**Docker Volumes**

**1. Create docker volumes**

docker volume create postgre-data

2. check directory

sudo ls /var/lib/docker/volumes/postgre-data

3. inspect volume

docker inspect postgre-data

4. Run postgrec on created vlume

docker run -d --name db -p 5432:5432 -e POSTGRES\_PASSWORD=password -v postgre-data:/var/lib/postgresql/data postgres:16-alpine

Lest do some hands on postgrec commands for that we need to run it ui mode

1) docker exec -it db /bin/bash

# connect with postgressql database

psql -U postgres -d postgres

# Create a new table Users

CREATE TABLE IF NOT EXISTS users (

id INTEGER PRIMARY KEY,

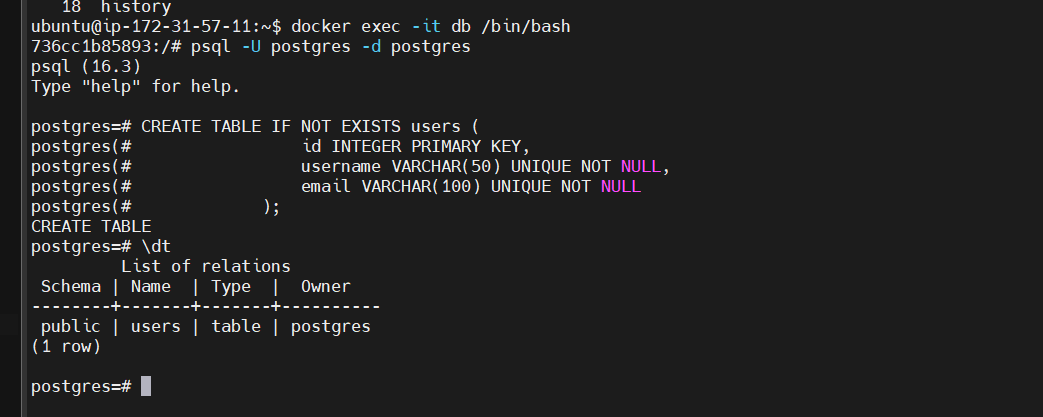
username VARCHAR(50) UNIQUE NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL

);

# List of all tables

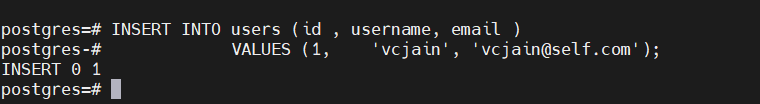
\dt



# Insert a record in users table

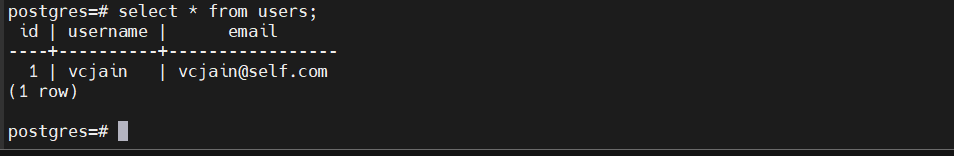
INSERT INTO users (id , username, email )

VALUES (1, 'vcjain', 'vcjain@self.com');



Select the records in the database

select \* from users;



# get exit from post grec

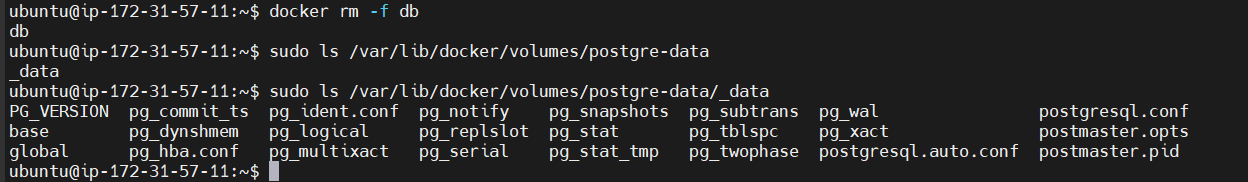


Check our data is still exist or not after removing container so first we will delete container then we will check data is persisted or not

docker rm -f db

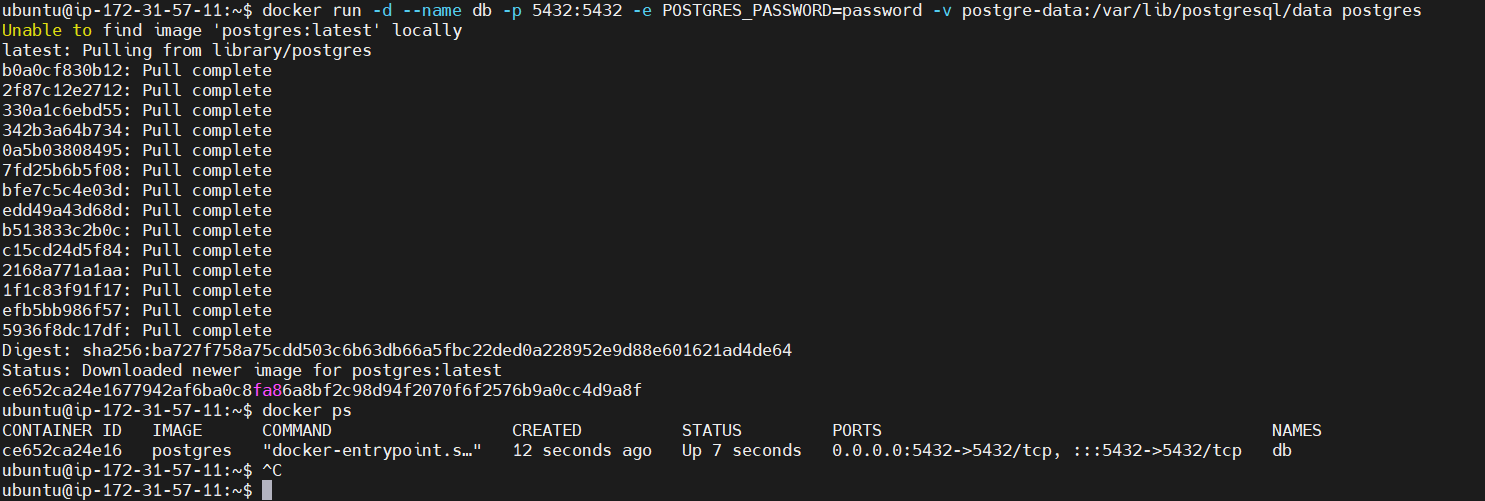
sudo ls /var/lib/docker/volumes/postgre-data/\_data

note data is still there even after we removed container



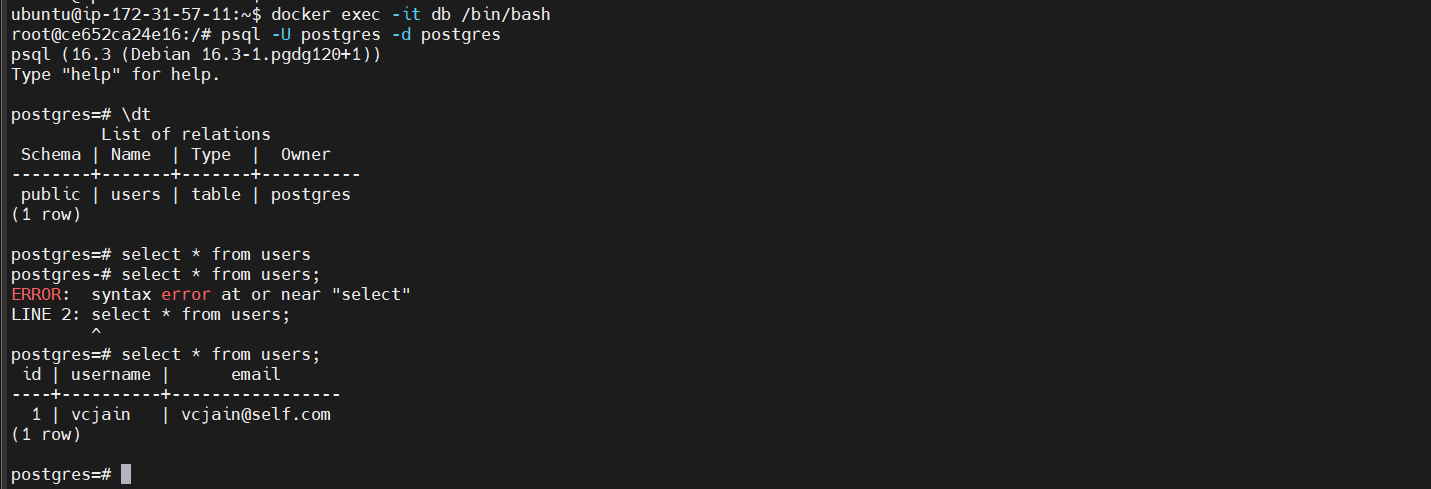
Now lets get into that postgrec and check weather we able to retrieve file that earlier we inserted one record firt we need to install postgress again using following commad

docker run -d --name db -p 5432:5432 -e POSTGRES\_PASSWORD=password -v postgre-data:/var/lib/postgresql/data postgres



Now make connection using following command

psql -U postgres -d postgres



**HOW MOUNT WORK AND INSTALLATION**

Create directory first and insde that created index.html file and put html code

<!DOCTYPE html>

<html>

<head>

<title>Welcome to My Website</title>

</head>

<body>

<h1>Hello, world!</h1>

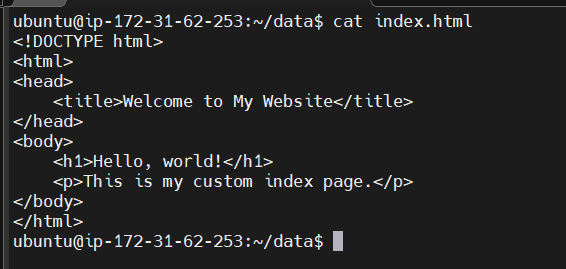
<p>This is my custom index page.</p>

</body>

</html>

mkdir data

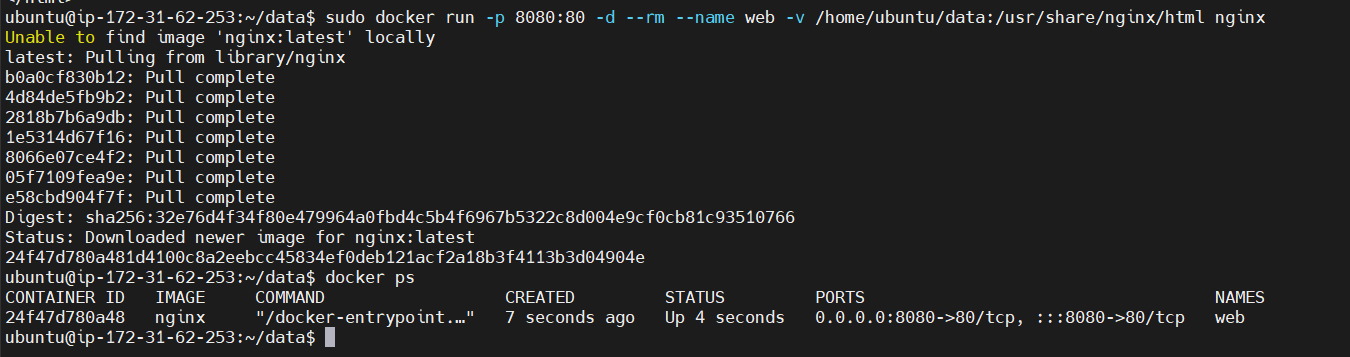
nano index.html



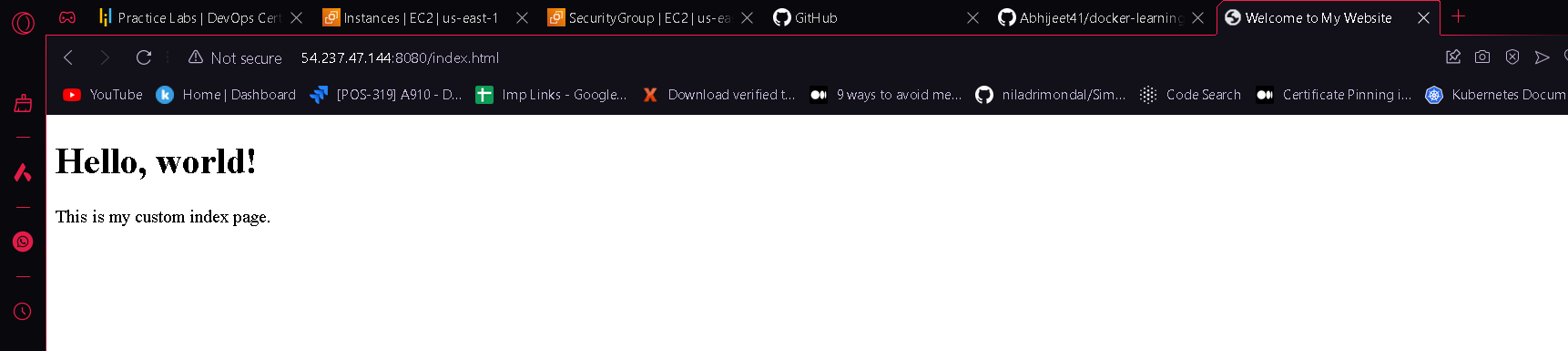
Now run bind mount command instead of vlume name we passing absolute path

Like /home/Ubuntu/data

sudo docker run -p 8080:80 -d --rm --name web -v /home/ubuntu/data:/usr/share/nginx/html nginx

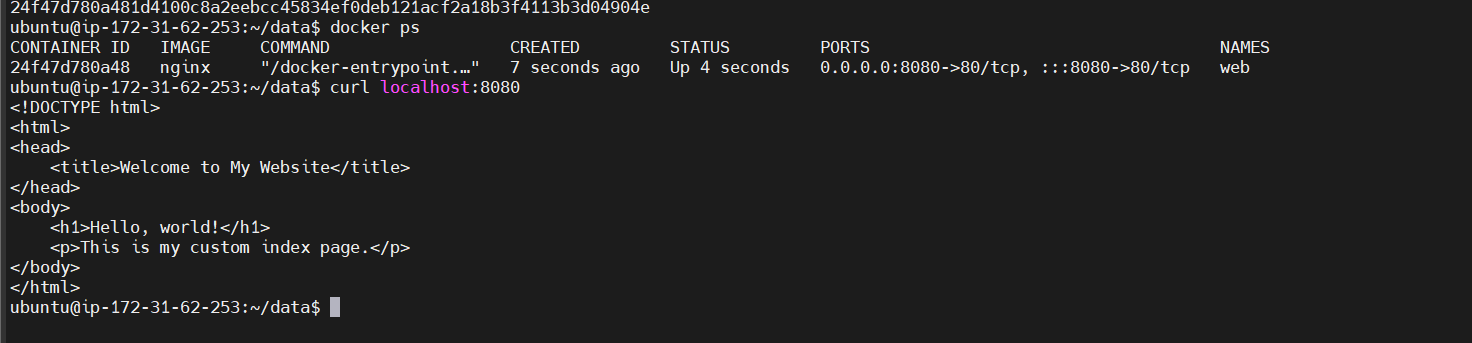


WebBrowser check index.html is hosted



You can also check in moboxterm like this using following command

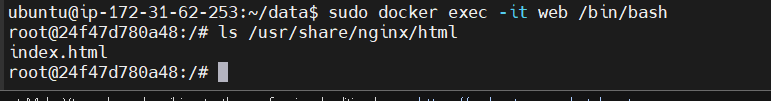
curl localhost:8080



Now get into the nginx container and check there is index.html is present or not

sudo docker exec -it web /bin/bash

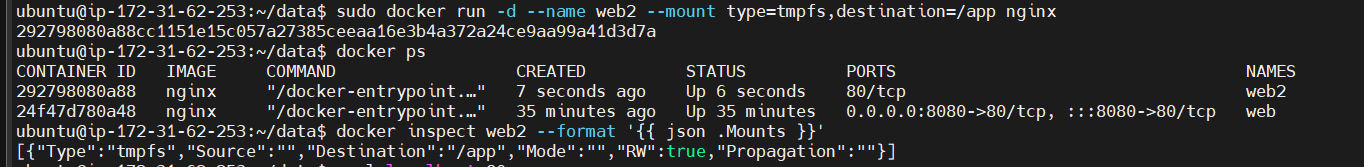
ls /usr/share/nginx/html



**tmpfs mount**

tmpfs mount is used to store the files temporarilt. i.e in memory

its disadvantage is when ever you container gets down the every data stored in memory will gone along with container.



sudo docker run -d --name web --mount type=tmpfs,destination=/app nginx

# Note: There is no source in tmpfs mount type

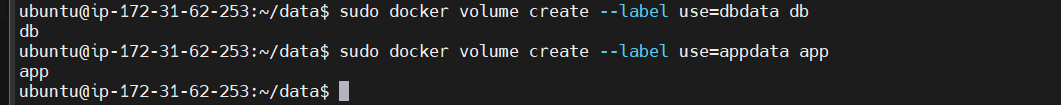
docker inspect web --format '{{ json .Mounts }}'

**Applying Labels and Filters**

Lest first create 2 volumes first is db and second is app

sudo docker volume create --label use=dbdata db

sudo docker volume create --label use=appdata app



**Docker Volume From Udemy**