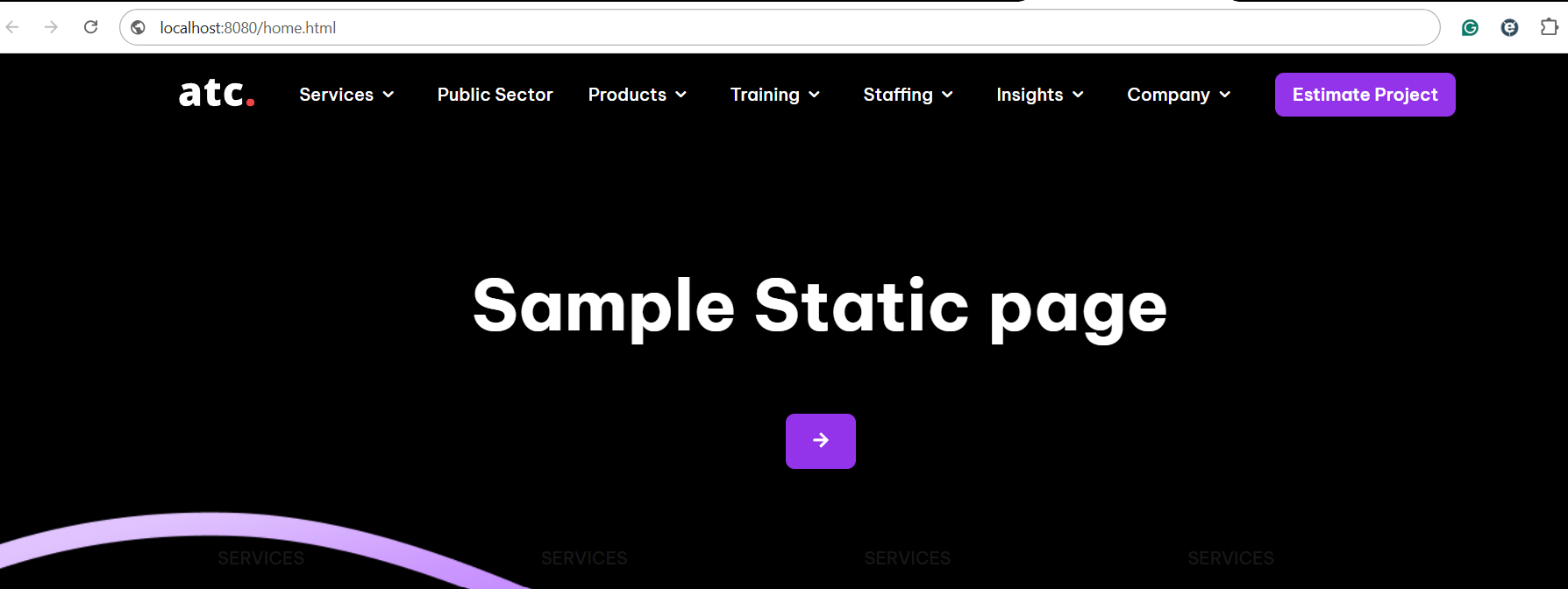
**Project: Simple Static Web Application using html and Javascript**

The server will start on port 8080. This can access it by navigating to

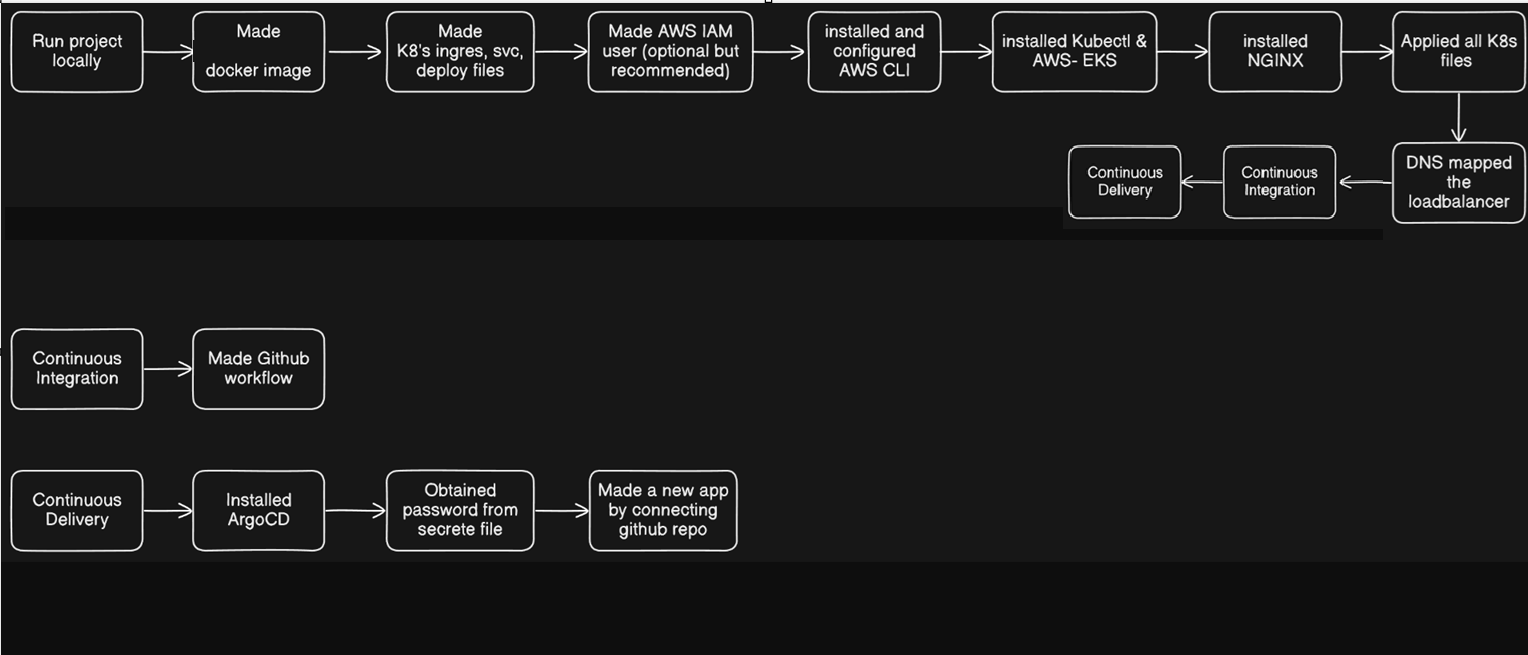
*http://localhost:8080/home.html*

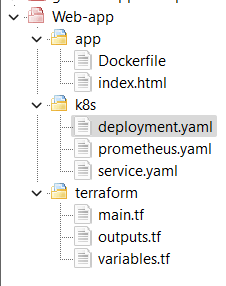


**DevOps practices include the following:**

* Creating Dockerfile
* Containerization
* Continuous Integration (CI)
* Continuous Deployment (CD)

**The Project Flow and folder Structure:**





**Docker File:**

The Dockerfile will be used to containerize a simple static web app. I used nginx for this static HTML page. If it’s a EP level Application We can go with *Multi-stage Docker Image (Distroless*)

*Deploying the docker Image*  
Build the Docker image and push it to a container registry (Docker Hub).

*# Use an official NGINX image as the base*

*FROM nginx:alpine*

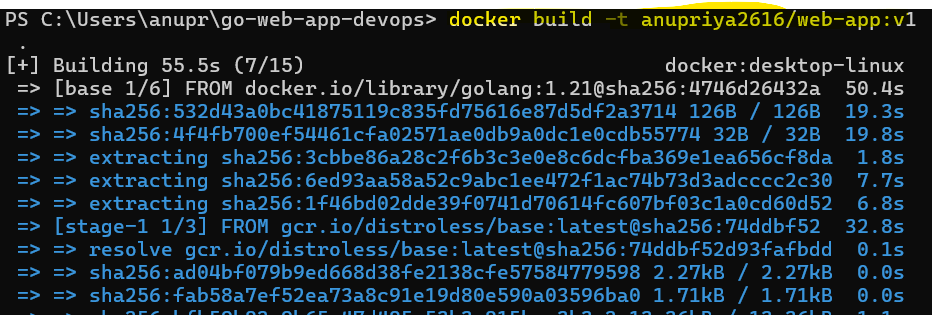
*COPY ./index.html /usr/share/nginx/html/*

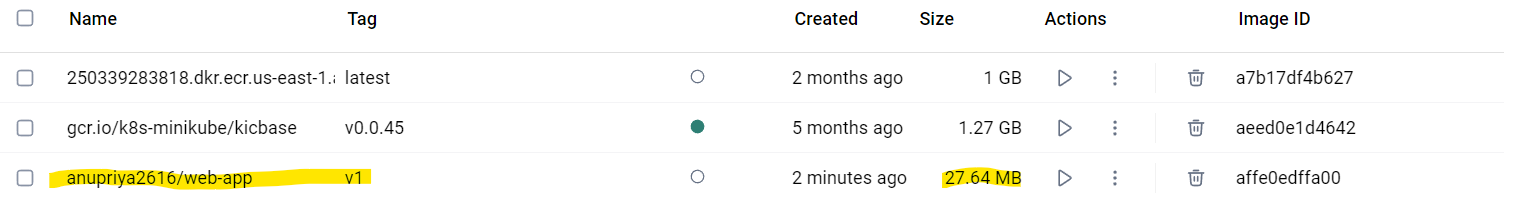
*EXPOSE 80*

*CMD ["nginx", "-g", "daemon off;"]*

docker build -t anupriya2616/web-app:v1 .

docker push anupriya2616/web-app:v1

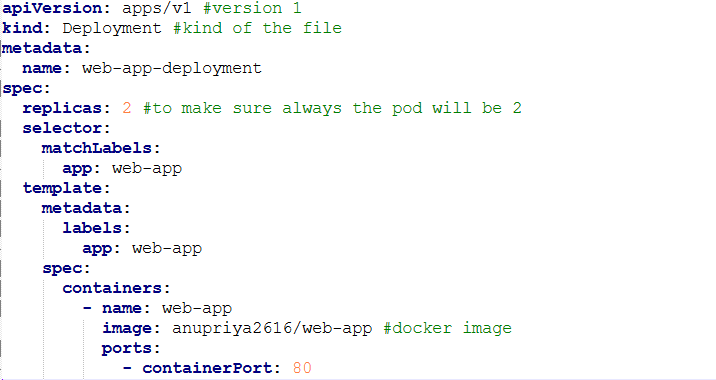




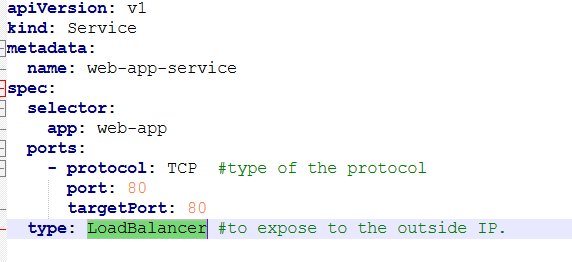
**Kubernetes deployment**

Apply the Kubernetes deployment and service:

The Kubernetes deployment and service files will allow you to deploy the web application and expose it.

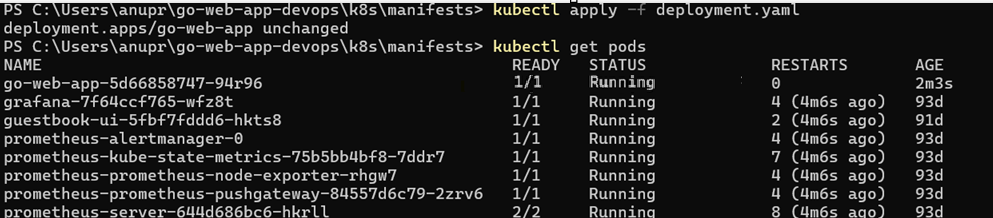
Deployment.yaml  


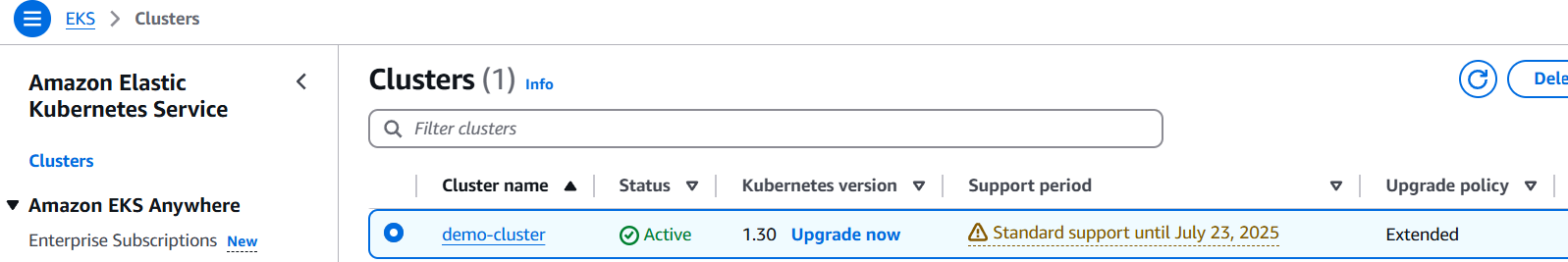
Service.yaml



kubectl apply -f ./k8s/deployment.yaml

kubectl apply -f ./k8s/service.yaml





**Prometheus monitoring**

Add the Prometheus monitoring configurations

You can add a basic Prometheus configuration to monitor the deployment.

**Terraform**

Run Terraform to create the infrastructure and deploy the application:

Used Terraform to create a Kubernetes cluster (local Minikube), configure Kubernetes, and deploy the resources.

**main.tf**

* **Purpose**: In this main file I have defined the primary infrastructure resources (EC2 instances,EKS clusters).

**variables.tf**

* **Purpose**: In this file I defined input variables that can be used in the configuration files. (AWS\_region setup)

**outputs.tf**

* **Purpose**: Used this file to define output values that are displayed after the Terraform apply operation is completed. (web\_app\_service\_url)

terraform init #initializes a working directory containing Terraform configuration files. (aws config)

terraform apply #ensures that the resources specified in the configuration are created/modified.