**CICD pipeline using jenkins to deploy kubernetes.**

1. Install docker

2. Install kubernetes cluster/Minikube/kubectl

3. Install Java

*sudo apt update*  
*sudo apt install openjdk-8-jdk*

4. Install jenkins

5. install ngrok

sudo snap install ngrok

6. Add your user to the 'docker' group:

sudo usermod -aG docker $USER && newgrp docker

7. Add jenkins to 'docker' group

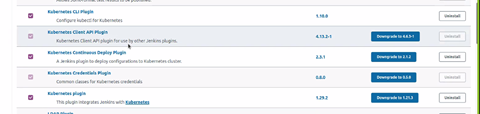
sudo usermod -aG docker jenkins

8. Open jenkins on localhost:8080

9. Go to manage jenkins->manage plugins & install following plugins ,

* Kubernetes continuous deployment
* Kubernetes Cli plugin
* Kubernetes
* Docker Pipeline

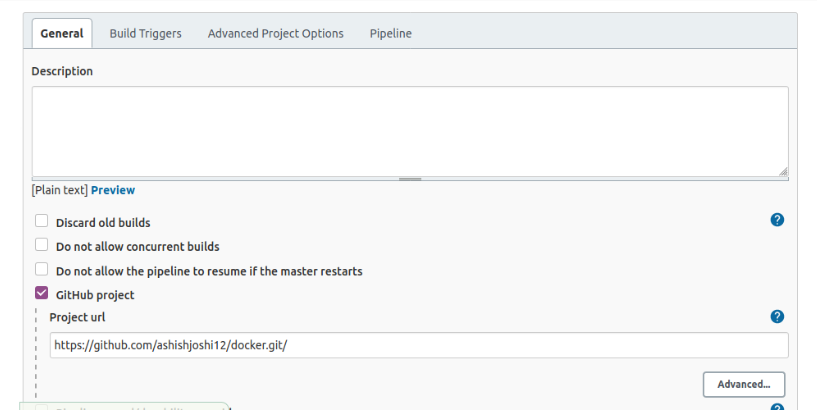
Click on install without restart.

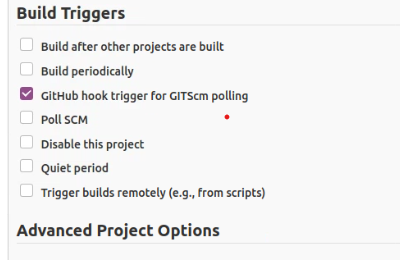


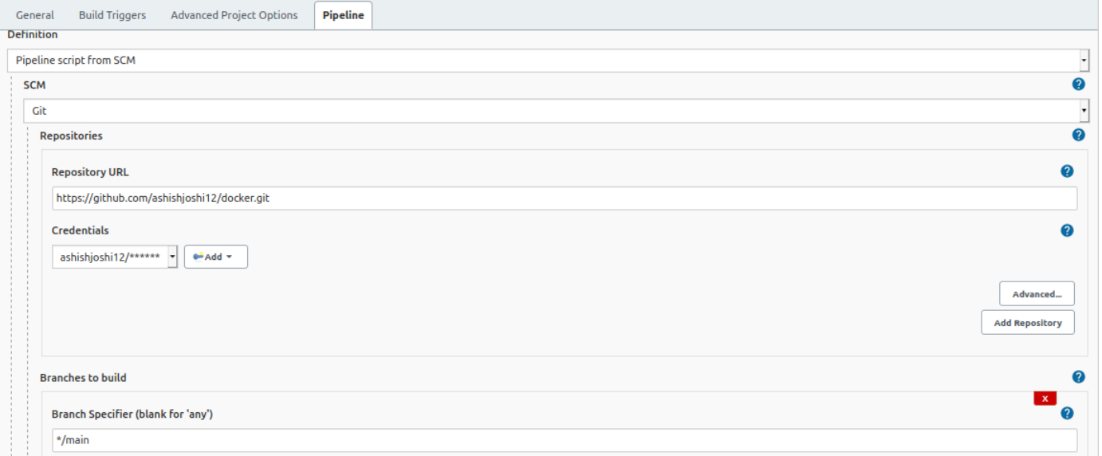
10. Create new pipeline as Demo1 and setup github scm polling,

* Click on New Item-> Select Pipeline Project -> OK
* General -> GitHub project -> Add your repository url (https://github.com/ashishjoshi12/docker.git)
* Build Trigger -> Check GitHub hook trigger for GITScm polling
* Pipeline -> Definition -> Pipeline Script From SCM
* SCM -> GIT -> Repository Url (https://github.com/ashishjoshi12/docker.git) (If private you will need to specify github creds.)
* Branch - \*/Master (or \*/main if git repository has been created recently)
* Script -> Jenkinsfile
* Uncheck Lightweight checkout
* Click on save

Make sure your repo contains dockerfile , Jenkinsfile and kubedeployment yml.





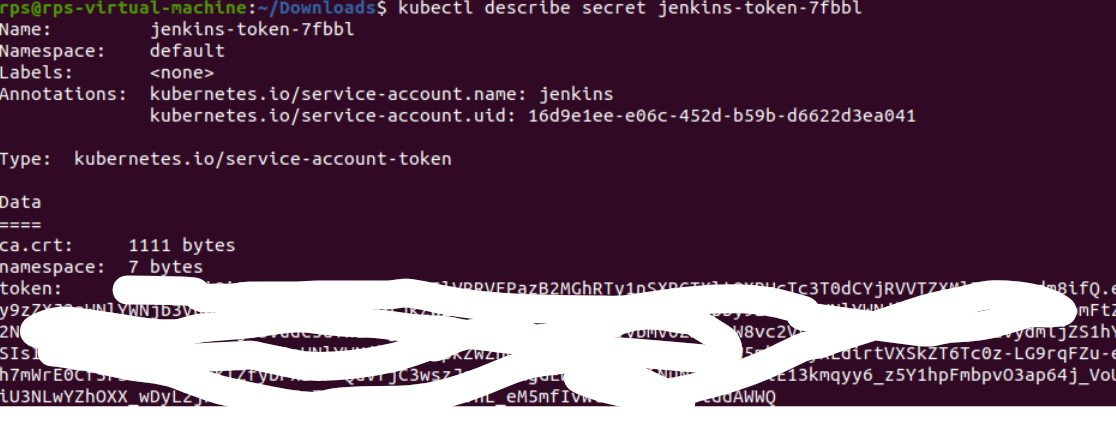


11. create jenkins service account

kubectl create serviceaccount jenkins

kubectl get secret

kubectl describe secret <secreteName>

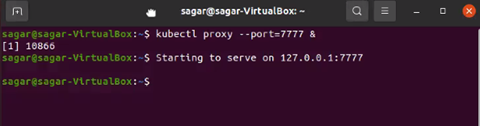


copy secrete text and create credential in jenkins->Jenkins(system)-> Global Creds->add credentials->kind(secrete text) ->paste secret text and give id as 'mysecret'



12. In order to use withKubeCredential OR kubeConfig cli in Jenkindfile to deploy yml file we need to access kubernetes server api using kube proxy,

kubectl proxy --port=7777 &



13. GO to jenkins after plugin installation. manage jenkins -> manage Nodes and Clouds -> configure cloud -> select kubernetes in list. ((http://localhost:8080/configureClouds/)

click kubernetes cloud details -> Kubernetes Url -> paste http://localhost:7777 (kubernetes proxy api which we started in above step) -> check direct connection -> Test (dont provide any credentials),

Output: Connected to Kubernetes v1.20.2

check Disable https certificate check checkbox

Jenkins url -> http://localhost:8080

Pod Template :,

Add Pod template:

Name - kube

Add Container:

Cotainer Template Name - jnlp-slave

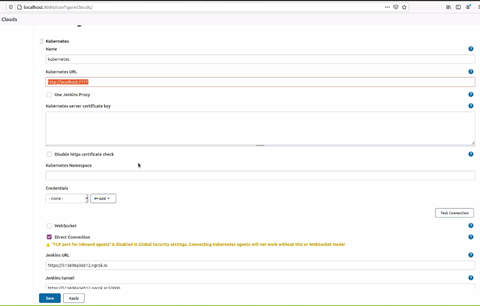
Docker Image - jenkinsci/jnlp-slave

Working directory- /home/jenkins/

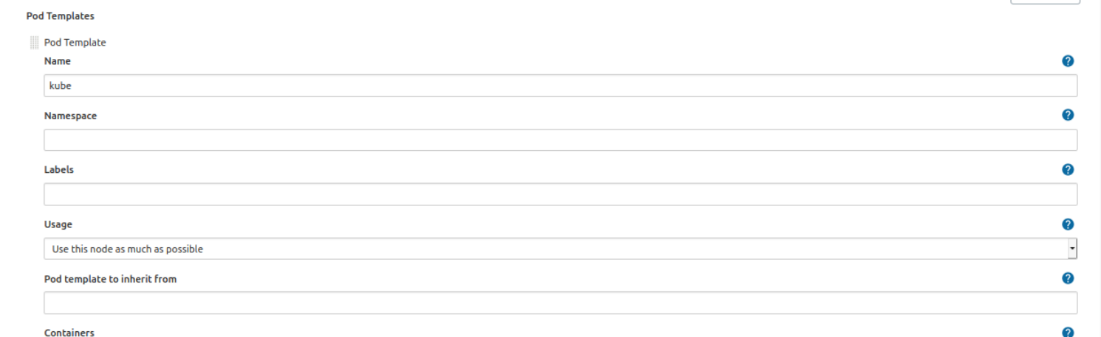
Command to run- /bin/sh -c

Arguments to pass to the command- cat

Click on save









13. Create credential for docker

* jenkins->credentials (http://localhost:8080/credentials/store/system/domain/\_/newCredentials) -> Jekkins -> Global credentials (unrestricted) -> Add Credentials,
* Kind -> Username with password
* Username -> dockerhub username
* password -> dockerhub password
* id -> dockerhub
* Save



14. Go to pipeline syntax on clicking on left side panel or http://localhost:8080/job/Demo/pipeline-syntax/

(Demo is pipeline name)

In sippet generator select following fields

* Sample Step - KubeConfig: Setup Kubernetes CLI (kubectl) OR select -> withKubeCredentials //any one of them is fine
* Kubernetes server endpoint - http://localhost:7777 (this is kubernetes proxy api)
* Certificate of certificate authority - Leave Blank
* Credentials - select mysecret from list (we created mysecret credential i above steps)
* Click on Generate Snippet

kubeconfig(credentialsId: 'mysecret', serverUrl: 'http://localhost:7777') {

// some block

}

We are going to use this block in Jenkinsfile -

<https://github.com/sagar-kale/kube_cicd_demo/blob/master/Jenkinsfile>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Jenkinsfile \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pipeline {,

agent any

stages {

stage('Checkout Source') {

steps {

git url:'https://github.com/ashishjoshi12/docker.git', branch:'main'

}

}

stage("Build image") {

steps {

script {

myapp = docker.build("ashish84/docker:${env.BUILD\_ID}")

}

}

}

stage("Push image") {

steps {

script {

docker.withRegistry('https://registry.hub.docker.com', 'dockerhub') {

myapp.push("latest")

myapp.push("${env.BUILD\_ID}")

}

}

}

}

stage('Deploy App') {

steps {

script {

kubeconfig(credentialsId: 'mysecret', serverUrl: 'http://localhost:7777') {

sh 'kubectl create -f $WORKSPACE/hellowhale.yml'

sh 'kubectl get pods'

sh 'kubectl get svc'

echo 'open minikube ip and svc port which is 31113 in browser'

echo 'deployment completed ........'

}

}

}

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Jenkinsfile \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

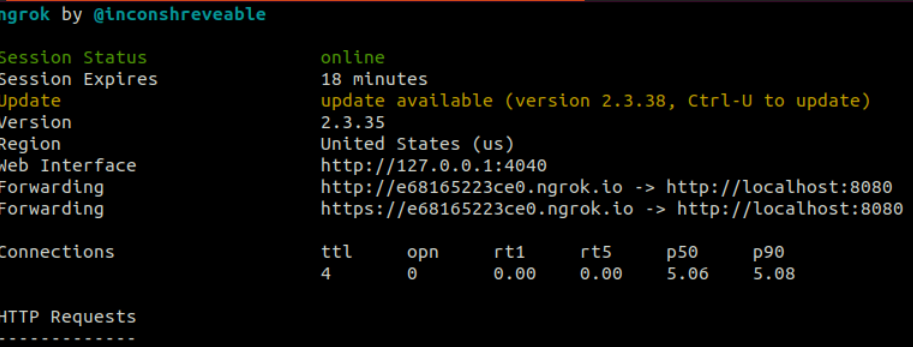
15. Restart jenikins -> sudo systemctl restart jenkins

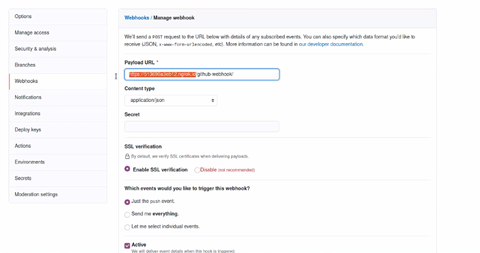
16. Run following command on terminal

ngrok http 8080

copy https url and create webhook in github

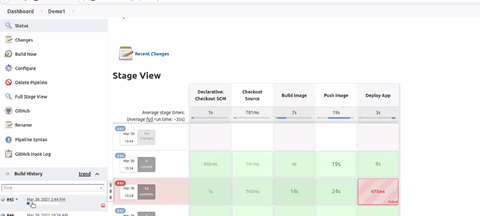
<https://9e639763de3d.ngrok.io/github-webhook/>



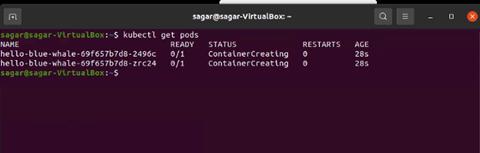


16. Commit some files in github repo. It should trigger jenkins build for Demo project.

Check if deployment is success







minikube ip,

kubectl get svc

open minikube ip with svc port in browser and you will see app message hello blue whale

<http://192.168.49.2:31113/>

