



Eyego-Task

Diaa Qassem
7/23/2025

```
+ enable_network_address_usage_metrics = (known after apply)
+ id                                    = (known after apply)
+ instance_tenancy                     = "default"
+ 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)
+ 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)
+ cluster_id           = (known after apply)
+ cluster_version       = (known after apply)
+ 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:~/eyego/Task/EKS_terraform
+ enable_dns_hostnames           = true
+ enable_dns_support             = true
+ enable_network_address_usage_metrics = (known after apply)
+ id                             = (known after apply)
+ instance_tenancy               = "default"
+ 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)
+ region                         = "us-east-1"
+ 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

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

Apply complete! Resources: 20 added, 0 changed, 0 destroyed.

Outputs:

```
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]$
```

Amazon S3 > Buckets > eyego-bucket

eyego-bucket info

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
terraform.tfstate	tfstate	July 23, 2025, 06:08:51 (UTC+03:00)	32.3 KB	Standard

Amazon Elastic Kubernetes Service > Clusters > eyego-eks

eyego-eks

Cluster info

Status: Active

Kubernetes version: 1.33

Support period: Standard support until July 29, 2026

Provider: EKS

Cluster health: 0

Upgrade insights: 4

Node health issues: 0

Overview | Resources | Compute | Networking | Add-ons | Access | Observability | Update history | Tags

Nodes (2)

Filter Nodes by property or value

Node name	Instance type	Compute	Managed by	Created	Status
ip-10-0-1-162.ec2.internal	t2.micro	Node group	public_node_Grp	37 minutes ago	Ready
ip-10-0-2-37.ec2.internal	t2.micro	Node group	public_node_Grp	37 minutes ago	Ready

Node groups (1)

Node groups implement basic compute scaling through EC2 Auto Scaling groups.

Group name	Desired size	AMI release version	Launch template	Status
public_node_Grp	2	1.33.0-20250715	-	Active

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
```

```
[diaa@ITI EKS_terraform]$ vi ../app/push-image.sh  
[diaa@ITI EKS_terraform]$ aws eks describe-cluster --name eyego-eks --region us-east-1 --query "cluster.status"  
"ACTIVE"  
[diaa@ITI EKS_terraform]$ aws eks --region us-east-1 update-kubeconfig --name eyego-eks  
Added new context arn:aws:eks:us-east-1:717279709688:cluster/eyego-eks to /home/diaa/.kube/config  
[diaa@ITI EKS_terraform]$ kubectl get nodes  
NAME                                STATUS    ROLES    AGE   VERSION  
ip-10-0-1-162.ec2.internal          Ready    <none>   23m   v1.33.0-eks-802817d  
ip-10-0-2-37.ec2.internal           Ready    <none>   22m   v1.33.0-eks-802817d  
[diaa@ITI EKS_terraform]$
```

Push image to ecr

```
CentOS 8 64-bit.tlp - diaa@192.168.1.100:22 - Bitvise xterm - diaa@ITI:~/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
```

```
~
```

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

=> [internal] load build definition from Dockerfile

=> => transferring dockerfile: 222B

=> [internal] load metadata for docker.io/library/node:18-alpine

=> [internal] load .dockerignore

=> => transferring context: 2B

=> [internal] load build context

=> => transferring context: 30.32kB

=> [1/5] FROM docker.io/library/node:18-alpine

=> CACHED [2/5] WORKDIR /app

=> CACHED [3/5] COPY package*.json ./

=> CACHED [4/5] RUN npm install

=> [5/5] COPY . .

=> exporting to image

=> => exporting layers

=> => writing image sha256:cfa236a820c2480603f2dd88eef0aaf820ef78104cef6cc6d95203ad19f2de96

=> => naming to 717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo:latest

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

Login Succeeded

building docker image

[+] Building 27.7s (10/10) FINISHED

=> [internal] load build definition from Dockerfile

=> => transferring dockerfile: 222B

=> [internal] load metadata for docker.io/library/node:18-alpine

=> [internal] load .dockerignore

=> => transferring context: 2B

=> [internal] load build context

=> => transferring context: 30.32kB

=> [1/5] FROM docker.io/library/node:18-alpine

=> CACHED [2/5] WORKDIR /app

=> CACHED [3/5] COPY package*.json ./

=> CACHED [4/5] RUN npm install

=> [5/5] COPY . .

=> exporting to image

=> => exporting layers

=> => writing image sha256:cfa236a820c2480603f2dd88eef0aaf820ef78104cef6cc6d95203ad19f2de96

=> => naming to 717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo:latest

push image to ECR

The push refers to repository [717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo]

c71c59047eae: Pushed

1e50dfc4a5af: Pushed

1a18c0ddbba2d: Pushed

437d58d72517: Pushed

82140d9a70a7: Pushed

f3b40b0cdb1c: Pushed

0b1f26057bd0: Pushed

08000c18d16d: Pushed

latest: digest: sha256:1b72bfc6c047b86170ba3fd5ba5ec97e9b45d4dcad6bab203ffc88abeb69000 size: 1991

[diao@ITI app]\$

Amazon ECR > Private registry > Repositories > eyego-repo

Amazon Elastic Container Registry

Private registry

- Repositories
- Summary
- Images
- Permissions
- Lifecycle Policy
- Repository tags
- Features & Settings

Public registry

- Repositories
- Settings

ECR public gallery

Amazon ECS

Amazon EKS

Images (1)

Search artifacts

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
latest	Image	July 22, 2025, 21:46:04 (UTC+03)	46.40	Copy URI	sha256:1b72bfc6c047b86170ba3fd5ba5ec9...	-

1

```
[diaa@ITI k8s]$ kubectl apply -f deployment.yml
deployment.apps/eyego-app created
[diaa@ITI k8s]$ kubectl get deployment
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
eyego-app	2/2	2	2	65s

```
[diaa@ITI k8s]$
```

Run deployment

Cluster info

- Status: Active
- Kubernetes version: 1.33
- Support period: Standard support until July 29, 2026
- Provider: EKS
- Cluster health: 0
- Upgrade insights: 4
- Node health issues: 0

Workloads: ReplicaSets (2)

Name	Namespace	Type	Created	Pod count	Status
coredns-5d849c4789	kube-system	replicasets	an hour ago	2	2 Ready 0 Failed 2 Desired
eyego-app-6bd4497bf9	default	replicasets	3 minutes ago	2	2 Ready 0 Failed 2 Desired

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
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
eyego-service	LoadBalancer	172.20.124.6	ab04ac5497a5345f6aa1472366e3e5f8-1271934481.us-east-1.elb.amazonaws.com	80:30334/TCP
kubernetes	ClusterIP	172.20.0.1	<none>	443/TCP

```
[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]$
```


Hello Eyego



Jenkins

Install aws cli , aws configure, git,eks(kubectl) ,docker {

```
sudo yum install -y docker
```

```
sudo service docker start
```

```
sudo usermod -a -G docker ec2-user
```

```
sudo usermod -aG docker jenkins
```

```
newgrp docker
```

```
sudo systemctl restart jenkins
```

Add aws credentials to jenkins (Manage Jenkins)

```
ssh -T git@github.com
```

```
[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
NAME      TYPE  SIZE USED PRIO
/swapfile file   2G   0B  -2
[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 tee -a /etc/fstab
/swapfile none swap sw 0 0
[ec2-user@ip-172-31-47-125 ~]$ sudo systemctl restart jenkins
```

`sudo fallocate -l 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

Global credentials (unrestricted)

[+ Add Credentials](#)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description	
 aws-credentials	AKIA2OAJTZH4B8F4PZOH (aws-credentials)	AWS Credentials	aws-credentials	

Icon: ☐ S ☐ M ☒ L

New Item

Enter an item name

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments.

[OK](#)

Jenkins file

```
1 pipeline {
2   agent any
3
4   environment {
5     AWS_REGION      = "us-east-1"
6     ECR_REPO        = "717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo"
7     IMAGE_TAG       = "${BUILD_NUMBER}"
8     CLUSTER_NAME     = "eyego-eks"
9     DEPLOYMENT_YML  = "k8s/deployment.yml"
10  }
11
12  stages {
13    stage('Checkout Code') {
14      steps {
15        git url: 'https://github.com/diaaqqassem/Deploy-App-EKS-Terraform.git', branch: 'main'
16      }
17    }
18
19    stage('Build Docker Image') {
20      steps {
21        sh "docker build -t $ECR_REPO:$IMAGE_TAG ./app"
22      }
23    }
24
25    stage('Login to AWS ECR') {
26      steps {
27        withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
28                          credentialsId: 'aws-credentials']]) {
29          sh '''
30            aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-stdin $ECR_REPO
31          '''
32        }
33      }
34    }
35
36    stage('Push Image to ECR') {
37      steps {
38        sh "docker push $ECR_REPO:$IMAGE_TAG"
39      }
40    }
41
42    stage('Update Kubeconfig') {
43      steps {
44        withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
45                          credentialsId: 'aws-credentials']]) {
46          sh "aws eks update-kubeconfig --region $AWS_REGION --name $CLUSTER_NAME"
47        }
48      }
49    }
50
51    stage('Update Deployment YAML with new image') {
52      steps {
53        sh 'sed -i "s|image: .*|image: ${ECR_REPO}:${IMAGE_TAG}|" ${DEPLOYMENT_YML}'
54      }
55    }
56
57    stage('Deploy to EKS') {
58      steps {
59        withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
60                          credentialsId: 'aws-credentials']]) {
61          sh "kubectl apply -f $DEPLOYMENT_YML"
62        }
63      }
64    }
65  }
66
67  post {
68    success {
69      echo 'Deployment completed successfully!'
70    }
71    failure {
72      echo 'Deployment failed.'
73    }
74  }
75 }
```

Configure

General

Triggers

Pipeline

Advanced

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/diaaqqassem/Deploy-App-EKS-Terraform

Credentials ?

diaaqqassem (GitHub-Access)

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

Save

Apply

Configure

General

Triggers

Pipeline

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Script Path ?

Jenkinsfile

☒ Lightweight checkout ?

Pipeline Syntax

Save

Apply

Dashboard > eyego > #12

Status

</> Changes

Console Output

Edit Build Information

Timings

Git Build Data

Pipeline Overview

Thread Dump

Pause/resume

Replay

Pipeline Steps

Workspaces

Previous Build

Console Output

Download

Copy

View as plain text

```
Started by user diaaqaassem
Obtained Jenkinsfile from git https://github.com/diaaqaassem/Deploy-App-EKS-Terraform
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/eyego
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential GitHub-Access
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/eyego/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/diaaqaassem/Deploy-App-EKS-Terraform # timeout=10
Fetching upstream changes from https://github.com/diaaqaassem/Deploy-App-EKS-Terraform
> git --version # timeout=10
> git --version # 'git version 2.47.1'
using GIT_SSH to set credentials GitHub-Access
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key verification.
> git fetch --tags --force --progress -- https://github.com/diaaqaassem/Deploy-App-EKS-Terraform +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 518a59c08f1bf54cb3bfc23f859b71857bb16e4 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
```

Dashboard > eyego > #12

```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to EKS)
[Pipeline] withCredentials
Masking supported pattern matches of $AWS_ACCESS_KEY_ID or $AWS_SECRET_ACCESS_KEY
[Pipeline] {
[Pipeline] sh
+ kubectl apply -f k8s/deployment.yml
deployment.apps/eyego-app configured
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deployment completed successfully!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Amazon Elastic Kubernetes Service

eyego-app

Structured view Raw view

Details

Status: 2 Desired | 2 Available | 2 Ready | 0 Pending

Created: 18 minutes ago

Namespace: default

Selector: app=eyego

Last transition time: 17 minutes ago

Strategy type: RollingUpdate

Images: 717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo:12

Pods (3)

Name	Status	Created	IP
eyego-app-5796cf8c7-2htdf	Pending	4 minutes ago	
eyego-app-7fc4c785ff-j4664	Running	18 minutes ago	10.0.2.47
eyego-app-7fc4c785ff-1528r	Running	18 minutes ago	10.0.2.11

Conditions (2)

Name	Status	Message
Available	True	Deployment has minimum availability.
Progressing	True	ReplicaSet "eyego-app-5796cf8c7" is progressing.

```
master.tlp - ec2-user@35.153.181.36:22 - Bitvise xterm - ec2-user@ip-172-31-33-215:~
[ec2-user@ip-172-31-33-215 ~]$ kubectl get svc
NAME                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
eyego-service       LoadBalancer  172.20.204.186  a2eb9dc9a9a724d8ca89902467fae561-1158404713.us-east-1.elb.amazonaws.com  80:32545/TCP    18m
kubernetes           ClusterIP      172.20.0.1      <none>           443/TCP          110m
[ec2-user@ip-172-31-33-215 ~]$
```

Not secure a2eb9dc9a9a724d8ca89902467fae561-1158404713.us-east-1.elb.amazonaws.com

Gmail YouTube Maps M All Bookmarks

Hello Eyego

Then add trigger

diaaqqassem / Deploy-App-EKS-Terraform

Type to search

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Models

Webhooks

Copilot

Environments

Codespaces

Pages

Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

diaaqqassem / Deploy-App-EKS-Terraform

Type to search

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Models

Webhooks

Copilot

Environments

Codespaces

Pages

Security

Advanced Security

Deploy keys

Secrets and variables

Webhooks / Add webhook

We'll send a post request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type *

Secret

SSL verification

Which events would you like to trigger this webhook?

Dashboard > eyego > Configuration

Configure

General

Triggers

Pipeline

Advanced

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Build after other projects are built

☐ Build periodically

☒ GitHub hook trigger for GITScm polling

☐ Poll SCM

☐ Trigger builds remotely (e.g., from scripts)

Pipeline

Test trigger


```
FreeComp@Deyaa-Qassem MINGW64 /e/Courses/ITI/task1/Task/Task (main)
$ 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), 420 bytes | 140.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:diaaqqassem/Deploy-App-EKS-Terraform.git
   06fcb20..6b772a5  main -> main
branch 'main' set up to track 'origin/main'.
```

```
FreeComp@Deyaa-Qassem MINGW64 /e/Courses/ITI/task1/Task/Task (main)
$
```

The screenshot shows the Amazon Elastic Kubernetes Service (EKS) console. The left sidebar contains navigation links for Dashboard, Clusters, Settings, Amazon EKS Anywhere, and Related services. The main content area displays the details for the 'eyego-app' cluster. The 'Details' section shows the cluster status as '2 Desired | 2 Available | 2 Ready | 0 Pending', created 2 hours ago, and using the 'app-eyego' selector. The 'Pods' section shows a table with 3 pods: one pending and two running. The 'Conditions' section shows 2 conditions.

Name	Status	Created	IP
eyego-app-6fd55697f6-5bxdp	Pending	7 minutes ago	
eyego-app-7fc4c785ff-j4664	Running	2 hours ago	10.0.2.47
eyego-app-7fc4c785ff-1528r	Running	2 hours ago	10.0.2.11

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

The screenshot shows the Jenkins configuration page for a Slack notification. The 'Slack' section is active. The 'Workspace' field is set to 'qassem'. The 'Credential' field is set to 'Secret text'. The 'Default channel / member id' field is set to '#devops'. There is a checkbox for 'Custom slack app bot user' which is currently unchecked. The 'Advanced' section is collapsed. The 'Save' button is highlighted in blue.



```
1 def COLOR_MAP = [
2   'SUCCESS': 'good',
3   'FAILURE': 'danger',
4 ]
5
6 pipeline {
7   agent any
8
9   environment {
10    AWS_REGION      = "us-east-1"
11    ECR_REPO        = "717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo"
12    IMAGE_TAG       = "${BUILD_NUMBER}"
13    CLUSTER_NAME    = "eyego-eks"
14    DEPLOYMENT_YML = "k8s/deployment.yml"
15  }
16
17  stages {
18    stage('Checkout Code') {
19      steps {
20        git url: 'https://github.com/diaaqaassem/Deploy-App-EKS-Terraform.git', branch: 'main'
21      }
22    }
23
24    stage('Build Docker Image') {
25      steps {
26        sh "docker build -t $ECR_REPO:$IMAGE_TAG ./app"
27      }
28    }
29
30    stage('Login to AWS ECR') {
31      steps {
32        withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
33                          credentialsId: 'aws-credentials']]) {
34          sh '''
35            aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-stdin $ECR_REPO
36          '''
37        }
38      }
39    }
40
41    stage('Push Image to ECR') {
42      steps {
43        sh "docker push $ECR_REPO:$IMAGE_TAG"
44      }
45    }
46
47    stage('Update Kubeconfig') {
48      steps {
49        withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
50                          credentialsId: 'aws-credentials']]) {
51          sh "aws eks update-kubeconfig --region $AWS_REGION --name $CLUSTER_NAME"
52        }
53      }
54    }
55
56    stage('Update Deployment YAML with new image') {
57      steps {
58        sh 'sed -i "s|image: .*|image: ${ECR_REPO}:${IMAGE_TAG}|" ${DEPLOYMENT_YML}'
59      }
60    }
61
62    stage('Deploy to EKS') {
63      steps {
64        withCredentials([[$class: 'AmazonWebServicesCredentialsBinding',
65                          credentialsId: 'aws-credentials']]) {
66          sh "kubectl apply -f $DEPLOYMENT_YML"
67        }
68      }
69    }
70  }
71
72  post {
73    always {
74      echo 'Slack Notifications.'
75      slackSend channel: '#devops',
76                color: COLOR_MAP[currentBuild.currentResult],
77                message: "${currentBuild.currentResult}:* Job ${env.JOB_NAME} build ${env.BUILD_NUMBER} \n More info at: ${env.BUILD_URL}"
78    }
79    success {
80      echo 'Deployment completed successfully!'
81    }
82    failure {
83      echo 'Deployment failed.'
84    }
85  }
86 }
```

```
$ 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:diaaqqassem/Deploy-App-EKS-Terraform.git
 6b772a5..49c019d  main -> main
branch 'main' set up to track 'origin/main'.
```

Search Qassem

devops

Messages

Today

Diaa Qassem 9:50 AM
added an integration to this channel: [incoming-webhook](#)

Diaa Qassem 9:55 AM
added an integration to this channel: [jenkins](#)

jenkins APP 9:57 AM
Slack/Jenkins plugin: you're all set on <http://13.223.82.135:8080/>

jenkins APP 10:04 AM
SUCCESS: Job eyego build 14
More info at: <http://13.223.82.135:8080/job/eyego/14/>

Message #devops

Slack needs your permission to enable notifications. [Enable notifications](#)

Amazon Elastic Kubernetes Service

Clusters > eyego-eks > eyego-app

eyego-app

Structured view Raw view

Details

Status
2 Desired | 2 Available | 2 Ready | 0 Pending

Created
3 hours ago

Namespace
default

Selector
app=eyego

Last transition time
4 minutes ago

Images
717279709688.dkr.ecr.us-east-1.amazonaws.com/eyego-repo:14

Strategy type
RollingUpdate

Pods (3) Info

Name	Status	Created	IP
eyego-app-56474df547-bm8rg	Pending	4 minutes ago	
eyego-app-7fc4c785ff-j4664	Running	3 hours ago	10.0.2.47
eyego-app-7fc4c785ff-1528r	Running	3 hours ago	10.0.2.11

Not secure a2eb9dc9a9a724d8ca89902467fae561-1158404713.us-east-1.elb.amazonaws.com

Gmail YouTube Maps

All Bookmarks

Hello Eyego