**ITI Docker Lab1**

**Mazen AbdelTawab Saad**

What is the difference between:

CMD & ENTRYPOINT

COPY & ADD

CMD:

Specifies the default command that will run when a container starts. This command can be overridden by passing a different command at runtime. It's primarily used to provide default arguments for ENTRYPOINT or to specify a default command for the container.

ENTRYPOINT:

Defines the main command that will always run in the container. Unlike CMD, ENTRYPOINT is not overridden by commands passed at runtime. Instead, any additional commands are passed as arguments to ENTRYPOINT. It’s commonly used to set the container’s primary executable, making it behave like a dedicated service or application.

COPY:

Copies files and directories from the host filesystem into the container's filesystem. Simple and explicit—used for straightforward copying of local files.

ADD:

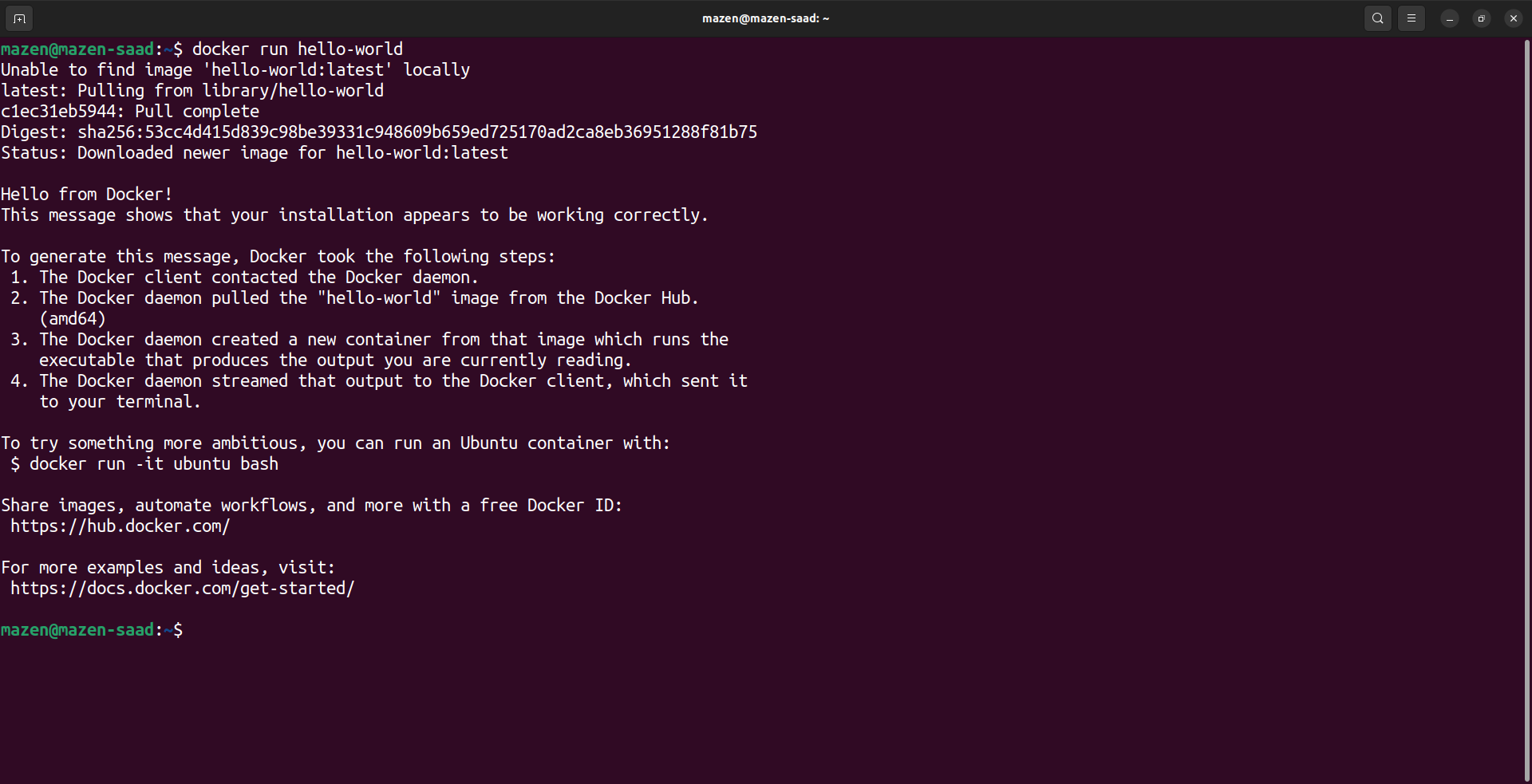
Also copies files and directories, but with additional functionality such as automatic extraction of compressed files (e.g., .tar) and the ability to download files from remote URLs. It is more powerful but can be less predictable than COPY.

Problem 1:

1 - Run the container hello-world:-

|

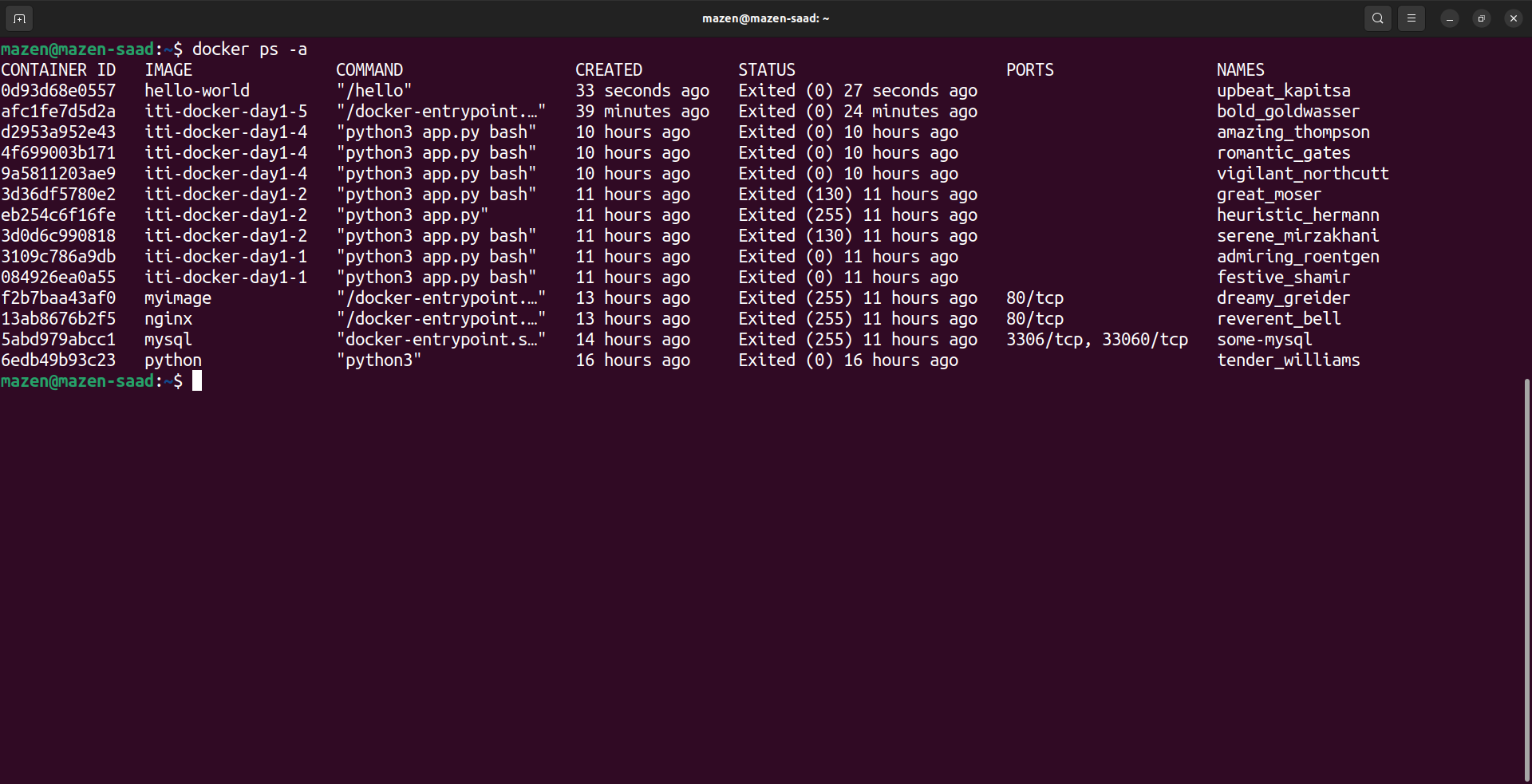
docker run hello-world



2- Check the container status

|

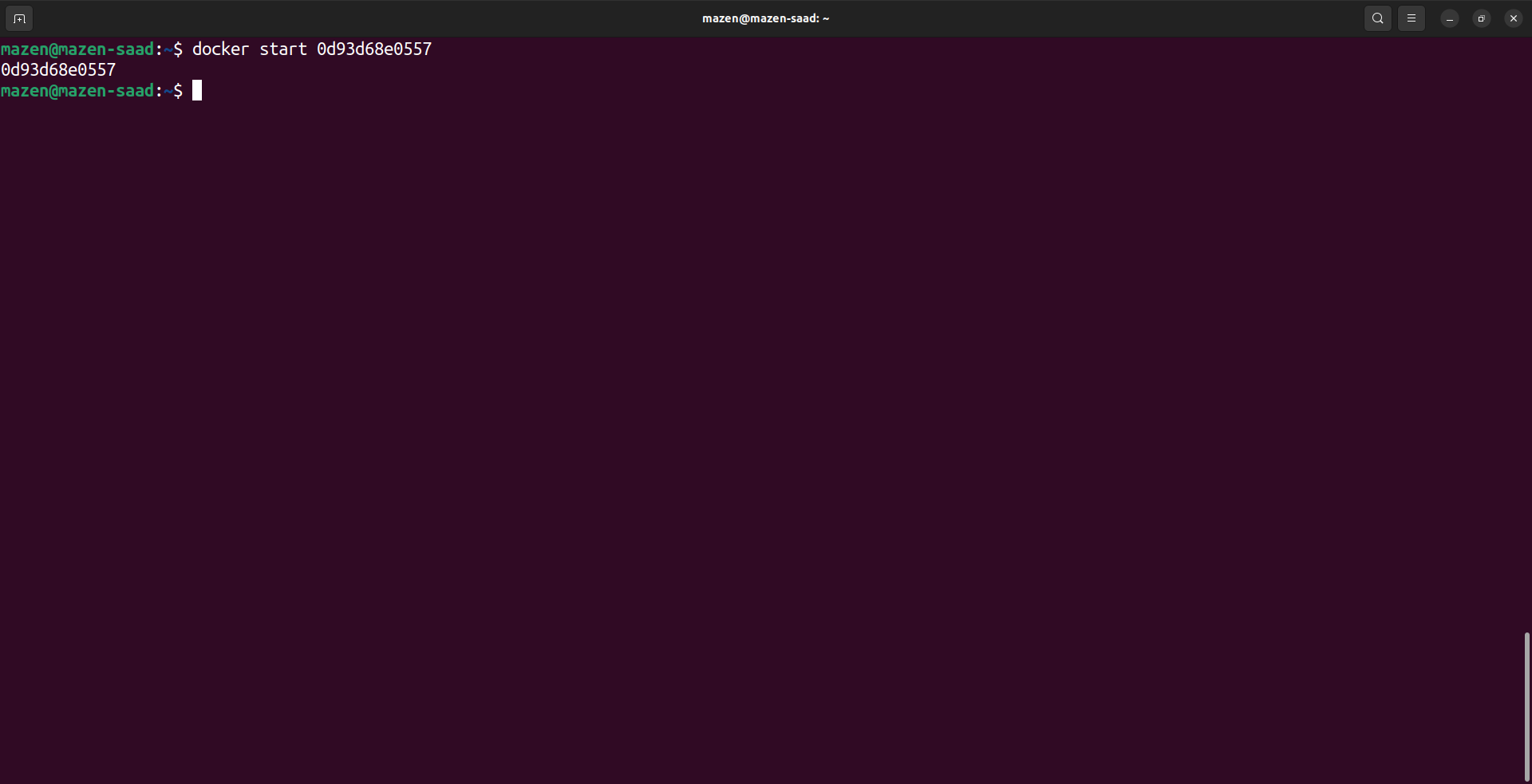
docker ps -a



3- Start the stopped container

|

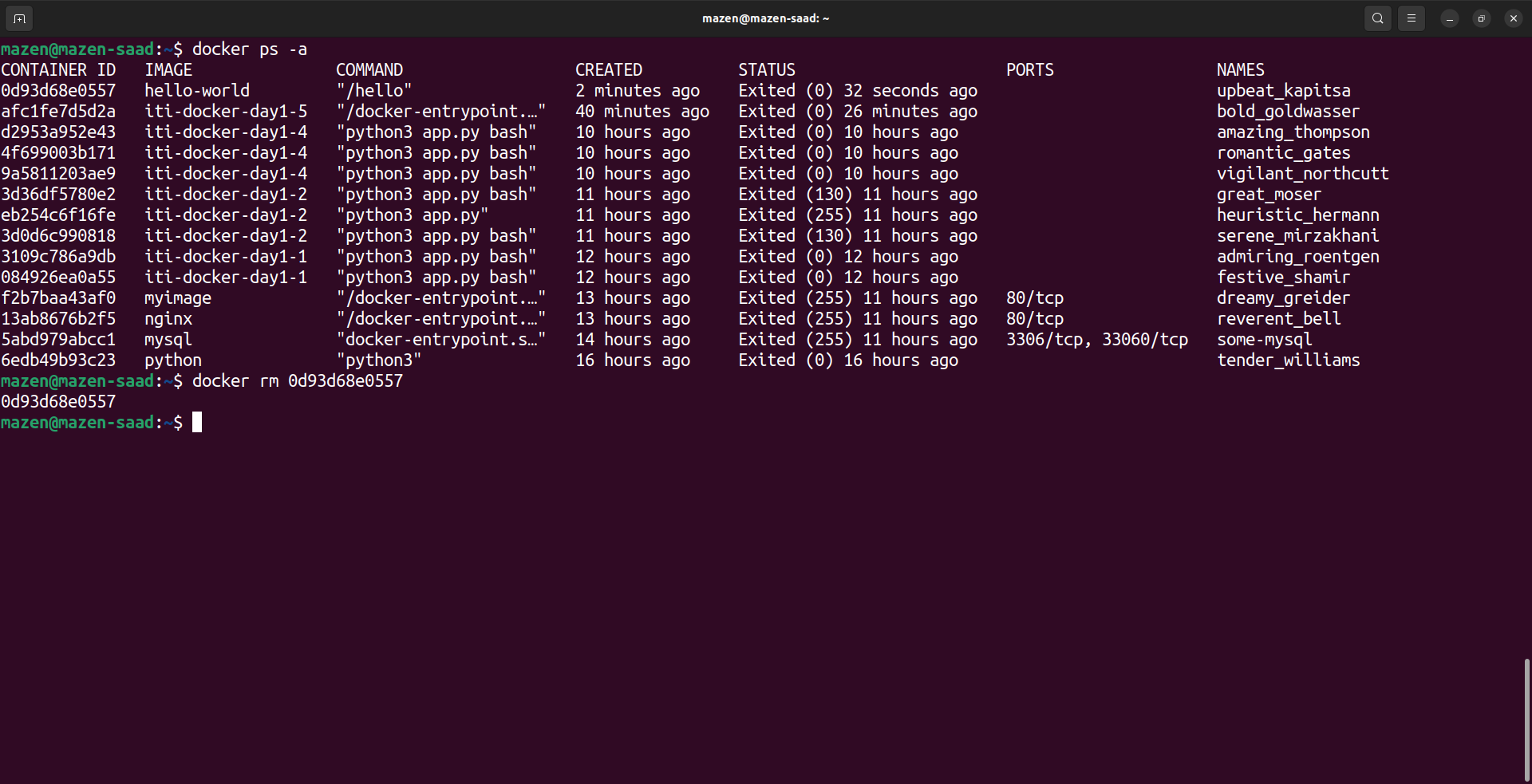
docker start 0d93d68e0557



4- Remove the container

|

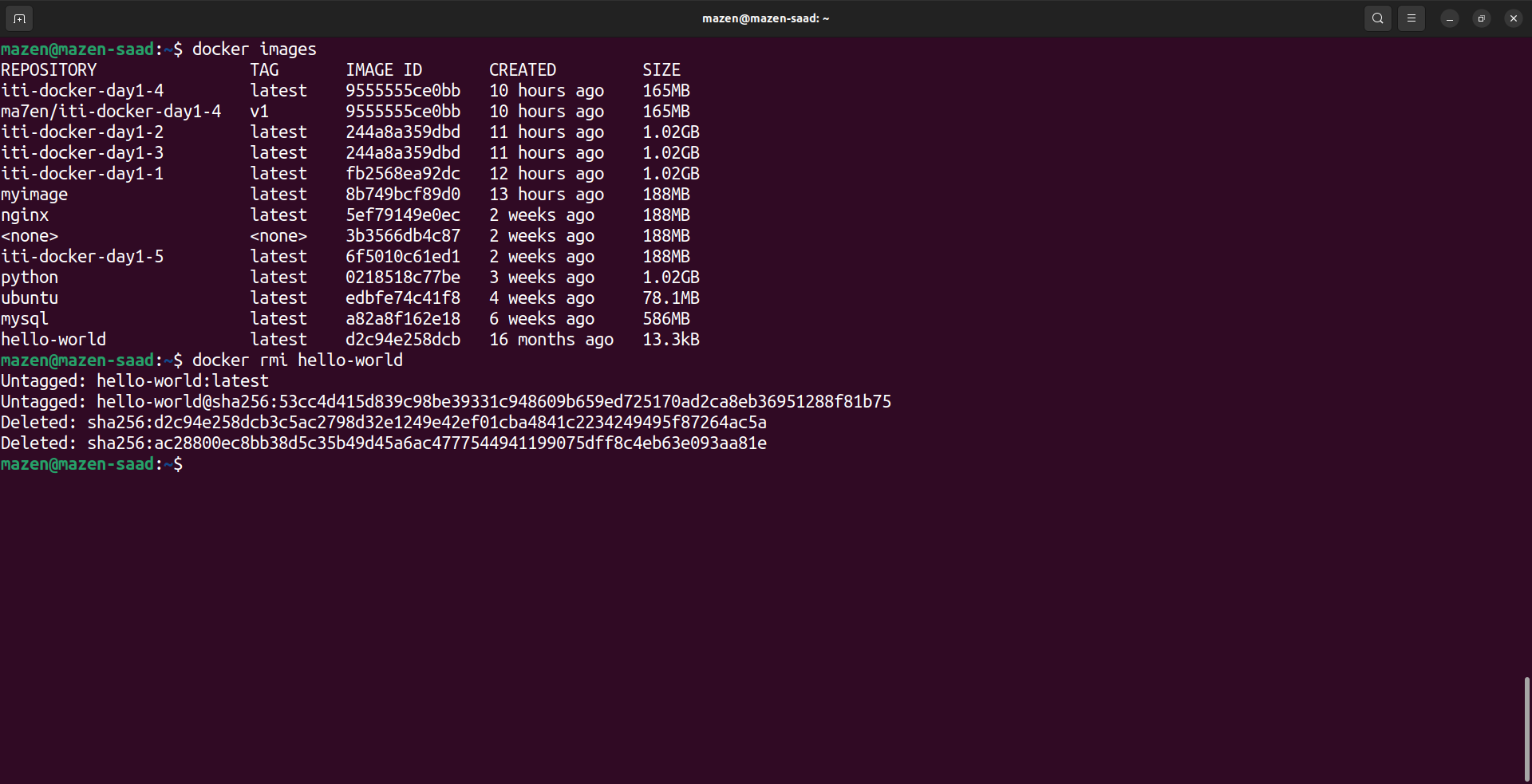
docker rm 0d93d68e0557



5- Remove the image

|

docker rmi hello-world

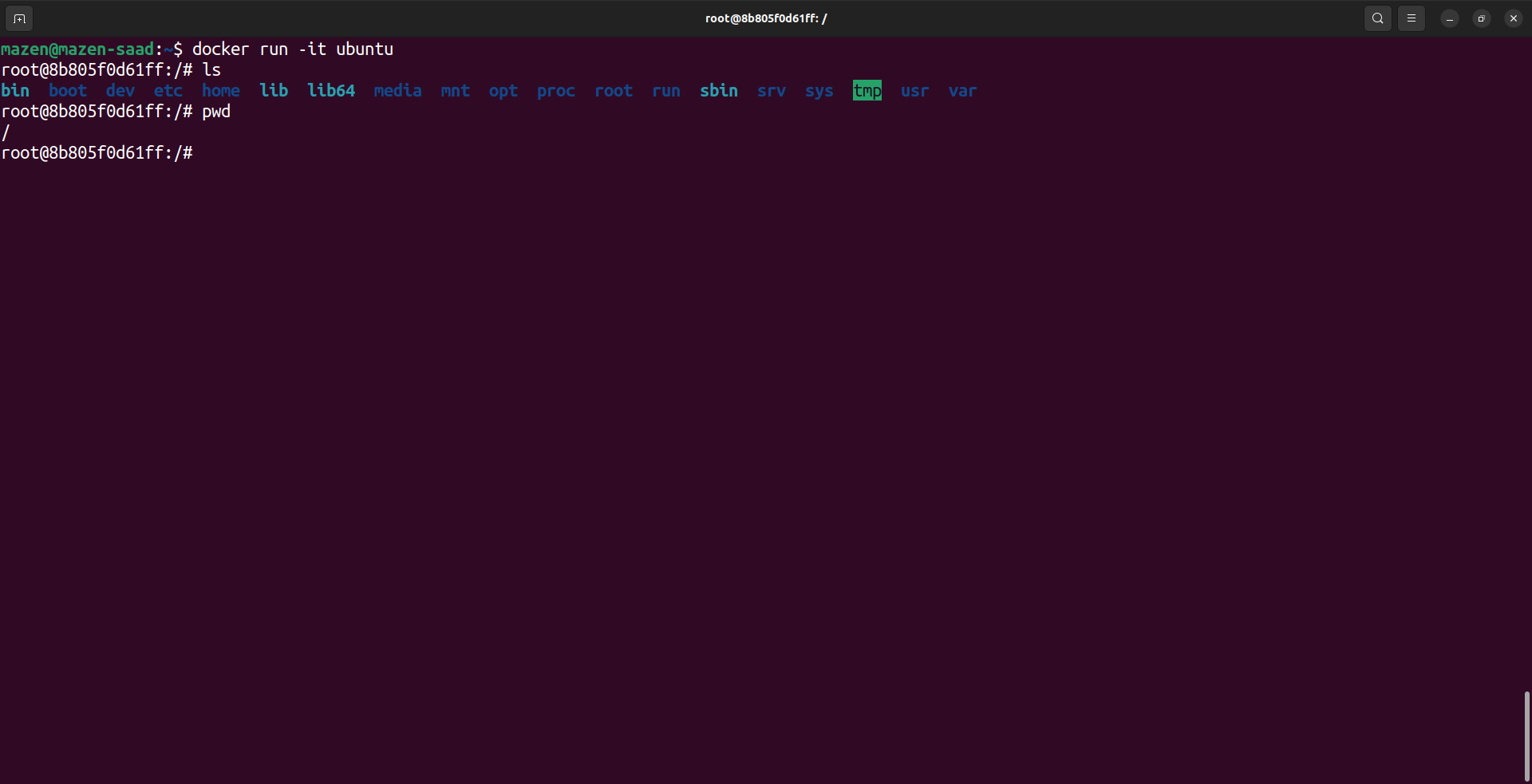


Problem 2:

1- Run container centos or ubuntu in an interactive mode

|

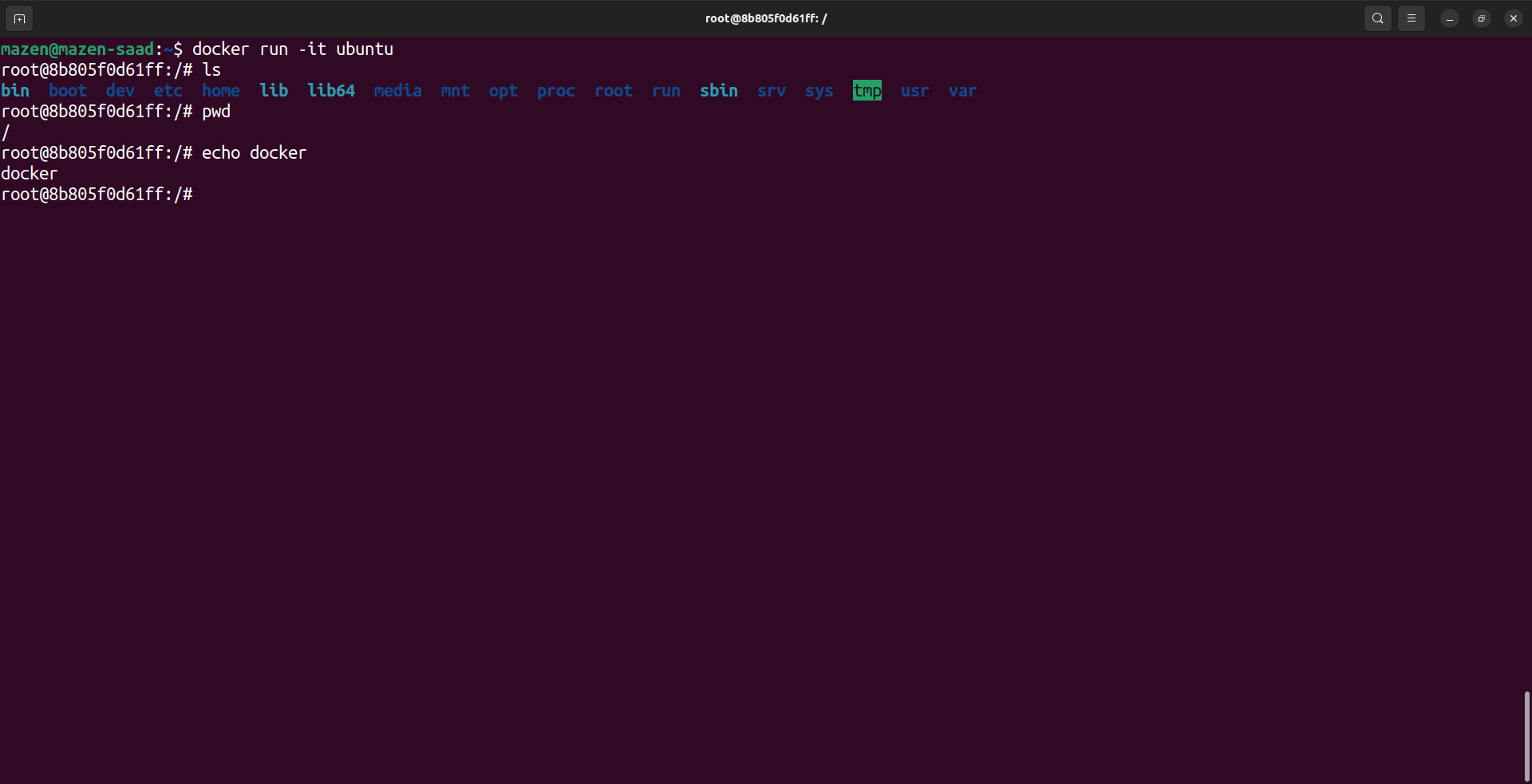
docker run -it ubuntu



2-Run the following command in the container “echo docker ”

|

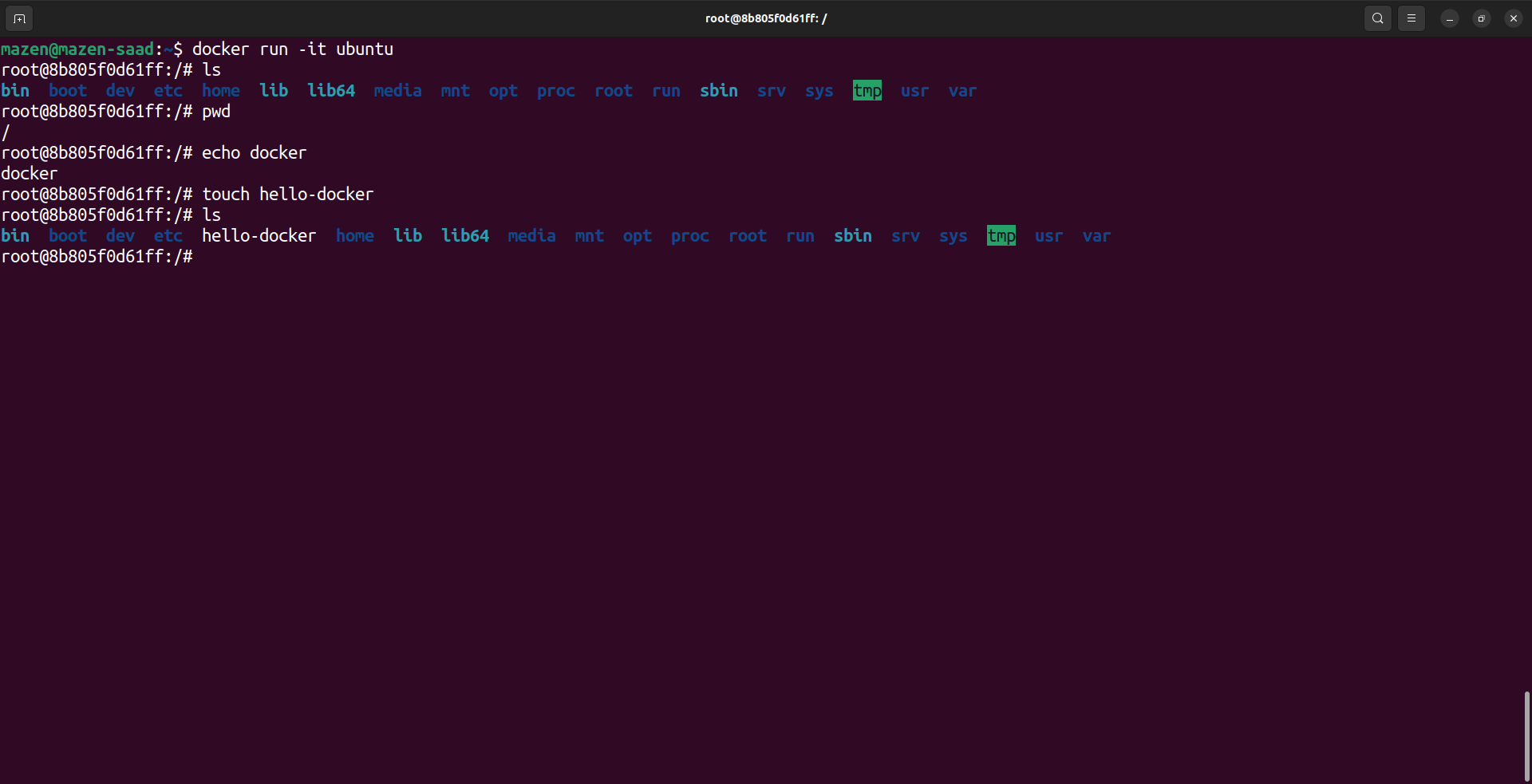
echo docker



3-Open a bash shell in the container and touch a file named hello-docker

|

touch hello-docker



4-Stop the container and remove it. Write your comment about the file hello-docker

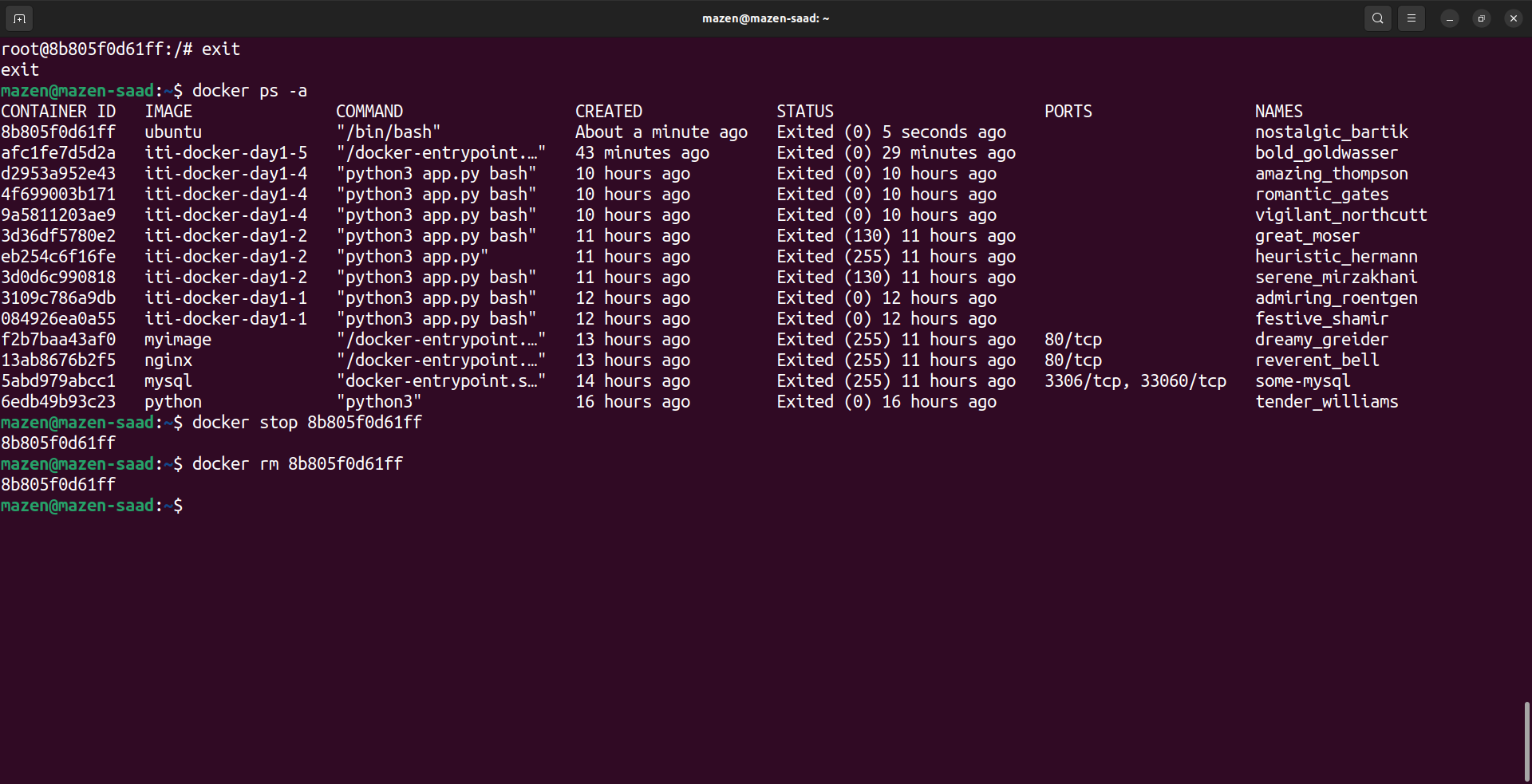
|

exit

docker ps -a

docker stop <CONTAINER\_ID>

docker rm <CONTAINER\_ID>

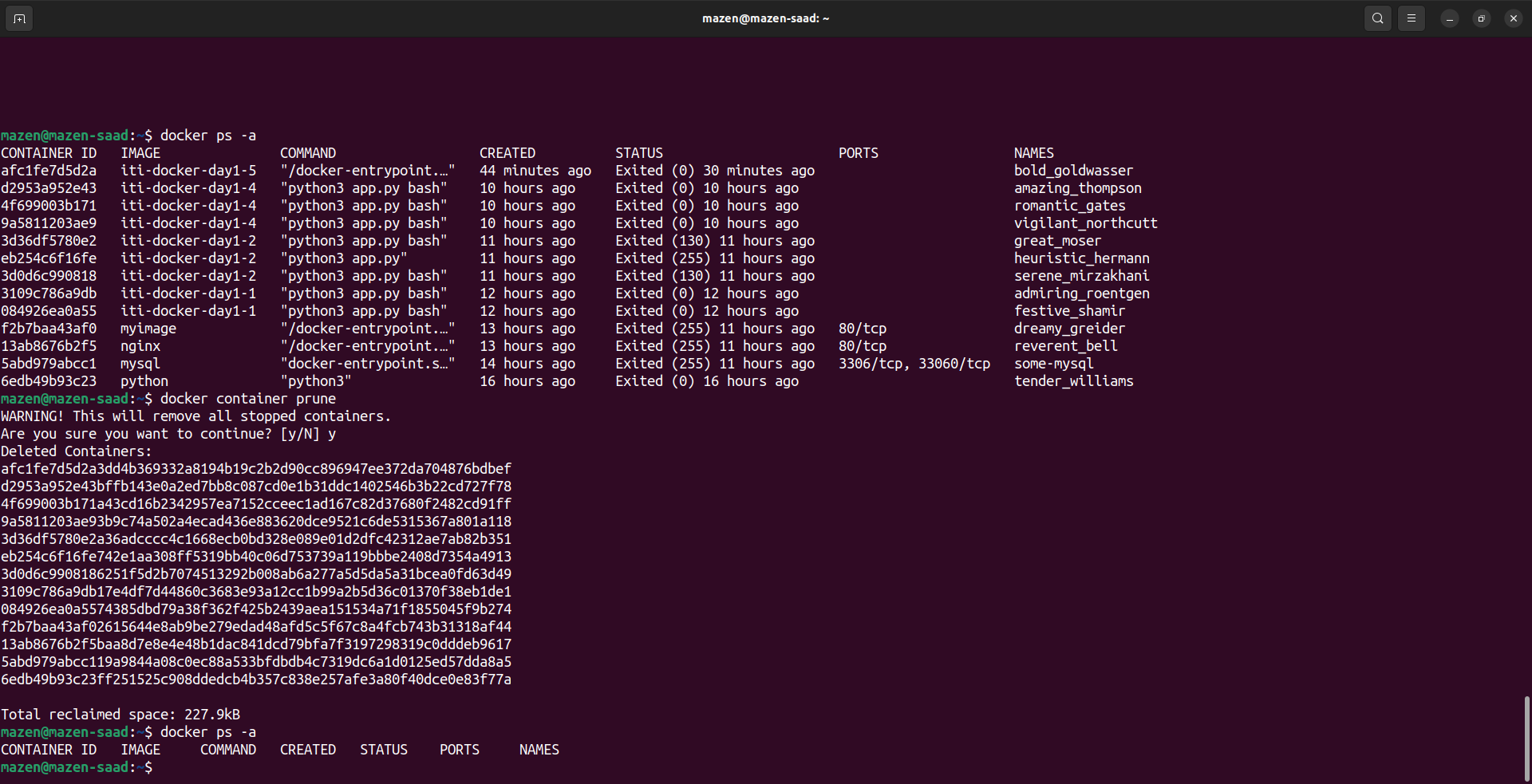


5-Remove all stopped containers

|

docker ps -a

docker container prune



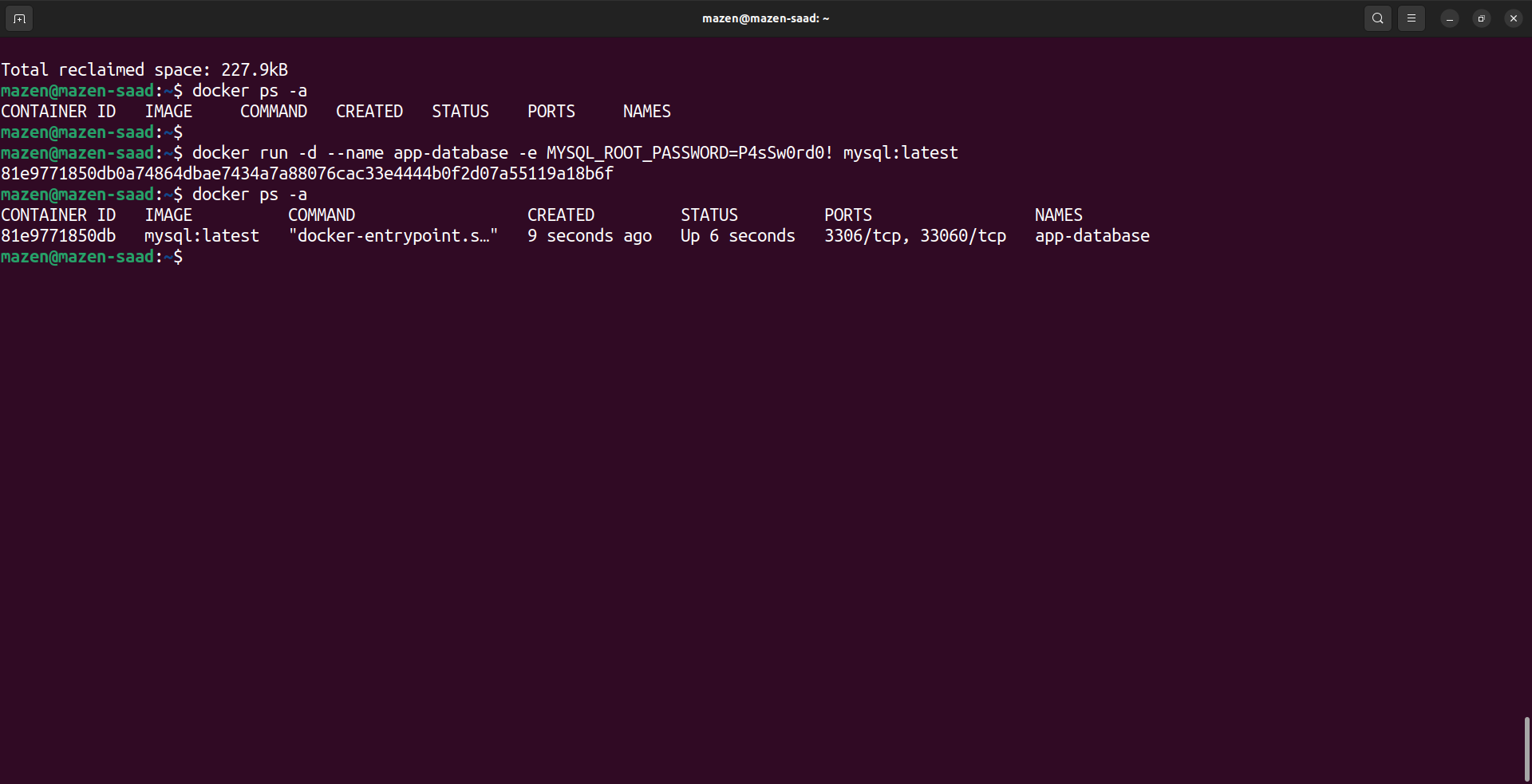
Problem 3:

Deploy a MySQL database called app-database. Use the mysql latest image, and use the -e flag to set MYSQL\_ROOT\_PASSWORD to P4sSw0rd0!. The container should run in the background.

|

docker run -d --name app-database -e MYSQL\_ROOT\_PASSWORD=P4sSw0rd0! mysql:latest

docker ps -a

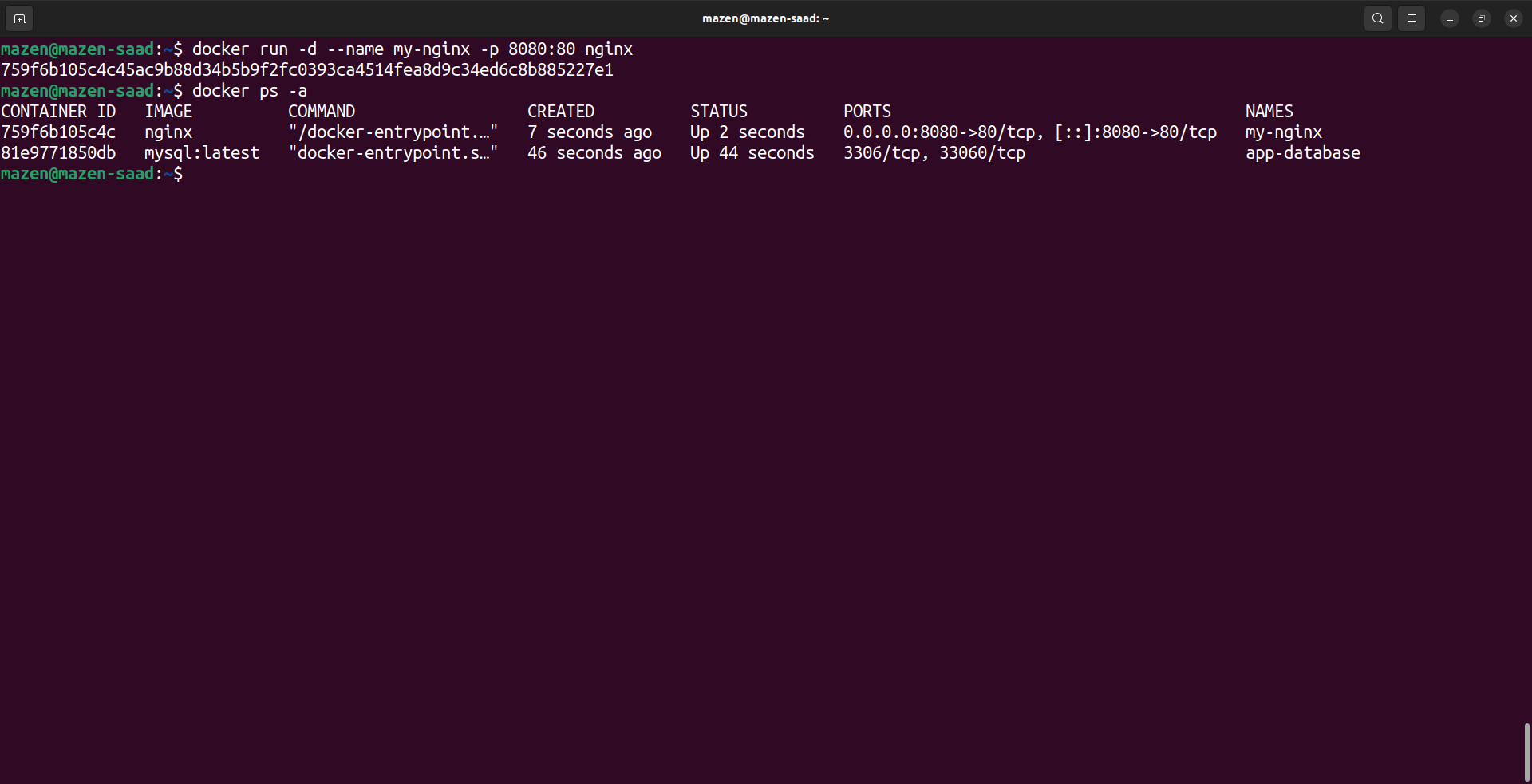


Problem 4:

1- Run the image Nginx

|

docker run -d --name my-nginx -p 8080:80 nginx

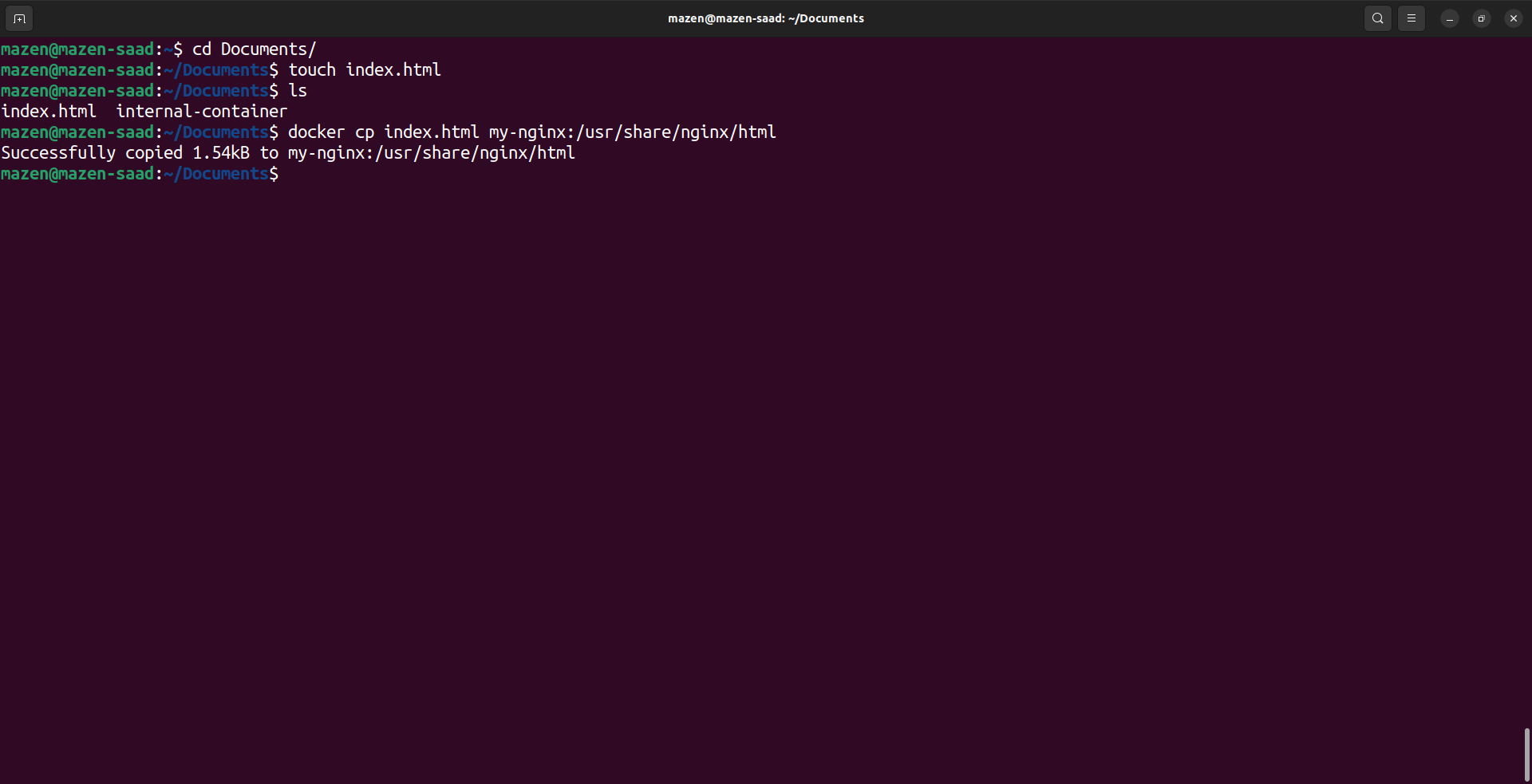


2-Add html static files to the container and make sure they are accessible

|

touch index.html

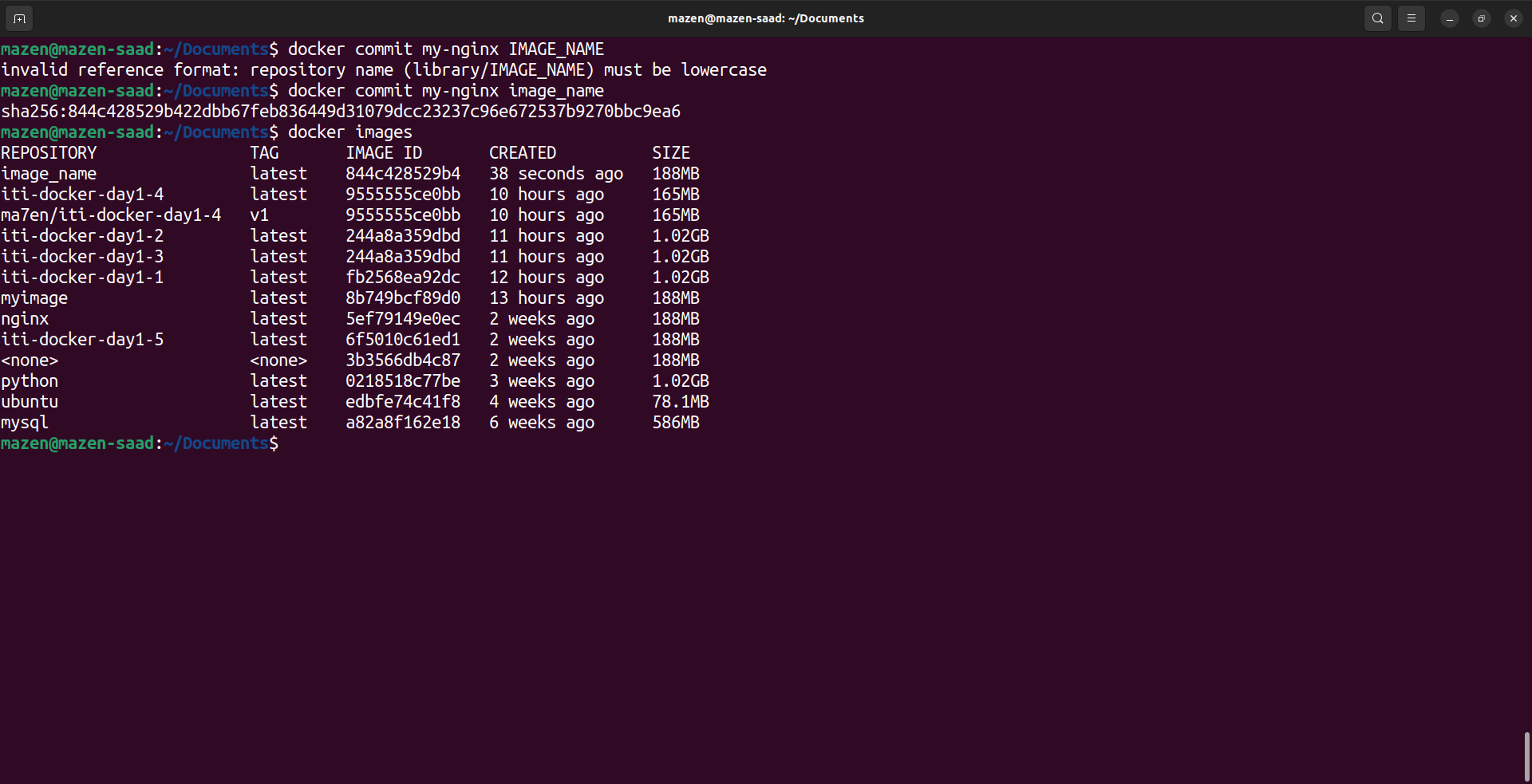
docker cp index.html my-nginx:/usr/share/nginx/html



3-Commit the container with image name IMAGE\_NAME

|

docker commit my-nginx IMAGE\_NAME



Problem 5:

1-Create a python simple app

|

print("Hello, Docker!")

Create a dockerfile to containerize the python app

|

FROM python

COPY app.py .

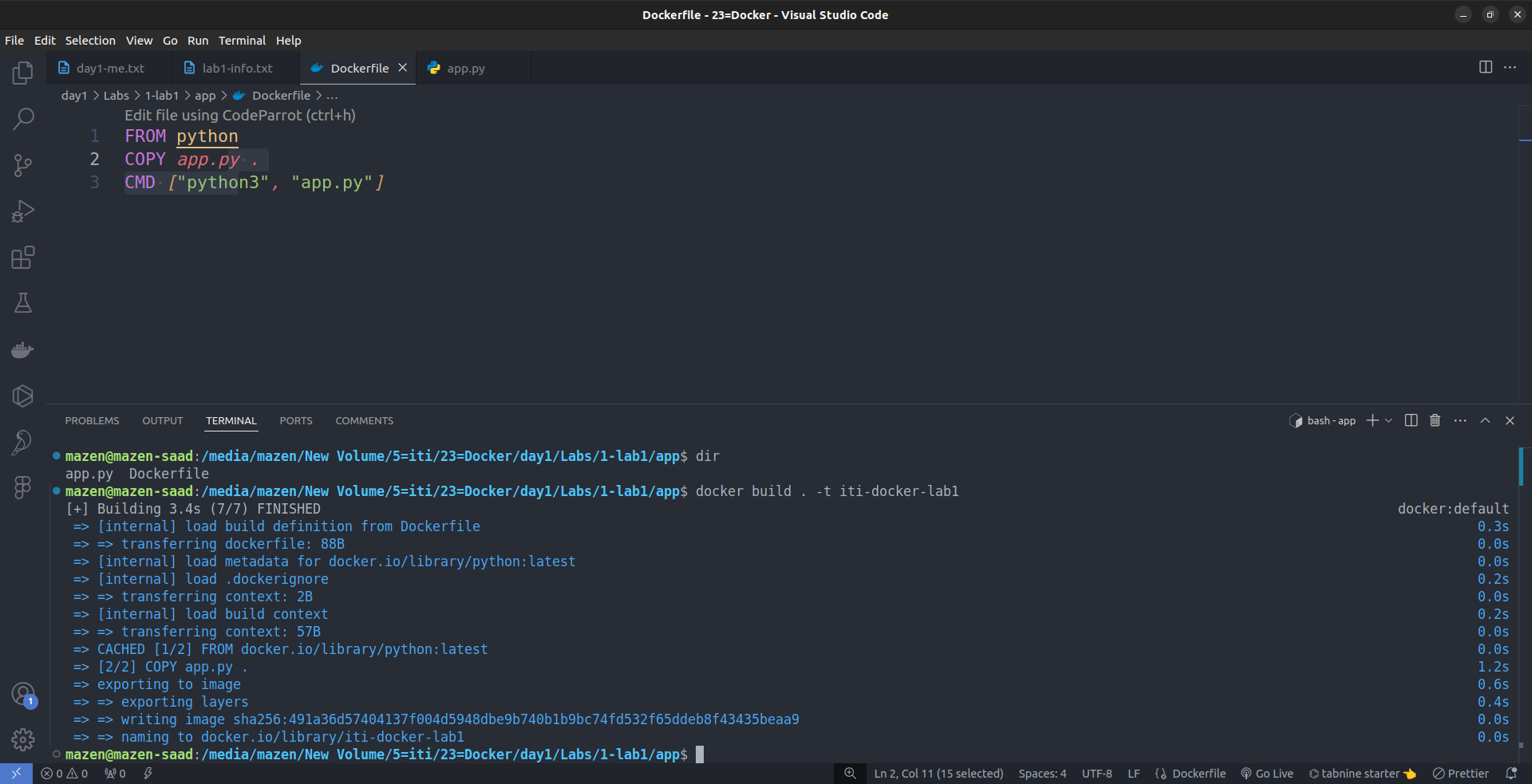
CMD ["python3", "app.py"]

Build the image and test it

|

docker build . -t iti-docker-lab1

docker run -it iti-docker-lab1 bash



5-Push the created image into your docker hub repo

|

docker tag iti-docker-lab1 ma7en/iti-docker-lab1

docker push ma7en/iti-docker-lab1

