**ITI Docker Lab2**

**Mazen AbdelTawab Saad**

Problem 1:

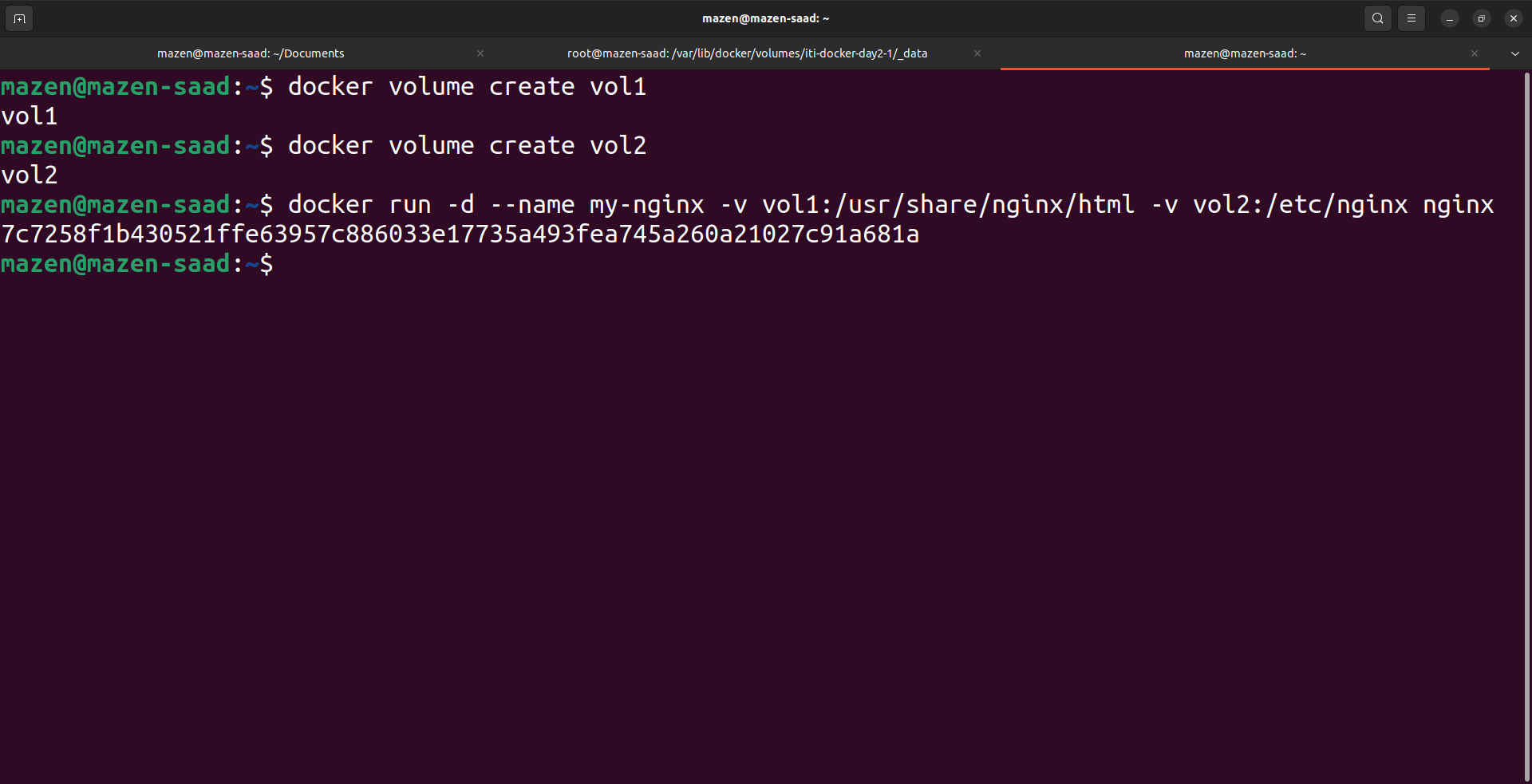
1- Run a container nginx with name my-nginx and attach 2 volumes to the container using volume mount

|

docker volume create vol1

docker volume create vol2

docker run -d --name my-nginx -v vol1:/usr/share/nginx/html -v vol2:/etc/nginx nginx



2- Volume1 for containing static html file

3- Volume2 for containing nginx configuration

4- Edit the html content

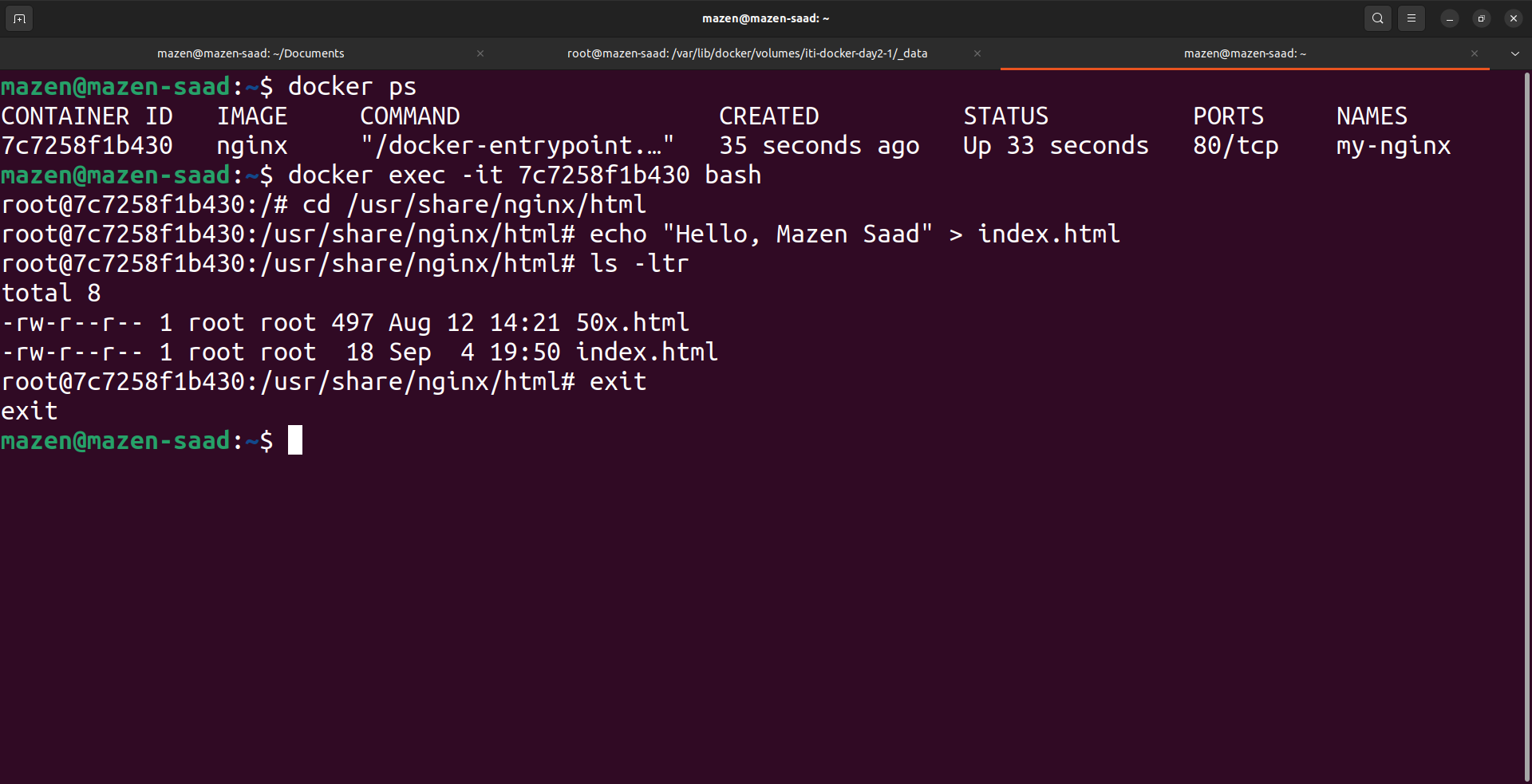
|

docker ps

docker exec -it my-nginx bash

cd /usr/share/nginx/html

echo "Hello, Mazen Saad" > index.html



5- Remove the container

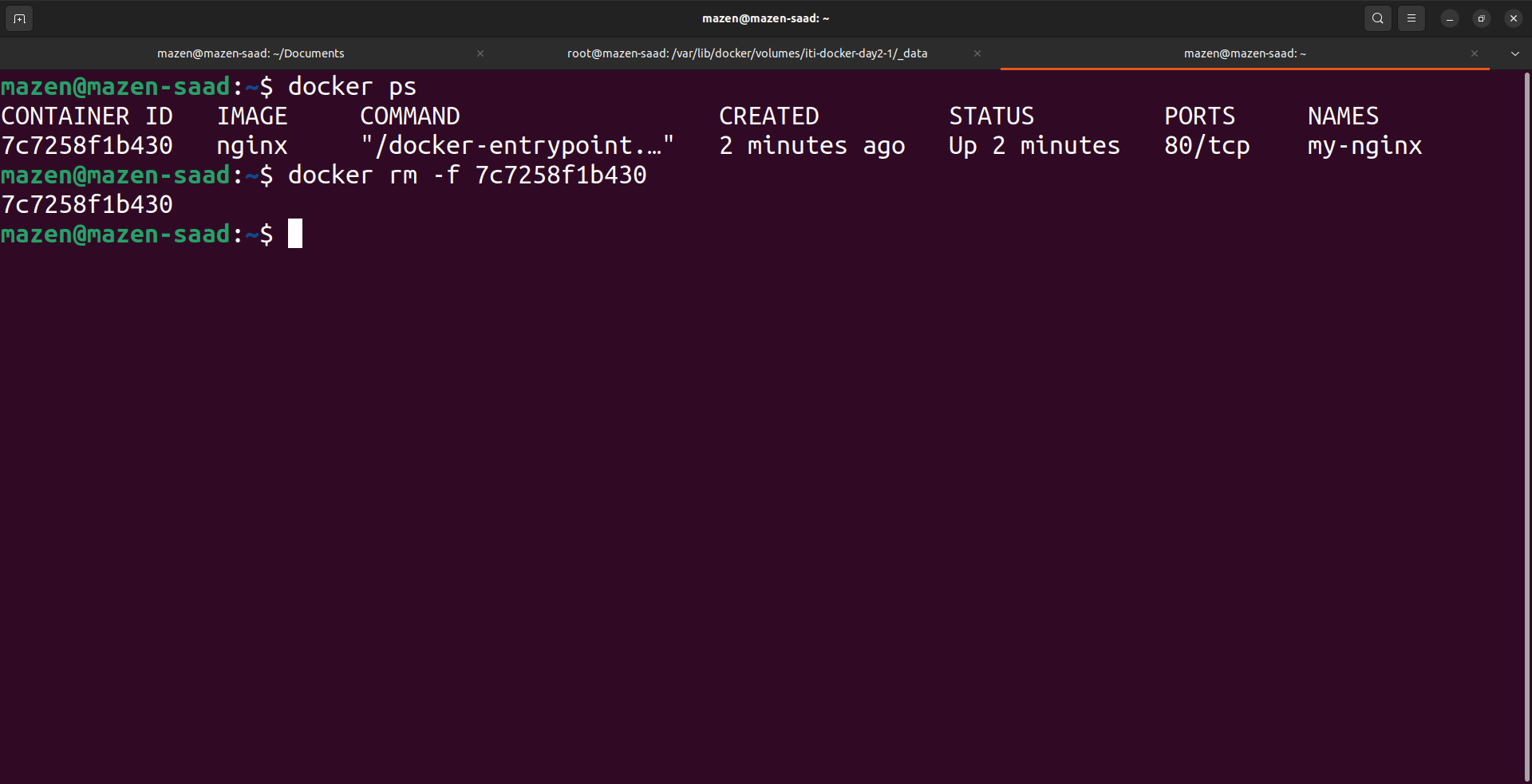
|

docker ps

docker rm -f my-nginx

or

docker rm -f md88j8d3



6- Run a new 2 containers with the following:

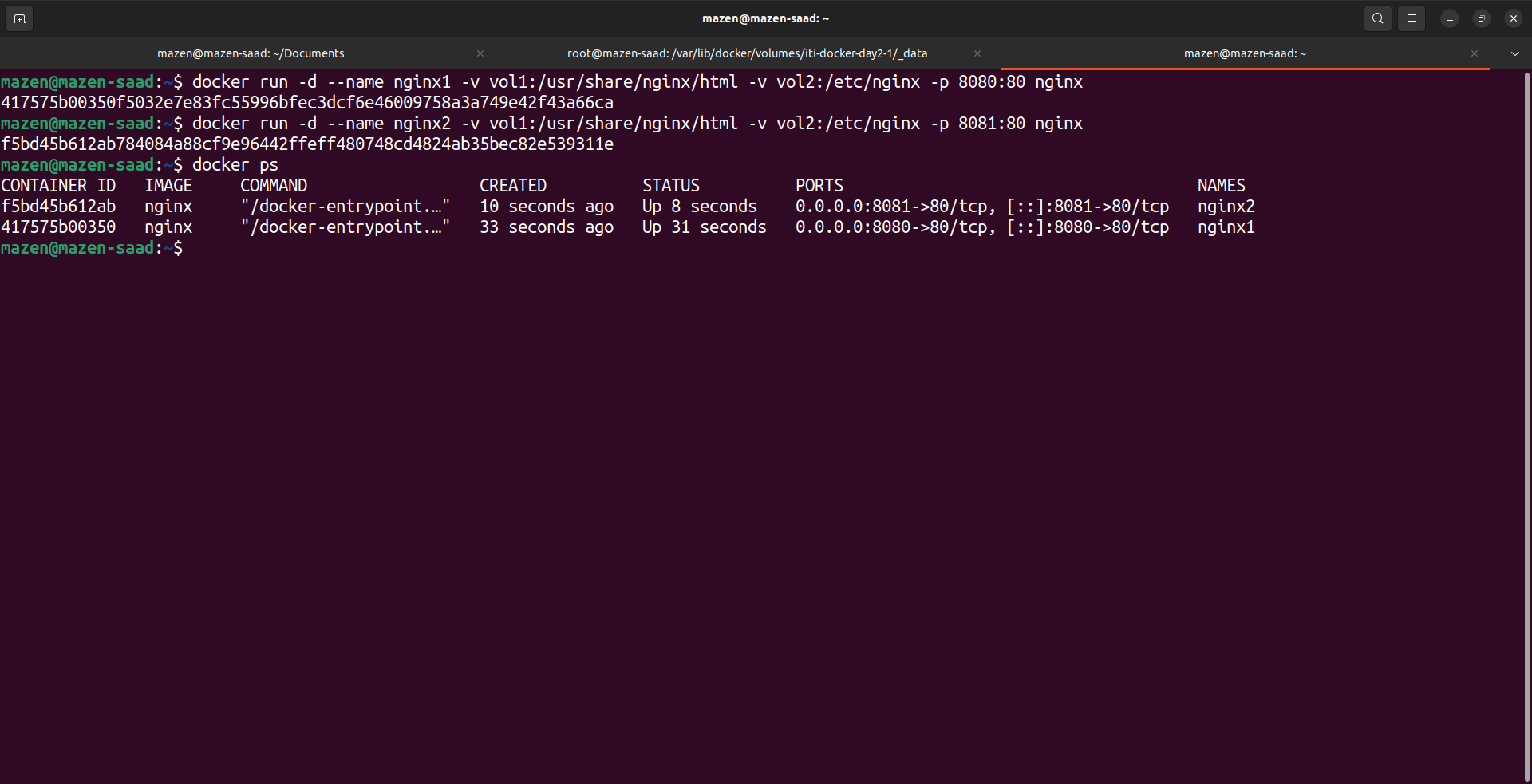
6-1 Attach the two volumes that were attached to the previous container using volume mount

6-2 Map port 80 to port 8080 on you host machine

|

docker run -d --name nginx1 -v vol1:/usr/share/nginx/html -v vol2:/etc/nginx -p 8080:80 nginx

docker run -d --name nginx2 -v vol1:/usr/share/nginx/html -v vol2:/etc/nginx -p 8081:80 nginx



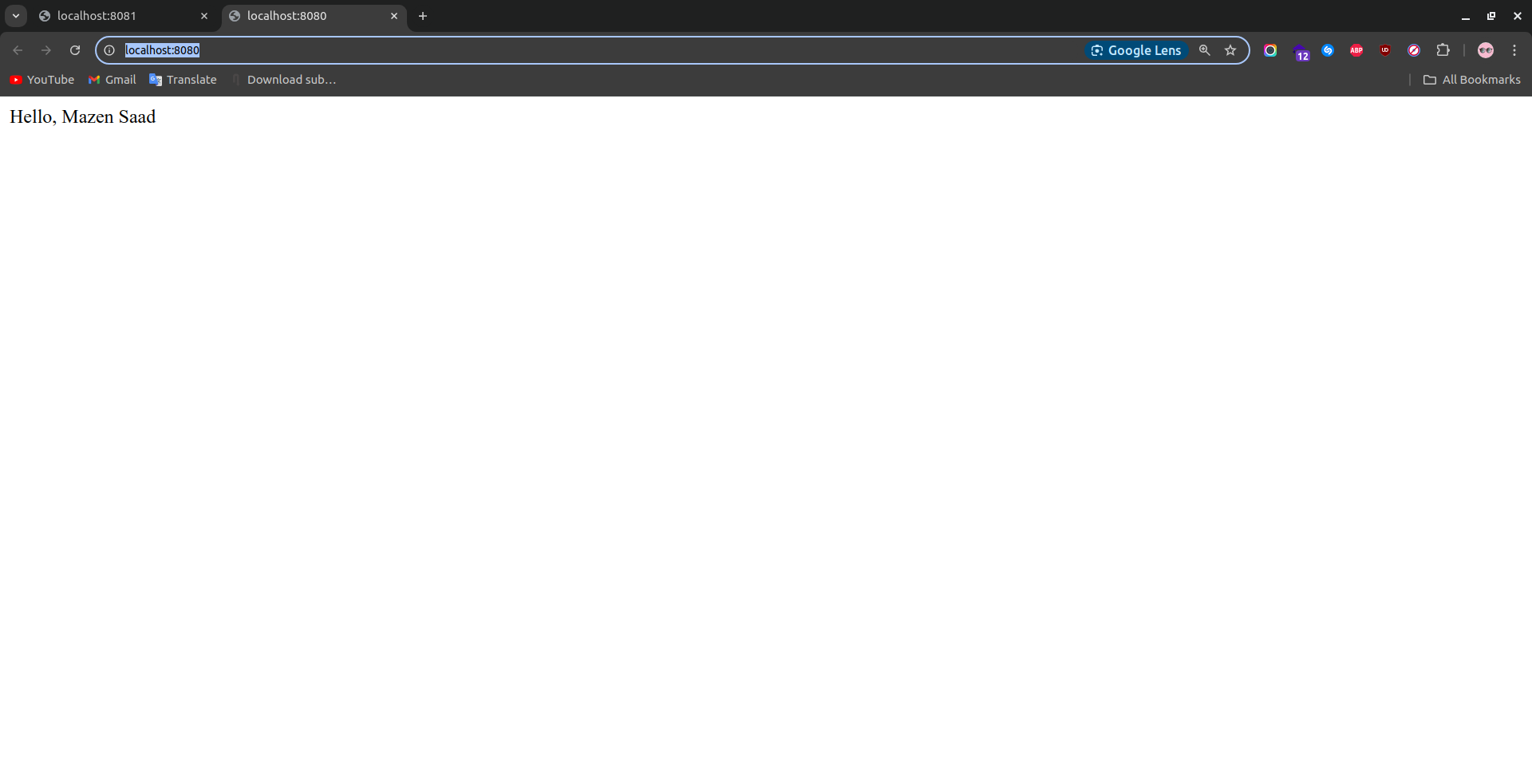
6-3 Access the html files from your browser

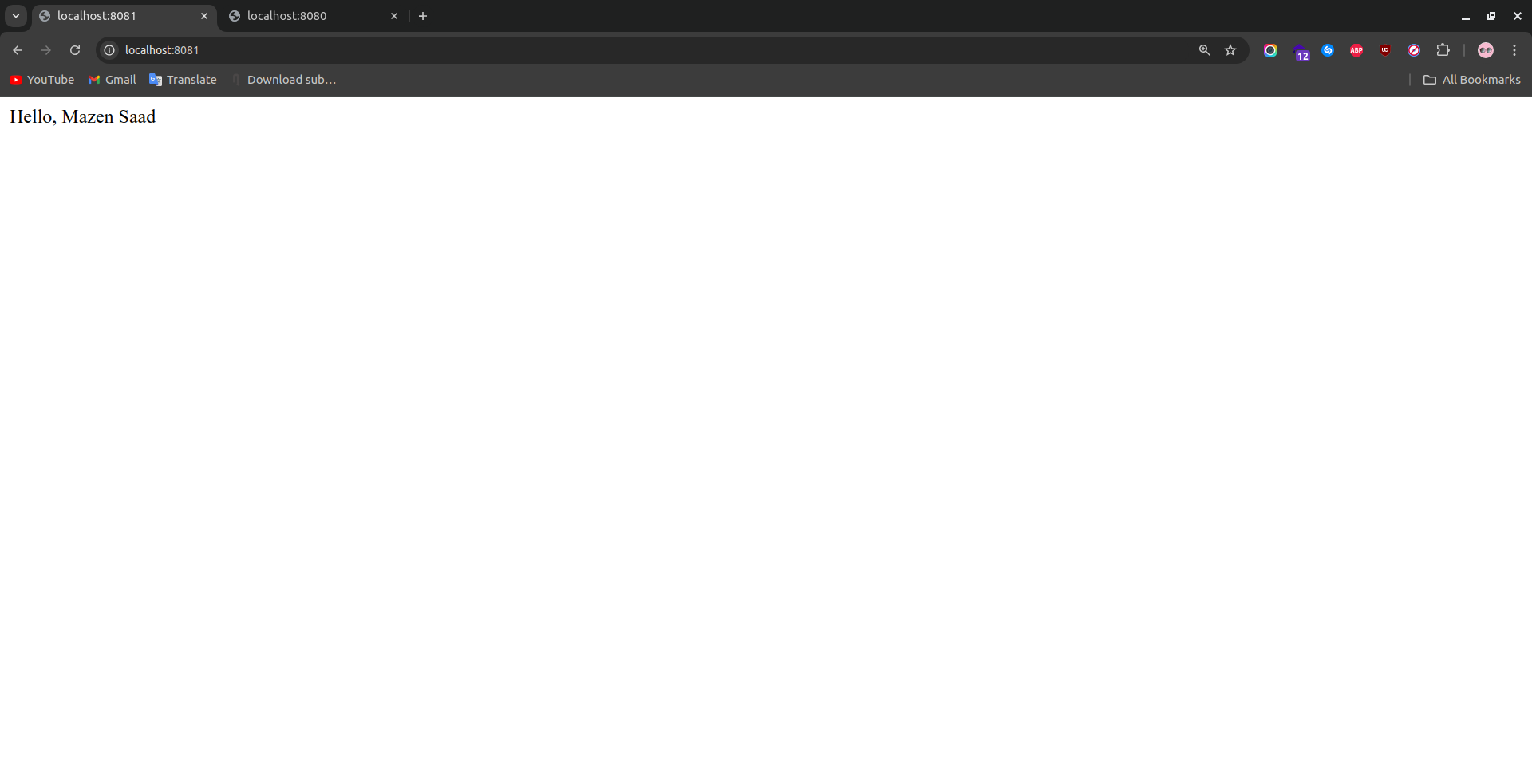
|

# Open a web browser and navigate to

http://localhost:8080

http://localhost:8081



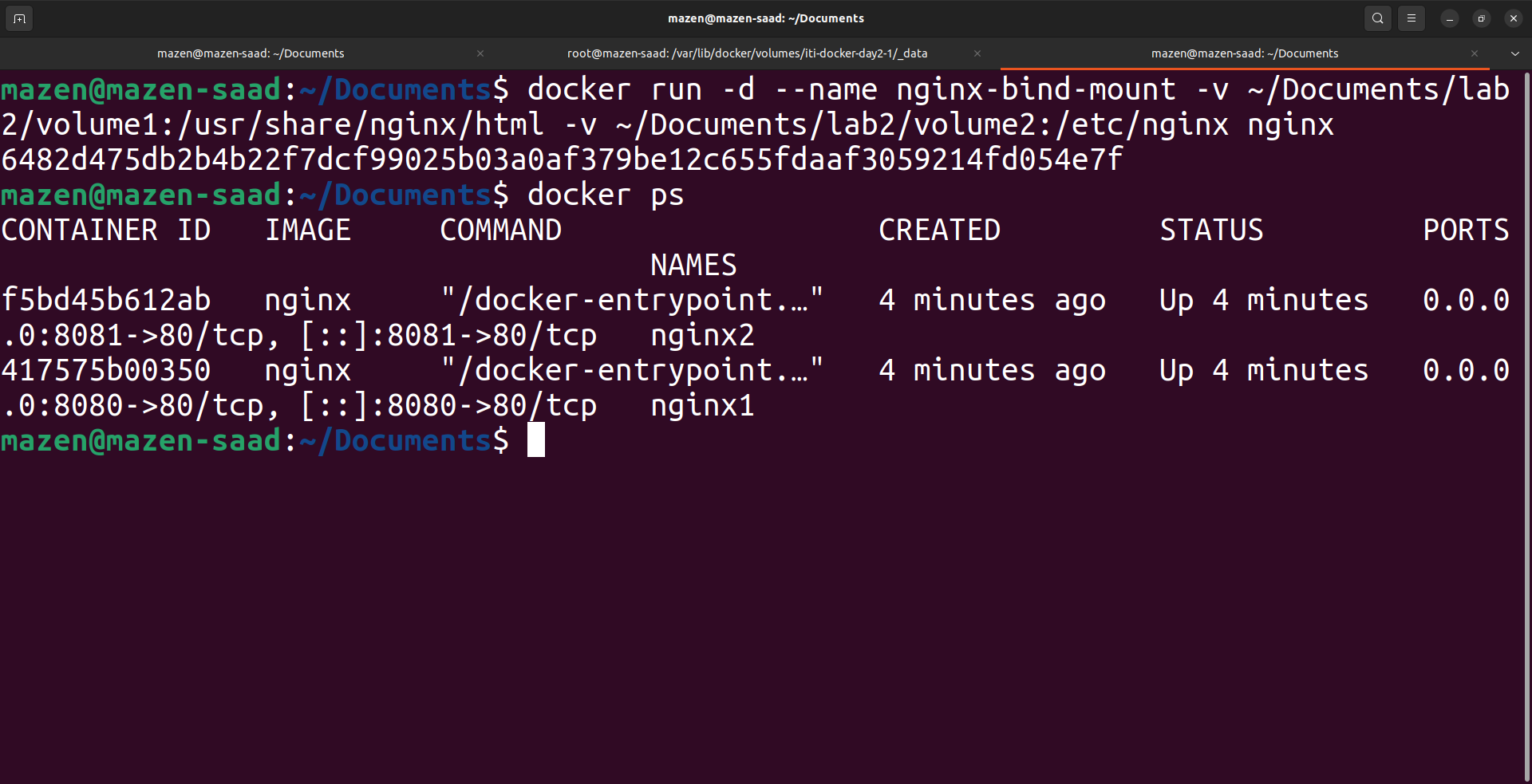


Problem 2:

1- Run a container Nginx with name nginx-bind-mount and attach 2 volumes using bind mount under any paths

|

docker run -d --name nginx-bind-mount -v ~/Documents/lab2/volume1:/usr/share/nginx/html -v ~/Documents/lab2/volume2:/etc/nginx nginx

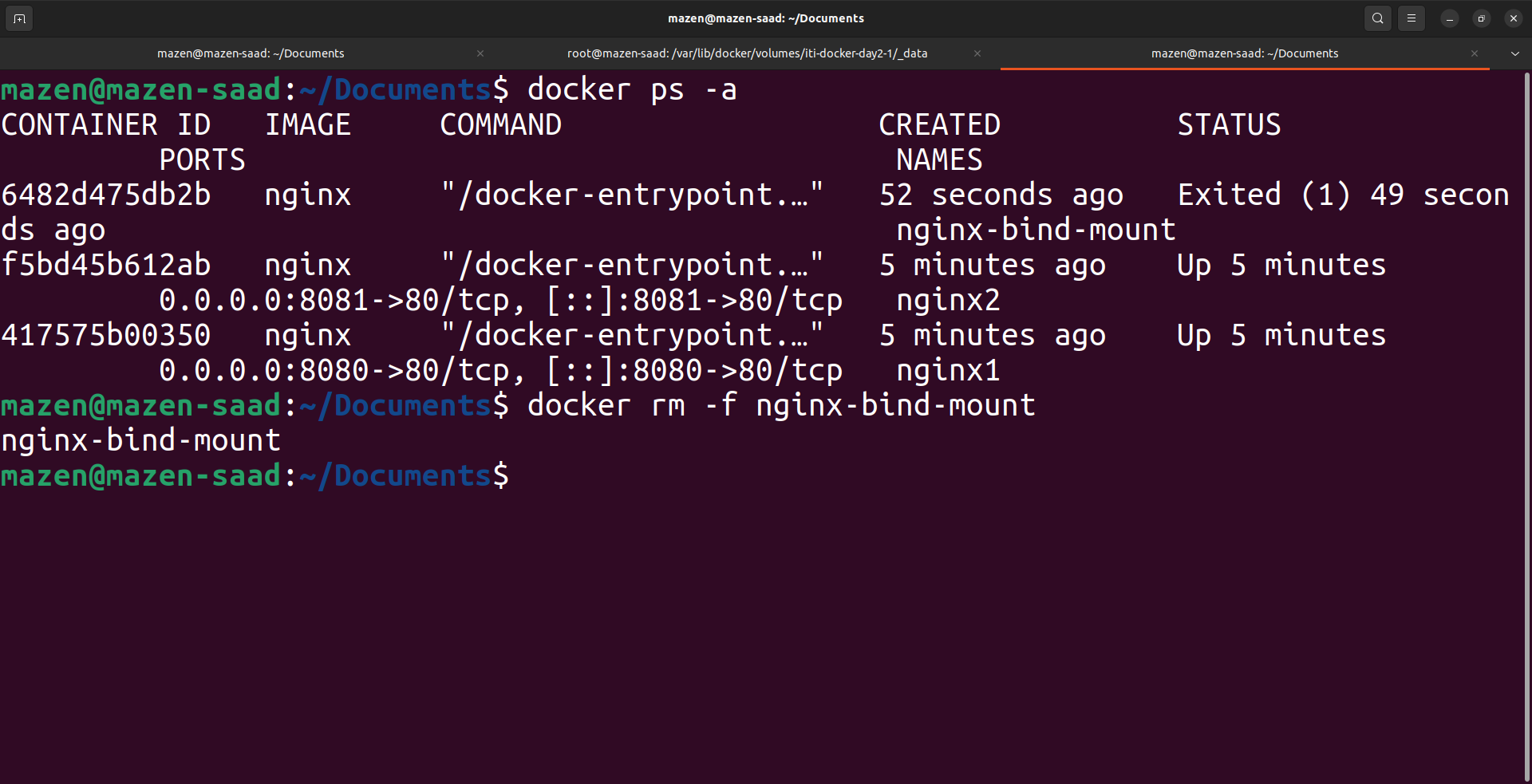


2- Remove the container

|

docker ps -a

docker rm -f nginx-bind-mount



3- Run a new container with the following:

3-1 Attach the two volumes that were attached to the previous container

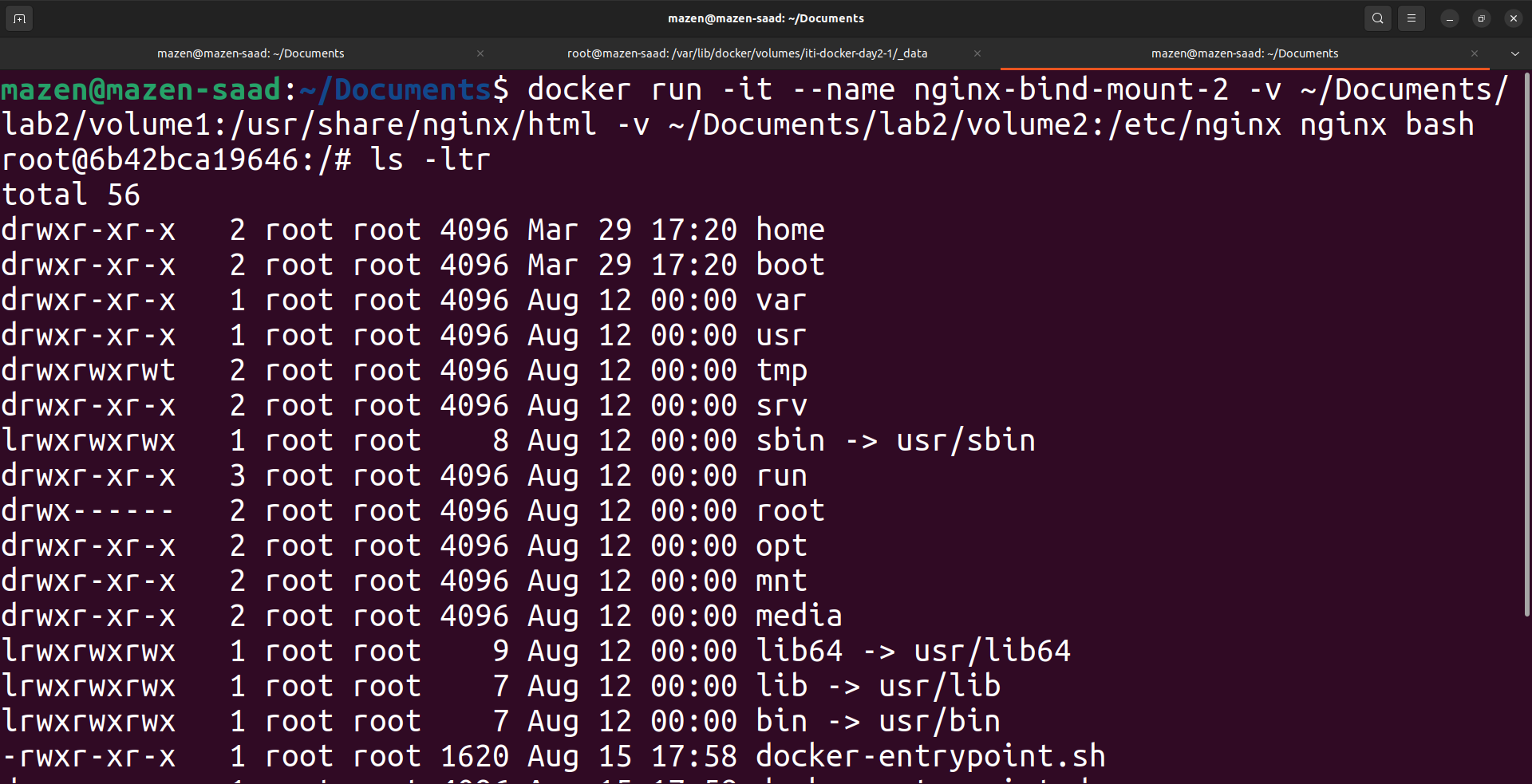
3-2 Check the old data in the new containers

|

docker run -it --name nginx-bind-mount-2 -v ~/Documents/lab2/volume1:/usr/share/nginx/html -v ~/Documents/lab2/volume2:/etc/nginx nginx bash

ls -ltr

exit



Problem 3:

1- Create 2 nginx containers with 2 different network of type bridge, enter to one of them and use curl command to view the content of the other container.

|

docker network create network1

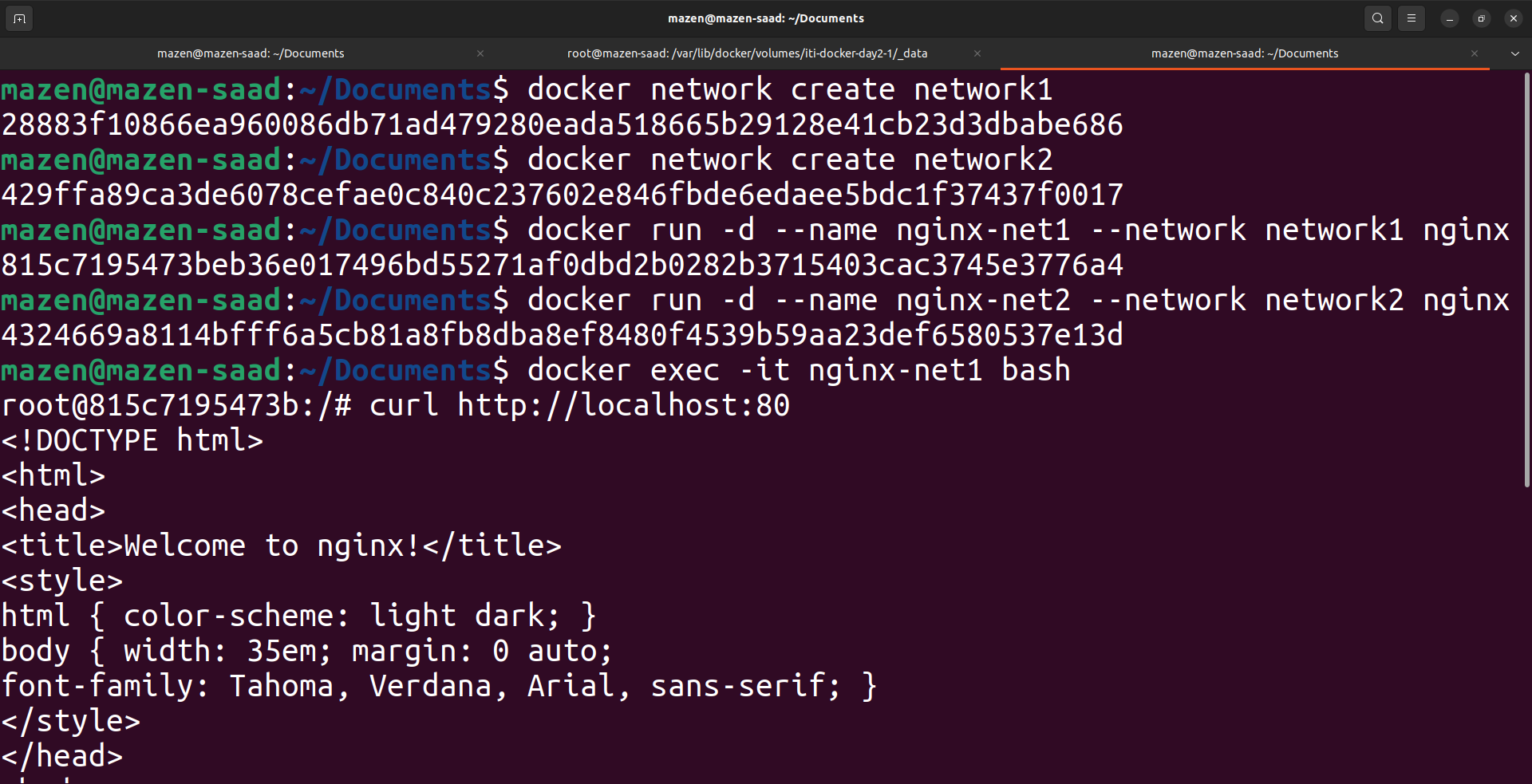
docker network create network2

docker run -d --name nginx-net1 --network network1 nginx

docker run -d --name nginx-net2 --network network2 nginx

docker exec -it nginx-net1 bash

curl http://localhost:80



Problem 4:

1- Create docker compose with: Two services nginx and mysql Add needed ports and environments for both services nginx service is depending on mysql service

|

docker-compose.yaml

services:

mysql:

image: mysql:latest

environment:

MYSQL\_ROOT\_PASSWORD: 123123

MYSQL\_DATABASE: iti\_docker\_lab2

MYSQL\_USER: u\_docker

MYSQL\_PASSWORD: 123123

ports:

- "3306:3306"

nginx:

image: nginx:latest

depends\_on:

- mysql

ports:

- "8080:80"

