**deploy application by Jenkins**

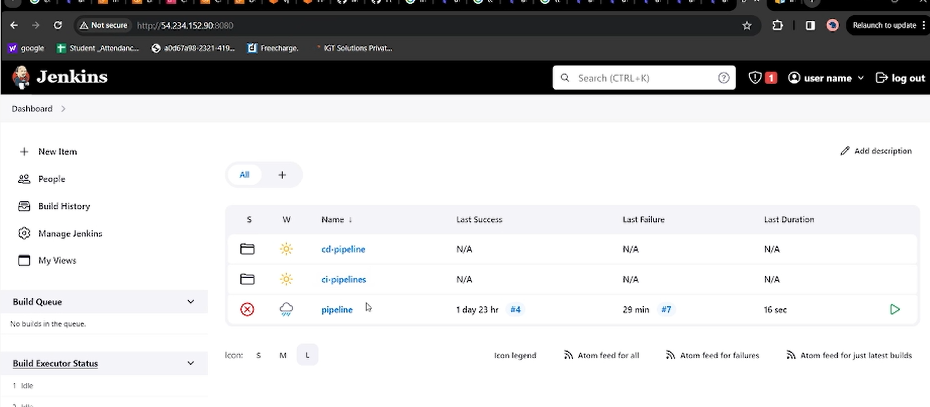
step 1:- create Jenkins master server

* Create cluster,ekctl,kubectl by using following commands:-
* K8s
* Creating cluster
* Step1:- download ekctl first
* k8s commands:-
* https://eksctl.io/installation/
* vim script.sh
* # Replace amd64 with armv6, armv7 or arm64
* (Get-FileHash -Algorithm SHA256 .\eksctl\_Windows\_amd64.zip).Hash -eq ((Get-Content .\eksctl\_checksums.txt) -match 'eksctl\_Windows\_amd64.zip' -split ' ')[0]
* ```
* #### Using Git Bash:
* ```sh
* # for ARM systems, set ARCH to: `arm64`, `armv6` or `armv7`
* ARCH=amd64
* PLATFORM=windows\_$ARCH
* curl -sLO "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl\_$PLATFORM.zip"
* # (Optional) Verify checksum
* curl -sL "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl\_checksums.txt" | grep $PLATFORM | sha256sum --check
* unzip eksctl\_$PLATFORM.zip -d $HOME/bin
* rm eksctl\_$PLATFORM.zip
* step 2:- download kubectl
* https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html
* curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.3/2023-11-14/bin/linux/amd64/kubectl
* chmod +x ./kubectl
* mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
* echo 'export PATH=$HOME/bin:$PATH' >> ~/.bashrc
* step 3:- create cluster
* eksctl create cluster --name cdecb5 --node-type=t3.medium --nodes=2 --region ap-south-1
* kubectl get nodes
* kubectl get nodes -o wide

step 3:- put the admin role on Jenkins master

step 4:- now create folder in Jenkins

* Ci pipeline (continuous integration pipline)
* cd pipeline (continuous deployment pipeline)



Step 5:- crate ne job in ci pipeline named as “student-backend-pipeline”

pipeline {

agent any

stages {

stage('git checkout') {

steps {

git 'https://github.com/Pritam-Khergade/student-ui.git'

}

}

stage('docker install ') {

steps {

sh '''sudo yum install -y docker

sudo systemctl start docker

sudo systemctl enable docker '''

}

}

stage('docker build ') {

steps {

sh 'sudo docker build -t tomcat-student .'

}

}

stage('docker push ') {

steps {

sh ''' aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin 423513534757.dkr.ecr.us-east-1.amazonaws.com

sudo docker tag tomcat-student 423513534757.dkr.ecr.us-east-1.amazonaws.com/cdec\_repo:latest

sudo docker push 423513534757.dkr.ecr.us-east-1.amazonaws.com/cdec\_repo:latest'''

}

}

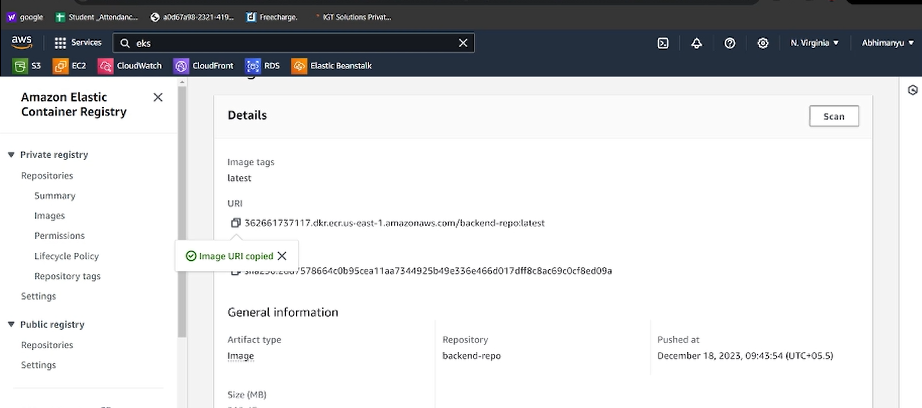
}

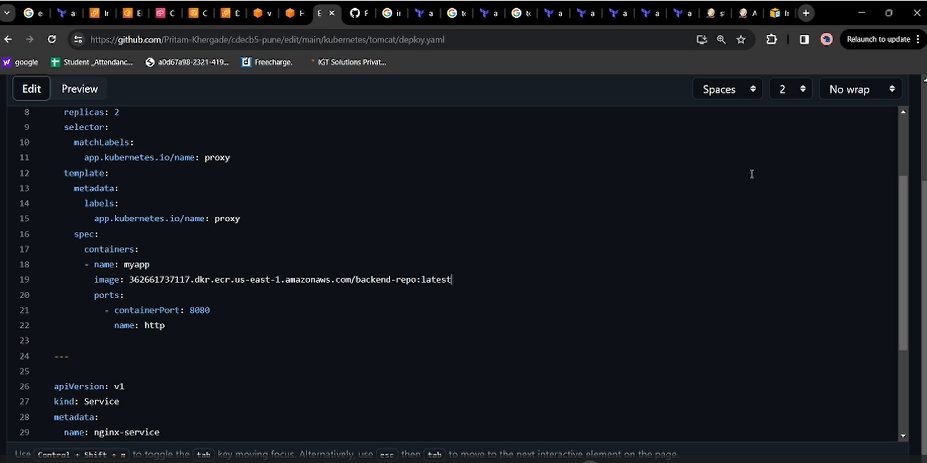
}

Do Apply- save- build.

Step6:- create cd pipeline

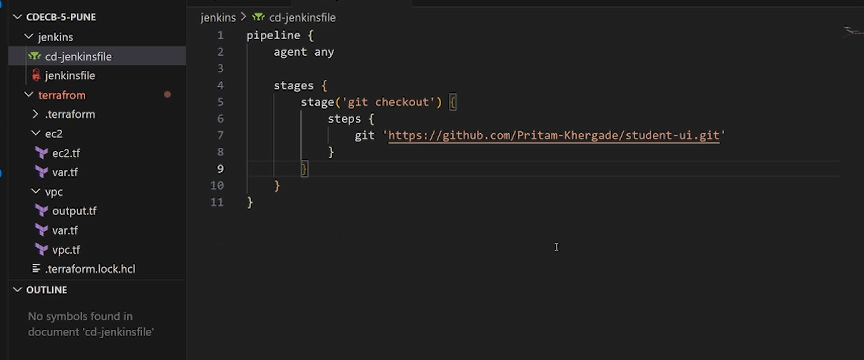
* copy the uri of latest push image from ecr af aws and paste to the deployment.yml





Commit the changes

This id the cd pipeline

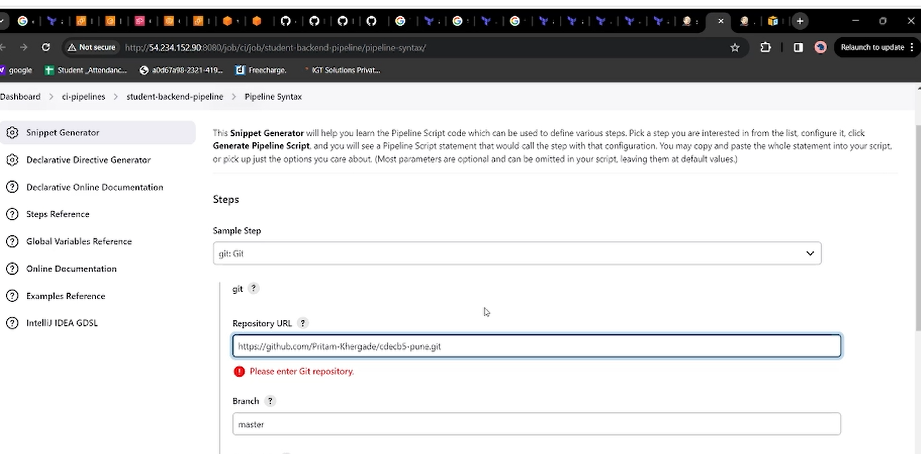


-Now generate tokens from git-settings-developer- personal access token-tokens(clasic)-generate new token

And copy that token on notepad or anyware else.

-go to the Jenkins console in syntax parameter in cd pipeline

-add git and copy the https clone address of git and paste to the repository <URL:->



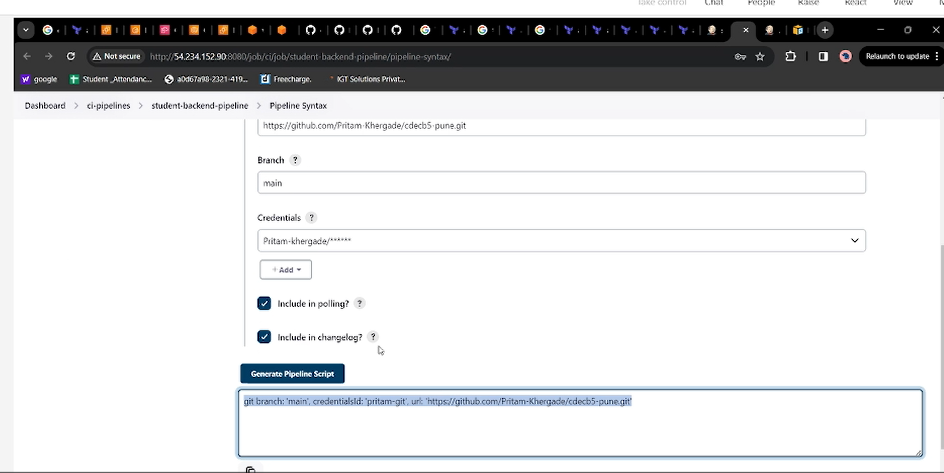
-Set the branch main /master

-add credentials = Jenkins

-username= pritam-kherghare (git username)

-password= paste the token which is generated

-Generate the pipeline syntax



pipeline {

agent

stages {

stage('git checkout') {

steps {

git branch: 'main', credentialsId: 'pritam-git', url: 'https://github.com/Pritam-Khergade/cdecb5-pune.git'

}

}

stage('deploy') {

steps {

sh ''' cd kubernetes/tomcat/.

aws eks update-kubeconfig --region us-east-2 --name demoeks

kubectl apply -f deploy.yaml

'''

}

}

}

}

Step 7:- now deply this image on cluter we made.

-create new pipeline in cd pipeline named ad student-backend-pipeline.

And paste the following pipeline

pipeline {

agent

stages {

stage('git checkout') {

steps {

git branch: 'main', credentialsId: 'pritam-git', url: 'https://github.com/Pritam-Khergade/cdecb5-pune.git'

}

}

stage('deploy') {

steps {

sh ''' cd kubernetes/tomcat/.

aws eks update-kubeconfig --region us-east-2 --name demoeks

kubectl apply -f deploy.yaml

'''

}

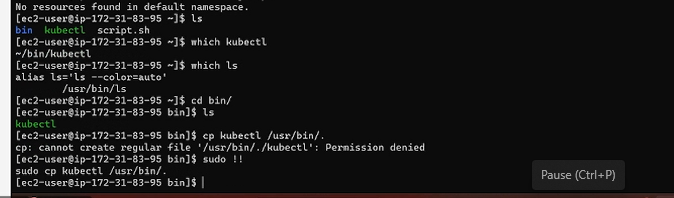
}

}

}

-now error will occur

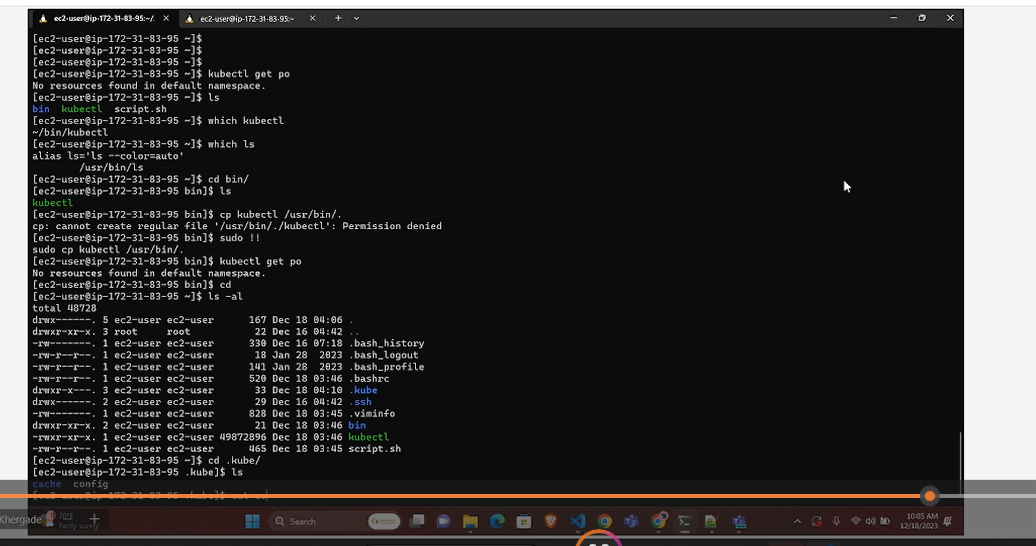
-for that on your server copy kubectl to /usr/bin/.



Now restart the pipeline from restart from the stage.(again error)

Now do:-

* ls –al
* cd .kube
* cat .kube/



Add the cconfig file to the Jenkins foe the access

Add the following command:-

(aws eks update-kubeconfig --region us-east-2 --name demoeks)

pipeline {

agent

stages {

stage('git checkout') {

steps {

git branch: 'main', credentialsId: 'pritam-git', url: 'https://github.com/Pritam-Khergade/cdecb5-pune.git'

}

}

stage('deploy') {

steps {

sh ''' cd kubernetes/tomcat/.

aws eks update-kubeconfig --region us-east-2 --name demoeks

kubectl apply -f deploy.yaml

'''

}

}

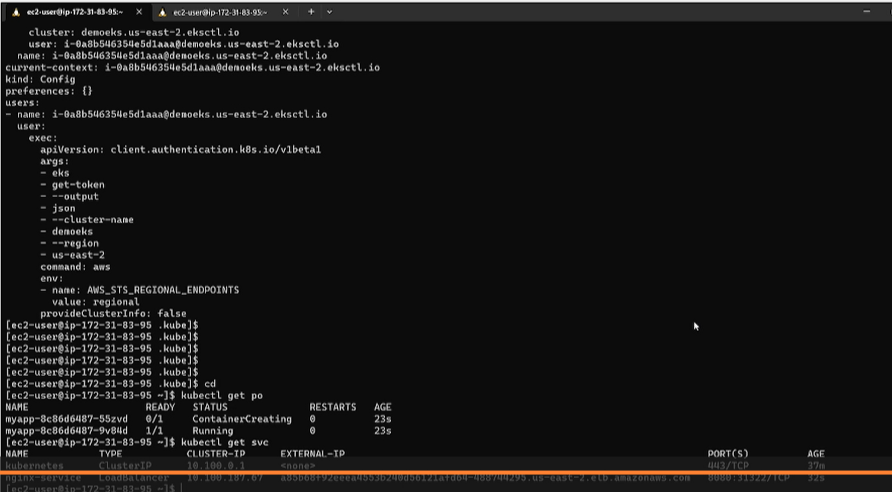
}

}

Step8:-

Kubectl get pods

Kubectl get svc



You will get the load balancer.