### END TO END PROJECT

<https://github.com/akshu20791/cicdakshat>

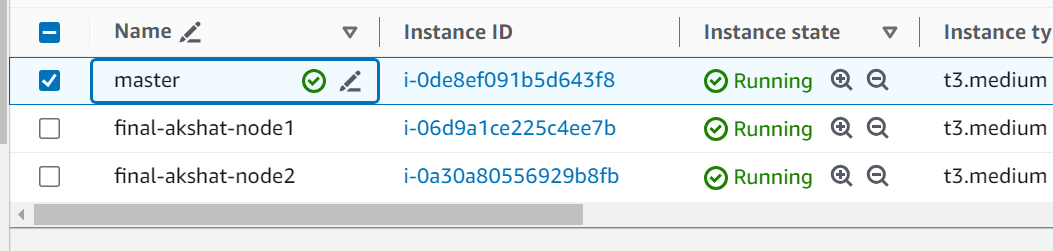
launch a 3 ubuntu machines

Ubuntu machines

Create key pair

Allow all traffic

T3.medium



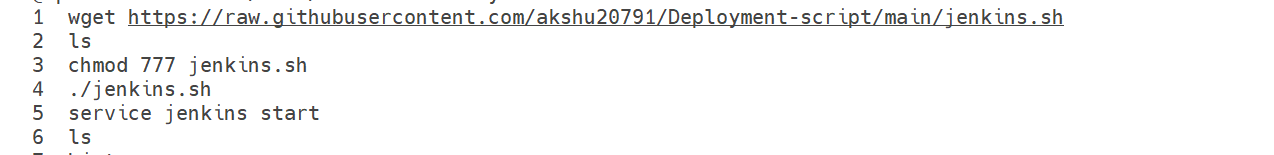
Login to each machine

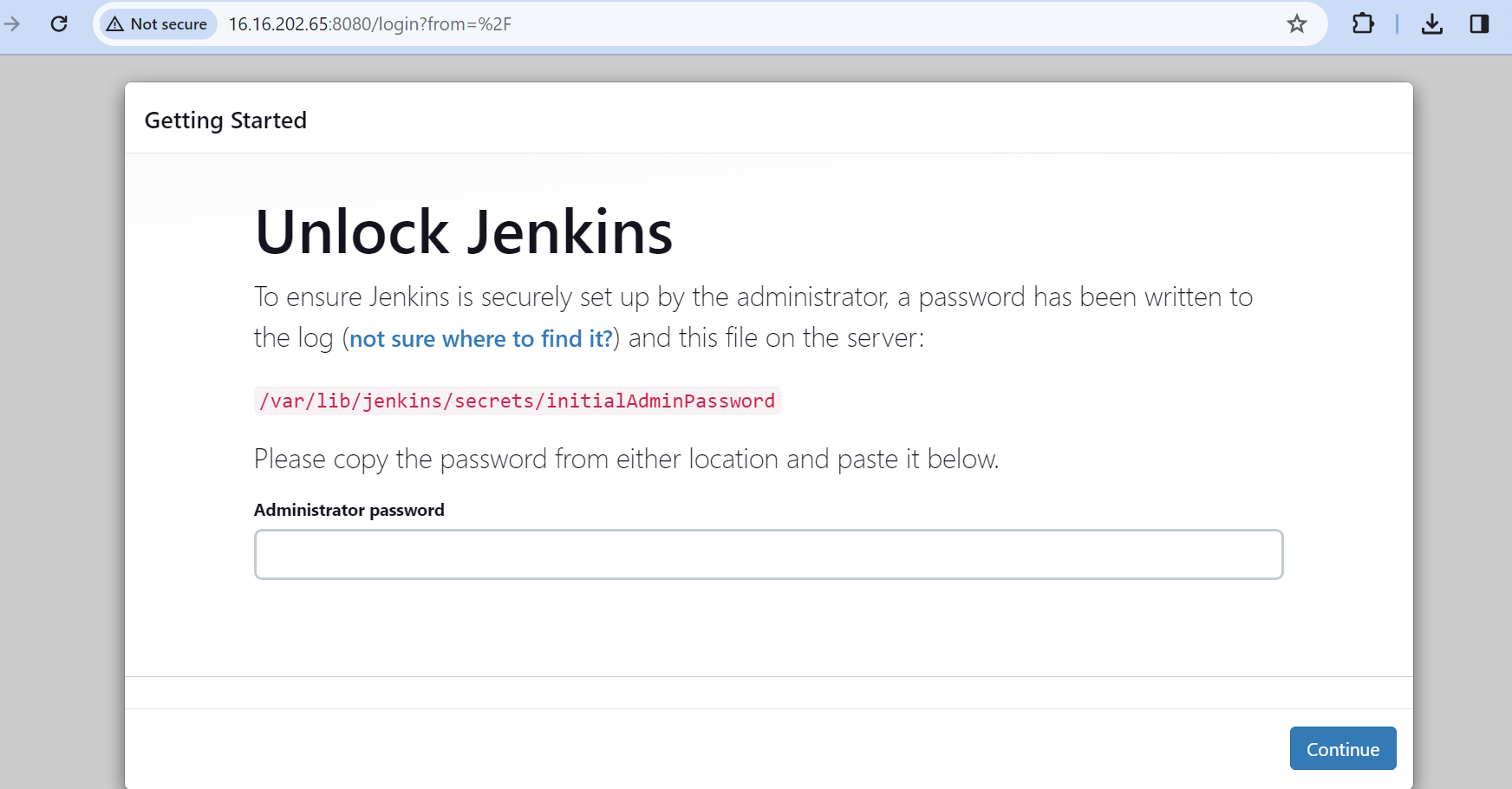
wget <https://raw.githubusercontent.com/akshu20791/Deployment-script/main/jenkins.sh>

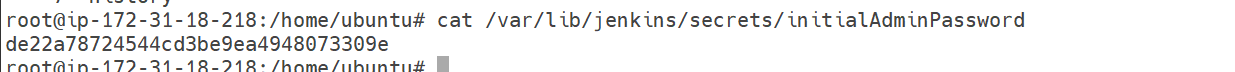
chmod 777 jenkins.sh

./jenkins.sh

service jenkins start



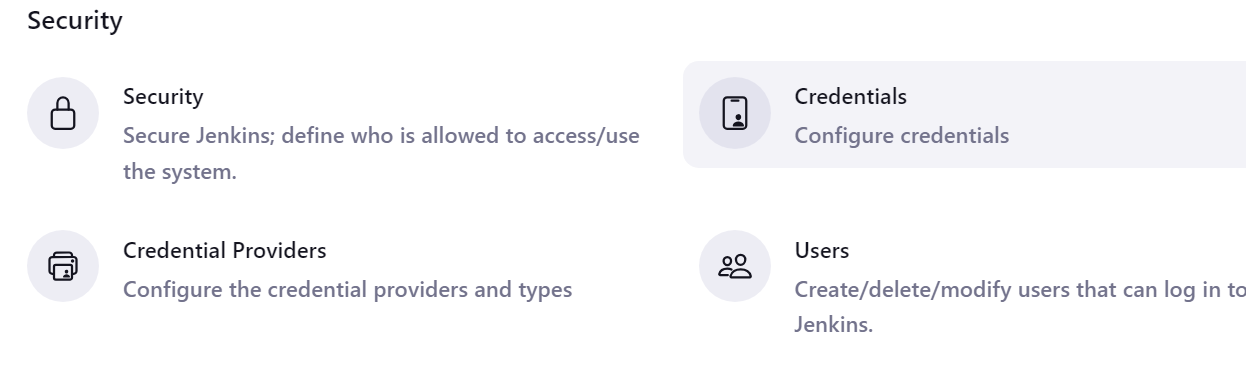
2) 

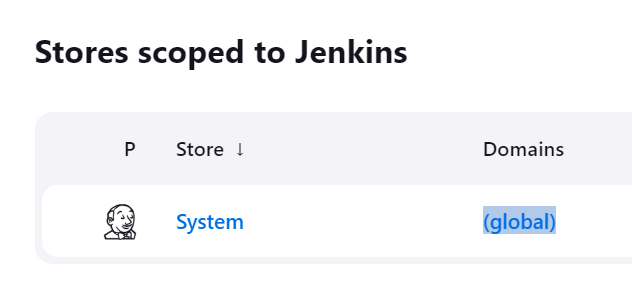


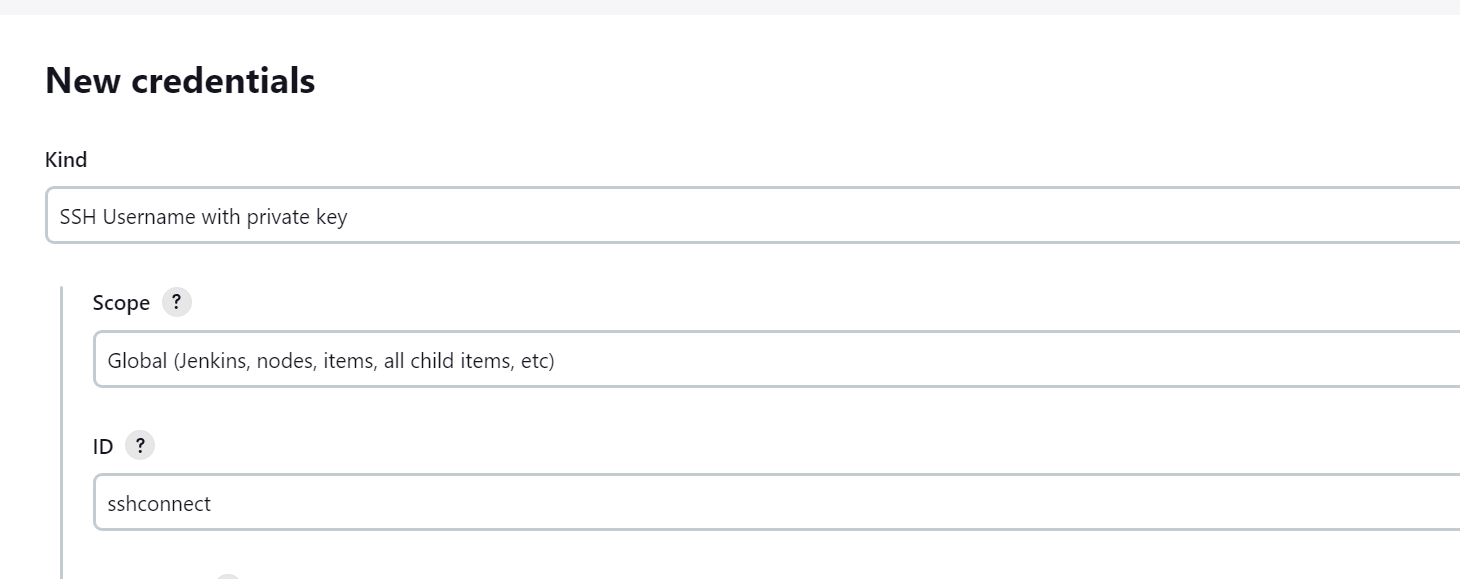
Copy paste this pass in the administrator pass

Get inside Jenkins

Go to manage Jenkins -> Credentials -> Click on global -> + add credentials

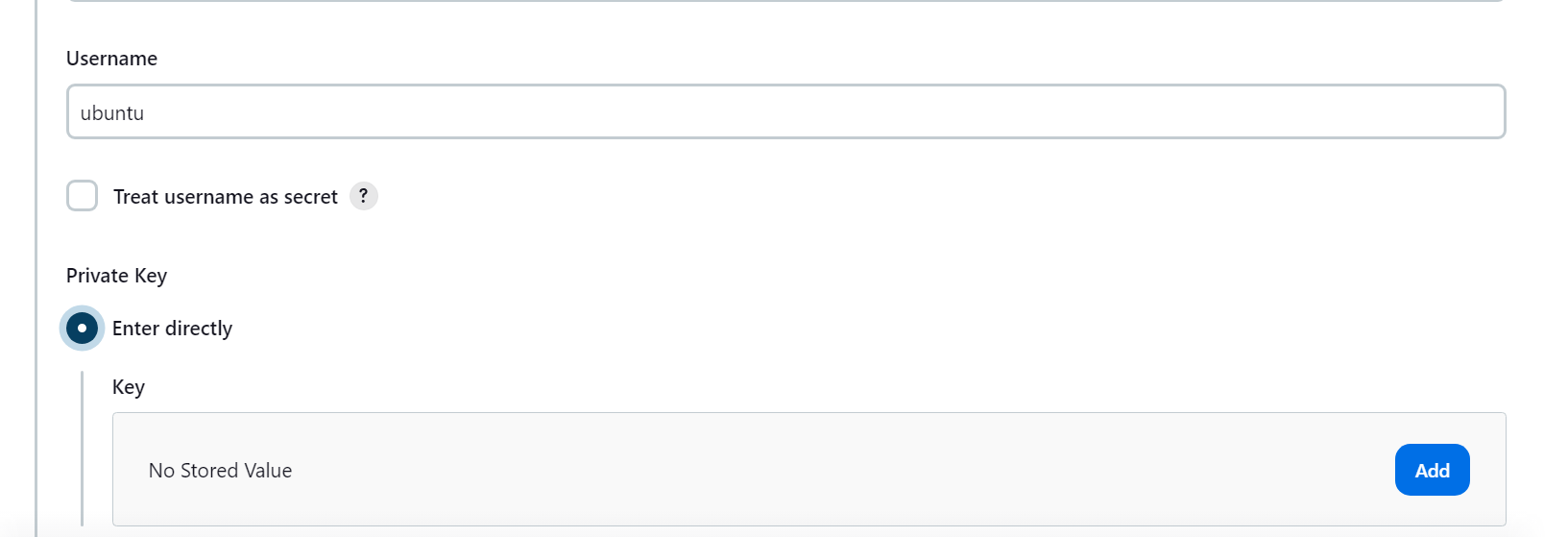




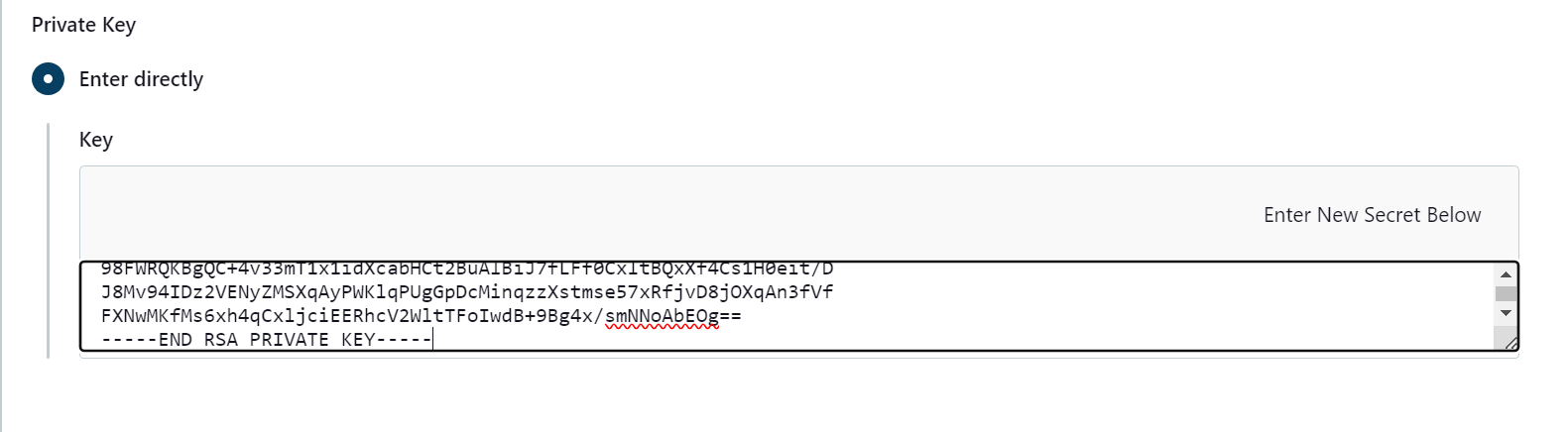


Username : ubuntu (if using linux then use ec2-user)

Click on add



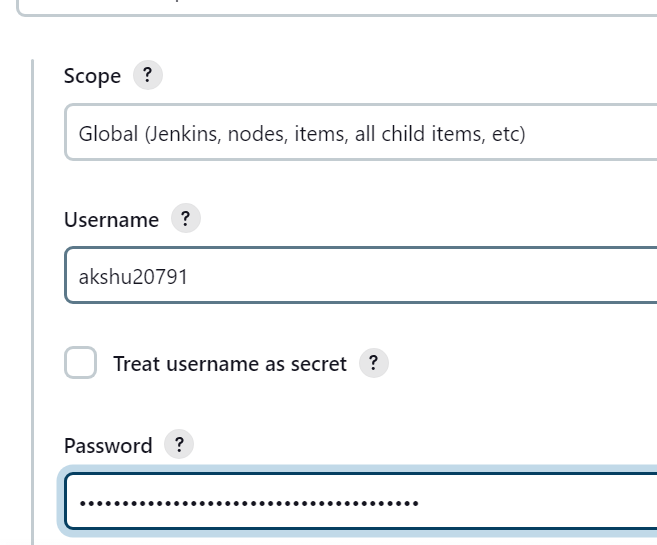
Copy the pem files of your nodes and paste it

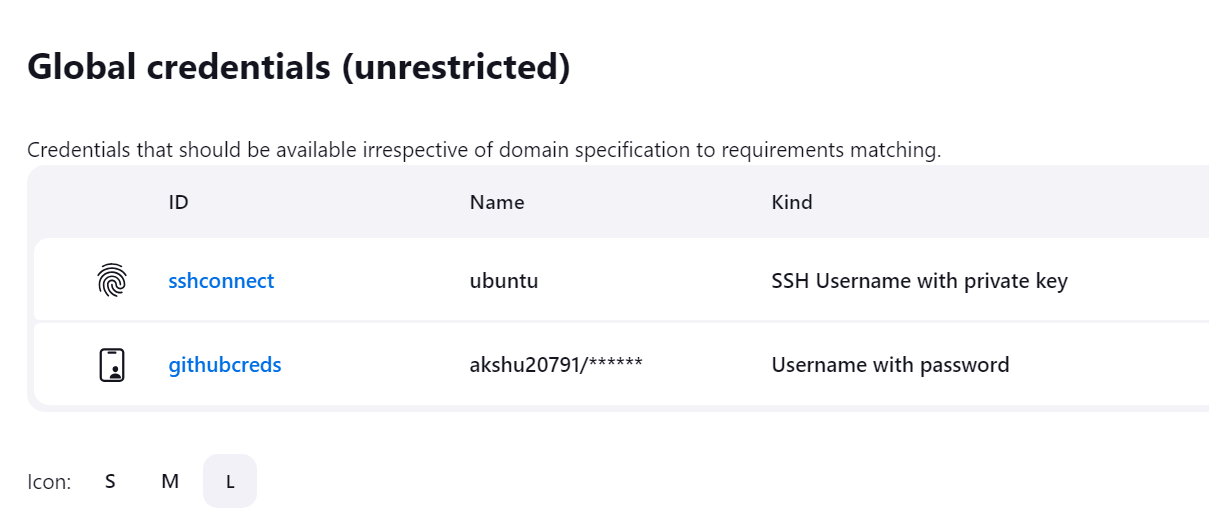


In case the github repo is public

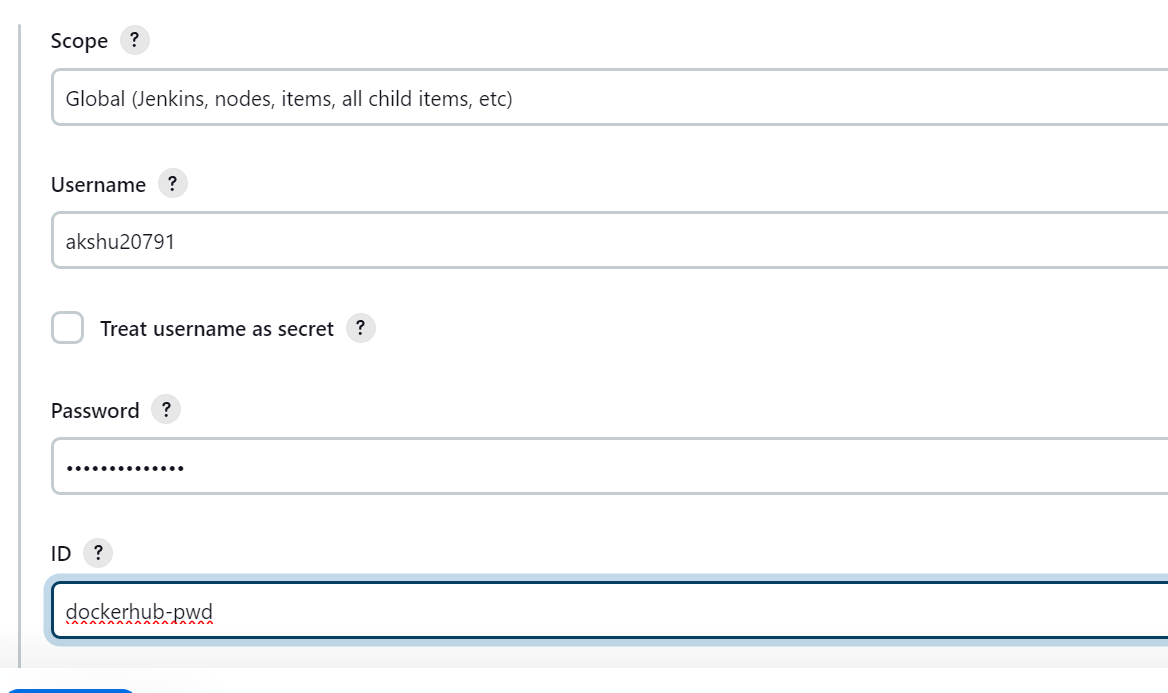


Use pat in password



save

Lets create docker credentials as well

Kind : username and pass

Put dockerhub username and pass

Now lets install kubernetes as well

IN master machine execute:

wget <https://raw.githubusercontent.com/akshu20791/Deployment-script/main/k8s-master.sh>

chmod 777 k8s-master.sh

./k8s-master.sh

####  
now lets install k8s in nodes

wget https://raw.githubusercontent.com/akshu20791/Deployment-script/main/k8s-nodes.sh

chmod 777 k8s-nodes.sh

./k8s-nodes.sh

#####

Go to master machine

## Run Below on Master Node to get join token

kubeadm token create --print-join-command

    copy the kubeadm join token from master & ensure to add --cri-socket unix:///var/run/cri-dockerd.sock as below & then run on worker nodes

kubeadm join 172.31.18.218:6443 --cri-socket unix:///var/run/cri-dockerd.sock --token ncomle.kuzlw14pr6shovt8 --discovery-token-ca-cert-hash sha256:6b32f3cd891f804ced97d64eefe9891f20db885a2796b7ba0b6a107d5efb486c

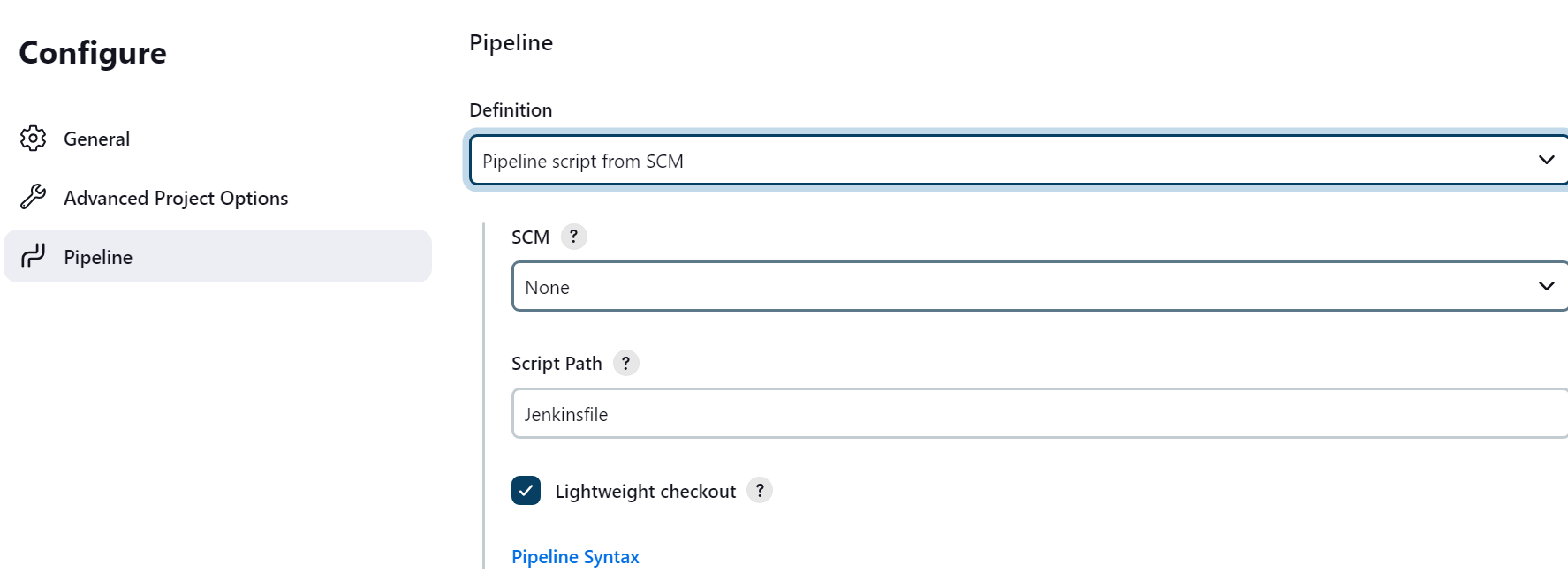
    Ex: kubeadm join 10.128.15.231:6443 --cri-socket unix:///var/run/cri-dockerd.sock --token mks3y2.v03tyyru0gy12mbt \

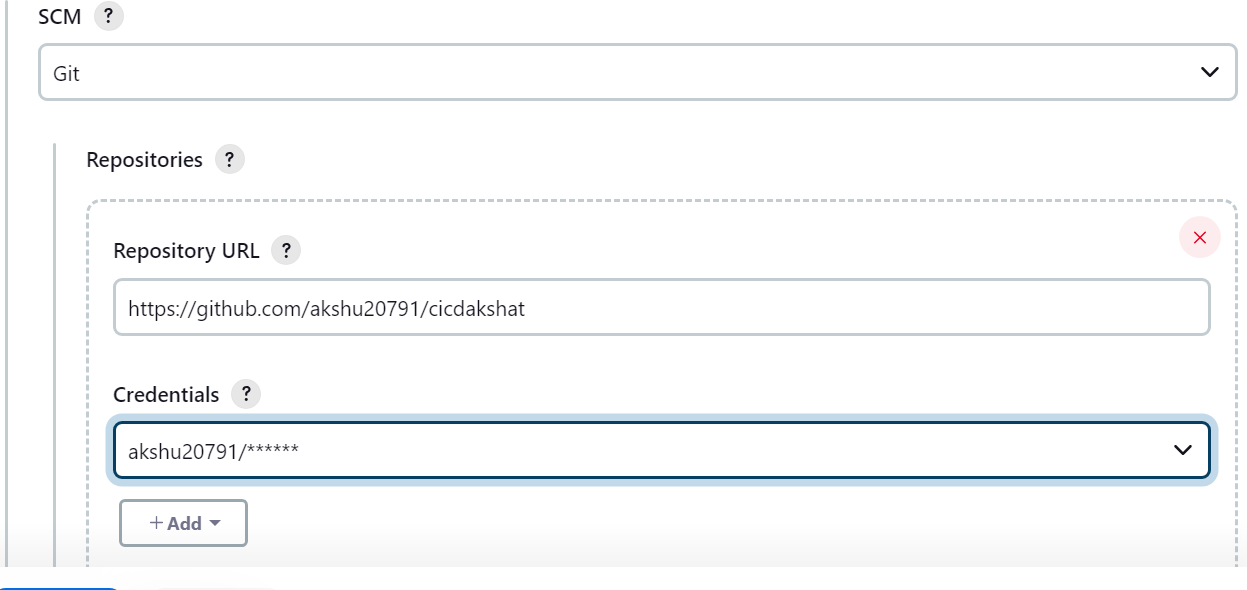
           --discovery-token-ca-cert-hash sha256:3de23d42c7002be0893339fbe558ee75e14399e11f22e3f0b34351077b7c4b56

Now lets create pipeline

Click on new item

Select pipeline





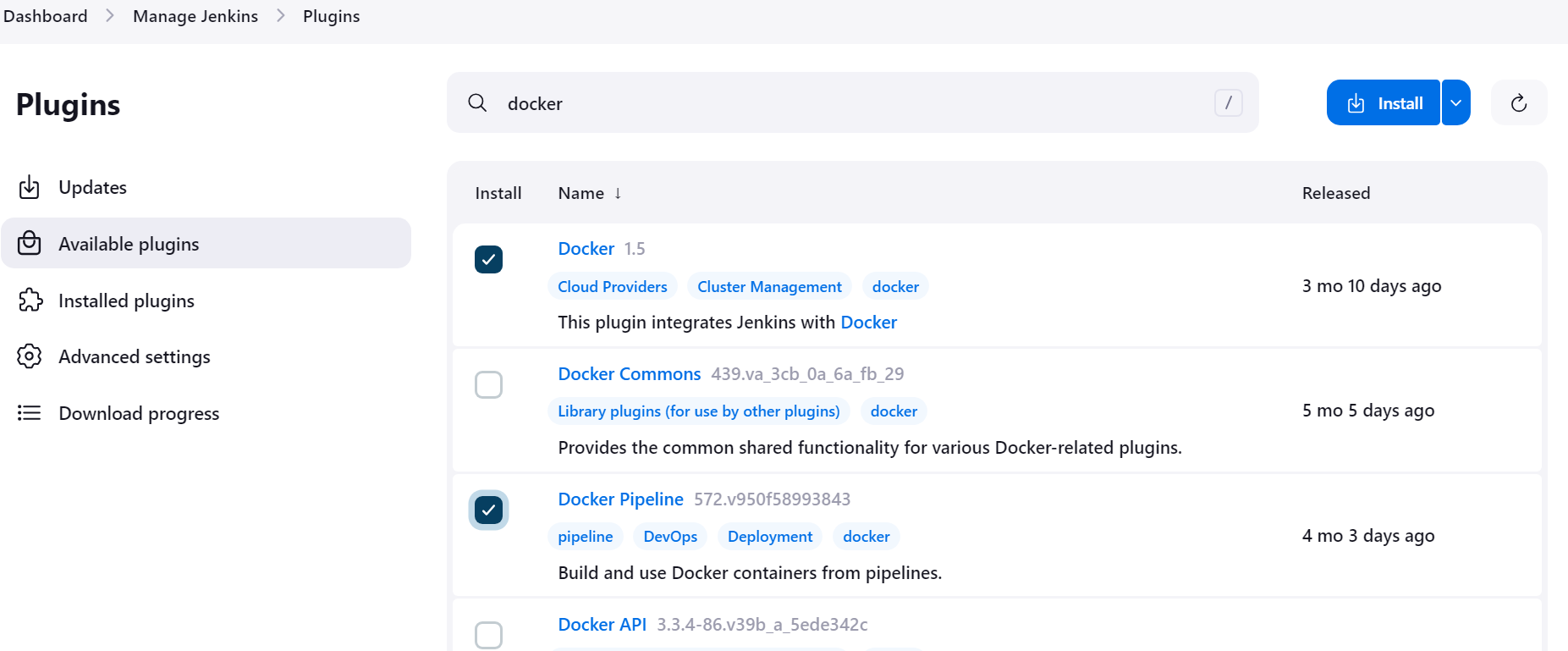
Save

Go to github.com

Update Jenkinsfile as per your requirement

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

Go to Jenkins -> install plugins

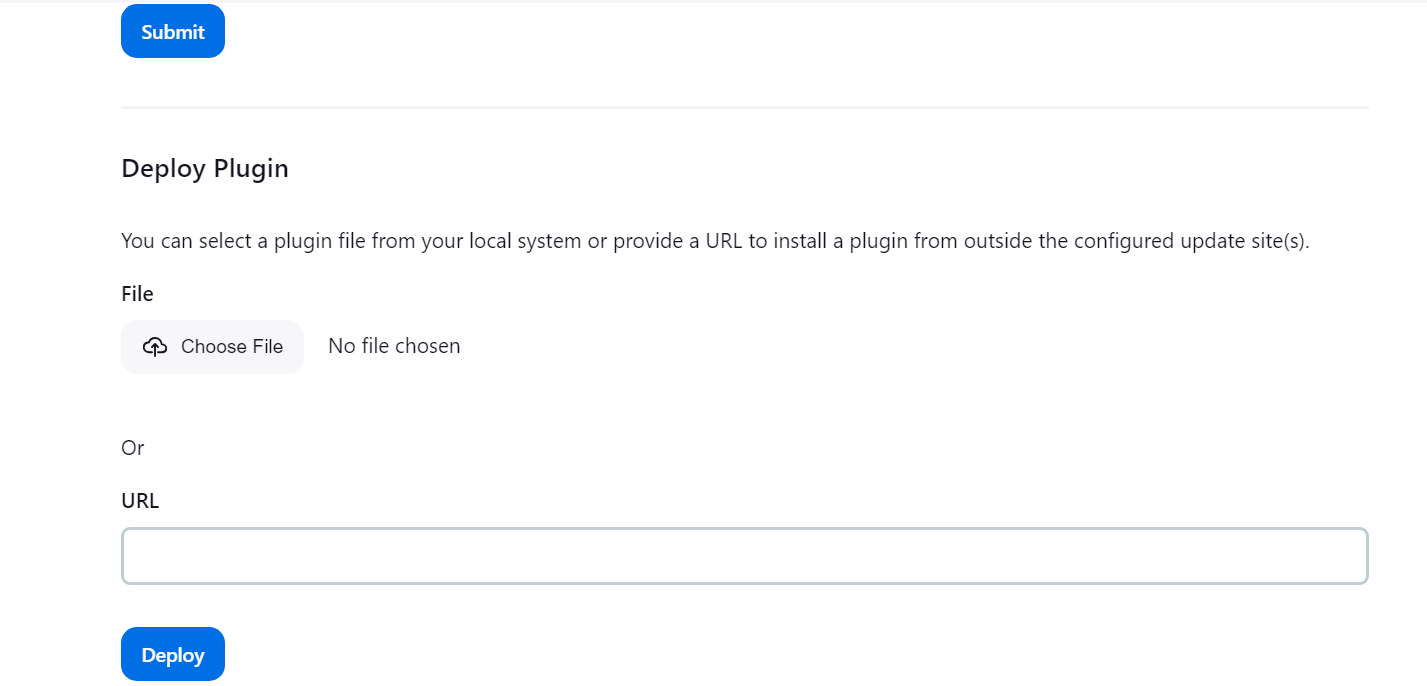


We will also install k8s plugin

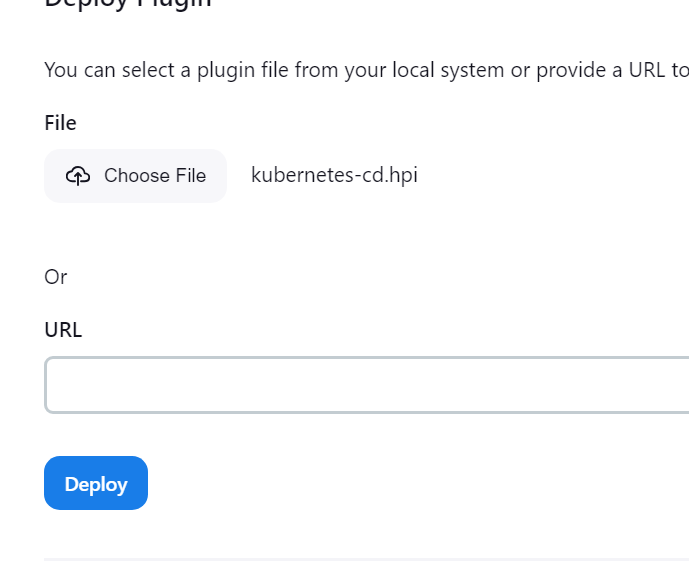
Download the file

<https://github.com/akshu20791/cicdakshat/blob/master/kubernetes-cd.hpi>

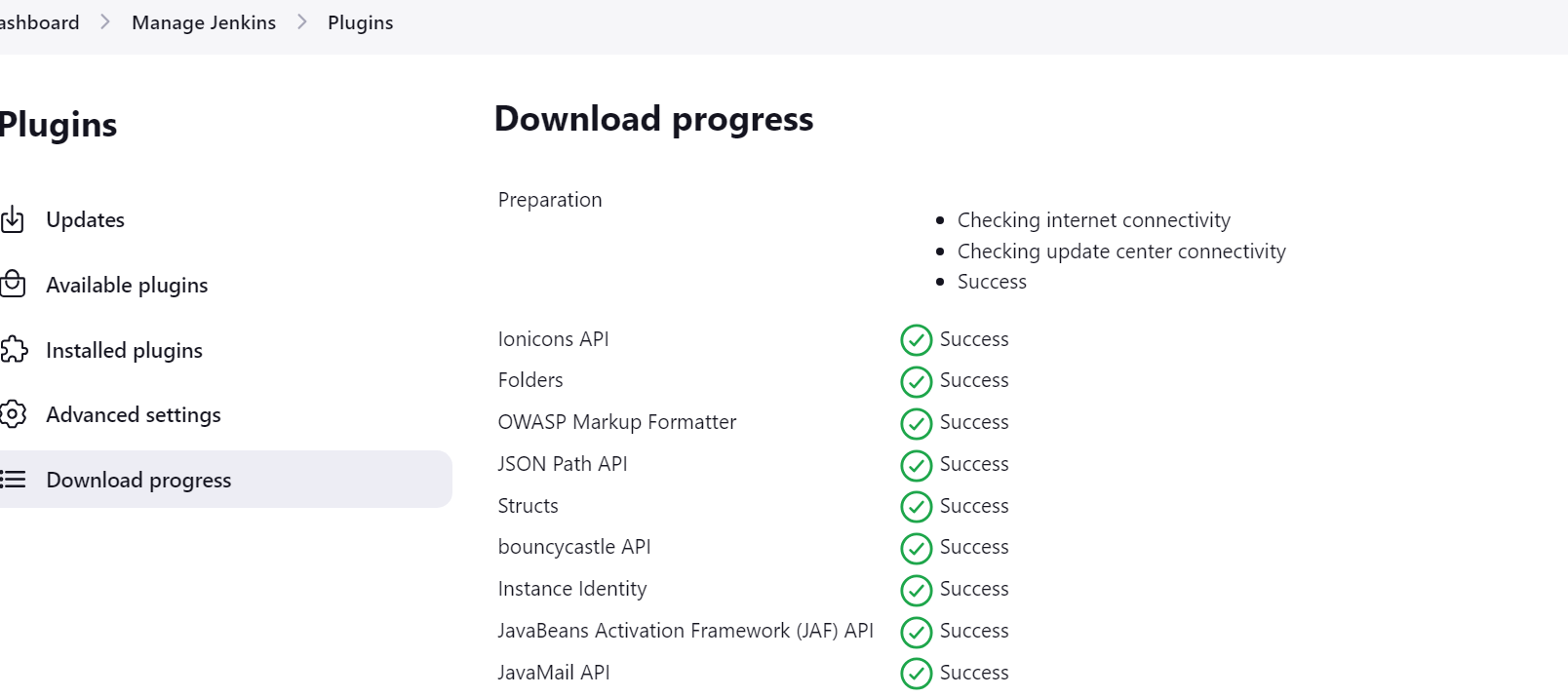
in Jenkins -> manage plugins -> plugins -> advance settings



Upload the plugin downloaded

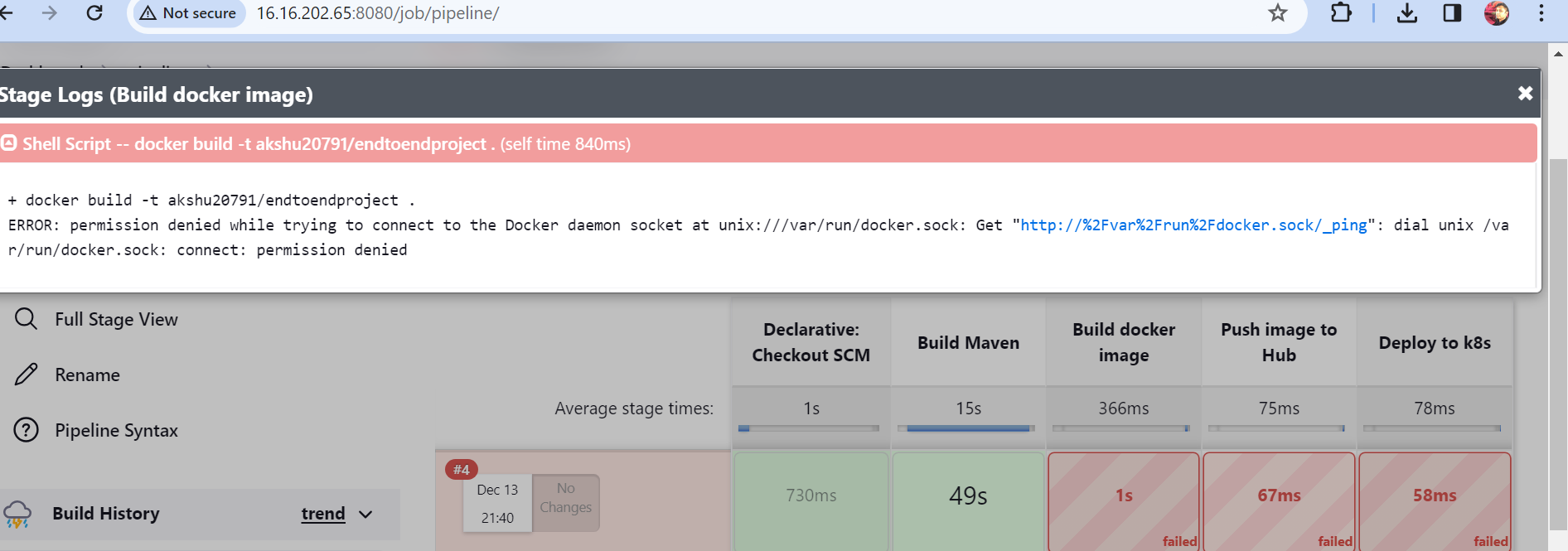


Deploy



Go back to Jenkins pipeline

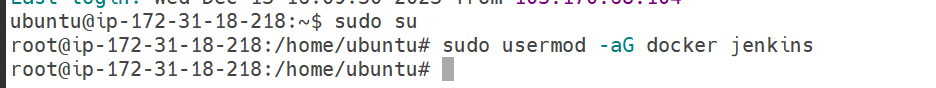
When you execute you wil get docker permission issues



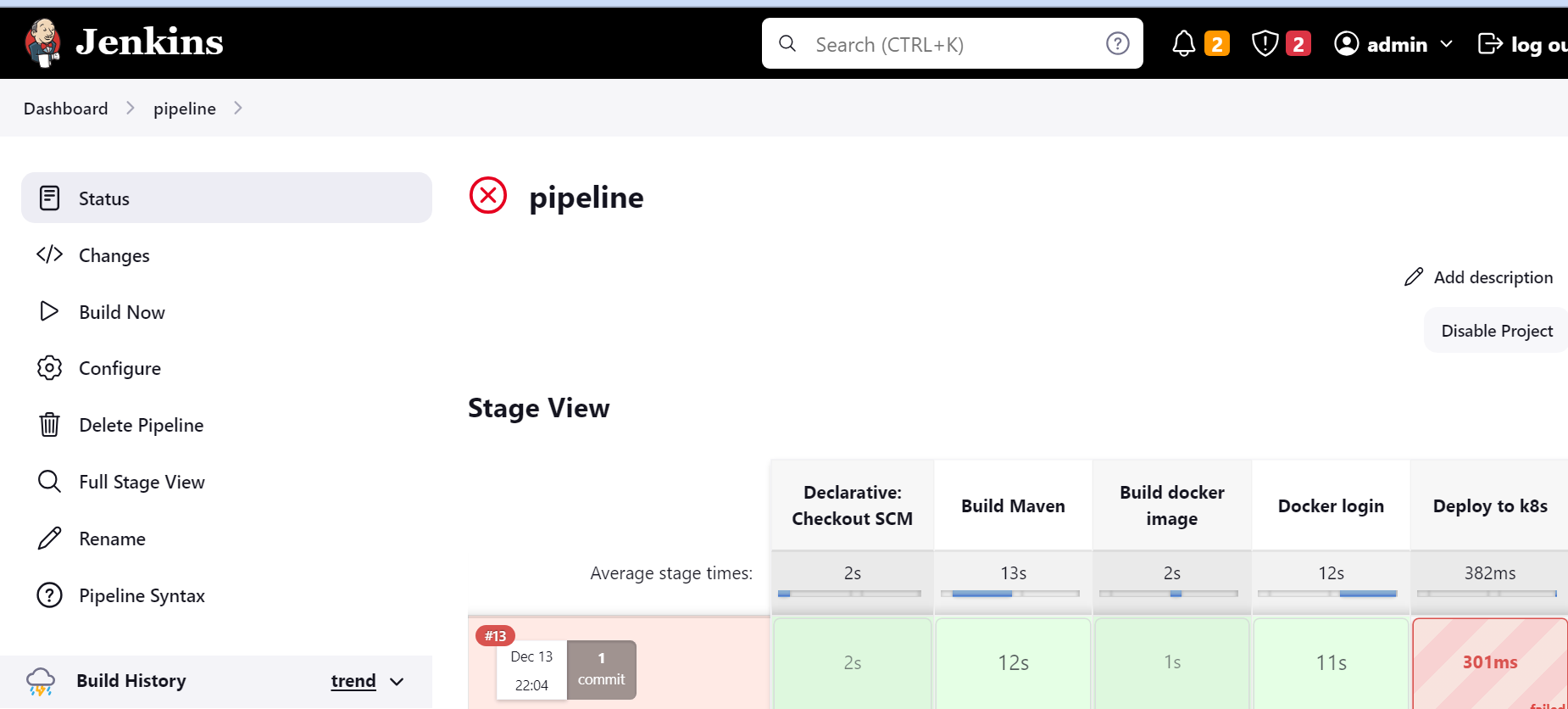
Go to machine and execute:

sudo usermod -aG docker jenkins

service jenkins restart (impotant step)



Now you will see it will build and push the image to docker hub

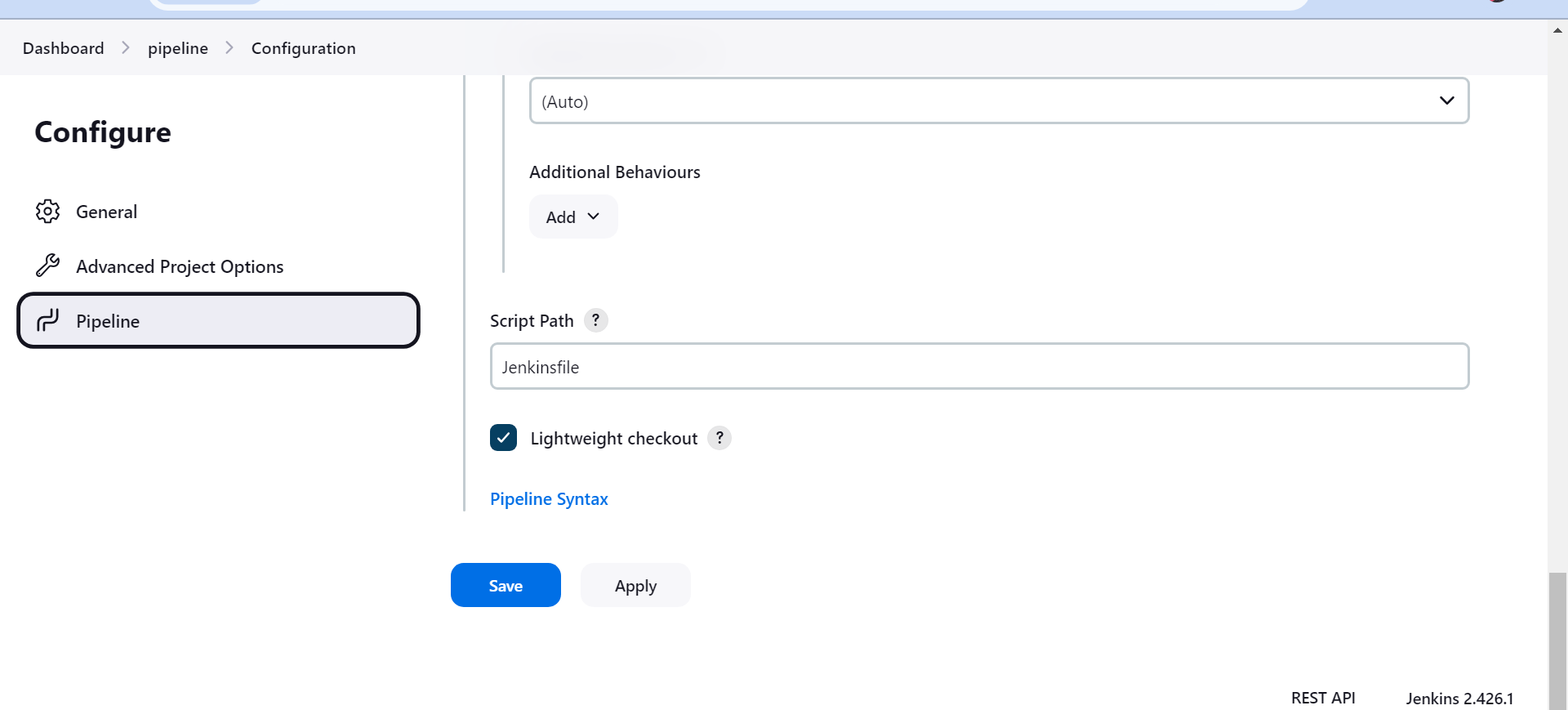


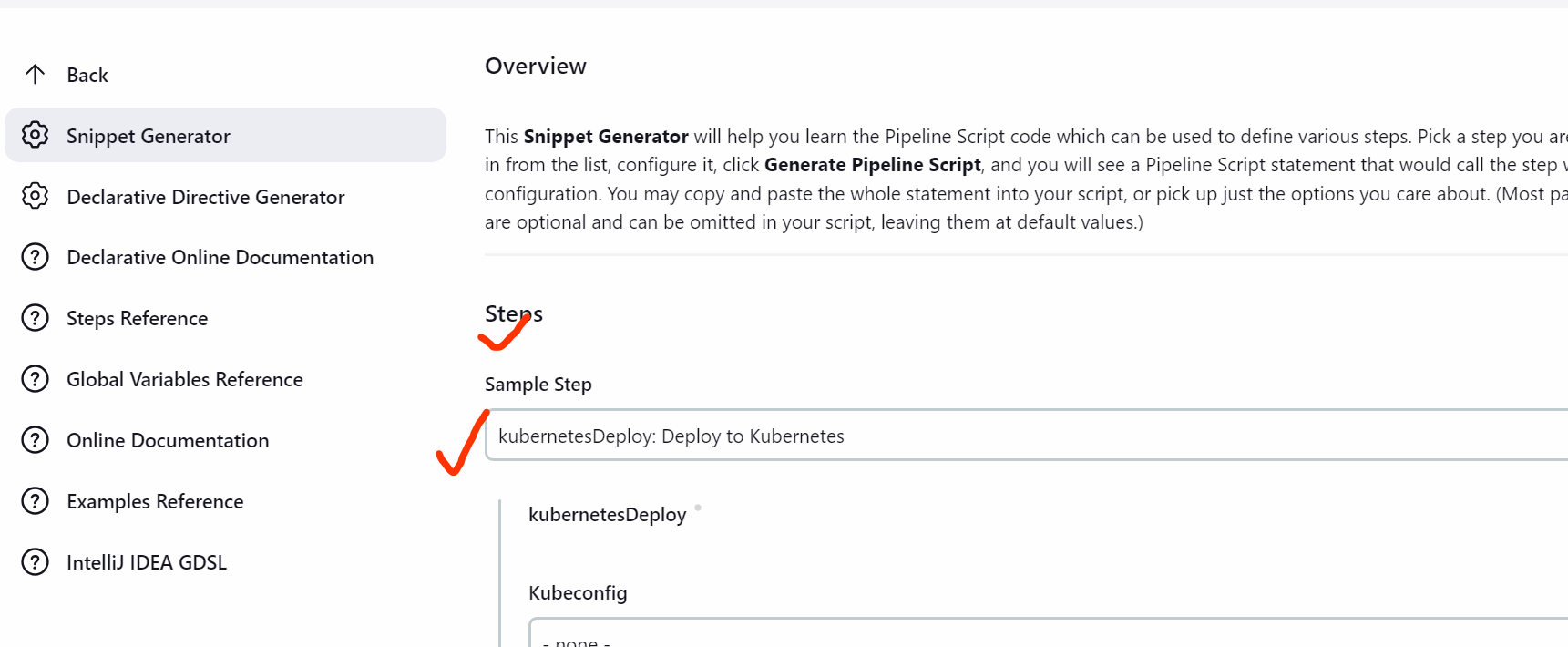
Lets configure kubeconfig

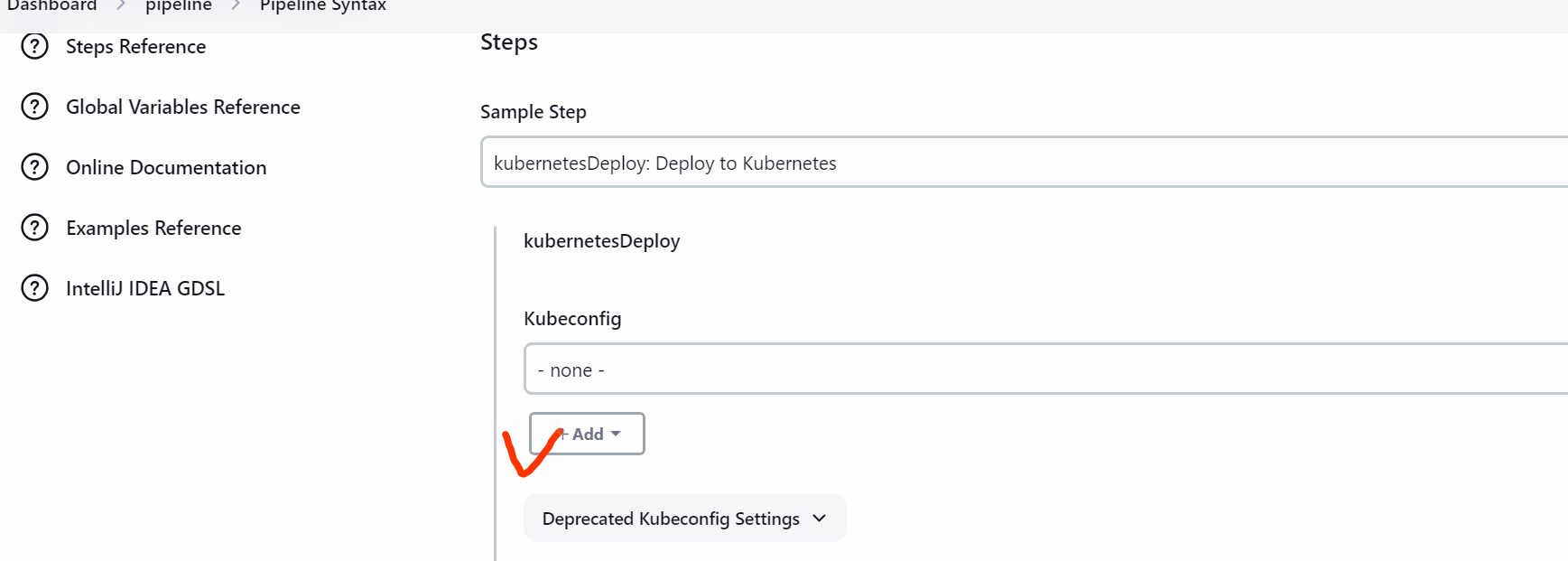
We need to generate the code

Go to your pipeline

Scroll down and click on pipeline syntax







Kind: Kubernetes configuration



Go to master

cd ~

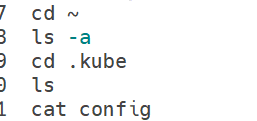
ls -la

cd .kube

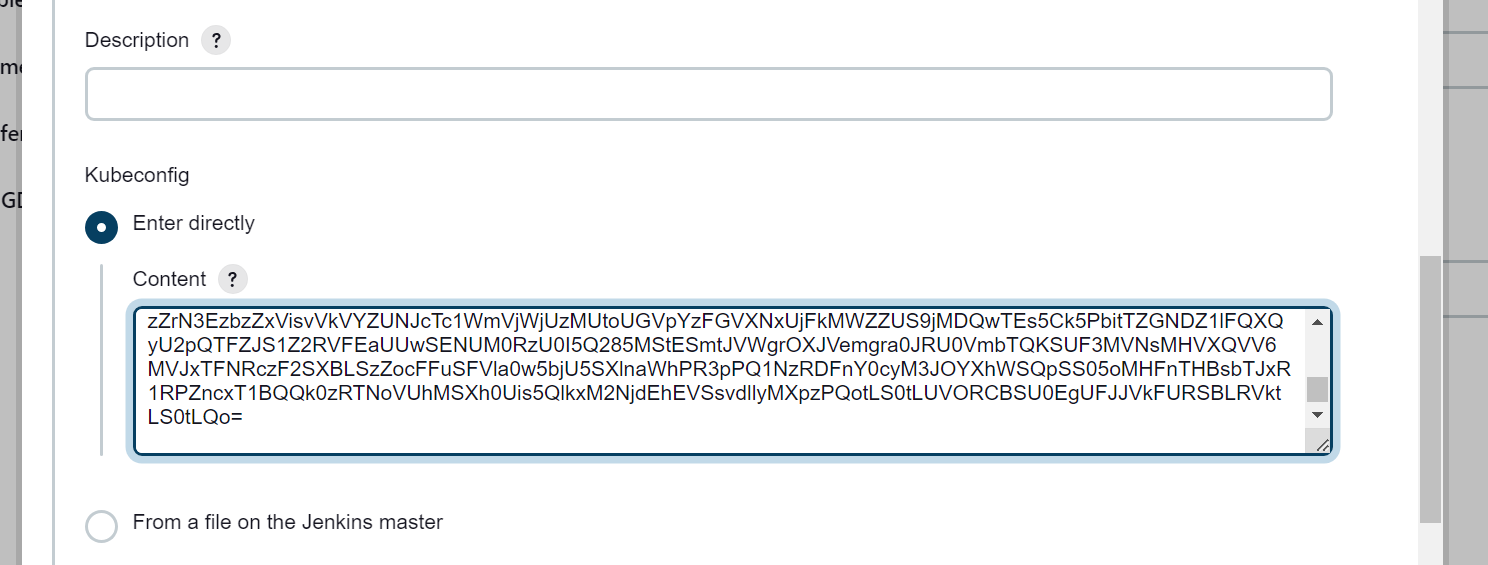
ls

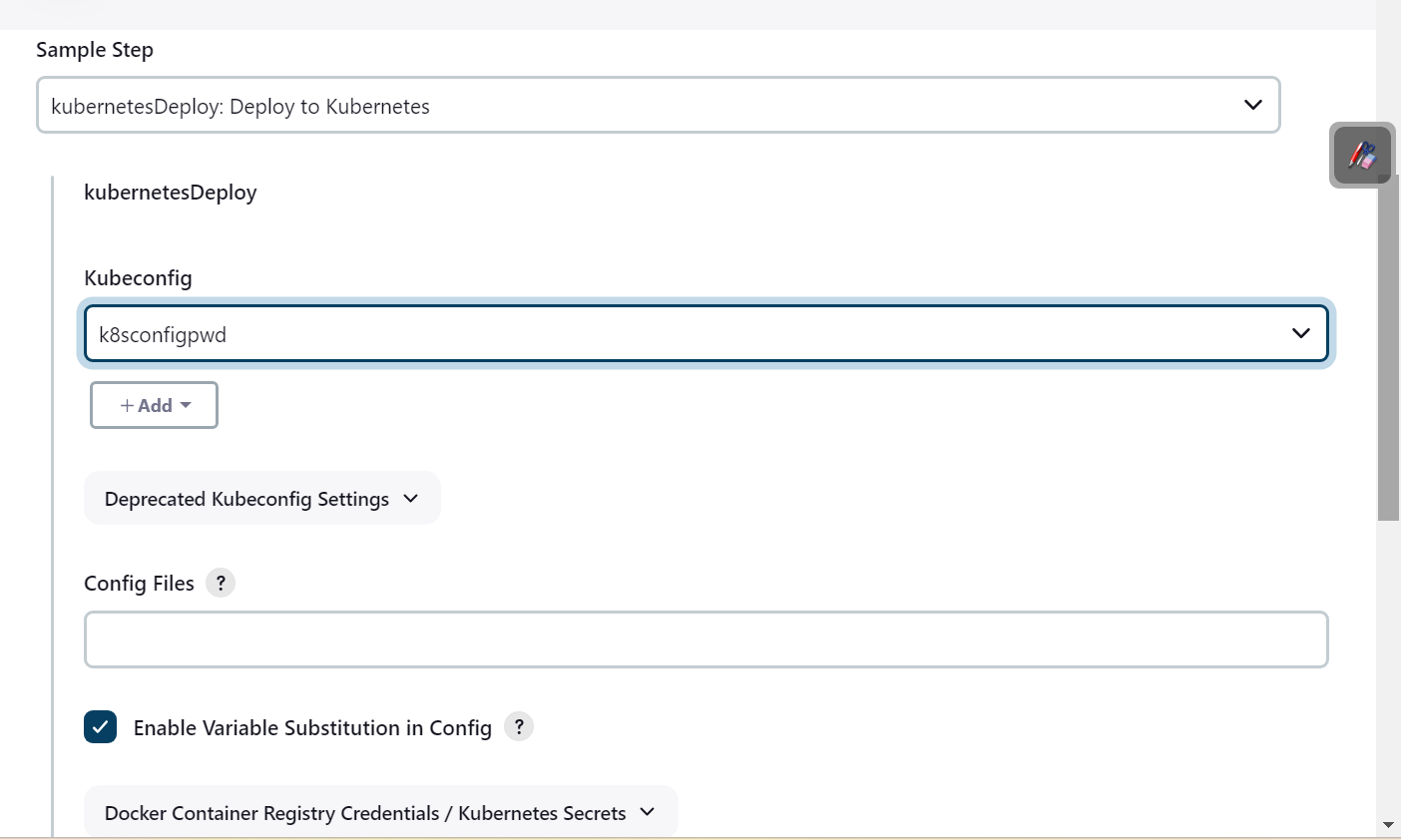
cat config

copy the o/p of this file and paste it in Jenkins

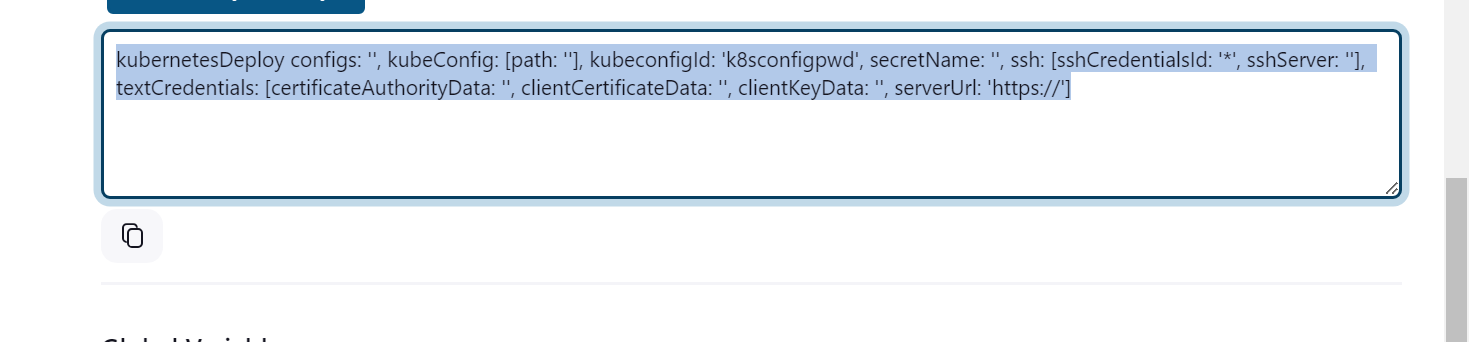


Paste here





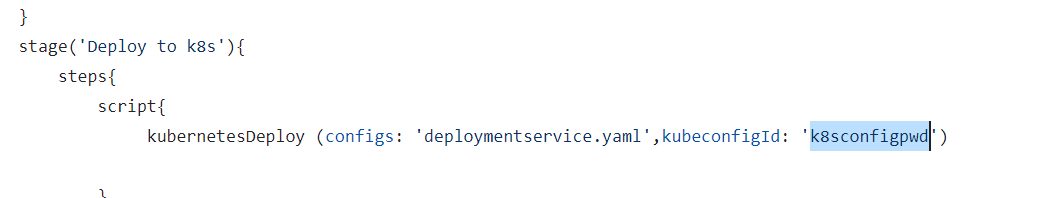
Generate pipeline script



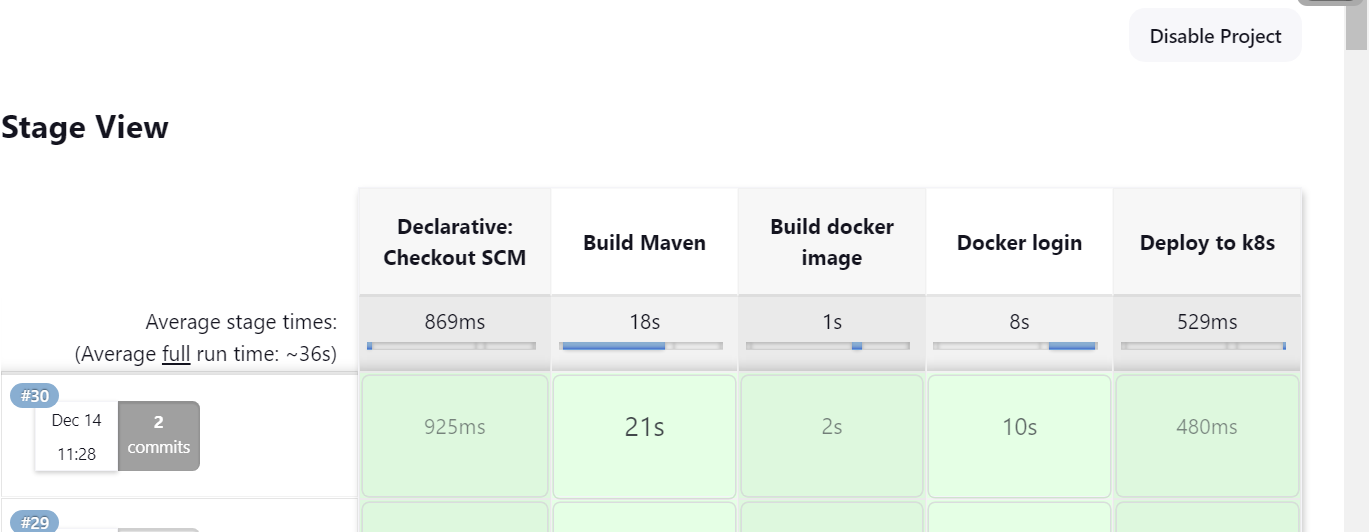
Now you can remove the not required content

You can also remove this kubeConfig: [path: ]

Update in Jenkins file



##excute the pipeline



Now

Go back to master:

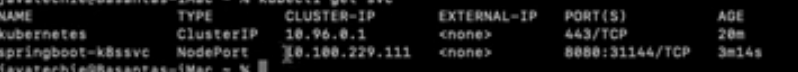
sudo su

kubectl get pods

copy one pod

kubectl logs <pod name>

kubectl get svc



Copy the port number

And cop public ip :port (in our case publicip:31144)