

Índice:

Ejercicio 1	3
1) Instalar docker y docker-compose en una máquina virtual Ubuntu Server	3
2) Conseguir ejecutar comandos de docker sin necesidad de sudo	4
3) Descargar la imagen Hello-world del repositorio oficial de docker.....	4
4) Listar las imágenes y contenedores existentes en tu MV, ejecutar la imagen anterior, comprobar su estado y finalmente borrar la imagen de tu MV.	4
5) Ejecutar el contenedor httpd poniéndole como nombre web y redirigiendo al puerto 8080 del host(tu MV) el puerto 80 del contenedor. Prueba el acceso a la web.....	6
6) Modificar la ejecución anterior para que tu contenido web se encuentre en una carpeta real de tu MV.....	7
7) Ejecutar 4 veces más el contenedor con las mismas características que el caso anterior, en cada caso a los puertos 8041, 8084 y con los nombres web2..web5	8
8) PostgreSQL	9
Ejercicio 2	12
9) Crear un volumen para almacenar la información de la base de datos	12
10) Montar la imagen de postgres así.....	12
11) Tomar pgAdmin de aquí.....	13
12) Ingresar a la web con las credenciales de superman.....	14
13) Intentar crear la conexión a la base de datos	14
14) Ohhh no!, no vemos la base de datos, se nos olvidó la red.....	15
15) Crear la red.....	16
16) Asignar ambos contenedores a la red.....	16
17) Conectar ambos contenedores.....	17
18) Intentar el paso 4. de nuevo.	17
19) Qué es Docker Compose	19
20) Crear Docker Compose	19
Ejercicio 3	21
21) Cread una capeta que se llame “pokemon-app”	21
22) Escribir en nuestro docker-compose	22
23) Probar la conexión en TablePlus seleccionando mongo.....	23
24) Ahora queremos que las conexiones al mongo estén autenticadas	25
25) Toca editar la conexión para que nos deje	27
26) ARCHIVO .env.....	29

27) Añadimos la imagen de mongo-express	30
28) Añadir a la imagen más variables de entorno, puertos y always	30
29) Levantamos el docker-compose	31
30) Probamos a acceder	32
31) Verificamos con table plus	33
32) Comentar los puertos de mongoDB.....	34
33) Probamos que sea inaccesible	35
34) Descargar la imagen Pokemon-Nest-App	36
35) Configurar la imagen, el depends, los puertos y always.....	37
36) Probamos el docker-compose	38
37) Para ejecutar la semilla	39
38) Acceder y ver las tablas.....	40
Ejercicio 4.....	41
39) Creamos el bulid context y archivos	41
40) Crearemos la imagen	42
41) Creando aplicaciones en contenedores.....	43
42) Balanceo de carga	46
43) Compartir imágenes.....	48
Ejercicio 5.....	51
44) Instalamos NGINX	51
45) Creación contenedor PHP	52
46) Creación de un contenedor para datos	55
47) Creación de contenedor MySQL	56

Ejercicio 1

1) Instalar docker y docker-compose en una máquina virtual Ubuntu Server

Solución:

Comandos utilizados:

-sudo apt update

-sudo apt upgrade

Paquetes de requisitos previos

-sudo apt-get install curl apt-transport-https ca-certificates software-properties-common

Agregar repositorios

-curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add

-sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable"

-sudo apt update

-apt-cache policy docker-ce

Instalamos Dockers

-sudo apt install docker-ce

Instalamos docker compose

-mkdir -p ~/.docker/cli-plugins/

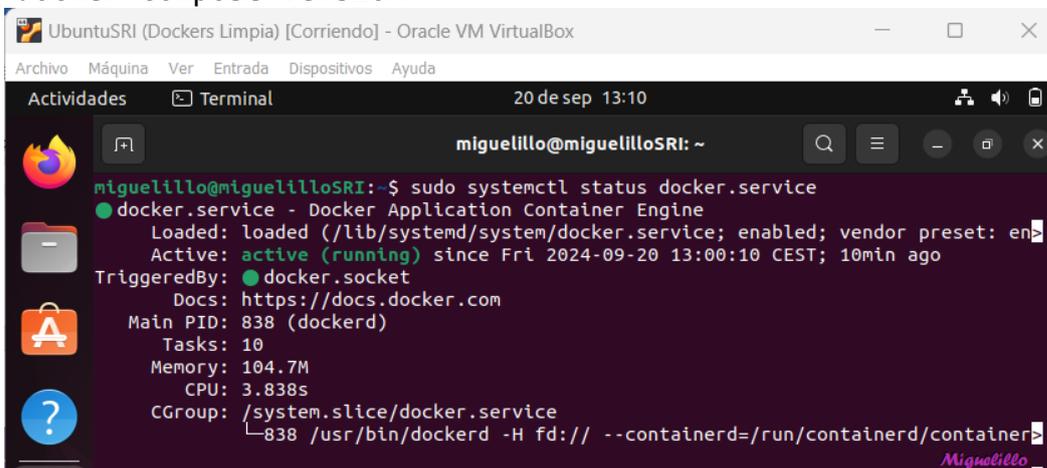
-curl

-SL

https://github.com/docker/compose/releases/download/v2.3.3/docker-compose-linux-x86_64 -o ~/.docker/cli-plugins/docker-compose

Comprobamos que se instaló

-docker compose version

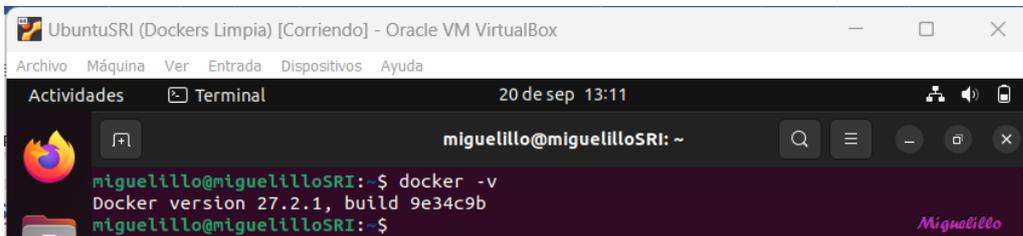


```

miguellillo@miguellilloSRI:~$ sudo systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: en
   Active: active (running) since Fri 2024-09-20 13:00:10 CEST; 10min ago
   TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 838 (dockerd)
   Tasks: 10
   Memory: 104.7M
   CPU: 3.838s
   CGroup: /system.slice/docker.service
           └─838 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd
  
```

2) Conseguir ejecutar comandos de docker sin necesidad de sudo

Solución:

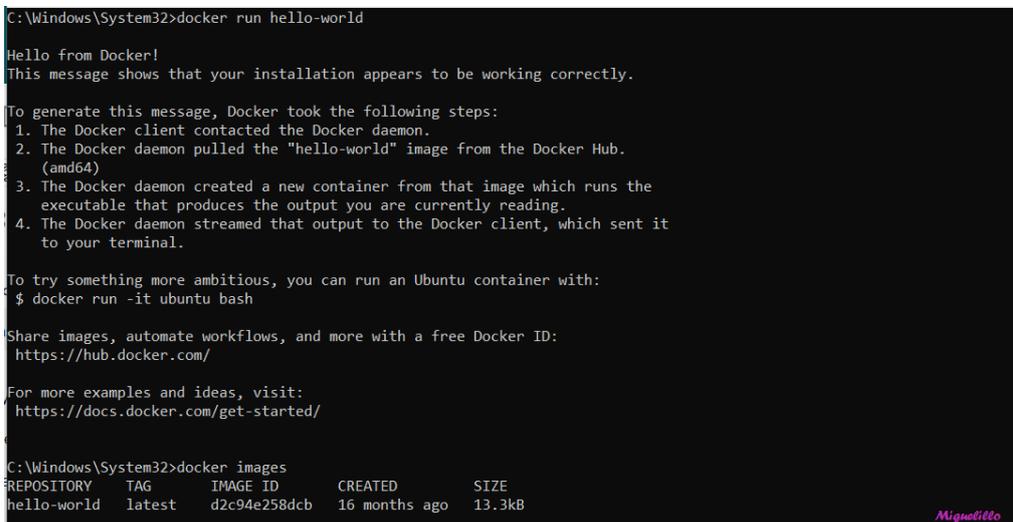


```

miguellillo@miguellilloSRI: ~
miguellillo@miguellilloSRI:~$ docker -v
Docker version 27.2.1, build 9e34c9b
miguellillo@miguellilloSRI:~$
  
```

3) Descargar la imagen Hello-world del repositorio oficial de docker.

Solución:



```

C:\Windows\System32>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

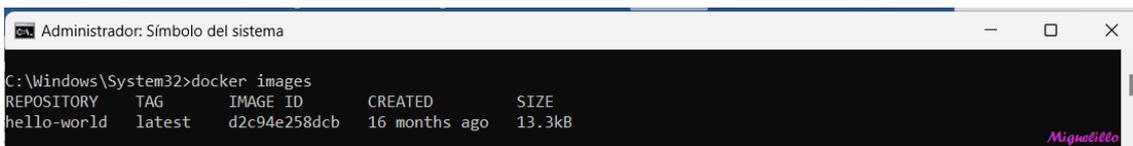
C:\Windows\System32>docker images
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
hello-world   latest   d2c94e258dcb  16 months ago  13.3kB
  
```

4) Listar las imágenes y contenedores existentes en tu MV, ejecutar la imagen anterior, comprobar su estado y finalmente borrar la imagen de tu MV.

Solución:

Para listarlas ´

docker images

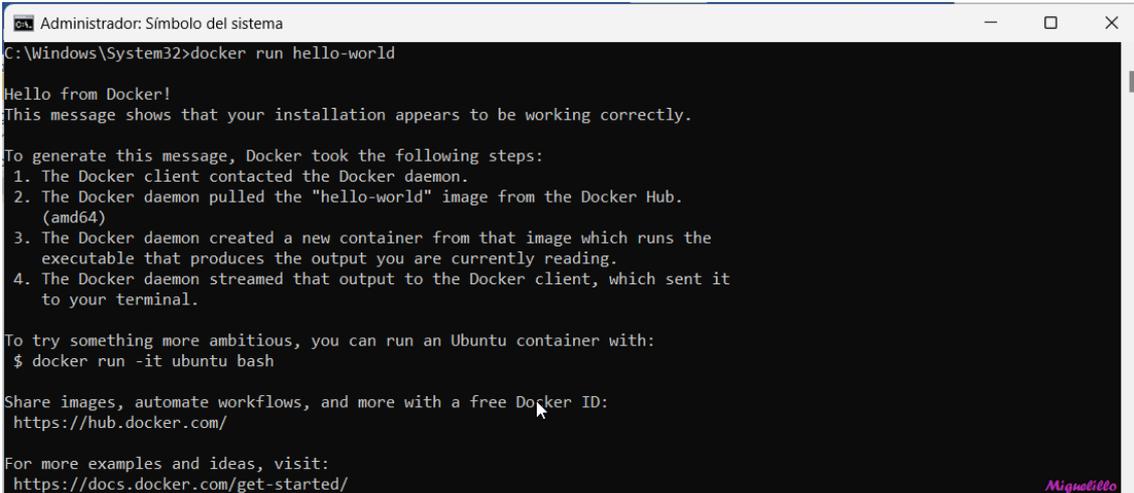


```

Administrador: Símbolo del sistema
C:\Windows\System32>docker images
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
hello-world   latest   d2c94e258dcb  16 months ago  13.3kB
  
```

Para ejecutarla:

```
docker run hello-world
```



```
Administrador: Símbolo del sistema
C:\Windows\System32>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

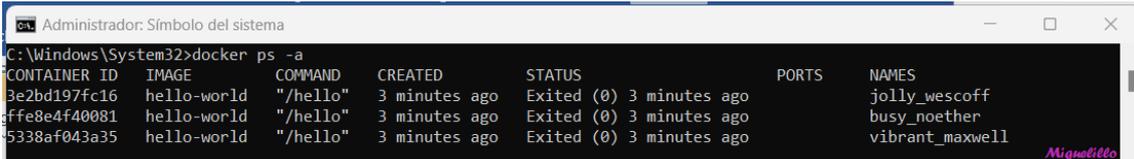
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Para comprobar el estado

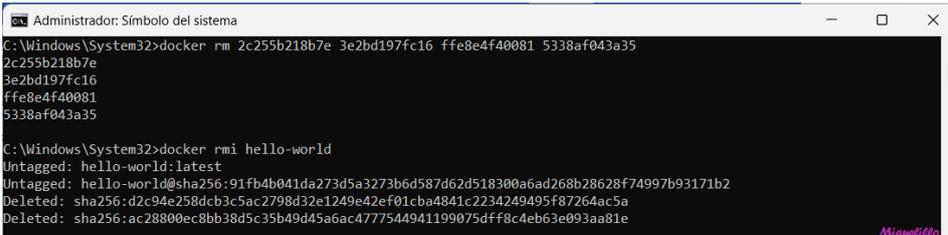
```
docker ps -a
```



```
Administrador: Símbolo del sistema
C:\Windows\System32>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS          PORTS          NAMES
3e2bd197fc16   hello-world    "/hello"                 3 minutes ago Exited (0) 3 minutes ago          jolly_wescoff
ffe8e4f40081   hello-world    "/hello"                 3 minutes ago Exited (0) 3 minutes ago          busy_noether
5338af043a35   hello-world    "/hello"                 3 minutes ago Exited (0) 3 minutes ago          vibrant_maxwell
```

Para eliminarlo, en este caso he tenido que forzar la detención

```
docker rmi hello-world
```



```
Administrador: Símbolo del sistema
C:\Windows\System32>docker rm 2c255b218b7e 3e2bd197fc16 ffe8e4f40081 5338af043a35
2c255b218b7e
3e2bd197fc16
ffe8e4f40081
5338af043a35

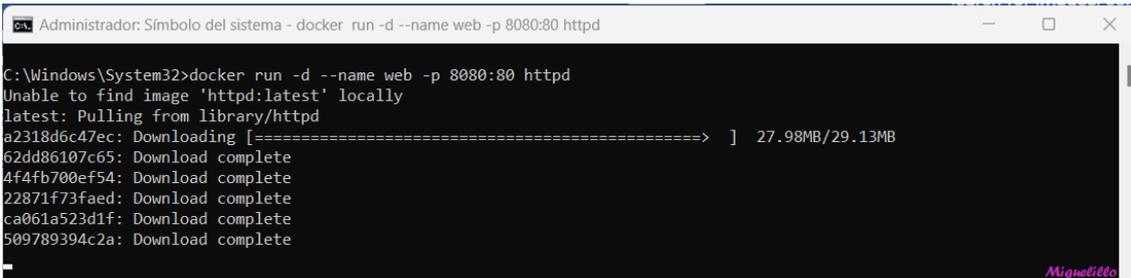
C:\Windows\System32>docker rmi hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:91fb4b041da273d5a3273b6d587d62d518300a6ad268b28628f74997b93171b2
Deleted: sha256:d2c94e258dcb3c5ac2798d32e1249e42ef01c8a4841c2234249495f87264ac5a
Deleted: sha256:ac28800ec8bb38d5c35b49d45a6ac4777544941199075dff8c4eb63e093aa81e
```

5) Ejecutar el contenedor httpd poniéndole como nombre web y redirigiendo al puerto 8080 del host(tu MV) el puerto 80 del contenedor. Prueba el acceso a la web.

Solución:

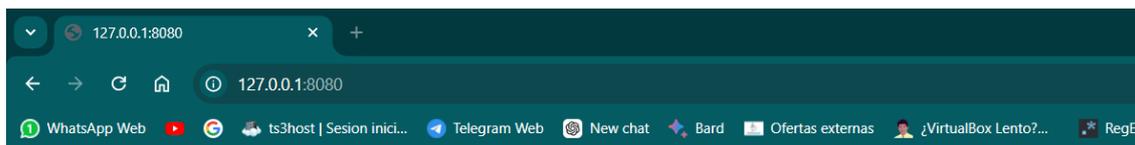
Descargamos el contenedor

```
docker run -p 8080:80 --name web httpd
```



```
Administrador: Símbolo del sistema - docker run -d --name web -p 8080:80 httpd
C:\Windows\System32>docker run -d --name web -p 8080:80 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
a2318d6c47ec: Downloading [=====] 27.98MB/29.13MB
62dd86107c65: Download complete
4f4fb700ef54: Download complete
22871f73faed: Download complete
ca061a523d1f: Download complete
509789394c2a: Download complete
```

Probamos que funcione introduciendo la IP y el puerto

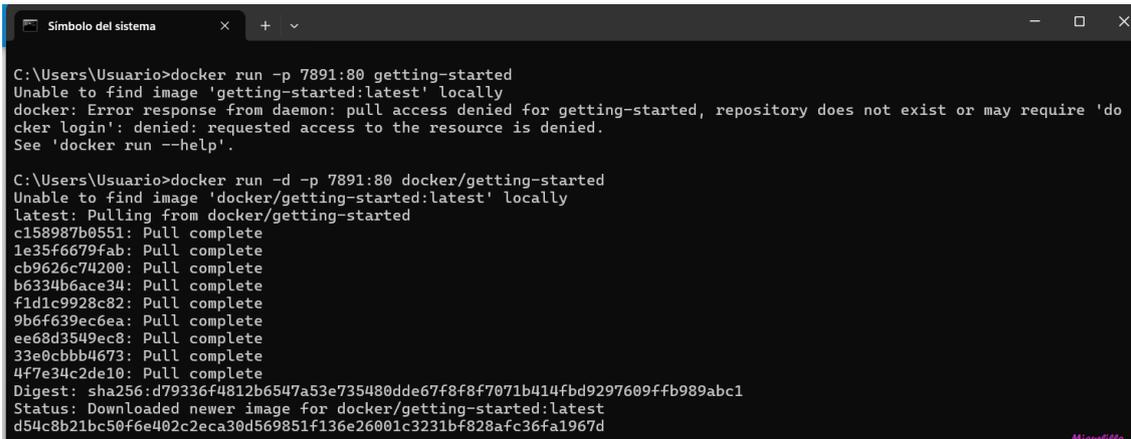


It works!

Miguelillo

Instamos el getting started para Windows

```
docker run -d -p 7891:80 docker/getting-started
```

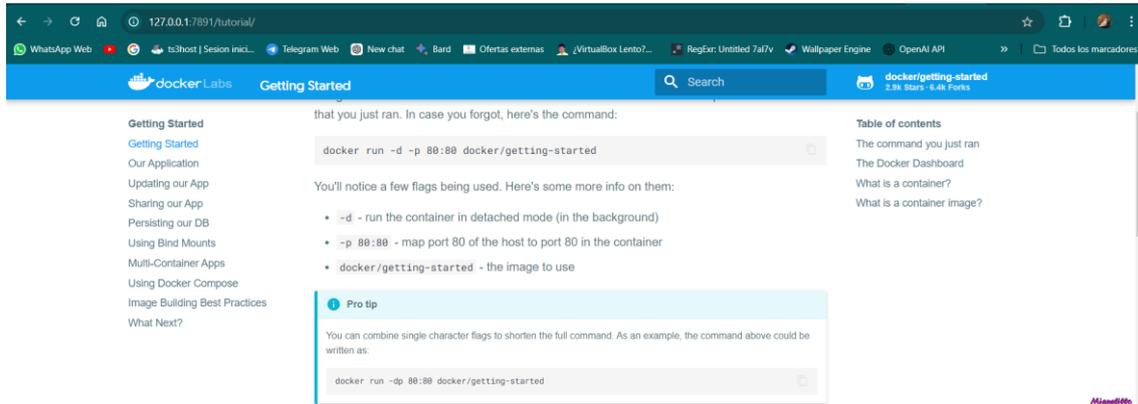


```
Símbolo del sistema
C:\Users\Usuario>docker run -p 7891:80 getting-started
Unable to find image 'getting-started:latest' locally
docker: Error response from daemon: pull access denied for getting-started, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.

C:\Users\Usuario>docker run -d -p 7891:80 docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
c158987b0551: Pull complete
1e35f6679fab: Pull complete
cb9626c74200: Pull complete
b6334b6ace34: Pull complete
fd1c9928c82: Pull complete
9b6f639ec6ea: Pull complete
ee68d3549ec8: Pull complete
33e0cbbb4673: Pull complete
4f7e34c2de10: Pull complete
Digest: sha256:d79336f4812b6547a53e735480dde67f8f8f7071b414fbd9297609ffb989abc1
Status: Downloaded newer image for docker/getting-started:latest
d54c8b21bc50f6e402c2eca30d569851f136e26001c3231bf828afc36fa1967d
```

Miguelillo

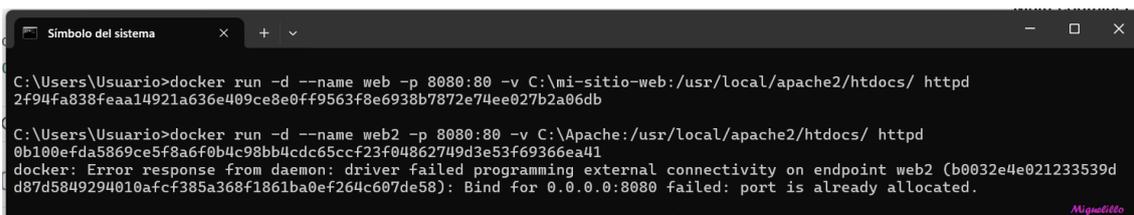
Comprobamos su funcionamiento



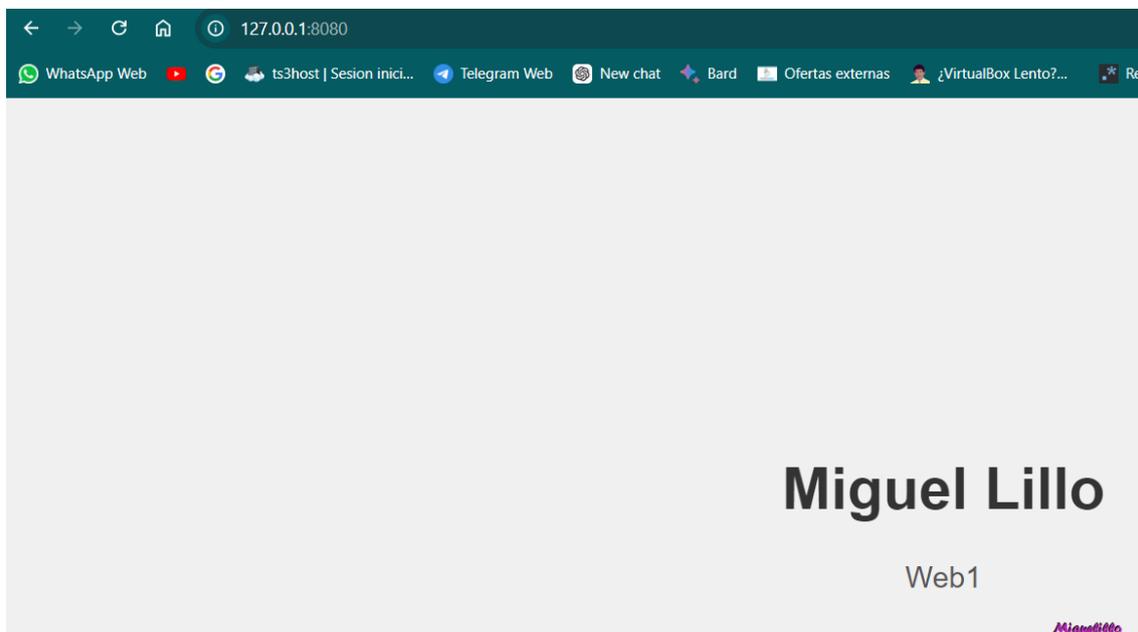
6) Modificar la ejecución anterior para que tu contenido web se encuentre en una carpeta real de tu MV.

Solución:

Instalas httpd y lo creas especificando la con la ruta de los archivos



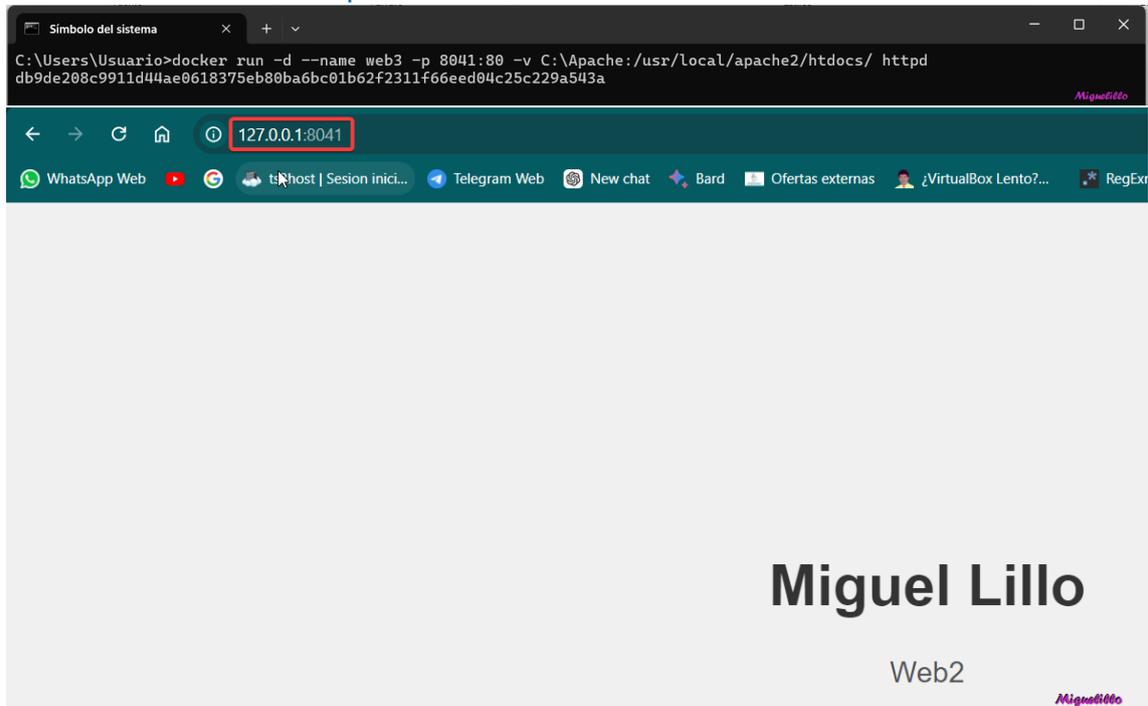
Pruebas que funcione



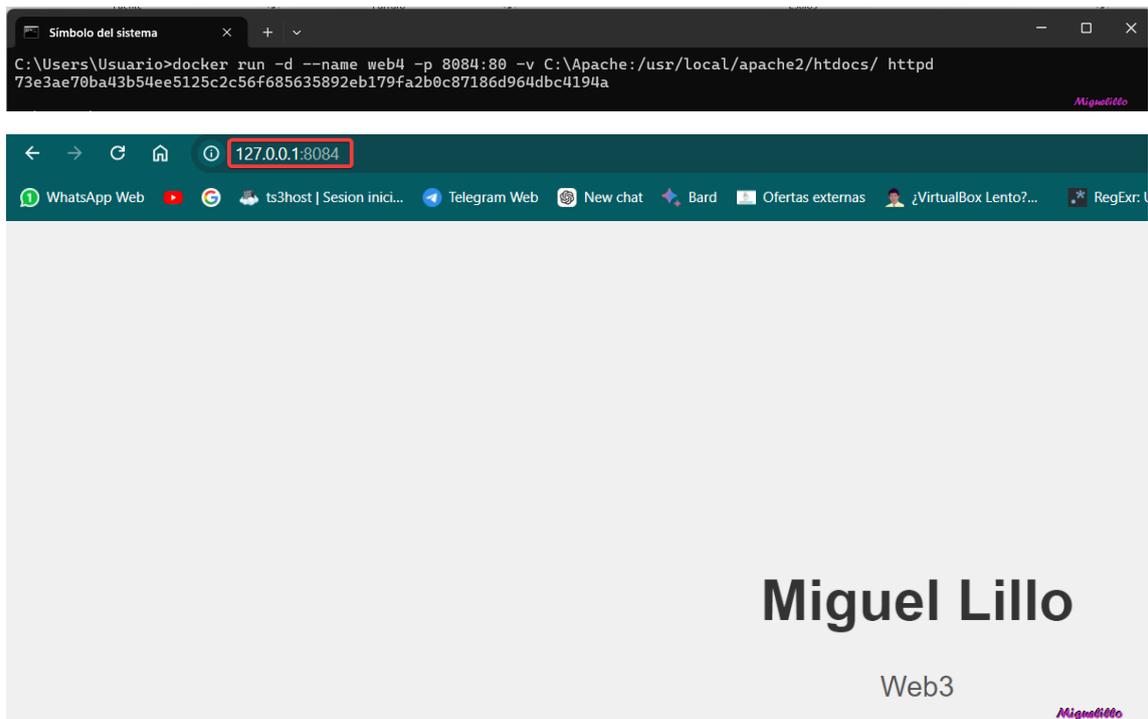
7) Ejecutar 4 veces más el contenedor con las mismas características que el caso anterior, en cada caso a los puertos 8041, 8084 y con los nombres web2..web5

Solución:

Creamos los diferentes puertos el 8041



El puerto 8084

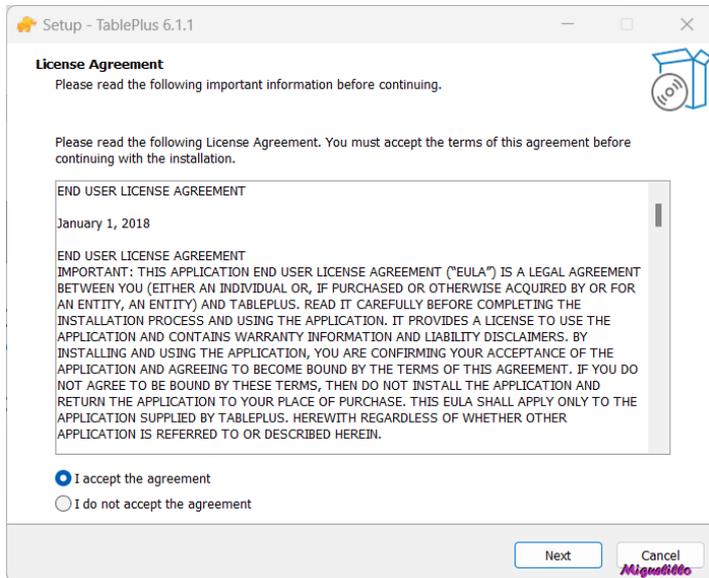


Y repetimos con las otras 2 páginas

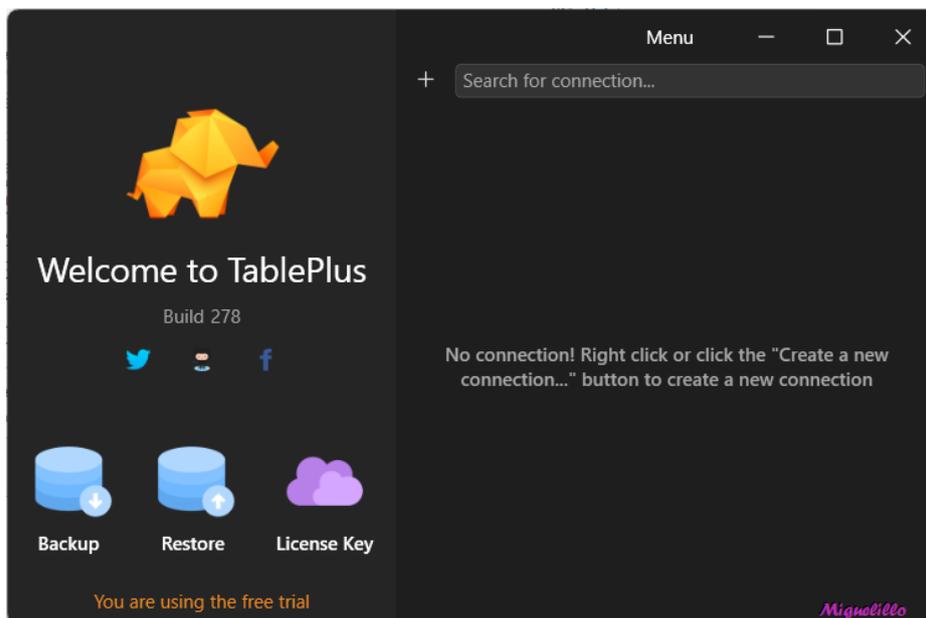
8) PostgreSQL

Solución:

Instalamos TablePlus

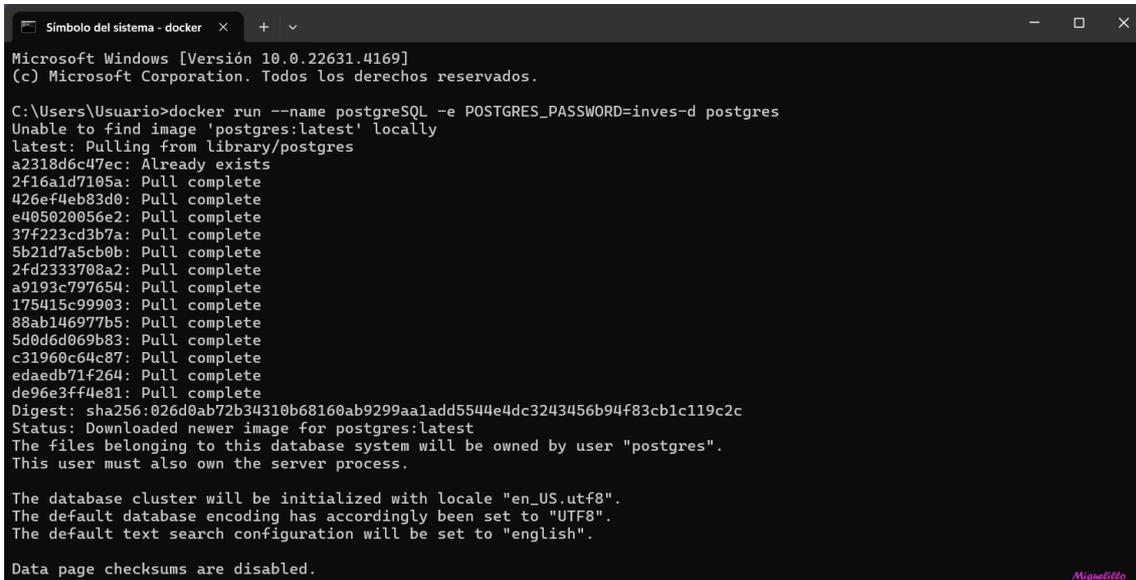


Se ha instalado correctamente, este nos servirá para conectarnos al postgre



Instalamos PostgreSQL

```
docker run --name postgresQL -e POSTGRES_PASSWORD=inves -p 8081:5432 -d postgres
```



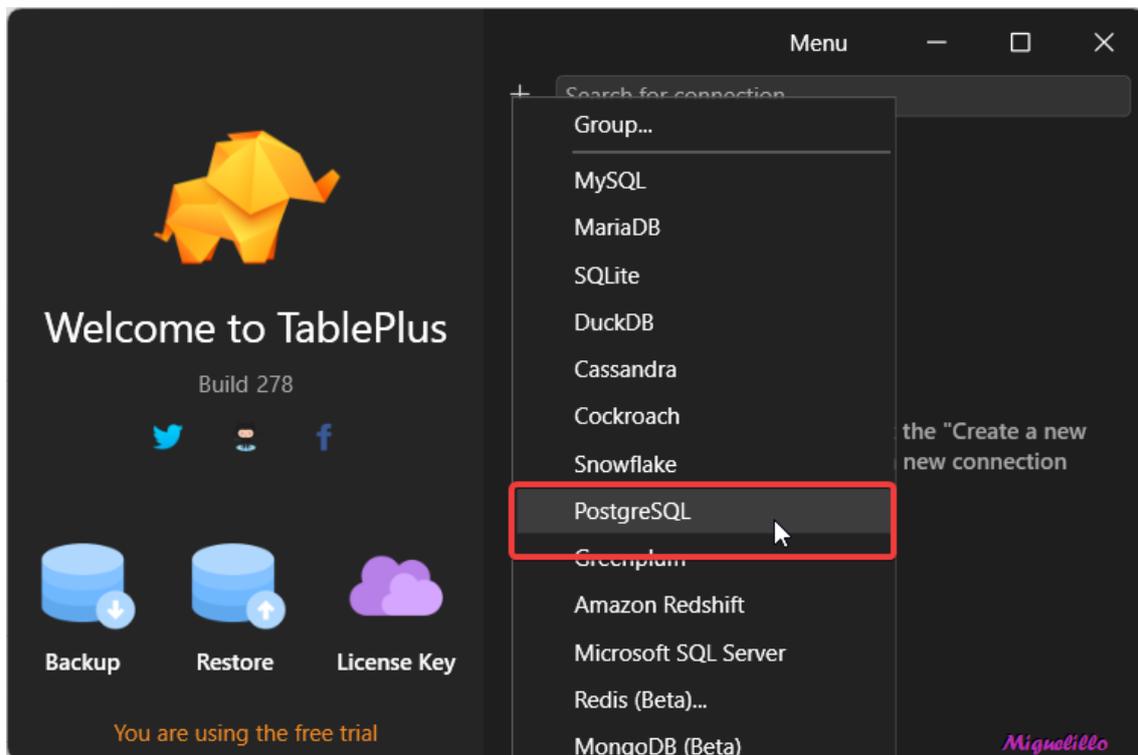
```
Microsoft Windows [Versión 10.0.22631.4169]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>docker run --name postgresQL -e POSTGRES_PASSWORD=inves-d postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
a2318d6c47ec: Already exists
2f16a1d7105a: Pull complete
426ef4eb83d0: Pull complete
e405020056e2: Pull complete
37f223cd3b7a: Pull complete
5b21d7a5cb0b: Pull complete
2fd2333708a2: Pull complete
a9193c707654: Pull complete
175415c99903: Pull complete
88ab146977b5: Pull complete
5d0d6d069b83: Pull complete
c31960c64c87: Pull complete
edaedb71f264: Pull complete
de96e3ff4e81: Pull complete
Digest: sha256:026d0ab72b34310b68160ab9299aa1add5544e4dc3243456b94f83cb1c119c2c
Status: Downloaded newer image for postgres:latest
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

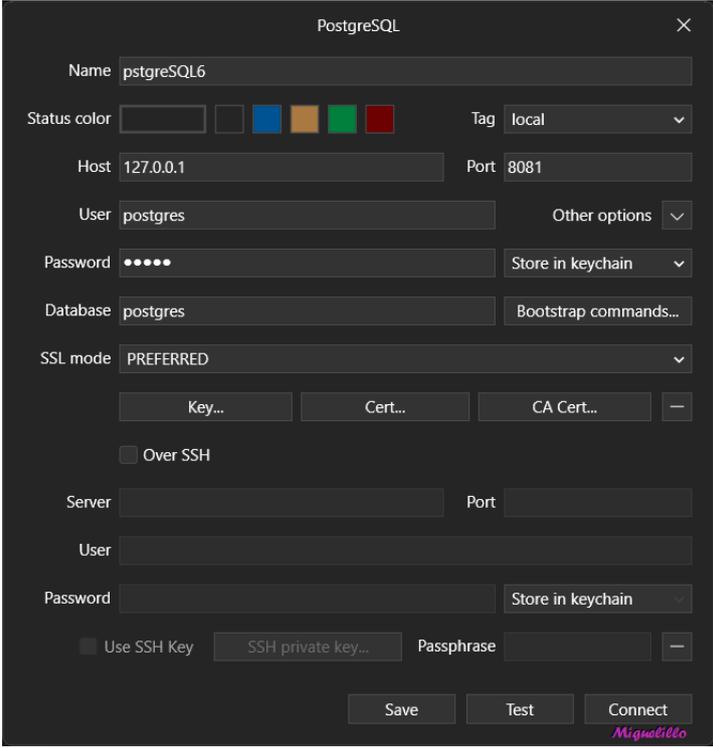
The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.
```

Para conectarnos



Añadimos los datos

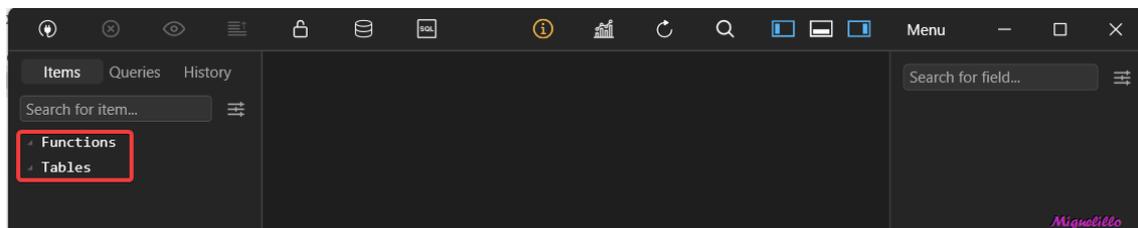


The image shows a configuration window for a PostgreSQL connection. The window title is "PostgreSQL" and it has a close button (X) in the top right corner. The configuration fields are as follows:

- Name: pstgreSQL6
- Status color: A row of five colored squares (grey, blue, orange, green, red).
- Tag: local (dropdown menu)
- Host: 127.0.0.1
- Port: 8081
- User: postgres
- Other options: (dropdown menu)
- Password: (masked with dots)
- Store in keychain: (dropdown menu)
- Database: postgres
- Bootstrap commands...: (button)
- SSL mode: PREFERRED (dropdown menu)
- Key...: (button)
- Cert...: (button)
- CA Cert...: (button)
- Over SSH: (checkbox, unchecked)
- Server: (text field)
- Port: (text field)
- User: (text field)
- Password: (text field)
- Store in keychain: (dropdown menu)
- Use SSH Key: (checkbox, unchecked)
- SSH private key...: (text field)
- Passphrase: (text field)

At the bottom, there are three buttons: "Save", "Test", and "Connect". A small pink signature "Miguelllo" is visible in the bottom right corner of the dialog.

Probamos a conectarnos



Ejercicio 2

9) Crear un volumen para almacenar la información de la base de datos

Solución:

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.4169]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\Usuario>docker volume create postgres-db-miguel
postgres-db-miguel
C:\Users\Usuario>
```

10) Montar la imagen de postgres así

Solución:

Comando

```
docker container run `
-d `
--name postgres-db `
-e POSTGRES_PASSWORD=123456 `
-v postgres-db:/var/lib/postgresql/data `
postgres:15.1
```

```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

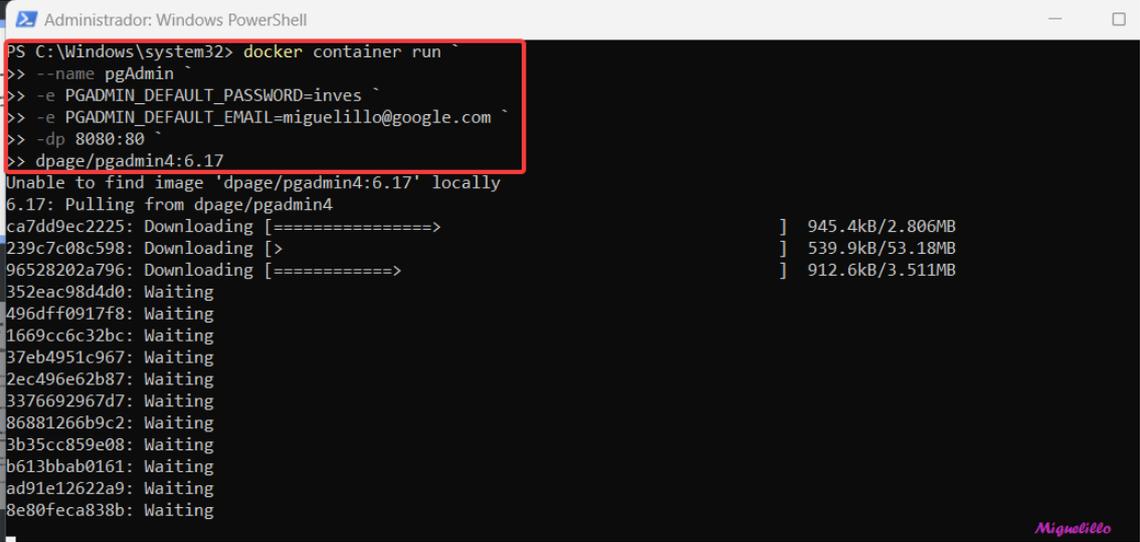
PS C:\Windows\system32> docker container run `
>> -d `
>> --name postgres-db `
>> -e POSTGRES_PASSWORD=123456 `
>> -v postgres-db:/var/lib/postgresql/data `
>> postgres:15.1
Unable to find image 'postgres:15.1' locally
15.1: Pulling from library/postgres
bb263680fed1: Downloading [=====] 18.2MB/31.41MB
75a54e59e691: Pulling fs layer
3ce7f8df2b36: Download complete
f30287ef02b9: Download complete
dc1f0e9024d8: Downloading [=====] 4.963MB/8.045MB
7f0a68628bce: Waiting
32b11818cae3: Waiting
48111fe612c1: Waiting
07b5cb2894c7: Waiting
7cca76b73db0: Waiting
87f7b375a7d2: Waiting
d9daaa1dc184: Waiting
536a8b356450: Waiting
```

11) Tomar pgAdmin de aquí

Solución:

Instalamos pgAdmin

```
docker container run `
--name pgAdmin `
-e PGADMIN_DEFAULT_PASSWORD=inves `
-e PGADMIN_DEFAULT_EMAIL=miguelillo@google.com `
-dp 8080:80 `
dpage/pgadmin4:6.17
```

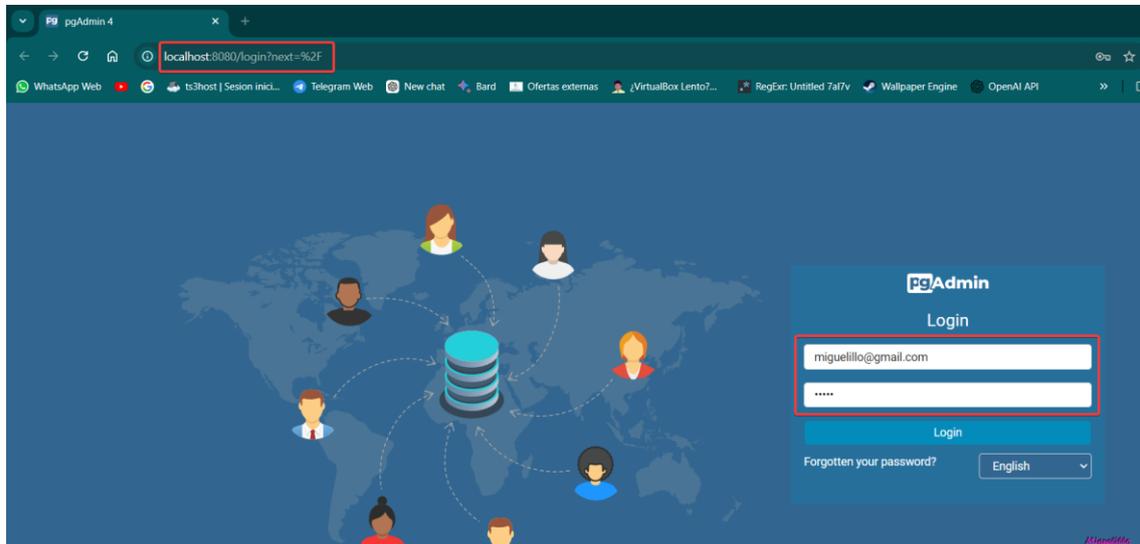


```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker container run `
>> --name pgAdmin `
>> -e PGADMIN_DEFAULT_PASSWORD=inves `
>> -e PGADMIN_DEFAULT_EMAIL=miguelillo@google.com `
>> -dp 8080:80 `
>> dpage/pgadmin4:6.17
Unable to find image 'dpage/pgadmin4:6.17' locally
6.17: Pulling from dpage/pgadmin4
ca7dd9ec2225: Downloading [=====>] 945.4kB/2.806MB
239c7c08c598: Downloading [>] 539.9kB/53.18MB
96528202a796: Downloading [=====>] 912.6kB/3.511MB
352eac98d4d0: Waiting
496dff0917f8: Waiting
1669cc6c32bc: Waiting
37eb4951c967: Waiting
2ec496e62b87: Waiting
3376692967d7: Waiting
86881266b9c2: Waiting
3b35cc859e08: Waiting
b613bbab0161: Waiting
ad91e12622a9: Waiting
8e80feca838b: Waiting
```

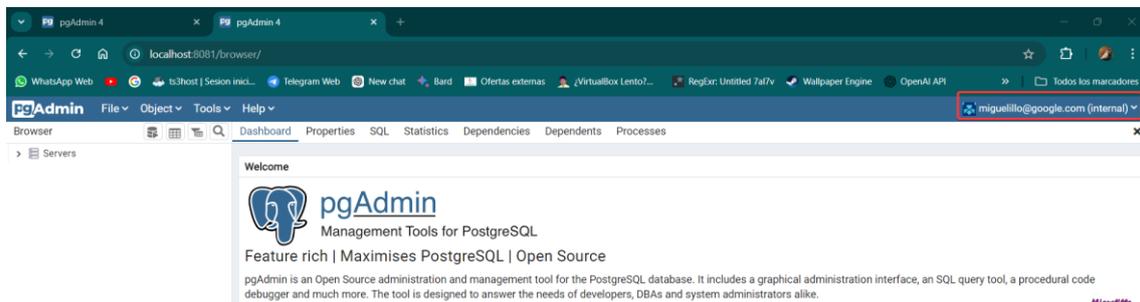
12) Ingresar a la web con las credenciales de superman

Solución:

Vamos a la web y nos logueamos

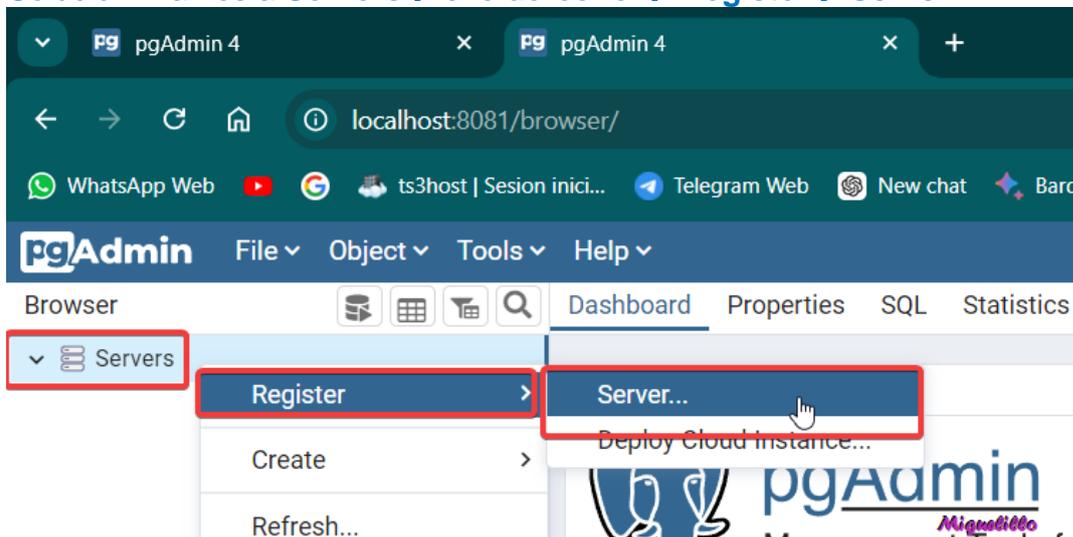


Hemos iniciado sesión

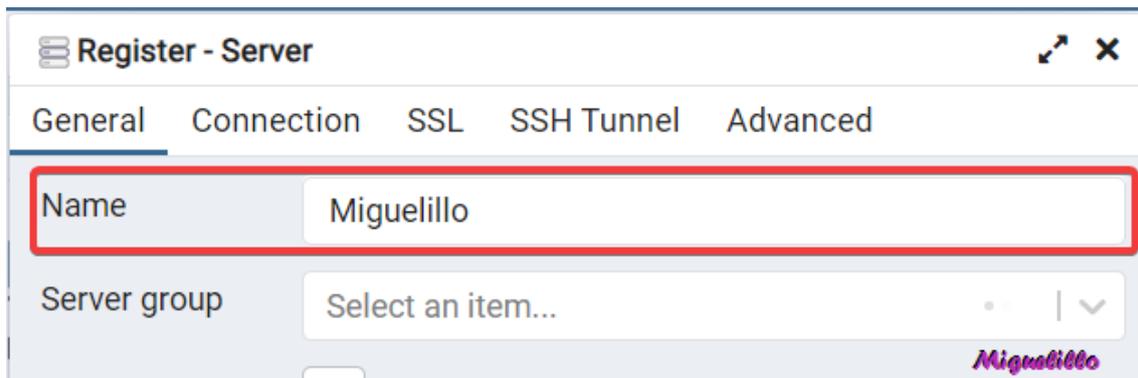


13) Intentar crear la conexión a la base de datos

Solución: Vamos a **Servers** → clic derecho → **Register** → **Server...**



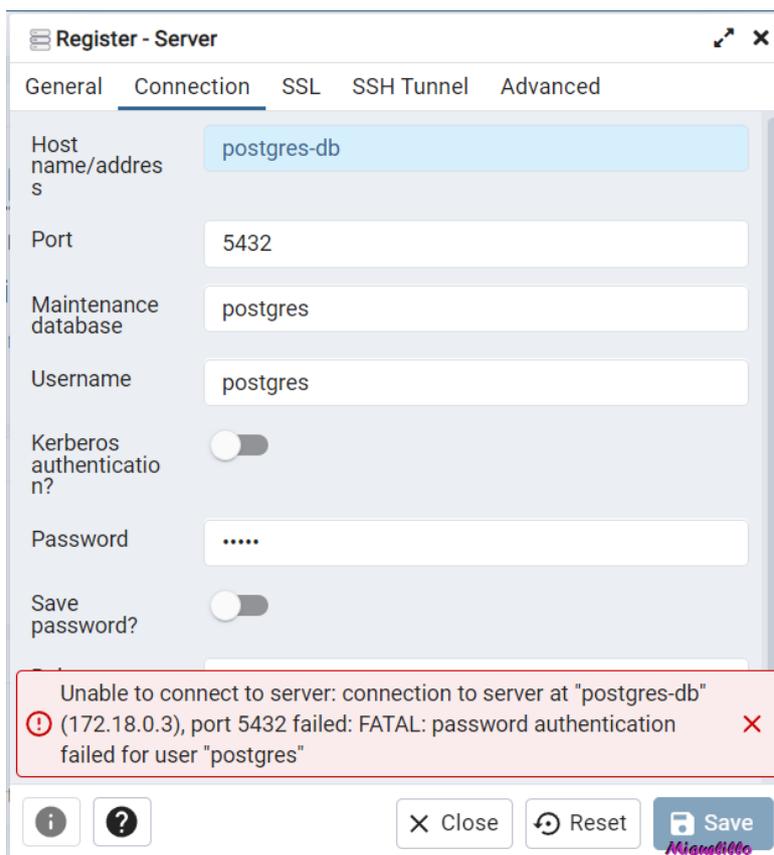
Le ponemos el nombre



14) Ohhh no!, no vemos la base de datos, se nos olvidó la red

Solución:

Nos da errores como los siguientes



Register - Server

General Connection SSL SSH Tunnel Advanced

Host name/addresses: postgres-dd

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication?

Password:

Save password?

Role:

Unable to connect to server: could not translate host name "postgres-dd" to address: Name does not resolve

Close Reset Save

15) Crear la red

Solución:

```
docker network create postgres-net
```

```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker network create postgres-net
df500312661ac86bbf39cfc2c40efa0e0ca6a89439ea783197ba760a72a74896
PS C:\Windows\system32>
```

16) Asignar ambos contenedores a la red

Solución:

```
docker container ps
```

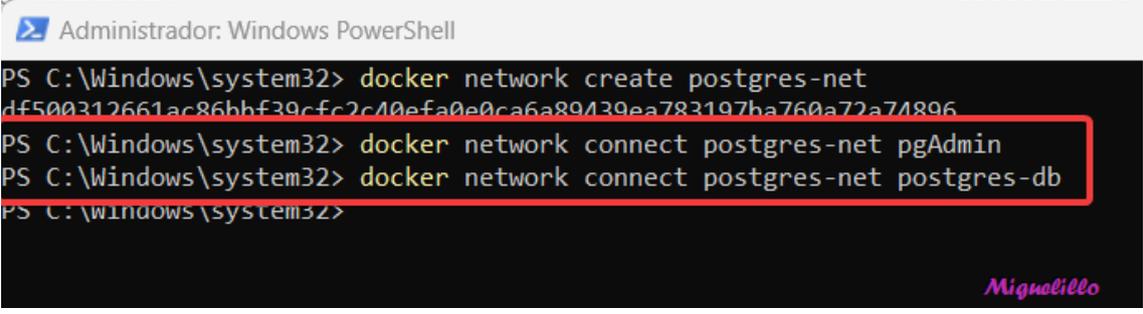
```
Administrador: Windows PowerShell
PS C:\Windows\system32> docker container ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
2e21c427c630   dpag/pgadmin4:6.17  "/entrypoint.sh"        19 minutes ago  Up 19 minutes  443/tcp, 0.0.0.0:8081->80/tcp      pgAdmin2
9b32638692a7   dpag/pgadmin4:6.17  "/entrypoint.sh"        23 minutes ago  Up 23 minutes  443/tcp, 0.0.0.0:8080->80/tcp      pgAdmin
22737bed6170   postgres:15.1     "docker-entrypoint.s..."  33 minutes ago  Up 33 minutes  5432/tcp                          postgres-db
PS C:\Windows\system32>
```

17) Conectar ambos contenedores

Solución:

```
docker network connect postgres-net
```

```
docker network connect pgAdmin
```



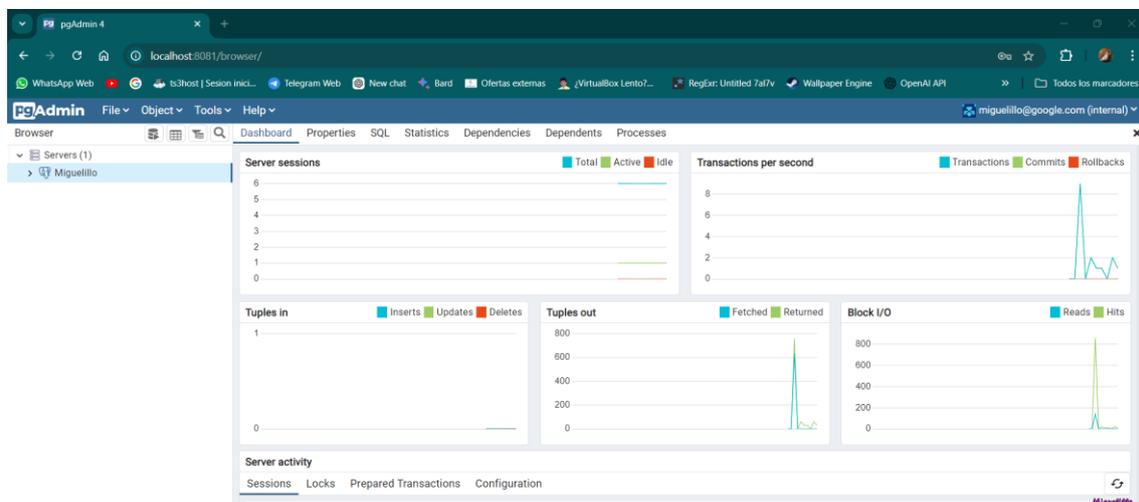
```
Administrador: Windows PowerShell  
PS C:\Windows\system32> docker network create postgres-net  
df500312661ac86bbf39cfc2c40efa0e0ca6a89439ea783197ba760a72a74896  
PS C:\Windows\system32> docker network connect postgres-net pgAdmin  
PS C:\Windows\system32> docker network connect postgres-net postgres-db  
PS C:\Windows\system32>
```

Miguelillo

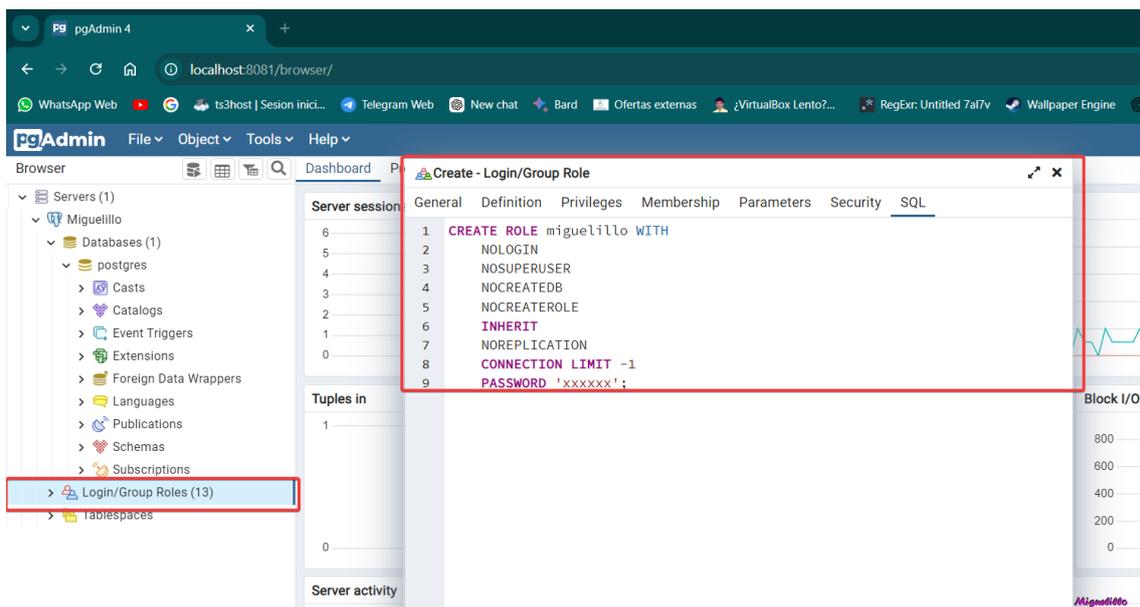
18) Intentar el paso 4. de nuevo.

Solución:

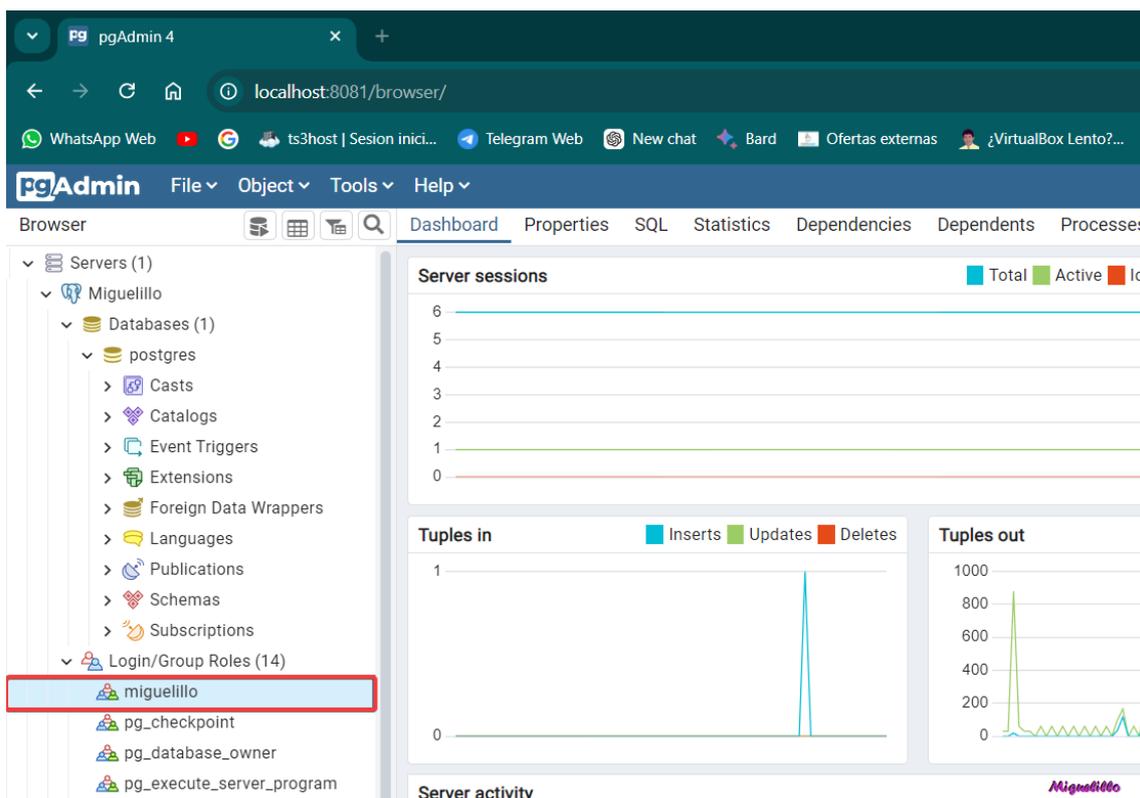
Se conecta correctamente



Creamos un usuario



Se ha creado el usuario



19) Qué es Docker Compose

Docker Compose es una utilidad de Docker que permite ejecutar varios contenedores simultáneamente reuniéndolo en un solo contenedor

20) Crear Docker Compose

Creamos un Docker Compose, este reúne todos los pasos anteriormente

```

51-55  pgAdmin.yaml
Archivo  Editar  Ver

version: '3'
services:
  postgres-db:
    image: postgres:15.1
    container_name: postgres-db-new
    environment:
      POSTGRES_PASSWORD: 123456
    volumes:
      - postgres-data:/var/lib/postgresql/data
    networks:
      - postgres-net

  pgadmin:
    image: dpage/pgadmin4:6.17
    container_name: pgAdmin-new
    environment:
      PGADMIN_DEFAULT_EMAIL: superman@google.com
      PGADMIN_DEFAULT_PASSWORD: 123456
    ports:
      - "8080:80"
    networks:
      - postgres-net

volumes:
  postgres-data:
  |
networks:
  postgres-net:

```

Probamos el docker Compose vamos a la ruta donde lo tenemos almacenado y lo ejecutamos con

```
docker-compose up -d
```

```

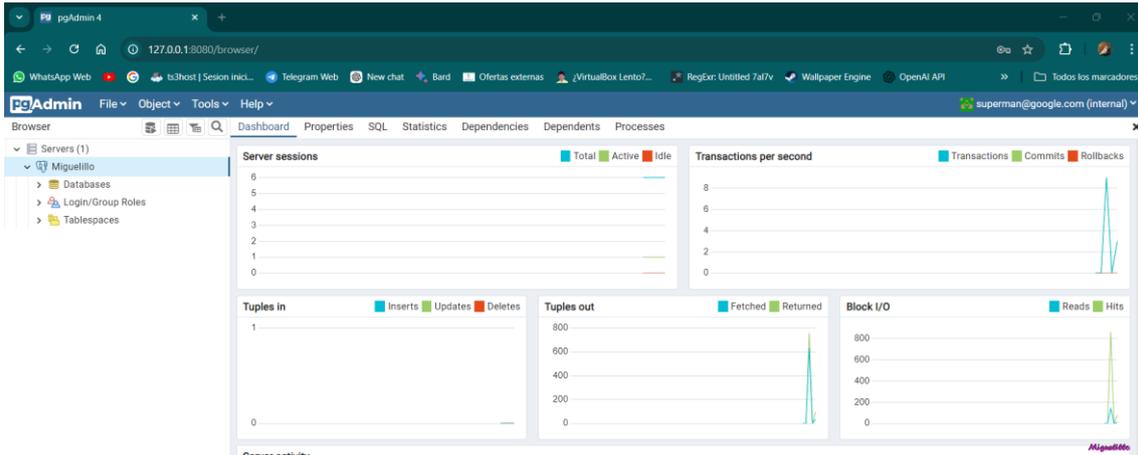
Administrador: Windows PowerShell
PS C:\Users\Usuario\Desktop> docker-compose up -d
time="2024-09-25T10:15:39+02:00" level=warning msg="C:\\Users\\Usuario\\Desktop\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
✔ Network desktop_postgres-net Created 0.1s
✔ Volume "desktop_postgres-data" Created 0.0s
✔ Container postgres-db-new Started 0.7s
✔ Container pgAdmin-new Started 0.7s
PS C:\Users\Usuario\Desktop>

```

Se ha creado correctamente

Nombre	Imagen	Estado	PUERTO	Uso de CPU	Uptime	Acciones
desktop		Running (2/2)		99.18%	31 seconds ago	🗑️
pgAdmin-new	dpage/pgadmin4:6.17	Running	8080:80	99.15%	31 seconds ago	🗑️
postgres-db-new	postgres:15.1	Running		0.03%	31 seconds ago	🗑️

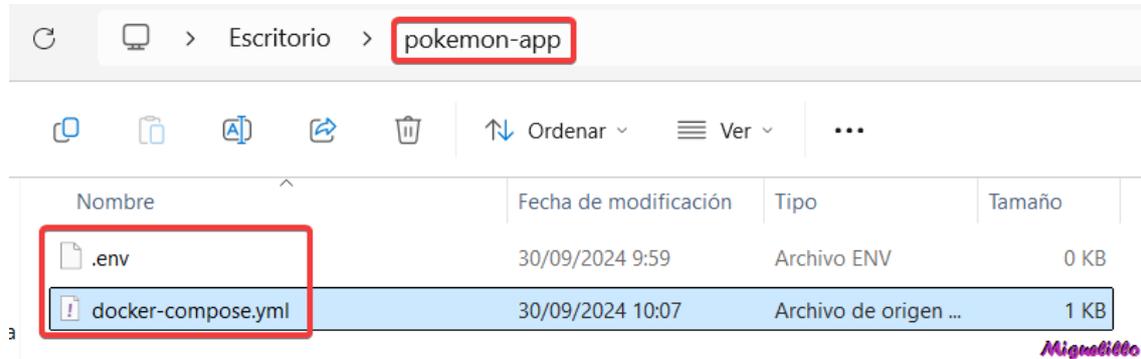
Funciona correctamente



Ejercicio 3

21) Cread una capeta que se llame “pokemon-app”

Solución:



22) Escribir en nuestro docker-compose

Solución:

Editamos el archivo

```
docker-compose.yml X
C: > Users > Usuario > Desktop > pokemon-app > docker-compose.yml
 1  services:
 2      db:
 3          container_name: miguel
 4          image: mongo:6.0
 5          volumes:
 6              - poke-vol:/data/db
 7          ports:
 8              - "27017:27017"
 9          restart: always
10
11  volumes:
12      poke-vol:
13          external: false
```

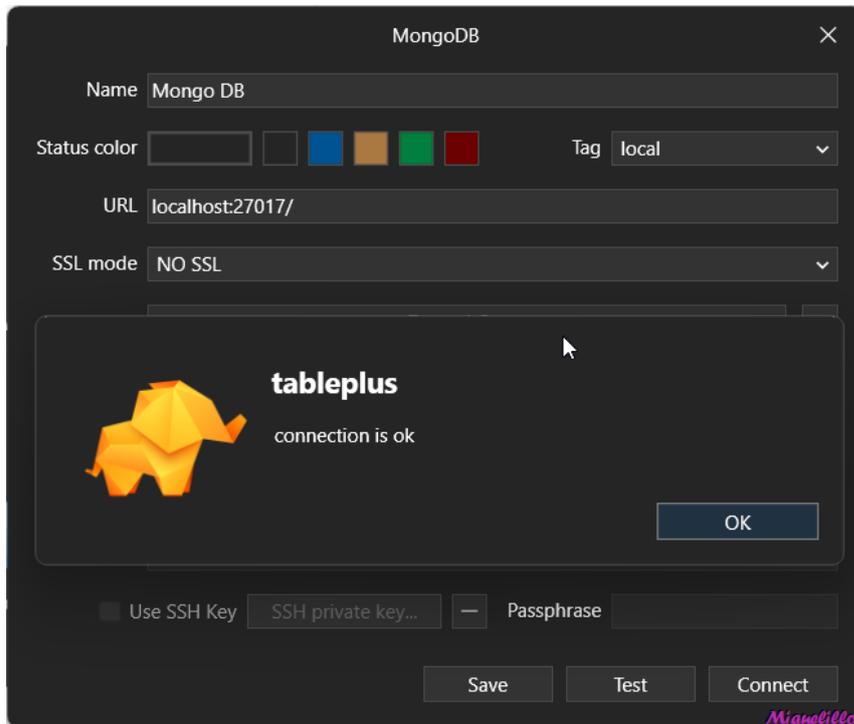
Lo ejecutamos

```
Administrador: Windows PowerShell
PS C:\Users\Usuario\Desktop\pokemon-app> docker-compose up -d
[+] Running 9/9
  ✓ db Pulled                                     48.7s
  ✓ 6414378b6477 Pull complete                    11.1s
  ✓ d7afefe395b7 Pull complete                    11.2s
  ✓ ebbd3a7433d0 Pull complete                    11.7s
  ✓ 2aa2f76c98aa Pull complete                    11.8s
  ✓ 7e2502eff1b1 Pull complete                    11.9s
  ✓ ce9d82391f28 Pull complete                    11.9s
  ✓ 685fe1322518 Pull complete                    45.1s
  ✓ bb550bff6854 Pull complete                    45.2s
[+] Running 3/3
  ✓ Network pokemon-app_default Created           0.1s
  ✓ Volume "pokemon-app_poke-vol" Created        0.0s
  ✓ Container miguel Started                     1.3s
PS C:\Users\Usuario\Desktop\pokemon-app>
```

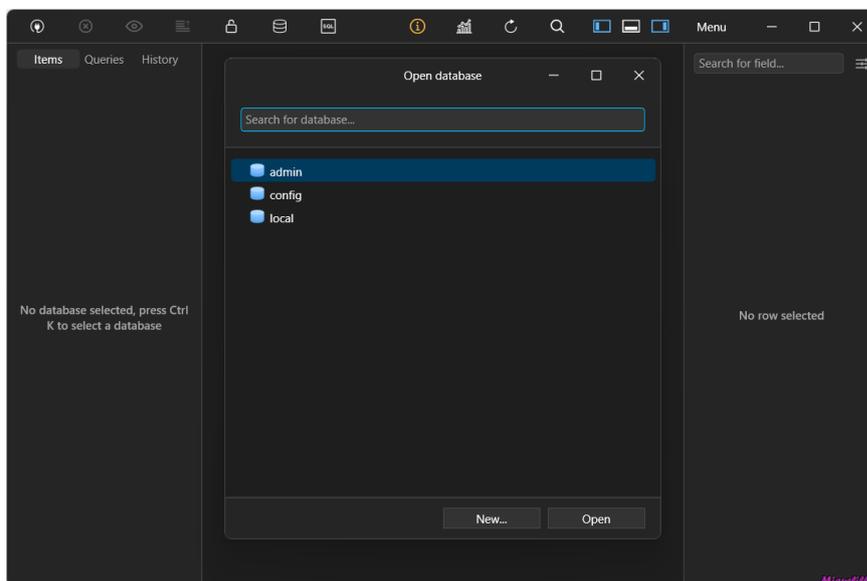
23) Probar la conexión en TablePlus seleccionando mongo

Solución:

Probamos la conexión y es exitosa



Vemos que accede correctamente



<  **pokemon-app**
C:\Users\Usuario\Desktop\pokemon-app

View Configurations



 **miguel**
[mongo:6.0](#)
Running
[27017:27017](#)

```
ecture":{"x86_64"},"platform":{"cfg=0x0204170063 CC=MSVC 1929 CFLAGS=\\\"/DWIN32 /D
_WINDOWS /W3\\\" LDFLAGS=\\\"/machine:x64\\\"}}}}
2024-09-30 13:16:34 miguel | {"t":{"$date":"2024-09-30T11:16:34.258+00:00"},"s
":"I", "c":"WTCHKPT", "id":22430, "ctx":"Checkpoint", "msg":"WiredTiger me
ssage", "attr":{"message":{"ts_sec":1727694994,"ts_usec":257962,"thread":"1:0x7f
c075ff1640","session_name":"WT_SESSION.checkpoint","category":"WT_VERB_CHECKPOI
NT_PROGRESS","category_id":6,"verbose_level":"DEBUG","verbose_level_id":1,"msg
":"saving checkpoint snapshot min: 312, snapshot max: 312 snapshot count: 0, old
est timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 1"}}}
2024-09-30 13:16:40 miguel | {"t":{"$date":"2024-09-30T11:16:40.984+00:00"},"s
":"I", "c":"NETWORK", "id":22943, "ctx":"listener", "msg":"Connection accept
ed", "attr":{"remote":"172.20.0.1:45320","uid":"a51723d4-d944-4584-b7f0-3f68ac8
57548","connectionId":5,"connectionCount":3}}
2024-09-30 13:16:40 miguel | {"t":{"$date":"2024-09-30T11:16:40.984+00:00"},"s
":"I", "c":"NETWORK", "id":51800, "ctx":"conn5", "msg":"client metadata", "at
```



Miguelillo

24) Ahora queremos que las conexiones al mongo estén autenticadas

Solución:

Añadimos las variables para introducir el usuario y la contraseña

```
docker-compose.yml X
C: > Users > Usuario > Desktop > pokemon-app > docker-compose.yml
1  services:
2    db:
3      container_name: miguel
4      image: mongo:6.0
5      volumes:
6        - poke-vol:/data/db
7      ports:
8        - "27017:27017"
9      restart: always
10     environment:
11       - MONGO_INITDB_ROOT_USERNAME=admin
12       - MONGO_INITDB_ROOT_PASSWORD=inves
13  volumes:
14    poke-vol:
15     external: false
```

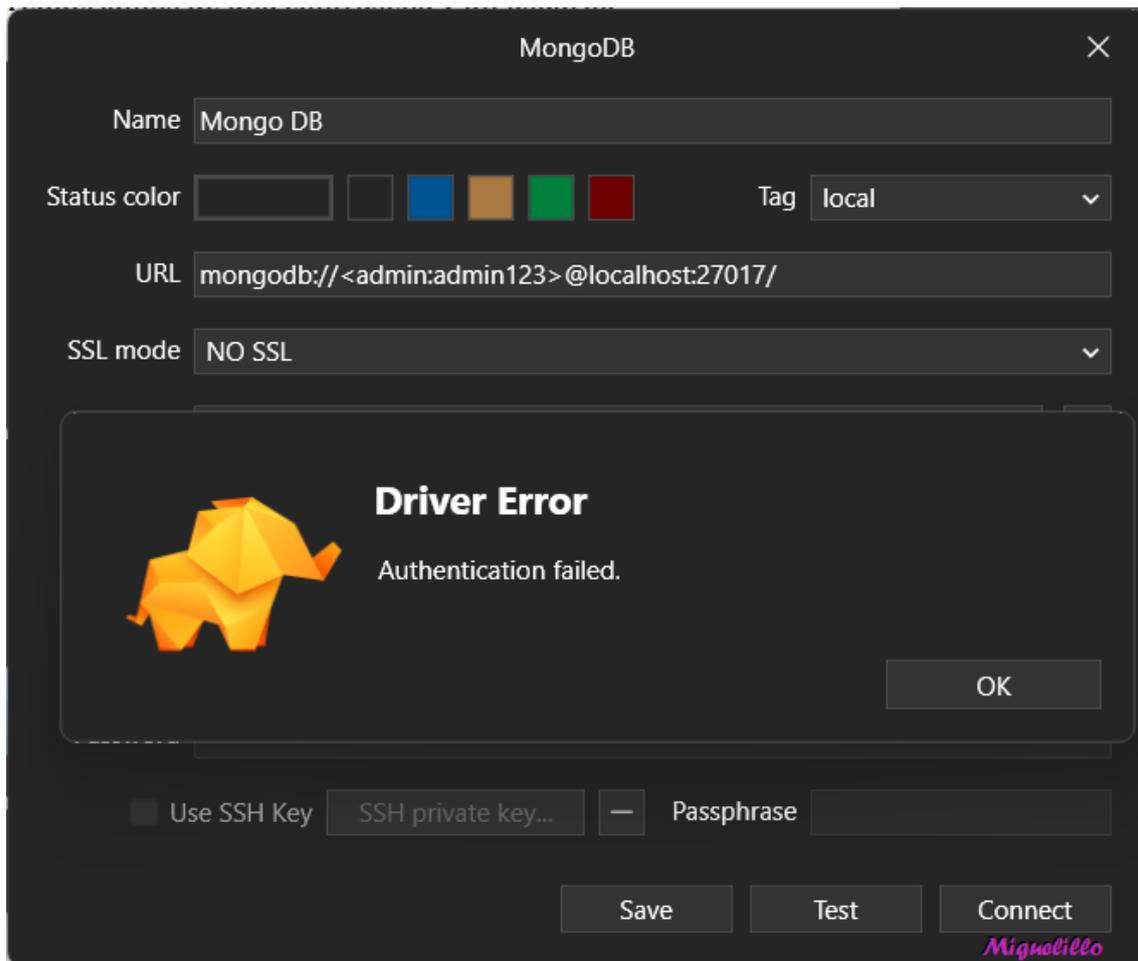
Añadimos la siguiente opción para que nos pida la autenticación

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
docker-compose.yml X
C: > Users > Usuario > Desktop > pokemon-app > docker-compose.yml
1  services:
2    db:
3      container_name: miguel
4      image: mongo:6.0
5      volumes:
6        - poke-vol:/data/db
7      ports:
8        - "27017:27017"
9      restart: always
10     command: mongod --auth
11     environment:
12       - MONGO_INITDB_ROOT_USERNAME=admin
13       - MONGO_INITDB_ROOT_PASSWORD=inves
14  volumes:
15    poke-vol:
16     external: false
```

Ejecutamos el docker compose

```
PS C:\users\usuario\desktop\pokemon-app> docker-compose down
[+] Running 2/2
✔ Container miguel Removed 0.5s
✔ Network pokemon-app_default Removed 0.3s
PS C:\users\usuario\desktop\pokemon-app> docker-compose up -d
[+] Running 2/2
✔ Network pokemon-app_default Created 0.1s
✔ Container miguel Started 0.6s
PS C:\users\usuario\desktop\pokemon-app>
```

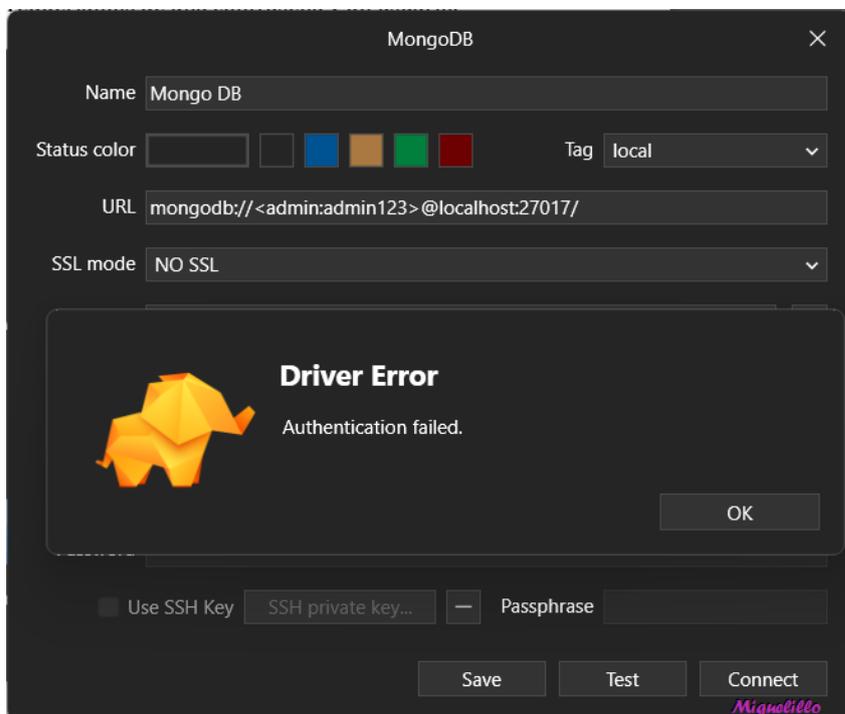
Probamos a conectarnos y nos da el siguiente fallo



25) Toca editar la conexión para que nos deje

Solución:

Fallo de autenticación



Listamos los volúmenes

```

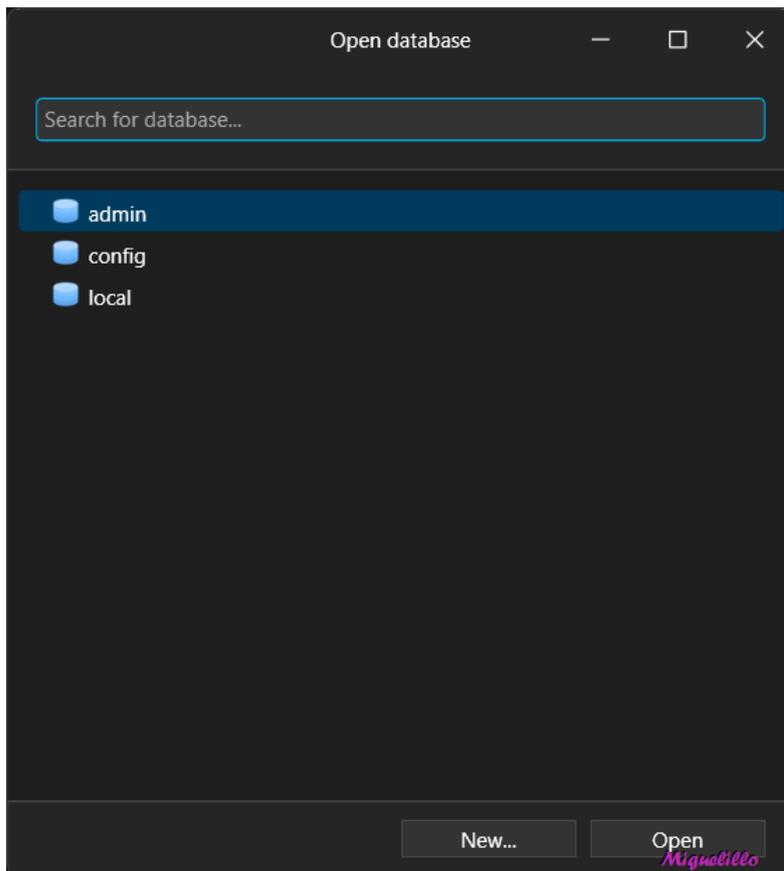
Administrador: Windows PowerShell
PS C:\users\usuario\Desktop\pokemon-app> docker volume ls
DRIVER      VOLUME NAME
local       3fdfb672701e80c4de2eb3c284b2bf07c0ce9803d96565cd825d8490df104bae
local       5ba9c1b0f08e813eddb41a7555f6cdc50657a97a52ff28e01652cdf602c633e8
local       7cb5993c47836a64a2bd6565822d05b180f03ca00b15ab244c83b6aa44595cf2
local       15ffa4f1aa022bd0882aba649c989e44279bd45e458a2177df4f3192b0784ad6
local       3137d93884946a8f17318818b42c2c8334a9da8eba5cc2c1a54de77d6974924b
local       80464f1b57f6ba20fe904e6f1f0be5c6ae3090451bc1b72ffe3758259e38cac0
local       3416274c6774de3d4db52d14d9a2c0aa78e7b1a5163e34f7ac53e3e486aa46a9
local       af7cf03aea7cba2d580fbb0498bdc1ee7430149bb8eff13e09279a4f909c126
local       bfb19ecc57e4433b7d6a9279c7044c4b2cd08042f1e2f5826ae6bc29c305a49b
local       cfaacebe6e0533eba1c8b823c76dd1d3ad312ccf5c08d3700552c5941d866034
local       d215d73e46650b1ed89364cd7f96bcf38b3bb2e7f066417ccca6ef227255a2cf
local       desktop-postgres-data
local       pokemon-app_poke-vol
local       postgres-db
local       postgres-db-miguel
  
```

Borramos el volumen

```

local       postgres-db-miguel
PS C:\users\usuario\Desktop\pokemon-app> docker volume rm pokemon-app_poke-vol
pokemon-app_poke-vol
PS C:\users\usuario\Desktop\pokemon-app> docker-compose up -d
[+] Running 3/3
 ✓ Network pokemon-app_default      Created
 ✓ Volume "pokemon-app_poke-vol"    Created
 ✓ Container miguel                  Started
PS C:\users\usuario\Desktop\pokemon-app>
  
```

Nos podemos conectar correctamente

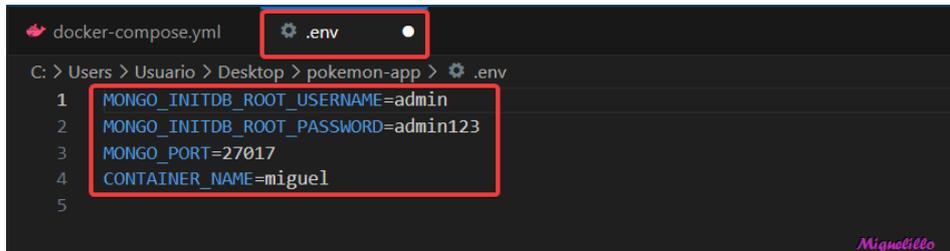


26) ARCHIVO .env

Solución:

Un archivo **.env** es un **archivo de texto** que contiene **variables de entorno** y se utiliza para **configurar parámetros**

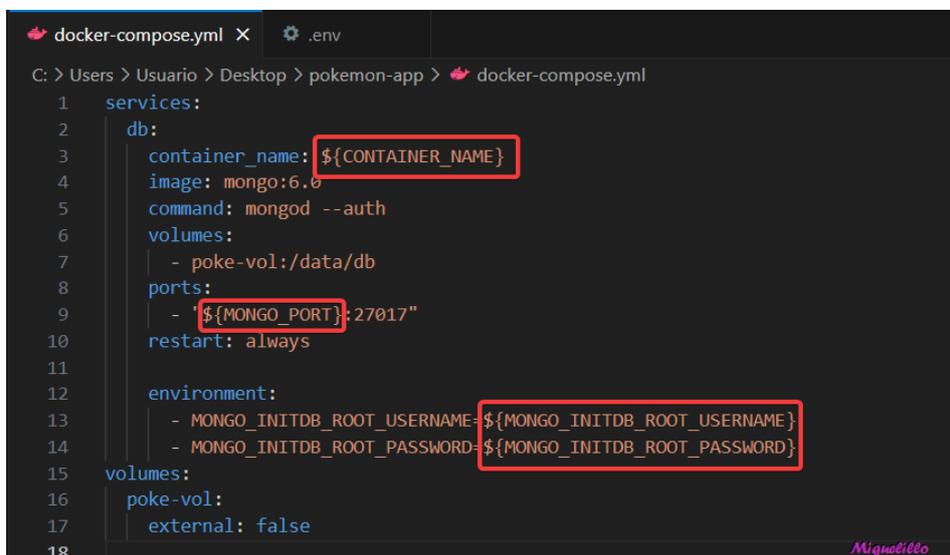
Editamos el **.env**



```
docker-compose.yml .env
C: > Users > Usuario > Desktop > pokemon-app > .env
1 MONGO_INITDB_ROOT_USERNAME=admin
2 MONGO_INITDB_ROOT_PASSWORD=admin123
3 MONGO_PORT=27017
4 CONTAINER_NAME=miguel
5
```

Miguelillo

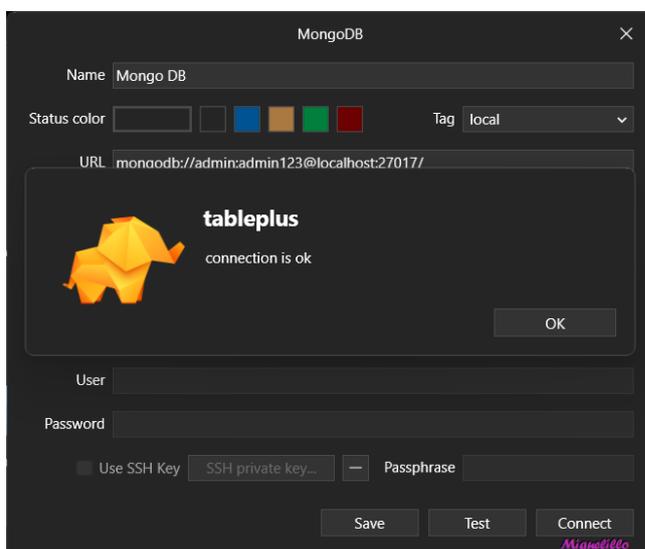
Editamos el **yml**



```
docker-compose.yml .env
C: > Users > Usuario > Desktop > pokemon-app > docker-compose.yml
1 services:
2   db:
3     container_name: ${CONTAINER_NAME}
4     image: mongo:6.0
5     command: mongod --auth
6     volumes:
7       - poke-vol:/data/db
8     ports:
9       - ${MONGO_PORT}:27017
10    restart: always
11
12    environment:
13      - MONGO_INITDB_ROOT_USERNAME=${MONGO_INITDB_ROOT_USERNAME}
14      - MONGO_INITDB_ROOT_PASSWORD=${MONGO_INITDB_ROOT_PASSWORD}
15  volumes:
16    poke-vol:
17      external: false
18
```

Miguelillo

Sigue funcionando correctamente



27) Añadimos la imagen de mongo-express

Solución:

```
mongo-express:
  image: mongo-express
  ports:
```

28) Añadir a la imagen más variables de entorno, puertos y always

Solución:

Configuramos el usuario contraseña, puertos, el always y el servidor de mongod

```
docker-compose.yml
services:
  db:
    container_name: ${CONTAINER_NAME}
    image: mongo:6.0
    command: mongod --auth
    volumes:
      - poke-vol:/data/db
    ports:
      - "${MONGO_PORT}:27017"
    restart: always
    environment:
      - MONGO_INITDB_ROOT_USERNAME=${MONGO_INITDB_ROOT_USERNAME}
      - MONGO_INITDB_ROOT_PASSWORD=${MONGO_INITDB_ROOT_PASSWORD}

  mongo-express:
    image: mongo-express
    ports:
      - "${MONGO_EXPRESS_PORT}:8081"
    restart: always
    environment:
      - ME_CONFIG_MONGODB_ADMINUSERNAME=${MONGO_INITDB_ROOT_USERNAME}
      - ME_CONFIG_MONGODB_ADMINPASSWORD=${MONGO_INITDB_ROOT_PASSWORD}
      - ME_CONFIG_MONGODB_SERVER=miguel
      - ME_CONFIG_BASICAUTH_USERNAME=${MONGO_EXPRESS_USERNAME}
      - ME_CONFIG_BASICAUTH_PASSWORD=${MONGO_EXPRESS_PASSWORD}

volumes:
  poke-vol:
    external: false
```

Configuramos las variables del .env

```
C: > Users > Usuario > Desktop > pokemon-app > .env
1 MONGO_INITDB_ROOT_USERNAME=admin
2 MONGO_INITDB_ROOT_PASSWORD=admin123
3 MONGO_PORT=27017
4 CONTAINER_NAME=miguel
5
6 MONGO_EXPRESS_PORT=8080
7 MONGO_EXPRESS_USERNAME=admin
8 MONGO_EXPRESS_PASSWORD=admin123
```

29) Levantamos el docker-compose

Solución:

Se levanta correctamente

```

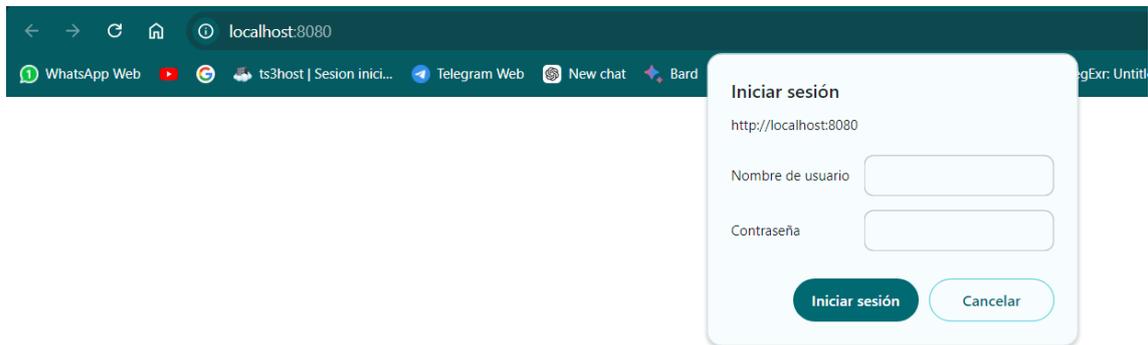
Administrador: Windows PowerShell
PS C:\users\usuario\desktop\pokemon-app> docker-compose down
[+] Running 2/2
  ✓ Container miguel          Removed
  ✓ Network pokemon-app_default Removed
PS C:\users\usuario\desktop\pokemon-app> docker-compose up -d
[+] Running 1/9
  mongo-express              Pulling
    619be1103602 Downloading [=====] 1.786MB/3.403MB 4.6s
    7e9a007eb24b Downloading [=====] 4.049MB/39.7MB 4.6s
  ✓ 5189255e31c8 Download complete 3.2s
    88f4f8a6bc8d Waiting 4.6s
    d8305ae32c95 Waiting 4.6s
    45b24ec126f9 Waiting 4.6s
    9f7f59574f7d Waiting 4.6s
    0bf3571b6cd7 Waiting 4.6s
  
```

```

Administrador: Windows PowerShell
PS C:\users\usuario\desktop\pokemon-app> docker-compose down
[+] Running 2/2
  ✓ Container miguel          Removed
  ✓ Network pokemon-app_default Removed
PS C:\users\usuario\desktop\pokemon-app> docker-compose up -d
[+] Running 9/9
  ✓ mongo-express Pulled
    619be1103602 Pull complete 57.9s
    7e9a007eb24b Pull complete 7.6s
    5189255e31c8 Pull complete 37.9s
    88f4f8a6bc8d Pull complete 38.2s
    d8305ae32c95 Pull complete 38.2s
    45b24ec126f9 Pull complete 38.4s
    9f7f59574f7d Pull complete 38.4s
    0bf3571b6cd7 Pull complete 47.8s
[+] Running 3/3
  ✓ Network pokemon-app_default Created 47.9s
  ✓ Container miguel          Started 0.1s
  ✓ Container pokemon-app-mongo-express-1 Started 4.0s
  
```

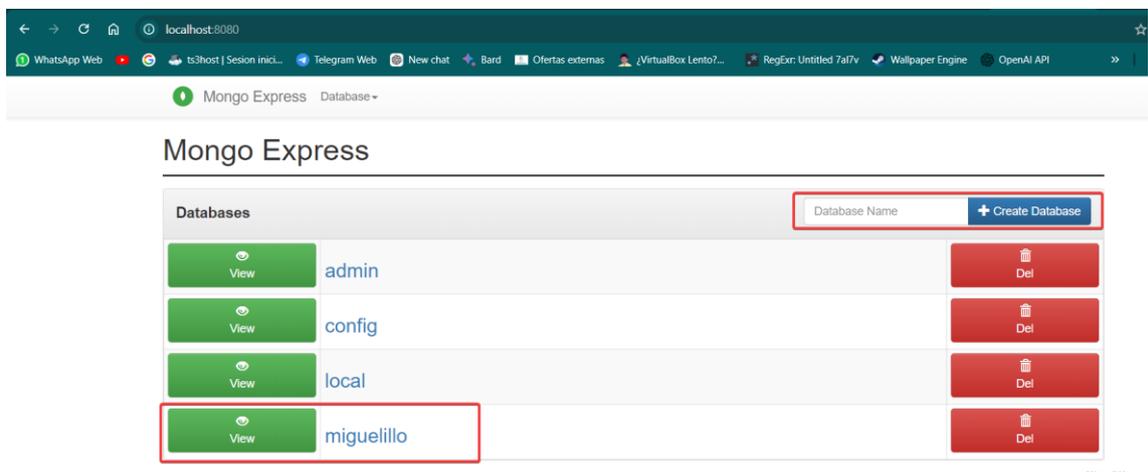
30) Probamos a acceder

Solución:



Miguelillo

Creamos la base de datos "miguelillo"

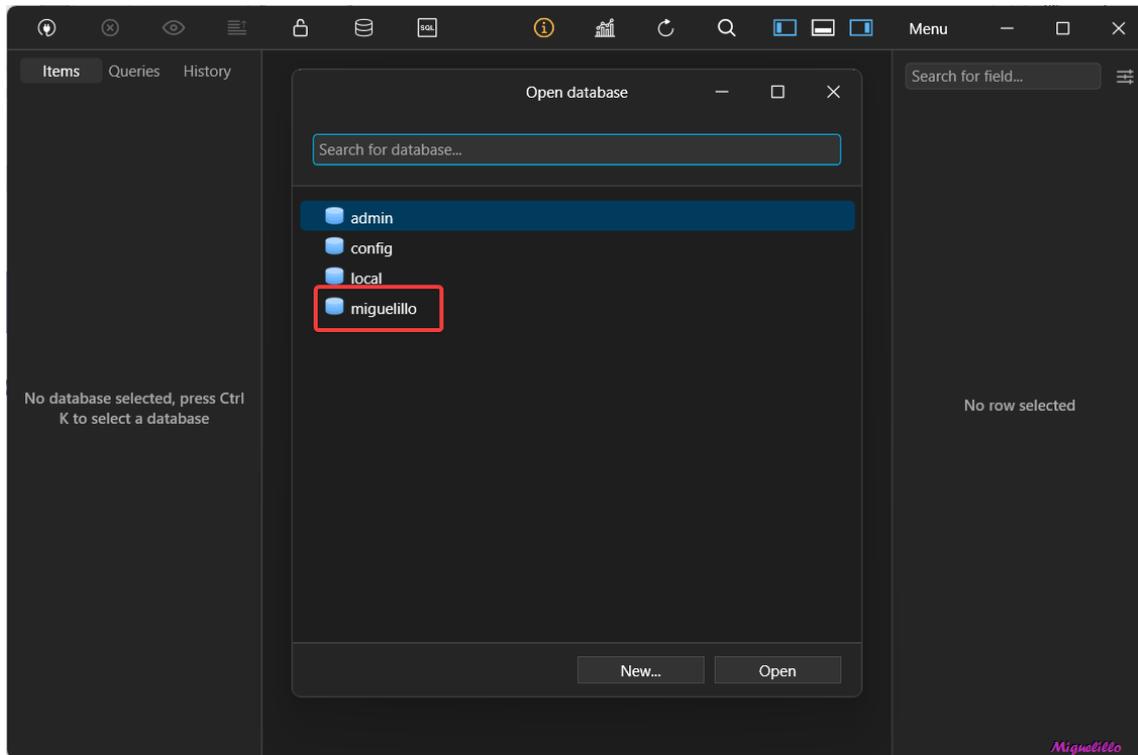


Miguelillo

31) Verificamos con table plus

Solución:

Se ve correctamente



32) Comentar los puertos de mongoDB

Solución:

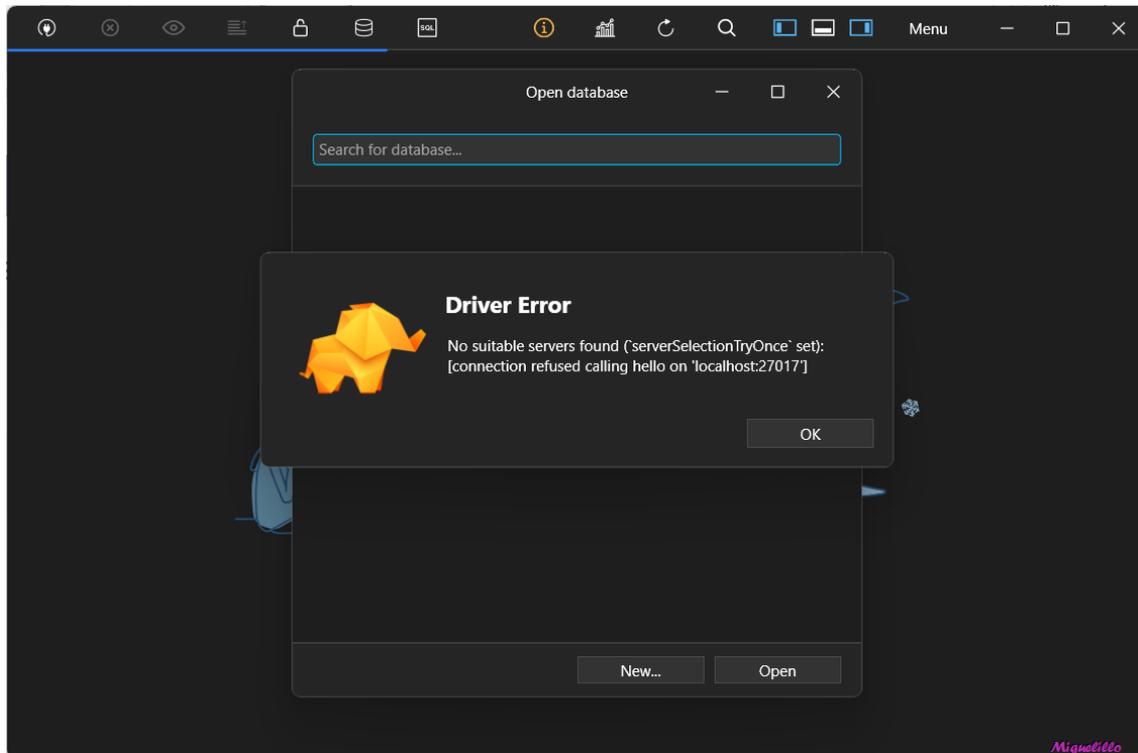
Los comento añadiendo un #

```
docker-compose.yml X .env
C: > Users > Usuario > Desktop > pokemon-app > docker-compose.yml
1  services:
2    db:
3      container_name: ${CONTAINER_NAME}
4      image: mongo:6.0
5      command: mongod --auth
6      volumes:
7        - poke-vol:/data/db
8      # ports:
9      # - "${MONGO_PORT}:27017"
10     restart: always
11     environment:
12       - MONGO_INITDB_ROOT_USERNAME=${MONGO_INITDB_ROOT_USERNAME}
13       - MONGO_INITDB_ROOT_PASSWORD=${MONGO_INITDB_ROOT_PASSWORD}
14
15     mongo-express:
16       image: mongo-express
17       ports:
18         - "${MONGO_EXPRESS_PORT}:8081"
19       restart: always
20       environment:
21         - ME_CONFIG_MONGODB_ADMINUSERNAME=${MONGO_INITDB_ROOT_USERNAME}
22         - ME_CONFIG_MONGODB_ADMINPASSWORD=${MONGO_INITDB_ROOT_PASSWORD}
23         - ME_CONFIG_MONGODB_SERVER=miguel
24         - ME_CONFIG_BASICAUTH_USERNAME=${MONGO_EXPRESS_USERNAME}
25         - ME_CONFIG_BASICAUTH_PASSWORD=${MONGO_EXPRESS_PASSWORD}
26
27     volumes:
28       poke-vol:
29         external: false
30
```

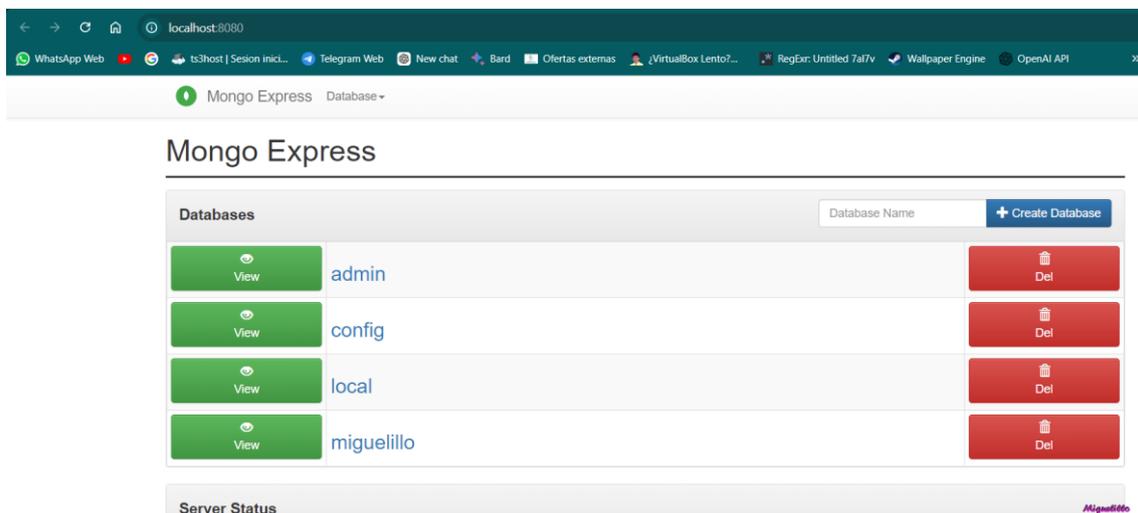
Miguelillo

33) Probamos que sea inaccesible

Solución:



Mongo Express es accesible



34) Descargar la imagen Pokemon-Nest-App

Solución:

Lo descargamos con `docker pull klerith/pokemon-nest-app`

```
PS C:\users\usuario\desktop\pokemon-app> docker pull klerith/pokemon-nest-app
Using default tag: latest
latest: Pulling from klerith/pokemon-nest-app
9621f1afde84: Pull complete
b2ff27170c03: Pull complete
857f24243633: Pull complete
f5234ba59f34: Pull complete
e09f54a0e125: Pull complete
08fdb445aaff: Pull complete
931c4e4daab4: Pull complete
4c3cfc937a62: Pull complete
Digest: sha256:ee109396125120312e886cd05cf0c4c871c26d6d7044bd1d1c7ec3a712fb26cf
Status: Downloaded newer image for klerith/pokemon-nest-app:latest
docker.io/klerith/pokemon-nest-app:latest

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview klerith/pokemon-nest-app
PS C:\users\usuario\desktop\pokemon-app> Miguelillo
```

35) Configurar la imagen, el depends, los puertos y always Solución:

Configuramos todo

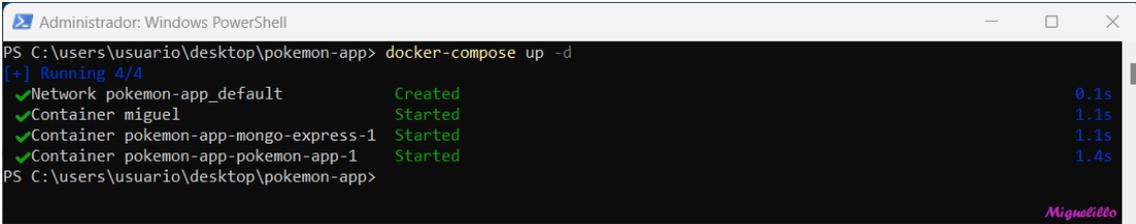
```
25 |   pokemon-nest-app:
26 |     container_name: pokemon-nest-app
27 |     image: klerith/pokemon-nest-app
28 |     depends_on:
29 |       - db
30 |       - mongo-express
31 |     ports:
32 |       - "3000:3000"
33 |     restart: always
34 |     environment:
35 |       - MONGODB=mongodb://admin:admin123@miguel:27017
36 |
37 |   volumes:
38 |     poke-vol:
39 |       external: false
```

Miguelillo

36) Probamos el docker-compose

Solución:

Se crea correctamente



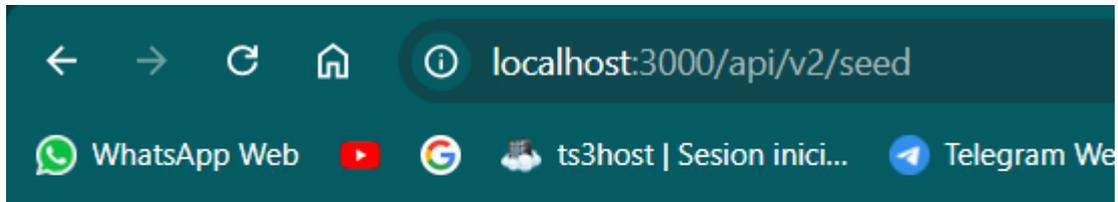
```
Administrador: Windows PowerShell
PS C:\users\usuario\desktop\pokemon-app> docker-compose up -d
[+] Running 4/4
✔ Network pokemon-app_default      Created      0.1s
✔ Container miguel                 Started      1.1s
✔ Container pokemon-app-mongo-express-1 Started      1.1s
✔ Container pokemon-app-pokemon-app-1 Started      1.4s
PS C:\users\usuario\desktop\pokemon-app>
```

Miguelillo

37) Para ejecutar la semilla

Solución:

Ponemos esta URL



Seed Executed

Miguelillo

38) Acceder y ver las tablas

Solución:

Vamos al mongo express y descargamos el json de la base de datos, vemos que contiene 650 Objetos

The screenshot shows the Mongo Express web interface. At the top, the URL is localhost:8080/db/pokemon-db/. Below the navigation bar, the database name 'pokemon-db' is displayed. The main area is titled 'Viewing Database: pokemon-db'. Under 'Collections', the 'pokemons' collection is highlighted with a red box, and its '[JSON]' button is also highlighted. Below this, the 'Database Stats' table is shown with the following data:

Database Stats	
Collections (incl. system.namespaces)	1
Data Size	37.4 KB
Storage Size	4.10 KB
Avg Obj Size #	57.5 Bytes
Objects #	650
Indexes #	3

Si abrimos el JSON podemos ver los distintos registros

The screenshot shows a code editor with the file 'pokemons.json' open. The JSON content is displayed, showing an array of 44 objects representing different Pokémon. The first object is highlighted with a red box:

```

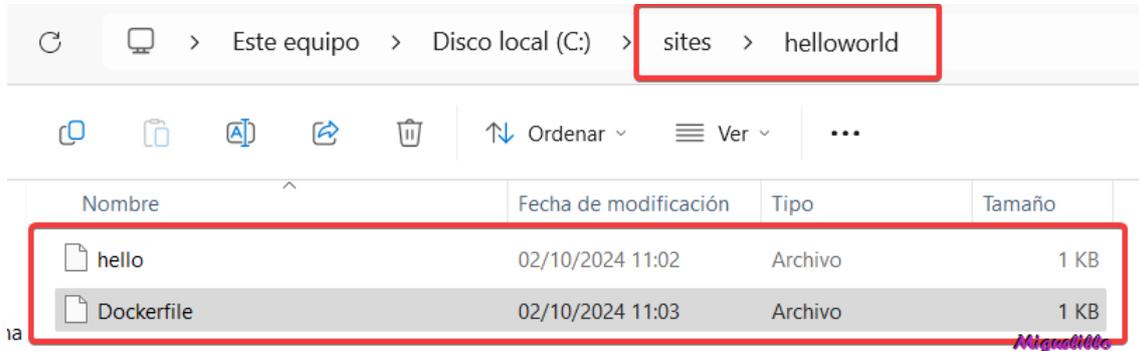
1 [{"_id":{"oid":"66fd05d3cc84b8435e5a2c2f"},"name":"bulbasaur","no":1,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c30"},"name":"ivysaur","no":2,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c31"},"name":"venusaur","no":3,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c32"},"name":"charmander","no":4,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c33"},"name":"charmeleon","no":5,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c34"},"name":"charizard","no":6,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c35"},"name":"squirtle","no":7,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c36"},"name":"wartortle","no":8,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c37"},"name":"blastoise","no":9,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c38"},"name":"caterpie","no":10,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c39"},"name":"metapod","no":11,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c3a"},"name":"butterfree","no":12,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c3b"},"name":"weedle","no":13,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c3c"},"name":"kakuna","no":14,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c3d"},"name":"beedrill","no":15,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c3e"},"name":"pidgey","no":16,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c3f"},"name":"pidgeotto","no":17,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c40"},"name":"pidgeot","no":18,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c41"},"name":"rattata","no":19,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c42"},"name":"raticate","no":20,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c43"},"name":"spearow","no":21,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c44"},"name":"fearow","no":22,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c45"},"name":"ekans","no":23,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c46"},"name":"arbok","no":24,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c47"},"name":"pikachu","no":25,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c48"},"name":"raichu","no":26,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c49"},"name":"sandshrew","no":27,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c4a"},"name":"sandslash","no":28,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c4b"},"name":"nidoran-f","no":29,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c4c"},"name":"nidorina","no":30,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c4d"},"name":"nidoqueen","no":31,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c4e"},"name":"nidoran-m","no":32,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c4f"},"name":"nidorino","no":33,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c50"},"name":"nidoking","no":34,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c51"},"name":"clefairy","no":35,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c52"},"name":"clefable","no":36,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c53"},"name":"vulpix","no":37,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c54"},"name":"ninetales","no":38,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c55"},"name":"jigglypuff","no":39,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c56"},"name":"wigglytuff","no":40,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c57"},"name":"zubat","no":41,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c58"},"name":"golbat","no":42,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c59"},"name":"oddish","no":43,"_v":0}, {"_id":{"oid":"66fd05d3cc84b8435e5a2c5a"},"name":"gloom","no":44,"_v":0}

```

Ejercicio 4

39) Creamos el bulid context y archivos

Solución:



Le añadimos contenido

```
Dockerfile X
C: > sites > helloworld > Dockerfile
1 FROM busybox
2 COPY /hello /
3 RUN cat /hello
```

Miguelillo

40) Crearemos la imagen

Solución:

Creamos la imagen con `docker build -t helloapp:v1` .

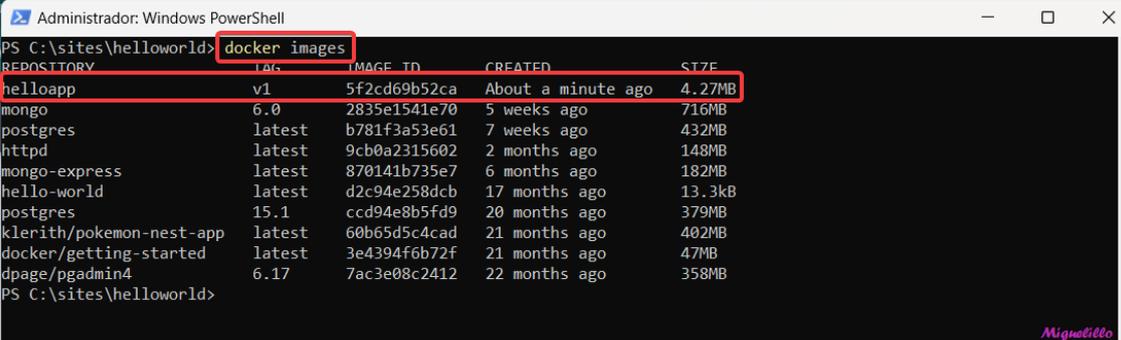
```
PS C:\sites\helloworld> docker build -t helloapp:v1.
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage: docker buildx build [OPTIONS] PATH | URL | -

Start a build
PS C:\sites\helloworld> docker build -t helloapp:v1 .
[+] Building 4.5s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.1s
=> => transferring dockerfile: 80B                             0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 2.6s
=> [auth] library/busybox:pull token for registry-1.docker.io   0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                   0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 37B                                  0.0s
=> [1/3] FROM docker.io/library/busybox:latest@sha256:768e5c6f5cb6db0794eec98dc7a967f40631746c32232b78a3105fb946 0.8s
=> => resolve docker.io/library/busybox:latest@sha256:768e5c6f5cb6db0794eec98dc7a967f40631746c32232b78a3105fb946 0.0s
=> => sha256:768e5c6f5cb6db0794eec98dc7a967f40631746c32232b78a3105fb946f3ab83 10.20kB / 10.20kB 0.0s
=> => sha256:22f27168517de1f58dae0ad51eacf1527e7e7ccc47512d3946f56bdbc913f564 610B / 610B 0.0s
=> => sha256:27a71e19c95622dce4d60d4a3760707495c9875f5c5322c5bd535214799593ce 372B / 372B 0.0s
=> => sha256:a46fbb00284bdd1a1d8d80d51333abc851371a7b8d44cc781c4469b5e54119ae 2.16MB / 2.16MB 0.4s
=> => extracting sha256:a46fbb00284bdd1a1d8d80d51333abc851371a7b8d44cc781c4469b5e54119ae 0.2s
=> [2/3] COPY /hello /                                         0.1s
=> [3/3] RUN cat /hello                                         0.5s
=> exporting to image                                           0.1s
=> => exporting layers                                          0.1s
=> => writing image sha256:5f2cd69b52ca827ba33d1802bf804db4debe110ad2bf5908818f6c2c73ff9701 0.0s
=> => naming to docker.io/library/helloapp:v1                  0.0s

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\sites\helloworld>
```

Vemos las imágenes instalada



```
Administrador: Windows PowerShell
PS C:\sites\helloworld> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
helloapp            v1                 5f2cd69b52ca       About a minute ago 4.27MB
mongo               6.0                2835e1541e70       5 weeks ago        716MB
postgres            latest             b781f3a53e61       7 weeks ago        432MB
httpd               latest             9cb0a2315602       2 months ago       148MB
mongo-express       latest             870141b735e7       6 months ago       182MB
hello-world         latest             d2c94e258dcb       17 months ago      13.3kB
postgres            15.1              ccd94e8b5fd9       20 months ago      379MB
klerith/pokemon-nest-app latest             60b65d5c4cad       21 months ago      402MB
docker/getting-started latest             3e4394f6b72f       21 months ago      47MB
dpape/pgadmin4     6.17              7ac3e08c2412       22 months ago      358MB
PS C:\sites\helloworld>
```

41) Creando aplicaciones en contenedores

Solución:

Creamos la carpeta friendlyhello y dentro creamos app.py, le ponemos el siguiente contenido

```

C:\> sites > friendlyhello > app.py
1  from flask import Flask
2  from redis import Redis, RedisError
3  import os
4  import socket
5  Y por último definimos nuestro Dockerfile:
6  Para conocer todas las directivas visita la documentación oficial de Dockerfile.
7  En total debemos tener 3 archivos:
8  # Connect to Redis
9  redis = Redis(host="redis", db=0, socket_connect_timeout=2, socket_timeout=2)
10 app = Flask(__name__)
11 @app.route("/")
12 def hello():
13     try:
14         visits = redis.incr("counter")
15     except RedisError:
16         visits = "<i>cannot connect to Redis, counter disabled</i>"
17     html = "<h3>Hello {name}!</h3>" \
18           "<b>Hostname:</b> {hostname}<br/>" \
19           "<b>Visits:</b> {visits}"
20     return html.format(name=os.getenv("NAME", "world"),
21                       hostname=socket.gethostname(), visits=visits)
22 if __name__ == "__main__":
23     app.run(host='0.0.0.0', port=80)

```

Creamos el archivo requirements.txt

```

requirements.txt
Archivo  Editar  Ver
Flask
Redis

```

Creamos el archivo Dockerfile

```

C:\> sites > friendlyhello > Dockerfile
1  # Partimos de una base oficial de python
2  FROM python:2.7-slim
3
4  # El directorio de trabajo es desde donde se ejecuta el contenedor al iniciarse
5  WORKDIR /app
6
7  # Copiamos todos los archivos del build context al directorio /app del contenedor
8  COPY . /app
9
10 # Ejecutamos pip para instalar las dependencias en el contenedor
11 RUN pip install --trusted-host pypi.python.org -r requirements.txt
12
13 # Indicamos que este contenedor se comunica por el puerto 80/tcp
14 EXPOSE 80
15
16 # Declaramos una variable de entorno
17 ENV NAME World
18
19 # Ejecuta nuestra aplicación cuando se inicia el contenedor
20 CMD ["python", "app.py"]

```

Construimos la imagen de nuestra aplicación

```

Administrador: Windows PowerShell
PS C:\sites\friendlyhello> docker build -t friendlyhello .
[+] Building 2.3s (2/3)                                docker:desktop-linux
=> [internal] load build definition from Dockerfile    0.0s
=> => transferring dockerfile: 657B                  0.0s
=> [internal] load metadata for docker.io/library/python:2.7-slim 2.3s
=> [auth] library/python:pull token for registry-1.docker.io 0.0s

```

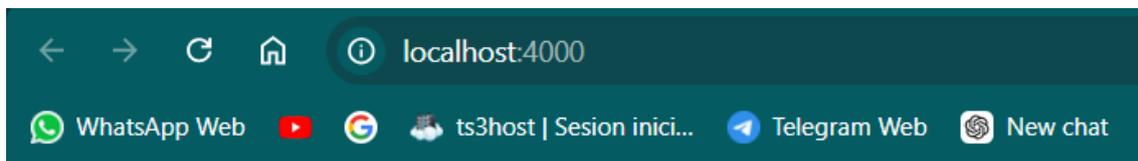
Comprobamos que se ha creado

```

Administrador: Windows PowerShell
PS C:\sites\friendlyhello> docker image ls
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
friendlyhello      latest      82d03e6508d7     About a minute ago 159MB
helloapp           v1         5f2cd69b52ca     About an hour ago 4.27MB
mongo              6.0        2835e1541e70     5 weeks ago      716MB
postgres           latest     b781f3a53e61     7 weeks ago      432MB
http               latest     9cb0a2315602     2 months ago    148MB
mongo-express      latest     870141b735e7     6 months ago    182MB
hello-world        latest     d2c94e258dcb     17 months ago   13.3kB
postgres           15.1      ccd94e8b5fd9     20 months ago   379MB
klerith/pokemon-nest-app latest     60b65d5c4cad     21 months ago   402MB
docker/getting-started latest     3e4394f6b72f     21 months ago   47MB
dpage/pgadmin4    6.17      7ac3e08c2412     22 months ago   358MB
PS C:\sites\friendlyhello>

```

Probamos la imagen corre correctamente



Hello World!

Hostname: 1e35fabda403

Visits: cannot connect to Redis, counter disabled

Miguelillo

Hacemos un docker-compose

```

docker-compose.yml
C: > sites > friendlyhello > docker-compose.yml
1  services:
2    web:
3      build: .
4      ports:
5        - "4000:80"
6    redis:
7      image: redis
8      ports:
9        - "6379:6379"
10     volumes:
11       - "./data:/data"
12     command: redis-server --appendonly yes

```

Vemos como se crea

```

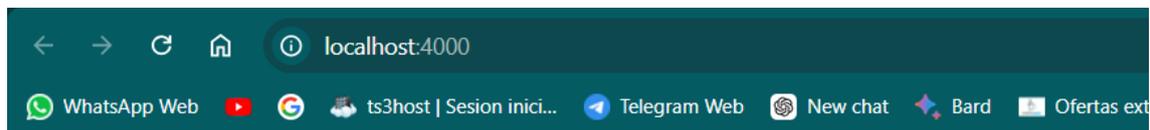
Administrador: Windows PowerShell
PS C:\sites\friendlyhello> docker-compose up -d
[+] Running 7/9
 - redis [.....] 34.73MB / 45.88MB Pulling
   302e3ee49805 Downloading [----->]...
   5d0249d9189d Download complete
   4825c5e95815 Download complete
   b0ce50685fa2 Download complete
   455886c7d31b Download complete
   96377887d476 Download complete
   4f4fb700ef54 Download complete
   5fac73c23c9b Download complete
10.5s
7.7s
1.0s
0.7s
2.8s
5.6s
5.0s
5.7s
6.2s
Miguellillo

```

Vemos como se ha creado

<input type="checkbox"/>	 friendlyhello		Running (2/2)	0.48%	2 minutes ago	<input type="checkbox"/>	:		
<input type="checkbox"/>	 web-1 7beb822fc344	friendlyhello-web	Running	4000:80	0.02%	2 minutes ago	<input type="checkbox"/>	:	
<input type="checkbox"/>	 redis-1 cfa96f3593d9	redis	Running	6379:6379	0.46%	2 minutes ago	<input type="checkbox"/>	:	

Lo probamos



Hello World!

Hostname: 7beb822fc344

Visits: 6

Miguellillo

42) Balanceo de carga

Solución:

Actualizamos el yaml anterior, cambiando el ports por un lb (un balanceador de carga)

```

docker-compose.yml
C: > sites > friendlyhello > docker-compose.yml
1  services:
2    web:
3      build: .
4    redis:
5      image: redis
6      volumes:
7        - ./data:/data
8      command: redis-server --appendonly yes
9    lb:
10     image: dockercloud/haproxy
11     ports:
12       - 4000:80
13     links:
14       - web
15     volumes:
16       - /var/run/docker.sock:/var/run/docker.sock

```

Arrancamos la aplicación

```

PS C:\sites\friendlyhello> docker-compose up -d --scale web=5
[+] Running 2/4
- lb [#####] 6.399MB / 16.38MB Pulling 10.5s
  ✓ 1160f4abead8 Pull complete 3.8s
  ✓ b0df9c632afc Pull complete 3.9s
  - a49b18c7cd3a Downloading [=====] 3.98... 7.1s

```

Esperamos a que se inicien todos los servicios

```

PS C:\sites\friendlyhello> docker-compose up -d --scale web=5
[+] Running 4/4
  ✓ lb Pulled 21.0s
  ✓ 1160f4abead8 Pull complete 3.8s
  ✓ b0df9c632afc Pull complete 3.9s
  ✓ a49b18c7cd3a Pull complete 17.6s
[+] Running 7/7
  ✓ Container friendlyhello-redis-1 Started 1.0s
  ✓ Container friendlyhello-web-5 Started 2.8s
  ✓ Container friendlyhello-web-2 Started 1.6s
  ✓ Container friendlyhello-web-1 Started 1.2s
  ✓ Container friendlyhello-web-4 Started 2.1s
  ✓ Container friendlyhello-web-3 Started 2.5s
  ✓ Container friendlyhello-lb-1 Started 2.8s
PS C:\sites\friendlyhello>

```

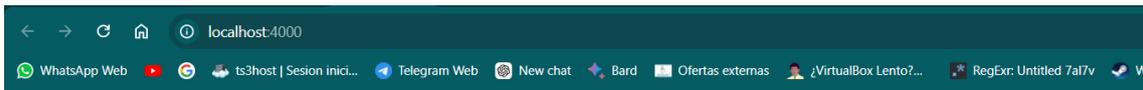
Vemos que se inician todos los servicios

```

Administrator: Windows PowerShell
PS C:\sites\friendlyhello> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
3f9432516979   dockercloud/haproxy                "/sbin/tini -- docke..." 2 minutes ago Up 2 minutes  443/tcp, 1936/tcp, 0.0.0.0:
4800->80/tcp
friendlyhello-lb-1
ddd65f20aafc   friendlyhello-web                   "python app.py"          2 minutes ago Up 2 minutes  80/tcp
friendlyhello-web-1
444e2ee076e8   friendlyhello-web                   "python app.py"          2 minutes ago Up 2 minutes  80/tcp
friendlyhello-web-2
48805059bf0a   friendlyhello-web                   "python app.py"          2 minutes ago Up 2 minutes  80/tcp
friendlyhello-web-4
f1e36d82268b   friendlyhello-web                   "python app.py"          2 minutes ago Up 2 minutes  80/tcp
friendlyhello-web-3
fce35aa8281b   redis                                "docker-entrypoint.s..." 2 minutes ago Up 2 minutes  6379/tcp
friendlyhello-redis-1
5da01b6c695b   friendlyhello-web                   "python app.py"          2 minutes ago Up 2 minutes  80/tcp
friendlyhello-web-5
PS C:\sites\friendlyhello>

```

No funciona tiene un fallo



Esta página no funciona

localhost no ha enviado ningún dato.

ERR_EMPTY_RESPONSE

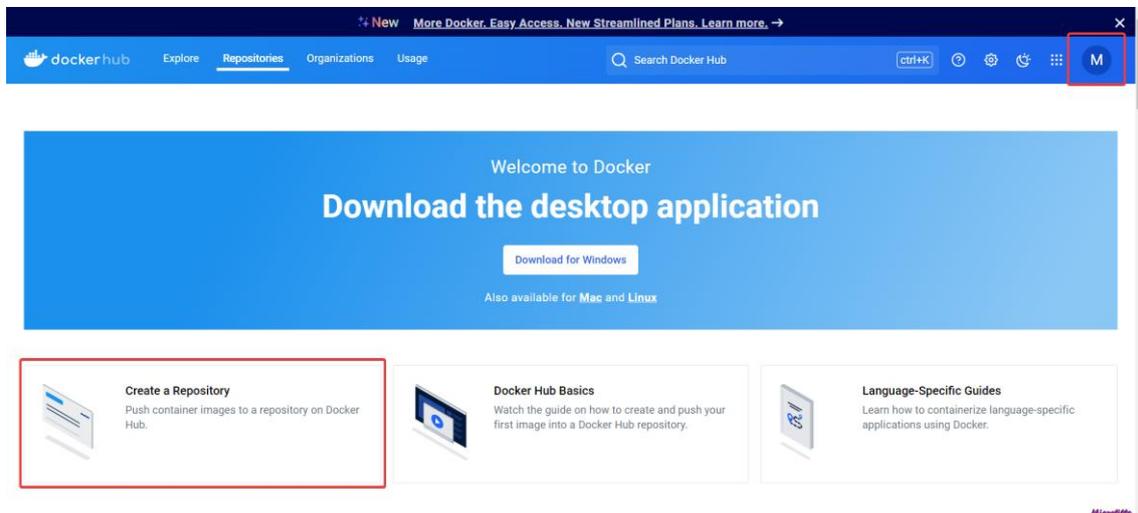
Volver a cargar

Miguelillo

43) Compartir imágenes

Solución:

Nos creamos una cuenta y clicamos en **Create a Repository**



Le ponemos un nombre y descripción

Create repository

Namespace
miguellillo89

Repository Name *
friendlyhello

Short description
 practica sri miguellillo

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 1 of 1 private repositories. [Get more](#)

Public
 Appears in Docker Hub search results

Private
 Only visible to you

Cancel

Create

Pushing images

You can push a new image to t

```
docker tag local-image
docker push new-repo:t
```

Make sure to replace tagnar

Iniciamos sesión en dockers

```

Administrador: Windows PowerShell
PS C:\sites\friendlyhello> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\sites\friendlyhello>
  
```

Subimos el dockerfile

```

Administrador: Windows PowerShell
PS C:\sites\friendlyhello> docker build -t miguelillo89/friendlyhello .
[+] Building 7.8s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 657B
=> [internal] load metadata for docker.io/library/python:2.7-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/python:2.7-slim@sha256:6c1ffdf499e29ea663e6e67c9b6b9a3b401d554d2c9f061f9a45344e3992363
=> [internal] load build context
=> => transferring context: 1.85kB
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install --trusted-host pypi.python.org -r requirements.txt
=> exporting image
=> => exporting layers
=> => writing image sha256:4f4b402443c0e2f35478ce0304d7778514d43e37cd0182702f63fec8a789787
=> => naming to docker.io/miguelillo89/friendlyhello

1 warning found (use docker --debug to expand):
- LegacyKeyValueFormat: "ENV key=value" should be used instead of legacy "ENV key value" format (line 17)

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\sites\friendlyhello>

```

Lo subimos

```

Administrador: Windows PowerShell
PS C:\sites\friendlyhello> docker push miguelillo89/friendlyhello
Using default tag: latest
The push refers to repository [docker.io/miguelillo89/friendlyhello]
590fd09c3686: Pushed
888614b7a008: Pushed
fc85689e8d13: Pushed
7a287aad297b: Mounted from library/python
7ea2b60b0a08: Mounted from library/python
568944187d93: Mounted from library/python
b60e5c3bcef2: Mounted from library/python
latest: digest: sha256:9bb7e48760d7c7721d6e807890489074d715f65e2eafac3810232fc48a012702 size: 1788
PS C:\sites\friendlyhello>

```

Probamos a usar la imagen

```

Administrador: Windows PowerShell
PS C:\sites\friendlyhello> docker run -d --name friendlyhello_container miguelillo89/friendlyhello:latest
0ead4b46a68ebba0ed957758938a9e8bf87c1479d29d6fca90565177bb7ff96f
PS C:\sites\friendlyhello>

```

```

docker-compose.yml
C:\sites> friendlyhello > docker-compose.yml
1  services:
2    web:
3      image: sergio1116/friendlyhello:latest
4    redis:
5      image: redis
6      volumes:
7        - ./data:/data
8      command: redis-server --appendonly yes
9    lb:
10     image: dockercloud/haproxy
11     ports:
12       - "4000:80"
13     links:
14       - web
15     volumes:
16       - /var/run/docker.sock:/var/run/docker.sock

```

```
docker-compose.yml ●
C: > sites > friendlyhello > docker-compose.yml
1  services:
2    web:
3      image: miguelillo89/friendlyhello:latest
4    redis:
5      image: redis
6      volumes:
7        - "./data:/data"
8      command: redis-server --appendonly yes
9    lb:
10     image: dockercloud/haproxy
11     ports:
12       - "4000:80"
13     links:
14       - web
15     volumes:
16       - /var/run/docker.sock:/var/run/docker.sock
```

Miguelillo

Ejercicio 5

44) Instalamos NGINX

Solución:

Creamos la estructura de directorios para el ejercicio

```
miguelillo@miguelilloSRI:~/practica62$ ls -r
www php nginx docker-compose.yml
miguelillo@miguelilloSRI:~/practica62$ ls www/
html
miguelillo@miguelilloSRI:~/practica62$ ls www/html/
index.php
miguelillo@miguelilloSRI:~/practica62$ ls php/
Dockerfile
miguelillo@miguelilloSRI:~/practica62$ ls nginx/
default.conf Dockerfile
```

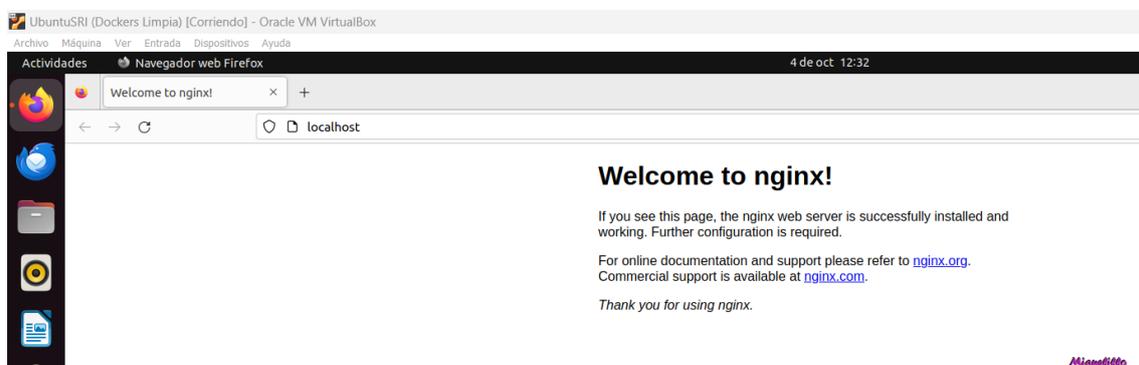
Miguelillo

```
Abrir [+] docker-compose.yml /home/miguelillo/practica62 Guardar [≡] [–] [□] [×]
1 services:
2   nginx:
3     image: nginx:latest
4     container_name: nginx-container
5     ports:
6     - 80:80
```

Miguelillo

```
miguelillo@miguelilloSRI: ~/practica62 [🔍] [≡] [–] [□] [×]
miguelillo@miguelilloSRI:~/practica62$ docker-compose up -d
Creating network "practica62_default" with the default driver
Pulling nginx (nginx:latest)...
latest: Pulling from library/nginx
302e3ee49805: Already exists
d07412f52e9d: Pull complete
9ab66c386e9c: Pull complete
4b563e5e980a: Pull complete
55af3c8feb2: Pull complete
5b8e768fb22d: Pull complete
85177e2c6f39: Pull complete
Digest: sha256:d2eb56950b84efe34f966a2b92efb1a1a2ea53e7e93b94cdf45a27cf3cd47fc0
Status: Downloaded newer image for nginx:latest
Creating nginx-container ... done
```

Miguelillo



Miguelillo

45) Creación contenedor PHP

Solución:

Dentro de index.php añadimos el siguiente código

```

GNU nano 6.2 practica62/www/html/index.php
<!DOCTYPE html>
<head>
<title>;Hola mundo!</title>
</head>
<body>
<h1>;Hola mundo!</h1>
<p><?php echo 'Estamos corriendo PHP, version: ' . phpversion(); ?></p>
</body>
[ El fichero «practica62/www/html/index.php» no es de escritura ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^_ Reemplazar  ^U Pegar      ^J Justificar ^/ Ir
  
```

Modificamos el default.conf

```

1 server {
2     listen 80 default_server;
3     root /var/www/html;
4     index index.html index.php;
5     charset utf-8;
6     location / {
7         try_files $uri $uri/ /index.php?$query_string;
8     }
9     location = /favicon.ico { access_log off; log_not_found off; }
10    location = /robots.txt { access_log off; log_not_found off; }
11    access_log off;
12    error_log /var/log/nginx/error.log error;
13    sendfile off;
14    client_max_body_size 100m;
15    location ~ .php$ {
16        fastcgi_split_path_info ^(.+\.php)(/.+)$;
17        fastcgi_pass php:9000;
18        fastcgi_index index.php;
19        include fastcgi_params;
20        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
21        fastcgi_intercept_errors off;
22        fastcgi_buffer_size 16k;
23        fastcgi_buffers 4 16k;
24    }
25    location ~ /\.ht {
26        deny all;
27    }
28 }
  
```

Añadimos el contenido al dockerfile

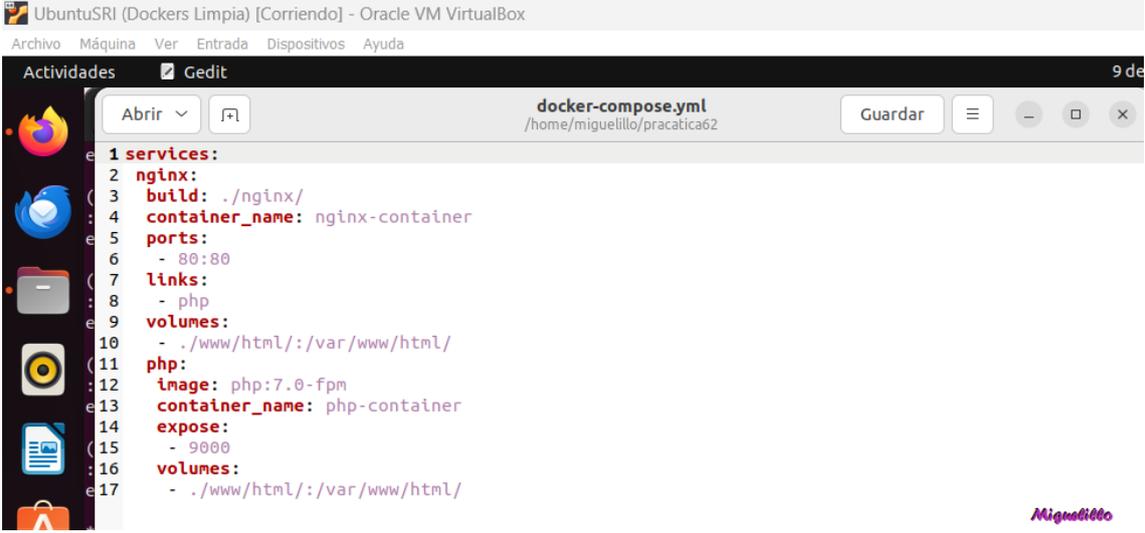


```

1 FROM nginx:latest
2 COPY ./default.conf /etc/nginx/conf.d/default.conf
3

```

Modificamos el docker-compose

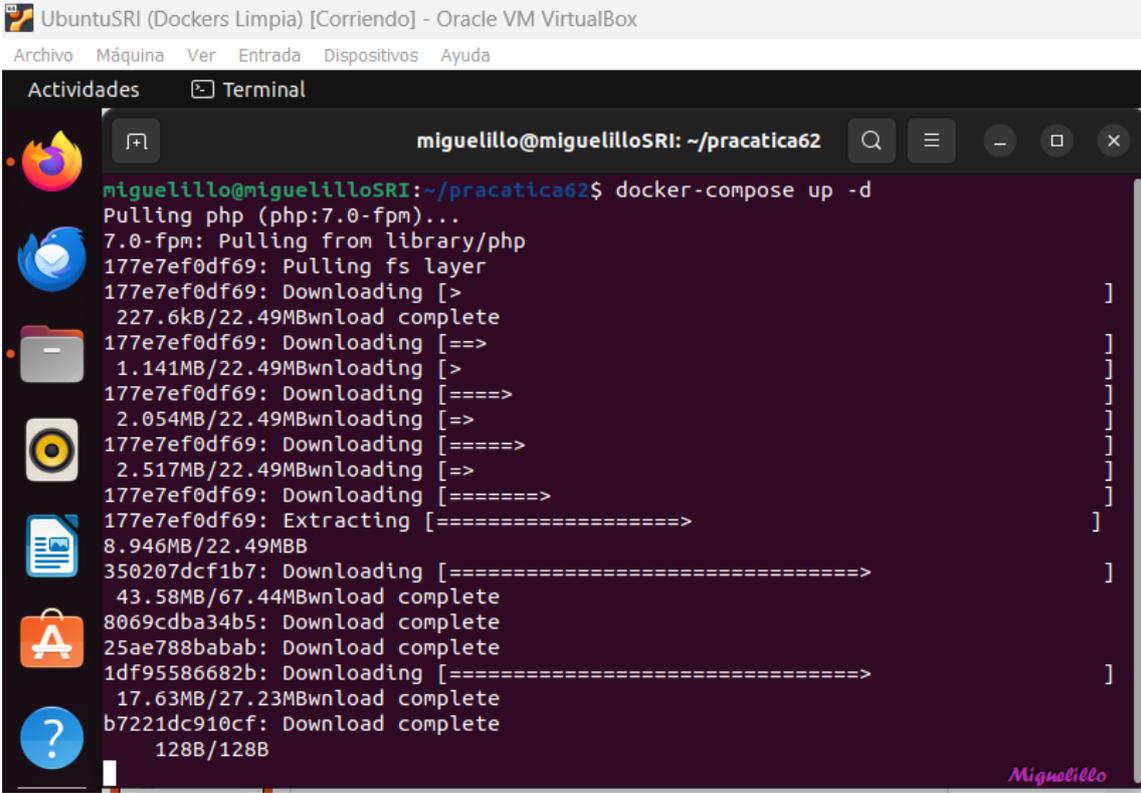


```

1 services:
2   nginx:
3     build: ./nginx/
4     container_name: nginx-container
5     ports:
6       - 80:80
7     links:
8       - php
9     volumes:
10      - ./www/html:/var/www/html/
11   php:
12     image: php:7.0-fpm
13     container_name: php-container
14     expose:
15       - 9000
16     volumes:
17       - ./www/html:/var/www/html/

```

Lo ejecutamos



```

miguellillo@miguellilloSRI: ~/practica62
miguellillo@miguellilloSRI:~/practica62$ docker-compose up -d
Pulling php (php:7.0-fpm)...
7.0-fpm: Pulling from library/php
177e7ef0df69: Pulling fs layer
177e7ef0df69: Downloading [> ]
 227.6kB/22.49MBwnload complete
177e7ef0df69: Downloading [==> ]
 1.141MB/22.49MBwnloading [> ]
177e7ef0df69: Downloading [====> ]
 2.054MB/22.49MBwnloading [=> ]
177e7ef0df69: Downloading [=====> ]
 2.517MB/22.49MBwnloading [==> ]
177e7ef0df69: Downloading [=====> ]
177e7ef0df69: Extracting [=====> ]
 8.946MB/22.49MBB
350207dcf1b7: Downloading [=====> ]
 43.58MB/67.44MBwnload complete
8069cdba34b5: Download complete
25ae788babab: Download complete
1df95586682b: Downloading [=====> ]
 17.63MB/27.23MBwnload complete
b7221dc910cf: Download complete
 128B/128B

```

Se ha creado y está corriendo

```
miguellillo@miguellilloSRI:~/practicas6$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
0812aafda69c   php:7.0-fpm   "docker-php-entrypoi..." 14 minutes ago Up 18 seconds 9000/tcp                php-container
39781c13b4dd   nginx:latest  "/docker-entrypoint..." 4 days ago    Up 18 seconds 0.0.0.0:80->80/tcp      39781c13b4dd_nginx-container
miguellillo@miguellilloSRI:~/practicas6$
```

46) Creación de un contenedor para datos

Solución:

Modificamos el docker-compose

```

docker-compose.yml
/home/miguelillo/pracatica62

1 services:
2   nginx:
3     build: ./nginx/
4     container_name: nginx-container
5     ports:
6       - 80:80
7     links:
8       - php
9     volumes:
10      - ./www/html:/var/www/html/
11  php:
12    image: php:7.0-fpm
13    container_name: php-container
14    expose:
15      - 9000
16    volumes:
17      - ./www/html:/var/www/html/
18  app-data:
19    image: php:7.0-fpm
20    container_name: app-data-container
21    volumes:
22      - ./www/html:/var/www/html/
23    command: "true"

```

Probamos que se cree

```

Terminal
9 de oct 11:54
miguelillo@miguelilloSRI: ~/pracatica62

miguelillo@miguelilloSRI:~/pracatica62$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
0812aafda69c   php:7.0-fpm   "/docker-php-entrypoi..." 56 minutes ago Up 42 minutes 9000/tcp                 php-container
39781c13b4dd   nginx:latest  "/docker-entrypoint..." 4 days ago    Up 42 minutes 0.0.0.0:80->80/tcp       39781c13b4dd_nginx-container
fc0c0f6bc484   httpd         "httpd-foreground"       5 days ago    Exited(0) 4 days ago                                miguel
ccf20e8105a0   hello-world   "/hello"                  5 days ago    Exited(0) 5 days ago                                interesting_ferni

```

UbuntuSRI (Dockers Limpia) [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Navegador web Firefox

¡Hola mundo!

Estamos corriendo PHP, version: 7.0.33

47) Creación de contenedor MySQL

Solución:

Añadimos información al Dockerfile

```

GNU nano 6.2
FROM php:7.0-fpm
RUN docker-php-ext-install pdo_mysql
  
```

Modificamos el docker compose

```

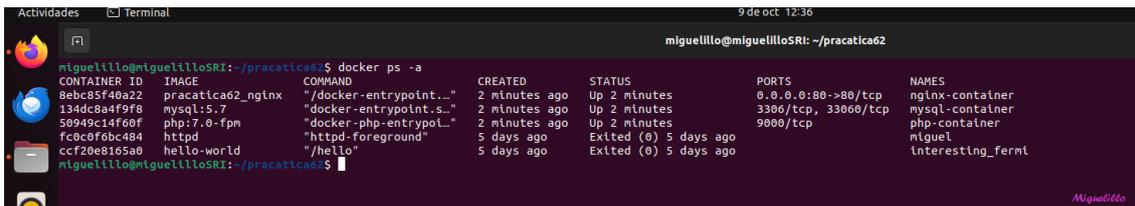
1 services:
2   nginx:
3     build: ./nginx/
4     container_name: nginx-container
5     ports:
6       - 80:80
7     links:
8       - php
9     volumes:
10      - ./www/html:/var/www/html/
11   php:
12     image: php:7.0-fpm
13     container_name: php-container
14     expose:
15       - 9000
16     volumes:
17       - ./www/html:/var/www/html/
18   app-data:
19     image: php:7.0-fpm
20     container_name: app-data-container
21     volumes:
22       - ./www/html:/var/www/html/
23     command: "true"
24   mysql:
25     image: mysql:5.7
26     container_name: mysql-container
27     volumes_from:
28       - mysql-data
29     environment:
30       MYSQL_ROOT_PASSWORD: secret
31       MYSQL_DATABASE: mydb
32       MYSQL_USER: myuser
33       MYSQL_PASSWORD: password
34   mysql-data:
35     image: mysql:5.7
36     container_name: mysql-data-container
37     volumes:
38       - /var/lib/mysql
39     command: "true"
  
```

Lo ejecutamos

```

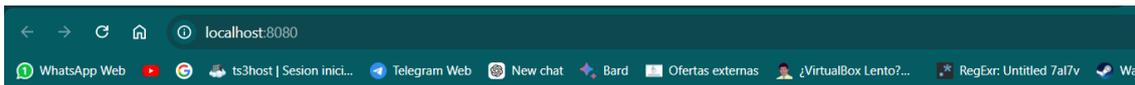
miguellillo@miguellilloSRI:~/practica62$ docker-compose up -d
Creating volume "practica62_mysql-data" with default driver
Pulling mysql (mysql:5.7)...
5.7: Pulling from library/mysql
20e4dcae4c69: Extracting [=====>] 6.291MB/50.5MB
1c56c3d4ce74: Download complete
e9f03a1c24ce: Download complete
68c3898c2015: Download complete
6b95a940e7b6: Download complete
90986bb8de6e: Download complete
ae71319cb779: Download complete
ffc89e9dfd88: Download complete
43d05e938198: Download complete
064b2d298fba: Download complete
df9a4d85569b: Download complete
  
```

Probamoso que se ejecute



```
Actividades Terminal 9 de oct 12:36
miguellillo@miguellilloSRI: ~/practica62
miguellillo@miguellilloSRI:~/practica62$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
8ebc85f40a22   practica62_nginx  "/docker-entrypoint.s..."  2 minutes ago  Up 2 minutes  0.0.0.0:80->80/tcp      nginx-container
134dc8a4f9f8   mysql:5.7      "/docker-entrypoint.s..."  2 minutes ago  Up 2 minutes  3306/tcp, 33060/tcp     mysql-container
50949c14f60f   php:7.0-fpm    "/docker-php-entrypoi..."  2 minutes ago  Up 2 minutes  9000/tcp                php-container
fc9c0f6bc484   httpd         "httpd-foreground"       5 days ago    Exited (0) 5 days ago    nigel
ccf2088165a0   hello-world    "/hello"                  5 days ago    Exited (0) 5 days ago    interesting_ferni
```

Probamos a acceder a la web funciona correctamente



```
localhost:8080
WhatsApp Web  ts3host | Sesion inici...  Telegram Web  New chat  Bard  Ofertas externas  VirtualBox Lento?...  RegExr: Untitled 7a17v  Wall
```

¡Hola mundo!

Estamos corriendo PHP, version: 7.0.33

La base de datos "mydb" tiene las siguientes tablas:

- columns_priv
- db
- engine_cost
- event
- func
- general_log
- gtid_executed
- help_category
- help_keyword
- help_relation

Miguellillo