

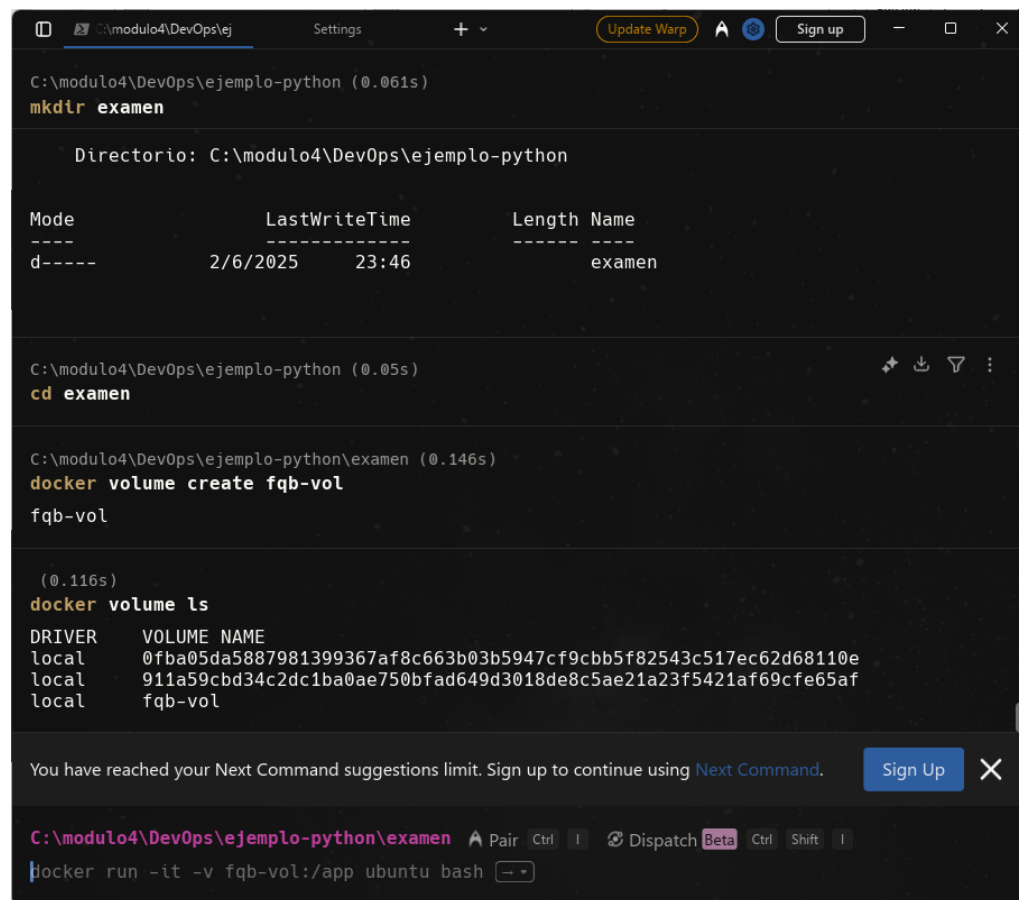
1. Crear un volumen para almacenar la información de la base de datos docker COMANDO  
CREAR NOMBRE-VOLUMEN = (Felicidad Quispe Balcazar => fqb-vol)

En esta pregunta se pide crear un volumen con las iniciales de nuestro nombre como se muestra en la pregunta fqb-vol, abrir el cmd Warp, donde se cre una carpeta con el nombre de examen, luego se ejecuta el siguiente comando:

**docker volume create fqb-vol**

Donde fqb-vol se usará para almacenar la base de datos. Para poder verificar que se a creado correctamente se ejecuta el siguiente comando:

**docker volume ls**



```
C:\modulo4\DevOps\ejemplo-python (0.061s)
mkdir examen

Directorio: C:\modulo4\DevOps\ejemplo-python

Mode                LastWriteTime         Length Name
----                -
d-----           2/6/2025   23:46             examen

C:\modulo4\DevOps\ejemplo-python (0.05s)
cd examen

C:\modulo4\DevOps\ejemplo-python\examen (0.146s)
docker volume create fqb-vol
fqb-vol

(0.116s)
docker volume ls
DRIVER      VOLUME NAME
local      0fba05da5887981399367af8c663b03b5947cf9cbb5f82543c517ec62d68110e
local      911a59cbd34c2dc1ba0ae750bfad649d3018de8c5ae21a23f5421af69cfe65af
local      fqb-vol

You have reached your Next Command suggestions limit. Sign up to continue using Next Command. Sign Up X

C:\modulo4\DevOps\ejemplo-python\examen A Pair Ctrl I Dispatch Beta Ctrl Shift I
docker run -it -v fqb-vol:/app ubuntu bash
```

Como se muestra en la imagen se describe todo lo realizado.

2. Montar la imagen de postgres
  - a. Docker container run .....
    - i. -e POSTGRES\_PASSWORD=PassDocker
    - ii. -v postgres-db:/PATH/DE/LA/BASE/DE/DATOS (esto verifica en dockerhub)

Para montar la imagen de postgres lo primero es verificar se tiene instalado el Docker.desktop para ello se pone el siguiente comando: Docker --version, una vez verificado, se descarga la imagen de postgres con el siguiente comando:

**docker pull postgres**

Esto descargara la imagen desde Docker Hub. Como se muestra en la siguiente imagen:

```
C:\modulo4\DevOps\ejemplo-python\examen (0.116s)
docker volume ls

C:\modulo4\DevOps\ejemplo-python\examen (0.081s)
docker --version
Docker version 28.1.1, build 4eba377

C:\modulo4\DevOps\ejemplo-python\examen (27.013s)
docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
28708ff4e046: Pull complete
82697a7976df: Pull complete
475b0e32b814: Pull complete
e7aba16d6a5e: Pull complete
6ce13d85dabe: Pull complete
2bb588ce4e67: Pull complete
3db9b37be7c3: Pull complete
e9a82aed48d7: Pull complete
bd1fa28722bb: Pull complete
410cd7ec9a40: Pull complete
7c852ebdd63e: Pull complete
89ba8b615fa9: Pull complete
7e11eb1421f3: Pull complete
Digest: sha256:6efd0df010dc3cb40d5e33e3ef84acecc5e73161bd3df06029ee8698e5e12c60
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest

You have reached your Next Command suggestions limit. Sign up to continue using Next Command. Sign Up X

C:\modulo4\DevOps\ejemplo-python\examen A Pair Ctrl | Dispatch Beta Ctrl Shift |
docker run --name postgres-db -e POSTGRES_PASSWORD=password -d -v fq-b-vol:/var/lib/postgresql/data postgres
```

Ahora se ejecuta y se crea el contenedor con el vol que se creo con el siguiente comando:

```
docker run -d --name postgres-db -e POSTGRES_PASSWORD=PassDocker -v fq-b-vol:/var/lib/postgresql/data -p 5432:5432 postgres
```

Para verificar si el contenedor esta funcionando se ejecuta el siguiente comando:

```
docker ps
```

Como se muestran en la imagen:

```
C:\modulo4\DevOps\ejemplo-python\examen (0.116s)
docker volume ls

C:\modulo4\DevOps\ejemplo-python\examen (0.081s)
docker --version
Docker version 28.1.1, build 4eba377

C:\modulo4\DevOps\ejemplo-python\examen (27.013s)
docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
28708ff4e046: Pull complete
82697a7976df: Pull complete
475b0e32b814: Pull complete
e7aba16d6a5e: Pull complete
6ce13d85dabe: Pull complete
2bb588ce4e67: Pull complete
3db9b37be7c3: Pull complete
e9a82aed48d7: Pull complete
bd1fa28722bb: Pull complete
410cd7ec9a40: Pull complete
7c852ebdd63e: Pull complete
89ba8b615fa9: Pull complete
7e11eb1421f3: Pull complete
Digest: sha256:6efd0df010dc3cb40d5e33e3ef84acecc5e73161bd3df06029ee8698e5e12c60
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest

C:\modulo4\DevOps\ejemplo-python\examen (0.543s)
docker run -d --name postgres-db -e POSTGRES_PASSWORD=PassDocker -v fq-b-vol:/var/lib/postgresql/data -p 5432:5432 postgres
f52679faeb9325388db771af1e98aebf23811364ab7a6f62e4bd7872243e

C:\modulo4\DevOps\ejemplo-python\examen (0.003s)
docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
f52679faeb93   postgres  docker-entrypoint.sh     2 minutes ago Up 2 minutes   0.0.0.0:5432->5432/tcp   postgres-db

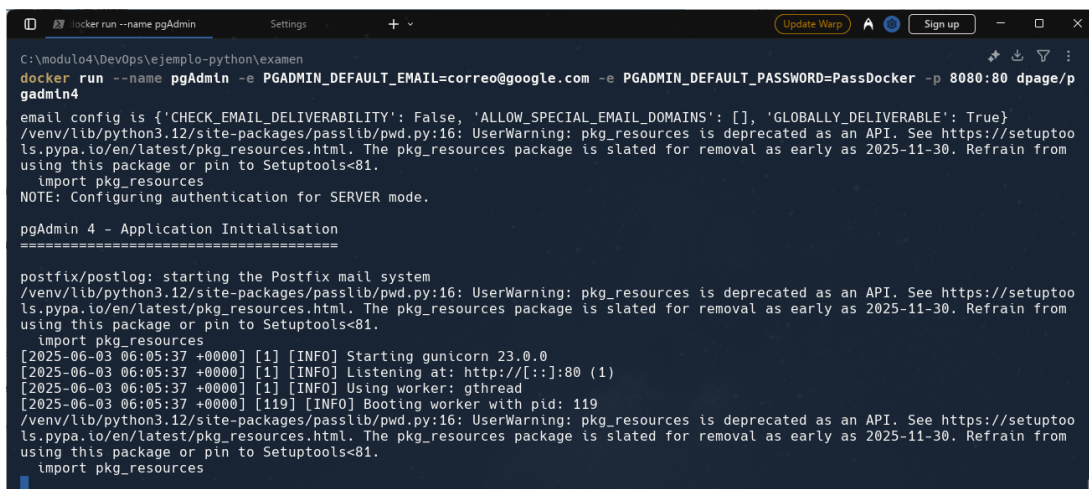
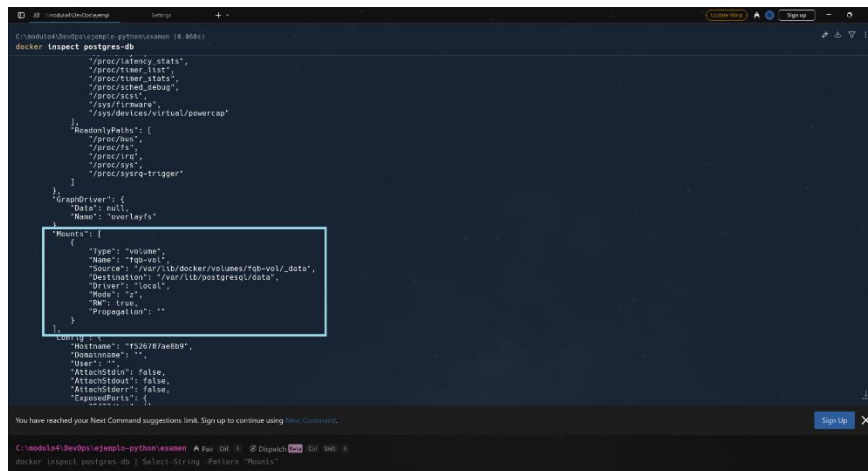
You have reached your Next Command suggestions limit. Sign up to continue using Next Command. Sign Up X

C:\modulo4\DevOps\ejemplo-python\examen A Pair Ctrl | Dispatch Beta Ctrl Shift |
docker inspect postgres-db | Select-String -Pattern "Mounts"
```

Para verificar ,los datos se coloca el siguiente comando:

```
docker inspect postgres-db
```

como se muestra en “Mounts” de la siguiente imagen:



Para verificar se coloca **docker ps**, si esta el pgadmin, entonces todo esta funcionando correctamente.

```
C:\modul4\DevOps\ejemplo-python\
ls
d-f--- 25/5/2025 23:38 tareagit

C:\modul4\DevOps\ejemplo-python\
cd ejemplo-python\

C:\modul4\DevOps\ejemplo-python\
ls

Directorio: C:\modul4\DevOps\ejemplo-python

Mode                LastWriteTime         Length Name
----                -
d----- 2/6/2025 19:19 Ejemplo1bno
d----- 2/6/2025 23:46 examen
d----- 2/6/2025 21:34 ml-app
d----- 2/6/2025 19:59 postgres-pgadmin

C:\modul4\DevOps\ejemplo-python\
cd examen

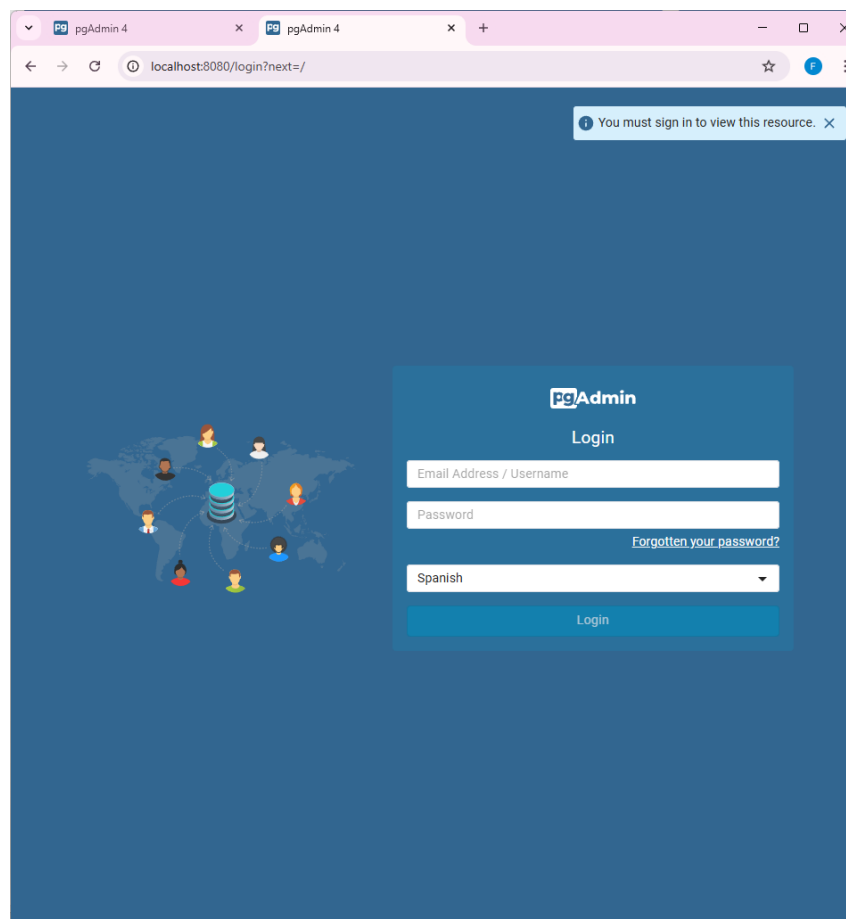
C:\modul4\DevOps\ejemplo-python\examen\
ls

C:\modul4\DevOps\ejemplo-python\examen\
docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS                               NAMES
f65d8f3eac4   dpago/pgadmin4  "docker-entrypoint.sh"  15 minutes ago  Up 15 minutes  443/tcp, 0.0.0.0:8080->80/tcp  pgadmin
f52676ae8b9   postgres    "docker-entrypoint.sh"  58 minutes ago  Up 58 minutes  0.0.0.0:5432->5432/tcp  postgres-db

C:\modul4\DevOps\ejemplo-python\examen\
docker stop pgadmin
```

4. Ingresar a pgAdmin con las credenciales.

Se va a la pagina y se coloca lo siguiente **localhost:8080**, te aparecerá lo siguiente:

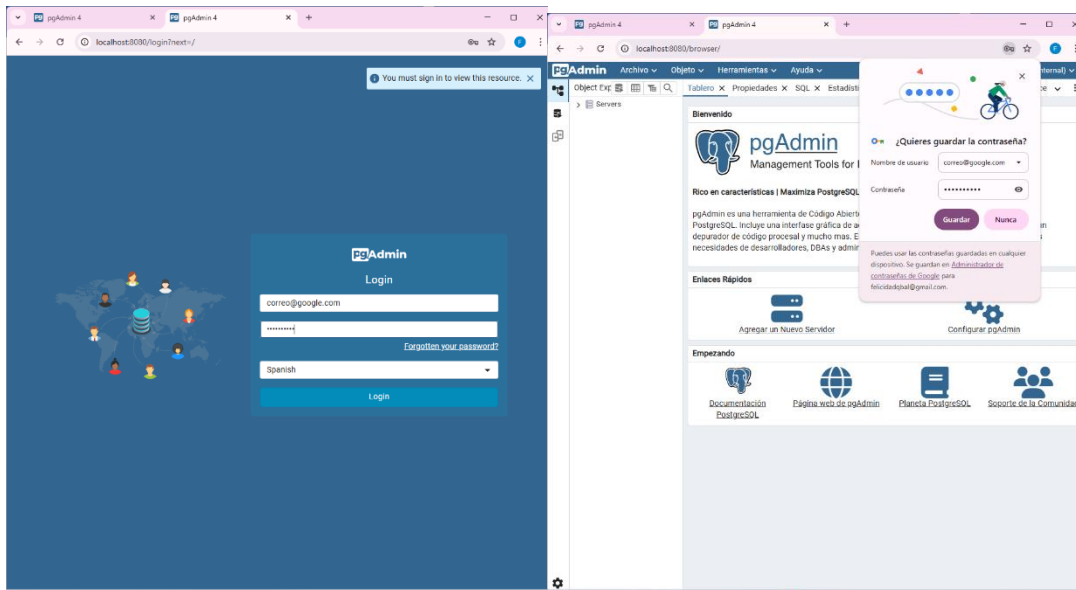


Luego pones las credenciales:

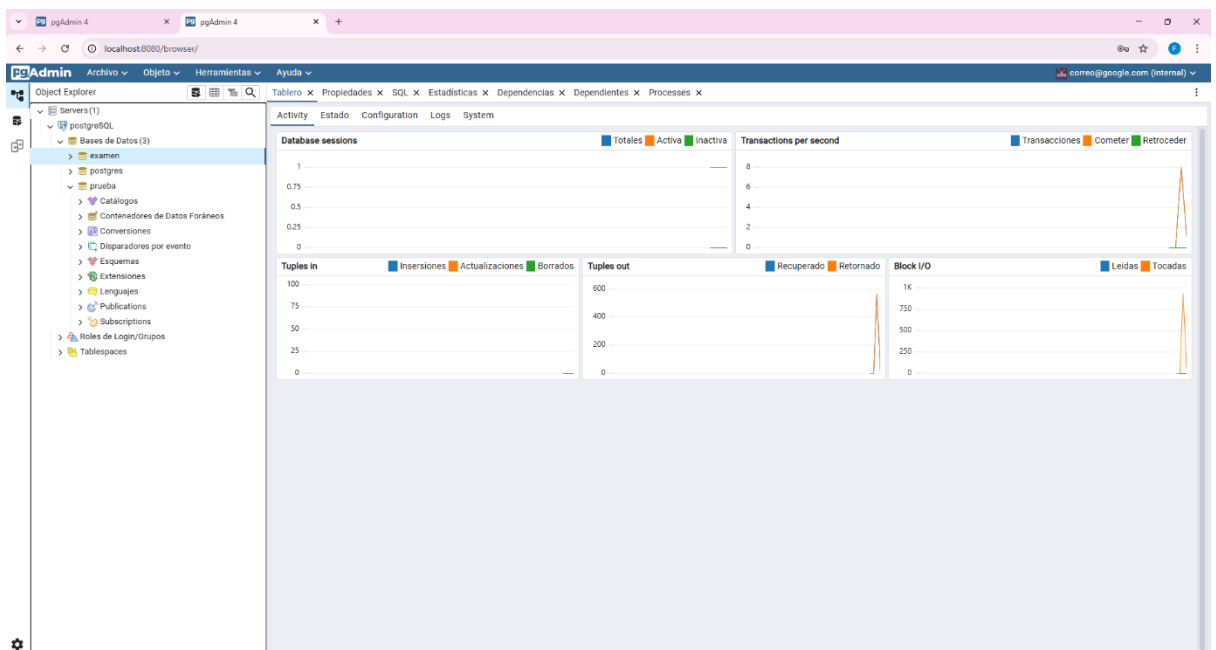
Correo: correo@google.com

Contraseña: PassDocker

Insertas los datos y luego presionas login, se muestra la otra imagen.



Presionas botón derecho en servers, luego eliges register, servidor, luego te aparece una ventana don de llenas los datos.



5. Conectar ambos contenedores a la red fq-bred

Se utilizaron los comandos que se muestra en la imagen para esta pregunta

```
C:\vnodeul04\DevOps\ejemplo-python>python examen (1h 26m 51s)
docker run --name pgAdmin -e PGADMIN_DEFAULT_EMAIL=correo@google.com -e PGADMIN_DEFAULT_PASSWORD=PassDocker -p 8080:80 dpape/pgadmin4

C:\vnodeul04\DevOps\ejemplo-python>python examen (0.122s)
docker network create fq-b-red
d9be3406696b960b727b15100a874fa868bcd77c228928753972b67fd06602197

C:\vnodeul04\DevOps\ejemplo-python>python examen (0.159s)
docker network connect fq-b-red postgres-db

C:\vnodeul04\DevOps\ejemplo-python>python examen (0.003s)
docker network connect fq-b-red pgAdmin

C:\vnodeul04\DevOps\ejemplo-python>python examen (0.093s)
docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED    STATUS    PORTS          NAMES
f526787ae0b9   postgres      "docker-entrypoint.s..."  2 hours ago  Up 2 hours    5432/tcp      postgres-db

C:\vnodeul04\DevOps\ejemplo-python>python examen (0.007s)
docker start postgres-db
postgres-db

C:\vnodeul04\DevOps\ejemplo-python>python examen (0.394s)
docker start pgAdmin
pgAdmin

C:\vnodeul04\DevOps\ejemplo-python>python examen (0.095s)
docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED    STATUS    PORTS          NAMES
f0bdaf3cae4c   dpape/pgadmin  "/entrypoint.sh"        2 hours ago  Up 3 seconds    443/tcp, 0.0.0.0:8080->80/tcp, 0.0.0.0:5432->5432/tcp      pgAdmin
f526787ae0b9   postgres      "docker-entrypoint.s..."  2 hours ago  Up 2 hours    5432/tcp      postgres-db
```