

Ejercicio 1 - Contenedores en red y Docker Desktop

Documento que resuelve el ejercicio 1 de la Tarea 3- Despliegue de Aplicaciones Web, realizado por Epifanía Peralta López.

Ejercicio 1 - Contenedores en red y Docker Desktop

Notas

Resolución del ejercicio.

Notas

Este ejercicio se resuelve con Docker Desktop.

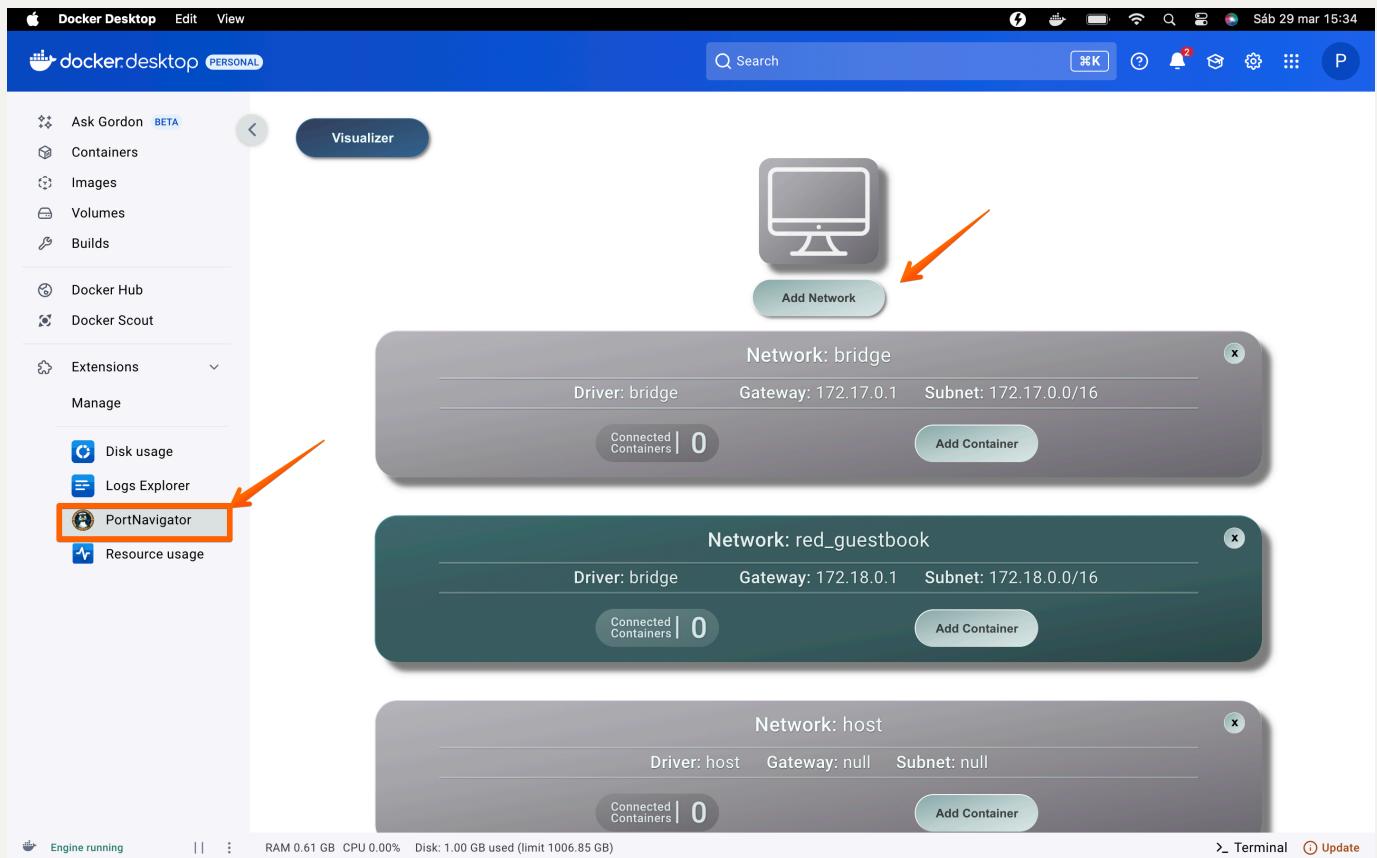
Si se necesitara utilizar comandos, se usa la Terminal integrada en Desktop.

Se sugiere utilizar la extensión **PortNavigator** para gestionar las redes en Docker Desktop.

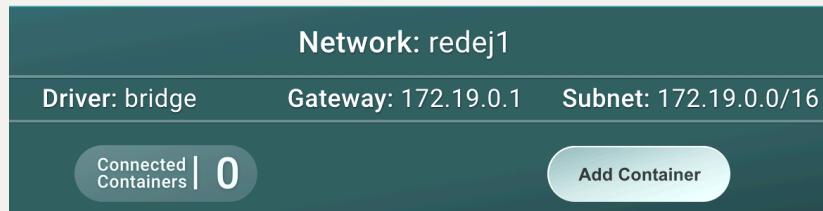
Resolución del ejercicio.

1. **Enunciado:** Crea una red bridge **redej1**

Para crear la red bridge utilizo la extensión PortNavigator:



Al hacer click en "Add Network" se crea la red que hemos llamado **redej1**

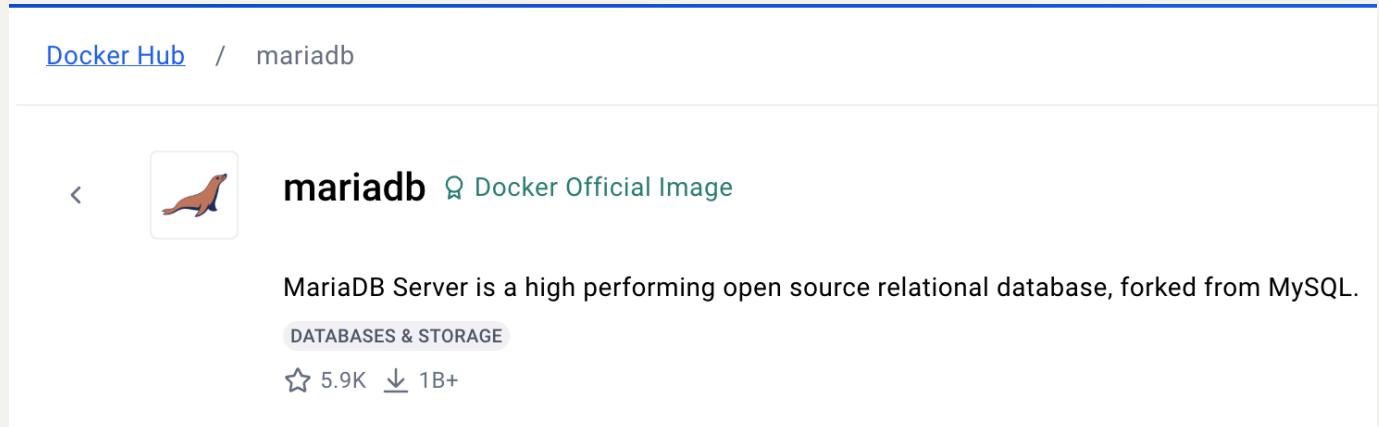


Las redes definidas por el usuario (por nosotros) nos ofrece un servidor DNS que reconoce los nombres de los contenedores. Por eso, cuando conectemos los dos contenedores, se puede hacer por el nombre. Podrían conectarse a través de la IP, pero nadie garantizaría que si en un momento dado se borra el contenedor de mariadb (porque se realice por ejemplo una actualización) y luego se vuelve a crear, que tome la misma dirección IP.

2. **Enunciado:** 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.

- Definir una contraseña para el usuario **root**, y un **usuario con tu nombre de pila** y con contraseña. La BD por defecto será **DAW**.
- Genera un script SQL que cree una tabla **módulos** con algunos registros con los nombres de los módulos que estás estudiando.

En Docker Desktop se puede acceder a Docker Hub, de donde se obtiene la imagen oficial de mariadb:



Además, las variables de entorno que se necesitan son:

```
$ docker run --detach --name some-mariadb --env MARIADB_USER=example-user --env MARIADB_PASSWORD=my_cool_secret --env MARIADB_DATABASE=example-database --env MARIADB_ROOT_PASSWORD=my-secret-pw mariadb:latest
```

Así, cuando se cree el contenedor, al haber ya definido esas 4 variables de entorno, antes de ejecutar el servicio se configura y se crean el usuario, contraseña y base de datos.

Los volúmenes y los bind mounts se configuran en el momento de crear el contenedor, no después. Con Docker Desktop, antes de arrancar el contenedor.

Estoy utilizando un macOS: voy a crear una carpeta en mi sistema donde se guardarán los datos de mariadb.

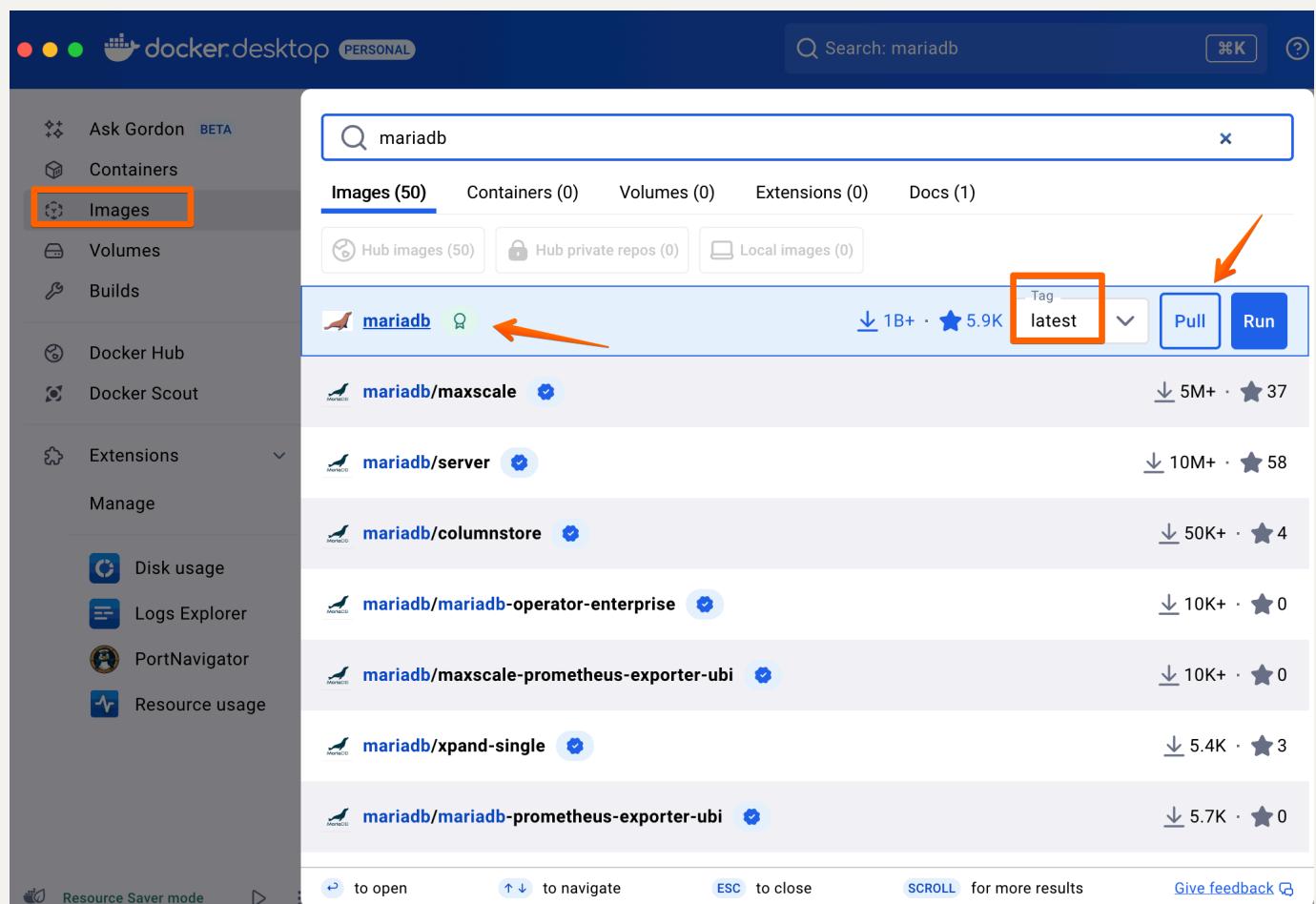
```

fanyperalta@MacBook-Air-de-Fany fanyperalta % pwd
/fanyperalta
[1] fanyperalta@MacBook-Air-de-Fany fanyperalta % mkdir -p docker/mariadb data
[1]+ 0  zsh
fanyperalta@MacBook-Air-de-Fany fanyperalta % ls
Applications Documents Movies Pictures docker      repositorios
CursoJavaScript Downloads Music Public libro-git2
Desktop Library OneDrive VirtualBox VMs mariadb
[1] fanyperalta@MacBook-Air-de-Fany fanyperalta % ls docker
mariadb_data
fanyperalta@MacBook-Air-de-Fany fanyperalta %

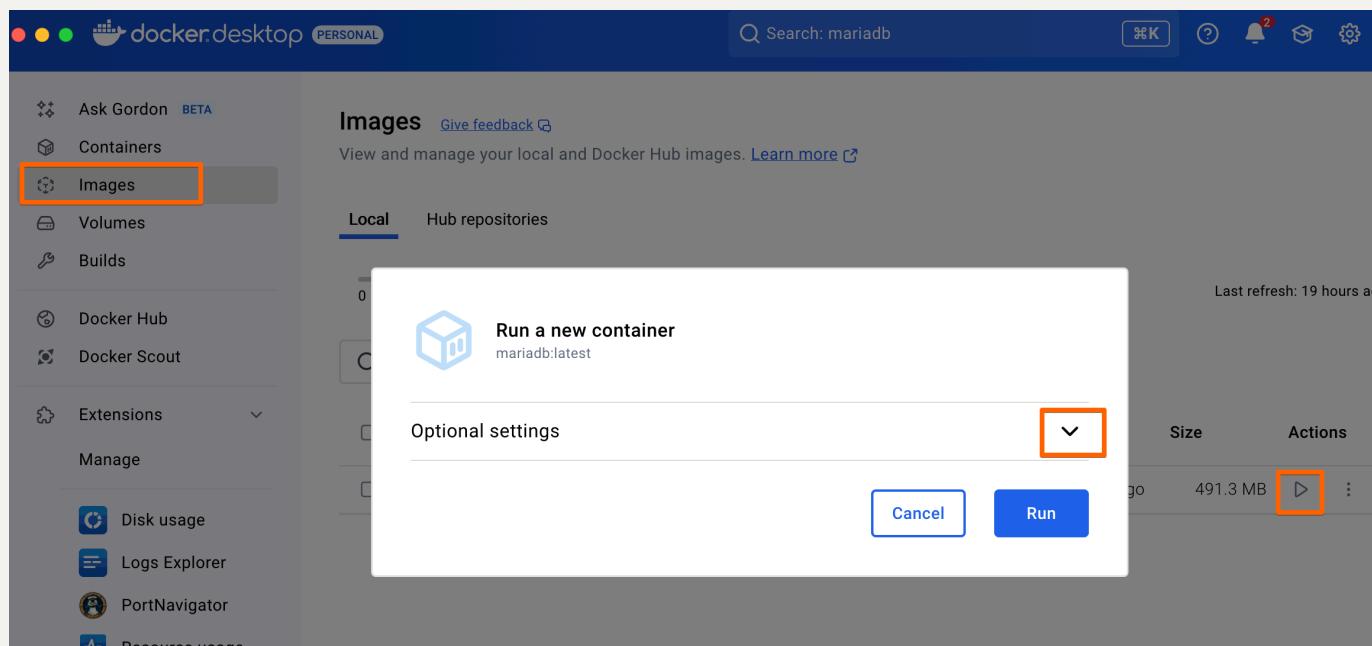
```

Ahora los pasos son los siguientes:

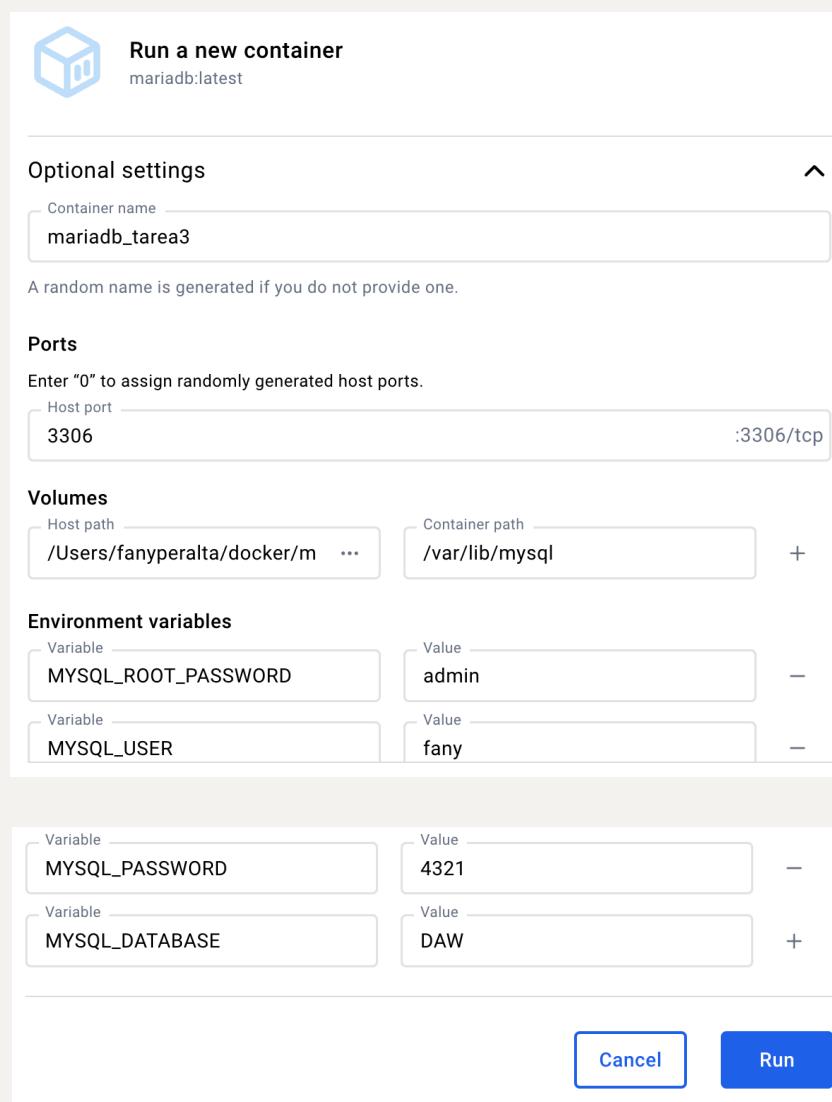
- selecciono la imagen `mariadb`:



- Una vez que tengo la imagen -> click en Run y despliego Opciones avanzadas:



- Se abren las Optional settings, para poder nombrar al contenedor, indicar el puerto, el almacenamiento y las 4 variables de entorno:



Así, el contenedor ya está creado. Lo compruebo:

The screenshot shows the Docker Desktop interface. On the left, there's a sidebar with various icons and links: Ask Gordon (BETA), Containers (selected), Images, Volumes, Builds, Docker Hub, Docker Scout, Extensions (Manage), Disk usage, Logs Explorer, PortNavigator, and Resource usage. The main area is titled 'Containers' with a 'Give feedback' link. It displays container statistics: Container CPU usage (0.02% / 800%), Container memory usage (346MB / 3.74GB), and a 'Show charts' button. Below this is a search bar and a toggle switch for 'Only show running containers'. A table lists the running container 'mariadb_tarea3' with details: Name (mariadb_tarea3), Container ID (6c80290c4cc0), Image (mariadb:lat 3306:3306), Port(s) (3306:3306), CPU (%)(0.02%), and Actions (button).

- En Docker Desktop no se puede elegir la red directamente al crear el contenedor desde la interfaz gráfica.

Una vez creado, la solución es utilizar la terminal incluida en Docker Desktop y escribir:

```
docker network connect redej1 mariadb_tarea3
```

Sería así:

The screenshot shows the Docker Desktop interface again. The top part is a table of containers, identical to the one above. Below it is a 'Terminal' window. The terminal window has a title bar 'Terminal' and a status bar with icons for '+' (new tab), '^' (minimize), and 'x' (close). The main area of the terminal shows the command 'fanyperalta@MacBook-Air-de-Fany ~ % docker network connect redej1 mariadb_tarea3' entered by the user. The output of the command is shown below the input line.

*Corrección: se puede conectar la red **redej1** al contenedor **mariadb_tarea3** a través de la extensión instalada de **Portnavigator**.*

Network: redej1

Driver: bridge Gateway: 172.19.0.1 Subnet: 172.19.0.0/16

Connected Containers | **1** Hide Containers Add Container

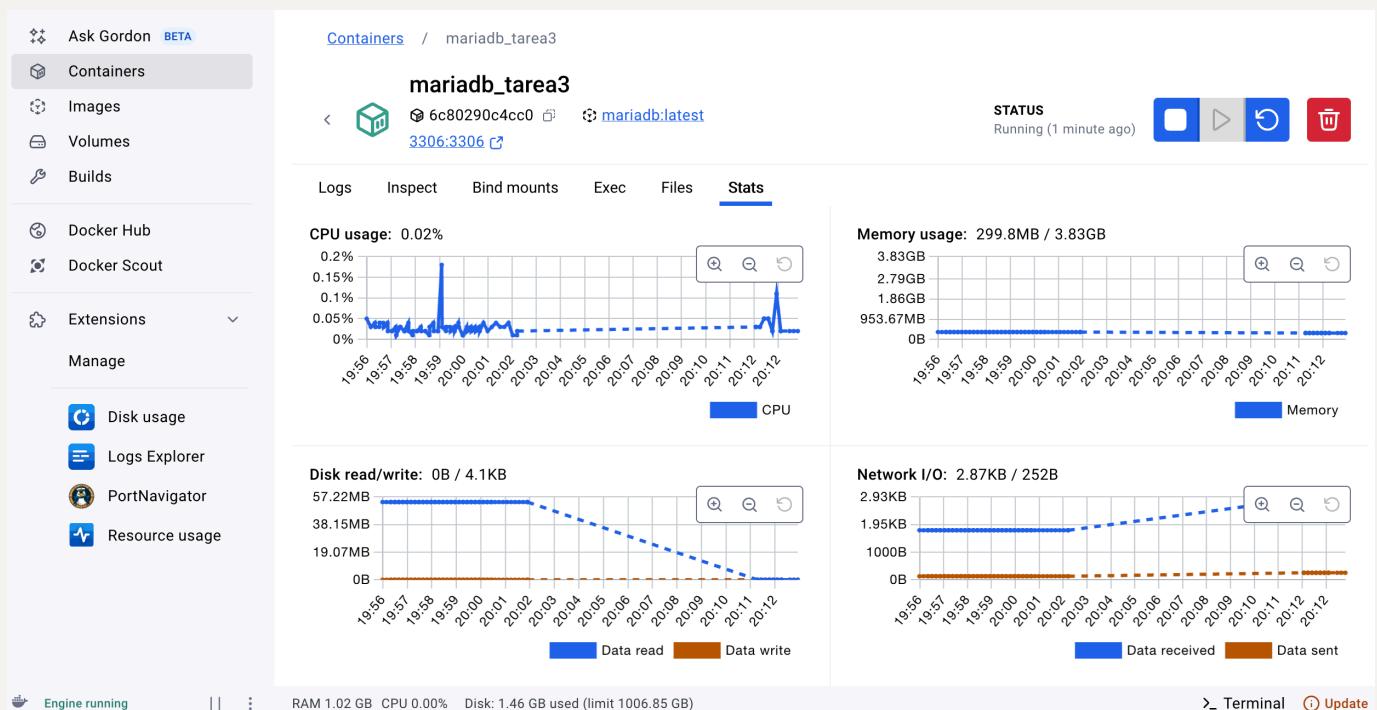
Container: mariadb_tarea3

ContainerID: 6c80290c4cc0a2c9c8e90...	Type: tcp IPv4Address: 172.19.0.2/16
Image: mariadb:latest	Published Ports: 3306:3306
Activity: Up 20 minutes	Private Ports: null

[Edit Ports](#)

[Disconnect](#) [Connect to Networks](#)

- Por último, en Action doy click en el triangulito para arrancar el contenedor. En Stats, compruebo que todo funciona:



- Dentro de la carpeta Ejercicio1 he guardado el archivo `init.sql` que he creado en SQL, con la tabla `módulos` que contiene varios registros con los módulos que estoy estudiando.

```

CREATE TABLE modulos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL
);

INSERT INTO modulos (nombre) VALUES
('Despliegue de Aplicaciones Web'),
('Programación'),
('Entornos de Desarrollo'),
('Lenguaje de Marcas');

```

3. Enunciado: Crea un contenedor con Adminer o con phpMyAdmin que se pueda conectar al contenedor de la BD.

Compruebo la info de la imagen de **phpMyAdmin** desde Docker Desktop:

Images (100) Containers (0) Volumes (0) Extensions (0) Docs (0)

Hub images (100) Hub private repos (0) Local images (0)

phpmyadmin	50M+ · ★ 1.1K
phpmyadmin	100M+ · ★ 1.2K
bitnami/p...	50M+ · ★ 48
bitnamich...	500K+ · ★ 0
elestio/php...	10K+ · ★ 0
linuxserver...	1M+ · ★ 21
vulnhub/php...	10K+ · ★ 1

Tag: latest **Pull:** **Run**

Docker Official Image 50M+ · ★ 1.1K [View on Hub](#) Updated 10 days ago

How to use this image

All of the following examples will bring you phpMyAdmin on `http://localhost:8080` where you can enjoy your happy MySQL and MariaDB administration.

Credentials

phpMyAdmin connects using your MySQL server credentials. Please check your

Los pasos a seguir para crear el contenedor **phpmyadmin_tarea3** son:

- Descargo la imagen anterior y accedo a las

Images [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

[Local](#) [Hub repositories](#)

385.99 MB / 386.03 MB in use 2 images

Last refresh: 26 minutes ago



Search



<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	mariadb	latest	310d29fbb581	1 month ago	491.3 MB	
<input type="checkbox"/>	phpmyadmin	latest	6cb0a7146734	2 months ago	818.79 MB	



- Incorporo los datos:

- Puerto: 8080:80, para acceder desde el navegador en

`http://localhost:8080`

- Variables de entorno:

- `PMA_HOST=mariadb_tarea3`, que es el nombre que le di al contenedor de mariadb como servidor de base de datos.
- `MYSQL_ROOT_PASSWORD=admin`, que es la contraseña que puse para **root** en mariadb_tarea3.

Run a new container
phpmyadmin:latest

Optional settings ^

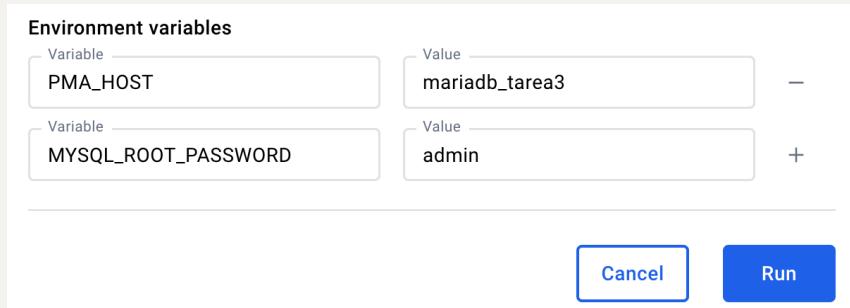
Container name A random name is generated if you do not provide one.

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

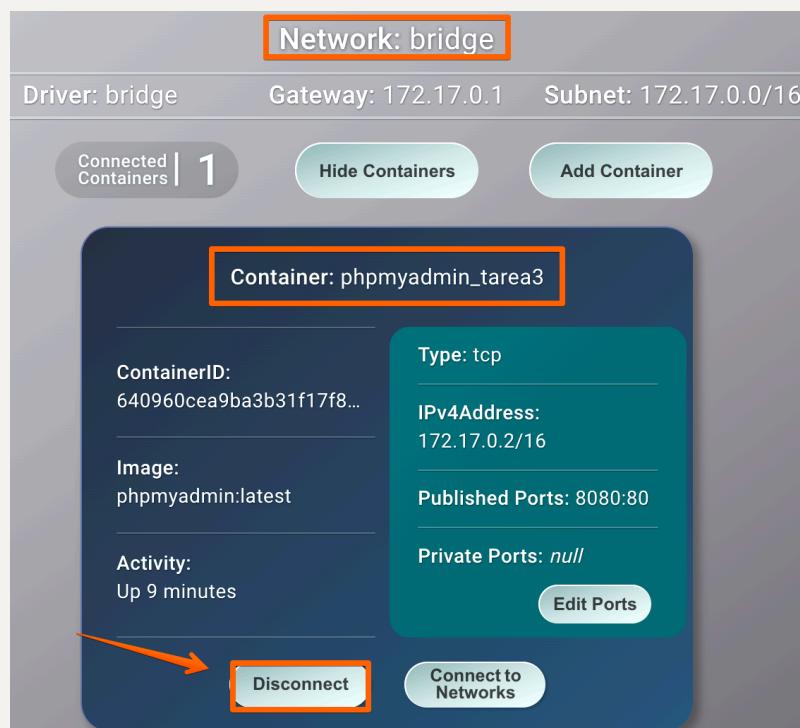
Host port :80/tcp

Volumes

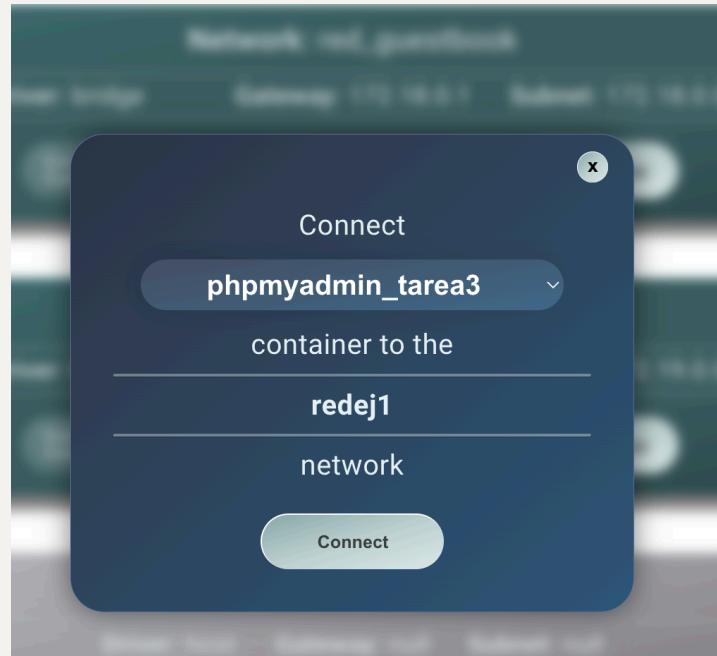
Host path ... Container path +



- Conecto phpmyadmin_tarea3 a la red redej1, utilizando el PortNavigator:
 - Primero tuve que cerrar Docker Desktop, como para que actualizara, porque no me salía el nuevo contenedor en ninguna red.
 - Al volver a abrirlo, ya me sale que está en la red `bridge`, por lo que lo desconecto de ahí.



* Y ahora ya puedo conectarlo a la red `redej1`:

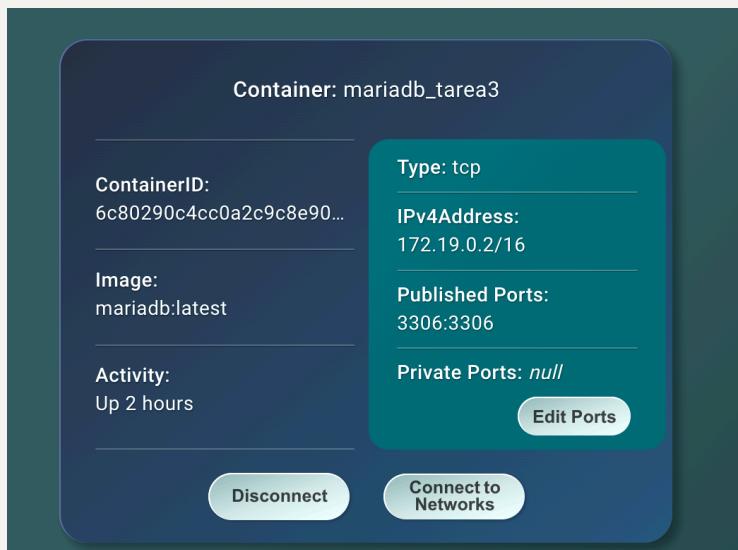
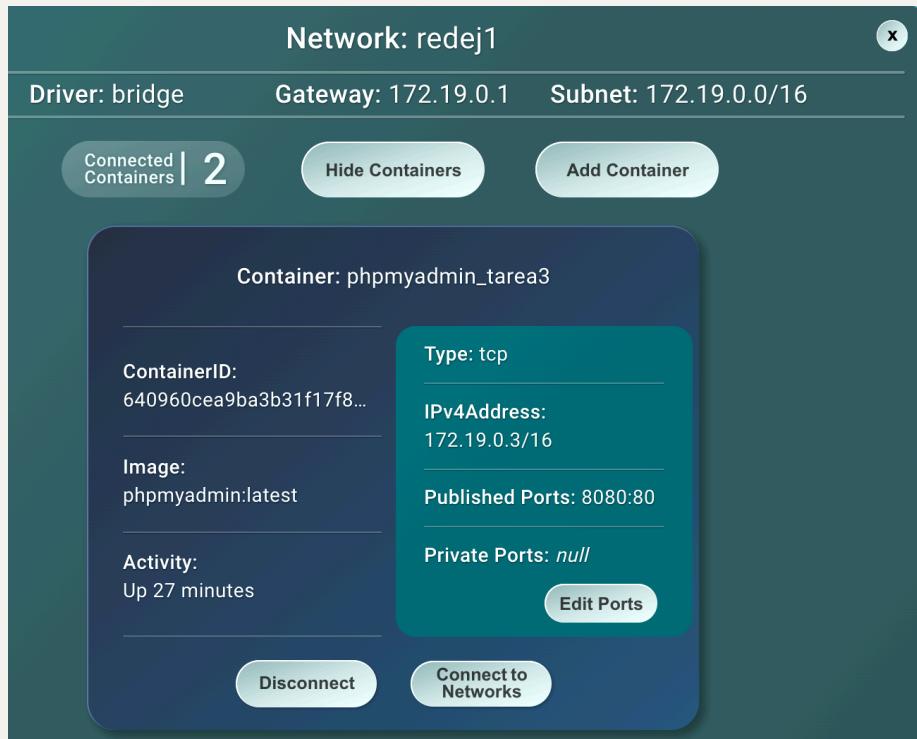


Directamente no deja `Add Container` a la `redej1`. Me dí cuenta que en `Inspect` del `phpmyadmin_tarea3` en `Network` figuraba `none`, por lo que en el `PortNavigator` me fui a esa red y ahí estaba. Desde ahí es donde me permitió conectarlo a la `redej1`.

Los contenedores creados y en ejecución son:

Containers Give feedback						
View all your running containers and applications. Learn more						
Container CPU usage			Container memory usage		Show charts	
0.03% / 800% (8 CPUs available)			313.91MB / 3.74GB			
Search	Filter	<input checked="" type="checkbox"/> Only show running containers	Name	Container ID	Image	Port(s)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	phpmyadmin_tarea3	640960cea9ba	phpmyadm	8080:80
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	mariadb_tarea3	6c80290c4cc0	mariadb:lat	3306:3306

Los parámetros de la red bridge utilizada: IPs



3. **Enunciado:** Desde la interfaz gráfica de `phpMyAdmin`, conectarse a la BD con el usuario personal, ejecuta el script con los datos de los módulos y muestra la BD y la tabla creados.

Desde Containers puedo acceder al navegador, haciendo click en los 3 puntitos según indico:

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage [\(i\)](#)

0.03% / 800% (8 CPUs available)

Container mem

334.72MB /

[Show charts](#)

Search



Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	
<input type="checkbox"/>	● phpmyadmin_tarea3	640960cea9ba	phpmyadmin: Open with browser	
<input type="checkbox"/>	● mariadb_tarea3	6c80290c4cc0	mariadb: lat 3306:3306	

[View details](#)

[View image packages and CVEs](#)

[Copy docker run](#)

[Open in terminal](#)

[View files](#)

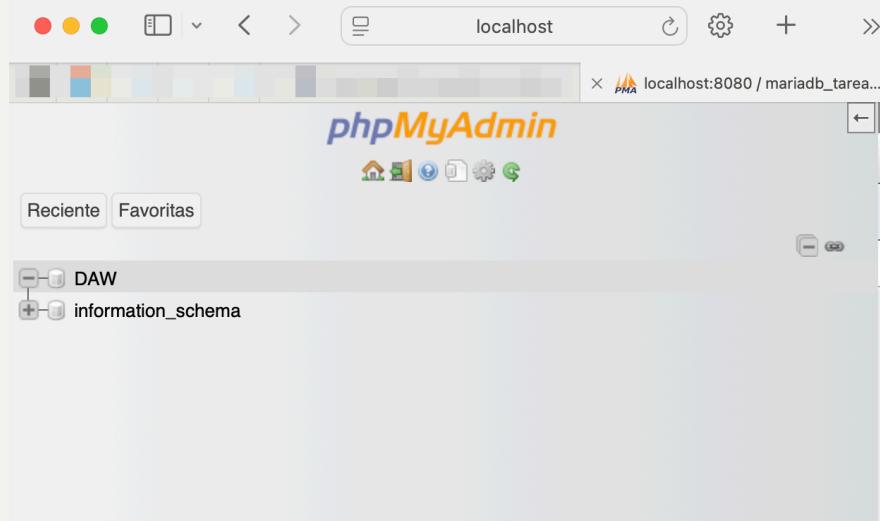
[Pause](#)

[Restart](#)

[Open with browser](#)

En el navegador se abre phpMyAdmin:

Al introducir los datos de usuario `fany` y la contraseña indicada al crear el contenedor de `mariadb_tarea3`:



Para ejecutar el script del archivo `init.sql` primeramente he de importarlo a la base de datos:

Servidor: mariadb_tarea3 » Base de datos: DAW

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Rutinas

Importando en la base de datos "DAW"

Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.
Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**

Buscar en su ordenador: (Máximo: 2,048KB)

Seleccionar archivo init.sql

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

utf-8

Y finaliza la importación:

Importación ejecutada exitosamente, 2 consultas ejecutadas. (init.sql)

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0053 segundos.)

```
CREATE TABLE modulos ( id INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL );
```

[Editar en línea] [Editar] [Crear código PHP]

4 filas insertadas.
La Id de la fila insertada es: 4 (La consulta tardó 0.0036 segundos.)

```
INSERT INTO modulos (nombre) VALUES ('Despliegue de Aplicaciones Web'), ('Programación'), ('Entornos de Desarrollo'), ('Lenguaje de Marcas');
```

[Editar en línea] [Editar] [Crear código PHP]

Donde se puede ver la tabla **módulos**:

The screenshot shows the MySQL Workbench interface with the following details:

- Header:** Servidor: mariadb_tarea3 > Base de datos: DAW
- Toolbar:** Estructura, SQL, Buscar, Generar una consulta, Exportar, Importar, Operaciones, Rutinas, Eventos, Disparadores, Más.
- Filtros:** Que contengan la palabra: [empty input field]
- Table View:** Shows the 'modulos' table with the following data:

Tabla	Acción
modulos	Examinar, Estructura, Buscar, Insertar, Vaciar, Eliminar

1 tabla Número de filas

Filas: 4, Tipo: InnoDB, Cotejamiento: utf8mb4_ucs2_ci, Tamaño: 16.0 KB, Residuo a depurar: 0 B
- Buttons:** Seleccionar todo, Para los elementos que están marcados: [dropdown menu]
- Other:** Imprimir, Diccionario de datos, Crear nueva tabla, Nombre de la tabla, Número de columnas, Crear.

Y al hacer click en **módulos**:

The screenshot shows the MySQL Workbench interface with the following details:

- Message Bar:** ✓ Mostrando filas 0 - 3 (total de 4, La consulta tardó 0.0007 segundos.)
- Query Editor:** SELECT * FROM `modulos`
- Results:** A table with the following data:

	id	nombre
<input type="checkbox"/>	1	Despliegue de Aplicaciones Web
<input type="checkbox"/>	2	Programación
<input type="checkbox"/>	3	Entornos de Desarrollo
<input type="checkbox"/>	4	Lenguaje de Marcas
- Options:** Mostrar todo, Número de filas: 25, Filtrar filas: [dropdown], Buscar en esta tabla, Ordenar según la clave: Ninguna.
- Buttons:** Perfilando [Editar en línea], Opciones extra, Selección todo, Para los elementos que están marcados: [checkbox], Editar, Copiar, Borrar, Exportar.

5. **Enunciado:** Elige entre estas dos opciones:

- Instala la extensión `Disk Usage`, muestra el espacio ocupado, borra algo...
- Instala la extensión `Resource usage` y muestra su salida cuando estén activos los contenedores.

Muestro el contenido de `Resource usage` instalado con los 2 contenedores activos:

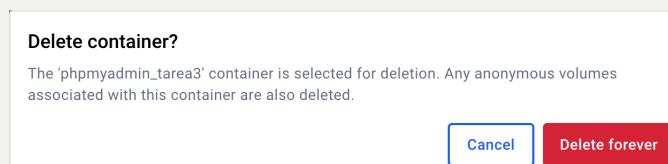
The screenshot shows the Docker Desktop interface. On the left, a sidebar lists various sections: Ask Gordon (BETA), Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, Extensions, Manage, Disk usage, Logs Explorer, PortNavigator, and Resource usage (which is selected). The main area is titled "Resource Usage" with a "Give Feedback" link. It displays real-time monitoring for CPU Cores Usage (Allocated: 8, 0.03% / 800%), Memory Usage (388.64MB / 3.83GB), and Containers (Application containers: 2, System containers: 0). Below this, there are tabs for "Table view" (selected) and "Chart view", and buttons for "Columns" and "Filters". A table lists two containers: "mariadb_tarea3" and "phpmyadmin_tarea3", both in a "running" status. The table includes columns for Name, Status, CPU (%), Memory Us..., MEM (%), Disk Read/..., Network I/O, and PIDS. At the bottom, status indicators show "Engine running", resource usage (RAM 1.96 GB, CPU 1.37%, Disk: 2.22 GB used / limit 1006.85 GB), and links for Terminal and Update.

- Borra los contenedores, la red y los volúmenes utilizados.

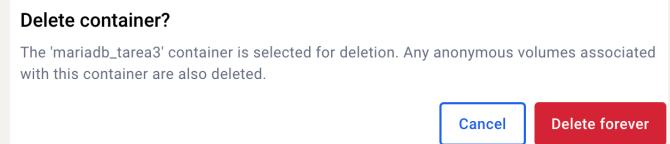
Para borrar los contenedores, lo primero es detenerlos, desde la pestaña de Containers.

The screenshot shows the Docker Desktop Containers table. It lists two containers: "phpmyadmin_tarea3" and "mariadb_tarea3", both marked as "running". Each container row has a delete icon at the end. Two orange arrows point from the text "primero paro los contenedores que están corriendo" to the delete icons of the two containers.

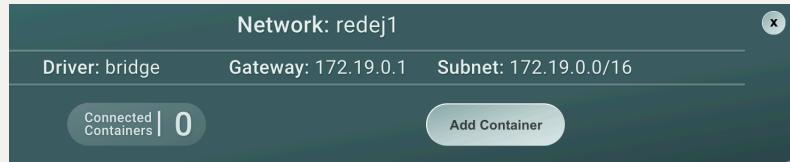
Y luego, click en el ícono de la papelera para borrar el contenedor:



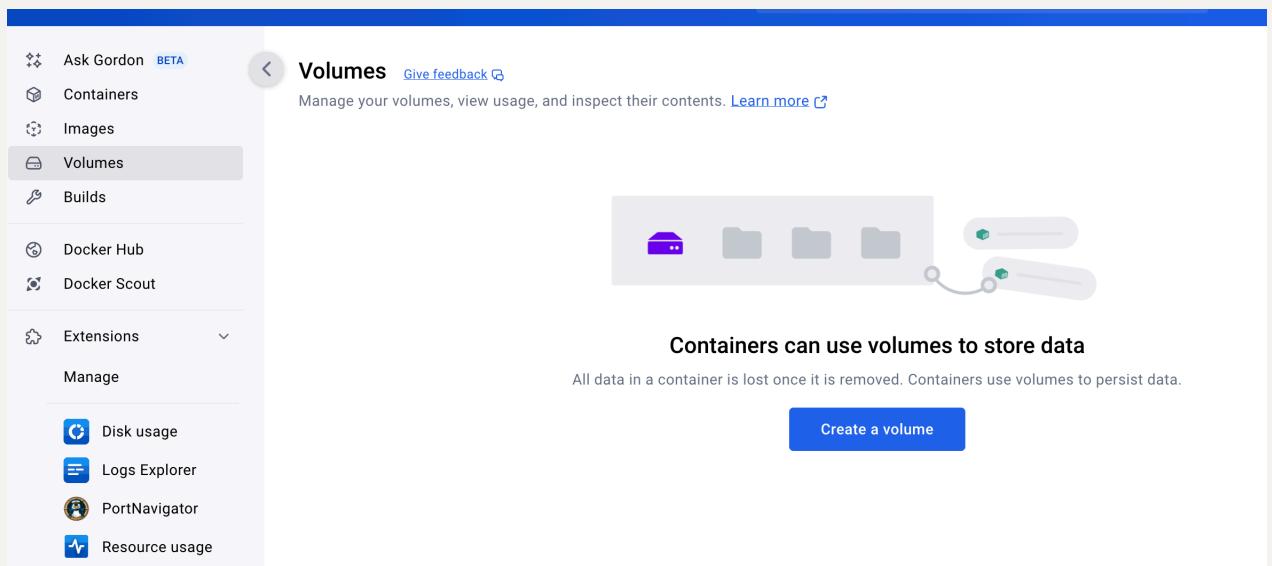
Realizo la misma operación con el contenedor de mariadb_tarea3:



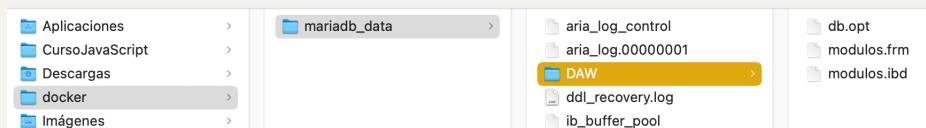
Y una vez que la red `redej1` no tiene contenedores conectados, puedo borrarla, cerrando la ventana en el `x` de la esquina superior derecha:



En cuanto a los volúmenes, en mi caso usé un Bind Mount, así que no aparece como volumen en Docker Desktop:



El directorio que usé para el Bind Mount figura en mi sistema de archivos (fuera de Docker): `users/fanyperalta/docker/mariadb_data` y en la carpeta DAW figura la tabla `módulos` creada.



En este caso, para borrar la carpeta `DAW` tengo que hacerlo manualmente desde la terminal o el Finder.

Ejercicio1 de la Tarea3 de Despliegue.