Documentation
Assessment – Part 01

Prepared by AGHU Dewmini

Table of Contents

Table of Figures

1. Introduction

This project has created a containerized environment that hosts multiple containers, each of which has its own NGINX web server. The bootstrap script will retrieve the current time zone's time from http://worldtimeapi.org/ and check to see if the server's local time is correct. And dynamically update the Home Page to include the fetched time, the local time, and the container from which the content is served. Using the 200-status code, this project also checks if the server is serving the expected content.

2. Deployed Steps

This project manually generated a container in the Ubuntu Virtual Environment. In this assessment, Docker is used for containerization. And also NGINX service used in web server.

```
CONTAINER ID   IMAGE       COMMAND                  CREATED              STATUS              PORTS                                       NAMES
9afb288bb5e2   webserver   "/docker-entrypoint.…"   3 seconds ago        Up 2 seconds        0.0.0.0:8081->80/tcp, :::8081->80/tcp       web2
fc7a791d921e   webserver   "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:8080->80/tcp, :::8080->80/tcp       web
```

Figure 1: Created 2 containers using Docker images

These containers are manually started and stopped. Kubernetes can be used to manage containers automatically. To manage how your containers run, AWS provides Amazon ECS and Amazon EKS.

Here created container has NGINX web servers.

```
root@fc7a791d921e:/#
root@fc7a791d921e:/#
root@fc7a791d921e:/# service nginx status
nginx is running.
root@fc7a791d921e:/#
root@fc7a791d921e:/#
root@fc7a791d921e:/# cd /etc/nginx/
root@fc7a791d921e:/etc/nginx# ls
conf.d  fastcgi_params  mime.types  modules  nginx.conf  scgi_params  uwsgi_params
root@fc7a791d921e:/etc/nginx#
root@fc7a791d921e:/etc/nginx#
```

Figure 3: NGINX is running in containers

```
FROM nginx:latest

COPY ./site-content/index.html /usr/share/nginx/html/index.html

# Add the script to the Docker Image
ADD bootstrap_script.sh /mnt/bootstrap_script.sh

# Give execution rights on the cron scripts
RUN chmod 0644 /mnt/bootstrap_script.sh

#Install Cron and JQ
RUN apt-get update && apt-get -y install cron && apt-get -y install jq

ADD crontab_source /mnt/crontab_source

RUN crontab /mnt/crontab_source

#CMD ["cron", "-f"]
CMD [ "sh", "-c", "cron && nginx -g 'daemon off;'" ]
```

Figure 2: Snapshot of Dockerfile

NGINX server is manually created inside the Docker file (Figure 01 first line).

```
#!/bin/bash


# Reading the server timezone

TIMEZONE=`cat /etc/timezone`
echo "Server time zone is: $TIMEZONE"


#Reading the time from worldtime API for server time zone

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

FETCH_UNIX_TIME=$(curl "$URL" | jq -r '.unixtime')
SERVER_UNIX_TIME=$(date '+%s')

echo "Fetched UNIX Time: $FETCH_UNIX_TIME"
echo "Server UNIX Time: $SERVER_UNIX_TIME"


# Comapre the Fetched and Server times in unix format
if [ "$FETCH_UNIX_TIME" = "$SERVER_UNIX_TIME" ]; then
    echo "Fetched Time and Server TIme are equal."
else
    echo "Fetched Time and Server TIme are not equal."
fi


# Update the fetched time and server time dynamically in the web server

DISPLAY_FETCH_TIME=$(printf '%(%F %T)T\n' $FETCH_UNIX_TIME)
DISPLAY_SERVER_TIME=$(printf '%(%F %T)T\n' $SERVER_UNIX_TIME)


sed -i "/Fetched time/c\Fetched time: '$DISPLAY_FETCH_TIME'" /usr/share/nginx/html/index.html
sed -i "/Local time/c\Local time: '$DISPLAY_SERVER_TIME'" /usr/share/nginx/html/index.html
#sed -i '/Container id/c\Container id: jsdhf' /usr/share/nginx/html/index.html
```

Figure 4: Snapshot of bootstrap_script.sh

This bootstrap script retrieves the current time zone from the provided API http://worldtimeapi.org/. And the response was converted to JSON using the jq tool. When you run the dockerfile, the jq tool is installed on the images (figure 02 line 12).

```
root@fc7a791d921e:/#
root@fc7a791d921e:/# curl http://worldtimeapi.org/api/timezone/Etc/UTC
{"abbreviation":"UTC","client_ip":"87.210.115.226","datetime":"2022-02-15T09:03:14.899922+00:00","day_of_week":2,"day_of_year":46,"dst":false,"dst_from":null,"dst_
0,"timezone":"Etc/UTC","unixtime":1644915794,"utc_datetime":"2022-02-15T09:03:14.899922+00:00","utc_offset":"+00:00","week_number":7}root@fc7a791d921e:/#
root@fc7a791d921e:/#
root@fc7a791d921e:/#
```

Figure 5: unixtime format was used in this project

In this project, the unixtime format was used instead of any other time format in API.

The fetch time and server time are then compared using the if-else command, which is in the same unix format. Then, both times, they are included in the web server's main page. The web server's main page.html file is contained within the side-content folder.

To test whether the web server is functioning properly, a curl request is generated using http code.

Using the cron job, the script was executed once every minute. When you run the dockerfile, cron jobs are automatically installed in your container.

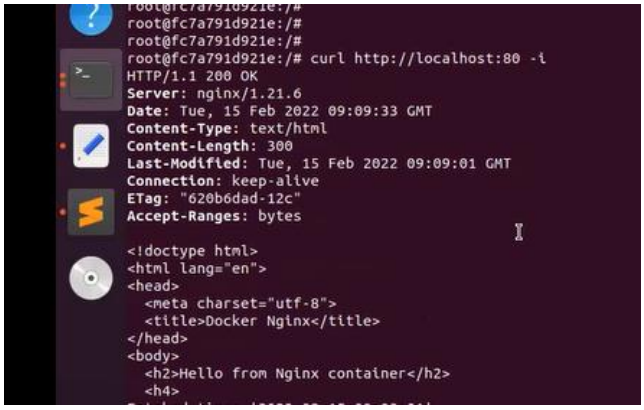

Figure 7: HTTP Status code to check web server status

Figure 6: server outputs

3. Diagram

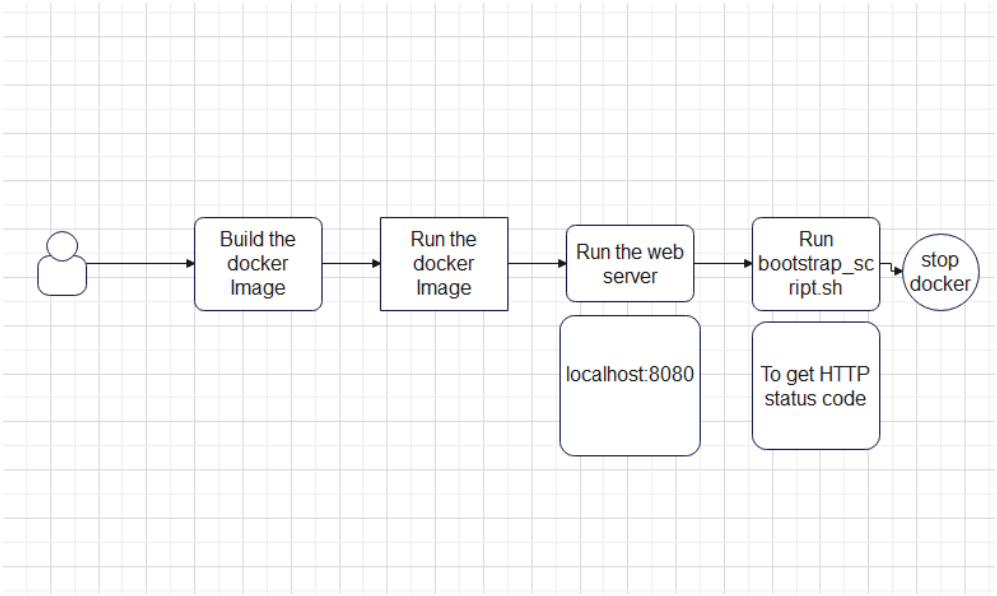

Figure 8: Project Flow chart