INSTITUT SUPÉRIEUR D'ÉLECTRONIQUE DE PARIS (ISEP)

COURSE: GOVERNANCE AND ARCHITECTURE OF INFORMATION SYSTEMS

ASSIGNMENT: A first small Python web application

SUBMITTED BY

BUHARI ALIYU 62788

December 2023

1 A first small Python web application

1.1 The operations to be carried out are:

1) we begin by creating a Flask directory and an AppliOne sub-directory in which we will place the Python file

$ mkdir -p /Docker/Flask/AppliOne

```
buharialiyu@Buhari:~$ sudo mkdir -p /Docker/Flask/AppliOne
```

$ cd /Docker/Flask/AppliOne

```
buharialiyu@Buhari:/$ cd /Docker/Flask/AppliOne
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

2) write the file app.py in the AppliOne directory

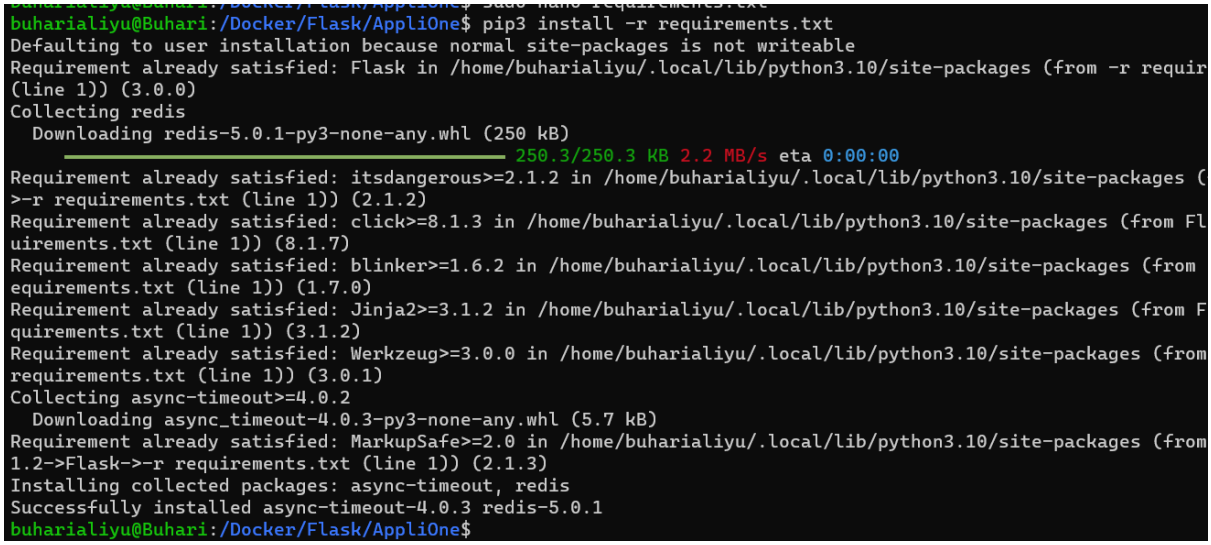
buharialiyu@Buhari: /Docker/

GNU nano 6.2 app.py *

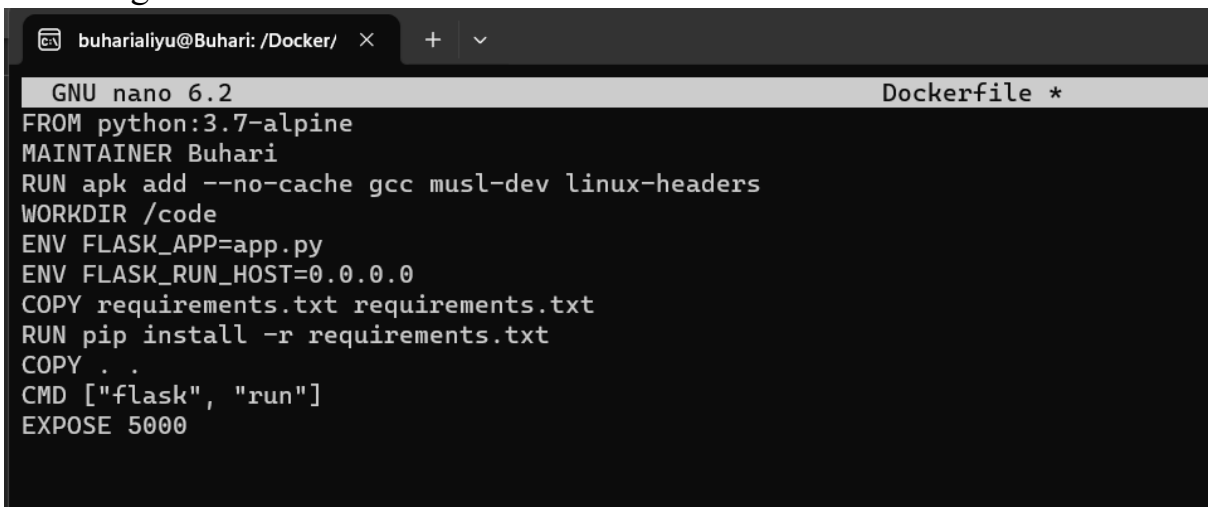
```
import redis
import time

from flask import Flask
app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)

def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)

@app.route("/")
def home():
    count = get_hit_count()
    return "Hello world! I have been seen {} times.".format(count)
```

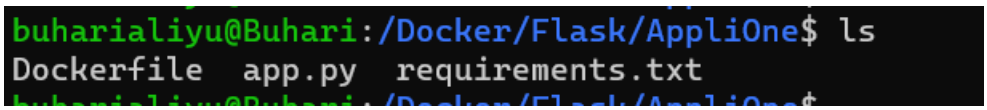
```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ ls
app.py
```

3) write a requirements.txt file containing the list of packages and dependencies needed to run the web application. This file, which will be executed via the pip install -r requirements.txt command, contains two flask redis lines



4) we move on to writing the Dockerfile that will be used to obtain the image of the container in charge of running the application. This image should naturally contain all the dependencies required by the Python application, including Python itself. To do this in a dockerfile, enter the following lines





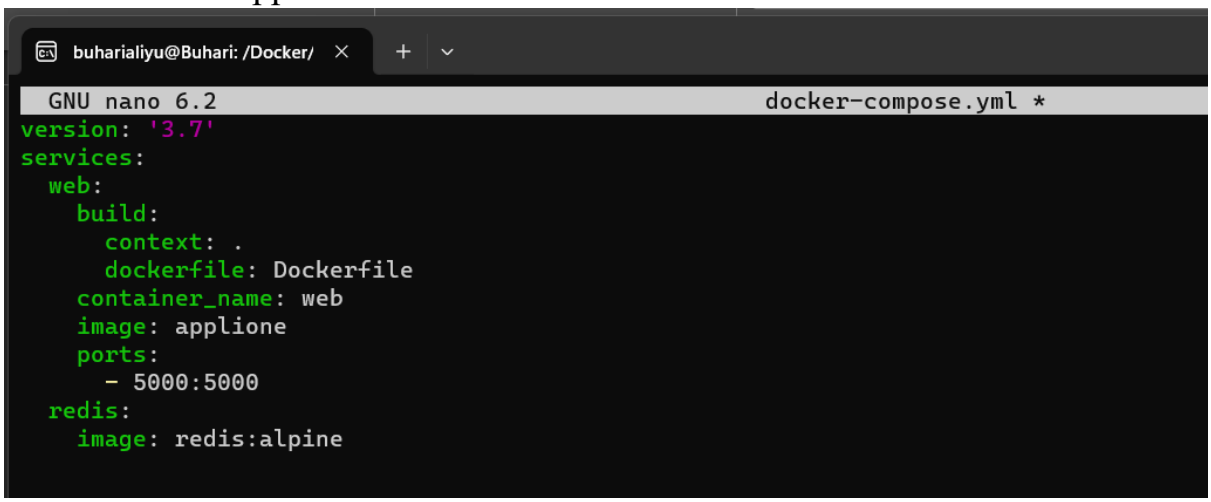
apt install docker-compose

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ sudo  apt install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz python3-attr python
  python3-docker python3-dockerpty python3-docopt python3-dotenv python3-idna python3-j
  python3-requests python3-texttable python3-urllib3 python3-websocket runc ubuntu-fan
```

service docker restart

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ sudo service docker restart
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

5) enter the file docker-compose.yml in the directory /Docker/Flask/AppliOne



6) launch docker-compose and look at the displays produced, in particular the warnings

$ docker-compose up -d

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker-compose up -d
Starting web                ... done
Starting applione_redis_1 ... done
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

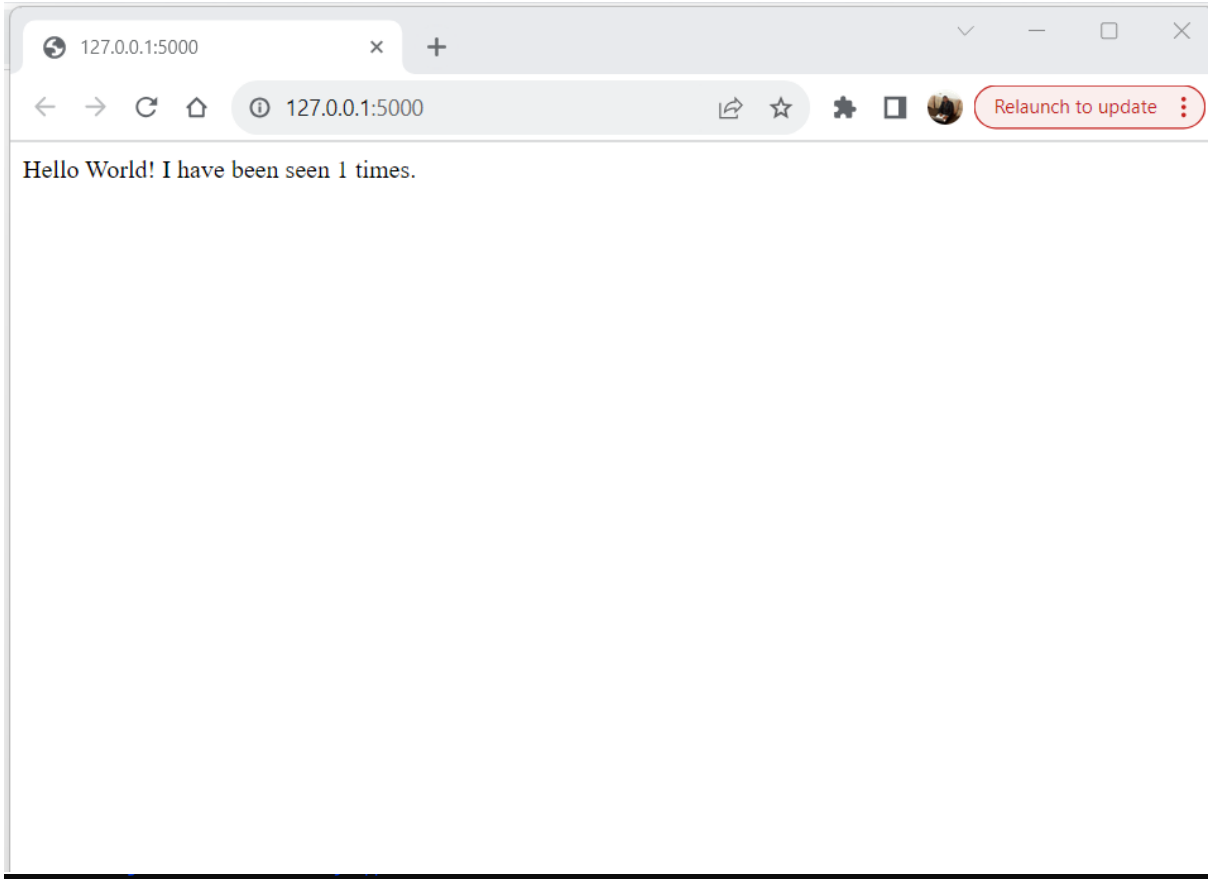
```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker-compose up --build
Removing web
Building web
[+] Building 3.8s (12/12) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                          0.1s
 => => transferring dockerfile: 309B                                                                                          0.0s
 => [internal] load .dockerignore                                                                                             0.1s
 => => transferring context: 2B                                                                                               0.0s
 => [internal] load metadata for docker.io/library/python:3.7-alpine                                                          3.4s
 => [auth] library/python:pull token for registry-1.docker.io                                                                 0.0s
 => [internal] load build context                                                                                             0.0s
 => => transferring context: 1.16kB                                                                                           0.0s
 => [1/6] FROM docker.io/library/python:3.7-alpine@sha256:f3d31c8677d03f0b3c724446077f229a6ce9d3ac430f5c08cd7dff0             0.0s
 => CACHED [2/6] RUN apk add --no-cache gcc musl-dev linux-headers                                                            0.0s
 => CACHED [3/6] WORKDIR /code                                                                                                0.0s
 => CACHED [4/6] COPY requirements.txt requirements.txt                                                                       0.0s
 => CACHED [5/6] RUN pip install -r requirements.txt                                                                          0.0s
 => [6/6] COPY . .                                                                                                            0.0s
 => exporting to image                                                                                                        0.1s
 => => exporting layers                                                                                                       0.0s
 => => writing image sha256:9490df47b476d3813d6a5e7a79a900a2018b1848dfc87a7bfc7a39d94f596a1b                                  0.0s
 => => naming to docker.io/library/applione                                                                                   0.0s
applione_redis_1 is up-to-date
Recreating e4a1c1df0257_web ... done
Attaching to applione_redis_1, web
web      |  * Serving Flask app 'app.py'
web      |  * Debug mode: off
web      | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
```

```
applione_redis_1 is up-to-date
Recreating e4a1c1df0257_web ... done
Attaching to applione_redis_1, web
web      |  * Serving Flask app 'app.py'
web      |  * Debug mode: off
web      | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
 instead.
web      |  * Running on all addresses (0.0.0.0)
web      |  * Running on http://127.0.0.1:5000
web      |  * Running on http://172.28.0.3:5000
web      | Press CTRL+C to quit
redis_1  | 1:C 30 Dec 2023 19:01:44.340 # WARNING Memory overcommit must be enabled! Without it, a background save or re
plication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition,
see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.co
nf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
redis_1  | 1:C 30 Dec 2023 19:01:44.342 * oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
redis_1  | 1:C 30 Dec 2023 19:01:44.342 * Redis version=7.2.3, bits=64, commit=00000000, modified=0, pid=1, just started
redis_1  | 1:C 30 Dec 2023 19:01:44.342 # Warning: no config file specified, using the default config. In order to speci
fy a config file use redis-server /path/to/redis.conf
redis_1  | 1:M 30 Dec 2023 19:01:44.343 * monotonic clock: POSIX clock_gettime
redis_1  | 1:M 30 Dec 2023 19:01:44.343 * Running mode=standalone, port=6379.
redis_1  | 1:M 30 Dec 2023 19:01:44.345 * Server initialized
redis_1  | 1:M 30 Dec 2023 19:01:44.345 * Ready to accept connections tcp
^CGracefully stopping... (press Ctrl+C again to force)
Stopping web               ... done
Stopping applione_redis_1  ... done
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

7) list running containers with docker ps

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker ps

CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS          PORTS                    NAMES
e973afbe58d4   applione       "flask run"              2 minutes ago   Up 30 seconds   0.0.0.0:5000->5000/tcp   web
52e3bd1459ea   redis:alpine   "docker-entrypoint.s…"   6 hours ago     Up 31 seconds   6379/tcp                 applione
_redis_1
1689c044aaba   ubuntu         "/bin/bash"              11 days ago     Up 11 days                               ubuntu
buharialiyu@Buhari:/Docker/Flask/AppliOne$
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

8) connect to the container's web server with a browser on the host machine and observe messages;




```
Creating web              ... done
Creating applione_redis_1 ... done
Attaching to applione_redis_1, web
redis_1  | 1:C 30 Dec 2023 13:44:26.140 # WARNING Memory overcommit must be enabled! Without it, a background save or re
plication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition,
see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.co
nf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
redis_1  | 1:C 30 Dec 2023 13:44:26.140 * oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
redis_1  | 1:C 30 Dec 2023 13:44:26.141 * Redis version=7.2.3, bits=64, commit=00000000, modified=0, pid=1, just started
redis_1  | 1:C 30 Dec 2023 13:44:26.141 # Warning: no config file specified, using the default config. In order to speci
fy a config file use redis-server /path/to/redis.conf
redis_1  | 1:M 30 Dec 2023 13:44:26.142 * monotonic clock: POSIX clock_gettime
redis_1  | 1:M 30 Dec 2023 13:44:26.144 * Running mode=standalone, port=6379.
redis_1  | 1:M 30 Dec 2023 13:44:26.145 * Server initialized
redis_1  | 1:M 30 Dec 2023 13:44:26.145 * Ready to accept connections tcp
web      |  * Serving Flask app 'app.py'
web      |  * Debug mode: off
web      | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
 instead.
web      |  * Running on all addresses (0.0.0.0)
web      |  * Running on http://127.0.0.1:5000
web      |  * Running on http://172.20.0.3:5000
web      | Press CTRL+C to quit
web      | 172.20.0.1 - - [30/Dec/2023 13:44:48] "GET / HTTP/1.1" 200 -
web      | 172.20.0.1 - - [30/Dec/2023 13:44:48] "GET /favicon.ico HTTP/1.1" 404 -
```

9) display the docker-compose logs.

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker-compose logs
Attaching to web, applione_redis_1
web      |  * Serving Flask app 'app.py'
web      |  * Debug mode: off
web      | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
 instead.
web      |  * Running on all addresses (0.0.0.0)
web      |  * Running on http://127.0.0.1:5000
web      |  * Running on http://172.28.0.3:5000
web      | Press CTRL+C to quit
web      |  * Serving Flask app 'app.py'
web      |  * Debug mode: off
web      | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
 instead.
web      |  * Running on all addresses (0.0.0.0)
web      |  * Running on http://127.0.0.1:5000
web      |  * Running on http://172.28.0.3:5000
web      | Press CTRL+C to quit
web      | 172.28.0.1 - - [31/Dec/2023 00:47:34] "GET / HTTP/1.1" 200 -
redis_1  | 1:C 30 Dec 2023 19:01:44.340 # WARNING Memory overcommit must be enabled! Without it, a background save or re
plication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition,
see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.co
nf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
redis_1  | 1:C 30 Dec 2023 19:01:44.342 * oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
redis_1  | 1:C 30 Dec 2023 19:01:44.342 * Redis version=7.2.3, bits=64, commit=00000000, modified=0, pid=1, just started
redis_1  | 1:C 30 Dec 2023 19:01:44.342 # Warning: no config file specified, using the default config. In order to speci
fy a config file use redis-server /path/to/redis.conf
redis_1  | 1:M 30 Dec 2023 19:01:44.343 * monotonic clock: POSIX clock_gettime
redis_1  | 1:M 30 Dec 2023 19:01:44.343 * Running mode=standalone, port=6379.
redis_1  | 1:M 30 Dec 2023 19:01:44.345 * Server initialized
```

10) stop the application via docker-compose down and check that the corresponding containers have been stopped and deleted

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker-compose down
Stopping web              ... done
Stopping applione_redis_1 ... done
Removing web              ... done
Removing applione_redis_1 ... done
Removing network applione_default
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED       STATUS       PORTS     NAMES
1689c044aaba   ubuntu    "/bin/bash"   11 days ago   Up 11 days             ubuntu
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

1) a mount point via volumes, i.e. make the current directory ., mount as /code ;

```
  GNU nano 6.2                                          docker-compose.yml
version: '3.7'
services:
  web:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: web
    image: applione
    ports:
      - 5000:5000
    volumes:
      - .:/code
    environment:
      FLASK_ENV: development
  redis:
    image: redis:alpine
```

2) the FLASK_ENV environment variable with the value development.

to begin with, we'll make a backup copy of the Dockerfile
$ cd /Docker/Flask/AppliOne
$ cp dockerfile dockerfile.old

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ ls
Dockerfile  Dockerfile.old  app.py  docker-compose.yml  requirements.txt
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

we also modify the requirements.txt file to add GUNICORN, flask, gunicorn, redis

```
 GNU nano 6.2                                        requirements.txt *
Flask
redis
gunicorn
```

3) The next step is to modify the dockerfile as follows

```
 GNU nano 6.2                                          Dockerfile *
FROM python:3.7-alpine
MAINTAINER Buhari
RUN apk add --no-cache gcc musl-dev linux-headers
WORKDIR /code
ENV APP_ENVIRONMENT=DEV
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
COPY . .
CMD ["./boot.sh"]
EXPOSE 5000
```

Create the boot.sh Script:

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ sudo nano boot.sh
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

```
 GNU nano 6.2                                          boot.sh
#!/bin/bash
set -e

if [ "$APP_ENVIRONMENT" = 'DEV' ]; then
  echo "Running Development Server"
  exec flask run -h 0.0.0.0
else
  echo "Running Production Server"
  exec gunicorn -b :5000 --access-logfile - --error-logfile - app_name:app
fi
```

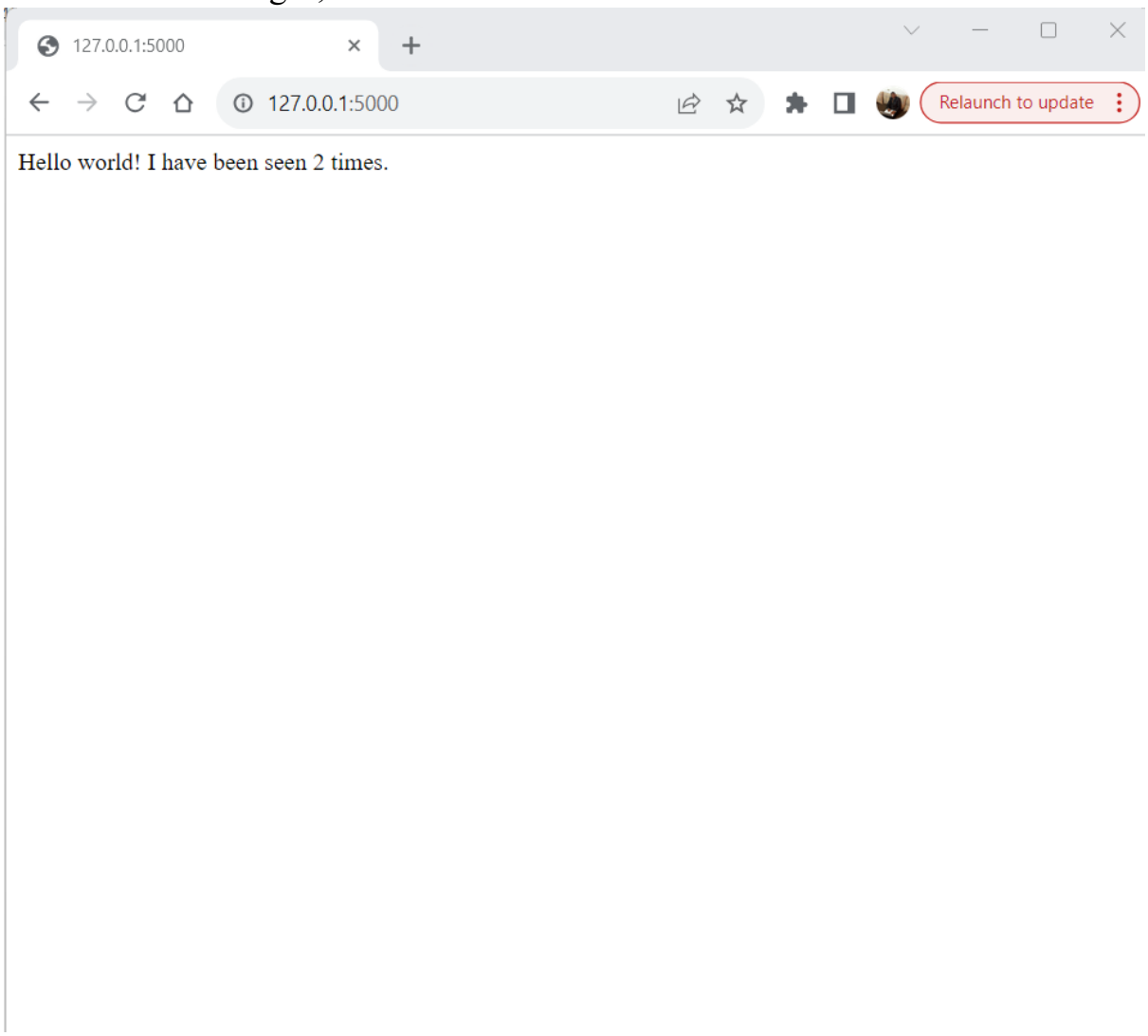
4) launch docker-compose and look at the displays produced, in particular the warnings
$ docker-compose up -d

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker-compose up -d
Creating network "applione_default" with the default driver
Creating web                ... done
Creating applione_redis_1 ... done
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

5) list running containers with docker ps

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                    NAMES
3e4cb792102b   applione       "flask run"              33 seconds ago   Up 31 seconds   0.0.0.0:5000->5000/tcp   web
dc0b71d2e2de   redis:alpine   "docker-entrypoint.s…"   33 seconds ago   Up 31 seconds   6379/tcp                 applion
e_redis_1
1689c044aaba   ubuntu         "/bin/bash"              11 days ago      Up 11 days                               ubuntu
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

6) connect to the container's web server with a browser on the host machine and observe messages;



7) display docker-compose logs

$ docker-compose logs

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker-compose logs
Attaching to web, applione_redis_1
redis_1  | 1:C 31 Dec 2023 01:03:10.290 # WARNING Memory overcommit must be enabled! Without it, a background save or re
plication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition,
see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.co
nf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
redis_1  | 1:C 31 Dec 2023 01:03:10.290 * oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
redis_1  | 1:C 31 Dec 2023 01:03:10.290 * Redis version=7.2.3, bits=64, commit=00000000, modified=0, pid=1, just started
redis_1  | 1:C 31 Dec 2023 01:03:10.290 # Warning: no config file specified, using the default config. In order to speci
fy a config file use redis-server /path/to/redis.conf
redis_1  | 1:M 31 Dec 2023 01:03:10.290 * monotonic clock: POSIX clock_gettime
redis_1  | 1:M 31 Dec 2023 01:03:10.291 * Running mode=standalone, port=6379.
redis_1  | 1:M 31 Dec 2023 01:03:10.291 * Server initialized
redis_1  | 1:M 31 Dec 2023 01:03:10.291 * Ready to accept connections tcp
web      | Running Development Server
web      |  * Serving Flask app 'app.py'
web      |  * Debug mode: on
web      | 'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
web      | 'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
web      | 'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
web      | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
 instead.
web      |  * Running on all addresses (0.0.0.0)
web      |  * Running on http://127.0.0.1:5000
web      |  * Running on http://172.30.0.3:5000
web      | Press CTRL+C to quit
web      |  * Restarting with stat
web      | 'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
web      | 'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
web      | 'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
web      |  * Debugger is active!
web      |  * Debugger PIN: 100-128-920
web      | 172.30.0.1 - - [31/Dec/2023 01:24:07] "GET / HTTP/1.1" 200 -
web      | 172.30.0.1 - - [31/Dec/2023 01:24:31] "GET / HTTP/1.1" 200 -
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```

8) stop the application via docker-compose down and check that the corresponding containers have been stopped and deleted.

```
buharialiyu@Buhari:/Docker/Flask/AppliOne$ docker-compose down
Stopping web              ... done
Stopping applione_redis_1 ... done
Removing web              ... done
Removing applione_redis_1 ... done
Removing network applione_default
buharialiyu@Buhari:/Docker/Flask/AppliOne$
```