Ejercicio 3

CIFP La Laboral - Módulo Despliegue de Aplicaciones Web

> 👥 **Autores:** Pelayo Rodríguez e Iker Pérez
> 📌 **Fecha de entrega y exposición:** Viernes, 21 de febrero de 2025.
> 📂 **Repositorio GitHub:**

📑 ÍNDICE

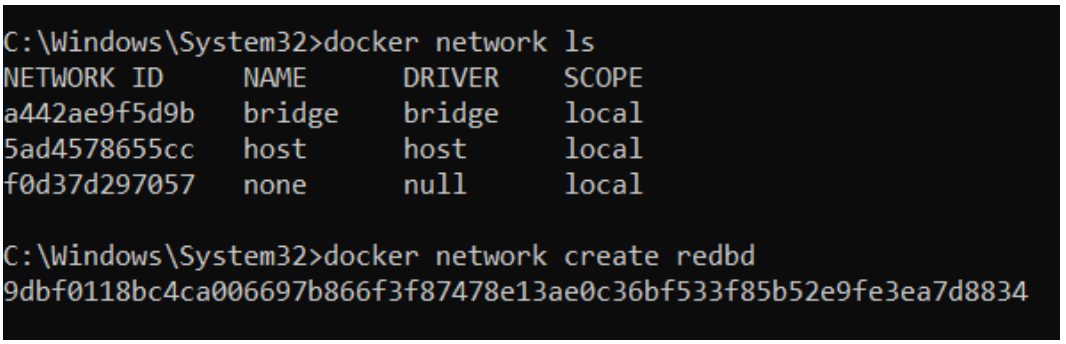
🌐 Ejercicio 3: Contenedores en Red

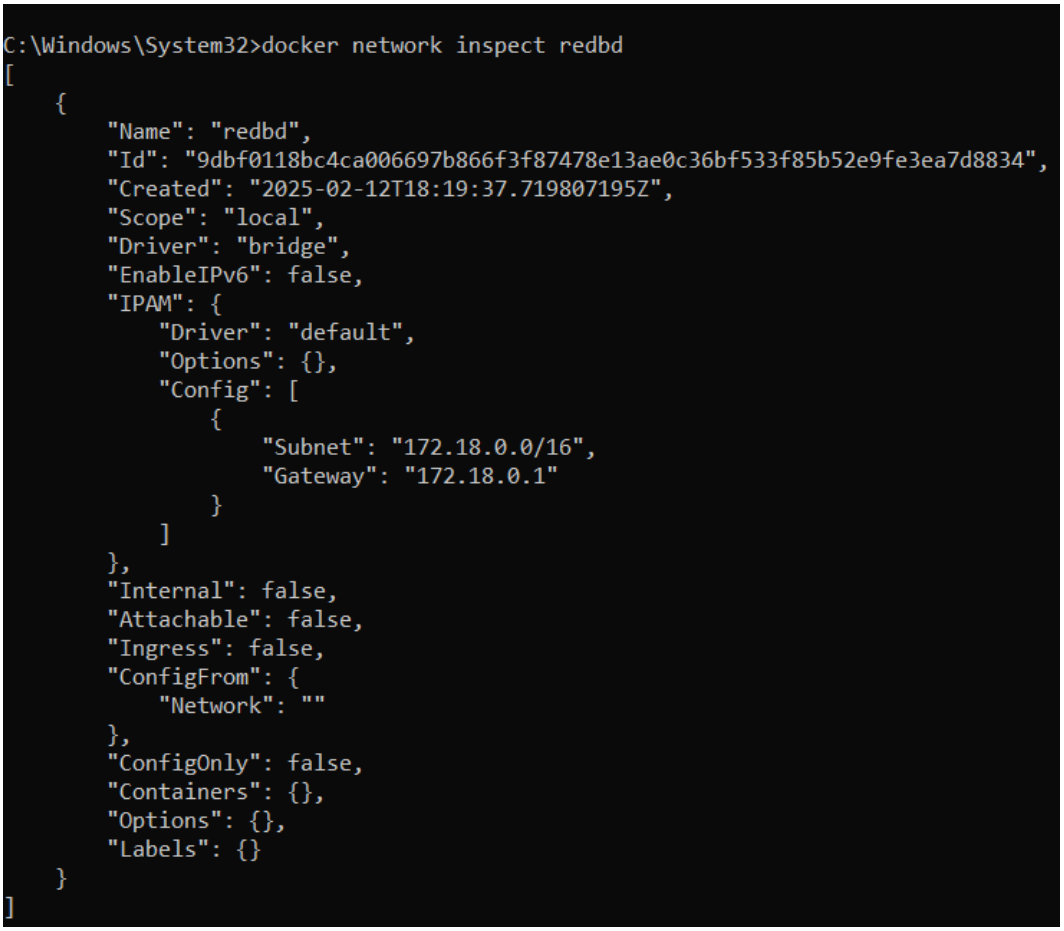
🛠️ Crear una red bridge (redbd)

En **Docker Desktop**, no hay una sección visual para redes en la interfaz, pero podemos verlas y administrarlas con comandos Docker.

Primero listamos las redes, luego, creamos la red bridge pedida y por ultimo vemos los detalles de esa red y comprobamos que se ha creado correctamente:

```
$docker network ls
$docker network create redbd
$docker network inspect redbd
```





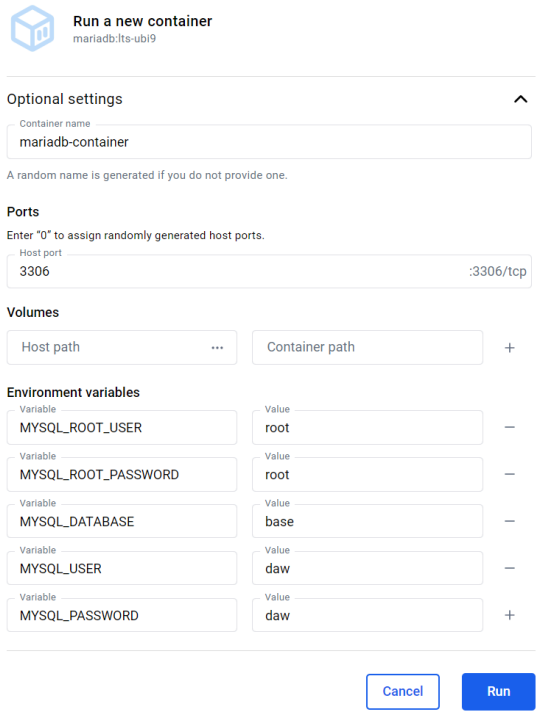
🐬 Crear el contenedor de MariaDB con comandos

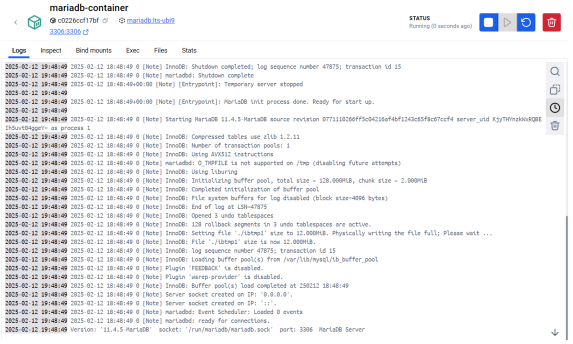
Ejecutamos en la terminal:

```
$docker run -d --name mariadb-container \
  --network redbd \
  -e MYSQL_ROOT_PASSWORD=root \
  -e MYSQL_DATABASE=base \
  -e MYSQL_USER=daw \
  -e MYSQL_PASSWORD=daw \
  -p 3306:3306 \
  -v mariadb_data:/var/lib/mysql \
  mariadb:latest
```

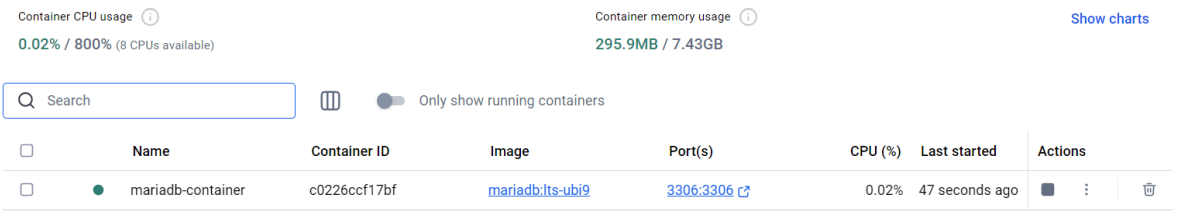
```
PS C:\Windows\system32> docker run -d --name mariadb-container `
>>    --network redbd `
>>    -e MYSQL_ROOT_PASSWORD=root `
>>    -e MYSQL_DATABASE=base `
>>    -e MYSQL_USER=daw `
>>    -e MYSQL_PASSWORD=daw `
>>    -p 3306:3306 `
>>    -v mariadb_data:/var/lib/mysql `
>>   mariadb:latest
>>
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
2cbd49ab14b1: Download complete
f67c6fbc0ef5: Download complete
640331c2cc76: Download complete
1f731489858b: Download complete
65dd09f27c61: Download complete
edb426f4a1af: Download complete
760f6e3db6bf: Download complete
5a7813e071bf: Download complete
Digest: sha256:bfb1298c06cd15f446f1c59600b3a856dae861705d1a2bd2a00edbd6c74ba748
Status: Downloaded newer image for mariadb:latest
e8c91772fc2d208558f8bcb5cc6eff647c6758b2192a35dc37f4aa588820390a
```

🐬 Crear el contenedor de MariaDB con Docker Desktop





Podemos ver en los logs que el contenedor de MariaDB se ha iniciado correctamente. El sistema nos muestra una secuencia de mensajes de inicialización que confirman que la base de datos está lista para aceptar conexiones y que todos los componentes necesarios se han cargado correctamente.



🌐 Crear el contenedor con Adminer o phpMyAdmin con comandos

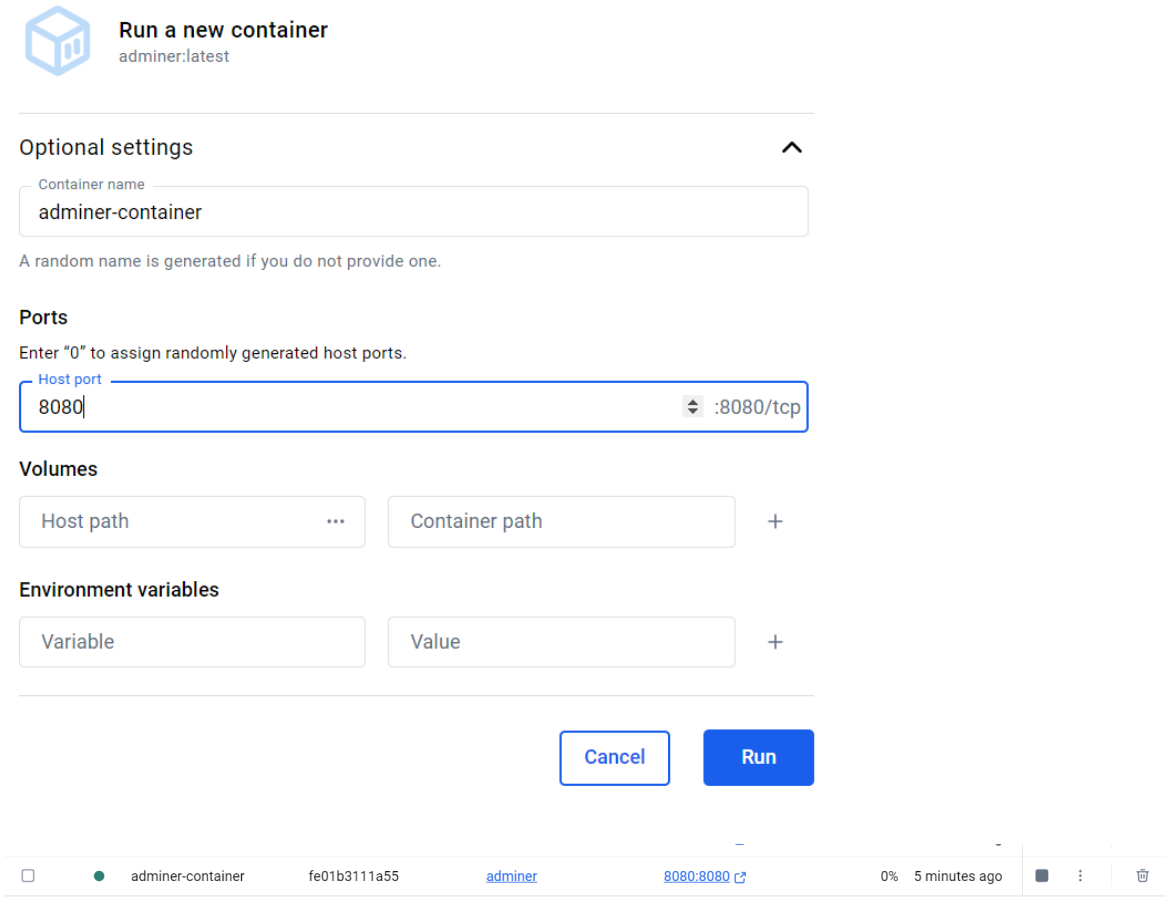
```
$docker run -d --name adminer-container \
  --network redbd \
  -p 8080:8080 \
  adminer
```

```
PS C:\Windows\system32> docker run -d --name adminer-container `
>>   --network redbd `
>>   -p 8080:8080 `
>>   adminer
>>
Unable to find image 'adminer:latest' locally
latest: Pulling from library/adminer
c82cd9b427d9: Download complete
ed94e1c95a57: Download complete
9a4cd7b75371: Download complete
798a45c9628c: Download complete
884bce373183: Download complete
73226dab8db5: Download complete
574dfab7cda2: Download complete
Digest: sha256:34d37131366c5aa84e1693dbed48593ed6f95fb450b576c1a7a59d3a9c9e8802
Status: Downloaded newer image for adminer:latest
fe01b3111a553aac63aac55e40ee1590564ea31f511daef9e2eadc605aeb6772
```

🌐 Crear el contenedor con Adminer o phpMyAdmin con Docker Desktop

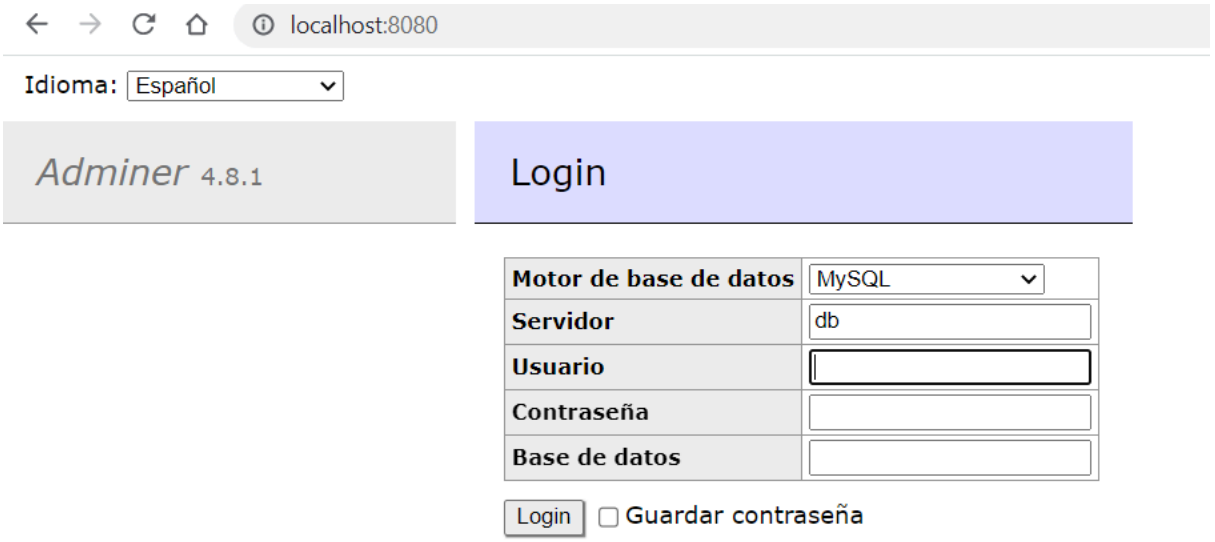
- Vamos a la pestaña **"Images"** y hacemos clic en **"Run"** con la imagen de Adminer.

- En **"Advanced settings"**:
  - Nombre: `adminer-container`.
  - Host Port: `8080`.
- Hacemos clic en **"Run"**.

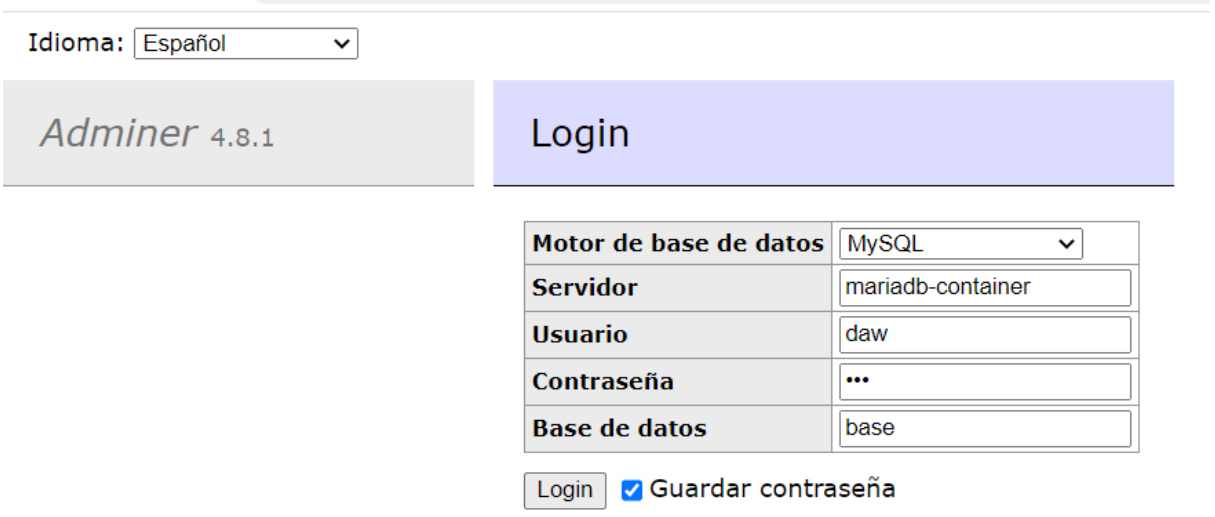


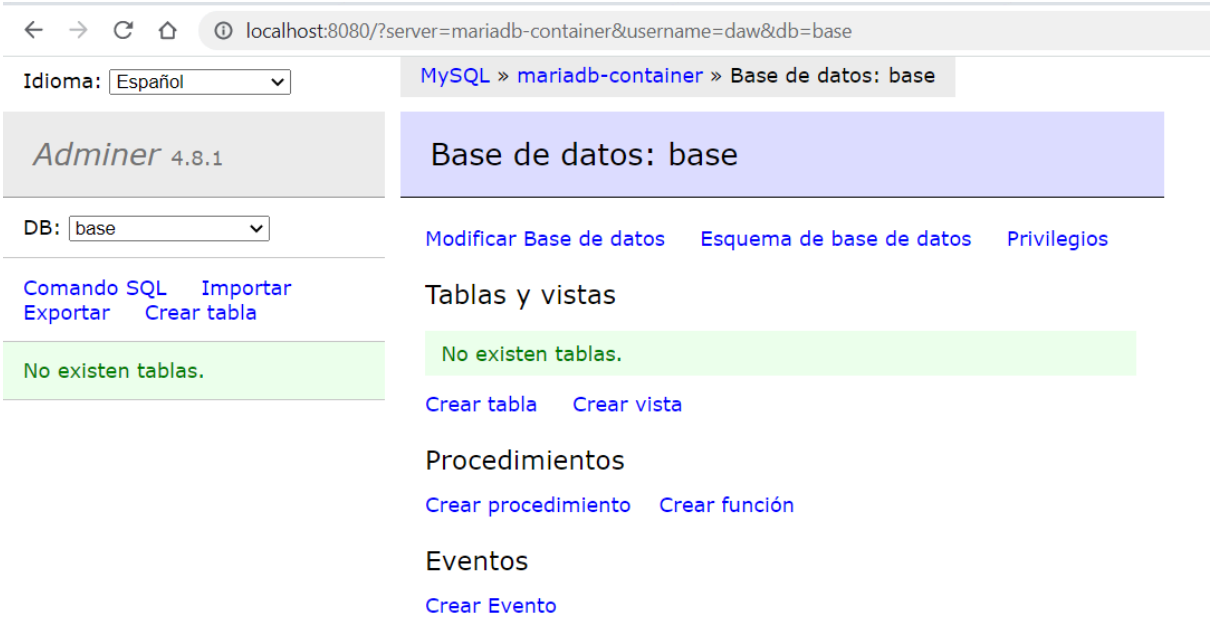
💻 Acceder a la base de datos desde la interfaz gráfica

Abrimos el navegador y vamos a `http://localhost:8080`.



- Rellenamos los datos de conexión:
  - **Servidor**: `mariadb-container` (nombre del contenedor).
  - **Usuario**: `daw`.
  - **Contraseña**: `daw`.
  - **Base de datos**: `base`.
- Hacemos clic en **"Login"**.



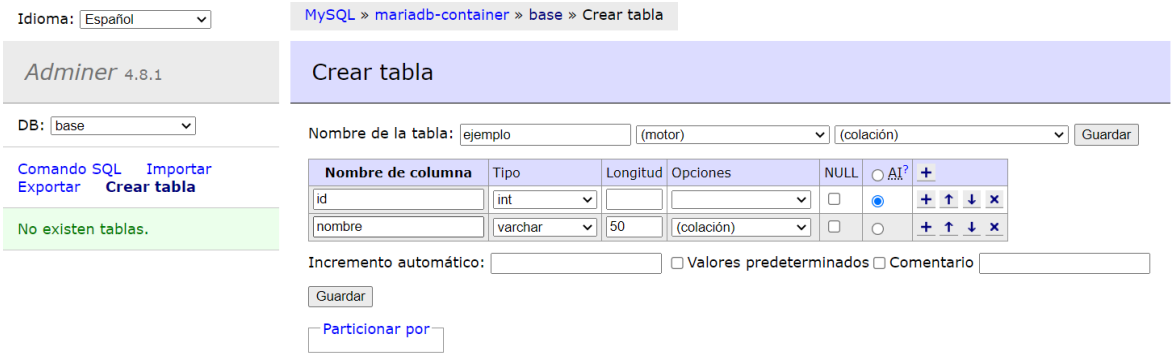


Crear una Tabla en la Base de Datos

- Entramos en la base de datos `base` y elegimos **"Crear tabla"**.
- Definimos una tabla de ejemplo:

Campo	Tipo	Clave primaria
id	INT	✅ (Auto-incremental)
nombre	VARCHAR(50)	❌

- Guardamos los cambios.



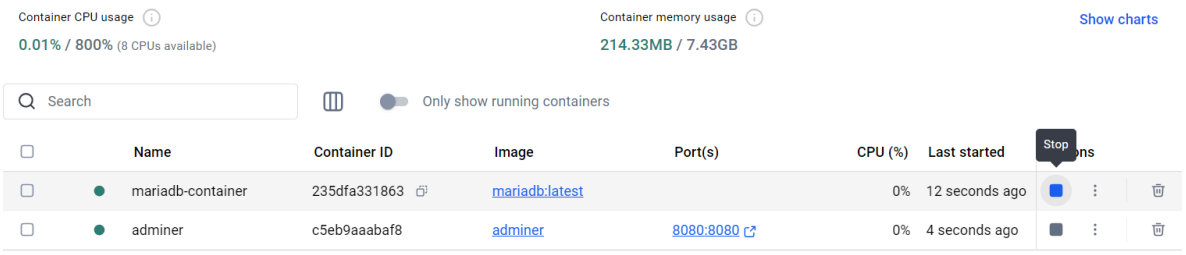
🧹 Borrar los contenedores, la red y los volúmenes utilizados con comandos

```
$docker stop mariadb-container adminer
$docker rm mariadb-container adminer
$docker network rm my_network
$docker volume prune -f
```

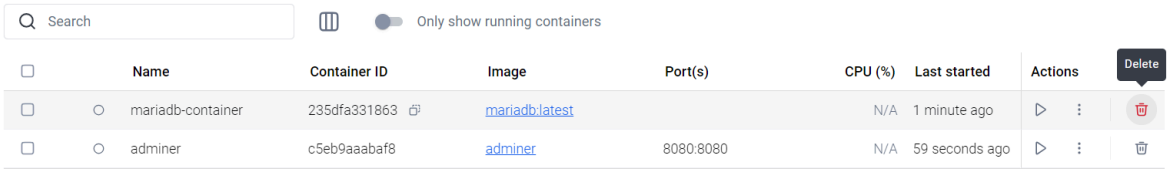
```
PS C:\Windows\system32> docker stop mariadb-container adminer
mariadb-container
adminer
PS C:\Windows\system32> docker rm mariadb-container adminer
mariadb-container
adminer
PS C:\Windows\system32> docker network rm my_network
my_network
PS C:\Windows\system32> docker volume prune -f
Deleted Volumes:
8e9a6bd9818e247f1aaaa94df39354bdd728d696cf7f40bad37ec1a21c19276e

Total reclaimed space: 156MB
```

🧹 Borrar los contenedores, la red y los volúmenes utilizados con Docker Desktop

Primero paramos los contenedores, haciendo click en Stop.



Y luego los borramos:



Una vez borrados podemos borrar sus imagenes: