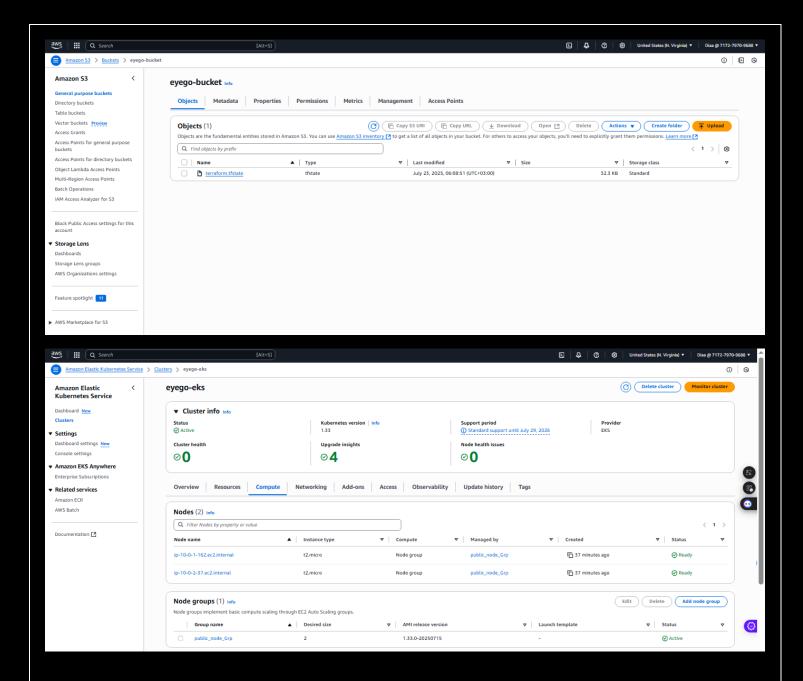


Diaa Qassem

7/23/2025

```
+ enable_network_address_usage_metrics = (known after apply)
                                               = (known after apply)
                                               = "default"
      + instance_tenancy
                                               = (known after apply)
      + ipv6_association_id
      + ipv6_cidr_block = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id
                                              = (known after apply)
      + owner id
                                               = (known after apply)
      + region
                                               = "us-east-1"
      + tags
          + "Name" = "eyego"
      + tags_all
            "Name" = "eyego"
Plan: 19 to add, 0 to change, 0 to destroy.
Changes to Outputs:
  + cluster_arn
                           = (known after apply)
  + cluster endpoint
                          = (known after apply)
                          = (known after apply)
  + cluster id
                         = (known after apply)
  + cluster_version
  + node_group_public_arn = (known after apply)
  + node_group_public_id = (known after apply)
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
  Enter a value:
```

```
🔼 🌣 🕂 CentOS 8 64-bit.tlp - diaa@192.168.1.100:22 - Bitvise xterm - diaa@ITI:-
       + enable dns hostnames
       + enable dns support
       + enable network address usage metrics = (known after apply)
                                                          (known after apply)
                                                          "default"
       + instance_tenancy
       + ipv6_association_id
                                                          (known after apply)
       + ipv6_cidr_block
                                                          (known after apply)
       + ipv6_cidr_block_network_border_group = (known after apply)
       + main route table id
                                                       = (known after apply)
       + owner id
                                                       = (known after apply)
                                                          "us-east-1"
       + region
       + tags
               "Name" = "eyego"
       + tags all
              "Name" = "eyego"
Plan: 20 to add, 0 to change, 0 to destroy.
Changes to Outputs:
  + cluster_arn
                                = (known after apply)
  + cluster endpoint
                                = (known after apply)
  + node group public arn = (known after apply)
  + repository url
                                = (known after apply)
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.
  Enter a value: yes
🔼 💠 🕂 Ce
aws_eks_cluster.eks_cluster: Still creating... [05m30s elapsed]
aws_eks_cluster.eks_cluster: Still creating... [05m40s elapsed]
aws eks cluster.eks cluster: Still creating... [05m50s elapsed]
aws_eks_cluster.eks_cluster: Still creating... [06m00s elapsed] aws_eks_cluster.eks_cluster: Still creating... [06m10s elapsed]
aws_eks_cluster.eks_cluster: Still creating... [06m20s elapsed]
aws_eks_cluster.eks_cluster: Still creating... [06m30s elapsed]
aws_eks_cluster.eks_cluster: Still creating... [06m40s elapsed]
aws_eks_cluster.eks_cluster: Still creating... [06m50s elapsed]
aws_eks_cluster.eks_cluster: Creation complete after 6m59s [id=eyego-eks]
aws_eks_node_group.eks_ng_public: Creating...
aws_eks_node_group.eks_ng_public: Still creating... [00m10s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [00m20s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [00m30s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [00m40s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [00m50s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [01m00s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [01m10s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [01m20s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [01m30s elapsed]
aws_eks_node_group.eks_ng_public: Still creating... [01m40s elapsed]
aws_eks_node_group.eks_ng_public: Creation complete after 1m49s [id=eyego-eks:public_node_Grp]
cluster arn = "arn:aws:eks:us-east-1:717279709688:cluster/eyego-eks"
cluster_endpoint = "https://5688685648B4E0F83FD4FAFB3C9F597A.gr7.us-east-1.eks.amazonaws.com"
node_group_public_arn = "arn:aws:eks:us-east-1:717279709688:nodegroup/eyego-eks/public_node_Grp/b2cc19aa-71a0-bffa-65b5-72dc44c5aabe"
repository url = "717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo
[diaa@ITI EKS_terraform]$
```



Install kubectl

curl --silent --location

"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_\$(uname - s) amd64.tar.gz" | tar xz - C /tmp

sudo mv /tmp/eksctl /usr/local/bin

another way to setup kubectl

curl -LO "https://dl.k8s.io/release/\$(curl -Ls

https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

aws eks describe-cluster --name eyego-eks --region us-east-1 --query "cluster.status"

aws eks --region us-east-1 update-kubeconfig --name eyego-eks

Push image to ecr

```
CentOS 8 64-bit.tlp - diaa@192.168.1.100:22 - Bitvise xterm - diaa@ITk:-/eyego/Task/EKS_terraform

#!/bin/bash

REPO_URL="717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo"

IMAGE_NAME="eyego-app"

echo "logging in to ECR"

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin $REPO_URL

echo "building docker image"

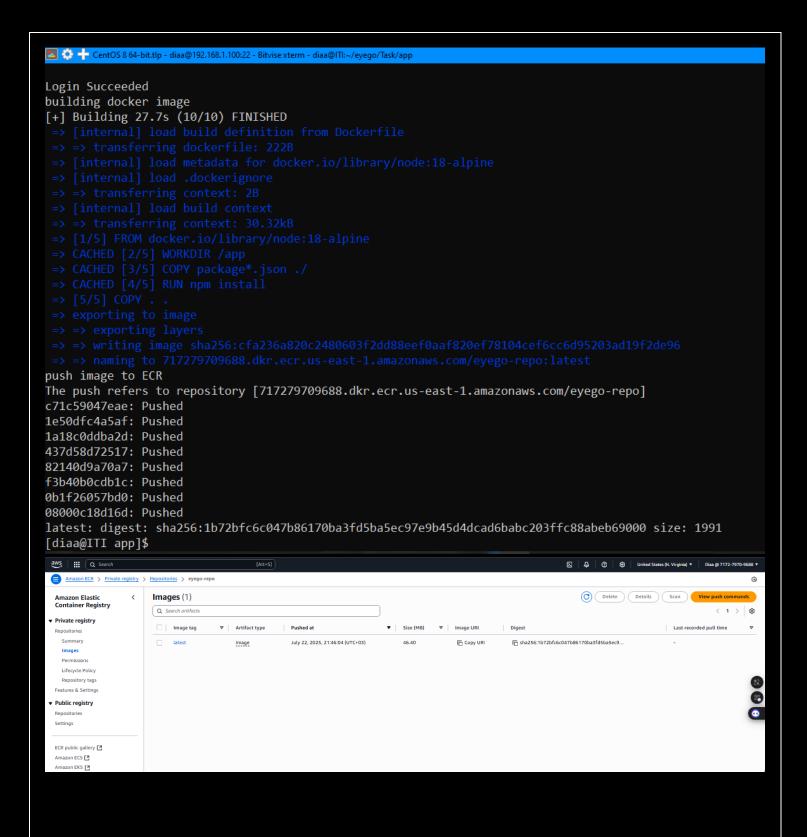
docker build -t $REPO_URL:latest .

echo "push image to ECR"

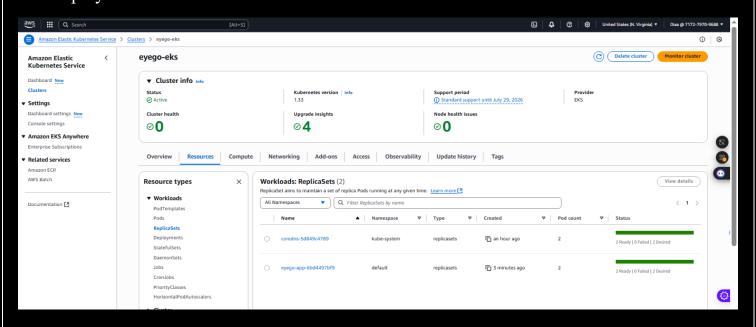
docker push $REPO_URL:latest

~
```

```
🔼 於 🕂 CentOS 8 64-bit.tlp - diaa@192.168.1.100:22 - Bitvise xterm - diaa@ITI:~/eyego/Task/app
logging in to ECR
WARNING! Your credentials are stored unencrypted in '/home/diaa/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
building docker image
[+] Building 27.7s (10/10) FINISHED
push image to ECR
The push refers to repository [717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo]
c71c59047eae: Pushed
1e50dfc4a5af: Pushed
1a18c0ddba2d: Pushed
437d58d72517: Pushed
82140d9a70a7: Pushed
```



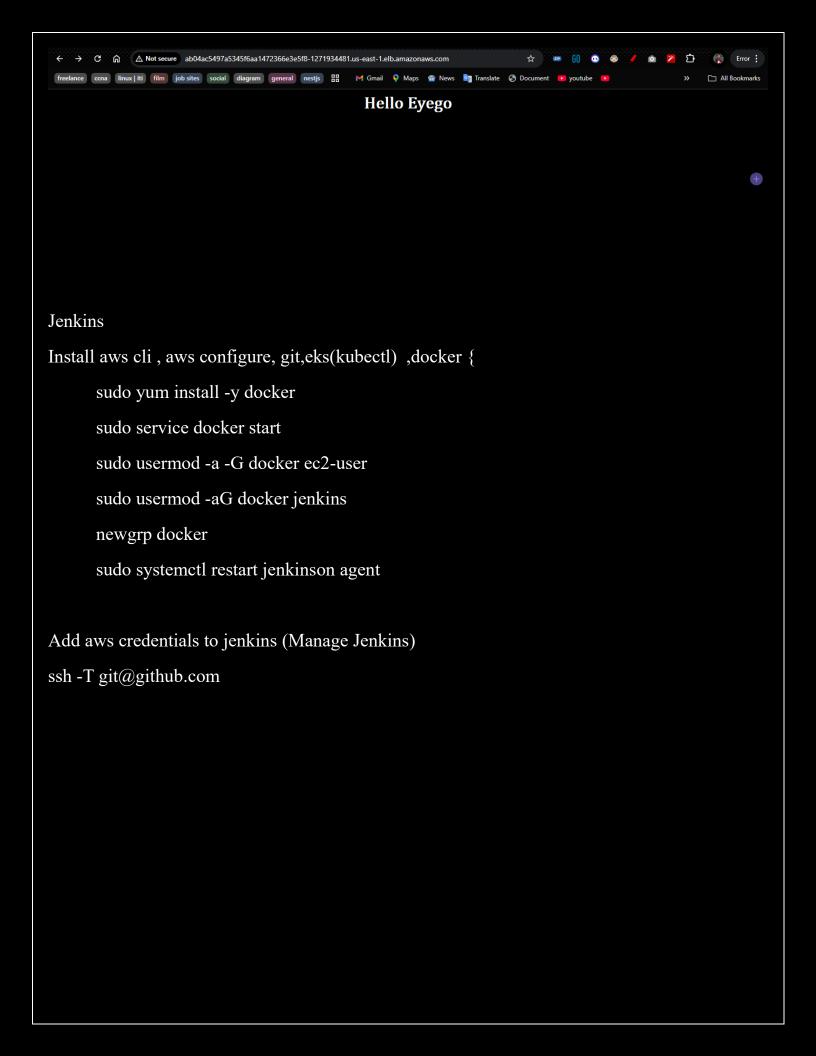
Run deployment [diaa@ITI k8s]\$



Create svc

```
[diaa@ITI k8s]$ vi service.yml
[diaa@ITI k8s]$ kubectl apply -f service.yml
service/eyego-service created
[diaa@ITI k8s]$ kubectl get svc
                               CLUSTER-IP
                                              EXTERNAL-IP
NAME
                TYPF
                                                                                                                          PORT(S)
AGE
                               172.20.124.6
eyego-service
               LoadBalancer
                                              ab04ac5497a5345f6aa1472366e3e5f8-1271934481.us-east-1.elb.amazonaws.com
                                                                                                                         80:30334/TCP
11s
kubernetes
                ClusterIP
                               172.20.0.1
                                              <none>
                                                                                                                          443/TCP
57m
```

[diaa@ITI k8s]\$ curl ab04ac5497a5345f6aa1472366e3e5f8-1271934481.us-east-1.elb.amazonaws.com <body style="background: black; color: white;"><h1><center>
Hello Eyego
</center></h1></body [diaa@ITI k8s]\$ |



```
[ec2-user@ip-172-31-47-125 ~]$ sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
[ec2-user@ip-172-31-47-125 ~]$ swapon --show
[ec2-user@ip-172-31-47-125 ~]$ sudo chmod 600 /swapfile
[ec2-user@ip-172-31-47-125 ~]$ sudo mkswap /swapfile
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=9d54e25d-665b-4ca4-afd3-c544ea927a7c
[ec2-user@ip-172-31-47-125 ~]$ sudo swapon /swapfile
[ec2-user@ip-172-31-47-125 ~]$ swapon --show
          TYPE SIZE USED PRIO
NAME
/swapfile file
                 2G
                      0B
[ec2-user@ip-172-31-47-125 ~]$ sudo rm -rf /tmp/*
[ec2-user@ip-172-31-47-125 ~]$ echo '/swapfile none swap sw 0 0' | sudo t
ee -a /etc/fstab
/swapfile none swap sw 0 0
[ec2-user@ip-172-31-47-125 ~]$ sudo systemctl restart jenkins
```

sudo fallocate -1 2G/swapfile

sudo chmod 600 /swapfile

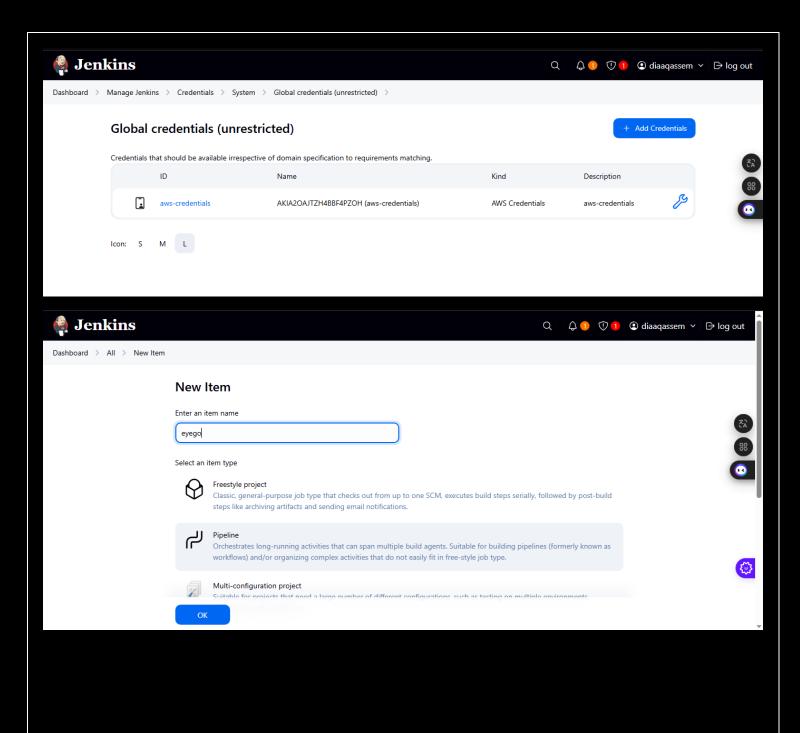
sudo mkswap /swapfile

sudo swapon /swapfile

swapon --show

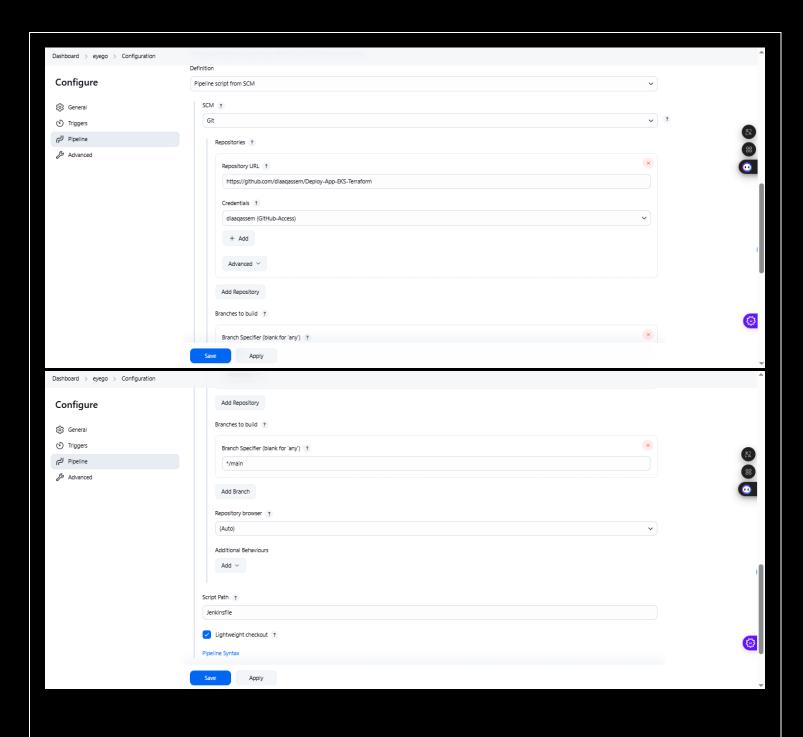
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab

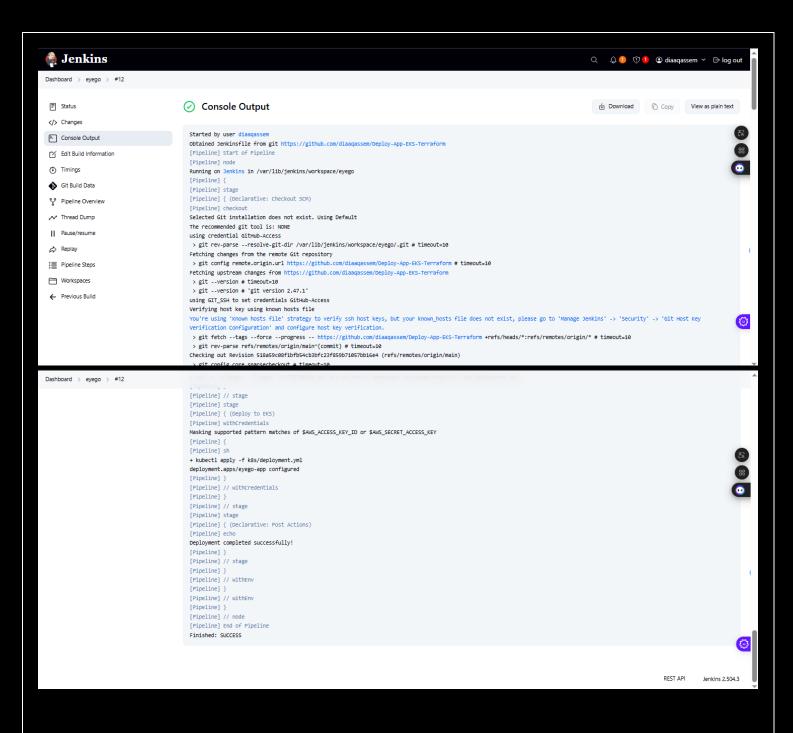
add aws credentials to jenkins

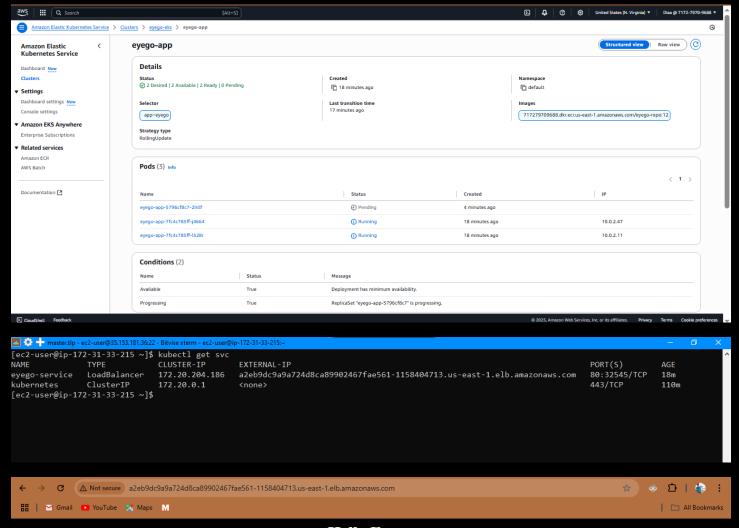


Jenkins file

```
pipeline {
    agent any
    environment {
       AWS_REGION
                     = "us-east-1"
                     = "717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo"
       ECR_REPO
                     = "${BUILD_NUMBER}"
       IMAGE TAG
       CLUSTER_NAME = "eyego-eks"
       DEPLOYMENT_YML = "k8s/deployment.yml"
    stages {
       stage('Checkout Code') {
           steps {
              git url: 'https://github.com/diaaqassem/Deploy-App-EKS-Terraform.git', branch: 'main'
       stage('Build Docker Image') {
               sh "docker build -t $ECR_REPO:$IMAGE_TAG ./app"
       stage('Login to AWS ECR') {
           steps {
               withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
                                credentialsId: 'aws-credentials']]) {
                  aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-stdin $ECR_REPO
       stage('Push Image to ECR') {
           steps {
sh "docker push $ECR_REPO:$IMAGE_TAG"
       stage('Update Kubeconfig') {
              withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
                               credentialsId: 'aws-credentials']]) {
                   sh "aws eks update-kubeconfig --region $AWS_REGION --name $CLUSTER_NAME"
       stage('Update Deployment YAML with new image') {
               sh 'sed -i "s|image: .*|image: ${ECR_REPO}:${IMAGE_TAG}|" ${DEPLOYMENT_YML}'
       stage('Deploy to EKS') {
             sh "kubectl apply -f $DEPLOYMENT_YML"
           echo 'Deployment completed successfully!'
       failure {
           echo 'Deployment failed.'
```

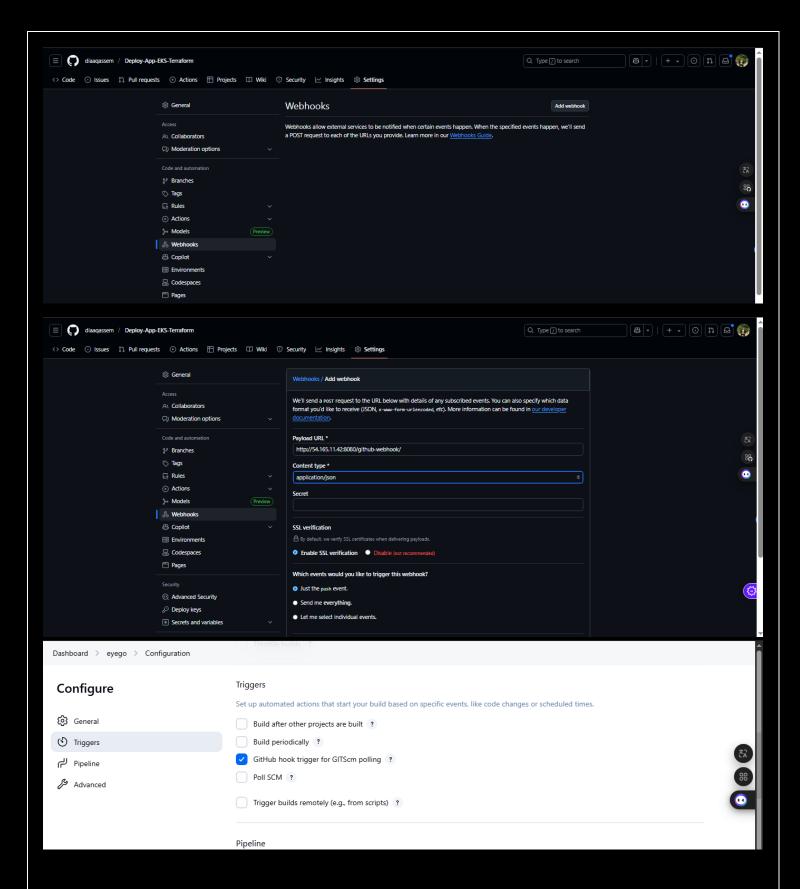




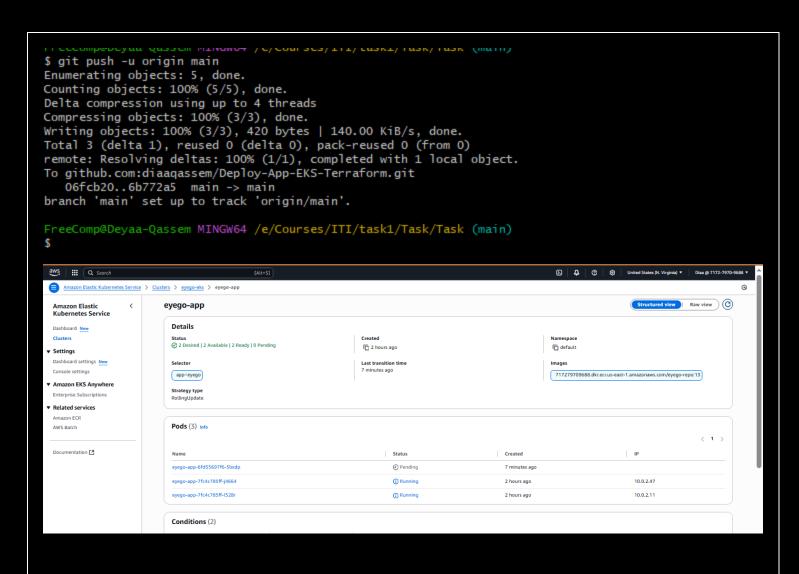


Hello Eyego

Then add trigger



Test trigger



Add slack notification https://qassem.slack.com/services/B096ZCX5TSS?added=1



```
• • •
    def COLOR_MAP = [
        'SUCCESS': 'good',
'FAILURE': 'danger',
    pipeline {
        agent any
        environment {
            AWS_REGION
                           = "us-east-1"
            ECR_REPO
                           = "717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo"
                           = "${BUILD_NUMBER}'
            IMAGE_TAG
            CLUSTER_NAME = "eyego-eks"
            DEPLOYMENT_YML = "k8s/deployment.yml"
        stages {
            stage('Checkout Code') {
               steps {
                    git url: 'https://github.com/diaaqassem/Deploy-App-EKS-Terraform.git', branch: 'main'
            stage('Build Docker Image') {
                steps {
    sh "docker build -t $ECR_REPO:$IMAGE_TAG ./app"
            stage('Login to AWS ECR') {
                    withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
                                      credentialsId: 'aws-credentials']]) {
                            aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-stdin $ECR_REPO
            stage('Push Image to ECR') {
                    sh "docker push $ECR_REPO:$IMAGE_TAG"
            stage('Update Kubeconfig') {
                    withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
                                      credentialsId: 'aws-credentials']]) {
                        sh "aws eks update-kubeconfig --region $AWS_REGION --name $CLUSTER_NAME"
            stage('Update Deployment YAML with new image') {
                    sh 'sed -i "s|image: .*|image: ${ECR_REPO}:${IMAGE_TAG}|" ${DEPLOYMENT_YML}'
            stage('Deploy to EKS') {
                steps {
                   withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
                                      credentialsId: 'aws-credentials']]){
                         sh "kubectl apply -f $DEPLOYMENT_YML"
            always {
                echo 'Slack Notifications.'
                slackSend channel: '#devops'
                    color: COLOR_MAP[currentBuild.currentResult],
                    message: "*${currentBuild.currentResult}:* Job ${env.JOB_NAME} build ${env.BUILD_NUMBER} \n More info at: ${env.BUILD_URL}"
            success {
    echo 'Deployment completed successfully!'
                echo 'Deployment failed.'
```

\$ git push -u origin main Enumerating objects: 5, done. Counting objects: 100% (5/5), done. Delta compression using up to 4 threads Compressing objects: 100% (3/3), done. Writing objects: 100% (3/3), 538 bytes | 269.00 KiB/s, done. Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0) remote: Resolving deltas: 100% (2/2), completed with 2 local objects. To github.com:diaaqassem/Deploy-App-EKS-Terraform.git 6b772a5..49c019d main -> main branch 'main' set up to track 'origin/main'. ← → © Search Qassem @ <u>a</u> 1 (6 | v) Q Qassem ~ ₩ # devops ○ Messages + • Today Y Diaa Qassem 9:50 AM െ Huddles added an integration to this channel: incoming-webhook Ω Diaa Qassem 9:55 AM ▼ Channels added an integration to this channel: jenkins # all-qassem Δ ienkins APP 9:57 AM # devops Slack/Jenkins plugin: you're all set on http://13.223.82.135:8080/ œ # social ienkins APP 10:04 AM + Add channels SUCCESS: Job eyego build 14 More info at: http://13.223.82.135:8080/job/eyego/14/ ▼ Direct messages Piaa Oassem vou + Invite people B I S | Ø ≒ ⊨ | F <>> 型 ▼ Apps Slackbot • + <u>Aa</u> ⁽¹⁾ ⁽²⁾ ⁽²⁾ ⁽²⁾ ⁽²⁾ + Add apps Slack needs your permission to enable notifications. Enable notifications aws | | Q Search Amazon Elastic Kubernetes Service > Clusters > eyego-eks > eyego-app Raw view C Amazon Elastic eyego-app **Kubernetes Service** Dashboard New Namespace Status

2 Desired | 2 Available | 2 Ready | 0 Pending Selector Last transition time Images Console settings 4 minutes ago app=eyego 717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo:14 Amazon EKS Anywhere Strategy type Enterprise Subscriptions Pods (3) Info AWS Batch Documentation [?] 3 hours ago eyego-app-7fc4c785ff-l528 10.0.2.11 👵 🖒 l 🕼 ᠄

Hello Eyego

All Bookmarks

🔡 | 🗹 Gmail 🔼 YouTube 🔀 Maps 🚺