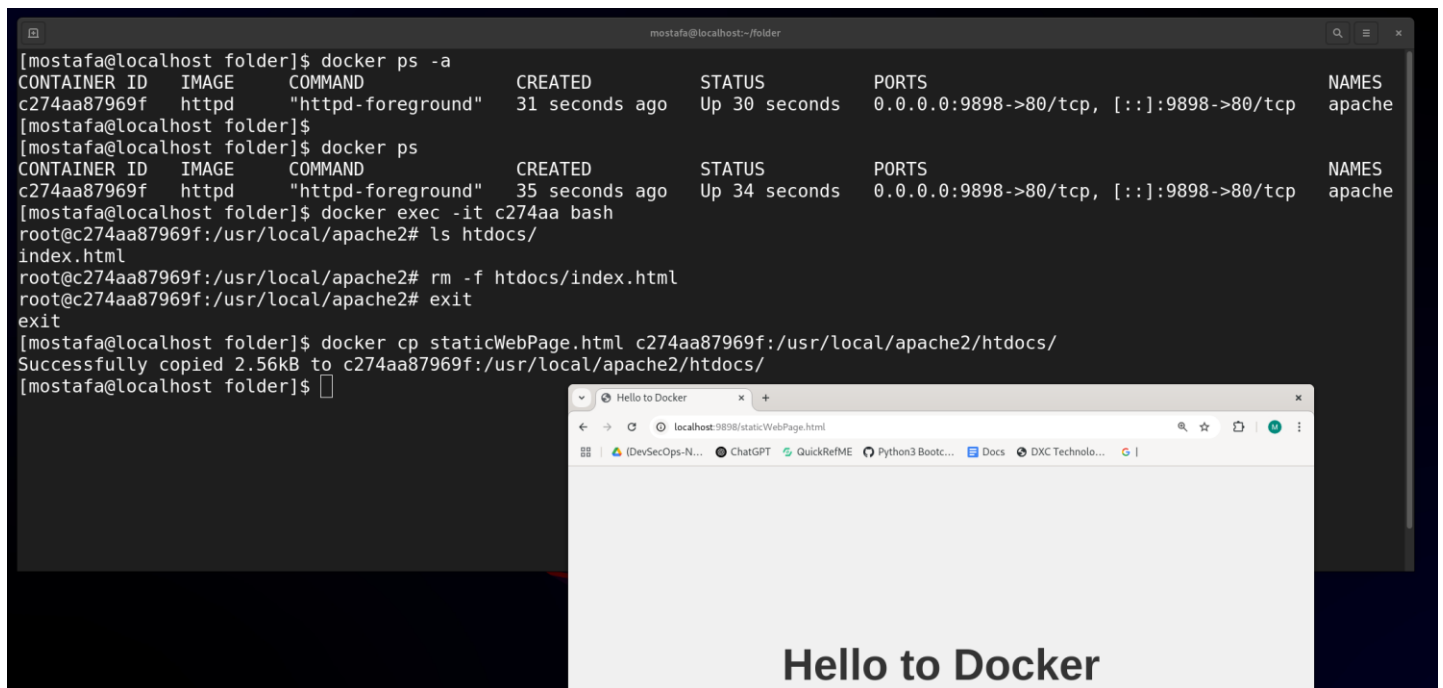


#### Problem 4:

#Run the image httpd again without attaching any volumes

> docker run -d httpd

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



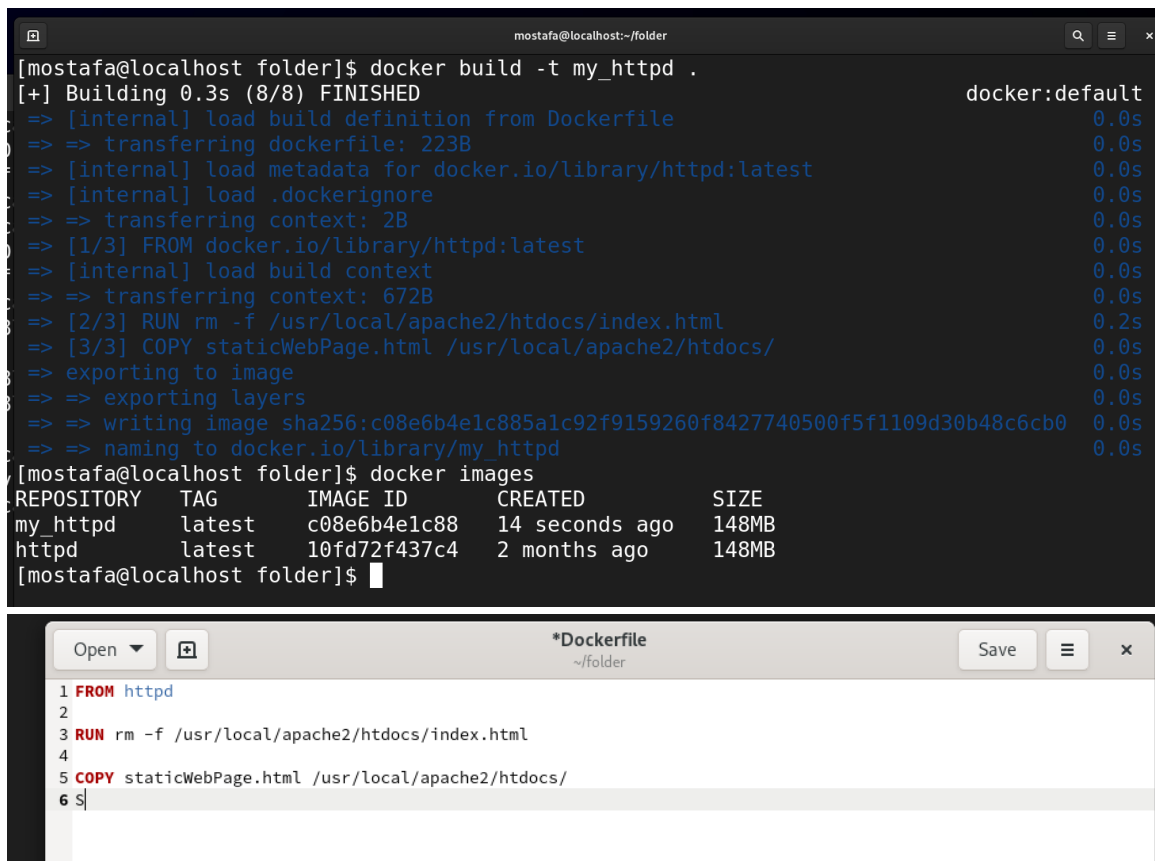
The image shows a terminal window and a web browser. The terminal window displays the following commands and output:

```
[mostafa@localhost folder]$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
c274aa87969f   httpd     "httpd-foreground"      31 seconds ago Up 30 seconds  0.0.0.0:9898->80/tcp, [::]:9898->80/tcp   apache
[mostafa@localhost folder]$
[mostafa@localhost folder]$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
c274aa87969f   httpd     "httpd-foreground"      35 seconds ago Up 34 seconds  0.0.0.0:9898->80/tcp, [::]:9898->80/tcp   apache
[mostafa@localhost folder]$ docker exec -it c274aa bash
root@c274aa87969f:/usr/local/apache2# ls htdocs/
index.html
root@c274aa87969f:/usr/local/apache2# rm -f htdocs/index.html
root@c274aa87969f:/usr/local/apache2# exit
exit
[mostafa@localhost folder]$ docker cp staticWebPage.html c274aa87969f:/usr/local/apache2/htdocs/
Successfully copied 2.56kB to c274aa87969f:/usr/local/apache2/htdocs/
[mostafa@localhost folder]$
```

The web browser shows a page titled "Hello to Docker" with the text "Hello to Docker" displayed in the center.

#Commit the container with image name IMAGE\_NAME

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



The image shows a terminal window and a Dockerfile editor. The terminal window displays the following commands and output:

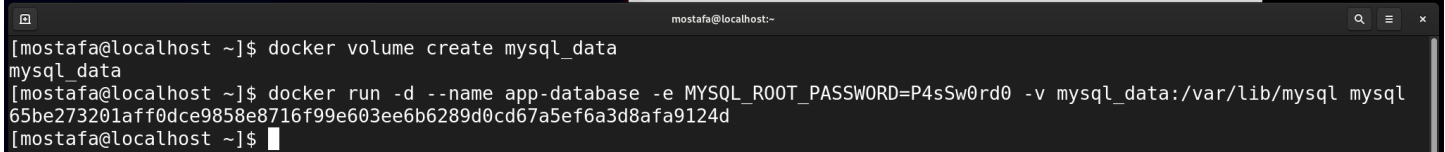
```
[mostafa@localhost folder]$ docker build -t my_httpd .
[+] Building 0.3s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 223B
=> [internal] load metadata for docker.io/library/httpd:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/httpd:latest
=> [internal] load build context
=> => transferring context: 672B
=> [2/3] RUN rm -f /usr/local/apache2/htdocs/index.html
=> [3/3] COPY staticWebPage.html /usr/local/apache2/htdocs/
=> exporting to image
=> => exporting layers
=> => writing image sha256:c08e6b4e1c885a1c92f9159260f8427740500f5f1109d30b48c6cb0
=> => naming to docker.io/library/my_httpd
[mostafa@localhost folder]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
my_httpd      latest    c08e6b4e1c88   14 seconds ago 148MB
httpd         latest    10fd72f437c4   2 months ago  148MB
[mostafa@localhost folder]$
```

The Dockerfile editor shows the following content:

```
1 FROM httpd
2
3 RUN rm -f /usr/local/apache2/htdocs/index.html
4
5 COPY staticWebPage.html /usr/local/apache2/htdocs/
6 $
```

### Problem 5:

#Create a volume calledmysql\_data, then deploy a MySQL database calledapp-database. Use the mysqllatestimage, and use the-e flag to setMYSQL\_ROOT\_PASSWORDtoP4sSw0rd0!. Mount themysql\_datavolume to/var/lib/mysql. The container should run in the background.

A terminal window with a dark background and light text. The window title is 'mostafa@localhost:~'. The first command is '[mostafa@localhost ~]\$ docker volume create mysql\_data', followed by the output 'mysql\_data'. The second command is '[mostafa@localhost ~]\$ docker run -d --name app-database -e MYSQL\_ROOT\_PASSWORD=P4sSw0rd0! -v mysql\_data:/var/lib/mysql mysql', followed by a long alphanumeric string '65be273201aff0dce9858e8716f99e603ee6b6289d0cd67a5ef6a3d8afa9124d'. The prompt returns to '[mostafa@localhost ~]\$' with a cursor.

```
[mostafa@localhost ~]$ docker volume create mysql_data
mysql_data
[mostafa@localhost ~]$ docker run -d --name app-database -e MYSQL_ROOT_PASSWORD=P4sSw0rd0! -v mysql_data:/var/lib/mysql mysql
65be273201aff0dce9858e8716f99e603ee6b6289d0cd67a5ef6a3d8afa9124d
[mostafa@localhost ~]$
```