**Task - 1**  
1)Go Application

● A simple Go application that serves "Hello from Botgauge" at the /hello path.

● Dockerfile to containerize the application  
  
  
  
apt-get update  
  
mkdir hello-botgauge

cd hello-botgauge  
  
Create a new file named main.go and add the following content

package main

import (

"fmt"

"net/http"

)

func helloHandler(w http.ResponseWriter, r \*http.Request) {

fmt.Fprintln(w, "Hello from Botgauge")

}

func main() {

http.HandleFunc("/hello", helloHandler)

http.ListenAndServe(":8080", nil)

}

Create a Dockerfile  
  
FROM golang:1.18-alpine

WORKDIR /app

COPY go.mod ./

RUN go mod download

COPY . .

RUN go build -o /hello-botgauge

EXPOSE 8080

CMD ["/hello-botgauge"]

create a file go.mod  
  
module hello-botgauge

go 1.18

require (

// your dependencies here

)

Then install golang  
  
  
apt install golang-go

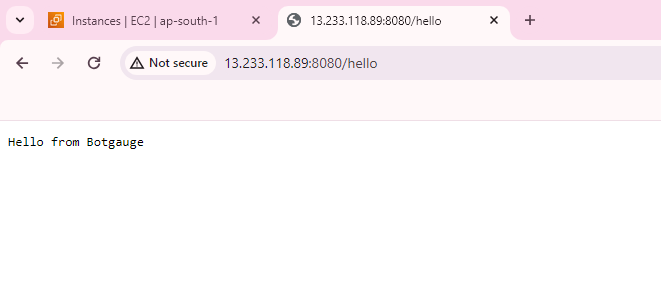
go mod init hello-botgauge

Install docker  
  
apt install docker.io  
  
  
Build the Docker image

docker build -t hello-botgauge .

docker run -d --name task -p 8080:8080 hello-botgauge

Hit this   
  
http://13.233.118.89:8080/hello



**Task 2**

Pipeline to build the Go application

1.First create a github repo  
  
2. Create a docker hub account

3. Create a workflow file in the .gihub/workflows directory of your repo

name: Build and Deploy

on:

push:

branches:

- main

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Checkout Repository

uses: actions/checkout@v2

- name: Set up Go

uses: actions/setup-go@v2

with:

go-version: 1.18

- name: Build Go Application

run: go build -o hello-botgauge

- name: Docker Build and Push

uses: docker/build-push-action@v2

with:

push: true

tags: username/hello-botgauge:latest

**Task 3**