Lab 5 Task Part 2

Name: Muhammad Sherjeel Akhtar

Roll No: 20p-0101

Subject: Operating Systems Lab

Submitted To Respected Sir: Muhammad Ahsan

Section: BCS-4A

####786####

Task Two:

In this task you are going to deploy your python application using docker. Source code for the application and the docker configuration files are also attached in zip file. This app is simple python flask app which counts the number of refresh of the page and store data in redis.

Step-1:

In first step build the image using Dockerfile(attached in zip) and spin the new container using docker-compose(compose file attached in zip). Once container starts try to access this app in

your
browser using proper port number which is mentioned in compose file. Once app open in browser take
the screenshot of the initial message of the app on browser.

Step-2:

Now app is running now refresh your page 10 times and see the app response one 10 refresh completed take the screenshot of the screen.

Answer:

Docker Images:

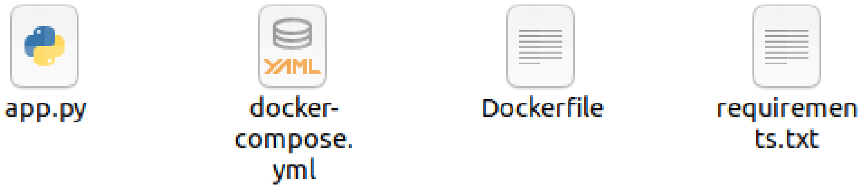
Use the command docker images first of all to see the available docker images.

Step 1:

First of all we’ve downloaded the Python Zip file from the given Lab Manual.
After extracting it, we’ve got two distinct files.

- Dockerfile
- docker-compose.yml

Visual Demonstration:



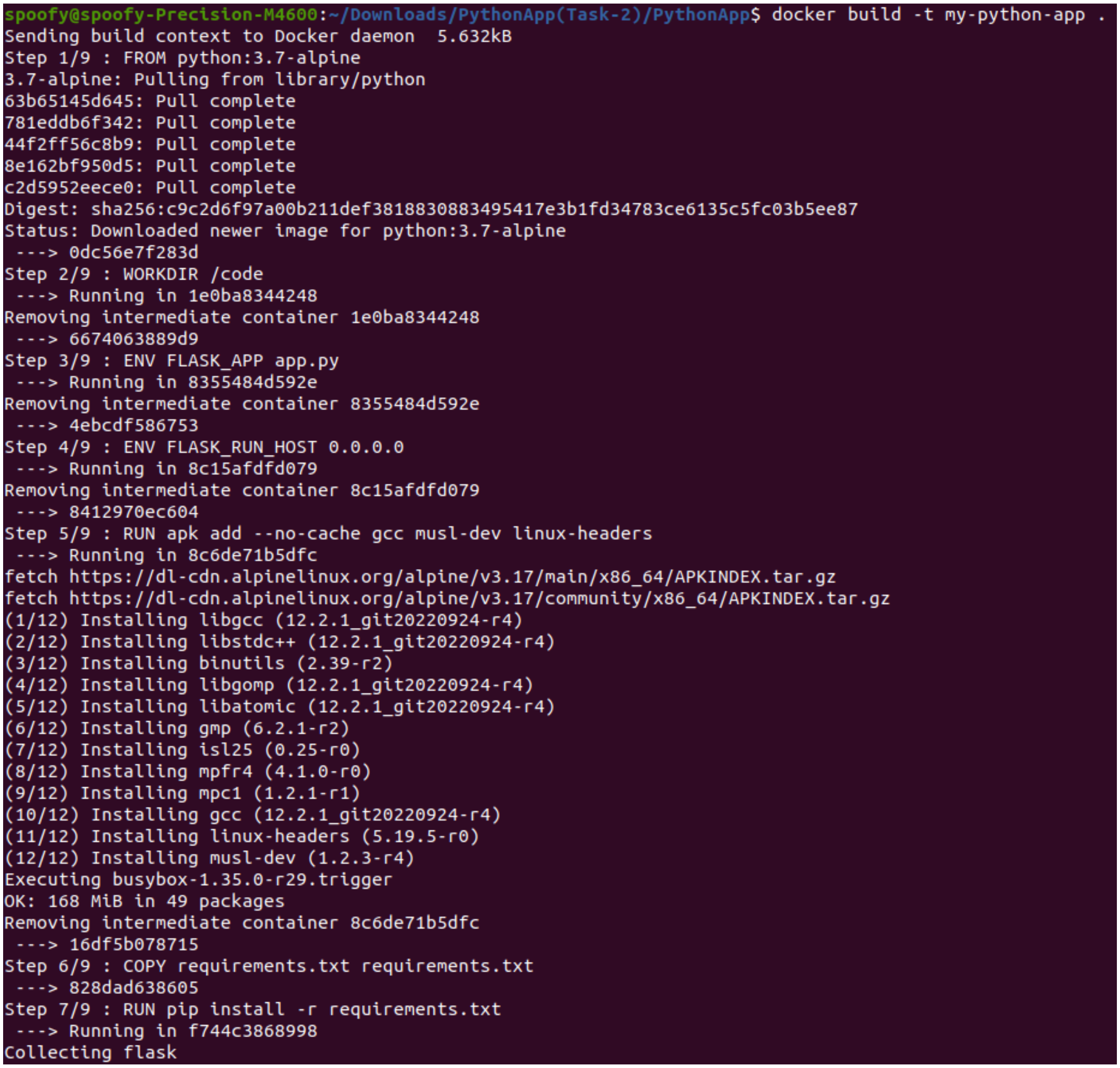
Step 2: Building Image using Dockerfile

Now we are going to build an image using the Dockerfile given in the zip folder.

```
spoofy@spoofy-Precision-M4600:~/Downloads/PythonApp(Task-2)/PythonApp$ docker build -t my-python-app .
```

Here “-t” is used to set the name of the image and “.” is representing the current directory.

Visual Demonstration:

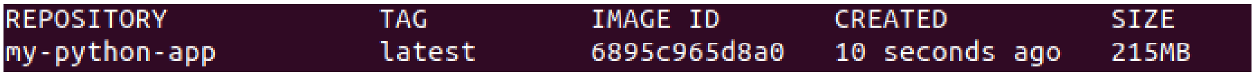


```
spoofy@spoofy-Precision-M4600:~/Downloads/PythonApp(Task-2)/PythonApp$ docker build -t my-python-app .
Sending build context to Docker daemon  5.632kB
Step 1/9 : FROM python:3.7-alpine
3.7-alpine: Pulling from library/python
63b65145d645: Pull complete
781eddb6f342: Pull complete
44f2ff56c8b9: Pull complete
8e162bf950d5: Pull complete
c2d5952eece0: Pull complete
Digest: sha256:c9c2d6f97a00b211def3818830883495417e3b1fd34783ce6135c5fc03b5ee87
Status: Downloaded newer image for python:3.7-alpine
 ---> 0dc56e7f283d
Step 2/9 : WORKDIR /code
 ---> Running in 1e0ba8344248
Removing intermediate container 1e0ba8344248
 ---> 6674063889d9
Step 3/9 : ENV FLASK_APP app.py
 ---> Running in 8355484d592e
Removing intermediate container 8355484d592e
 ---> 4ebcdf586753
Step 4/9 : ENV FLASK_RUN_HOST 0.0.0.0
 ---> Running in 8c15afdfd079
Removing intermediate container 8c15afdfd079
 ---> 8412970ec604
Step 5/9 : RUN apk add --no-cache gcc musl-dev linux-headers
 ---> Running in 8c6de71b5dfc
fetch https://dl-cdn.alpinelinux.org/alpine/v3.17/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.17/community/x86_64/APKINDEX.tar.gz
(1/12) Installing libgcc (12.2.1_git20220924-r4)
(2/12) Installing libstdc++ (12.2.1_git20220924-r4)
(3/12) Installing binutils (2.39-r2)
(4/12) Installing libgomp (12.2.1_git20220924-r4)
(5/12) Installing libatomic (12.2.1_git20220924-r4)
(6/12) Installing gmp (6.2.1-r2)
(7/12) Installing isl25 (0.25-r0)
(8/12) Installing mpfr4 (4.1.0-r0)
(9/12) Installing mpc1 (1.2.1-r1)
(10/12) Installing gcc (12.2.1_git20220924-r4)
(11/12) Installing linux-headers (5.19.5-r0)
(12/12) Installing musl-dev (1.2.3-r4)
Executing busybox-1.35.0-r29.trigger
OK: 168 MiB in 49 packages
Removing intermediate container 8c6de71b5dfc
 ---> 16df5b078715
Step 6/9 : COPY requirements.txt requirements.txt
 ---> 828dad638605
Step 7/9 : RUN pip install -r requirements.txt
 ---> Running in f744c3868998
Collecting flask
```

Using docker images:

Now use the command docker images observer the newly build image of “my-python-app”.

Visual Demonstration:



```
REPOSITORY              TAG         IMAGE ID        CREATED           SIZE
my-python-app           latest      6895c965d8a0    10 seconds ago    215MB
```

Step 2: Spinning the container using the docker compose command

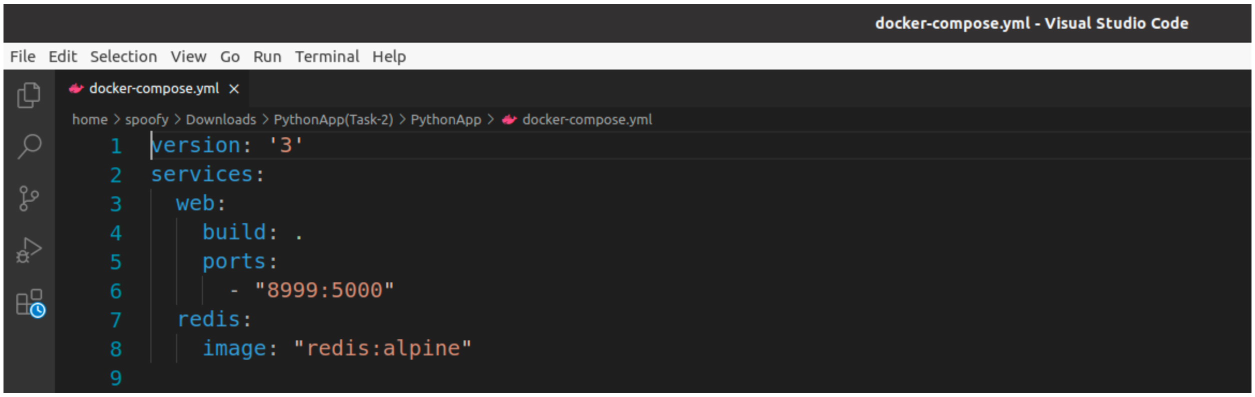
We’ve a docker compose file in our lab manual.

Now we are going to spin a docker container using this docker compose file.

For this purpose, go to the directory where your “**yaml**” file is present.
At there, run the following command:

```
spoofy@spoofy-Precision-M4600:~/Downloads/PythonApp(Task-2)/PythonApp$ docker-compose up
```

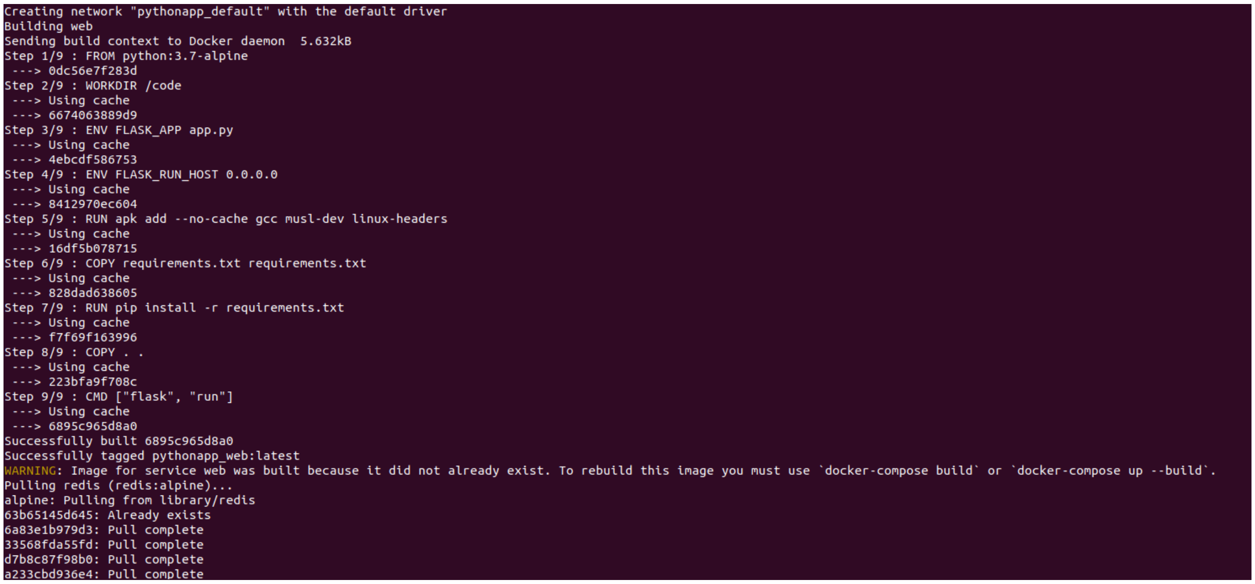
This will spin the container with the following configurations:


docker-compose.yml - Visual Studio Code
File Edit Selection View Go Run Terminal Help
docker-compose.yml
home > spoofy > Downloads > PythonApp(Task-2) > PythonApp > docker-compose.yml

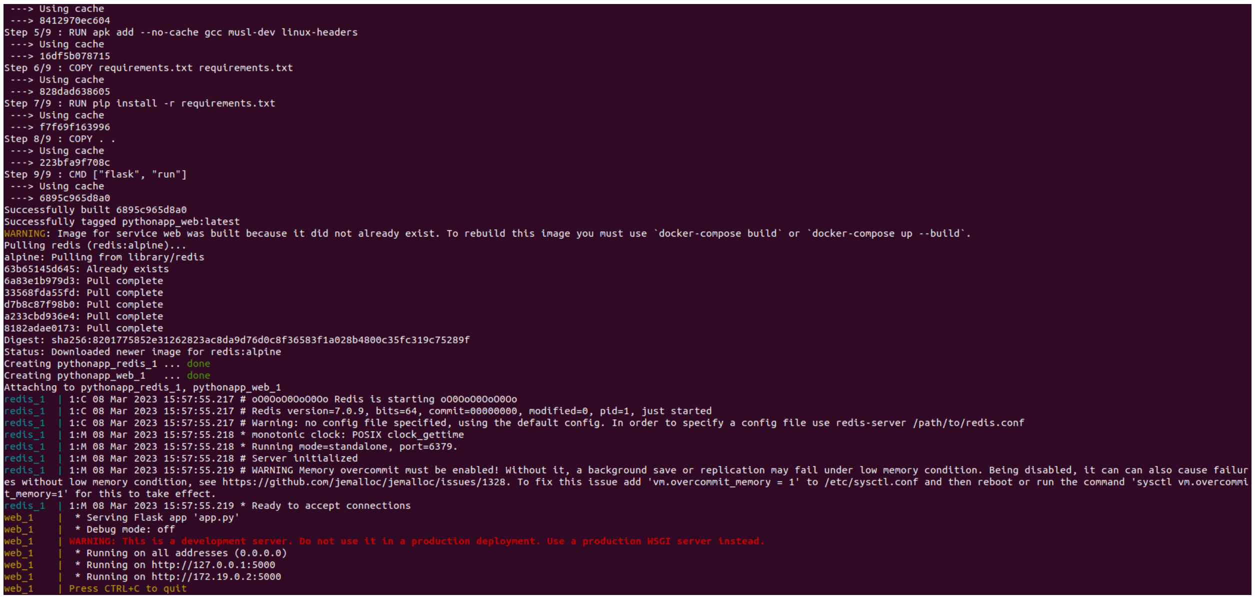
```
version: '3'
services:
  web:
    build: .
    ports:
      - "8999:5000"
  redis:
    image: "redis:alpine"
```

Overview of visual demonstration of terminal on running the “docker-compose up”

```
spoofy@spoofy-Precision-M4600:~/Downloads/PythonApp(Task-2)/PythonApp$ docker-compose up
```

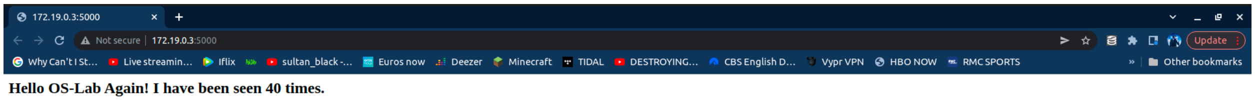


```
Creating network "pythonapp_default" with the default driver
Building web
Sending build context to Docker daemon  5.632kB
Step 1/9 : FROM python:3.7-alpine
 ---> 0dc56e7f283d
Step 2/9 : WORKDIR /code
 ---> Using cache
 ---> 6674063889d9
Step 3/9 : ENV FLASK_APP app.py
 ---> Using cache
 ---> 4ebcdf586753
Step 4/9 : ENV FLASK_RUN_HOST 0.0.0.0
 ---> Using cache
 ---> 8412970ec604
Step 5/9 : RUN apk add --no-cache gcc musl-dev linux-headers
 ---> Using cache
 ---> 16df5b078715
Step 6/9 : COPY requirements.txt requirements.txt
 ---> Using cache
 ---> 828dad638605
Step 7/9 : RUN pip install -r requirements.txt
 ---> Using cache
 ---> f7f69f163996
Step 8/9 : COPY . .
 ---> Using cache
 ---> 223bfa9f708c
Step 9/9 : CMD ["flask", "run"]
 ---> Using cache
 ---> 6895c965d8a0
Successfully built 6895c965d8a0
Successfully tagged pythonapp_web:latest
WARNING: Image for service web was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compose up --build`.
Pulling redis (redis:alpine)...
alpine: Pulling from library/redis
63b65145d645: Already exists
6a83e1b979d3: Pull complete
33568fda55fd: Pull complete
d7b8c87f98b0: Pull complete
a233cbd936e4: Pull complete
```

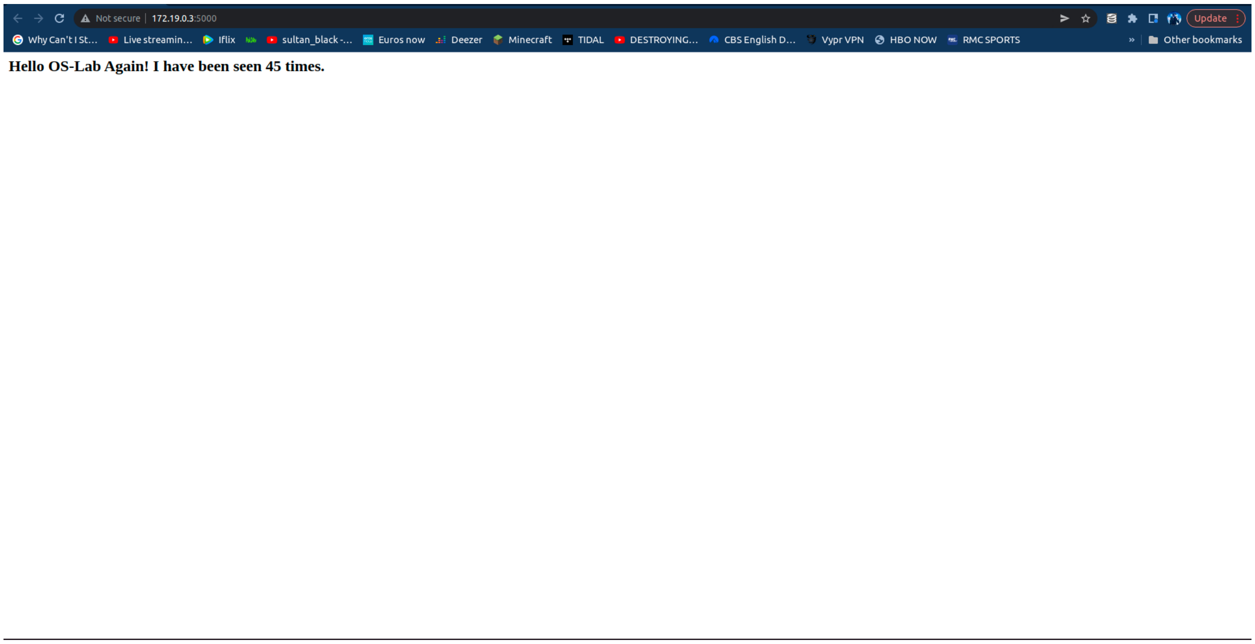


```
 ---> Using cache
 ---> 8412970ec604
Step 5/9 : RUN apk add --no-cache gcc musl-dev linux-headers
 ---> Using cache
 ---> 16df5b078715
Step 6/9 : COPY requirements.txt requirements.txt
 ---> Using cache
 ---> 828dad638605
Step 7/9 : RUN pip install -r requirements.txt
 ---> Using cache
 ---> f7f69f163996
Step 8/9 : COPY . .
 ---> Using cache
 ---> 223bfa9f708c
Step 9/9 : CMD ["flask", "run"]
 ---> Using cache
 ---> 6895c965d8a0
Successfully built 6895c965d8a0
Successfully tagged pythonapp_web:latest
WARNING: Image for service web was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compose up --build`.
Pulling redis (redis:alpine)...
alpine: Pulling from library/redis
63b65145d645: Already exists
6a83e1b979d3: Pull complete
33568fda55fd: Pull complete
d7b8c87f98b0: Pull complete
a233cbd936e4: Pull complete
8182adae0173: Pull complete
Digest: sha256:8201775852e31262823ac8da9d76d0c8f36583f1a028b4800c35fc319c75289f
Status: Downloaded newer image for redis:alpine
Creating pythonapp_redis_1 ... done
Creating pythonapp_web_1   ... done
Attaching to pythonapp_redis_1, pythonapp_web_1
redis_1  | 1:C 08 Mar 2023 15:57:55.217 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
redis_1  | 1:C 08 Mar 2023 15:57:55.217 # Redis version=7.0.9, bits=64, commit=00000000, modified=0, pid=1, just started
redis_1  | 1:C 08 Mar 2023 15:57:55.217 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
redis_1  | 1:M 08 Mar 2023 15:57:55.218 * monotonic clock: POSIX clock_gettime
redis_1  | 1:M 08 Mar 2023 15:57:55.218 * Running mode=standalone, port=6379.
redis_1  | 1:M 08 Mar 2023 15:57:55.218 # Server initialized
redis_1  | 1:M 08 Mar 2023 15:57:55.219 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can can also cause failur
es without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommi
t_memory=1' for this to take effect.
redis_1  | 1:M 08 Mar 2023 15:57:55.219 * Ready to accept connections
web_1    |  * Serving Flask app 'app.py'
web_1    |  * Debug mode: off
web_1    | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
web_1    |  * Running on all addresses (0.0.0.0)
web_1    |  * Running on http://127.0.0.1:5000
web_1    |  * Running on http://172.19.0.2:5000
web_1    | Press CTRL+C to quit
```

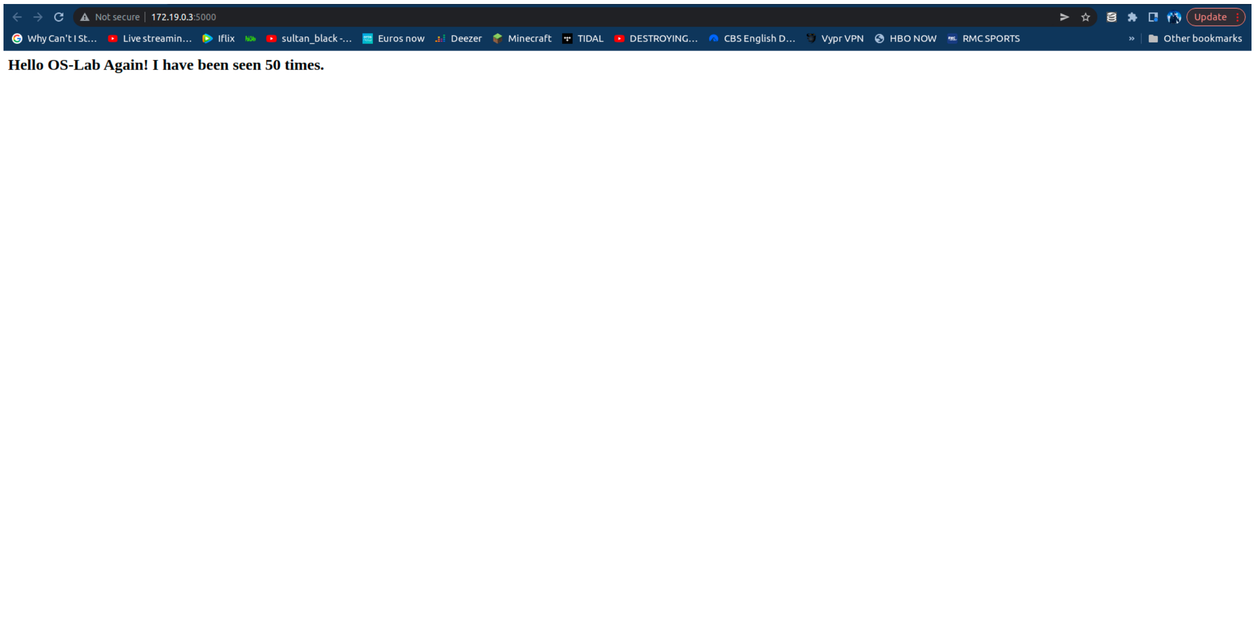
Visual Demonstration:



Hello OS-Lab Again! I have been seen 40 times.

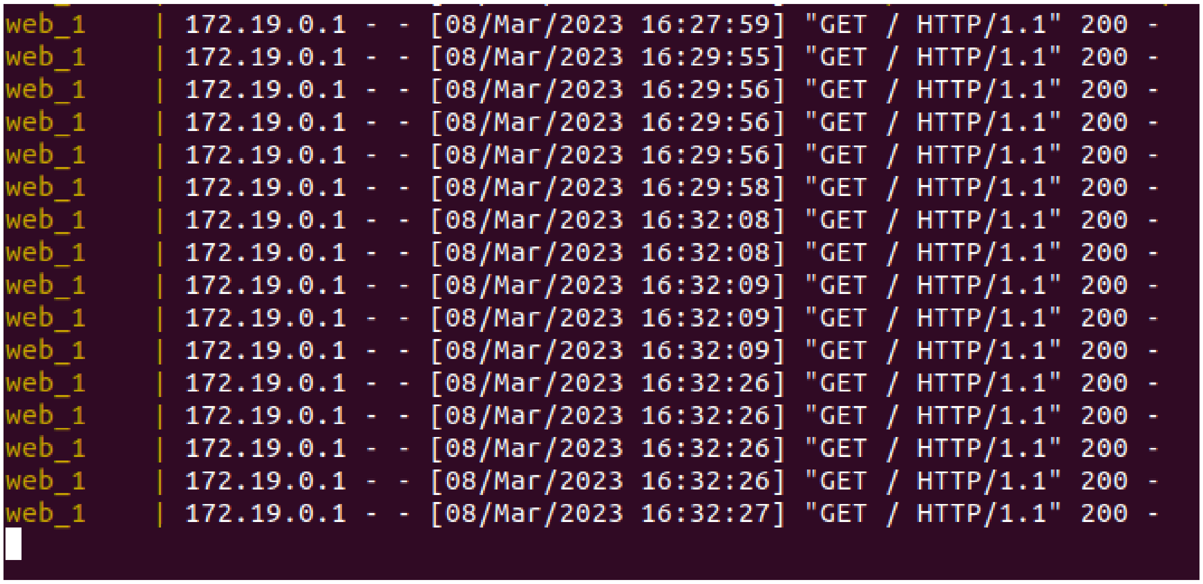


Hello OS-Lab Again! I have been seen 45 times.



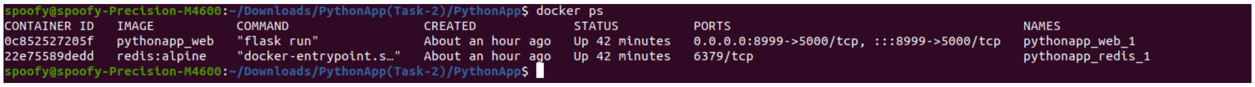
Hello OS-Lab Again! I have been seen 50 times.

Visualization Of Requests Via Terminal:



```
web_1        | 172.19.0.1 - - [08/Mar/2023 16:27:59] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:29:55] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:29:56] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:29:56] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:29:56] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:29:58] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:08] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:08] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:09] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:09] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:09] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:26] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:26] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:26] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:26] "GET / HTTP/1.1" 200 -
web_1        | 172.19.0.1 - - [08/Mar/2023 16:32:27] "GET / HTTP/1.1" 200 -
```

Running Containers:

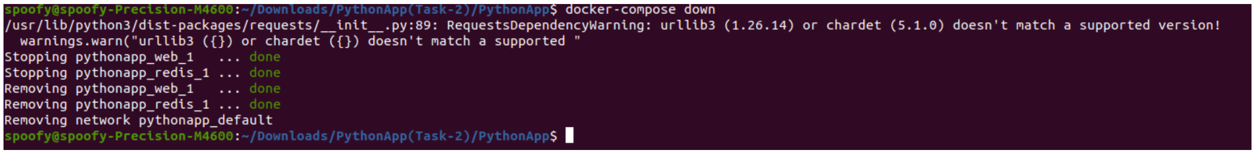


```
spoofy@spoofy-Precision-M4600:~/Downloads/PythonApp(Task-2)/PythonApp$ docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED             STATUS          PORTS                                         NAMES
0c852527205f   pythonapp_web    "flask run"              About an hour ago   Up 42 minutes   0.0.0.0:8999->5000/tcp, :::8999->5000/tcp   pythonapp_web_1
22e75589dedd   redis:alpine     "docker-entrypoint.s…"   About an hour ago   Up 42 minutes   6379/tcp                                      pythonapp_redis_1
spoofy@spoofy-Precision-M4600:~/Downloads/PythonApp(Task-2)/PythonApp$ 
```

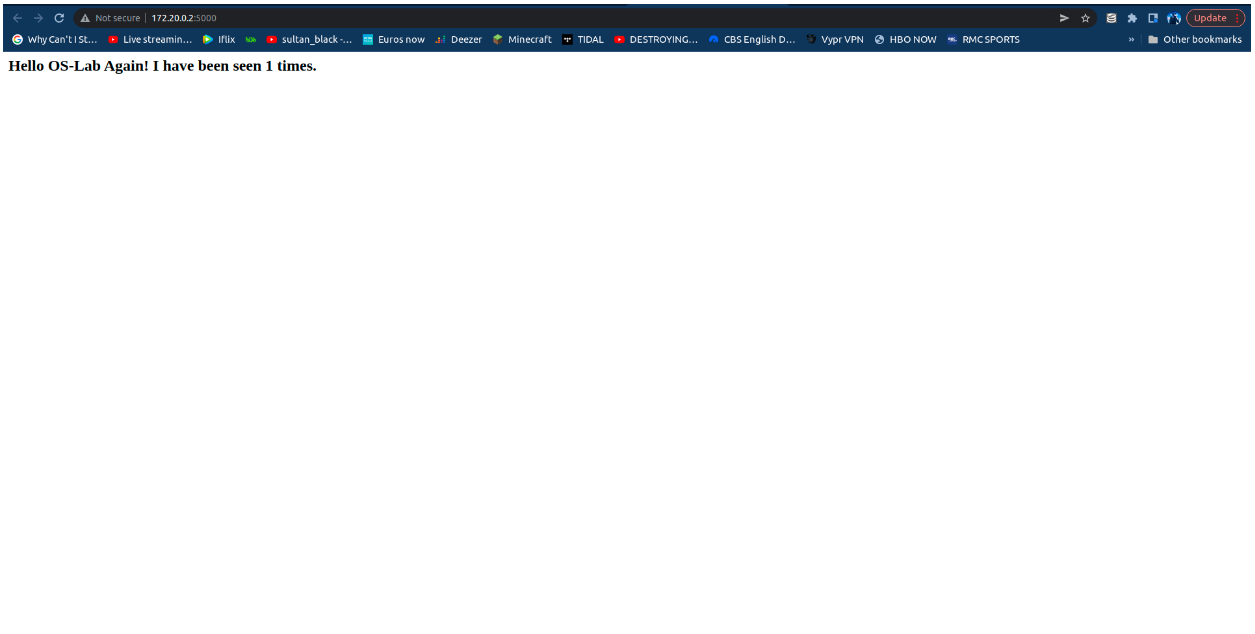
Down A Container:

In order to down a container, use the command docker-compose down.

Visual Demonstration:



```
spoofy@spoofy-Precision-M4600:~/Downloads/PythonApp(Task-2)/PythonApp$ docker-compose down
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning: urllib3 (1.26.14) or chardet (5.1.0) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "
Stopping pythonapp_web_1   ... done
Stopping pythonapp_redis_1 ... done
Removing pythonapp_web_1   ... done
Removing pythonapp_redis_1 ... done
Removing network pythonapp_default
spoofy@spoofy-Precision-M4600:~/Downloads/PythonApp(Task-2)/PythonApp$ 
```



```
Hello OS-Lab Again! I have been seen 1 times.
```

FIN————————————————————————!