

Tarea Evaluable Docker - 2EV

- Ejercicio 1 -

Ejercicio 1 - Contenedores en red y Docker
Desktop

Realizado por ORTIZ GONZALEZ Fernando

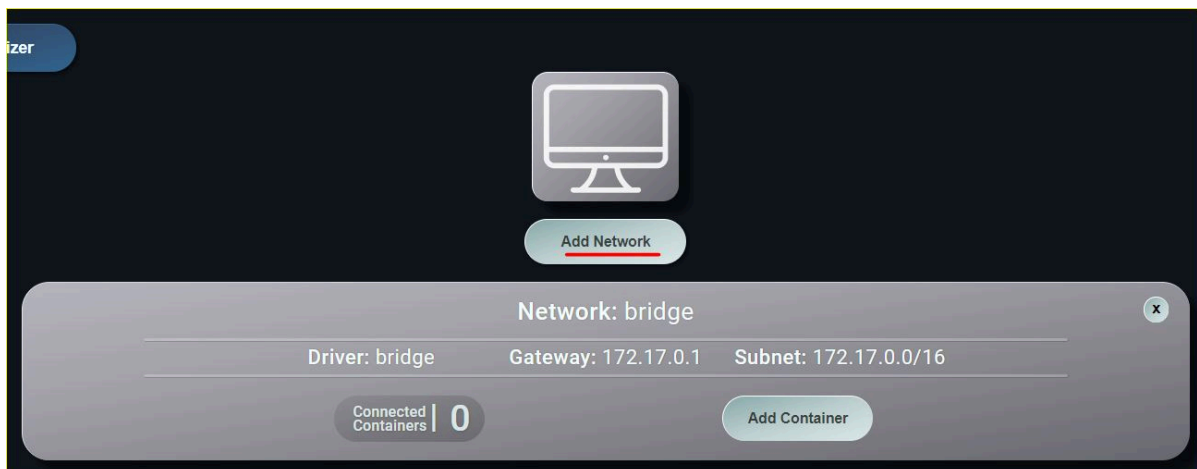
Tarea Evaluable Docker - 2EV

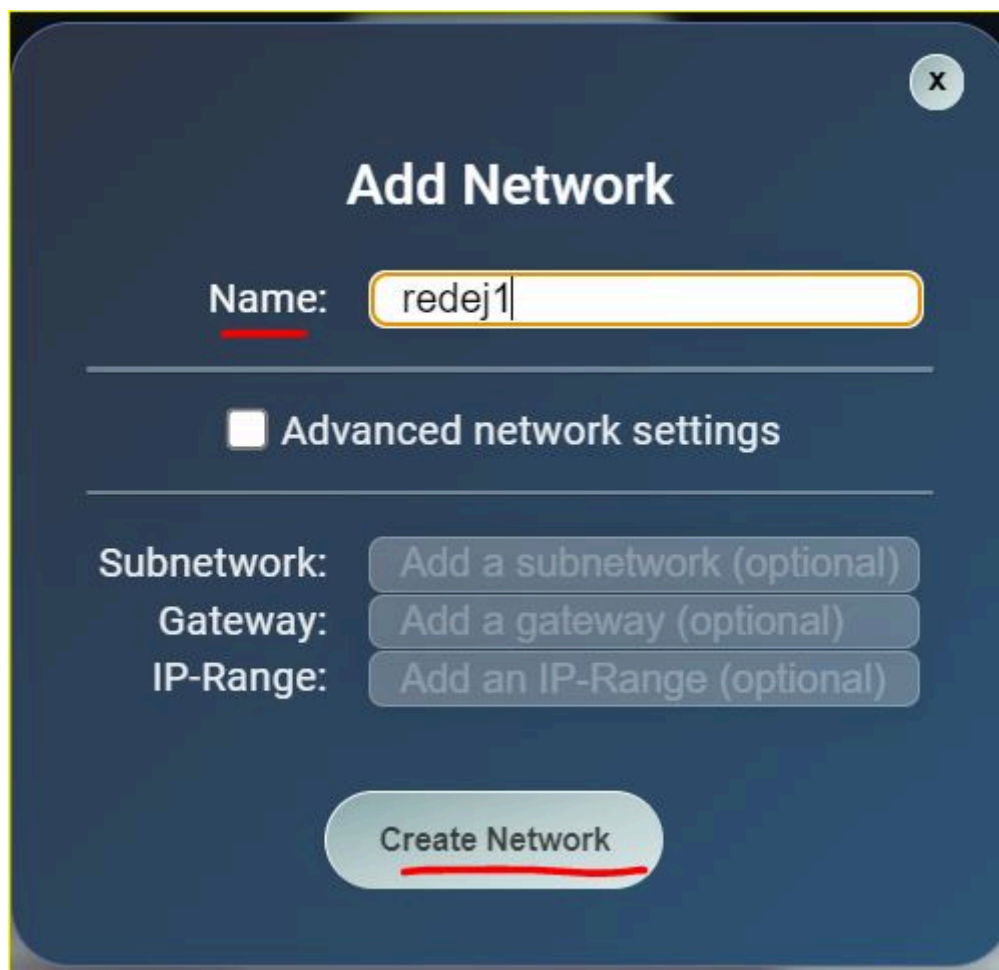
- Ejercicio 1 -

1. Crea una red bridge `redej1`
2. Crea un contenedor con una imagen de `mysql`
3. Crear un contenedor con `Adminer`
4. Mostrar BD y tablas creadas
5. Opcion elegida: Disk Usage

1. Crea una red bridge `redej1`

Luego de instalar la extension solicitada en los *pasos previos* para la realizacion del ejercicio 1. Dentro de la extensión abrimos el panel de *network* y procedemos a crear la red de tipo *bridge* `redej1`. En la misma se conectaran los contenedores creados posteriormente.





Add Network

Name:

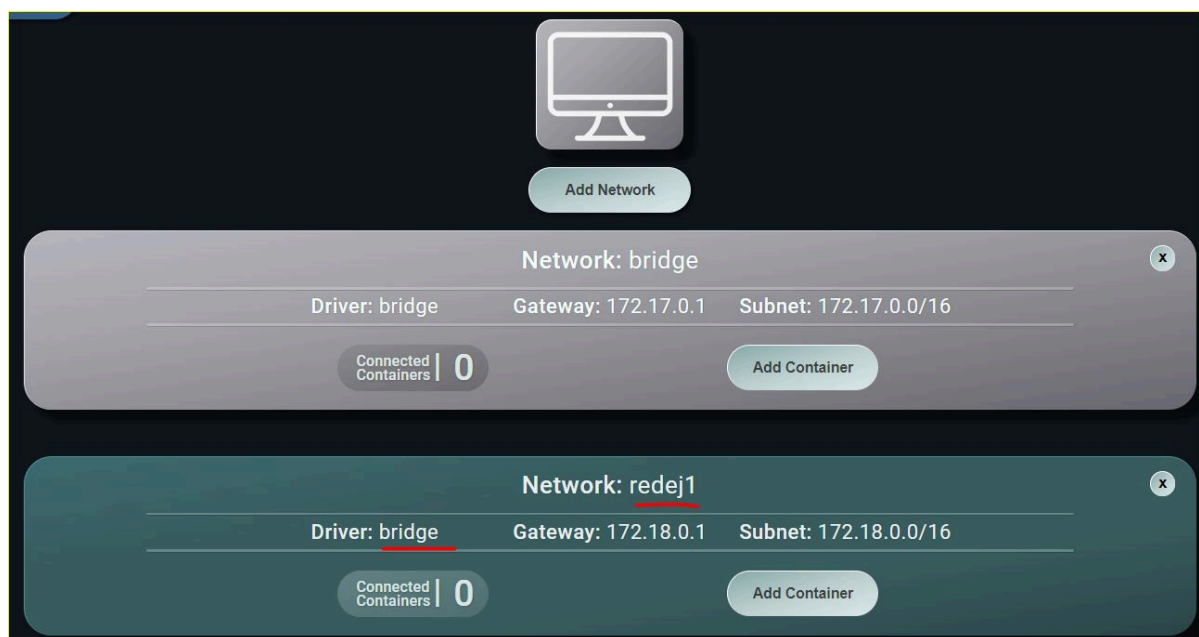
☐ **Advanced network settings**

Subnetwork:

Gateway:

IP-Range:

Create Network



Add Network

Network: bridge

Driver: bridge Gateway: 172.17.0.1 Subnet: 172.17.0.0/16

Connected Containers | 0

Add Container

Network: redej1

Driver: bridge Gateway: 172.18.0.1 Subnet: 172.18.0.0/16

Connected Containers | 0



Add Container


































2. Crea un contenedor con una imagen de mariaDB

Crea un contenedor con una imagen de mariaDB que estará en la red **redej1**. Este contenedor se ejecutará en segundo plano, y será accesible a través del puerto 3306.


Se realiza una descarga de la imagen mariaDB del DockerHub y luego se genera el contenedor (*run*) como se muestra en las imágenes.

Container CPU usage ⓘ 0.03% / 800% (8 CPUs available) Container memory usage ⓘ 93.55MB / 3.66GB [Show charts](#)

Q Search   Only show running containers


<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	prueba-web	5e1cbe67e67d	ubuntu		0%	12 days ago	  
<input type="checkbox"/>	mimariadb	29bbd426831b	mariadb:10.5	3306:3306	0%		  
<input type="checkbox"/>	hora-contenedor2	066d9295cb8b	ubuntu		0%	13 days ago	  
<input type="checkbox"/>	hora-contenedor	2c49e8a46c48	ubuntu		0%	13 days ago	  
<input type="checkbox"/>	contenedor5	767a45577ace	ubuntu		0%	13 days ago	  
<input type="checkbox"/>	contenedor2	75f6383572c7	ubuntu		0%	13 days ago	  
<input type="checkbox"/>	contenedor1	baf05938b5b4	ubuntu		0%	13 days ago	  
<input type="checkbox"/>	hungry_franklin	0dbd542687cb	ubuntu		0%	13 days ago	  
<input type="checkbox"/>	zen_cartwright	6a6c817d44b3	ubuntu		0%	13 days ago	  
<input type="checkbox"/>	quirky_bouman	5cf6823e3809	hello-world:latest		0%	13 days ago	  
<input type="checkbox"/>	<u>mariadb_daw_ej1</u>	27113cd13461	mariadb:10.5		0.03%	9 minutes ago	  

[Images](#) / [mariadb:10.5](#)

mariadb:10.5 IN USE CREATED 3 months ago SIZE 538.08 MB Recommended fixes ▾ **Run** ▾  Pull Push to Docker Hub [Give feedback](#)

Layers (26) [Vulnerabilities](#) [Packages](#)

0	ARG RELEASE	0 B
1	ARG LAUNCHPAD_BUILD_ARCH	0 B
2	LABEL org.opencontainers.image.ref.name=ubuntu	0 B
3	LABEL org.opencontainers.image.version=20.04	0 B
4	ADD file:f9ee450324e6ff2c946bc9aae5cf7e35e240dbd3...	81.66 MB
5	CMD ["/bin/bash"]	0 B
6	RUN /bin/sh -c groupadd -r mysql && useradd -r -g mysql ...	376.83 KB
7	ENV GOSU_VERSION=1.17	0 B




This image has not been analyzed

You can use Docker Scout to analyze local images and list its vulnerabilities.

[Start analysis](#)

[Enable background indexing in Settings](#) so your results are always ready.

- Definir una contraseña para el usuario root (elegimos *root*), y un usuario con tu nombre de pila **fernando** y con contraseña **123123**. La BD por defecto será **DAW**



Run a new container
 mariadb:10.5

Optional settings
^

Container name

A random name is generated if you do not provide one.

Ports
 Enter "0" to assign randomly generated host ports.

Host port

Volumes

Host path
 ...

Container path
 +

Environment variables

Variable	Value	
<input type="text" value="MARIADB_ROOT_PASSWORD"/>	<input type="text" value="root"/>	—
<input type="text" value="MARIADB_DATABASE"/>	<input type="text" value="DAW"/>	—
<input type="text" value="MARIADB_USER"/>	<input type="text" value="Fernando"/>	—
<input type="text" value="MARIADB_PASSWORD"/>	<input type="text" value="123123"/>	+

- Genera un *script SQL* que cree una tabla *módulos* con algunos registros con los nombres de los módulos que estás estudiando (DIW, DWES, DWEC)

```

CREATE TABLE módulos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50) NOT NULL
);
INSERT INTO módulos (nombre) VALUES
('DIW'),
('DWEC'),
('DWES'),
('DAW');
  
```

3. Crear un contenedor con Adminer

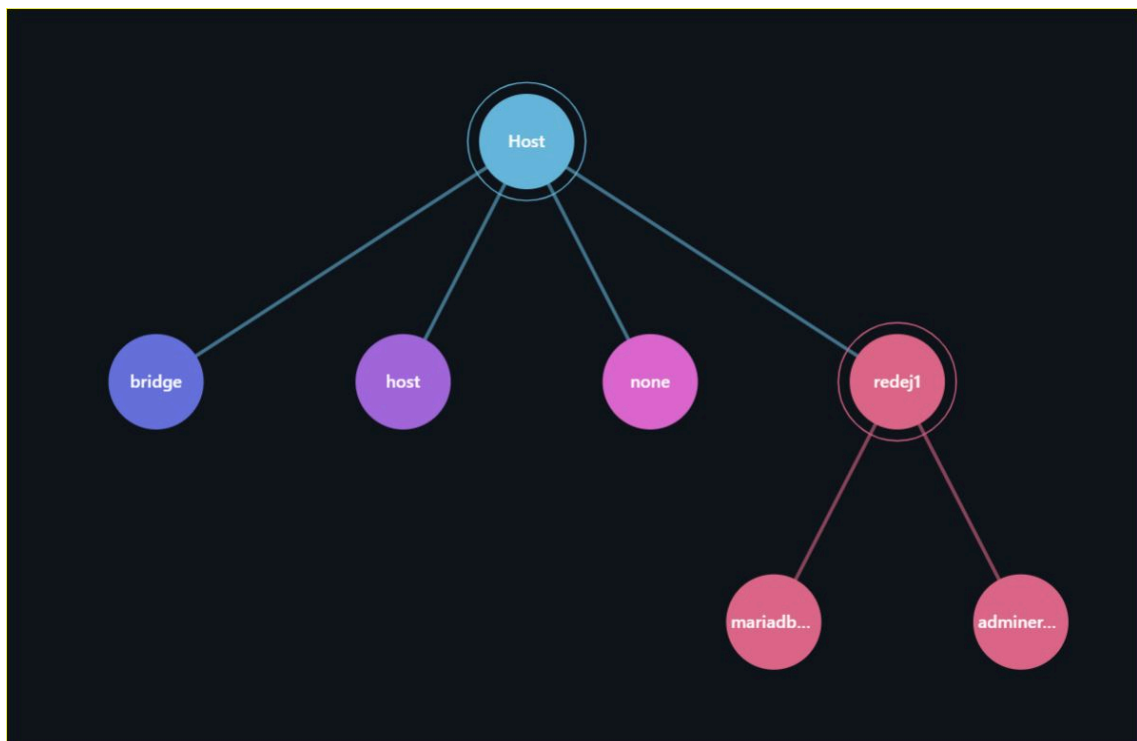
- Crear un contenedor con **Adminer** que se pueda conectar al contenedor de la BD.

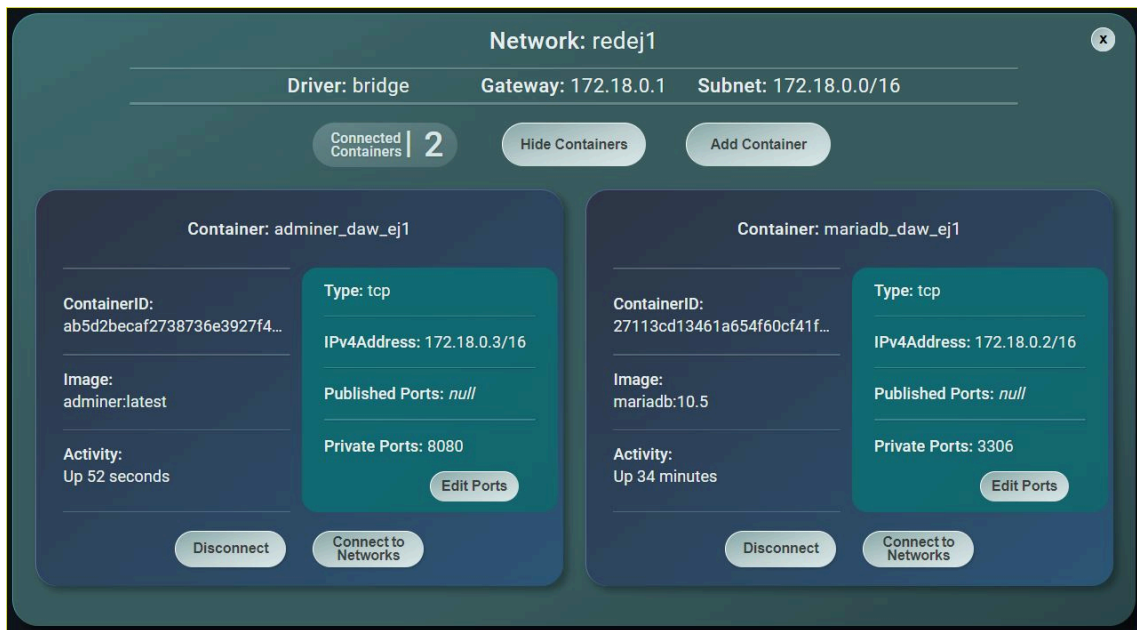


The top screenshot shows the Docker Hub page for the **adminer** image. It includes the Docker Official Image logo, a description "Database management in a single PHP file.", and a command to run the container: `$ docker run --link some_database:db -p 9000:9000 adminer:fastcgi`. Below the command, it says "Then point your web server to port 9000 of the container." and a note about exposing the FastCGI socket to the Internet.

The bottom screenshot shows the Docker Desktop interface for a container named **adminer_daw_ej1**. The container is running the **adminer:latest** image. The logs show: `[Fri May 9 19:04:47 2025] PHP 8.4.7 Development Server (http://[::]:8080) started`.

- Descargamos la imagen correspondiente del Hub y luego *corremos* el nuevo contenedor y realizamos la conexión a la red junto a el contenedor de **mariaDB**.





4. Mostrar BD y tablas creadas

Desde la interfaz gráfica elegida, conéctate a la BD con tu usuario personal, ejecuta el script con los datos de los módulos y muestra la BD y la tabla creadas



Adminer 5.2.1 5.3.0

BD: **DAW** ▼

Comando SQL Importar
Exportar Crear tabla

registros modulos

```
CREATE TABLE modulos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50) NOT NULL
)
```

Consulta ejecutada, 0 registros afectados. (0.385 s)

```
INSERT INTO modulos (nombre) VALUES
('DIW'),
('DWEC'),
('DWES'),
('DAW')
```

Consulta ejecutada, 4 registros afectados. (0.047 s)

```
CREATE TABLE modulos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50) NOT NULL
);
INSERT INTO modulos (nombre) VALUES
('DIW'),
('DWEC'),
('DWES'),
('DAW');
```

BD: **DAW** ▼

Comando SQL Importar
Exportar Crear tabla

registros modulos

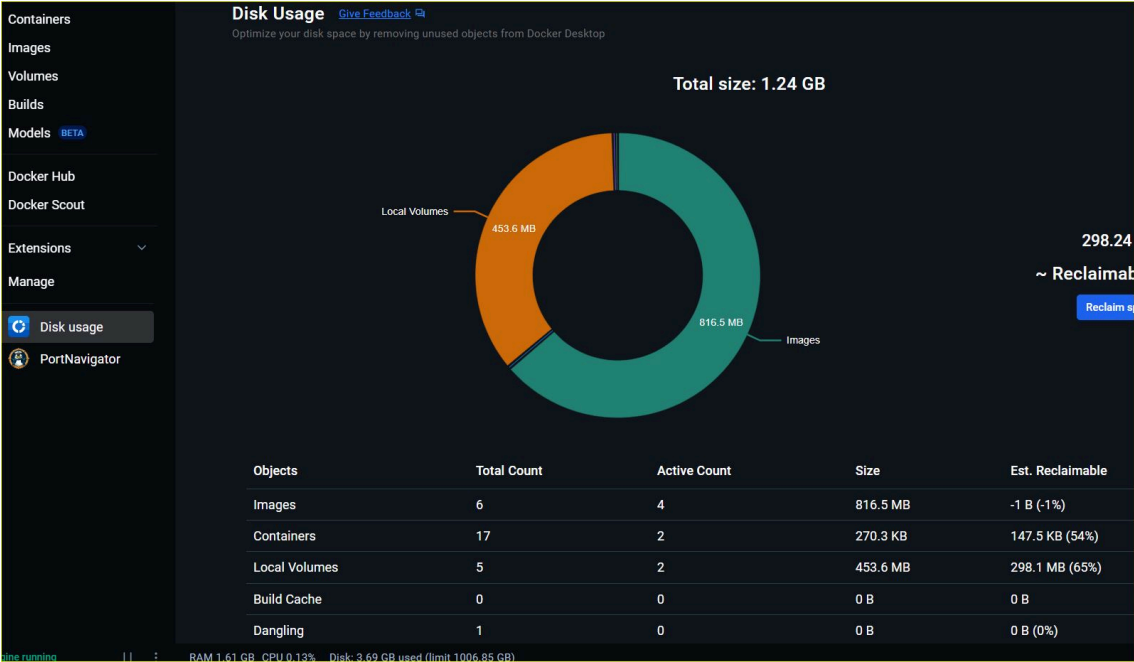
```
SELECT * FROM modulos
```

id	nombre
1	DIW
2	DWEC
3	DWES
4	DAW

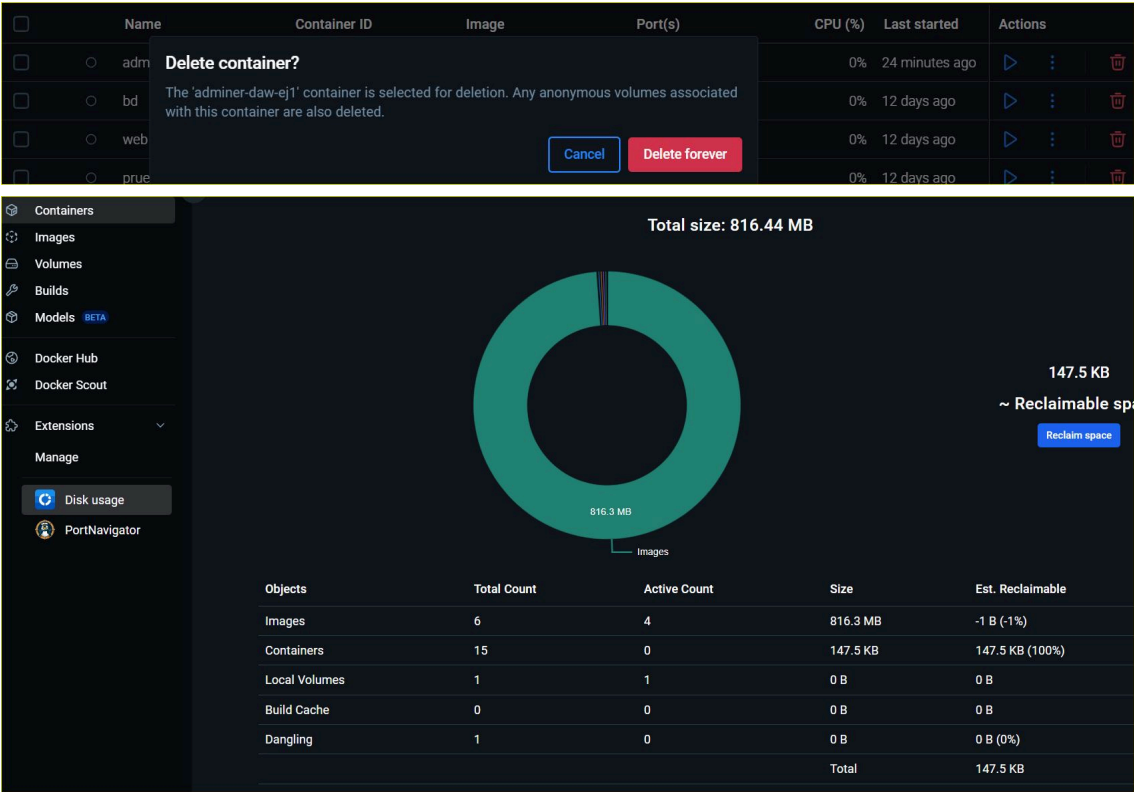
4 registros (0.001 s) Modificar, Explain, Exportar

5. Opcion elegida: Disk Usage

- Instala la extensión **Disk Usage**.
- Muestra el espacio ocupado.



- Borramos red, contenedores y volumen ocupado por las persistencias.



Se puede observar la diferencia en *Containers* y el *Total Size* que analiza la extensión para observar la eliminación de archivos, imágenes, red y contenedores.