



Ejercicio 2: Servidor de base de datos

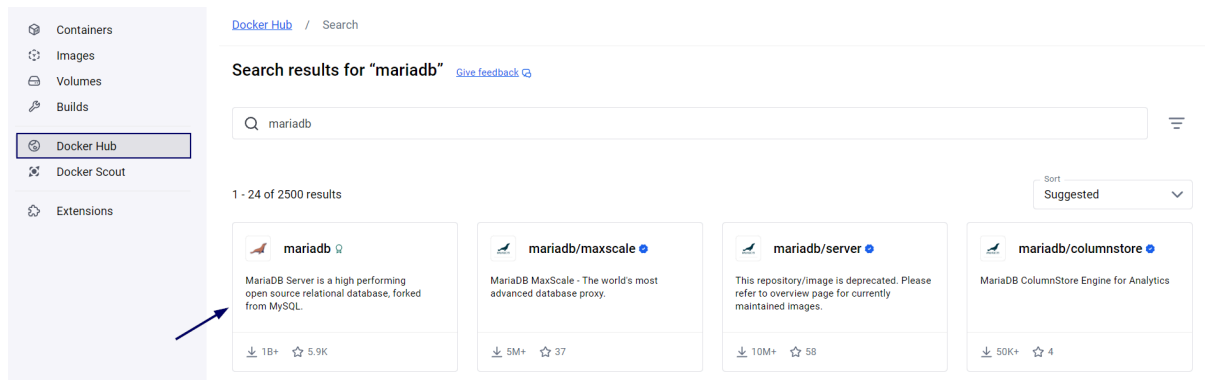
Autores: Ángel Villabrille Fernández, Cristina Mellado Malacara.

1. Abre Docker Desktop. Busca `mariadb` en la selección de imágenes. Selecciona la imagen oficial. Descárgala si no la tienes.
2. Despliega un contenedor utilizando esta imagen:
3. Arranca el contenedor
4. Accede a la base de datos usando una herramienta gráfica, como, por ejemplo `dbeaver`. Conéctate con el usuario `daw`. Crea una base de datos y alguna tabla.
5. Borra el contenedor.
6. Ver en Docker Desktop que el volumen que contiene los datos no se ha borrado.
7. Crear otro contenedor con un servidor de base de datos que use el mismo volumen. Llamar al contenedor `bbdd-2`. Comprobar que la base de datos y la tabla creada anteriormente siguen ahí.
8. Intenta borrar la imagen de `mariadb`. ¿Qué sucede?
9. Borra todo, volumen, imagen y contenedor.

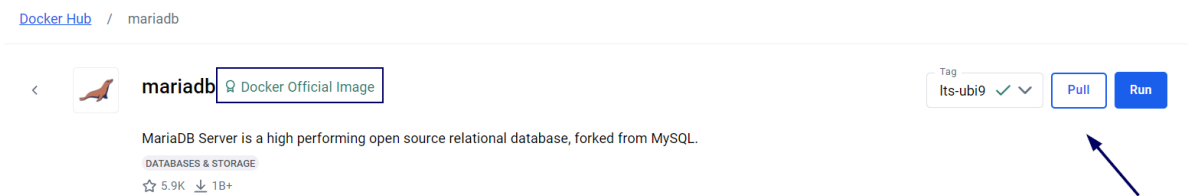
1. Abre Docker Desktop. Busca `mariadb` en la selección de imágenes. Selecciona la imagen oficial.

Descárgala si no la tienes.

- Nos situamos en la pestaña **Docker hub** y buscamos la imagen oficial en el buscador.



- Seleccionamos el botón pull para descargar la imagen. Se puede observar que se bajó correctamente.



- Nos dirigimos a nuestra pestaña **images** para hacer la comprobación de que se encuentra la imagen en la lista.

Images

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

Local Hub repositories

84.74 MB / 10.07 KB in use 3 images

Last refresh: 1 day ago

	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	hello-world	latest	74cc54e27dc4	22 days ago	10.07 KB	▶ ⋮ 🗑
<input type="checkbox"/>	nginx	stable-alpine3.20-perl	25895edb3072	7 days ago	84.74 MB	▶ ⋮ 🗑
<input type="checkbox"/>	mariadb	lts-ubi9	90e4bcac427b	7 days ago	465.31 MB	▶ ⋮ 🗑

2. Despliega un contenedor utilizando esta imagen:

Características:

Nombre del contenedor `bdd`

Puerto: 3306 - debe poder conectarse externamente

Utiliza un volumen llamado `datos-mariadb`

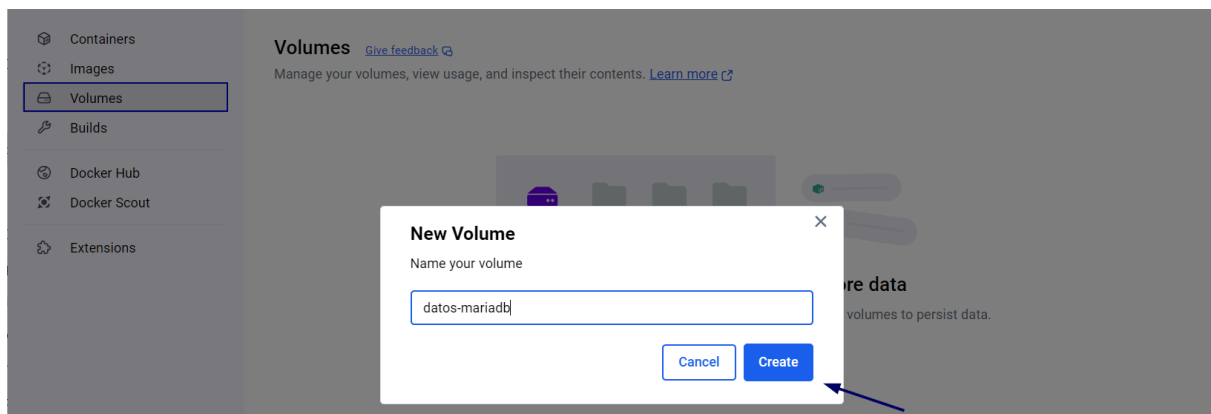
Usa las variables de entorno necesarias para que el usuario `root` tenga la password `root` , la base

de datos por defecto sea `base` , y se cree un usuario `daw` , con password `daw`

- Comprobamos la ruta para proceder a introducir los datos en nuestro contenedor.

```
PS C:\Users\cristi> docker volume inspect datos-mariadb
[
  {
    "CreatedAt": "2025-02-13T08:03:54Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/datos-mariadb/_data",
    "Name": "datos-mariadb",
    "Options": null,
    "Scope": "local"
  }
]
```

- Seleccionamos la pestaña `volumes` y hacemos click en el botón create.



- Configuramos los datos.



Run a new container

mariadb:latest-ubi9

Container name

bbdd

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

Ports

Enter "0" to assign randomly generated host ports.

Host port

3306

:3306/tcp

Volumes

Host path

datos-mariadb

...

Container path

/var/lib/mysql

+

Environment variables

Variable

MARIADB_ROOT_PASSWORD

Value

root

—

Variable

MARIADB_DATABASE

Value

base

—

Variable

MARIADB_USER

Value

daw

—

Variable

MARIADB_PASSWORD

Value

daw

+

Cancel

Run

- Comprobamos en Docker Desktop que se creó el contenedor y se está haciendo uso del volumen `datos-mariadb`.

Containers [Give feedback](#)

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

Container CPU usage

No containers are running.

Container memory usage

No containers are running.

[Show charts](#)

Q Search

Only show running containers

<div></div>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<div><div><div></div></div><div><div></div></div></div>	bbdd	5fd5d681d5e6	mariadb:its-ubi9	3306:3306	N/A	6 minutes ago	<div><div></div><div></div><div></div></div>

Volumes [Give feedback](#)

Manage your volumes, view usage, and inspect their contents. [Learn more](#)

Q

Search

Create

<input type="checkbox"/>	Name ↑	Created	Size	Actions
<input type="checkbox"/>	<div><div></div>datos-mariadb</div>	1 hour ago	0 Bytes	<div><div></div><div></div></div>

3. Arranca el contenedor

- Pulsamos el botón de arrancar y comprobamos que funcionó con éxito

Containers [Give feedback](#)

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

Container CPU usage

0.03% / 800% (8 CPUs available)

Container memory usage

252.4MB / 7.23GB

[Show charts](#)

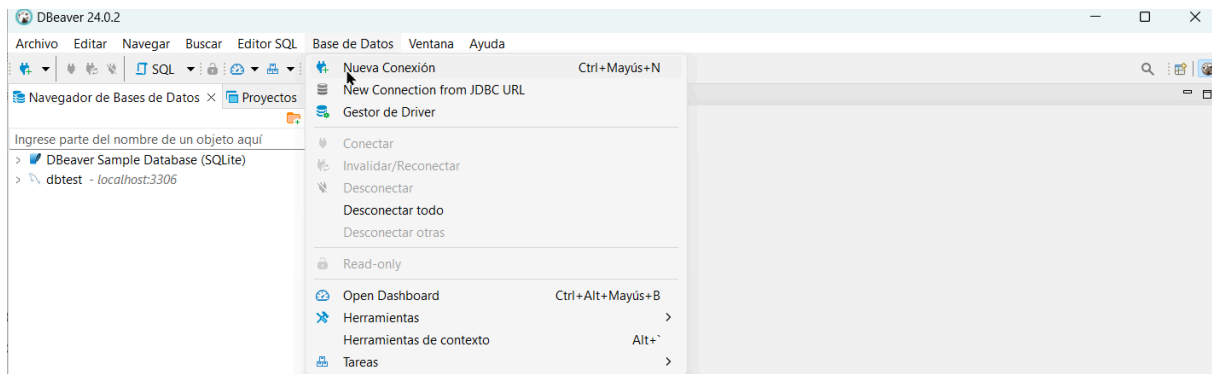
Q Search

Only show running containers

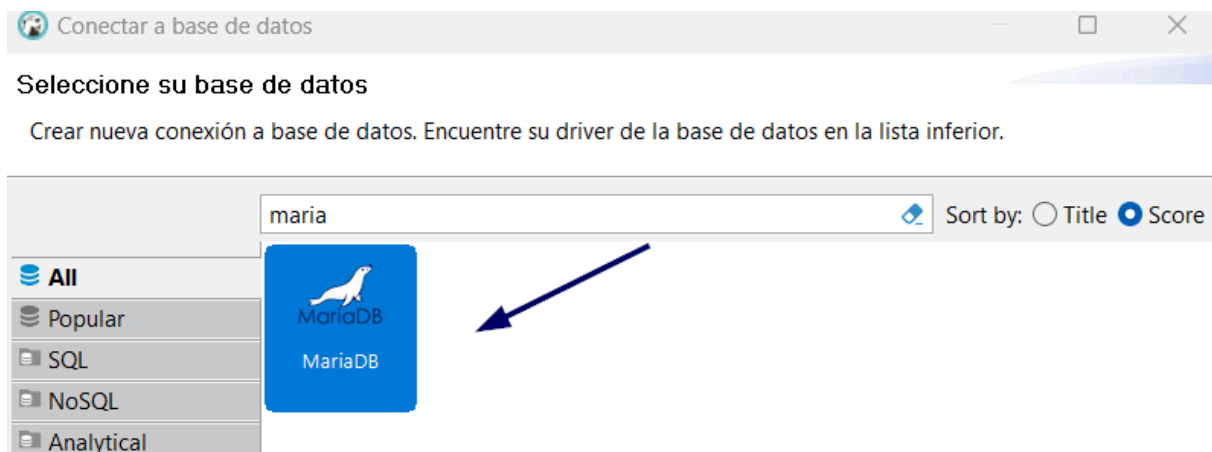
<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<div><div></div>bbdd</div>	5fd5d681d5e6	mariadb:its-ubi9	3306:3306	0.03%	0 seconds ago	<div><div></div><div></div><div></div></div>

4. Accede a la base de datos usando una herramienta gráfica, como, por ejemplo **dbeaver**. Conéctate con el usuario **daw**. Crea una base de datos y alguna tabla.

- Accedemos a **dbeaver** y pulsamos en nueva conexión.



- Seleccionamos nuestra base de datos **mariadb**




- Me conecto exitosamente con el usuario **daw** .

Configuración de la conexión "base"

Ajustes de conexión

MariaDB ajustes de conexión



▼ Ajustes de conexión

- Inicialización
- Comandos de shell
- Identificación de usuarios
- Transacciones
- General
- Metadatos
- Errores y timeouts
- Data Transfer
- Data Editor
- Editor SQL

General | Driver properties | SSH | SSL | + Network configurations...

Server

Connect by: ☒ Host ☐ URL

URL: jdbc:mariadb://localhost:3306/base

Server Host: localhost Port: 3306

Database: base

Authentication (Database Native)

Nombre de usuario: daw

Contraseña: ... ☒ Save password

Advanced

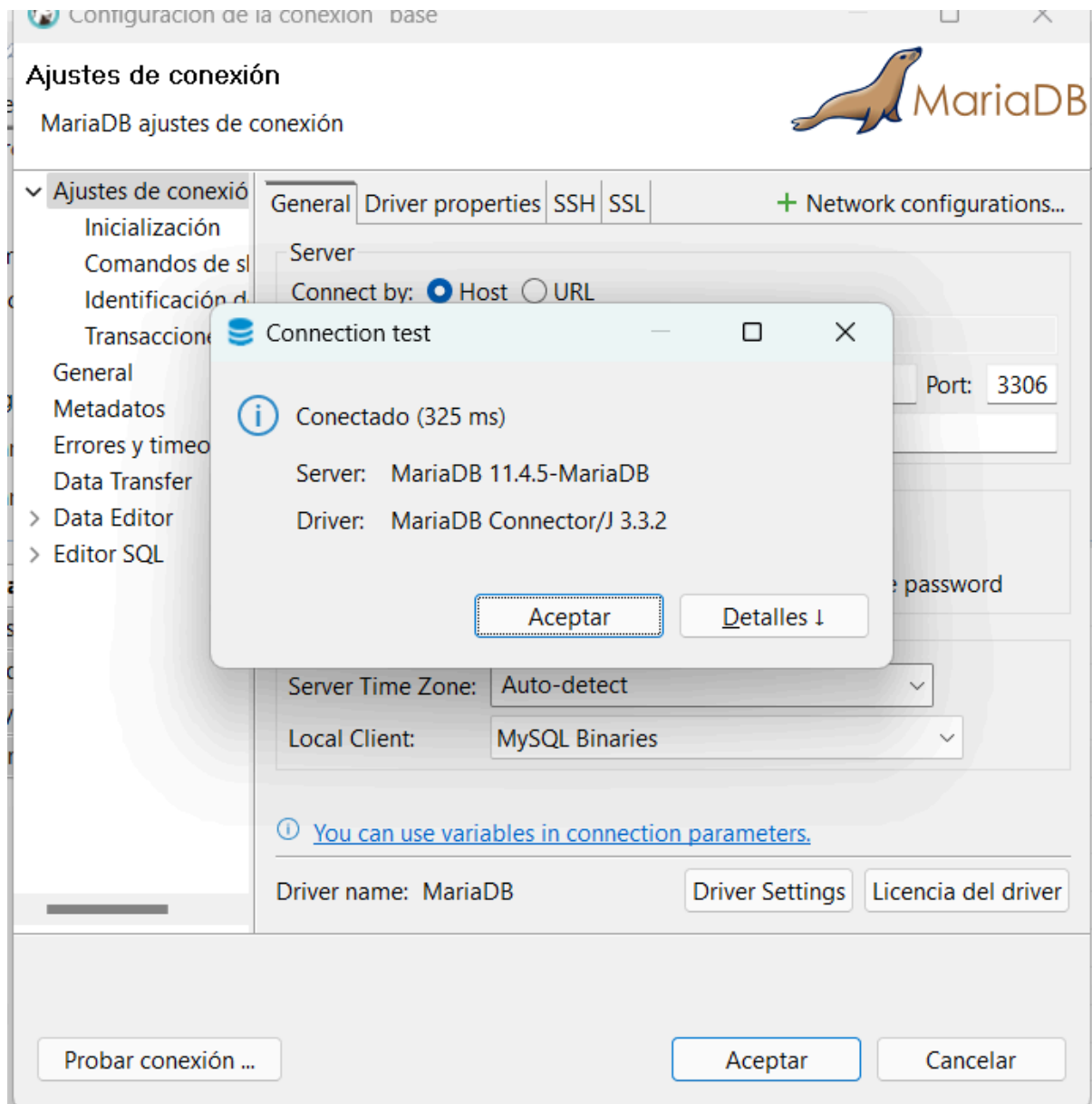
Server Time Zone: Auto-detect

Local Client: MySQL Binaries

[You can use variables in connection parameters.](#)

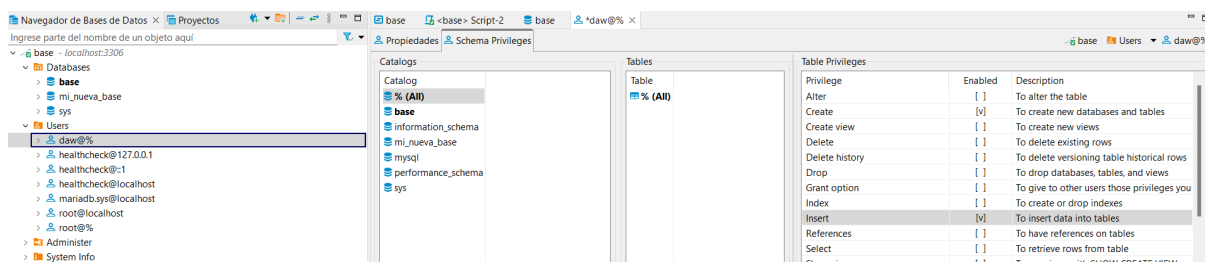
Driver name: MariaDB Driver Settings Licencia del driver

Probar conexión ... Aceptar Cancelar



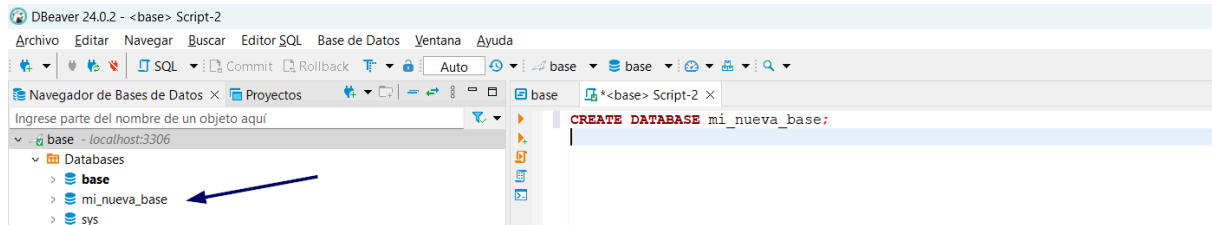
c. Creamos una base de datos y una tabla.

- Tenemos que otorgarle permisos al usuario **daw** para que pueda crear base de datos y tabla, eso lo hacemos desde nuestro usuario **root**.



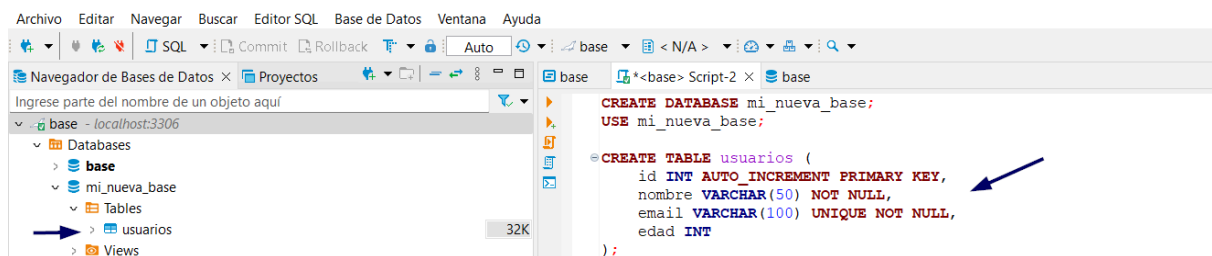
- Ahora que tenemos los permisos, volvemos a nuestro usuario `daw` , y creamos la base de datos `mi_nueva_base`.

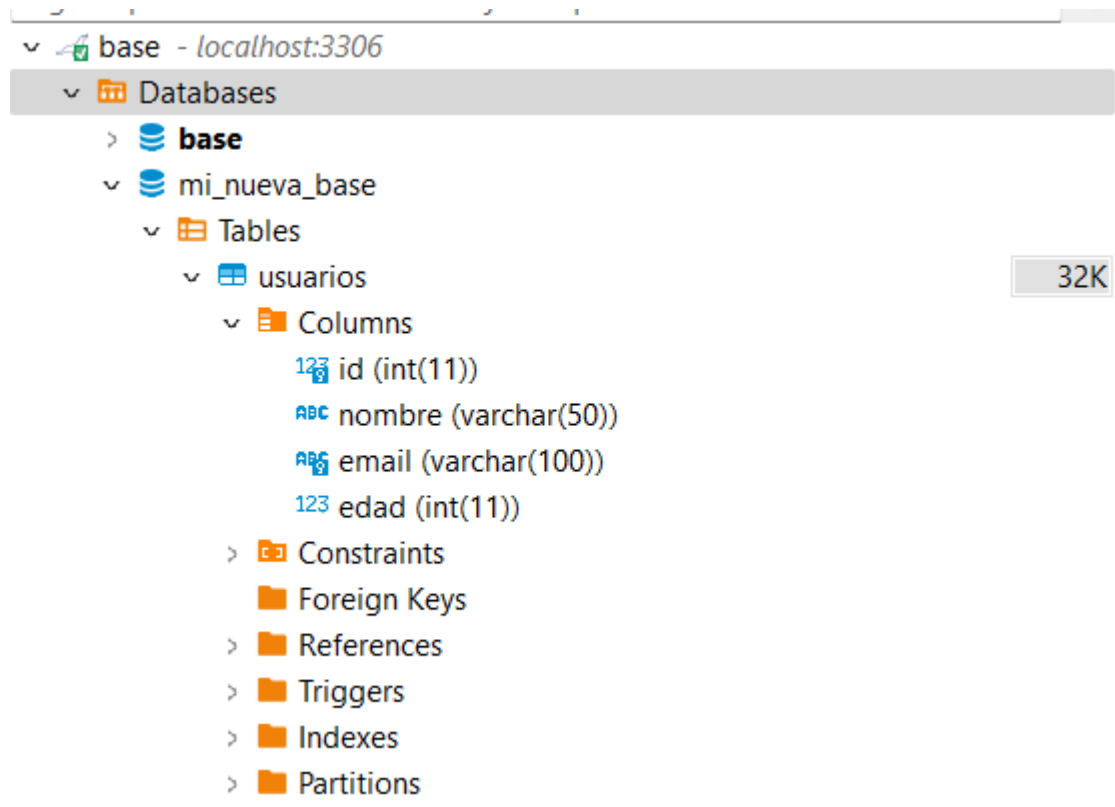
```
CREATE DATABASE mi_nueva_base;
```



- Creamos una tabla llamada `usuarios` y comprobamos que se creó correctamente.

```
CREATE TABLE usuarios (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(50) NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  edad INT
);
```





5. Borra el contenedor.

Containers [Give feedback](#)

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

Container CPU usage
No containers are running.

Container memory usage
No containers are running.

[Show charts](#)

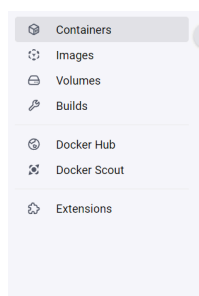
Search

Only show running containers

Delete

Play Pause Stop

<input checked="" type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input checked="" type="checkbox"/>	bbdd	5fd5d681d5e6	mariadb:10.5-ubi9	3306:3306	N/A	4 hours ago	Play ... Stop



Containers [Give feedback](#)

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

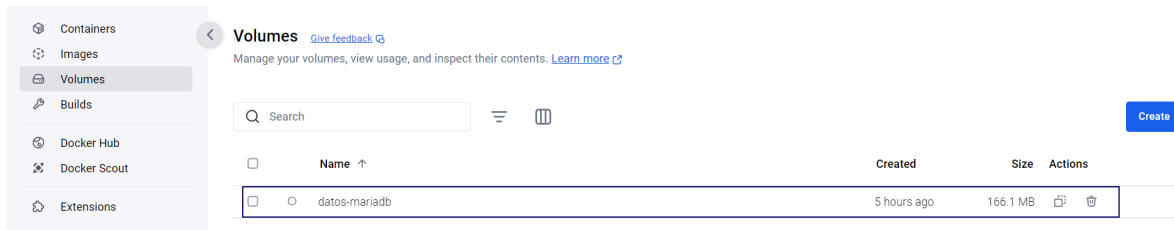
NOS DAMOS CUENTA QUE YA NO APARECE EL CONTENEDOR.

Your running containers show up here

A container is an isolated environment for your code

6. Ver en Docker Desktop que el volumen que contiene los datos no se ha borrado.

- En la imagen se puede observar que no se eliminó el volumen `datos-mariadb`



7. Crear otro contenedor con un servidor de base de datos que use el mismo volumen. Llamar al contenedor `bbdd-2`. Comprobar que la base de datos y la tabla creada anteriormente siguen ahí.

- Creamos el nuevo contenedor



Run a new container

mariadb:latest-ubi9

Container name

bbdd-2

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

Ports

Enter "0" to assign randomly generated host ports.

Host port

3306

:3306/tcp

Volumes

Host path

datos-mariadb

...

Container path

/var/lib/mysql

+

Environment variables

Variable

MARIADB_ROOT_PASSWORD

Value

root

—

Variable

MARIADB_DATABASE

Value

base

—

Variable

MARIADB_USER

Value

daw

—

Variable

MARIADB_USER_PASSWORD

Value

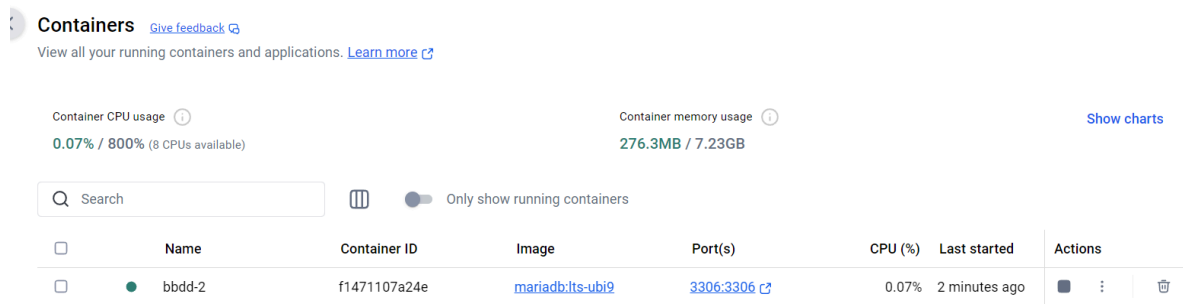
daw

+

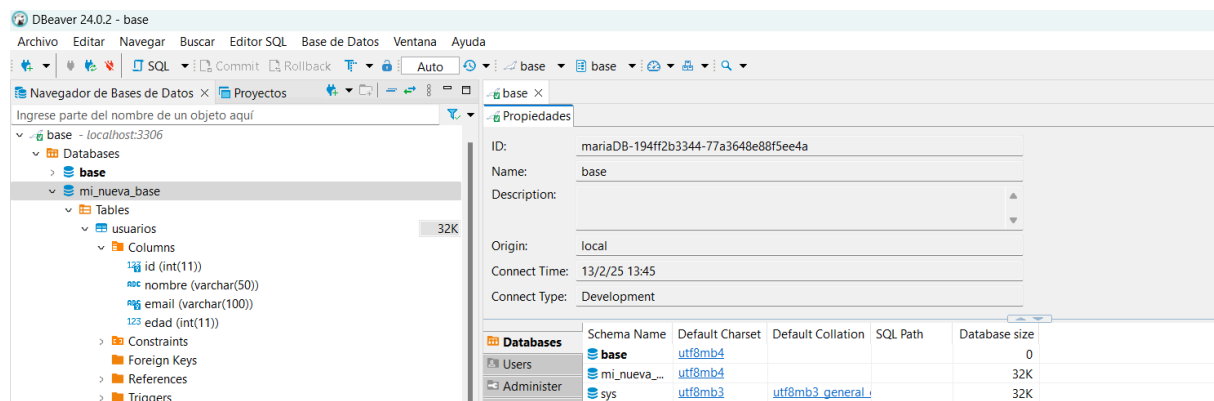
Cancel

Run

- Comprobamos que arrancó correctamente.



- Se comprueba que la base de datos y la tabla creada anteriormente siguen ahí.

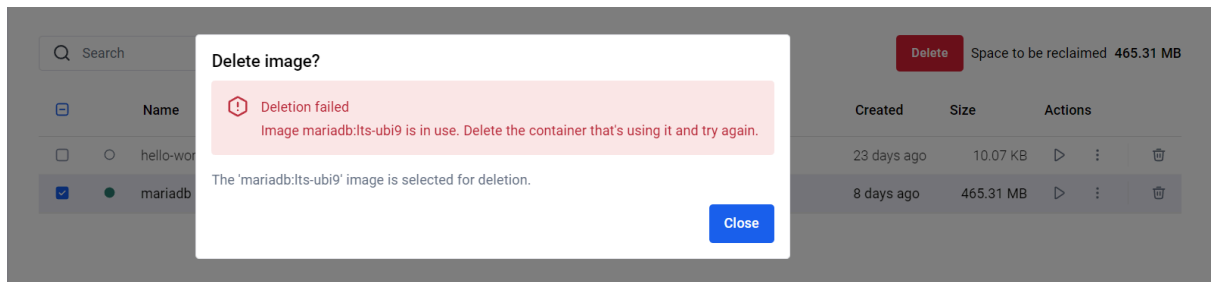


8. Intenta borrar la imagen de **mariadb** ¿Qué sucede?

- Al intentar eliminar la imagen **mariadb** nos pone el siguiente mensaje:

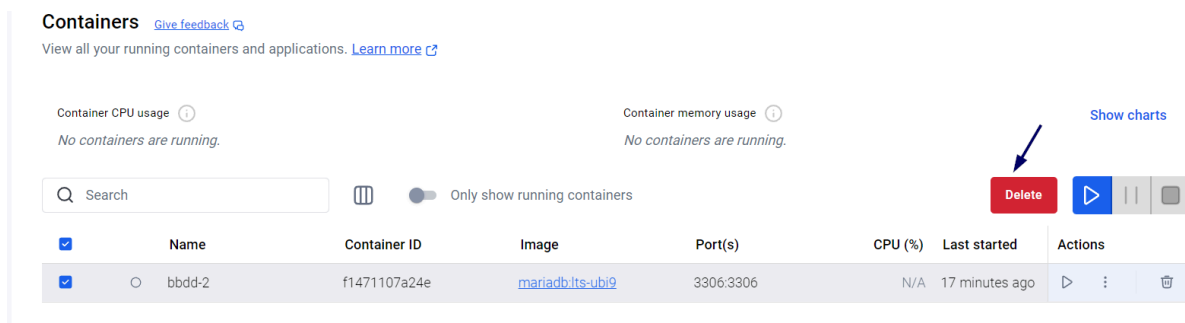
Error al eliminar: La imagen mariadb:its-ubi9 está en uso. Elimina el contenedor que la está usando y vuelve a intentarlo.

Esto sucede porque Docker no permite borrar una imagen si hay algún contenedor que la está utilizando. Para poder eliminarla, primero es necesario detener y borrar todos los contenedores que dependen de esa imagen.

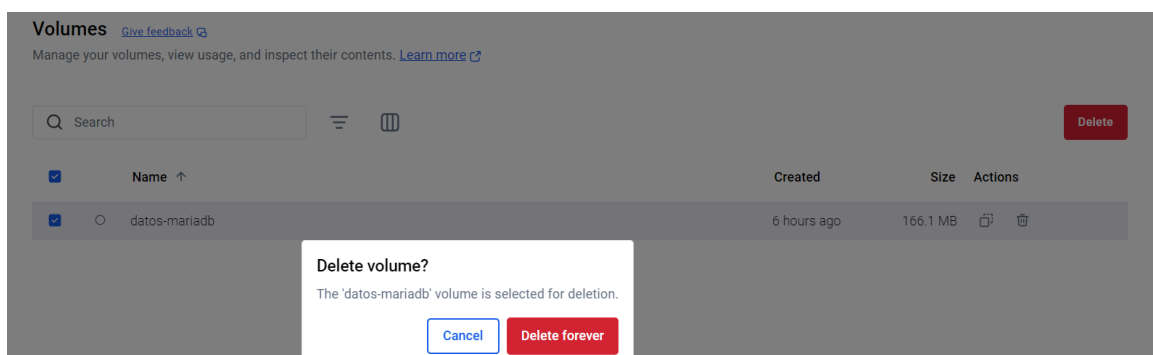


9. Borra todo, volumen, imagen y contenedor.

- Eliminamos primero el contenedor: Para que posteriormente nos permita eliminar el volumen y después la imagen.



- Eliminamos el volumen.



- Eliminamos la imagen.

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

Local Hub repositories

0 Bytes / 550.07 MB in use 2 Images



Delete

Space to be reclaimed **465.31 MB**

Last refresh: 6 hours ago

		Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	<input type="radio"/>	hello-world	latest	74cc54e27dc4	23 days ago	10.07 KB	<div><div></div><div></div><div></div></div>
<input checked="" type="checkbox"/>	<input type="radio"/>	mariadb	lts-ubi9	90e4bcac427b	8 days ago	465.31 MB	<div><div></div><div></div><div></div></div>