Mohanad Mahmoud Sayed

1. Task4:

- Run the image httpd again without attaching any volumes

```
[root@localhost ~]# docker run -dit --name my_httpd -p 8080:80 httpd
5af4642419c3bb9ff71e46485b855b93f2a9d0caf0ca18c8dac36ad33e2c6e12
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED         STATUS         PORTS                                       NAMES
5af4642419c3   httpd     "httpd-foreground"   3 seconds ago   Up 2 seconds   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   my_httpd
```

- Add html static files to the container and make sure they are accessible

```
[root@localhost ~]# docker cp /home/Task3/html/index.html my_httpd:/usr/local/apache2/htdocs/index.html
Successfully copied 2.05kB to my_httpd:/usr/local/apache2/htdocs/index.html
```

→Confirming access:


```
[root@localhost ~]# docker cp /home/Task3/html/index.html my_httpd:/usr/local/apache2/htdocs/index.html
Successfully copied 2.05kB to my_httpd:/usr/local/apache2/htdocs/index.html
```


- Commit the container with image name IMAGE_NAME

```
[root@localhost ~]# docker commit my_httpd mohanad_httpd_image
sha256:034e1ff285b91a10e4c74cf5190bfaded1383247fd820850dcec980fe555b2f1
```

- Create a dockerfile for the previous image and build the image from this dockerfile

```
  GNU nano 5.6.1                                                Dockerfile
FROM mohanad_httpd_image
COPY index.html /usr/local/apache2/htdocs/index.html
```

```
[root@localhost Task4]# mkdir docker_from_commit
[root@localhost Task4]# cd docker_from_commit/
[root@localhost docker_from_commit]# nano Dockerfile
[root@localhost docker_from_commit]# nano Dockerfile
[root@localhost docker_from_commit]# cp /home/Task3/html/index.html .
[root@localhost docker_from_commit]# ls
Dockerfile  index.html
[root@localhost docker_from_commit]#
```

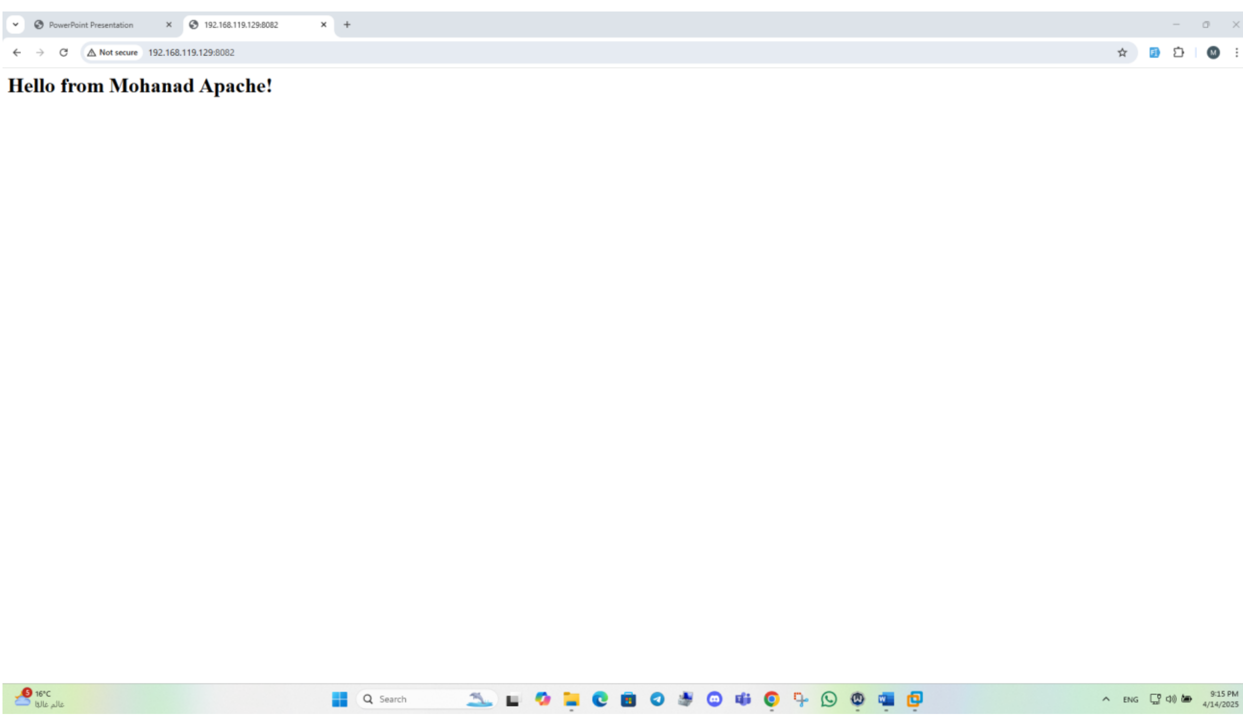
And now let's build the new image:

```
[root@localhost docker_from_commit]# docker build -t final_httpd_image .
[+] Building 0.1s (7/7) FINISHED                                                          docker:default
 => [internal] load build definition from Dockerfile                                                0.0s
 => => transferring dockerfile: 174B                                                                0.0s
 => [internal] load metadata for docker.io/library/mohanad_httpd_image:latest                       0.0s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load build context                                                                   0.0s
 => => transferring context: 132B                                                                   0.0s
 => [1/2] FROM docker.io/library/mohanad_httpd_image:latest                                         0.0s
 => [2/2] COPY index.html /usr/local/apache2/htdocs/index.html                                      0.0s
 => exporting to image                                                                              0.0s
 => => exporting layers                                                                             0.0s
 => => writing image sha256:bdfd86337e6f24448b2c40ea15c92b728f0f58e49a6f19170858258b4790bb8e        0.0s
 => => naming to docker.io/library/final_httpd_image                                                0.0s
[root@localhost docker_from_commit]#
```

```
[root@localhost docker_from_commit]# docker run -dit --name final_httpd -p 8082:80 final_httpd_image

391b0a296d172f4c0b2776181c50d31e8017fd7b6396ff9b7b57b287c60e88eb
[root@localhost docker_from_commit]# docker ps
CONTAINER ID   IMAGE               COMMAND              CREATED         STATUS         PORTS                                   NAMES
391b0a296d17   final_httpd_image   "httpd-foreground"   5 seconds ago   Up 5 seconds   0.0.0.0:8082->80/tcp, [::]:8082->80/tcp   final_httpd
```

Finally let's check by accessing container from my host machine browser:



2. Task5:

▪ Create a volume called mysql_data, then deploy a MySQL database called app-database. Use the mysql latest image, and use the -e flag to set MYSQL_ROOT_PASSWORD to P4sSw0rd0!.M ount the mysql_data volume to /var/lib/mysql. The container should run in the background.

- 1. Creating the volume:

```
[root@localhost ~]# docker volume create mysql_data
mysql_data
[root@localhost ~]# docker volume ls
DRIVER    VOLUME NAME
local     mysql_data
[root@localhost ~]#
```

- 2. Running the MySQL container with all options:

```
[root@localhost ~]# docker run -dit \
  --name app-database \
  -e MYSQL_ROOT_PASSWORD='P4sSw0rd0!.' \
  -v mysql_data:/var/lib/mysql \
  mysql:latest
c84c16950466f6d477f7446acfde8423de9bb009d27fea68aa3b149056a2dd91
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                 NAMES
c84c16950466   mysql:latest   "docker-entrypoint.s…"   14 seconds ago  Up 14 seconds  3306/tcp, 33060/tcp   app-database
```