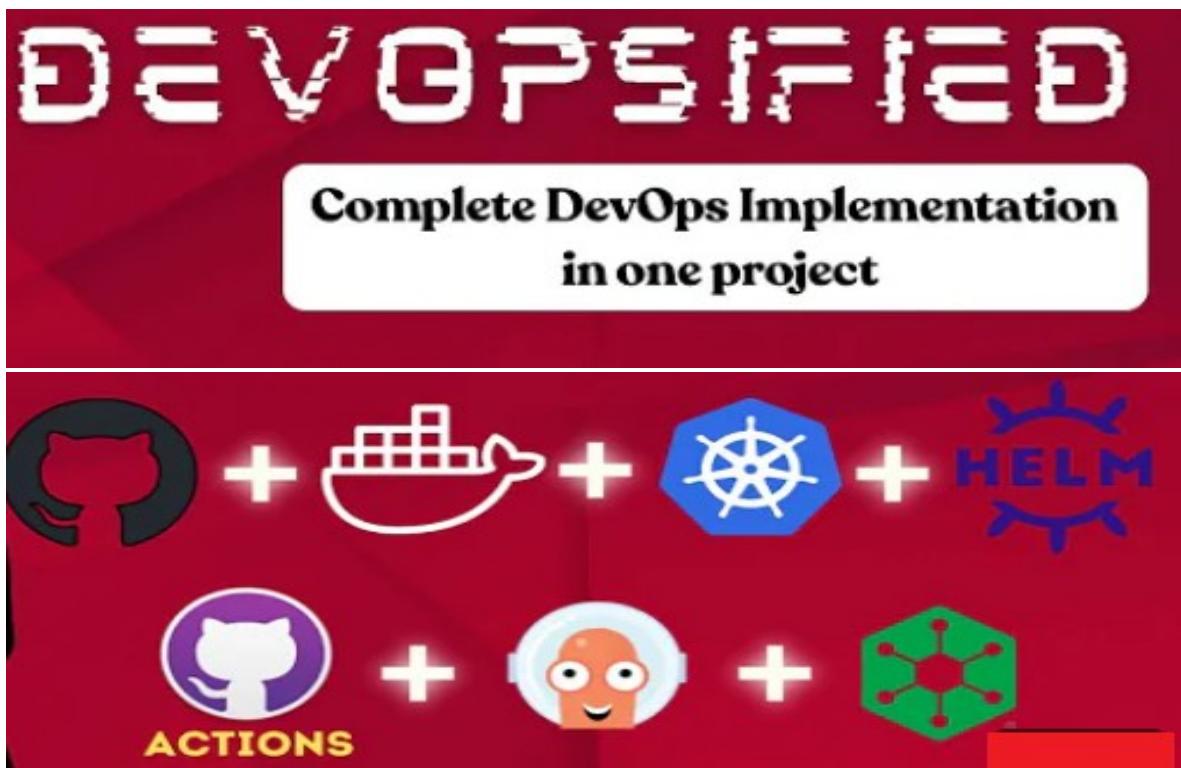


**End to End DevOps on a Golang web application. I will implement the following things**

- Containerization (Multi Stage Docker Build)
- Creating Kubernetes Manifests
- Continuous Integration using GitHub Actions
- Continuous Delivery using Argo CD
- Kubernetes Cluster creation and setup
- Helm chart creation and configuration for multiple environments
- Ingress controller creation, configuration to expose application
- DNS mapping for our domain
- End to End CI/CD demonstration



```
# install – go app - 1st Step  
# Builld the Binary go - 2nd step  
# run the binary Go --- 3rd Step
```

A screenshot of a terminal window titled "ws1host". The window shows a command-line session where a user is navigating to a directory and attempting to build a Go application. The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL (which is selected), along with other icons for PORTS and ESP-IDF. The command "go build -o main ." is entered, followed by an error message about the 'go' command not being found. The user then runs "go build -o main ." again, and finally executes "./main".

```
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects__/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP  
/go-web-app$ go build -o main .  
Command 'go' not found, but can be installed with:  
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects__/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ go build -o main .  
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects__/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ ./main
```

Localhot:8080

The screenshot shows a Microsoft Edge browser window with a dark theme. The address bar displays 'localhost:8080/about'. The page content includes a heading 'About My portfolio', a brief introduction, and a list of skills and interests. At the bottom, there is a link to a GitHub repository.

Hi Everyone, My name is Siddhartha from Bangalore :)

I am a AWS Cloud devops enthusiast and a great believer in sharing knowledge.

I have my keen interest in learning the tradition ML/DL/NLP/Gen-AI LLM models along with AWS -DevOps- Microservices.

For every section of my learning i have demonstrated the demo + real time Projects

- Data Analyst - Python- sql-PowerBI- Advance Python- Matplotlib-seaborn-pandas-numpy
- Traditional /Advance method of ML/DL/NLP/Gen-AI-LLM Models
- DevOps/ SRE Microservices -ArgoCD-docker-k8s-Observability-CI/CD- Gitlab/Github/AWS Codepipeline/Jenkins
- AWS Cloud Services - Migration-Deployment-Compute-Database
- This is my brief about portfolio

If you want more info on my Project repo plscheck out my GitRepo, checkout the playlist

[Github- Repo](#)

The screenshot shows a Microsoft Edge browser window with a dark theme. The address bar displays 'localhost:8080/home'. The page content includes a heading 'my portfolio', a brief introduction, and a section titled 'Overview'. Below 'Overview', there is a note about DevOps tools and the AWS Cloud platform.

The main purpose of this website is to know about my portfolio and to test Devopsified project-go-Web-App- deployment

## Overview

DevOps tools AWS Cloud(AWS) platform

A screenshot of a web browser window. The address bar shows 'localhost:8080/contact'. The page content includes a navigation bar with 'Home' and 'About' links, and a main section titled 'Contact me'. Below this, there is a note: 'For any doubts Pls visit my linkedin Profile' and a link to 'Join Slack'. The overall theme is dark.

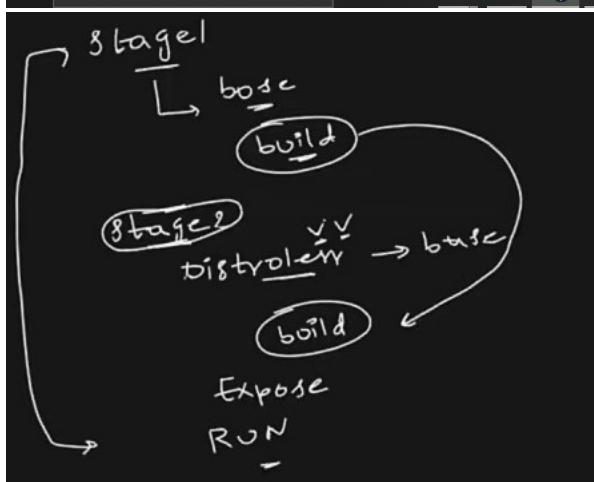
## Contact me

For any doubts Pls visit my linkedin Profile

[Join Slack](#)

# Click on Join slack - my LinkedIn get opened

A screenshot of a LinkedIn feed page. The top navigation bar shows 'Feed | LinkedIn'. The main content area displays a post from 'AWS DevOps Cloud' with the heading 'Hi Siddhartha, are you hiring?'. Below the post are two buttons: 'Yes, I'm hiring' and 'No, not right now'. To the right, there is a 'Trending Now' sidebar with various news items like 'India's export sector' and 'Credit card sales'. At the bottom, there is a search bar and a taskbar with icons for various applications.



# now Start Creating Docker File

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP
b_APP/go-web-app$ ls
Dockerfile LICENSE README.md go.mod main main.go main_test.go static
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP
/go-web-app$ docker build -t siddhartha082/go-web-app:v1 .
[+] Building 1.7s (2/4)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 309B
=> WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 1)
=> [internal] load metadata for docker.io/library/golang:1.21
=> [internal] load metadata for gcr.io/distroless/base:latest
=> [auth] library/golang:pull token for registry-1.docker.io
```

```
FROM golang:1.21 as base
WORKDIR /app
COPY go.mod .
COPY go.sum .
RUN go mod download
RUN go build -o main .
# Final Stage Distroless image
FROM gcr.io/distroless/base
COPY --from=base app/main .
COPY --from=base /app/static ./static
EXPOSE 8080
CMD [ "./main" ]
```

[+] Building 33.2s (7/15)  
=> WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 1)  
=> [base 1/6] FROM docker.io/library/golang:1.21@sha256:4746d26432a9117a5f58e95cb9f954ddf0de128e9d5816886514199316e4a2fb  
=> => sha256:1f46bd02de39f0741d7f614fc607bf03c1a0cd6d9d52a7dd06c44f8fb9358709 85.98MB / 92.23MB  
=> => sha256:1ed03aa58a52c9ab1ee472f1ac74b73d3adcccc2c30744498fd5f14f3f5d22c 56.62MB / 64.14MB  
=> => sha256:3ccbe86a28c2f6b3c3e0e8c6dcfb3a39e1ea656c78da76b6e789e0fe2105982b 24.05MB / 24.05MB  
=> => sha256:903681d8777d28dc56866a07a2774c3fd5b65fd734024c9de0cd9a13c9f636 28.31MB / 49.55MB  
=> => extracting sha256:3214acf345c8cc6bbdb5b698a41ccdefc624a09d6beb0d38b5de0b2303ecaf4  
=> => extracting sha256:5664ab15f108bf9436ce3312090a767300880edb7df4511a1a1ad6d4357024d5dd  
=> => extracting sha256:0bab15eea81d0fe6ab56ebf5fba14e02c4c1775a7f7436fbdd3505add4e18fa  
=> => extracting sha256:4aaea1413d37a58615488592a0b827ea4b2e48fa5a77cf707d0e35f025e13f  
=> => extracting sha256:da7816fa955ea24533c388143c78804c28682eeff9b9a4ee3723b548c70148ba6  
=> => extracting sha256:9aee425378dzcl6cd44177dc54a274b312897f5860a8e78fdfda55aa0d79d71  
=> => extracting sha256:0ee8e7084be533dbd7a0c28c749b81000b74a688d6b16d72888359a69ab31d  
=> => extracting sha256:f823d6cf5f751e7a04c19f387d0f855af517303accf17f49811a9d25d7109a

## # Image Build

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ docker images
REPOSITORY          TAG           IMAGE ID        CREATED         SIZE
siddhartha082/go-web-app   v1           9170efd33aa6  About a minute ago  45.9MB
gcr.io/k8s-minikube/kicbase  v0.0.45    e7c9bc3bc515  2 months ago   1.81GB
gcr.io/k8s-minikube/kicbase  <none>     81df28859520  2 months ago   1.81GB
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$
```

## # Run the Container

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ docker images
REPOSITORY          TAG           IMAGE ID        CREATED         SIZE
siddhartha082/go-web-app   v1           9170efd33aa6  4 minutes ago  45.9MB
gcr.io/k8s-minikube/kicbase  v0.0.45    e7c9bc3bc515  2 months ago   1.81GB
gcr.io/k8s-minikube/kicbase  <none>     81df28859520  2 months ago   1.81GB
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ docker run -p 8080:8080 -it siddhartha082/go-web-app:v1
```

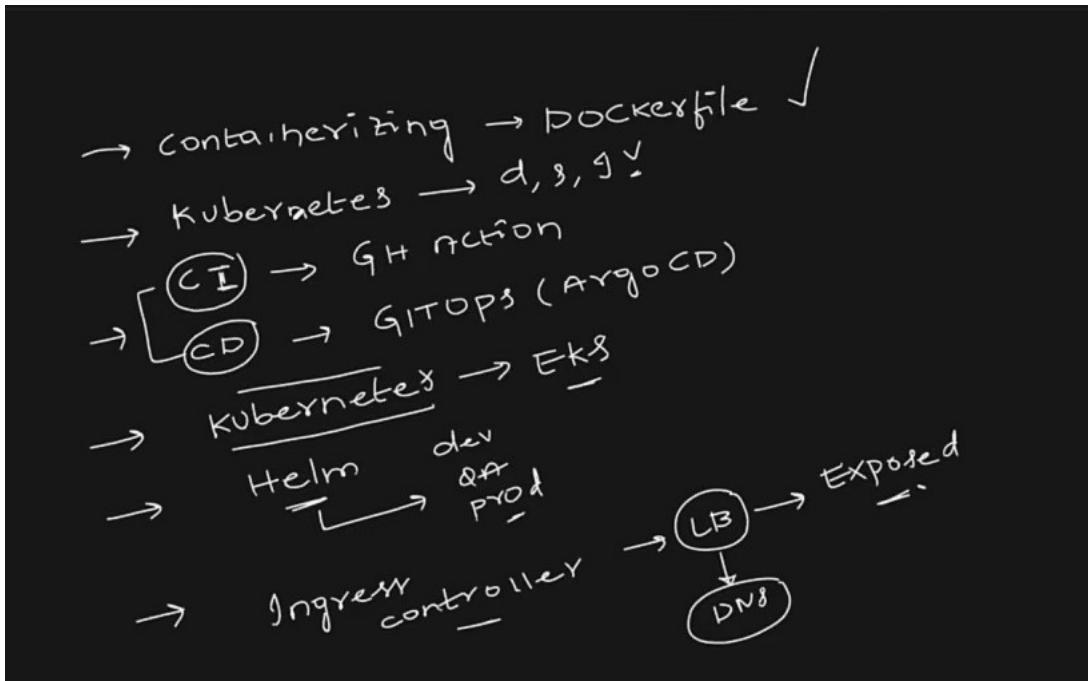
# app -Running

The screenshot shows a Microsoft Edge browser window with two tabs open:

- About**: The current tab, displaying the "About My portfolio" page. The content includes:
  - Hi Everyone, My name is Siddhartha from Bangalore :)
  - I am a AWS Cloud devops enthusiast and a great believer in sharing knowledge.
  - I have my keen interest in learning the tradition ML/DL/NLP/Gen-AI LLM models along with AWS -DevOps- Microservices.
  - For every section of my learning i have demonstrated the demo + real time Projects
  - Data Analyst - Python- sql-PowerBI- Advance Python- Matplotlib-seaborn-pandas-numpy
  - Traditional /Advance method of ML/DL/NLP/Gen-AI-LLM Models
  - DevOps/ SRE Microservices -ArgoCD-docker-k8s-Observability-CI/CD- Gitlab/Github/AWS Codepipeline/Jenkins
  - AWS Cloud Services - Migration-Deployment-Compute-Database
  - This is my brief about portfolio

If you want more info on my Project repo plscheck out my GitRepo, checkout the playlist  
[Github- Repo](#)
- Home**: The other tab, displaying the "my portfolio" page. The content includes:
  - The main purpose of this website is to know about my portfolio and to test Devopsified project-go-Web-App- deployment

The browser interface includes a top navigation bar with tabs like "DevOpsified | Complete", "Siddhartha082/go-web-0", "Build images | Docker Do", "Docker Hub Container In", and "About". The address bar shows "localhost:8080/about" and "localhost:8080/home". The status bar at the bottom shows "© Siddhartha", "JPY/INR +0.48%", "14:43", "08-11-2024", and "ENG".



## # Docker Push

```
siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects___/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ docker push siddhartha082/go-web-app:v1
The push refers to repository [docker.io/siddhartha082/go-web-app]
3214acf345c0: Waiting
a62778643d56: Pushed
7c12895b777b: Pushed
bfb59bb82a9b6: Pushed
44629ccb8b1: Pushed
0bab15eea81d: Pushed
a8f178eb3da3: Pushed
da7816fa955e: Pushed
```

DevOpsified | Complete DevOp x Siddhartha082/go-web-app x siddhartha082/go-web-app gen +

hub.docker.com/repository/docker/siddhartha082/go-web-app/general

\* New More Docker. Easy Access. New Streamlined Plans. Learn more. →

**dockerhub** Explore Repositories Organizations Usage Search Docker Hub ctrl+K

siddhartha082 / [Repositories](#) / [go-web-app](#) / General Using 0 of 1 private repositories.

[General](#) Tags Builds Collaborators Webhooks Settings

**siddhartha082/go-web-app** ⓘ Last pushed less than a minute ago

This repository does not have a description ⚡ ⓘ INCOMPLETE

This repository does not have a category ⚡ ⓘ INCOMPLETE

**Docker commands** To push a new tag to this repository: [Public View](#)

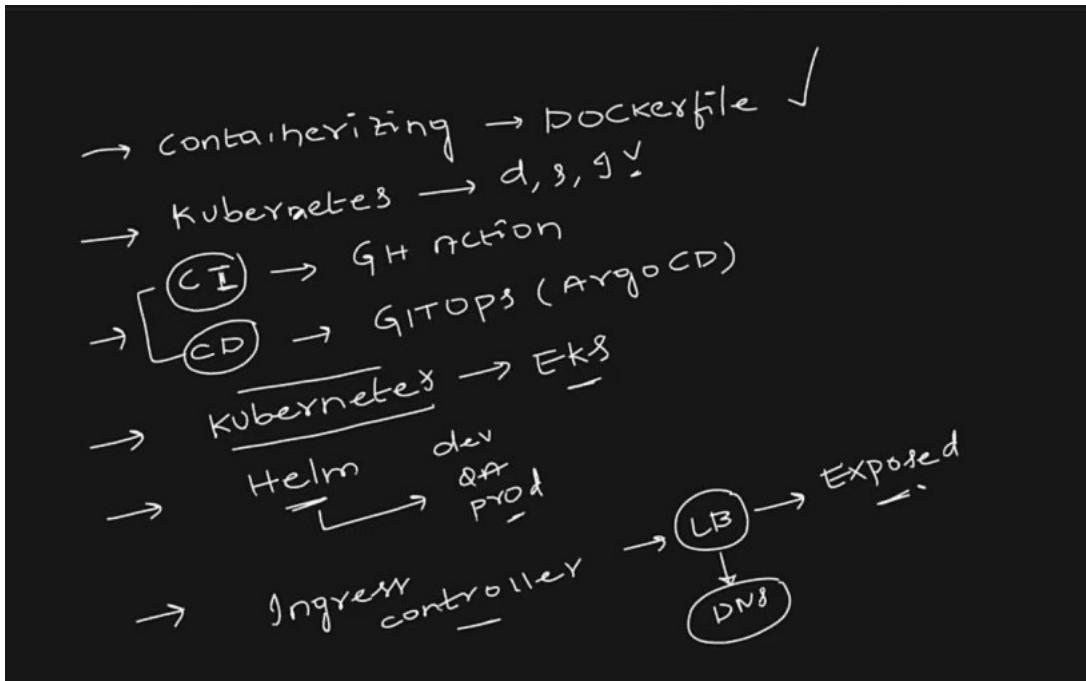
```
docker push siddhartha082/go-web-app:tagname
```

**Tags** This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed

**Automated Builds** Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Windows Type here to search ⚡ 27°C Mostly sunny 14:50 ENG 08-11-2024



# 1<sup>st</sup> Stage Docker img Created – Run- Push to the Docker – Hub

# 2<sup>nd</sup> Stage Create & Deploy the Kubernetes / manifests - deployment + service + ingress- yaml files

The screenshot shows a browser window with several tabs open, including 'DevOpsified | Complete DevOp', 'go-web-app-devops/k8s/manifests', and 'Siddhartha082/go-web-app'. The active tab is 'Deployments | Kubernetes'. The URL in the address bar is 'kubernetes.io/docs/concepts/workloads/controllers/deployment/'.

**Kubernetes Documentation**

**Deployments**

A Deployment manages a set of Pods to run an application workload, usually one that doesn't maintain state.

**Note:**  
Do not manage ReplicaSets owned by a Deployment. Consider opening an issue in the main Kubernetes repository if your use case is not covered

**Right Sidebar:**

- Deployment API reference
- Edit this page
- Create child page
- Create documentation issue
- Print entire section
- Use Case
- Creating a Deployment
- Pod-template-hash label
- Updating a Deployment
- Rollover (aka multiple updates in-flight)
- Label selector updates
- Rolling Back a Deployment
- Checking Rollout History of a Deployment
- Rolling Back to a Previous Revision
- Scaling a Deployment
- Proportional scaling
- Pausing and Resuming a rollout of a Deployment

The screenshot shows a Windows desktop environment with two windows open:

- Top Window (Browser):** Displays the Kubernetes documentation page for "Deployment controllers". The URL is [kubernetes.io/docs/concepts/workloads/controllers/deployment/](https://kubernetes.io/docs/concepts/workloads/controllers/deployment/). The page content includes a search bar and a sidebar with navigation links for Documentation, Getting started, Concepts, Workloads, Deployments, and more. A code snippet titled "nginx-deployment.yaml" is shown, defining a Deployment object with three replicas of an Nginx application.
- Bottom Window (Code Editor):** Shows a code editor interface with multiple tabs. The active tab is "deployment.yaml" under the "GO-WEB-APP" project. The file content is a Deployment manifest for a "go-web-app" service, specifying one replica, a selector, and a template with a single container named "go-web-app" running the image "siddhartha082/go-web-app:v1" on port 8080. Other tabs in the editor include "Welcome", "Dockerfile U", and "ESP-IDF: Search Error Hint". The left sidebar of the code editor shows the project structure with files like .gitignore, Dockerfile, go.mod, LICENSE, main.go, and README.md.

The screenshot shows a Microsoft Edge browser window with the URL [kubernetes.io/docs/concepts/services-networking/service/](https://kubernetes.io/docs/concepts/services-networking/service/). The page title is "Service". The left sidebar has a search bar and a navigation menu with sections like Documentation, Getting started, Concepts (Overview, Cluster Architecture, Containers, Workloads, Services, Load Balancing, Networking), and Service (Ingress, Ingress Controllers, Gateway API, EndpointSlices, Network Policies, DNS for Services and Pods, IPv4/IPv6 dual-stack, Topology Aware Routing). The main content area discusses what a Service is, how it exposes applications, and its role in load balancing. It includes a sidebar with links to Service API reference, edit this page, create child page, create documentation issue, print entire section, and various service types (ClusterIP, NodePort, LoadBalancer, ExternalName, Headless Services, With selectors, Without selectors). The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows a Microsoft Edge browser window with the URL [kubernetes.io/docs/concepts/services-networking/service/](https://kubernetes.io/docs/concepts/services-networking/service/). The page title is "Defining a Service". The left sidebar has a search bar and a navigation menu with sections like Documentation, Getting started, Concepts (Overview, Cluster Architecture, Containers, Workloads, Services, Load Balancing, Networking), and Service (Ingress, Ingress Controllers, Gateway API, EndpointSlices, Network Policies, DNS for Services and Pods, IPv4/IPv6 dual-stack, Topology Aware Routing). The main content area explains how to define a Service using a YAML manifest. A code block shows the following YAML:

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app.kubernetes.io/name: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

Text below the code block states: "Applying this manifest creates a new Service named "my-service" with the default ClusterIP service type. The Service targets TCP port 9376 on any Pod with the app.kubernetes.io/name: MyApp label." The bottom of the screen shows the Windows taskbar with various pinned icons.

```

1 # Service for the application
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: go-web-app
6   labels:
7     app: go-web-app
8 spec:
9   ports:
10    - port: 80
11      targetPort: 8080
12      protocol: TCP
13   selector:
14     app: go-web-app
15   type: ClusterIP
16

```

[KubeCon + CloudNativeCon 2024](#)

Join us for three days of incredible opportunities to collaborate, learn and share with the cloud native community.  
Buy your ticket now! 12 – 15 November | Salt Lake City

[Ingress API reference](#)  
[IngressClass API reference](#)  
[Edit this page](#)  
[Create child page](#)  
[Create documentation issue](#)  
[Print entire section](#)

[What is Ingress?](#)  
[Prerequisites](#)  
[The Ingress resource](#)  
[Ingress rules](#)  
[DefaultBackend](#)  
[Resource backends](#)  
[Path types](#)  
[Examples](#)  
[Hostname wildcards](#)  
[Ingress class](#)

# I use Host + path based ingress yaml file

The screenshot shows a Windows desktop environment with two main windows open:

- Browser Window:** Displays the Kubernetes documentation page for Ingress at <https://kubernetes.io/docs/concepts/services-networking/ingress/>. The page content shows a YAML manifest for an Ingress resource with two rules: one for "foo.bar.com" mapping to service1 and another for "\*.foo.com" mapping to service2.
- VS Code Editor:** Shows a file named "ingress.yaml" containing the same YAML manifest. The code editor interface includes a sidebar with icons for file operations, a search bar, and tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and ESP-IDF. The terminal tab shows the command "siddhartha@Siddhartha:/mnt/d/2\_AWS\_Devops\_Projects\_\_\_\_/0006\_DevOps\_notes\_Proj----/10\_Devopsified\_Project/1\_Go\_Web\_APP/go-web-app\$".

At the bottom of the screen, the taskbar shows the Windows Start button, a search bar, and pinned icons for various applications like File Explorer, Edge, and File History. The system tray indicates the date as 08-11-2024, the time as 15:09, and the weather as 27°C Partly sunny.

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-wildcard-host
spec:
  rules:
    - host: "foo.bar.com"
      http:
        paths:
          - pathType: Prefix
            path: "/bar"
            backend:
              service:
                name: service1
                port:
                  number: 80
    - host: "*.foo.com"
      http:
        paths:
          - pathType: Prefix
            path: "/foo"
            backend:
              service:
                name: service2
                port:
                  number: 80
```

```
# Ingress resource for the application
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: go-web-app
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  ingressClassName: nginx
  rules:
    - host: go-web-app.local
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: go-web-app
                port:
                  number: 80
```

# Time to Test Kubernetes Manifest

```

1 # Service for the application
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: go-web-app
6   labels:
7     app: go-web-app
8 spec:
9   ports:
10    - port: 80
11      targetPort: 8080
12      protocol: TCP
13   selector:
14     app: go-web-app
15   type: ClusterIP
16

```

# To Run the Kuberneeted Cluster we use EKS Cluster

## prerequisites

kubectl – A command line tool for working with Kubernetes clusters. For more information, see [Installing or updating kubectl](#).

eksctl – A command line tool for working with EKS clusters that automates many individual tasks. For more information, see [Installing or updating eksctl](#).

AWS CLI – A command line tool for working with AWS services, including Amazon EKS. For more information, see [Installing, updating, and uninstalling the AWS CLI in the AWS Command Line Interface User Guide](#). After installing the AWS CLI, we recommend that you also configure it. For more information, see [Quick configuration with aws configure](#) in the AWS Command Line Interface User Guide.

### To install or upgrade eksctl on Linux using curl

1. Download and extract the latest release of eksctl with the following command.

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

2. Move the extracted binary to /usr/local/bin.

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

3. Test that your installation was successful with the following command.

```
eksctl version
```

# Ubuntu

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

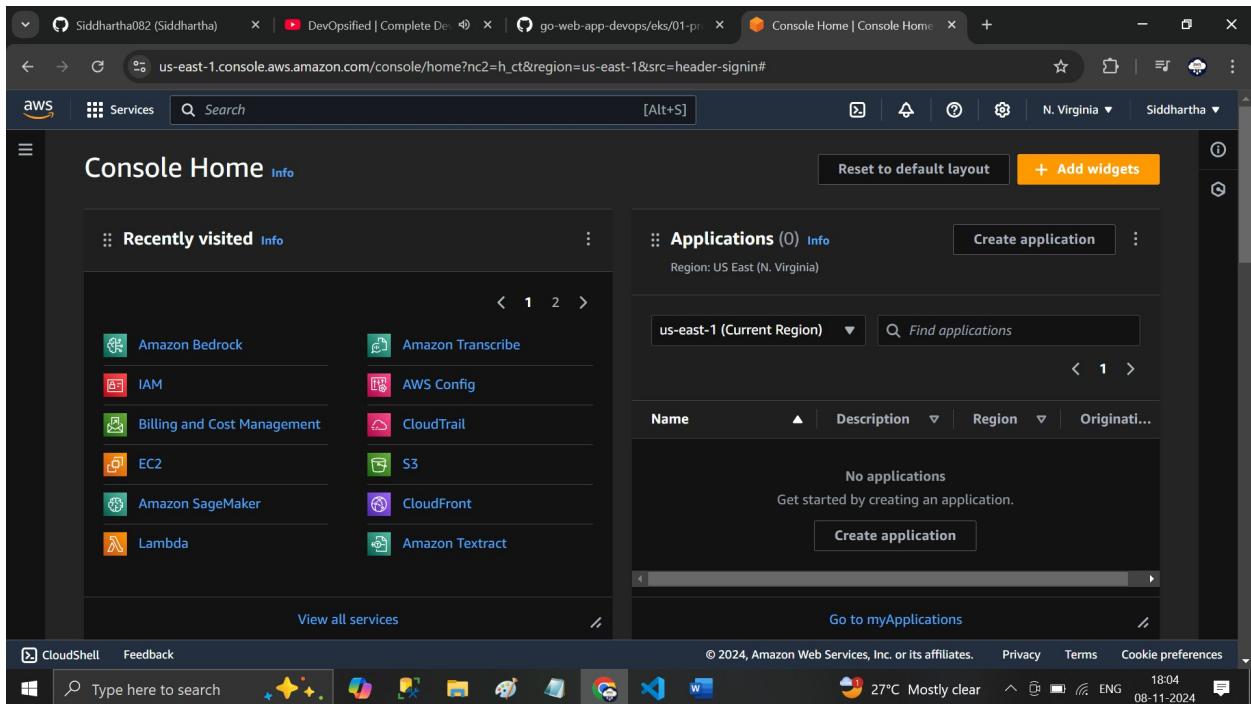
```
sudo mv /tmp/eksctl /usr/local/bin
```

**eksctl version**

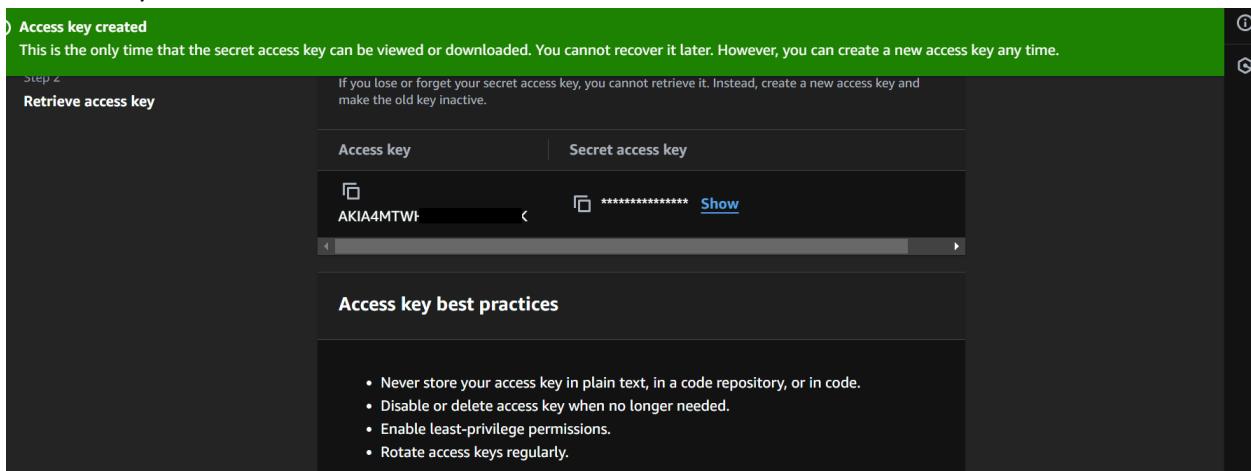
# AWS CLI –

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP
/go-web-app$ aws --version
aws-cli/1.22.34 Python/3.10.12 Linux/5.15.133.1-microsoft-standard-WSL2 botocore/1.34.72
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP
/go-web-app$
```

# Aws Console



# access key



```
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP
/go-web-app$ aws configure
AWS Access Key ID [*****GZUF]: AKIA4MI          HXK
AWS Secret Access Key [*****QxPq]: VPXv3           :+f/SL7R
Default region name [us-east-1]: us-east-1
Default output format [json]: json
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP
/go-web-app$
```

# Test aws cli connection to aws cloud(my account)

```
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP
/go-web-app$ aws sts get-caller-identity
{
    "UserId": "851725192411",
    "Account": "851725192411",
    "Arn": "arn:aws:iam::851725192411:root"
}
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP
/go-web-app$
```

The screenshot shows a GitHub repository page titled 'Install EKS'. It contains two main sections: 'Install EKS' and 'Install a EKS cluster with EKSCTL'. Each section has a code block showing the command 'eksctl create cluster --name demo-cluster --region us-east-1'. Below these sections is a 'Delete the cluster' section with a code block showing the command 'eksctl delete cluster --name demo-cluster --region us-east-1'. At the bottom of the screenshot, the Windows taskbar is visible, showing various icons and system status.

# Creating EK Cluster

# Note – have patience to install this cluster into Aws Cloud .. till completion of the installation do not work on any other toll

The screenshot shows the VS Code interface with the Dockerfile tab selected. The terminal window displays the command `eksctl create cluster --name demo-cluster --region us-east-1` being run. The output shows the creation of a cluster named "demo-cluster" in the "us-east-1" region, including the creation of a nodegroup "ng-a6e27579" using AmazonLinux2/1.30. It also mentions the creation of two separate CloudFormation stacks for the cluster itself and initial managed nodes.

```

FROM golang:1.22.5 as base
WORKDIR /app
COPY go.mod .

siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP
/go-web-app$ eksctl create cluster --name demo-cluster --region us-east-1
2024-11-09 19:59:22 [i] eksctl version 0.194.0
2024-11-09 19:59:22 [i] using region us-east-1
2024-11-09 19:59:24 [i] setting availability zones to [us-east-1d us-east-1c]
2024-11-09 19:59:24 [i] subnets for us-east-1d - public:192.168.0.0/19 private:192.168.64.0/19
2024-11-09 19:59:24 [i] subnets for us-east-1c - public:192.168.32.0/19 private:192.168.96.0/19
2024-11-09 19:59:24 [i] nodegroup "ng-a6e27579" will use "" [AmazonLinux2/1.30]
2024-11-09 19:59:24 [i] using Kubernetes version 1.38
2024-11-09 19:59:24 [i] creating EKS cluster "demo-cluster" in "us-east-1" region with managed nodes
2024-11-09 19:59:24 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed no
degroup
2024-11-09 19:59:24 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-st
acks --region=us-east-1 --cluster=demo-cluster'
2024-11-09 19:59:24 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false
} for cluster "demo-cluster" in "us-east-1"
2024-11-09 19:59:24 [i] CloudWatch logging will not be enabled for cluster "demo-cluster" in "us-east-1"
2024-11-09 19:59:24 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east
-1 --cluster=demo-cluster'
2024-11-09 19:59:24 [i] default addons vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2024-11-09 19:59:24 [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-a6e27579",
  }
}

```

This screenshot is nearly identical to the one above, showing the same command and output for creating an EKS cluster named "demo-cluster" in the "us-east-1" region. The log output is identical, detailing the creation of a nodegroup and the generation of CloudFormation stacks.

```

package main
import (
    "log"
)

/go-web-app$ eksctl create cluster --name demo-cluster --region us-east-1
2024-11-08 18:14:25 [i] eksctl version 0.194.0
2024-11-08 18:14:25 [i] using region us-east-1
2024-11-08 18:14:26 [i] setting availability zones to [us-east-1d us-east-1a]
2024-11-08 18:14:26 [i] subnets for us-east-1d - public:192.168.0.0/19 private:192.168.64.0/19
2024-11-08 18:14:26 [i] subnets for us-east-1a - public:192.168.32.0/19 private:192.168.96.0/19
2024-11-08 18:14:26 [i] nodegroup "ng-52d71727" will use "" [AmazonLinux2/1.30]
2024-11-08 18:14:26 [i] using Kubernetes version 1.38
2024-11-08 18:14:26 [i] creating EKS cluster "demo-cluster" in "us-east-1" region with managed nodes
2024-11-08 18:14:26 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed no
degroup
2024-11-08 18:14:26 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-st
acks --region=us-east-1 --cluster=demo-cluster'
2024-11-08 18:14:26 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false
} for cluster "demo-cluster" in "us-east-1"
2024-11-08 18:14:26 [i] CloudWatch logging will not be enabled for cluster "demo-cluster" in "us-east-1"
2024-11-08 18:14:26 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east
-1 --cluster=demo-cluster'
2024-11-08 18:14:26 [i] default addons vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2024-11-08 18:14:26 [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-52d71727",
  }
}

```

## # Kubectl Working Condition

```

siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl version
Client Version: v1.31.2
Kustomize Version: v5.4.2
Server Version: v1.30.6-eks-7f9249a
siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ 

```

Screenshot of the AWS EKS Clusters console. The left sidebar shows navigation options like 'Clusters', 'Amazon EKS Anywhere', and 'Related services'. The main area displays a table titled 'Clusters (1) Info' with one entry: 'Cluster name: demo-cluster', 'Status: Creating', 'Kubernetes version: 1.30', and 'Support period: Standard support until July 28, 2025'. The bottom status bar shows system information including the date (08-11-2024), time (18:15), and location (N. Virginia).

## # Cloud Formation

Screenshot of the AWS CloudFormation console. The left sidebar shows navigation options like 'Stacks', 'StackSets', and 'Exports'. The main area displays a table titled 'Stacks (1)' with one entry: 'Stack name: eksctl-demo-cluster-cluster', 'Status: CREATE\_IN\_PROGRESS', and 'Created time: 2024-11-08 18:14:27 UTC+0530'. The bottom status bar shows system information including the date (08-11-2024), time (18:17), and location (N. Virginia).

# now Creating Kubernetes Cluster manifest

## # Pods Created

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl apply -f k8s/manifests/deployment.yaml
deployment.apps/go-web-app created
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
go-web-app-66c6b64548-t87nt  1/1     Running   0          26s
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl apply -f k8s/manifests/service/go-web-app
service/go-web-app created
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl apply -f k8s/manifests/ingress.yaml
ingress.networking.k8s.io/go-web-app created
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
go-web-app-66c6b64548-t87nt  1/1     Running   0          108s
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$
```

## # Address not Assign

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get ing
NAME      CLASS      HOSTS      ADDRESS      PORTS      AGE
go-web-app  nginx      go-web-app.local      80      108s
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$
```

## # to Get it ingress controller will help

# now check the svc .. whether it in nodeport mode / Clustermode ??

# It s Cluster IP – highlighted .. Change it to nodeport mode

```
4 RUN go mod download
5 COPY .
6 RUN go build -o main .
7 # Final Stage Distributes image
{"apiVersion": "v1", "kind": "Service", "metadata": {"annotations": {}, "labels": {"app": "go-web-app"}, "name": "go-web-app", "namespace": "default"}, "spec": {"ports": [{"port": 80, "protocol": "TCP", "targetPort": 8080}], "selector": {"app": "go-web-app"}, "type": "ClusterIP"}}

labels:
  app: go-web-app
  name: go-web-app
  namespace: default
  resourceVersion: "6349"
  uid: d28cabe5-bd48-478e-9405-834c5b3b34fc
spec:
  clusterIP: 10.100.192.220
  clusterIPs:
  - 10.100.192.220
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - port: 80
    protocol: TCP
    targetPort: 8080
  selector:
    app: go-web-app
    sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}

34,1 Bot
```

# Change Cluster ip to nodeport mode

```
4 RUN go mod download
5 COPY .
6 RUN go build -o main .
7 # Final Stage Distributes image
{"apiVersion": "v1", "kind": "Service", "metadata": {"annotations": {}, "labels": {"app": "go-web-app"}, "name": "go-web-app", "namespace": "default"}, "spec": {"ports": [{"port": 80, "protocol": "TCP", "targetPort": 8080}], "selector": {"app": "go-web-app"}, "type": "NodePort"}}

labels:
  app: go-web-app
  name: go-web-app
  namespace: default
  resourceVersion: "6349"
  uid: d28cabe5-bd48-478e-9405-834c5b3b34fc
spec:
  clusterIP: 10.100.192.220
  clusterIPs:
  - 10.100.192.220
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - port: 80
    protocol: TCP
    targetPort: 8080
  selector:
    app: go-web-app
    sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}

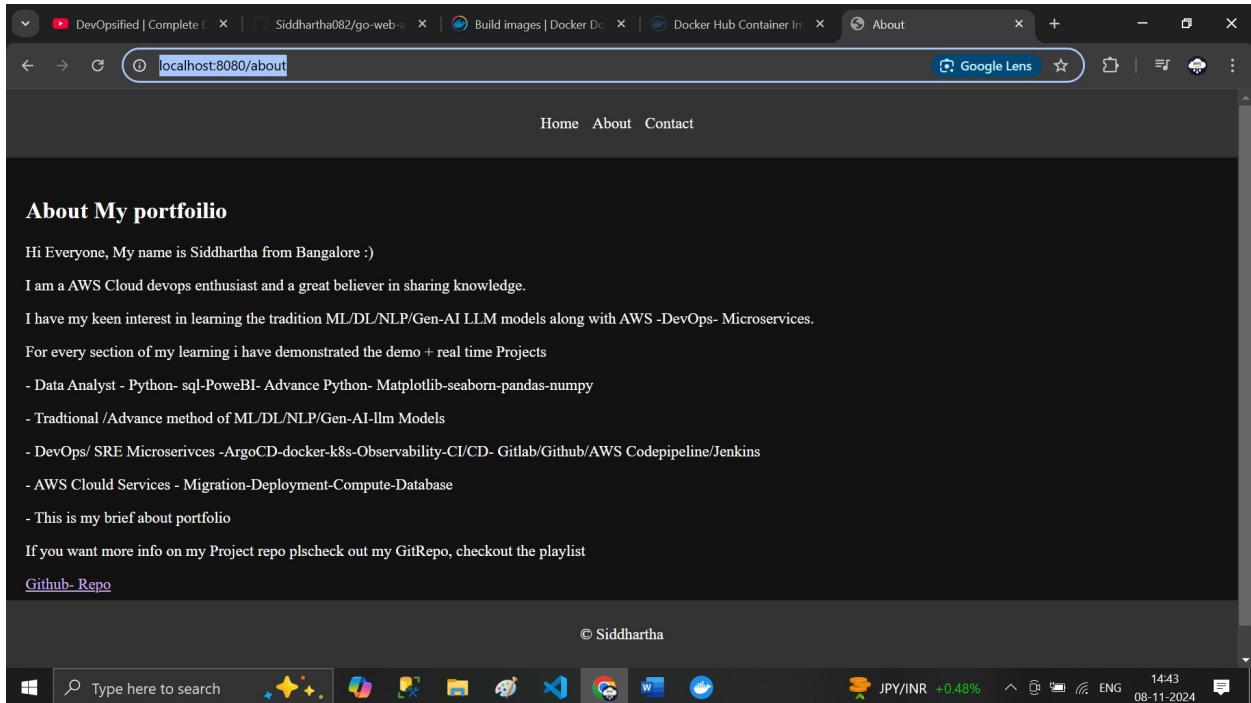
-- INSERT --
```

# now My svc is running in nodePort mode

```

siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl edit svc go-web-app
service/go-web-app edited
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get svc
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
go-web-app   NodePort    10.100.192.220  <none>          80:32517/TCP  11m
kubernetes   ClusterIP  10.100.0.1     <none>          443/TCP       42m
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get nodes -o wide
NAME           STATUS   ROLES   AGE   VERSION
ip-192-168-38-129.ec2.internal   Ready   <none>   35m   v1.30.4-eks-a737599   192.168.30.129   54.159.42.202   Amazon Linux 2   5.10.227-219.884.amzn2.x86_6
4   containerd://1.7.22
ip-192-168-54-23.ec2.internal   Ready   <none>   35m   v1.30.4-eks-a737599   192.168.54.23    3.93.148.90    Amazon Linux 2   5.10.227-219.884.amzn2.x86_6
4   containerd://1.7.22
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ 

```



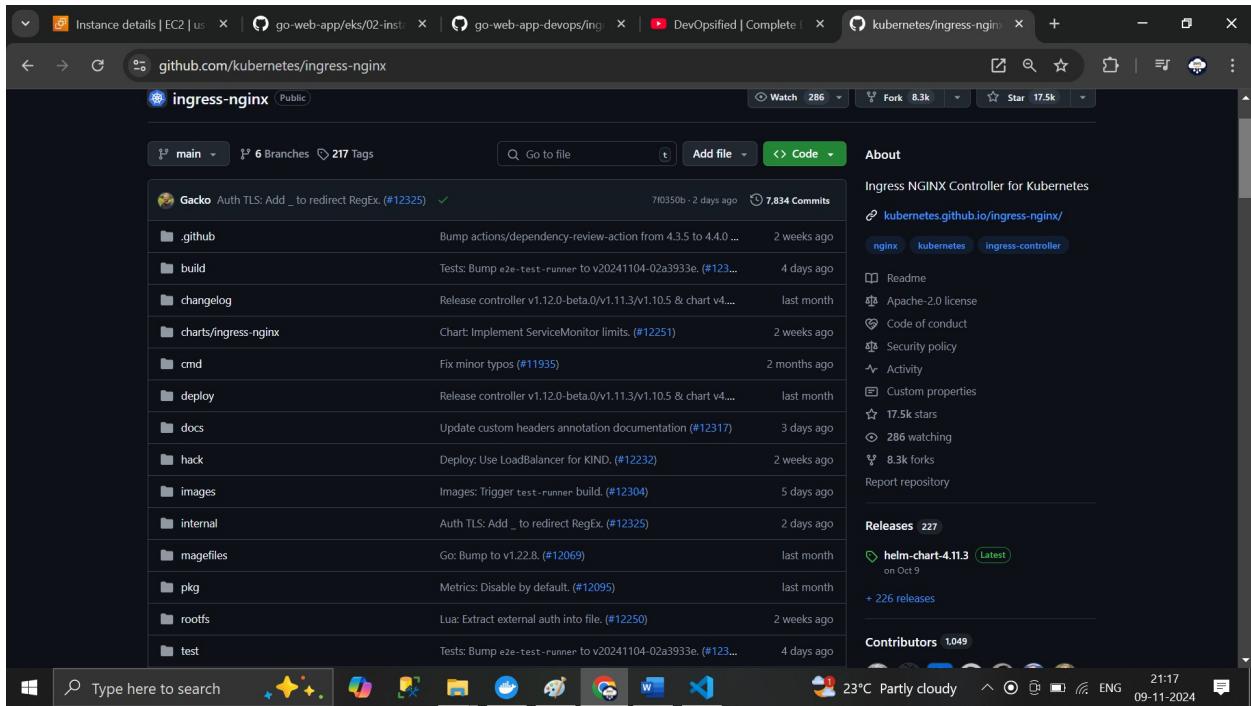
# ingress Controller Configuration

## Install Nginx Ingress Controller on AWS

### Step 1: Deploy the below manifest

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.11.1/deploy/static/provider/aws/deploy.yaml
```

[## <https://github.com/kubernetes/ingress-nginx>](https://github.com/kubernetes/ingress-nginx)



## # Getting Started

The screenshot shows the GitHub README page for the ingress-nginx repository. Key sections include:

- Overview:** ingress-nginx is an Ingress controller for Kubernetes using NGINX as a reverse proxy and load balancer.
- Get started:** See the [Getting Started](#) document. Note: Do not use in multi-tenant Kubernetes production installations.
- Troubleshooting:** If you encounter issues, review the [troubleshooting docs](#), [file an issue](#), or talk to us on the [#ingress-nginx channel](#) on the Kubernetes Slack server.
- Changelog:** See the [list of releases](#) for all changes. For detailed changes for each release, please check the `changelog-$version.md` file for the release version. For detailed changes on the `ingress-nginx` helm chart, please check the `changelog` folder for a specific version. [CHangelog-\\$current-version.md](#).
- Supported Versions table:** Supported versions for the ingress-nginx project mean that we have completed E2E tests, and they are passing for the versions listed. Ingress-Nginx versions **may** work on older versions, but the project does not make that guarantee.

<https://kubernetes.github.io/ingress-nginx/deploy/>

Instance details | EC2 | us | go-web-app/eks/02-inst | go-web-app-devops/mg | DevOpsified | Complete | Installation Guide - Ingress-Nginx

Ingress-Nginx Controller

Deployment Installation Guide Bare-metal considerations Role Based Access Control (RBAC) Upgrade Hardening guide

## Installation Guide

There are multiple ways to install the Ingress-Nginx Controller:

- with [Helm](#), using the project repository chart;
- with [kubectl apply](#), using YAML manifests;
- with specific add-ons (e.g. for [minikube](#) or [MicroK8s](#)).

On most Kubernetes clusters, the ingress controller will work without requiring any extra configuration. If you want to get started as fast as possible, you can check the [quick start](#) instructions. However, in many environments, you can improve the performance or get better logs by enabling extra features. We recommend that you check the [environment-specific instructions](#) for details about optimizing the ingress controller for your particular environment or cloud provider.

Table of contents

Contents Quick start Firewall configuration Pre-flight check Local testing Online testing Environment-specific instructions Local development clusters minikube MicroK8s Docker Desktop Rancher Desktop Cloud deployments AWS Network Load Balancer (NLB) TLS termination in AWS Load Balancer (NLB) NLB Idle Timeouts GCE-GKE Azure Digital Ocean Scaleway

Type here to search 21:18 09-11-2024

Instance details | EC2 | us | go-web-app/eks/02-inst | go-web-app-devops/mg | DevOpsified | Complete | Installation Guide - Ingress-Nginx

Ingress-Nginx Controller

Deployment Installation Guide Bare-metal considerations Role Based Access Control (RBAC) Upgrade Hardening guide

## Installation Guide

• [Quick start](#)

• [Environment-specific instructions](#)

• ... [Docker Desktop](#)

• ... [Rancher Desktop](#)

• ... [minikube](#)

• ... [MicroK8s](#)

• ... [AWS](#)

• ... [GCE-GKE](#)

• ... [Azure](#)

• ... [Digital Ocean](#)

• ... [Scaleway](#)

• ... [Exoscale](#)

• ... [Oracle Cloud Infrastructure](#)

• ... [OVHcloud](#)

• ... [Bare-metal](#)

• [Miscellaneous](#)

Table of contents

Contents Quick start Firewall configuration Pre-flight check Local testing Online testing Environment-specific instructions Local development clusters minikube MicroK8s Docker Desktop Rancher Desktop Cloud deployments AWS Network Load Balancer (NLB) TLS termination in AWS Load Balancer (NLB) NLB Idle Timeouts GCE-GKE Azure Digital Ocean Scaleway Exoscale

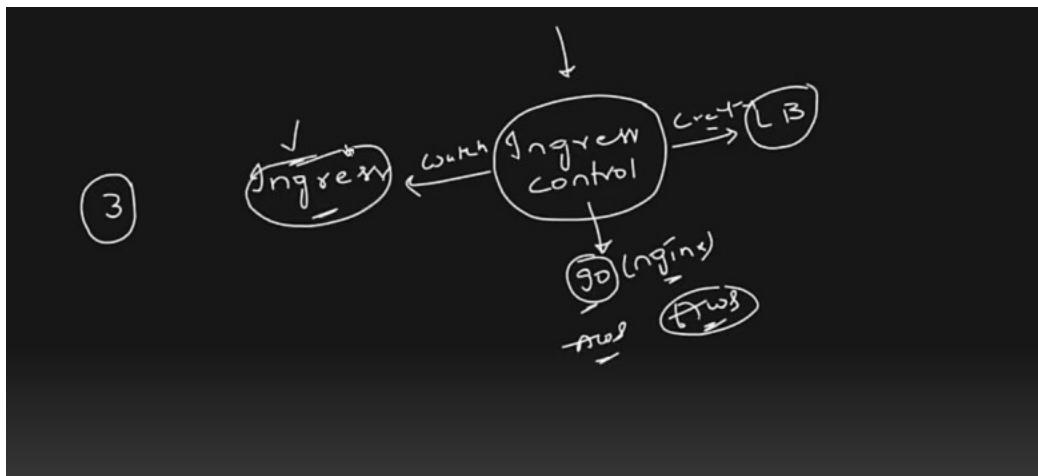
Type here to search 21:18 09-11-2024

# aws

<https://kubernetes.github.io/ingress-nginx/deploy/#aws>

Screenshot of a web browser showing the Kubernetes Ingress-Nginx deployment guide for AWS. The page details the setup of an Ingress-Nginx Controller behind a Network Load Balancer (NLB) in AWS. It includes sections on TLS termination, VPC CIDR configuration, and AWS Certificate Manager (ACM) integration. The browser taskbar shows multiple tabs related to DevOps and AWS.

```
siddhartha@siddhartha:/mnt/d/2_AWS_DevOps_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.11.1/deploy/static/provider/aws/deploy.yaml
namespace/ingress-nginx created
serviceaccount/ingress-nginx created
serviceaccount/ingress-nginx-admission created
role.rbac.authorization.k8s.io/ingress-nginx created
role.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrole.rbac.authorization.k8s.io/ingress-nginx created
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission created
rolebinding.rbac.authorization.k8s.io/ingress-nginx created
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx created
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission created
configmap/ingress-nginx-controller created
service/ingress-nginx-controller created
service/ingress-nginx-controller-admission created
deployment.apps/ingress-nginx-controller created
job.batch/ingress-nginx-admission-create created
job.batch/ingress-nginx-admission-patch created
ingressclass.networking.k8s.io/nginx created
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission created
siddhartha@siddhartha:/mnt/d/2_AWS_DevOps_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$
```



A screenshot of the Visual Studio Code interface. The top bar shows the file path: go-web-app. The left sidebar has icons for file, edit, selection, view, go, run, and search. The main area has tabs for Welcome, Dockerfile, and ESP-IDF. The Dockerfile tab shows the following code:

```
4 RUN go mod download
5 COPY .
6 RUN go build -o main .
7 # Final Stage Distributes image
```

The terminal tab is active and displays the following command output:

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get all -n ingress-nginx
NAME                                     READY   STATUS    RESTARTS   AGE
pod/ingress-nginx-admission-create-gt2n6   0/1     Completed   0          5m5s
pod/ingress-nginx-admission-patch-4bjdm    0/1     Completed   0          5m4s
pod/ingress-nginx-controller-666487-9k9x2  1/1     Running    0          5m5s

NAME                           TYPE      CLUSTER-IP       EXTERNAL-IP
PORT(S)                      AGE
service/ingress-nginx-controller   LoadBalancer   10.100.245.196   a29621a020a9a4f41b04d02e7c9097b7-0a5fb0f940721375.elb.us-east-1.amazonaws.com
80:31376/TCP, 443:30895/TCP   5m8s
service/ingress-nginx-controller-admission ClusterIP   10.100.180.229   <none>
443/TCP                      5m7s

NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/ingress-nginx-controller   1/1     1           1           5m6s

NAME                      DESIRED   CURRENT   READY   AGE
replicaset.apps/ingress-nginx-controller-666487  1        1         1         5m7s

NAME                      STATUS    COMPLETIONS   DURATION   AGE
job.batch/ingress-nginx-admission-create   Complete   1/1        5s        5m8s
job.batch/ingress-nginx-admission-patch   Complete   1/1        4s        5m7s
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$
```

The status bar at the bottom shows the following information: main\*, Type here to search, 23°C Partly cloudy, 21:25, 09-11-2024.

A screenshot of a terminal window showing the following command output:

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get pods -n ingress-nginx
NAME                                     READY   STATUS    RESTARTS   AGE
ingress-nginx-admission-create-gt2n6   0/1     Completed   0          21m
ingress-nginx-admission-patch-4bjdm    0/1     Completed   0          21m
ingress-nginx-controller-666487-9k9x2  1/1     Running    0          21m
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl edit pod ingress-nginx-controller-666487-9k9x2 -n ingress-nginx
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$
```

The screenshot shows the Visual Studio Code interface with two open files:

- Dockerfile**:

```
4 RUN go mod download
5 COPY .
6 RUN go build -o main .
7 # Final Stage Distributes image
```
- ingress.yaml**:

```
controller: true
kind: ReplicaSet
name: ingress-nginx-controller-666487
uid: 7d6d94a3-9121-4f4e-b976-85016e346ba9
resourceVersion: "13576"
uid: 3642971f-6e37-4841-973d-8f7beeabd4dc
spec:
  containers:
    - args:
        - /nginx-ingress-controller
        - --publish-service=$(POD_NAMESPACE)/ingress-nginx-controller
        - --election-id=ingress-nginx-leader
        - --controller-class=k8s.io/ingress-nginx
        - --ingress-class=nginx
        - --configmap=$(POD_NAMESPACE)/ingress-nginx-controller
        - --validating-webhook=:8443
        - --validating-webhook-certificate=/usr/local/certificates/cert
        - --validating-webhook-key=/usr/local/certificates/key
        - --enable-metrics=false
    env:
      - name: POD_NAME
        valueFrom:
          fieldRef:
            apiVersion: v1
            fieldPath: metadata.name
      - name: POD_NAMESPACE
        valueFrom:
```

The status bar at the bottom indicates the file is 43,1 lines long and 11% complete. The system tray shows it's 21:45, 22°C, Mostly clear, and the date is 09-11-2024.

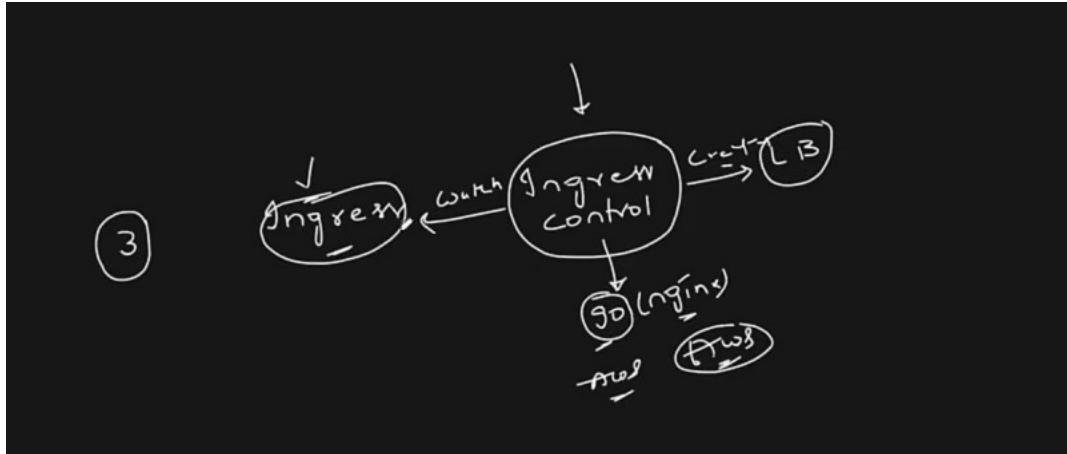
The screenshot shows the Visual Studio Code interface with the Explorer sidebar expanded, displaying the project structure:

- OPEN EDITORS**: Dockerfile, ingress.yaml
- GO-WEB-APP**: eks, ingress-controller\nginx (containing 01-installation.md), k8s\manifests (containing deploymentyaml, ingress.yaml, service.yaml), static, .gitignore, Dockerfile, eksctl\_.tar.gz, eksctl\_.zip, go.mod, go.mod, kubectl, LICENSE, main (containing main.go, main\_test.go), main.go

The **TERMINAL** tab is active, showing the contents of the **ingress.yaml** file:

```
1 # Ingress resource for the application
2 apiVersion: networking.k8s.io/v1
3 kind: Ingress
4 metadata:
5   name: go-web-app
6   annotations:
7     | nginx.ingress.kubernetes.io/rewrite-target: /
8 spec:
9   - ingressClassName: nginx
10  rules:
11    - host: go-web-app.local
12      http:
13        paths:
14          - path: /
15            pathType: Prefix
16            backend:
17              service:
18                name: go-web-app
19                port:
20                  number: 80
21
22
```

The status bar at the bottom indicates the file is 19 lines long and 25 selected, and the system tray shows it's 21:47, 22°C, Mostly clear, and the date is 09-11-2024.



# Load Balancer Running

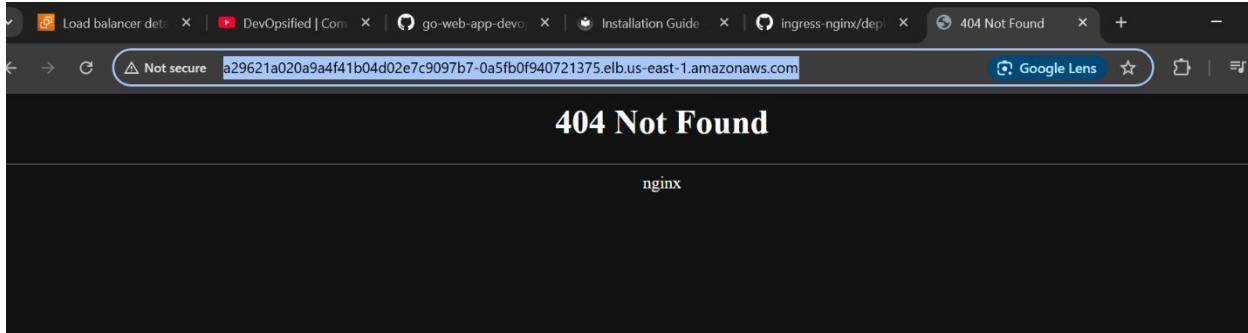
Load balancer type	Status	VPC	Load balancer IP address type
Network	Active	vpc-0c5480446ee4bcd22	IPv4
Scheme	Internet-facing	Hosted zone Z26RNL4JYFTOTI	Availability Zones subnet-07072a27de6afe3b9 us-east-1d (use1-az4) subnet-0adff4f0196caeaa89 us-east-1c (use1-az2)

```
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj---/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get ing
NAME      CLASS   HOSTS          ADDRESS
go-web-app  nginx   go-web-app.local  a29621a020a9a4f41b04d02e7c9097b7-0a5fb0f940721375.elb.us-east-1.amazonaws.com  80    72m
siddhartha@siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj---/10_Devopsified_Project/1_Go_Web_APP/go-web-app$
```

# Loadbalancer is matching

# now access the application on load balancer

# Error



@ # why ???

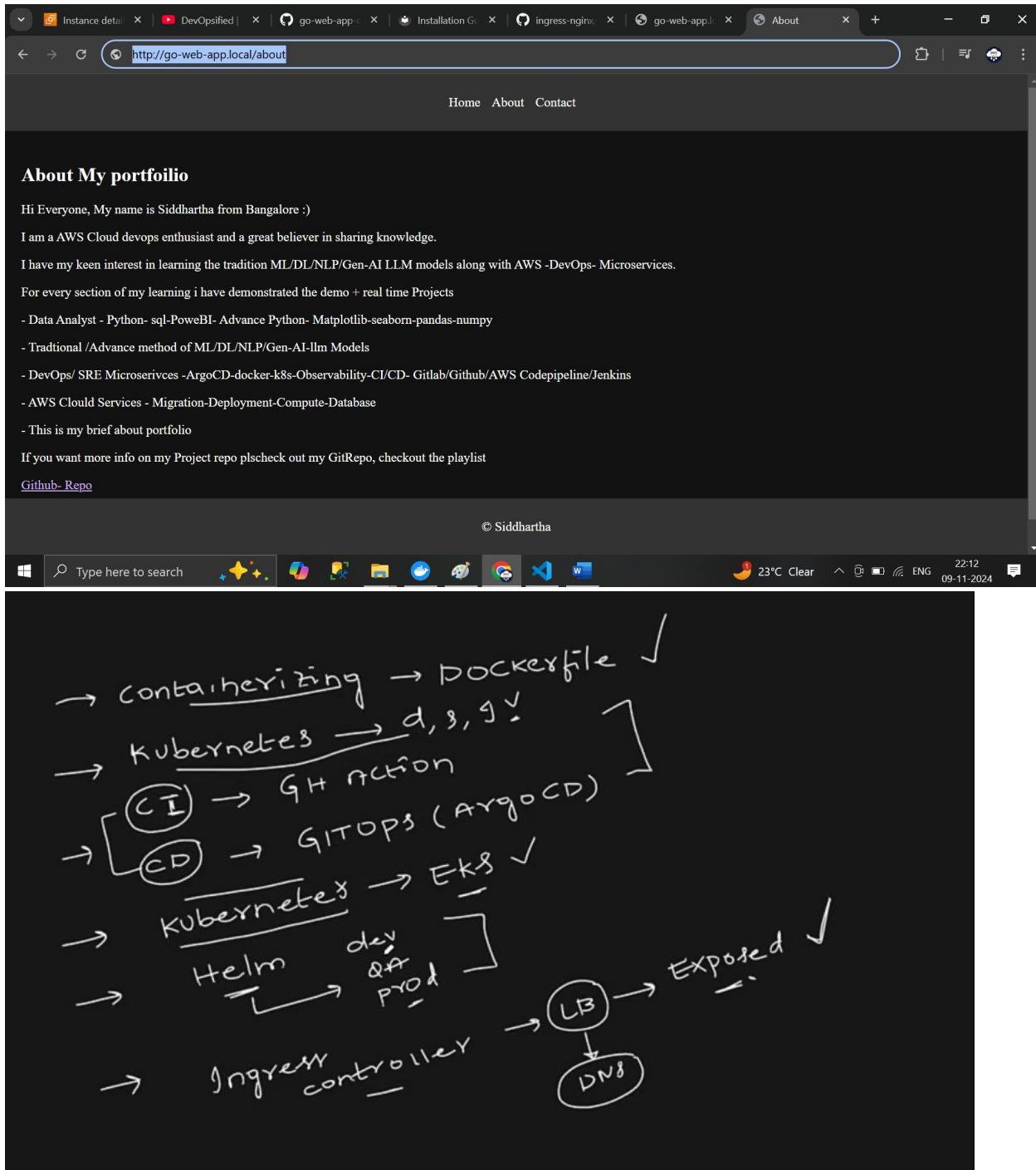
```
File Edit Selection View Go Run ... ← → go-web-app
Welcome Dockerfile U ingressyaml U
1 # Ingress resource for the application
2 apiVersion: networking.k8s.io/v1
3 kind: Ingress
4 metadata:
5   name: go-web-app
6   annotations:
7     | nginx.ingress.kubernetes.io/rewrite-target: /
8 spec:
9   ingressClassName: nginx
10  rules:
11    - host: go-web-app.local
12      http:
13        paths:
14          - path: /
15            pathType: Prefix
16            backend:
17              service:
18                name: go-web-app
19                port:
20                  number: 80
21
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ESP-IDF

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APPsiddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APPsiddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APPsiddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ nslookup a29621a020a9a4f41b04d02e7c9097b7-0a5fb0f940721375.elb.us-east-1.amazonaws.com
;; Got recursion not available from 172.24.16.1
Server: 172.24.16.1
Address: 172.24.16.1#53

Non-authoritative answer:
Name: a29621a020a9a4f41b04d02e7c9097b7-0a5fb0f940721375.elb.us-east-1.amazonaws.com
Address: 52.1.147.102
Name: a29621a020a9a4f41b04d02e7c9097b7-0a5fb0f940721375.elb.us-east-1.amazonaws.com
Address: 3.213.35.88

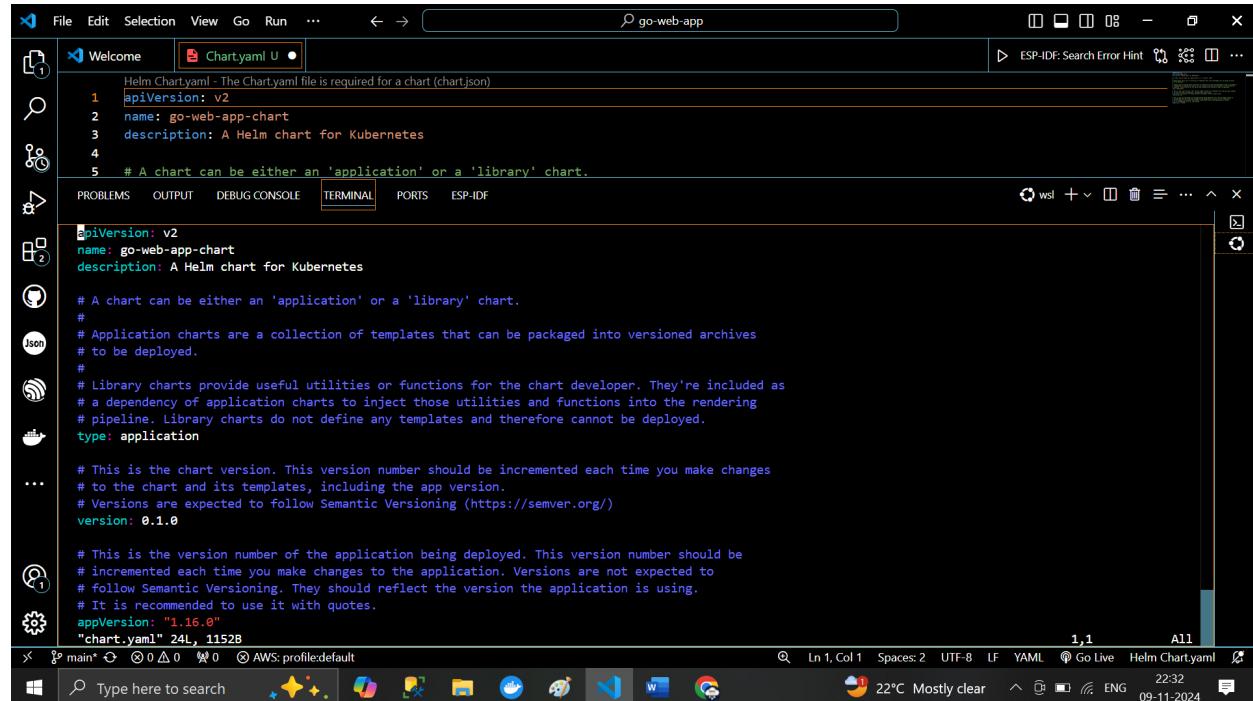
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$
```



# Helm chart + version

```
siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects____/0006_DevOps_notes_Prj----/10_DevOpsified_Project/1_Go_Web_APP/go-web-app/helm$ helm version  
version.BuildInfo{Version:"v3.16.2", GitCommit:"13654a52f7c70a143b1dd51416d633e1071fafbb", GitTreeState:"clean", GoVersion:"go1.22.7"}  
siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects____/0006_DevOps_notes_Prj----/10_DevOpsified_Project/1_Go_Web_APP/go-web-app/helm$ helm create go-web-app-c  
hart  
Creating go-web-app-chart  
siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects____/0006_DevOps_notes_Prj----/10_DevOpsified_Project/1_Go_Web_APP/go-web-app/helm$ █
```

## # About chart



The screenshot shows the VS Code interface with the title bar "go-web-app". The left sidebar has icons for file operations like Open, Save, Find, and Refresh. The main editor area displays the "Chart.yaml" file content:

```
apiVersion: v2
name: go-web-app-chart
description: A Helm chart for Kubernetes

# A chart can be either an 'application' or a 'library' chart.

# Application charts are a collection of templates that can be packaged into versioned archives
# to be deployed.

# Library charts provide useful utilities or functions for the chart developer. They're included as
# a dependency of application charts to inject those utilities and functions into the rendering
# pipeline. Library charts do not define any templates and therefore cannot be deployed.

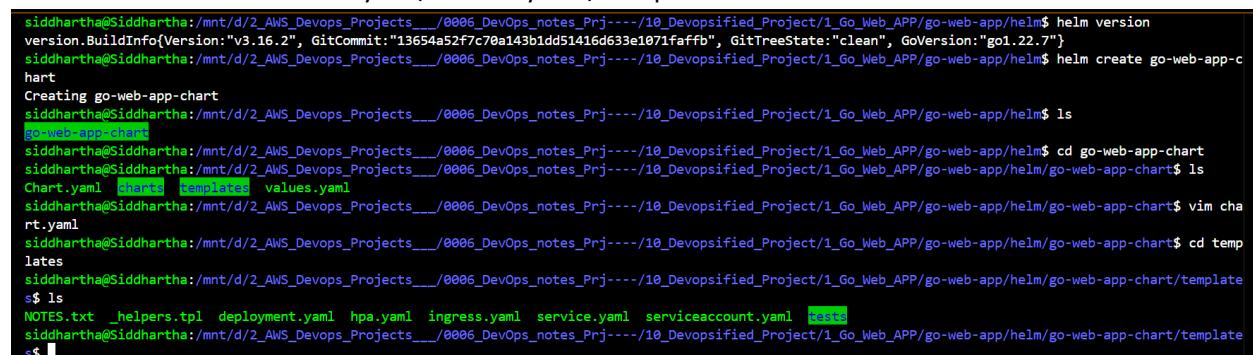
type: application

# This is the chart version. This version number should be incremented each time you make changes
# to the chart and its templates, including the app version.
# Versions are expected to follow Semantic Versioning (https://semver.org/)
version: 0.1.0

# This is the version number of the application being deployed. This version number should be
# incremented each time you make changes to the application. Versions are not expected to
# follow Semantic Versioning. They should reflect the version the application is using.
# It is recommended to use it with quotes.
appVersion: "0.16.0"
"chart.yaml" 24L, 1152B
```

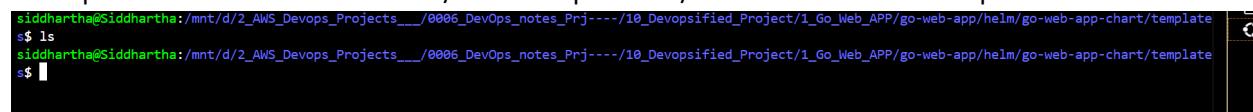
The status bar at the bottom shows "main" and "AWS: profile:default". The bottom right corner shows the date and time: "09-11-2024".

## # inside helm folder -- Chart.yaml/ values.yaml / templates- Dir



```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm$ helm version
version.BuildInfo{Version:"v3.16.2", GitCommit:"13654a52f7c70a143b1dd51416d633e1071fafffb", GitTreeState:"clean", GoVersion:"go1.22.7"}
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm$ helm create go-web-app-chart
Creating go-web-app-chart
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm$ ls
go-web-app-chart
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm$ cd go-web-app-chart
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart$ ls
Chart.yaml  charts  templates  values.yaml
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart$ vim charts.yaml
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart$ cd temp
lates
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
s$ ls
NOTES.txt  _helpers.tpl  deployment.yaml  ingress.yaml  service.yaml  serviceaccount.yaml  tests
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
s$
```

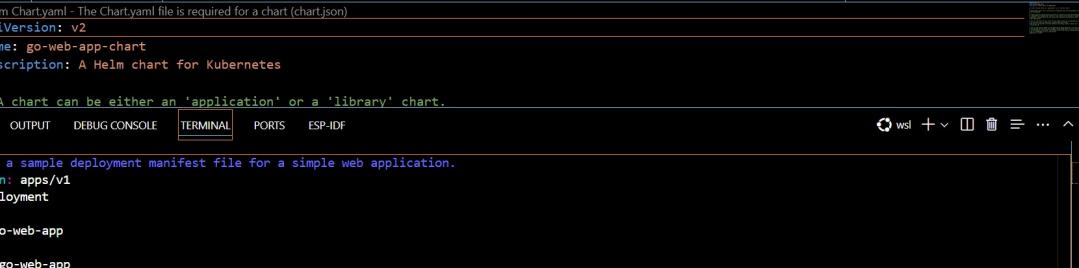
## # Template Dir – remove all the files / folder .. cp the k8s/ manifests – data to templates – helm chat



```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
s$ ls
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
s$
```

```
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
s$ cp ../../k8s/manifests/* .
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
s$ ls
deployment.yaml  ingress.yaml  service.yaml
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
s$
```

# Vim deployment.yaml



The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** On the left, there are icons for files, folders, search, and other development tools.
- Search Bar:** At the top center, it says "go-web-app".
- Terminal:** The "TERMINAL" tab is active at the bottom of the interface.
- Code Editor:** The main area displays two files:
  - Chart.yaml:** A Helm chart manifest with the following content:

```
apiVersion: v2
name: go-web-app-chart
description: A Helm chart for Kubernetes
# A chart can be either an 'application' or a 'library' chart.
```
  - deployment.yaml:** A Kubernetes deployment manifest with the following content:

```
# This is a sample deployment manifest file for a simple web application.
apiVersion: apps/v1
kind: Deployment
metadata:
  name: go-web-app
  labels:
    app: go-web-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: go-web-app
  template:
    metadata:
      labels:
        app: go-web-app
    spec:
      containers:
        - name: go-web-app
          image: siddhartha082/go-web-app:{{ .Values.image.tag}}
          ports:
            - containerPort: 8080
```
- Status Bar:** At the bottom right, it shows "20,63" (likely lines of code), "All", and "Helm Chart.yaml".
- Taskbar:** At the very bottom, it includes the Windows Start button, a search bar ("Type here to search"), and several pinned icons for various applications like File Explorer, GitHub, and Microsoft Edge.

---

## # Values.yaml



```
 1 # Default values for go-web-app-chart.
 2 # This is a YAML-formatted file.
 3 # Declare variables to be passed into your templates.
 4
 5 replicaCount: 1
 6
 7 image:
 8   repository: siddhartha082/go-web-app
 9   pullPolicy: IfNotPresent
10   # Overrides the image tag whose default is the chart appVersion.
11   tag: "10016307834"
12
13 ingress:
14   enabled: false
15   className: ""
16   annotations: {}
17   # kubernetes.io/ingress.class: nginx
18   # kubernetes.io/tls-acme: "true"
19   hosts:
20     - host: chart-example.local
21       paths:
22         - path: /
23           pathType: ImplementationSpecific
24
25
```

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    ESP-IDF
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
$ kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/go-web-app-66c6b64548-t87nt   1/1     Running   0          136m

NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/go-web-app  NodePort  10.100.192.220  <none>        80:38706/TCP  135m
service/kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP     166m

NAME          READY   UP-TO-DATE  AVAILABLE   AGE
deployment.apps/go-web-app  0/1       1            0          136m

NAME          DESIRED  CURRENT  READY   AGE
replicaset.apps/go-web-app-66c6b64548  1         1         0      136m
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
$ 

```

# deleted all

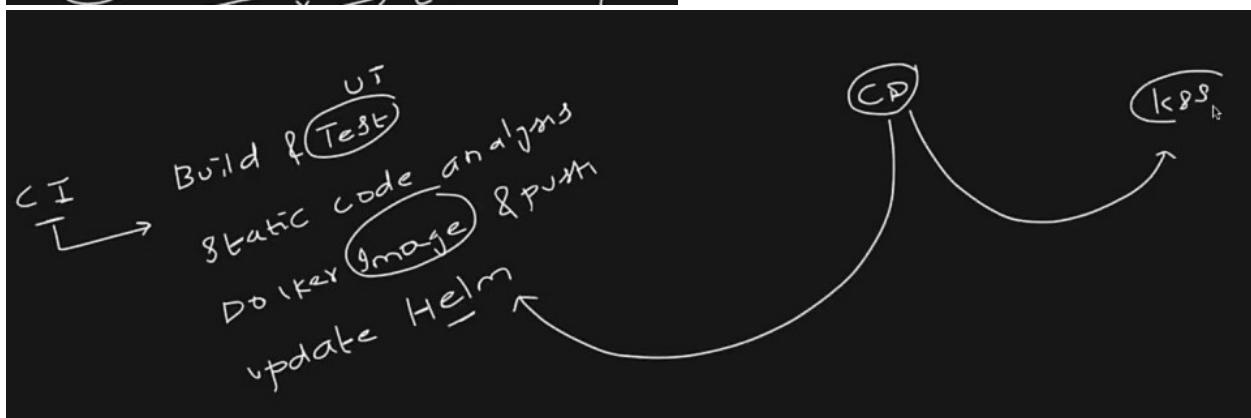
```

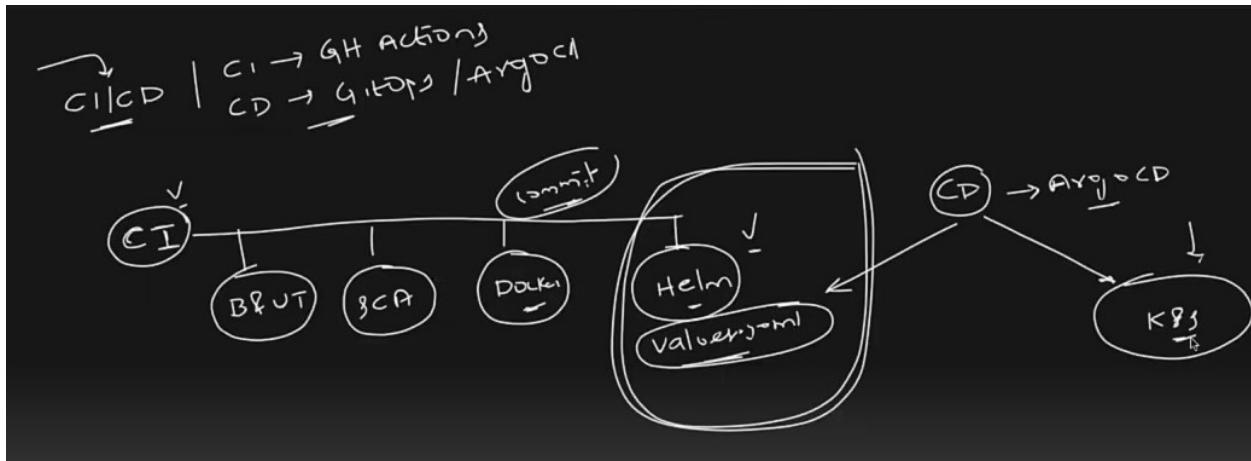
NAME          DESIRED  CURRENT  READY   AGE
replicaset.apps/go-web-app-66c6b64548  1         1         0      136m
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
$ kubectl delete deploy go-web-app
deployment.apps "go-web-app" deleted
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
$ kubectl delete svc go-web-app
service "go-web-app" deleted
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
$ kubectl delete ingc go-web-app
error: the server doesn't have a resource type "ingc"
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
$ kubectl delete ing go-web-app
ingress.networking.k8s.io "go-web-app" deleted
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template
$ kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/go-web-app-66c6b64548-t87nt   1/1     Terminating   0          138m

NAME           TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
service/kubernetes  ClusterIP  10.100.0.1    <none>        443/TCP     168m
siddhartha@Siddhartha:/mnt/d/2_AWS_Devops_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app/helm/go-web-app-chart/template

```

# Last part





github.com/Siddhartha082/go-web-app/settings

**General**

Repository name: go-web-app

Code and automation

- Branches
- Tags
- Rules
- Actions
- Webhooks
- Environments
- Codespaces
- Pages

Secrets and variables

- Actions
- Codespaces
- Dependabot

Default branch: main

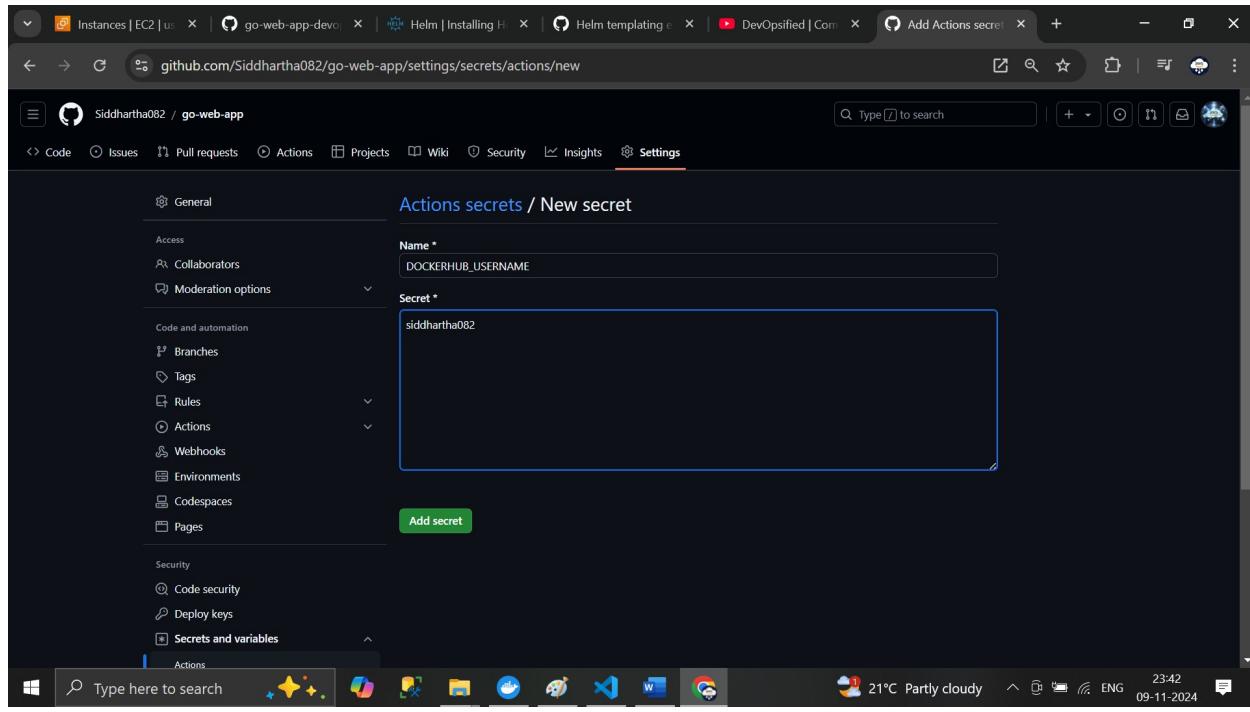
Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640x320px (1280x640px for best display).

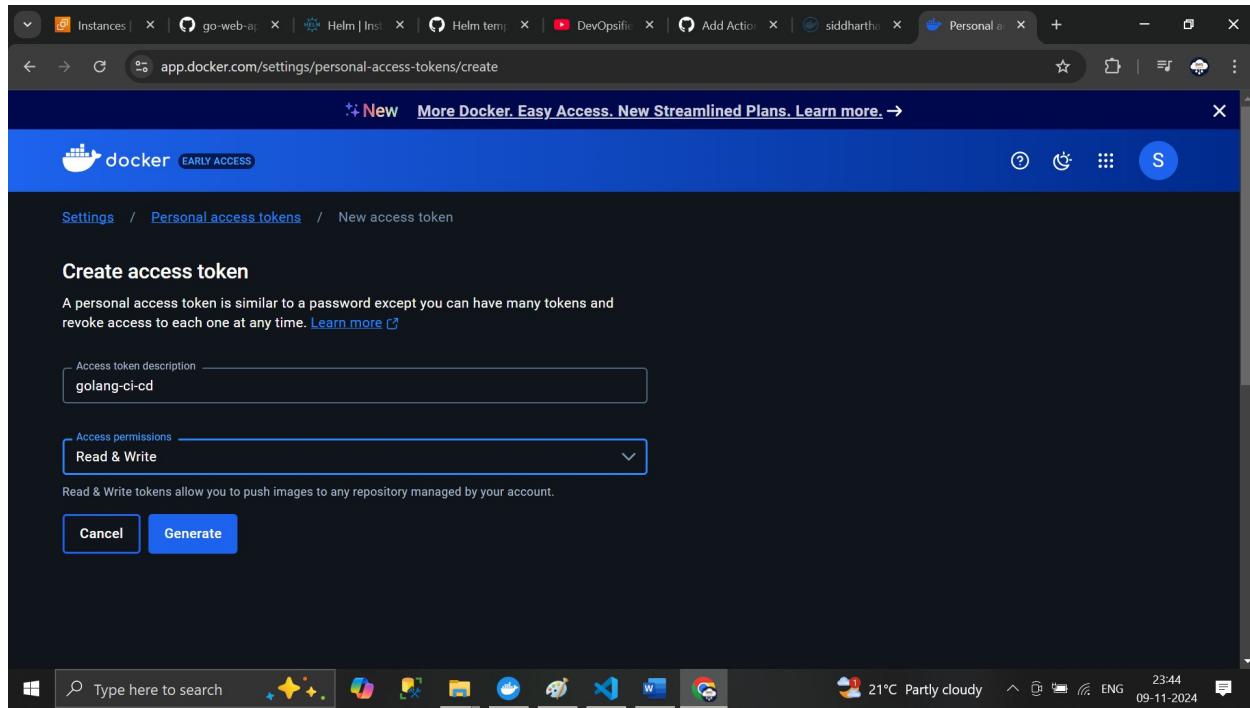
Download template

https://github.com/Siddhartha082/go-web-app/settings/secrets/actions



The screenshot shows the GitHub Actions secrets configuration page for a repository named 'go-web-app'. The 'General' section is selected. A new secret is being created with the name 'DOCKERHUB\_USERNAME' and the value 'siddhartha082'. The 'Add secret' button is visible at the bottom right of the secret input field.

# docker hub



The screenshot shows the Docker Personal Access Tokens creation page. A new access token is being created with the description 'golang-ci-ci'. The access permissions are set to 'Read & Write'. The 'Generate' button is visible at the bottom left.

The image shows two screenshots of the GitHub web interface. The top screenshot displays the 'Repository secrets' page for a repository named 'go-web-app'. It lists two secrets: 'DOCKERHUB\_TOKEN' (created 'now') and 'DOCKERHUB\_USERNAME' (created '3 minutes ago'). A 'Manage environment secrets' button is visible. The bottom screenshot shows the 'New personal access token (classic)' page. The 'Tokens (classic)' tab is selected. A note indicates that personal access tokens function like ordinary OAuth access tokens. A 'Note' section contains the identifier 'golang-CICD'. The 'Expiration' dropdown is set to '30 days', with a note that it will expire on Mon, Dec 9 2024. The 'Select scopes' section lists several options under the 'repo' scope, each with a description: 'repo' (Full control of private repositories), 'repo:status' (Access commit status), 'repo\_deployment' (Access deployment status), 'public\_repo' (Access public repositories), 'repo:invite' (Access repository invitations), and 'security\_events' (Read and write security events). Both screenshots are taken from a Windows 10 desktop environment.

The screenshot shows the GitHub Actions secrets management interface. On the left, there's a sidebar with options like Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, and Pages. The main area has tabs for 'Secrets' (which is selected) and 'Variables'. Under 'Environment secrets', it says 'This environment has no secrets.' and has a 'Manage environment secrets' button. Under 'Repository secrets', there's a table with three rows:

Name	Last updated
DOCKERHUB_TOKEN	9 minutes ago
DOCKERHUB_USERNAME	12 minutes ago
TOKEN	now

There's also a green 'New repository secret' button. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating 'Rain on Tuesday' at 23:54 on 09-11-2024.

## # last part – Argo CD

The screenshot shows a GitHub blob page for 'gitops/argocd/01-install.md'. The page contains sections for 'Install Argo CD using manifests', 'Access the Argo CD UI (Loadbalancer service)', 'Access the Argo CD UI (Loadbalancer service) -For Windows', and 'Get the Loadbalancer service IP'. Each section includes a code block with a copy icon:

- Install Argo CD using manifests**

```
kubectl create namespace argocd
kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
```
- Access the Argo CD UI (Loadbalancer service)**

```
kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
```
- Access the Argo CD UI (Loadbalancer service) -For Windows**

```
kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
```
- Get the Loadbalancer service IP**

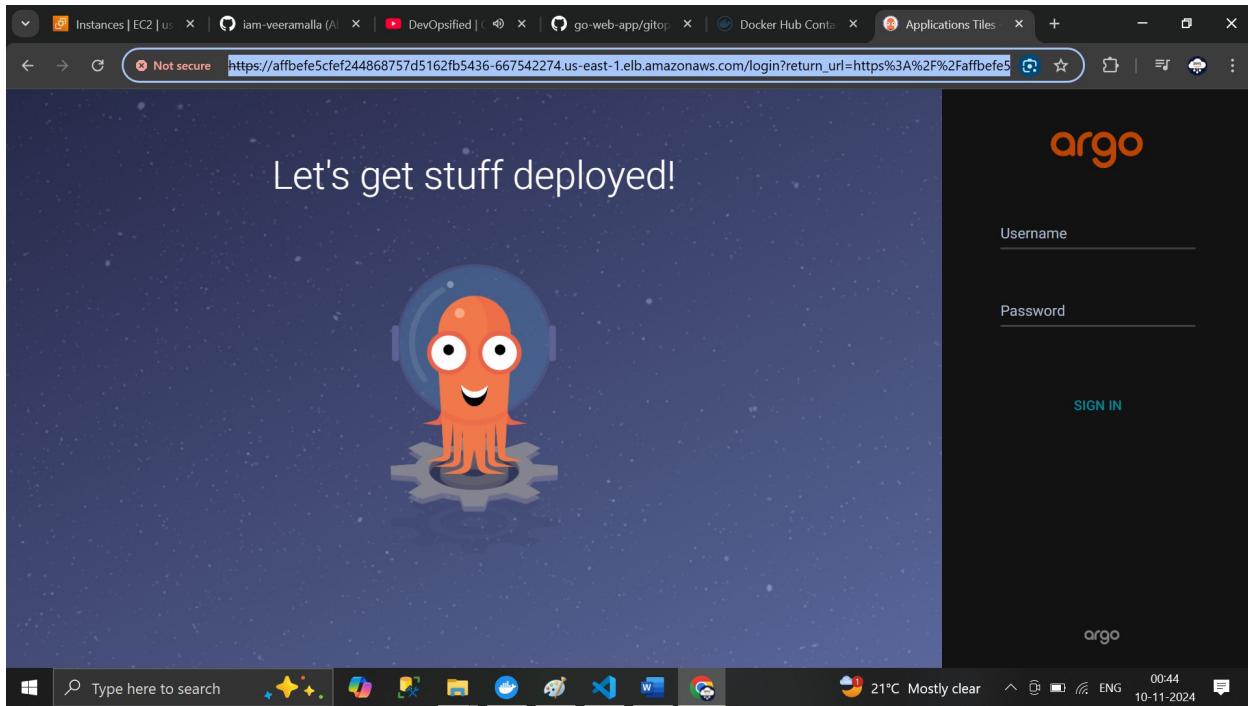
```
kubectl get svc argocd-server -n argocd
```

The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating '21°C Mostly clear' at 00:41 on 10-11-2024.

The screenshot shows two nearly identical terminal windows side-by-side, both displaying the same command-line session. The session starts with a GitHub Workflow YAML file named `cicd.yaml`, followed by the command `kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'`. This command patches the Kubernetes Service `argocd-server` in the namespace `argocd` to change its type from `ClusterIP` to `LoadBalancer`. After the patch is applied, the user runs `kubectl get svc -n argocd` to list all services in the `argocd` namespace. The output of this command is a table showing the following services:

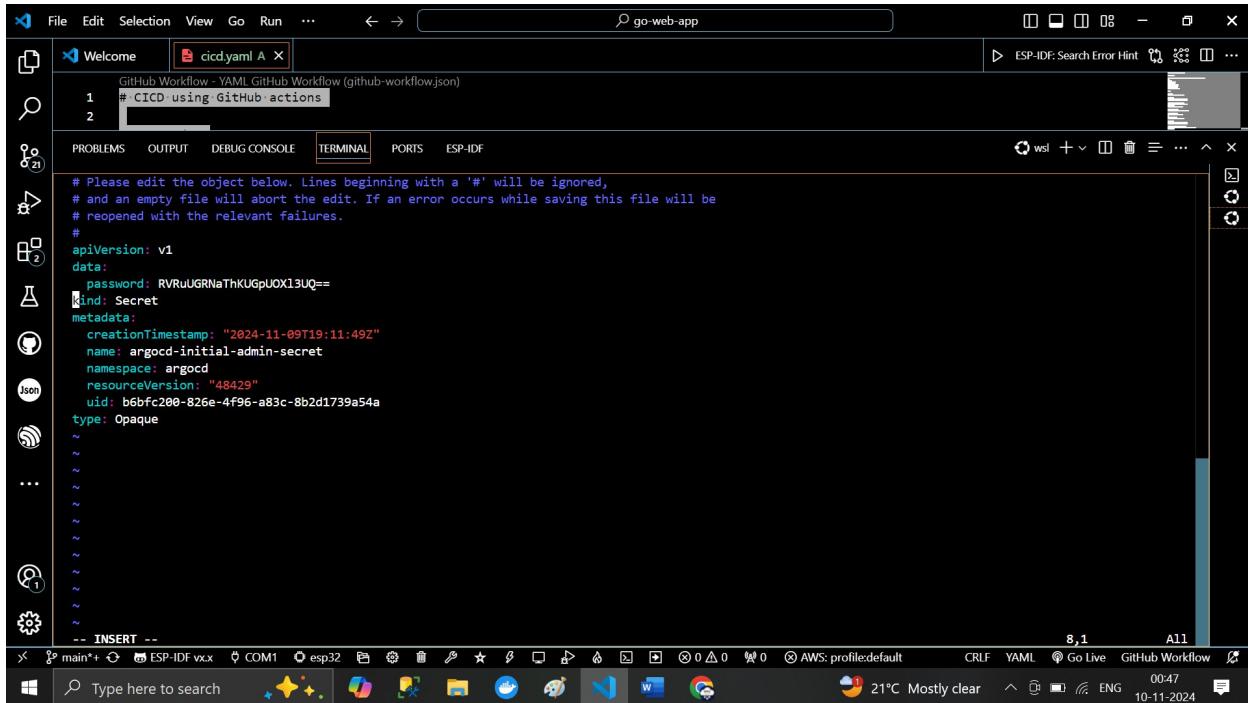
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
argocd-applicationset-controller	ClusterIP	10.100.142.45	<none>	7000/TCP,8
080/TCP	96s			
argocd-dex-server	ClusterIP	10.100.203.79	<none>	5556/TCP,5
557/TCP,5558/TCP	96s			
argocd-metrics	ClusterIP	10.100.172.5	<none>	8082/TCP
	95s			
argocd-notifications-controller-metrics	ClusterIP	10.100.169.91	<none>	9001/TCP
	94s			
argocd-redis	ClusterIP	10.100.251.31	<none>	6379/TCP
	94s			
argocd-repo-server	ClusterIP	10.100.187.28	<none>	8081/TCP,8
084/TCP	93s			
argocd-server	LoadBalancer	10.100.229.21	afffbefc5cfef244868757d5162fb5436-667542274.us-east-1.elb.amazonaws.com	80:31614/T
CP,443:30830/TCP	92s			
argocd-server-metrics	ClusterIP	10.100.154.203	<none>	8083/TCP
	92s			

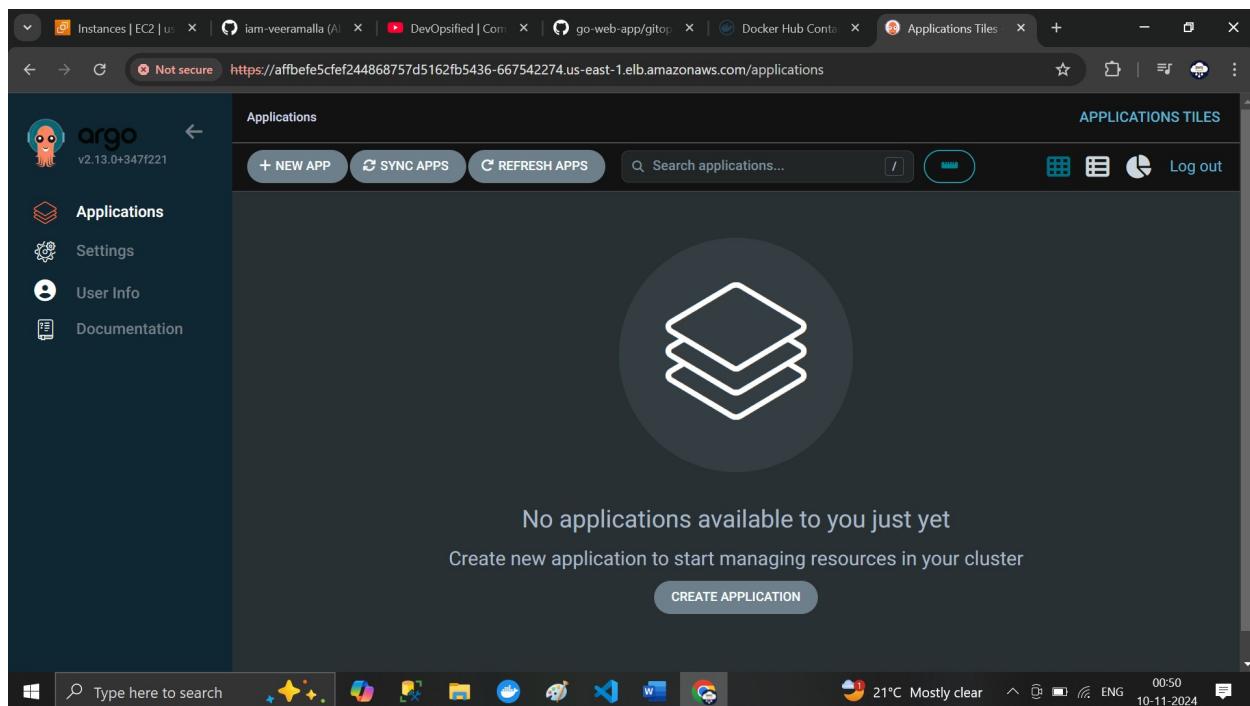
The terminal session concludes with the command `siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects____/0006_DevOps_notes_Proj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$`.



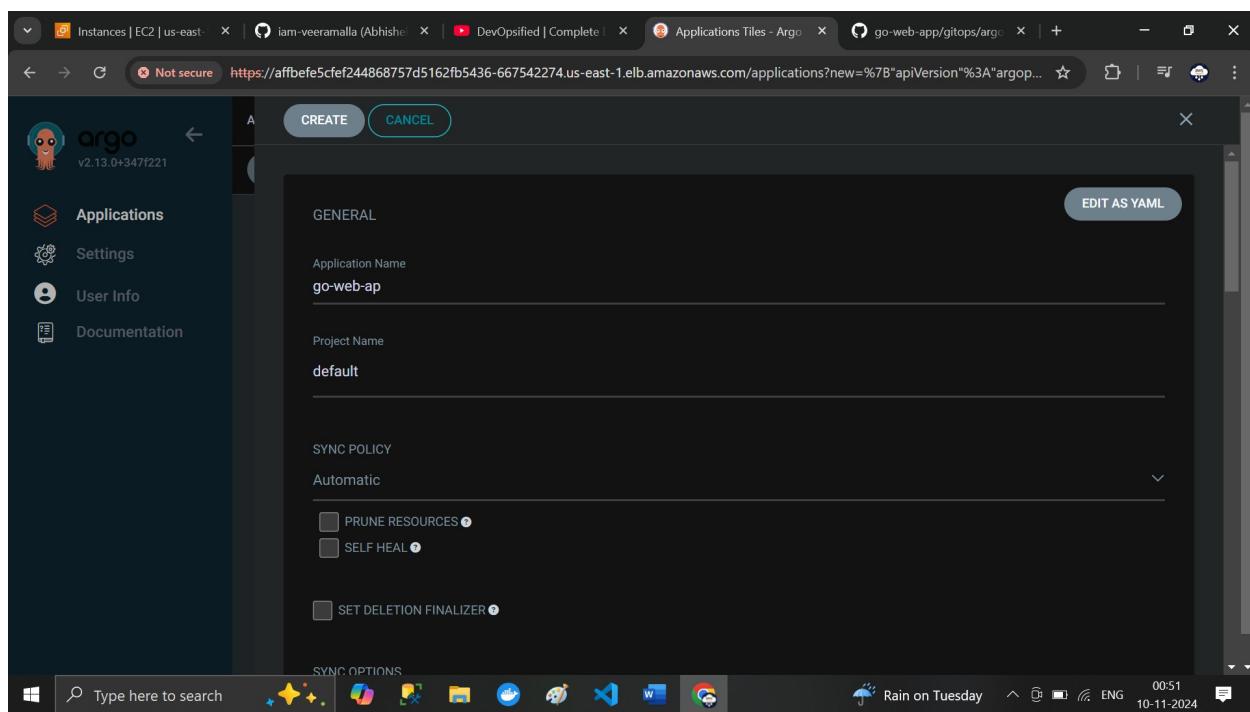
# password

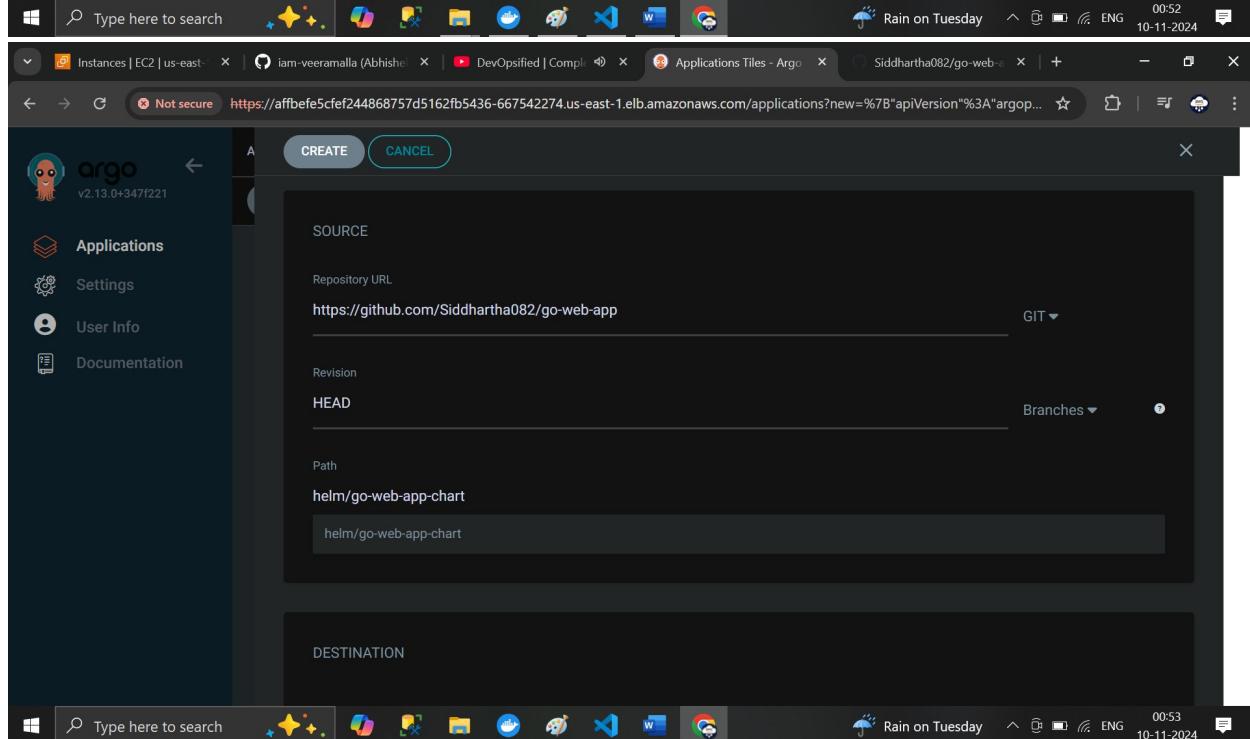
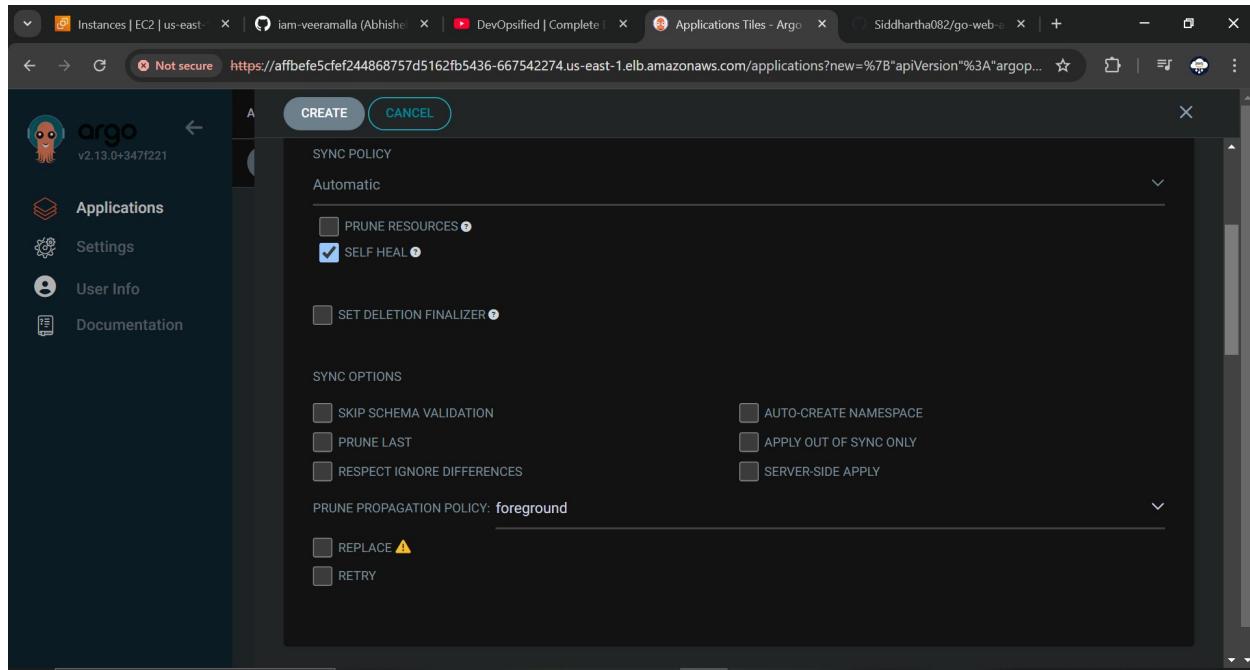
```
siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl get secrets -n argocd
NAME          TYPE    DATA  AGE
argocd-initial-admin-secret  Opaque  1    4m
argocd-notifications-secret Opaque  0    4m28s
argocd-redis      Opaque  1    4m12s
argocd-secret     Opaque  5    4m27s
siddhartha@Siddhartha:/mnt/d/2_AWS_DevOps_Projects___/0006_DevOps_notes_Prj----/10_Devopsified_Project/1_Go_Web_APP/go-web-app$ kubectl edit secret argocd-initial-admin-secret -n argocd
```





# create





The screenshot shows the Argo UI interface for creating a new application. The left sidebar has links for Applications, Settings, User Info, and Documentation. The main area is titled 'CREATE' with tabs for 'APPLICATIONS' and 'HELM'. Under 'HELM', there is a 'VALUES FILES' field containing 'values.yaml'. The 'VALUES' section is currently empty. The 'PARAMETERS' section contains the following key-value pairs:

Parameter	Value
image.pullPolicy	IfNotPresent
image.repository	siddhartha082/go-web-app
image.tag	11758764658

This screenshot is identical to the one above, showing the Argo UI interface for creating a new application. The left sidebar and 'CREATE' tab are the same. The 'HELM' section shows 'values.yaml' selected in the 'VALUES FILES' field. The 'VALUES' section is empty. The 'PARAMETERS' section contains the following key-value pairs:

Parameter	Value
image.pullPolicy	IfNotPresent
image.repository	siddhartha082/go-web-app
image.tag	11758764658

The screenshot shows the Argo UI interface for creating a new Helm application. The left sidebar includes links for Instances, Applications, Settings, User Info, and Documentation. The main area has tabs for Applications and Helm. Under Helm, there's a 'VALUES FILES' section with a dropdown set to 'values.yaml'. Below it are sections for 'VALUES' and 'PARAMETERS'. The 'PARAMETERS' section contains the following configuration:

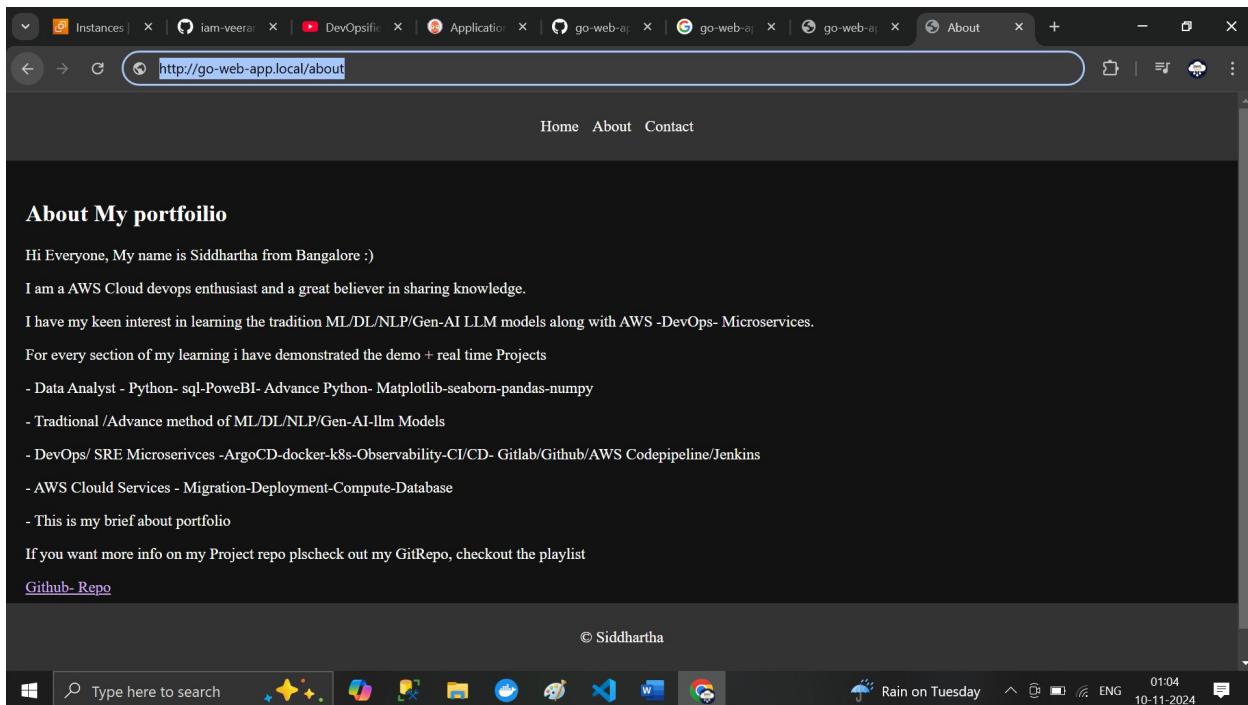
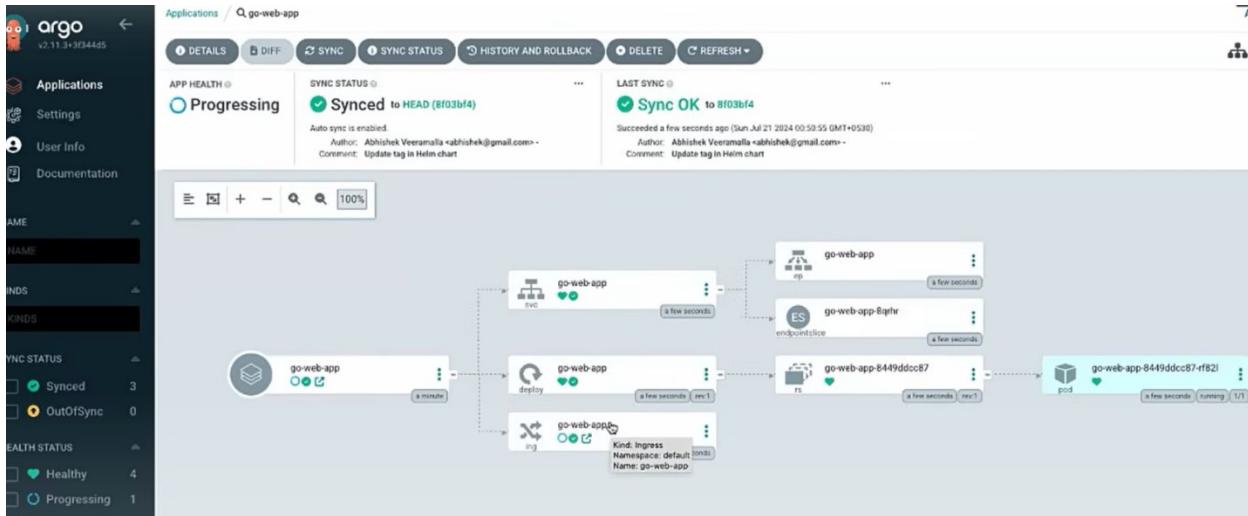
```
image.pullPolicy: IfNotPresent
image.repository: siddhartha082/go-web-app
image.tag: 11758764658
ingress.className:
ingress.enabled: false
ingress.hosts[0].host: chart-example.local
ingress.hosts[0].paths[0].path: /
```

The screenshot shows the GitHub code editor displaying the contents of the 'values.yaml' file from the 'go-web-app/helm/go-web-app-chart' repository. The file content is as follows:

```
replicaCount: 1

image:
  repository: siddhartha082/go-web-app
  pullPolicy: IfNotPresent
  # Overrides the image tag whose default is the chart appVersion.
  tag: "11758764658"

ingress:
  enabled: false
  className: ""
  annotations: {}
  # kubernetes.io/ingress.class: nginx
  # kubernetes.io/tls-acme: "true"
  hosts:
    - host: chart-example.local
      paths:
        - path: /
          pathType: ImplementationSpecific
```



**About My portfolio**

Hi Everyone, My name is Siddhartha from Bangalore :)

I am a AWS Cloud devops enthusiast and a great believer in sharing knowledge.

I have my keen interest in learning the tradition ML/DL/NLP/Gen-AI LLM models along with AWS -DevOps- Microservices.

For every section of my learning i have demonstrated the demo + real time Projects

- Data Analyst - Python- sql-PowerBI- Advance Python- Matplotlib-seaborn-pandas-numpy
- Traditional /Advance method of ML/DL/NLP/Gen-AI-LLM Models
- DevOps/ SRE Microservices -ArgoCD-docker-k8s-Observability-CI/CD- Gitlab/Github/AWS Codepipeline/Jenkins
- AWS Cloud Services - Migration-Deployment-Compute-Database
- This is my brief about portfolio

If you want more info on my Project repo plscheck out my GitRepo, checkout the playlist

[Github- Repo](#)

© Siddhartha

# Docker new Tag

[hub.docker.com/repository/docker/siddhartha082/go-web-app/general](#)

Docker commands

To push a new tag to this repository:

```
docker push siddhartha082/go-web-app:tagname
```

**Tags**

This repository contains 4 tag(s).

Tag	OS	Type	Pulled	Pushed
11758764658		Image	--	27 minutes ago
11758752925		Image	--	28 minutes ago
11758654732		Image	--	44 minutes ago
v1		Image	7 minutes ago	a day ago

**Automated Builds**

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#).

[Upgrade](#)

[Code](#)[Blame](#)

24 lines (20 loc) · 583 Bytes

Code 55% faster with GitHub Copilot

```
1  # Default values for go-web-app-chart.
2  # This is a YAML-formatted file.
3  # Declare variables to be passed into your templates.
4
5  replicaCount: 1
6
7  image:
8    repository: siddhartha082/go-web-app
9    pullPolicy: IfNotPresent
10   # Overrides the image tag whose default is the chart appVersion.
11   tag: "11758764658"
12
13  ingress:
14    enabled: false
15    className: ""
16    annotations: {}
17    # kubernetes.io/ingress.class: nginx
18    # kubernetes.io/tls-acme: "true"
19    hosts:
20      - host: chart-example.local
```