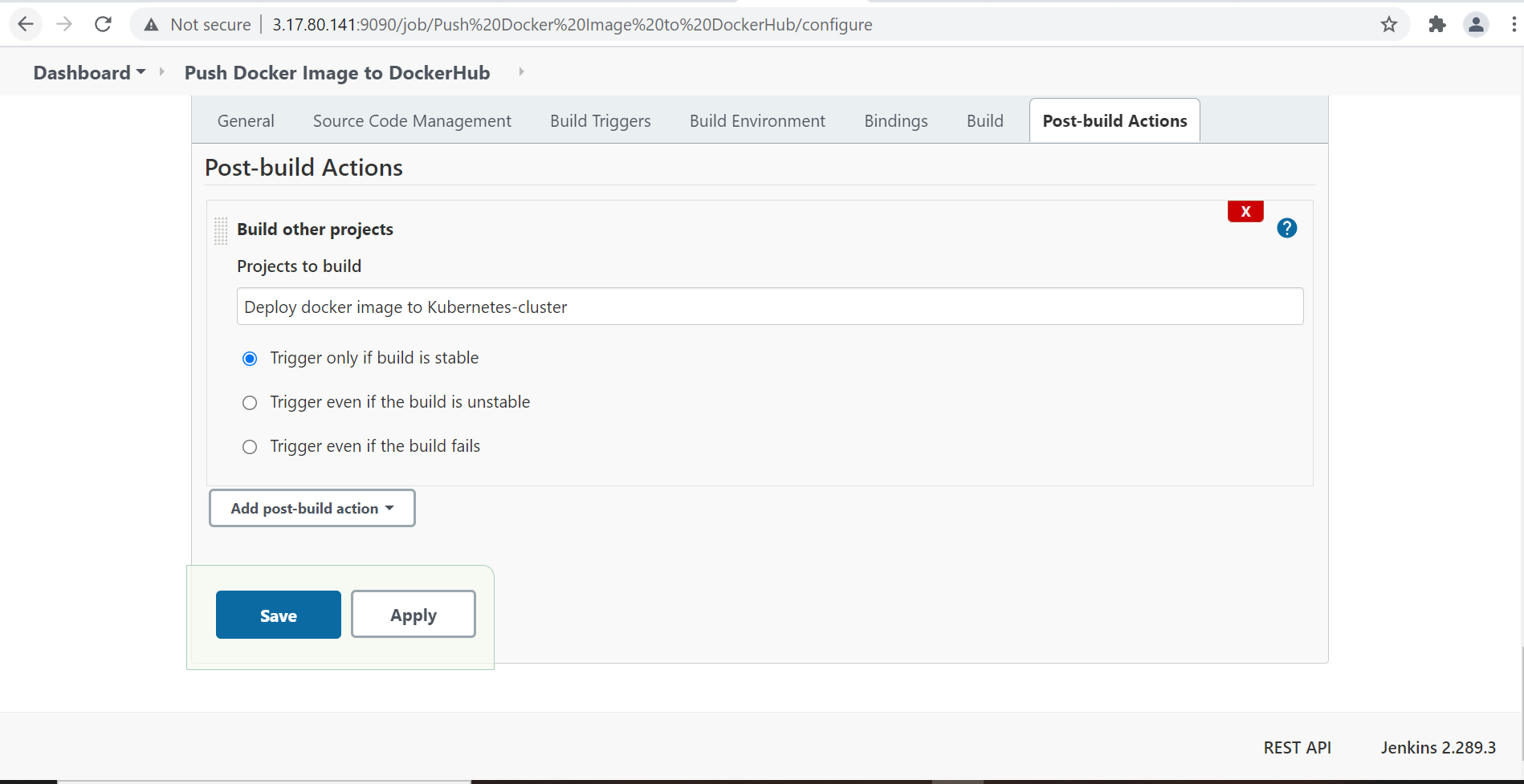




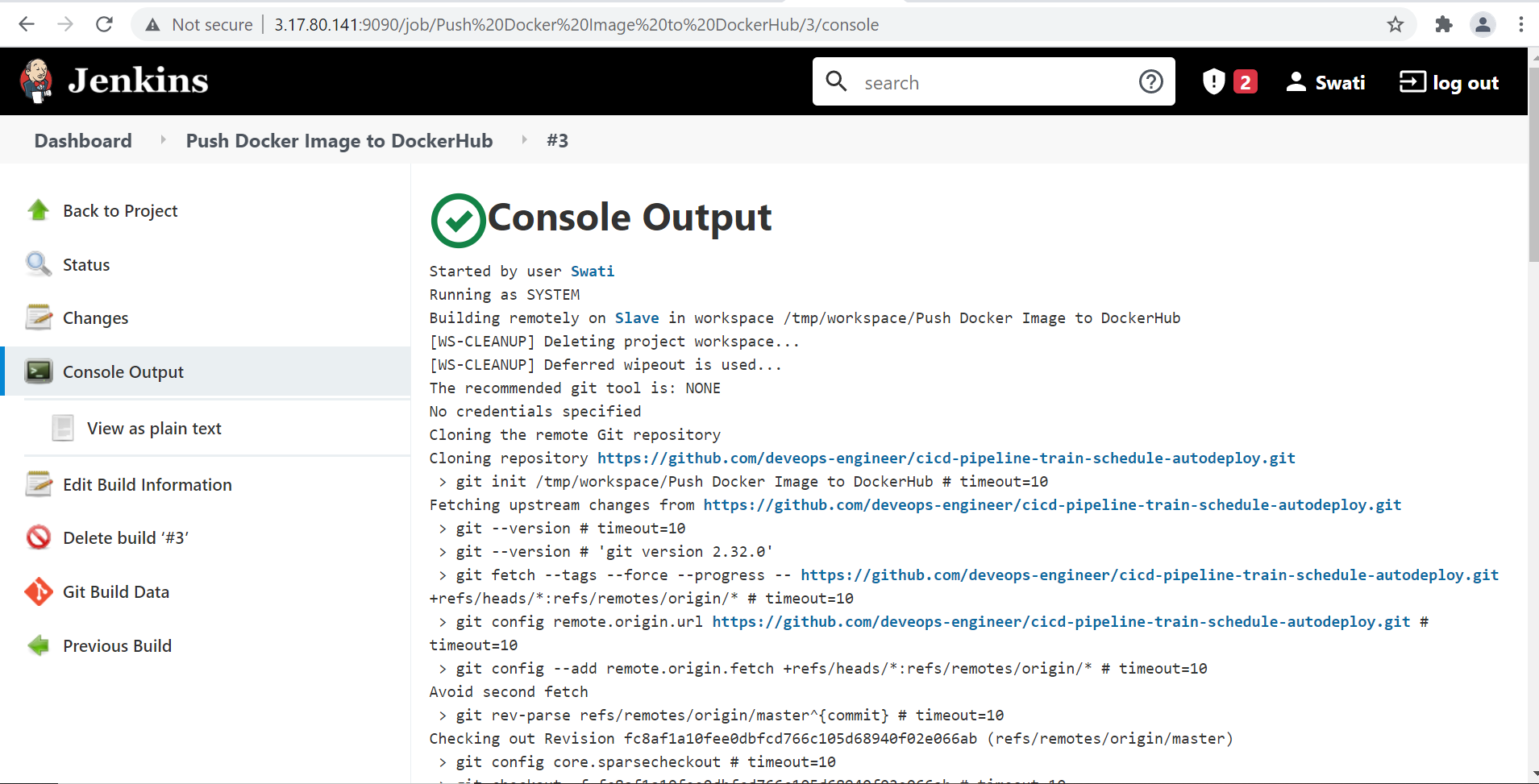


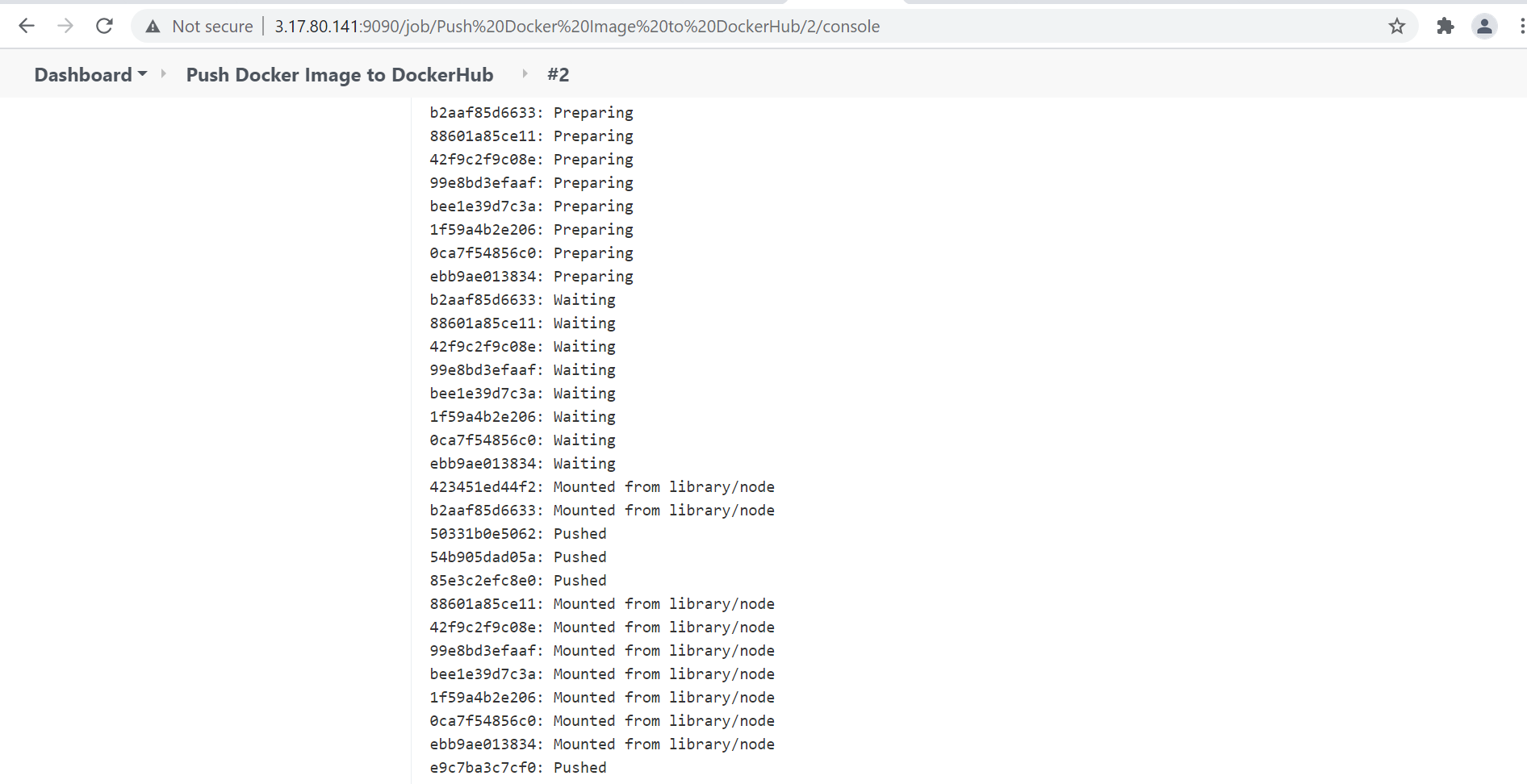


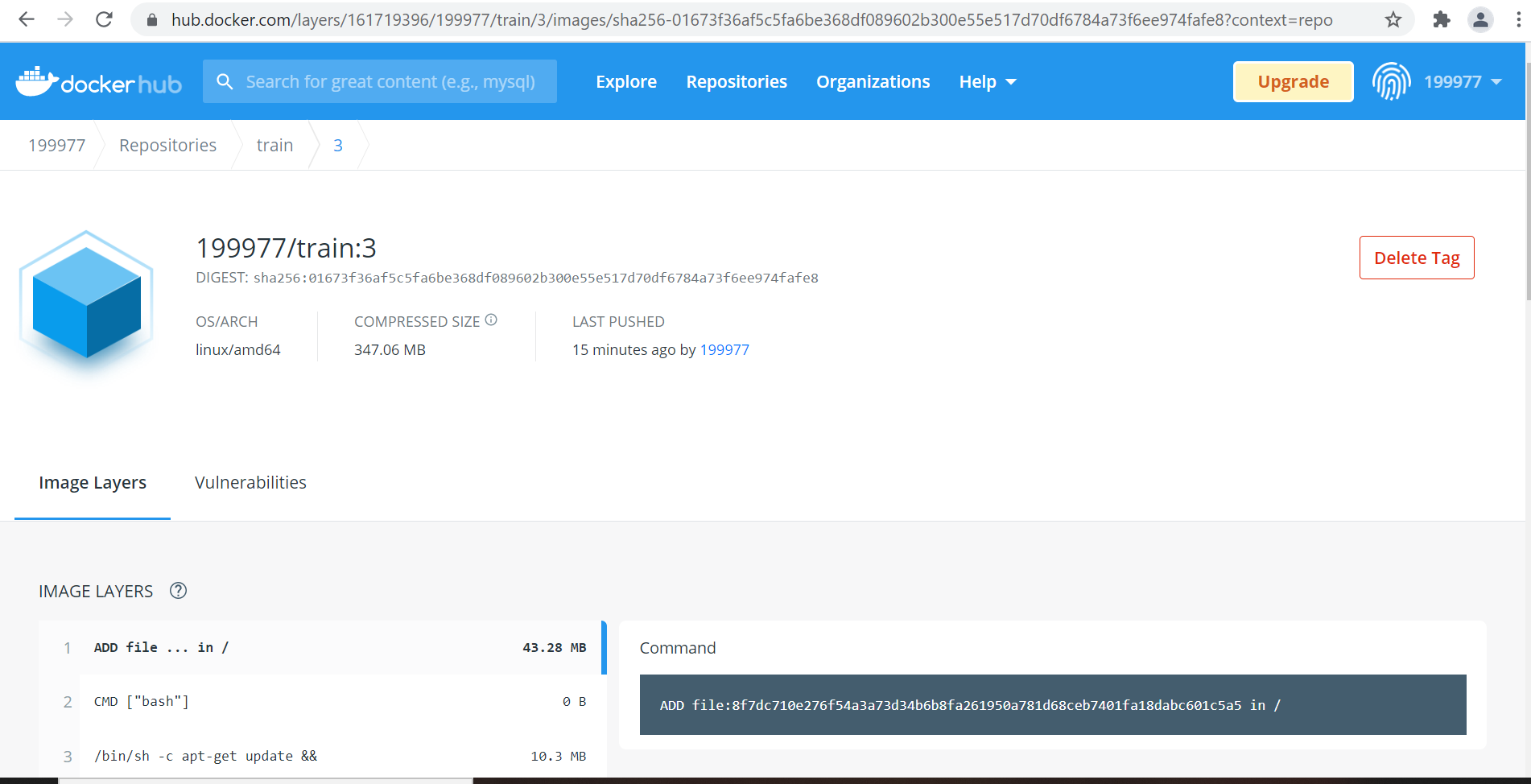




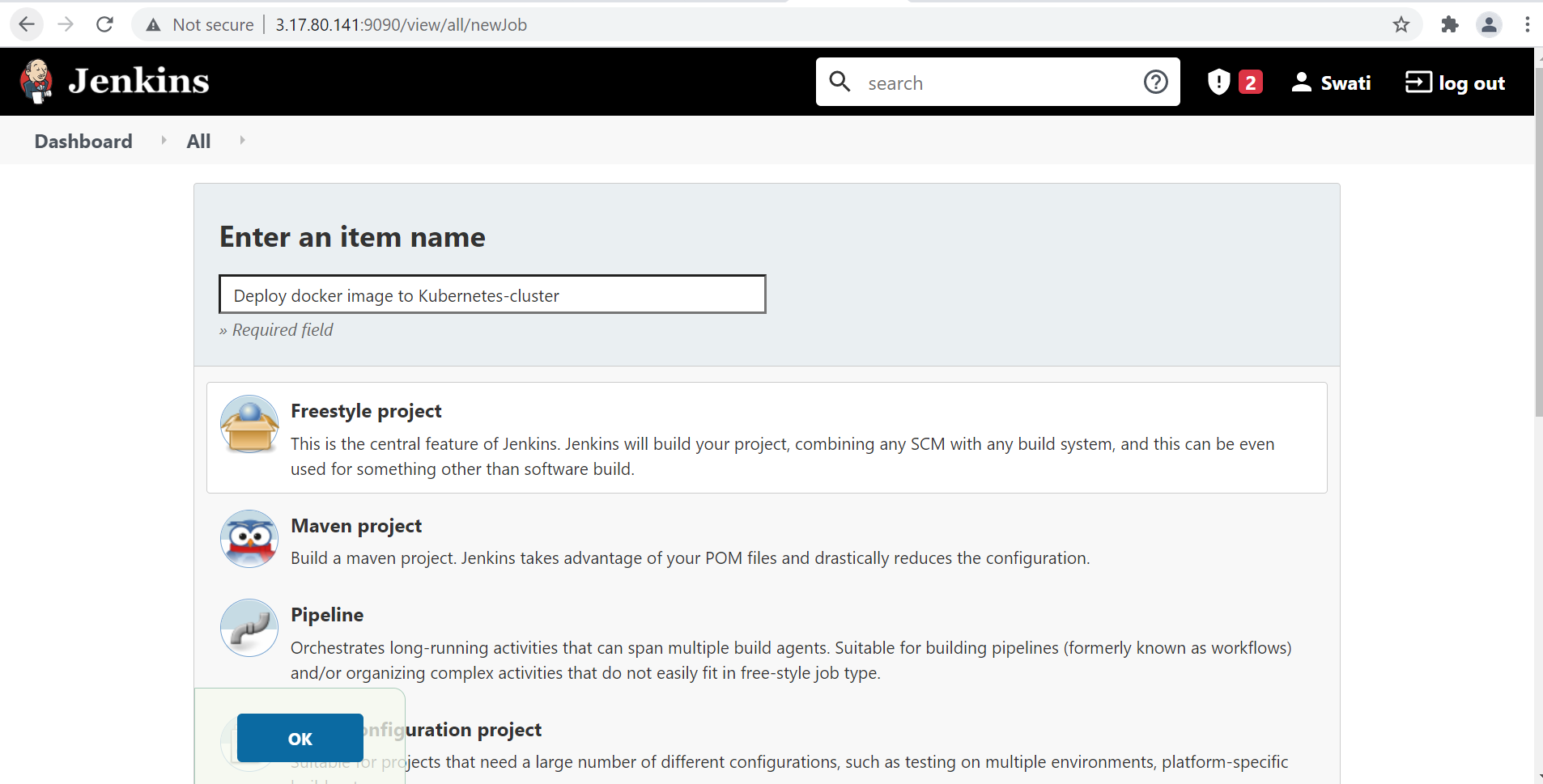
Output:

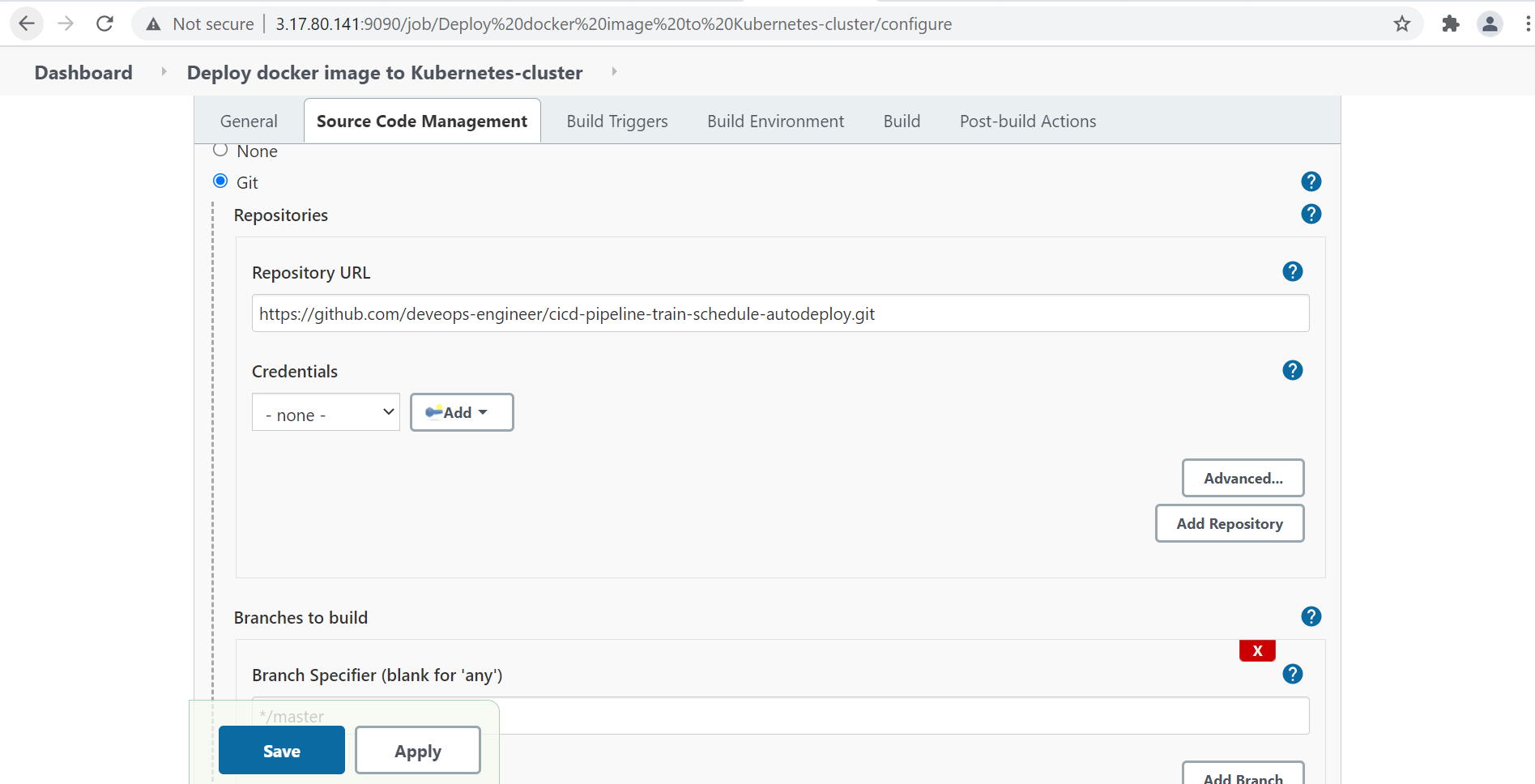


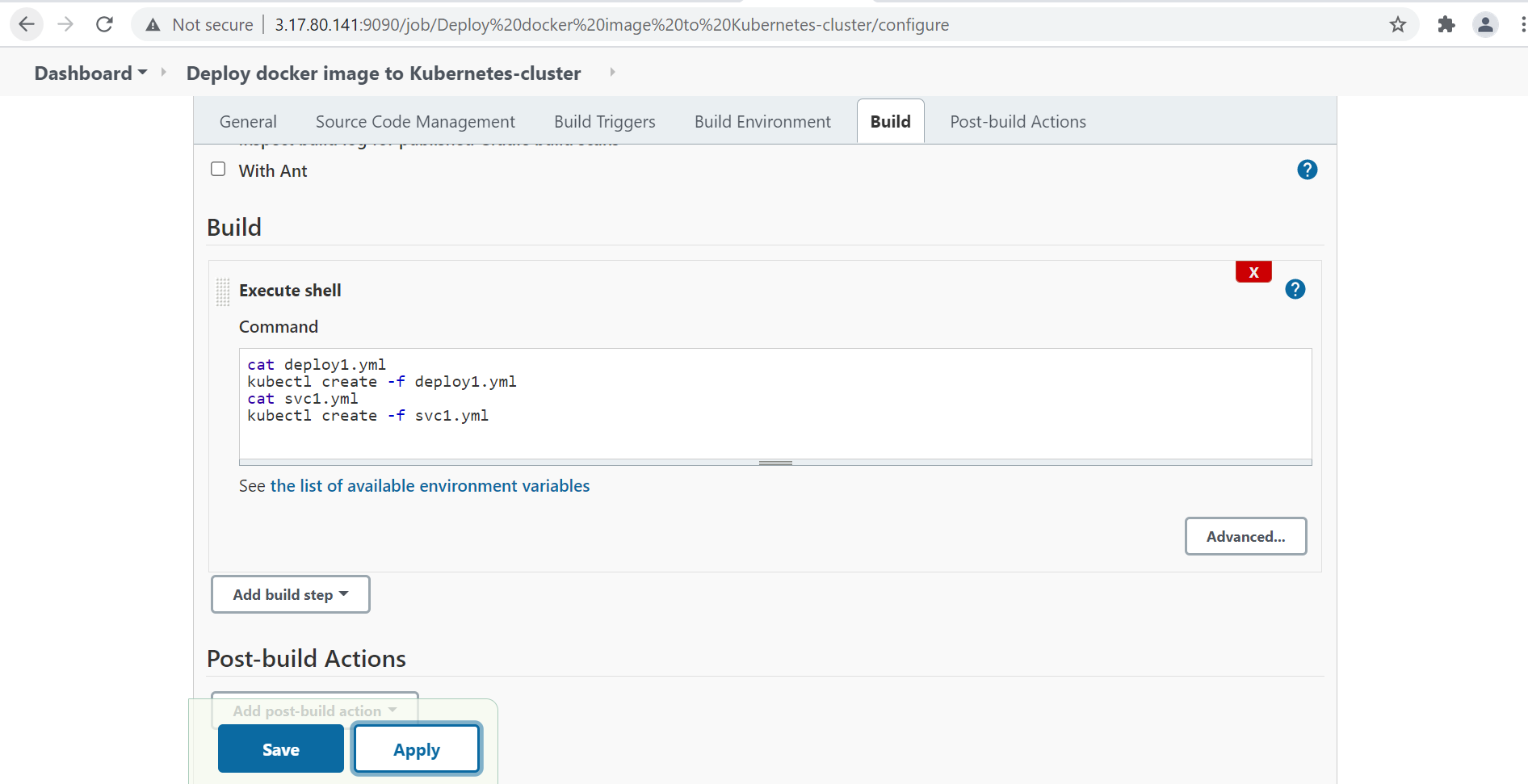


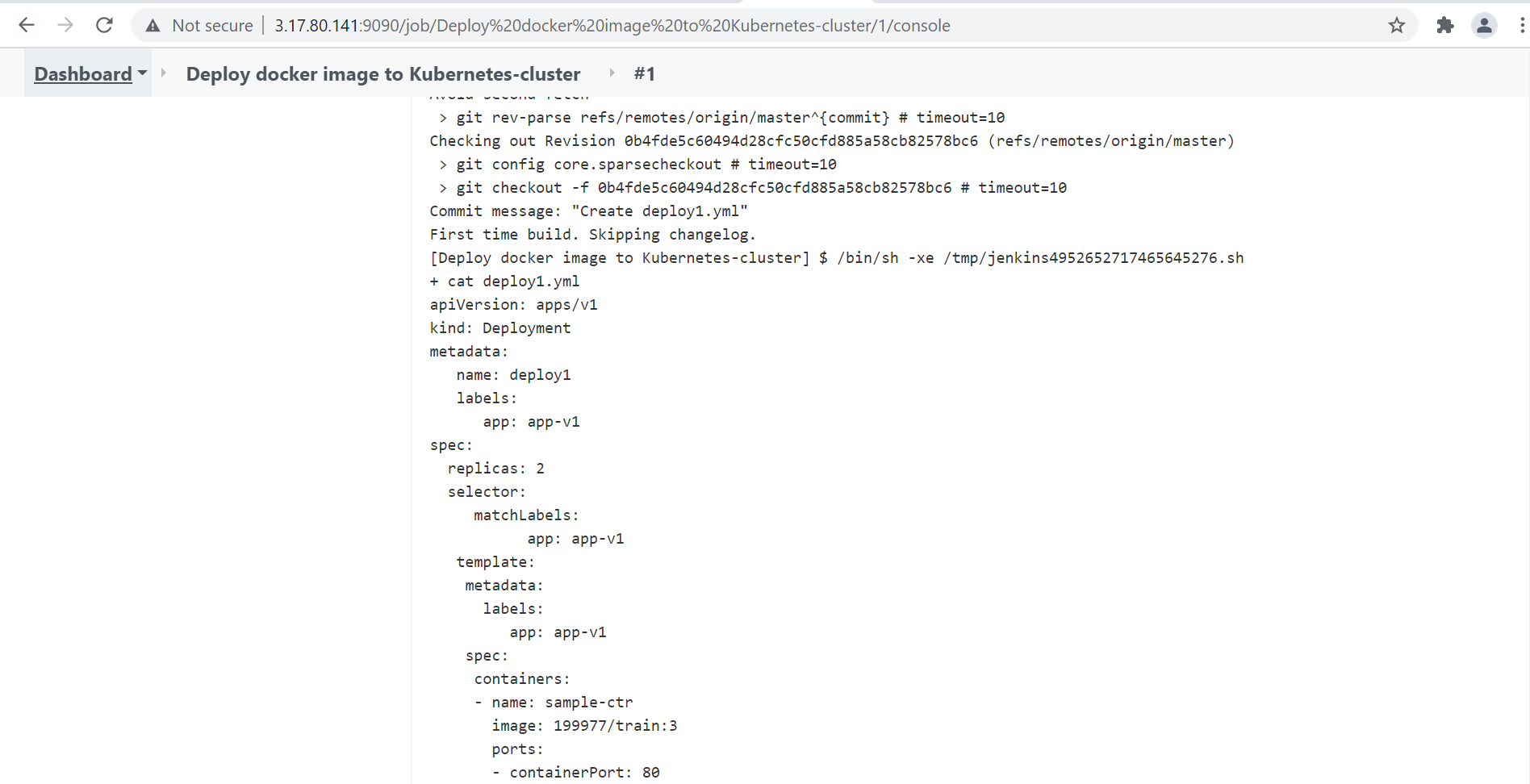


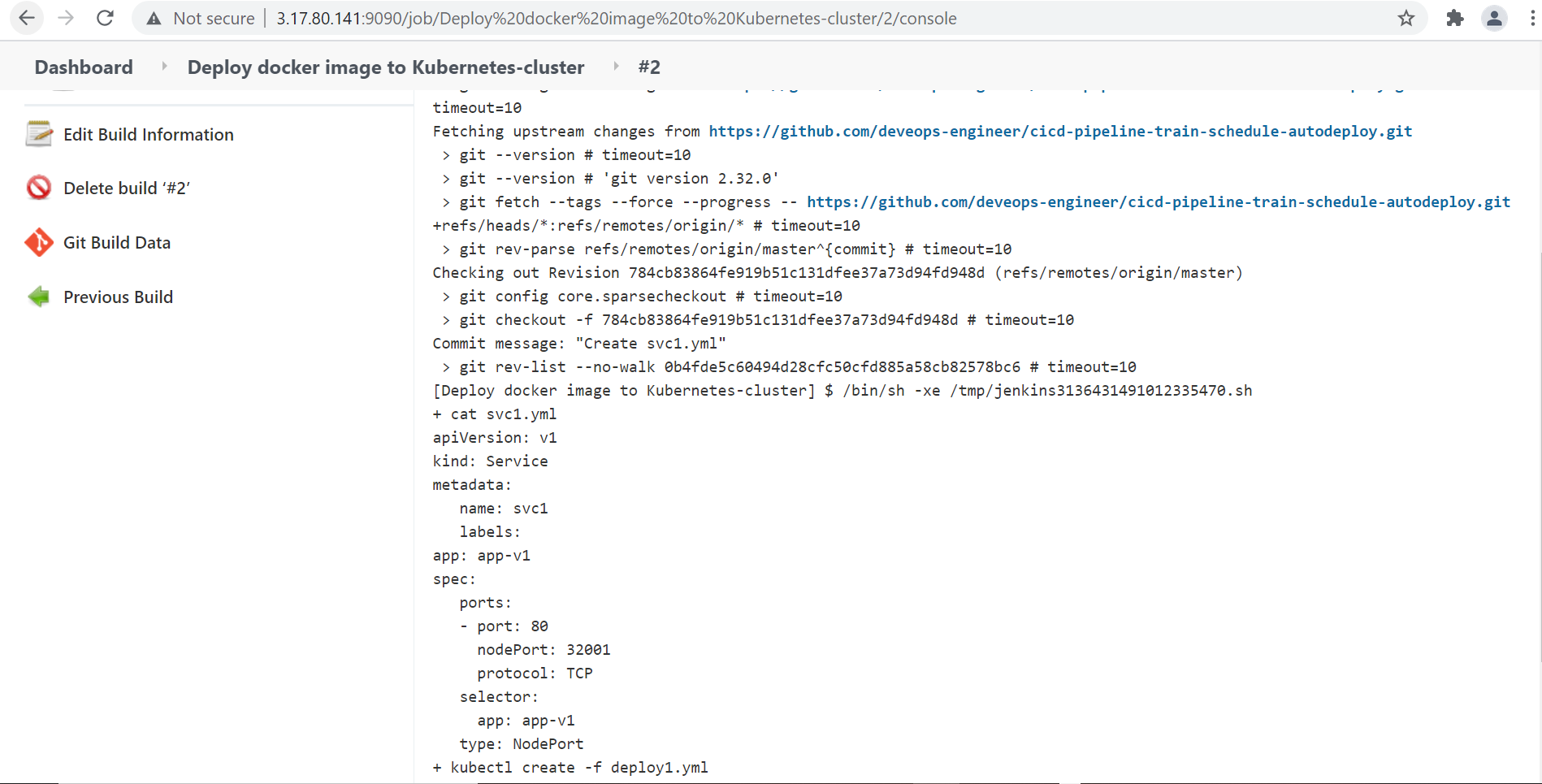
**Job 2** :- Deploy created docker image to kubernetes-cluster.



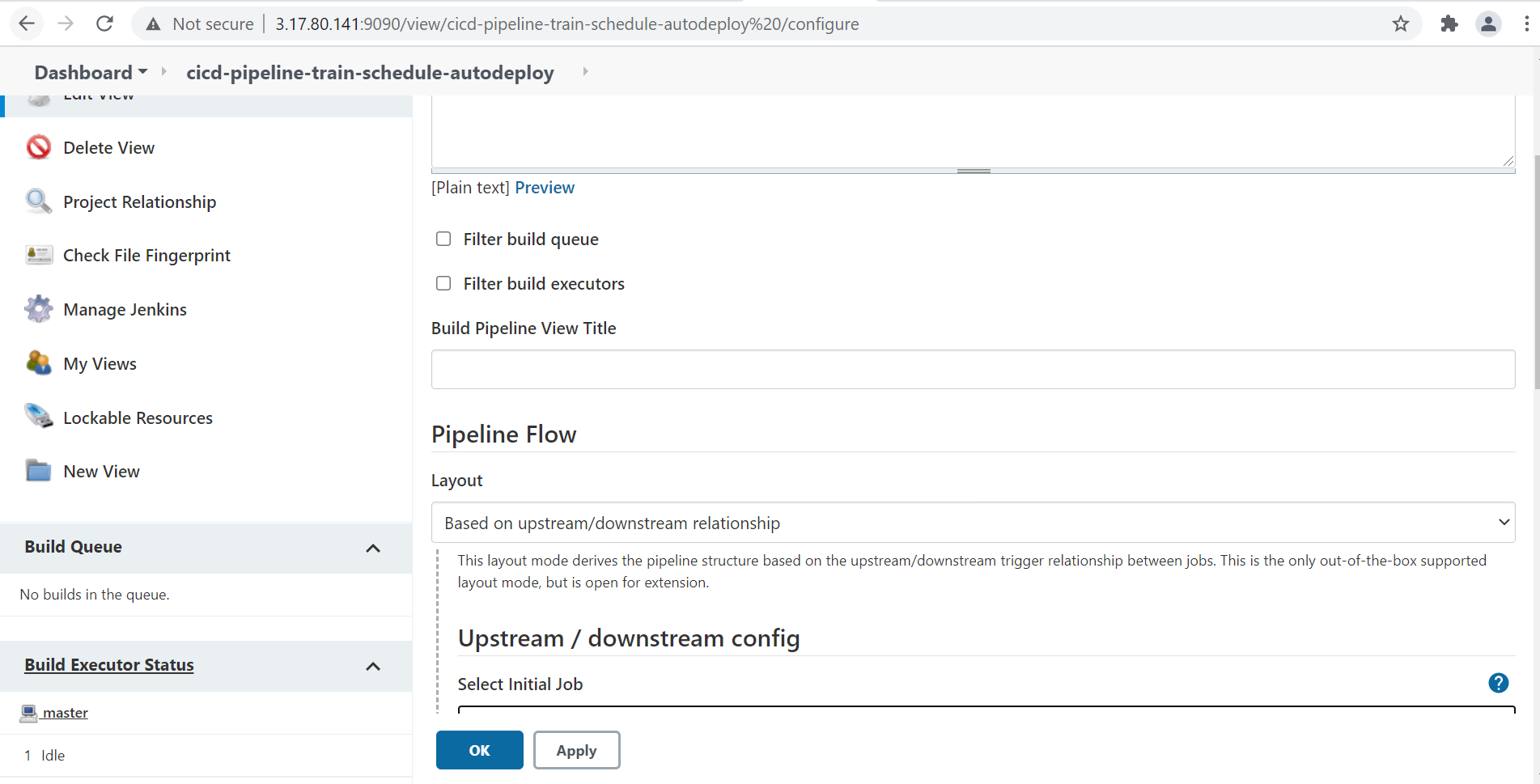


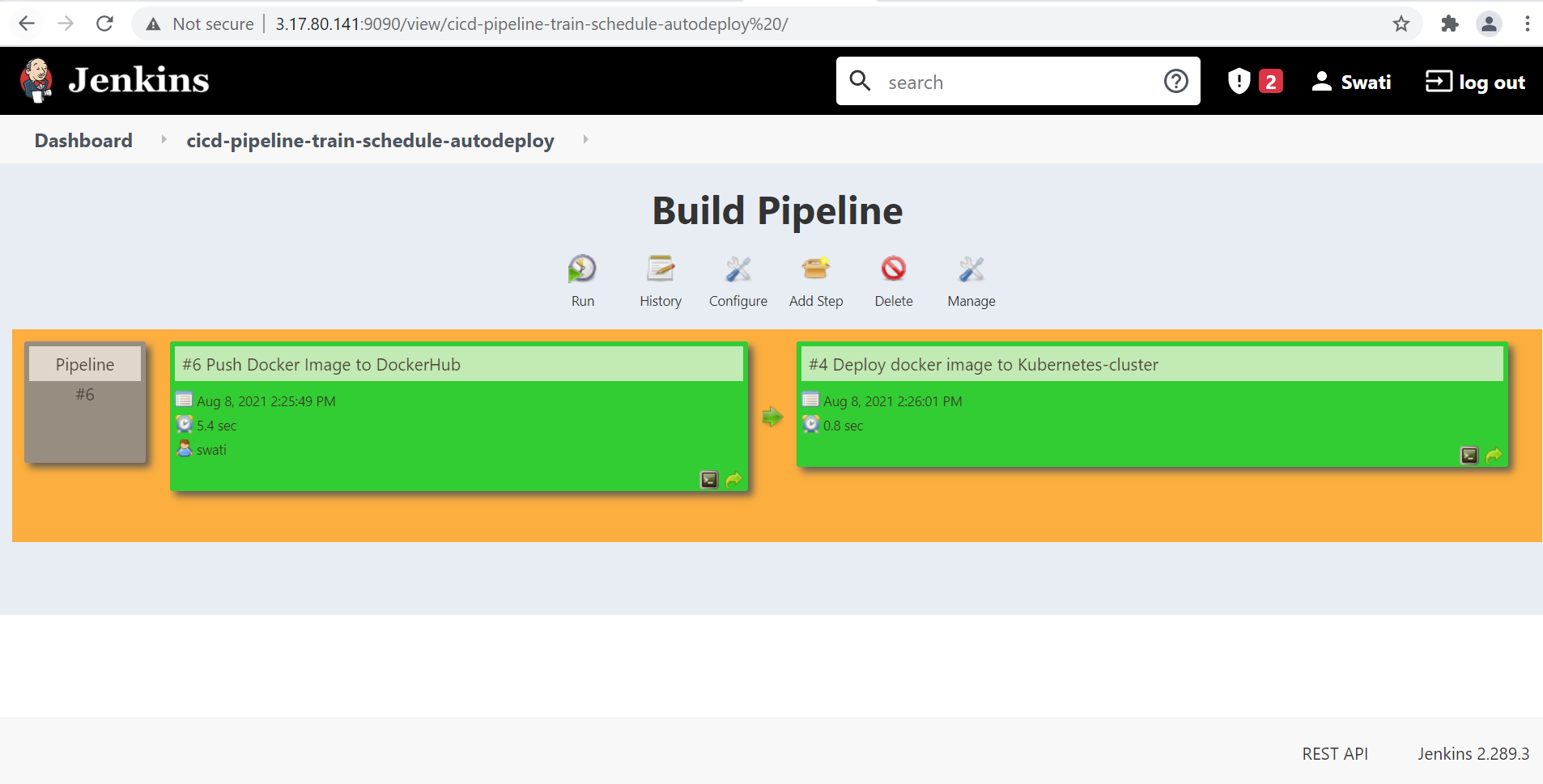






Make pipeline:





Deployment done :-

