Docker Part 2 Assignment – 3

1. Create 5 custom container with 5 different default pages

2. Using Docker Compose deploy these 5 containers on port 81, 82, 83, 84 and 85 respectively

- A docker-compose file is created with a manifesting script to create 5 containers on ports 81,82,83,84 & 85
- And default pages are customised by mounting filepaths of customised html files to each container

```
ubuntu@ip-172-31-45-255:~$ vi docker-compose.yml
ubuntu@ip-172-31-45-255:~$ cat docker-compose.yml
version: '3'

services:
  container1:
    image: nginx:latest
    ports:
      - "81:80"
    volumes:
      - ./container1:/usr/share/nginx/html
    container_name: container1

  container2:
    image: nginx:latest
    ports:
      - "82:80"
    volumes:
      - ./container2:/usr/share/nginx/html
    container_name: container2

  container3:
    image: nginx:latest
    ports:
      - "83:80"
    volumes:
      - ./container3:/usr/share/nginx/html
    container_name: container3

  container4:
    image: nginx:latest
    ports:
      - "84:80"
    volumes:
      - ./container4:/usr/share/nginx/html
    container_name: container4

  container5:
    image: nginx:latest
    ports:
      - "85:80"
    volumes:
      - ./container5:/usr/share/nginx/html
    container_name: container5

ubuntu@ip-172-31-45-255:~$
```

- Then the 5 directories are created for hosting each container's html files naming container1 to container5 in the same directory where docker-compose.yml file is created.

```
ubuntu@ip-172-31-45-255:~$ ls
container1  container2  container3  container4  container5  docker-compose.yml  dockerfile  index.html  mountbind  snap
ubuntu@ip-172-31-45-255:~$
```

- Then navigated into each directory and created an customised html file for each container

```
ubuntu@ip-172-31-45-255:~$ cd container1
ubuntu@ip-172-31-45-255:~/container1$ ls
index.html
ubuntu@ip-172-31-45-255:~/container1$ cat index.html
<html>
        <title>"Venkata Narayana"</title>
        <body>"This is container1"</body>
</html>
ubuntu@ip-172-31-45-255:~/container1$
```

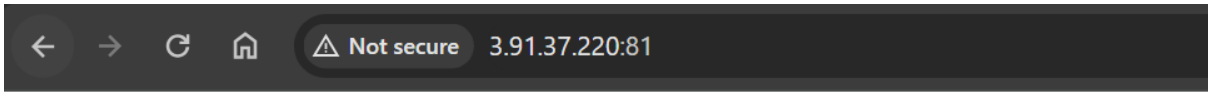
- Finally 5 containers are created by running “docker-compose up -d” command

```
ubuntu@ip-172-31-45-255:~/container1$ docker-compose up -d
Creating network "ubuntu_default" with the default driver
Creating container2 ... done
Creating container4 ... done
Creating container3 ... done
Creating container1 ... done
Creating container5 ... done
ubuntu@ip-172-31-45-255:~/container1$
```

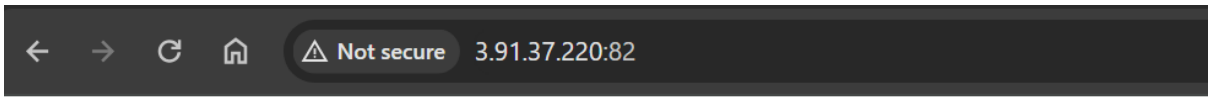
- Ensuring the docker compose containers are running successfully and ports are allocated properly to the respective containers

```
ubuntu@ip-172-31-45-255:~/container1$ docker compose ps
NAME         IMAGE          COMMAND                                  SERVICE      CREATED          STATUS          PORTS
container1   nginx:latest   "/docker-entrypoint.sh nginx -g 'daemon off;'"   container1   48 seconds ago   Up 45 seconds   0.0.0.0:81->80/tcp
, :::81->80/tcp
container2   nginx:latest   "/docker-entrypoint.sh nginx -g 'daemon off;'"   container2   48 seconds ago   Up 46 seconds   0.0.0.0:82->80/tcp
, :::82->80/tcp
container3   nginx:latest   "/docker-entrypoint.sh nginx -g 'daemon off;'"   container3   48 seconds ago   Up 45 seconds   0.0.0.0:83->80/tcp
, :::83->80/tcp
container4   nginx:latest   "/docker-entrypoint.sh nginx -g 'daemon off;'"   container4   48 seconds ago   Up 46 seconds   0.0.0.0:84->80/tcp
, :::84->80/tcp
container5   nginx:latest   "/docker-entrypoint.sh nginx -g 'daemon off;'"   container5   48 seconds ago   Up 45 seconds   0.0.0.0:85->80/tcp
, :::85->80/tcp
```

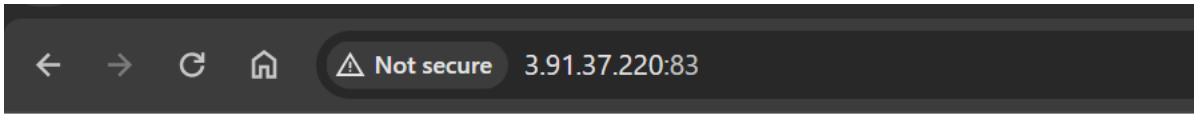
- Created containers are successfully accessible on their respective ports on local browser



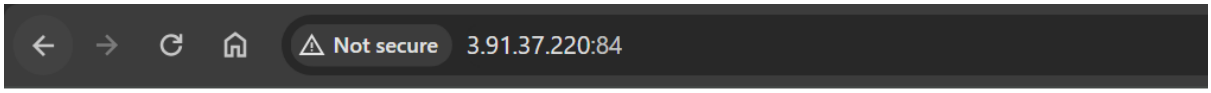
"This is container1"



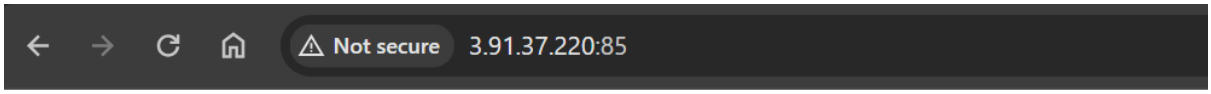
This is container2



"This is container3"



"This is container4"



"This is container5"