Documentation AWS-Docker Assingment

Sahiru Galappaththi

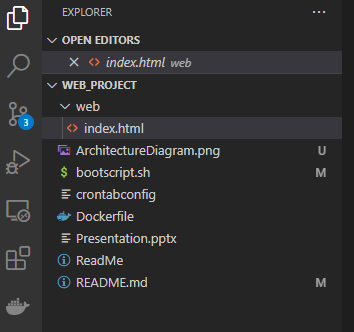
Introduction

Docker is used to create a containerized environment to host multiple webservers. NGINX is the type of webserver in use. Dockerfile was added into the project directory which holds the vital steps. These steps will be explained in detail in the following sections.

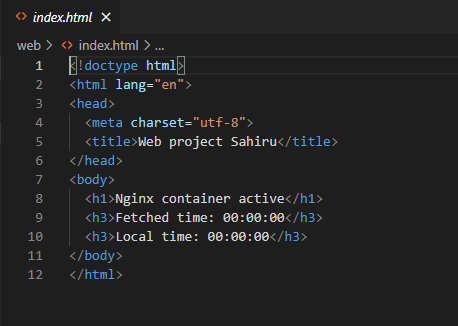
A bootstrap script is attached to the Docker image, this script will retrieve the time from <http://worldtimeapi.org/> and check with the local server time. These times will dynamically update the homepage as well.

How I approached the assignment

Use VScode to manipulate and edit the files as it’s easier,



Create the index.html file



Create the bootstrap script

#!/bin/bash

TIMEZONE=`cat /etc/timezone`

#Printing out server timezone

echo "This is the server timezone - $TIMEZONE"

#Reading the time from worldtime API for server time zone

WORLDTIMELINK="http://worldtimeapi.org/api/timezone/$TIMEZONE"

WORLD\_TIME=$(curl "$WORLDTIMELINK" | jq -r '.unixtime')

LOCAL\_TIME=$(date '+%s')

echo "Fetched UNIX Time: $WORLD\_TIME"

echo "Server UNIX Time: $LOCAL\_TIME"

#Check if local and world times are equal or not

if [ "$WORLD\_TIME" = "$LOCAL\_TIME" ]; then

    echo "Fetched Time and Server Time are equal."

else

    echo "Fetched Time and Server Time are not equal."

fi

#Update index.html with world time and local time

DISPLAY\_WORLD\_TIME=$(printf '%(%F %T)T\n' $WORLD\_TIME)

DISPLAY\_LOCAL\_TIME=$(printf '%(%F %T)T\n' $LOCAL\_TIME)

sed -i "/Fetched time/c\Fetched time: '$DISPLAY\_WORLD\_TIME'" /usr/share/nginx/html/index.html

sed -i "/Local time/c\Local time: '$DISPLAY\_LOCAL\_TIME'" /usr/share/nginx/html/index.html

#Checking response code of the server

STATUS\_CODE=$(curl --write-out "%{http\_code}\n" "http://localhost:80" --output output.txt --silent)

echo "Response Code from web server: $STATUS\_CODE"

if [ "$STATUS\_CODE" = "200" ]; then

    echo "Web server is responding the status code 200"

    echo "Server is responding as expected. " | mail -v -s "Server status report" rowsahiru@gmail.com

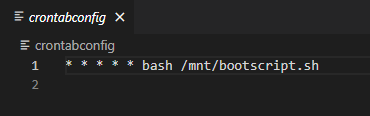
else

    echo "Web server is not responding the status code 200"

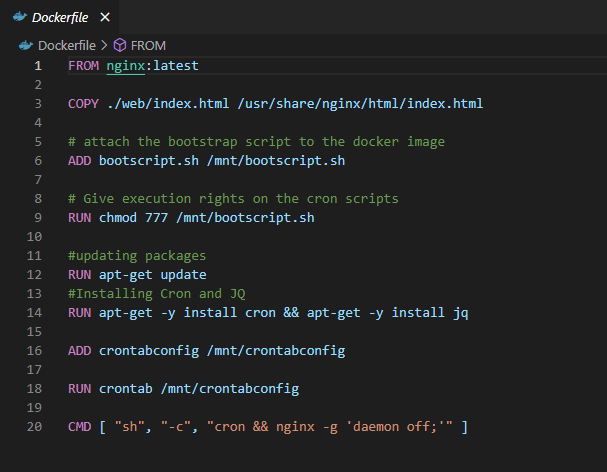
    echo "Server not returning code 200, please take a look. " | mail -v -s "Server status report" rowsahiru@gmail.com

fi

Create the Crontab config file and set it to run the bootstrap script every 1 minute

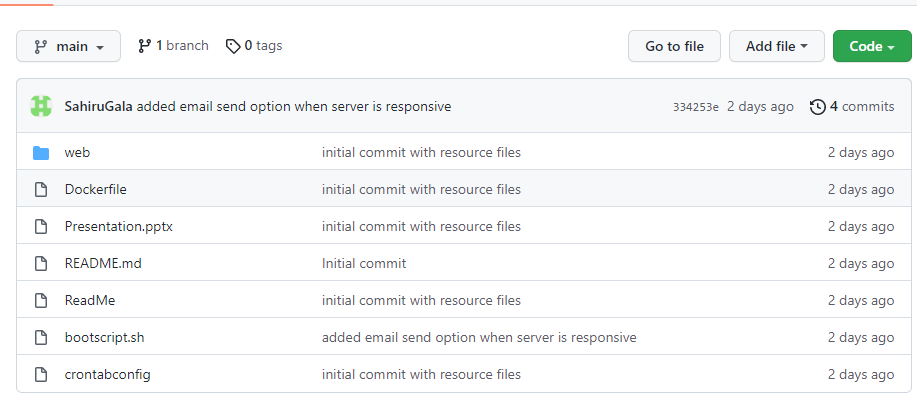


Create the Dockerfile in the same directory,



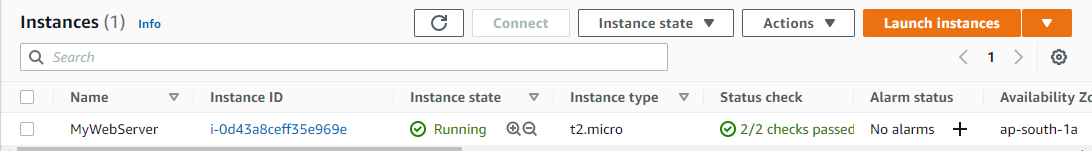
This file contains the necessary steps for our image to run.

Next step is to push these files to github.

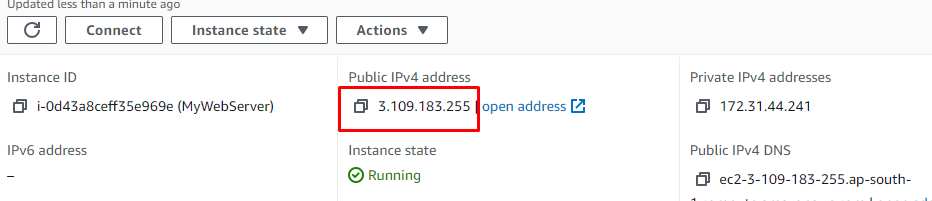


Setting up in Cloud environment.

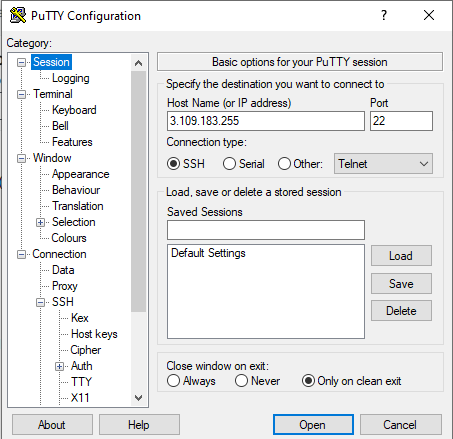
Login into amazon AWS and create an EC2 instance.



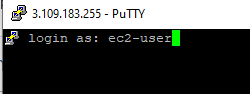
Use the following IP address to login to the instance.



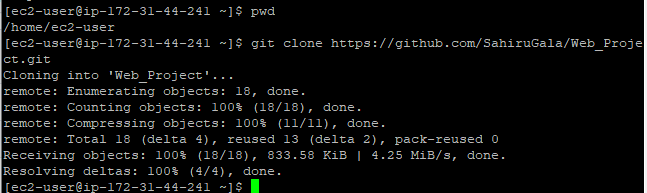
Using an app like PuTTY will be helpful.



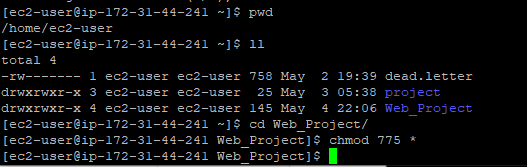
Login as ec2-user



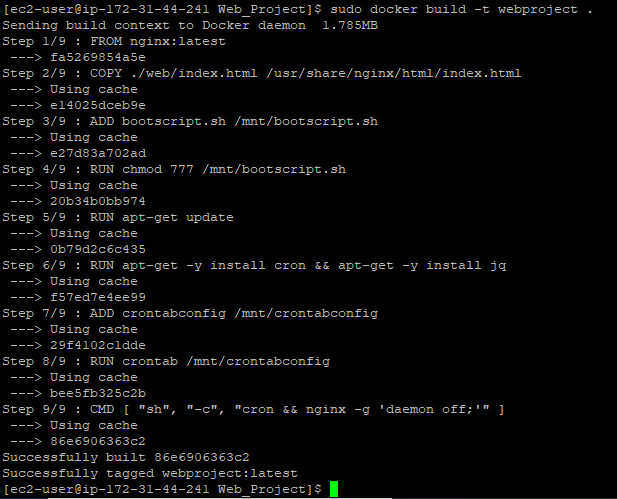
Clone the repo which I pushed to earlier,



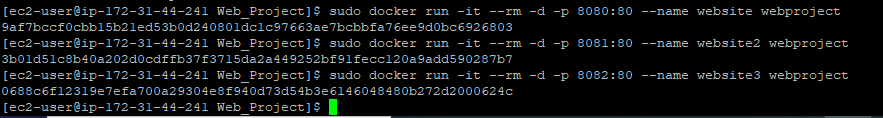
Move into the Web\_Project folder and give permissions to the files,



Build Image in aws environment.



Run the docker image 3 times on different ports,



To validate, retrieve the IP address again from EC2 instance and add port at the end,

