Objective:

The Main Objective of this assignment is to understanding Docker and containerization by Dockerizing a simple HTML page using Nginx as the web server.

Requirements:

Install docker in ubuntu machine

sudo apt install docker.io -y

Please find below a Sample HTMP Page which we will be using for deploying.

Basic HTML Page:

Create a plain HTML page named index.html with some content (e.g., "Hello, Docker!").

#------------------index.html---------------------------------

<!DOCTYPE html>

<html>

<head>

<title>Welcome to Tilak's World of Devops!</title>

</head>

<body>

<h1>Hello, Welcome to Tilak's World of Devops!!!</h1>

<p>This is a plain HTML page for Dockerizing a Plain HTML Page with Nginx.</p>

</body>

</html>#-------------------------------------------------------------------

**Nginx Configuration:**

**Since we are hosting the Webserver using nginx below is the configuration file that we need to copy to the container using docker file**

Create an Nginx configuration file named nginx.conf that serves the index.html page.

Configure Nginx to listen on port 80.

#-----------------------------nginx.conf-------------------------

events {}

http {

server {

listen 80;

location / {

root /usr/share/nginx/html;

index index.html;

}

}

}

#----------------------------------------------------------------

Dockerfile:

Create a Dockerfile to define the Docker image.

Use an official Nginx base image.

Copy the index.html and nginx.conf files into the appropriate location in the container.

Ensure that the Nginx server is started when the container is run.

#---------------------------Dockerfile----------------------

FROM nginx:latest

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

COPY nginx.conf /etc/nginx/nginx.conf

#-----------------------------------------------------------

Building the Docker Image:

Build the Docker image using the Dockerfile.

docker build -t webserver:v1 -f dockerfile .

A black screen with white text

Description automatically generated

Now we Test the image

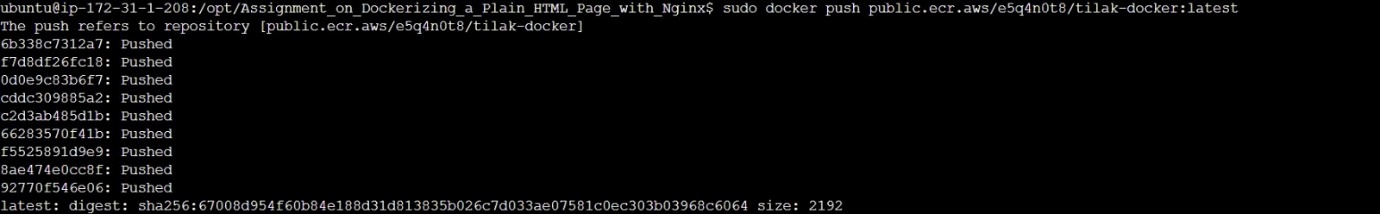
docker run -it -d -p 8080:80 webserver:v1

A computer screen with white text

Description automatically generated

Push the image on ECR

Make the public repository and push them on the ECR



A screenshot of a computer

Description automatically generated

public.ecr.aws/e5q4n0t8/tilak-docker:latest