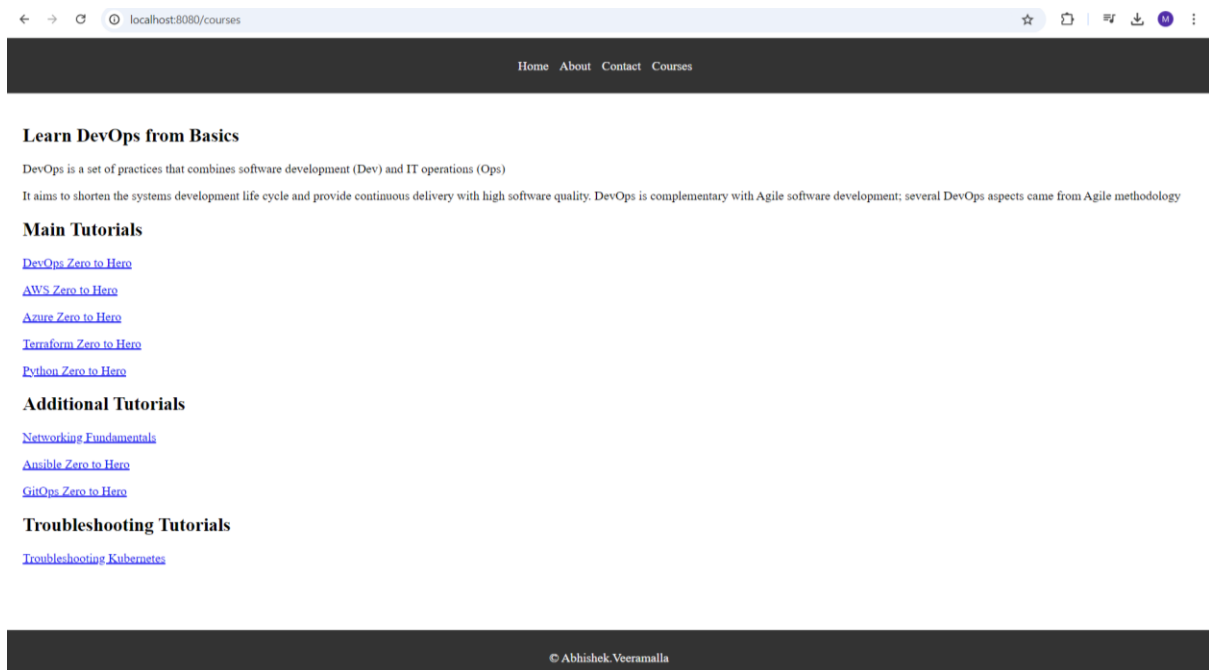
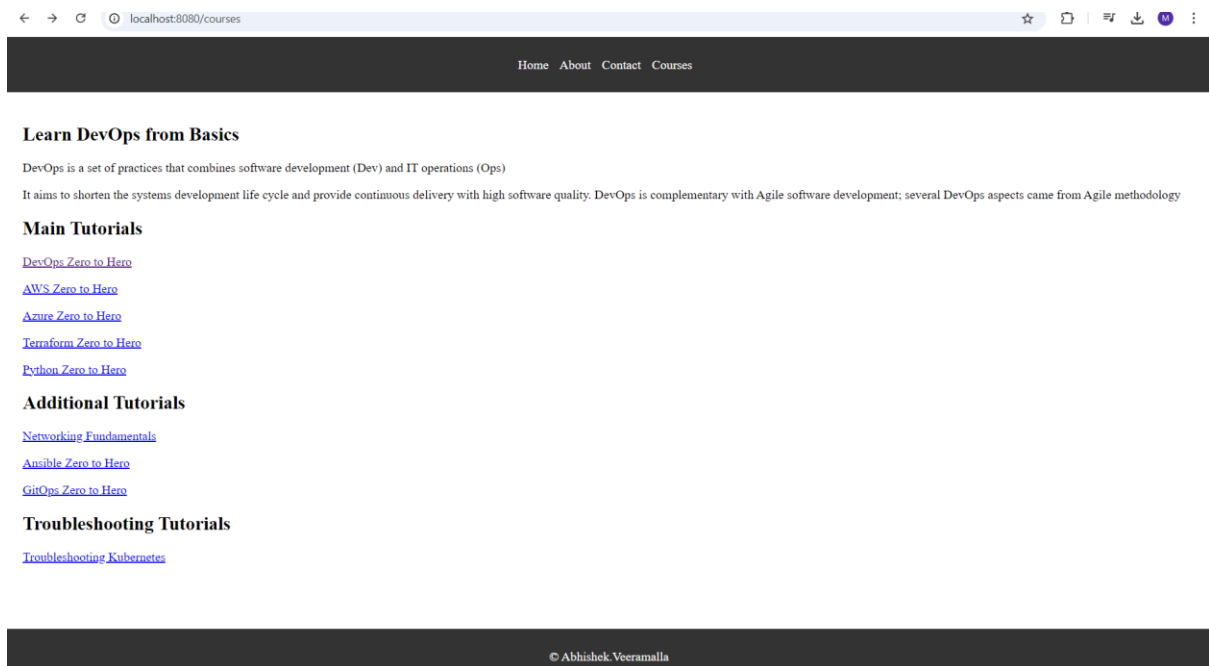


## 1.Run locally the application is running



## 2.Create the dockerfile then run the application using dockerfile.



## 3.Eks cluster creation

```
MINGW64/C:/Users/Amanudeen
$ eksctl create cluster --name demo-cluster --region us-east-1
2024-08-29 09:57:39 [i] eksctl version 0.189.0
2024-08-29 09:57:39 [i] using region us-east-1
2024-08-29 09:57:51 [i] setting availability zones to [us-east-1f us-east-1d]
2024-08-29 09:57:51 [i] subnets for us-east-1f - public:192.168.0.0/19 private:
192.168.64.0/19
2024-08-29 09:57:51 [i] subnets for us-east-1d - public:192.168.32.0/19 private:
192.168.96.0/19
2024-08-29 09:57:51 [i] nodegroup "ng-cf76e8b8" will use "" [AmazonLinux2/1.30]
2024-08-29 09:57:51 [i] using Kubernetes version 1.30
2024-08-29 09:57:51 [i] creating EKS cluster "demo-cluster" in "us-east-1" regi
on with managed nodes
2024-08-29 09:57:51 [i] will create 2 separate CloudFormation stacks for cluste
r itself and the initial managed nodegroup
2024-08-29 09:57:51 [i] if you encounter any issues, check CloudFormation conso
le or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=demo-cluste
r'
2024-08-29 09:57:51 [i] Kubernetes API endpoint access will use default of [pub
licAccess=true, privateAccess=false] for cluster "demo-cluster" in "us-east-1"
2024-08-29 09:57:51 [i] Cloudwatch logging will not be enabled for cluster "dem
o-cluster" in "us-east-1"
2024-08-29 09:57:51 [i] you can enable it with 'eksctl utils update-cluster-log
ging --enable-types=SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all) --region=us-east-1
--cluster=demo-cluster'
2024-08-29 09:57:51 [i] default addons vpc-cni, kube-proxy, coredns were not sp
ecified, will install them as EKS addons
2024-08-29 09:57:51 [i]
2 sequential tasks: { create cluster control plane "demo-cluster",
2 sequential sub-tasks: {
2 sequential sub-tasks: {
1 task: { create addons },
},
wait for control plane to become ready,
},
create managed nodegroup "ng-cf76e8b8",
}
2024-08-29 09:57:51 [i] building cluster stack "eksctl-demo-cluster-cluster"
2024-08-29 09:58:03 [i] deploying stack "eksctl-demo-cluster-cluster"
2024-08-29 09:58:33 [i] waiting for CloudFormation stack "eksctl-demo-cluster-c
luster"
2024-08-29 09:59:14 [i] waiting for CloudFormation stack "eksctl-demo-cluster-c
luster"
2024-08-29 10:00:26 [i] waiting for CloudFormation stack "eksctl-demo-cluster-c
luster"
2024-08-29 10:01:38 [i] waiting for CloudFormation stack "eksctl-demo-cluster-c
luster"
2024-08-29 10:02:49 [i] waiting for CloudFormation stack "eksctl-demo-cluster-c
luster"
2024-08-29 10:12:21 [i] recommended policies were found for "vpc-cni" addon, bu
t since OIDC is disabled on the cluster, eksctl cannot configure the requested p
ermissions; the recommended way to provide IAM permissions for "vpc-cni" addon i
s via pod identity associations; after addon creation is completed, add all reco
mmended policies to the config file, under 'addon.PodIdentityAssociations', and
run 'eksctl update addons'
2024-08-29 10:12:21 [i] creating addon
2024-08-29 10:12:22 [i] successfully created addon
2024-08-29 10:12:22 [i] creating addon
2024-08-29 10:12:23 [i] successfully created addon
2024-08-29 10:12:23 [i] creating addon
2024-08-29 10:12:25 [i] successfully created addon
2024-08-29 10:15:01 [i] building managed nodegroup stack "eksctl-demo-cluster-n
odegroup-ng-cf76e8b8"
2024-08-29 10:15:24 [i] deploying stack "eksctl-demo-cluster-nodegroup-ng-cf76e
```

```
MINGW64/C:/Users/Amanudeen/OneDrive/Desktop/go-web-app
$ kubectl get pods
NAME READY STATUS RESTARTS AGE
go-web-app-c7687b757-hdthp 0/1 InvalidImageName 0 30s

$ kubectl get pods
NAME READY STATUS RESTARTS AGE
go-web-app-c7687b757-hdthp 0/1 InvalidImageName 0 49s

$ kubectl get pods
NAME READY STATUS RESTARTS AGE
go-web-app-c7687b757-hdthp 0/1 InvalidImageName 0 58s

$ kubectl apply -f k8s/manifests/deployment.yaml
deployment.apps/go-web-app configured

$ kubectl get pods
NAME READY STATUS RESTARTS AGE
go-web-app-566c74c488-fmbrv 1/1 Running 0 8s

$ kubectl apply -f k8s/manifests/service.yaml
error: no objects passed to apply

$ kubectl apply -f k8s/manifests/service.yaml
service/go-web-app created

$ kubectl apply -f k8s/manifests/ingress.yaml
ingress.networking.k8s.io/go-web-app created

$ kubectl get ing
NAME CLASS HOSTS ADDRESS PORTS AGE
go-web-app nginx go-web-app.local 80 6m57s

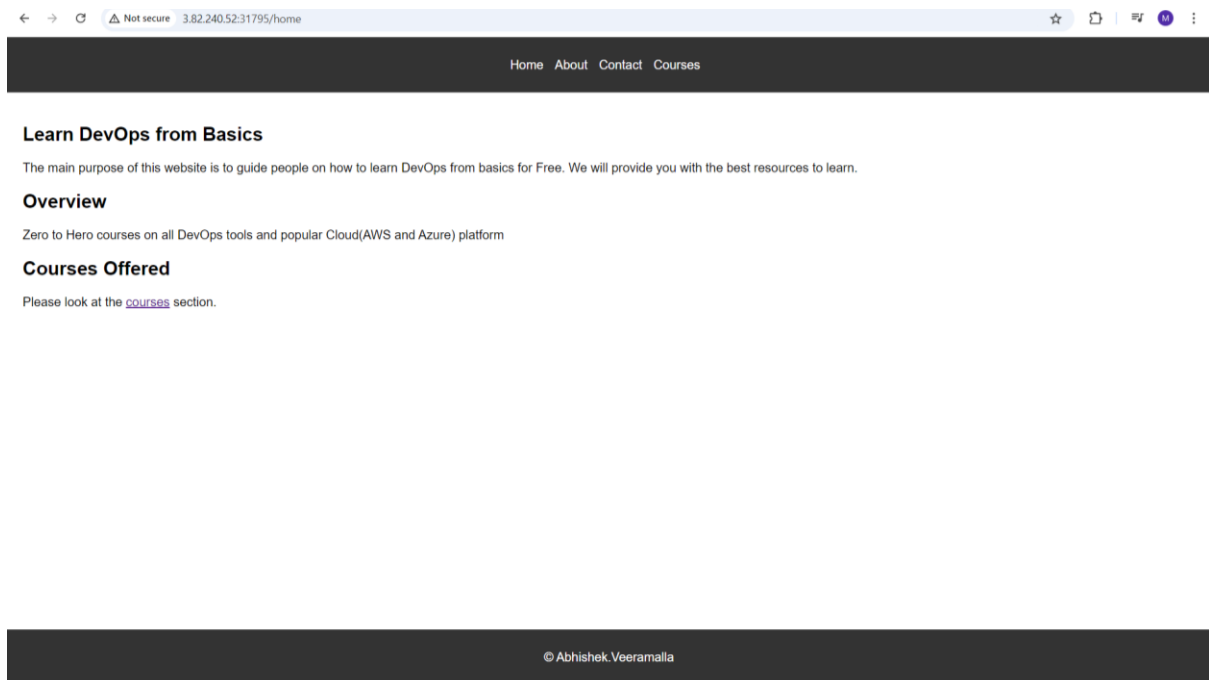
$ kubectl edit svc go-web-app

$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
go-web-app ClusterIP 10.100.200.25 <none> 80/TCP 10m
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 75m

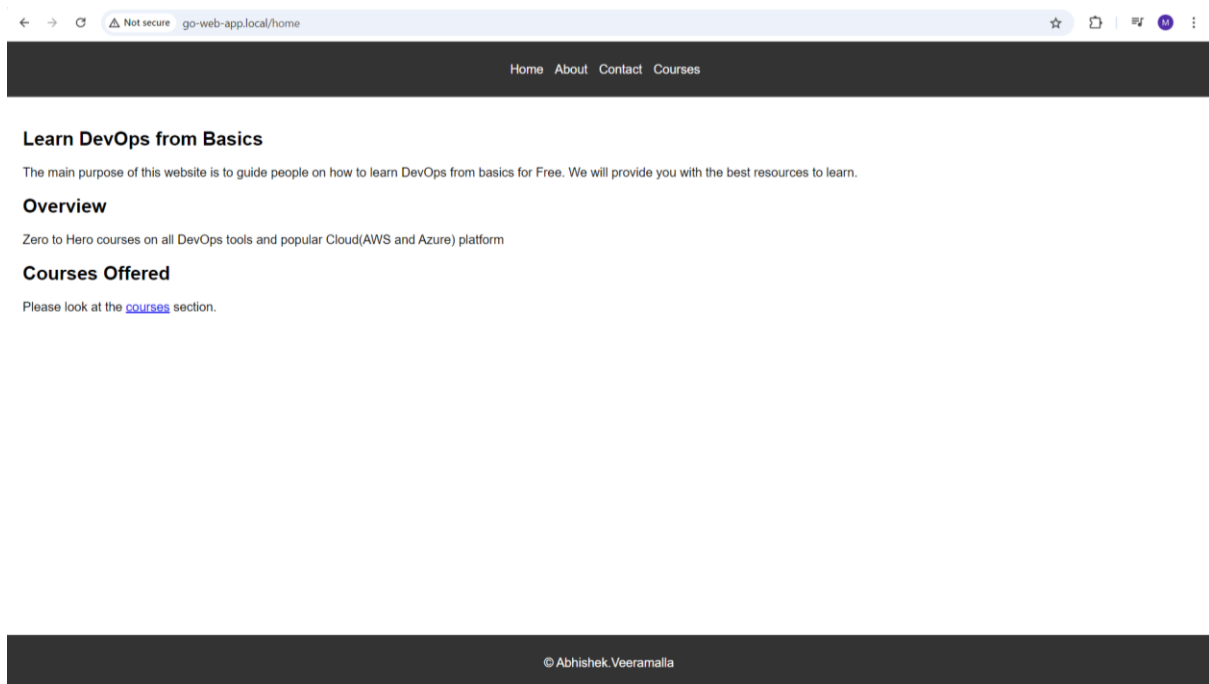
$ kubectl edit svc go-web-app
service/go-web-app edited

$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
go-web-app NodePort 10.100.200.25 <none> 80:31293/TCP 12m
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 76m
```

#### 4. Able to access using public ip with nodeport its running



## 5.Access the portal using go-web-app.local is running



## 6.Create and install Helm part done

```
MINGW64:/c:/Amanudeen/helm
$ helm install go-web-app ./go-web-app-chart
Error: INSTALLATION FAILED: template: go-web-app-chart/templates/deployment.yaml:20:46: executing "go-web-app-chart/templates/deployment.yaml" at <.values.image.tag>: nil pointer evaluating interface {}.image

MINGW64:/c:/Amanudeen/helm
$ helm install go-web-app ./go-web-app-chart
Error: INSTALLATION FAILED: template: go-web-app-chart/templates/deployment.yaml:20:46: executing "go-web-app-chart/templates/deployment.yaml" at <.values.image.tag>: nil pointer evaluating interface {}.image

MINGW64:/c:/Amanudeen/helm
$ AC

MINGW64:/c:/Amanudeen/helm
$ helm install go-web-app ./go-web-app-chart
NAME: go-web-app
LAST DEPLOYED: Sat Aug 31 14:04:39 2024
NAMESPACE: default
STATUS: deployed
REVISION: 3
TEST SUITE: None

MINGW64:/c:/Amanudeen/helm
$ }
```

## 7.CI part also done

github.com/Amanudeen/go-web-app/actions/runs/10644662564

Amanudeen / go-web-app

CI/CD

implemented CI #1

Summary

Jobs

- build
- code-quality
- push
- update-newtag-in-helm-chart

Run details

- Usage
- Workflow file

Triggered via push 3 minutes ago

Amanudeen pushed · adb7344 · main

Status: Success

Total duration: 1m 20s

Artifacts: 1

ci.yaml

on: push

build (15s) → push (36s) → update-newtag-in-helm-chart (4s)

code-quality (21s)

push summary


Docker Build summary

For a detailed look at the build, download the following build record archive and import it into Docker Desktop's Builds view.

hub.docker.com/r/amanudeen/go-web-app/tags

dockerhub Explore Repositories Organizations Search Docker Hub

Explore / amanudeen/go-web-app

 **amanudeen/go-web-app** ☆0  
By amanudeen · Updated 3 minutes ago Manage Repository

Pulls 10

Overview **Tags**

Sort by **Newest** Filter Tags

TAG	Digest	OS/ARCH	Last pull	Compressed Size
10644662564 Last pushed 3 minutes ago by amanudeen	a5623b47eef	linux/amd64	---	13.14 MB
v1 Last pushed 3 days ago by amanudeen		linux/amd64	3 hours ago	12.08 MB

docker pull amanudeen/go-web-app:10644662564 Copy

docker pull amanudeen/go-web-app:v1 Copy


<https://hub.docker.com/r/amanudeen/go-web-app/tags>

## 8.Argocd running

Not secure https://a0240e40706b542958b857365d7b88f6-1443905462.us-east-1.elb.amazonaws.com/login?return\_url=https%3A%2F%2Fa0240e40706b542958b857365d7b88f6-14...

argo

Let's get stuff deployed!

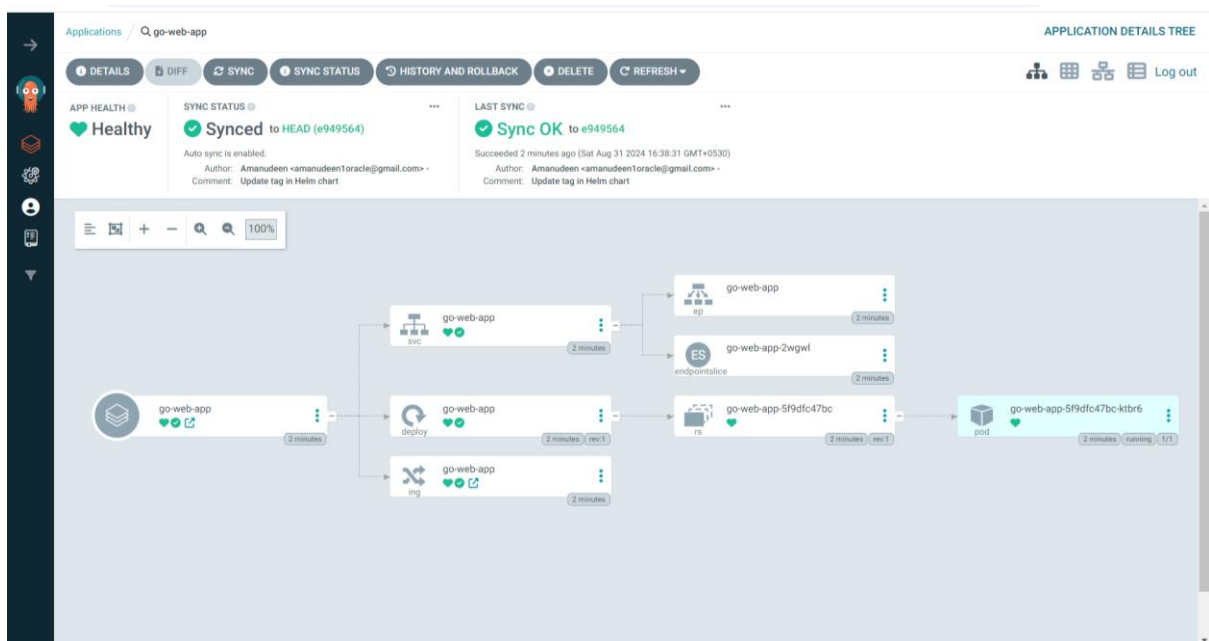
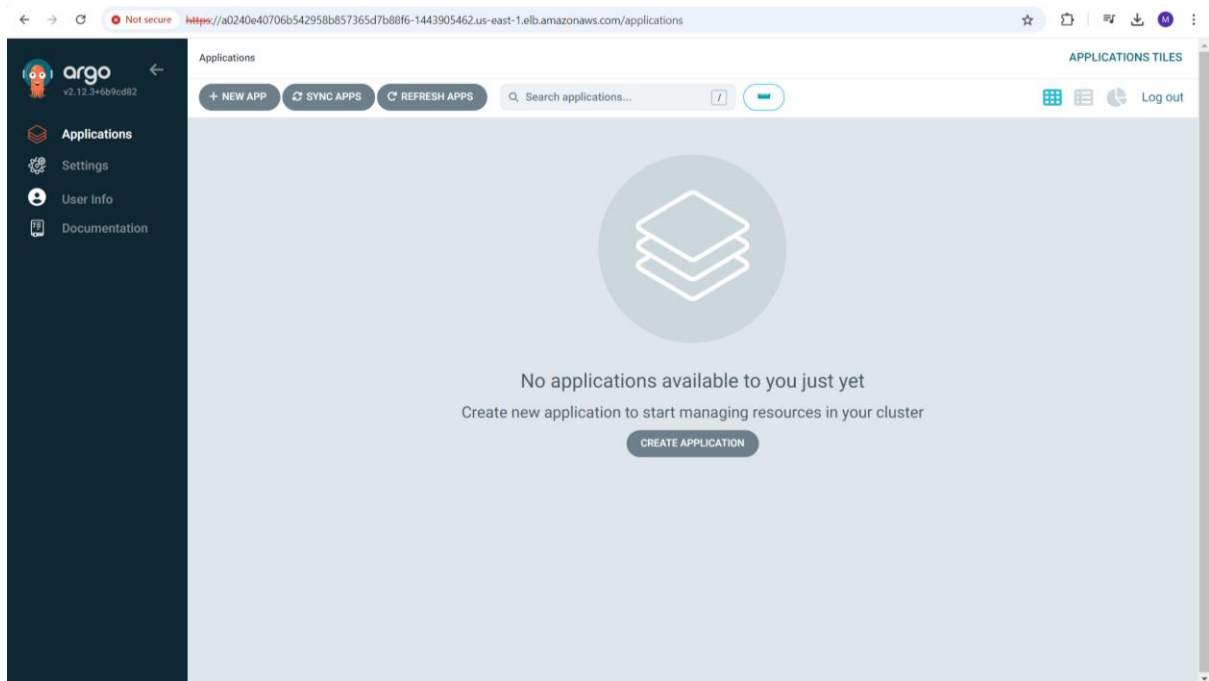


Username

Password

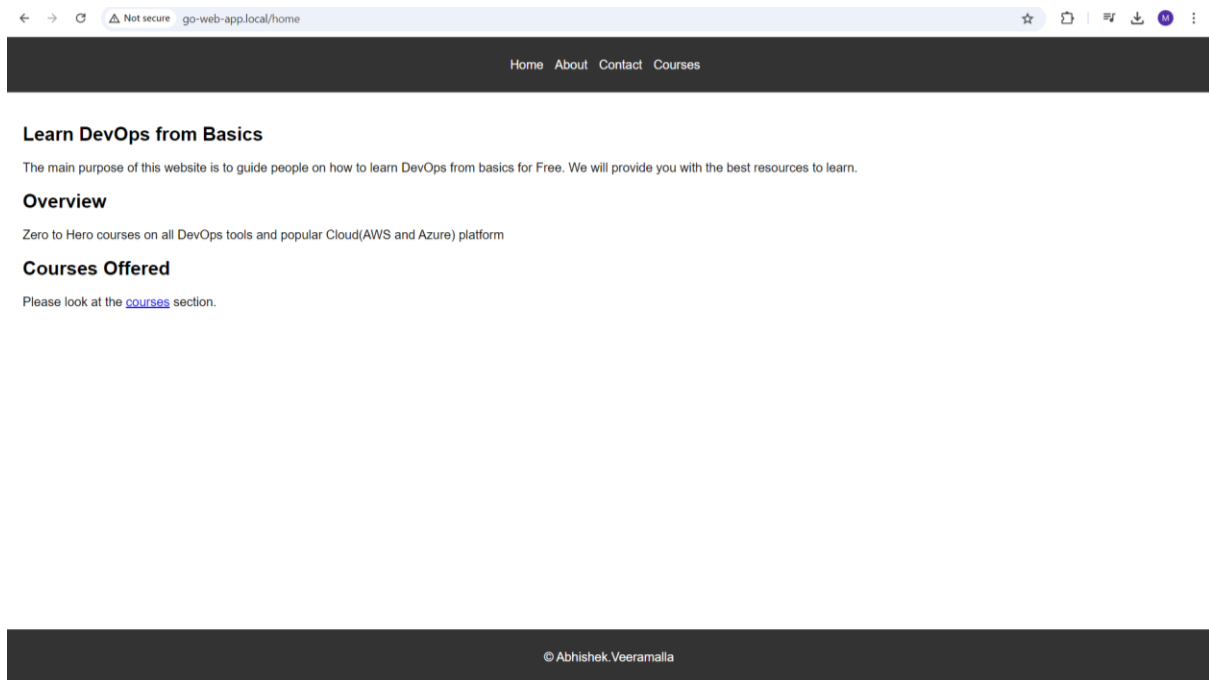
[SIGN IN](#)

argo

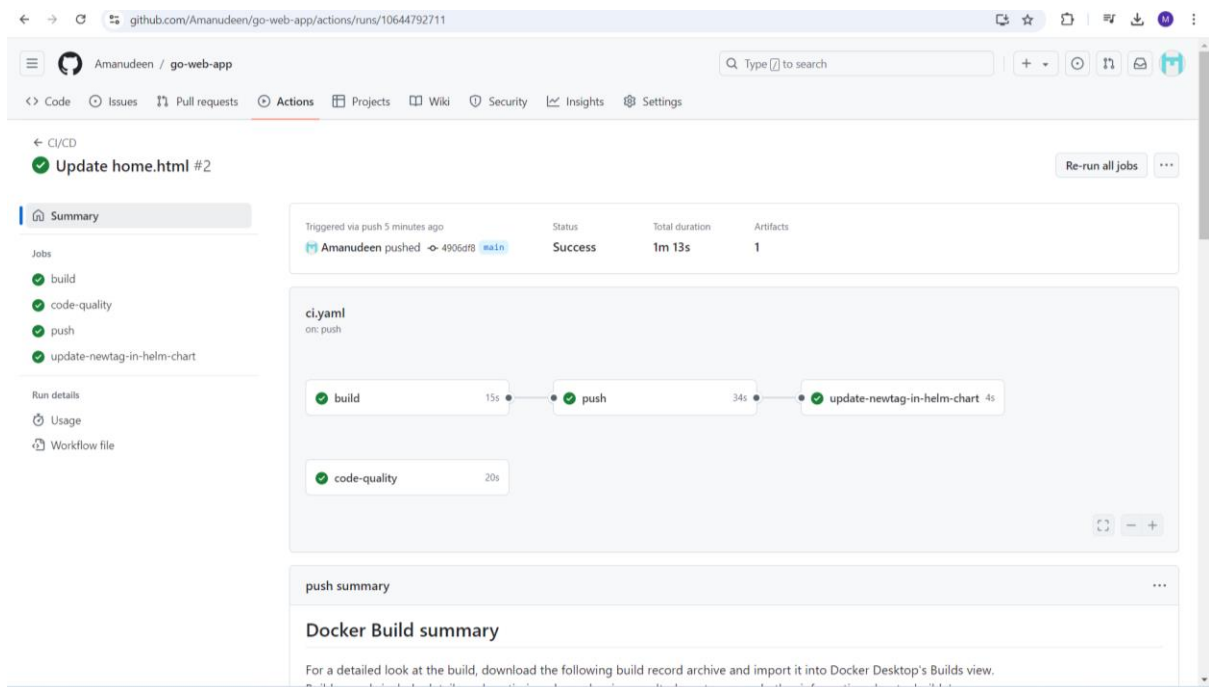


CICD is running successfully

We can have access the portal




## I done some changes in home.html



## New tags created in dockerhub

hub.docker.com/r/amanudeen/go-web-app/tags



**amanudeen/go-web-app** ☆0
 [Manage Repository](#)

By amanudeen · Updated 5 minutes ago

↓ Pulls 12

Overview **Tags**

Sort by **Newest** Filter Tags

TAG	Digest	OS/ARCH	Last pull	Compressed Size
10644792711 Last pushed 5 minutes ago by amanudeen	20e83b3b07ba	linux/amd64	---	13.14 MB
10644662564 Last pushed 28 minutes ago by amanudeen	a5623bf47eef	linux/amd64	11 minutes ago	13.14 MB

My changes appeared in home page

Not secure go-web-app.local/home

Home About Contact Courses

## Learn DevOps from Basics updated

The main purpose of this website is to guide people on how to learn DevOps from basics for Free. We will provide you with the best resources to learn.

### Overview

Zero to Hero courses on all DevOps tools and popular Cloud(AWS and Azure) platform

### Courses Offered

Please look at the [courses](#) section.

© Abhishek Veeramalla

**DevOpsified | Complete DevOps Implementation in one project**

**Completed sucessfully**