



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

ÍNDICE

-  [Ejercicio 3: Contenedores en Red](#)
-  [Crear una red bridge \(redbd\)](#)
-  [Crear el contenedor de MariaDB con comandos](#)
-  [Crear el contenedor de MariaDB con Docker Desktop](#)
-  [Crear el contenedor con Adminer o phpMyAdmin con comandos](#)
-  [Crear el contenedor con Adminer o phpMyAdmin con Docker Desktop](#)
-  [Acceder a la base de datos desde la interfaz gráfica](#)
-  [Borrar los contenedores, la red y los volúmenes utilizados con comandos](#)
-  [Borrar los contenedores, la red y los volúmenes utilizados con Docker Desktop](#)

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
```

```
C:\Windows\System32>docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
a442ae9f5d9b        bridge              bridge              local
5ad4578655cc        host                host                local
f0d37d297057        none                null                local

C:\Windows\System32>docker network create redbd
9dbf0118bc4ca006697b866f3f87478e13ae0c36bf533f85b52e9fe3ea7d8834
```

```
C:\Windows\System32>docker network inspect redbd
[
  {
    "Name": "redbd",
    "Id": "9dbf0118bc4ca006697b866f3f87478e13ae0c36bf533f85b52e9fe3ea7d8834",
    "Created": "2025-02-12T18:19:37.719807195Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
```

Crear el contenedor de MariaDB con comandos

Ejecutamos en la terminal:

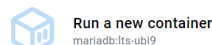
```
$docker run -d --name mariadb-container \
  --network redbd \
  -e MYSQL_ROOT_PASSWORD=root \
  -e MYSQL_DATABASE=base \
  -e MYSQL_USER=daw \
  -e MYSQL_PASSWORD=daw \
  -p 3306:3306 \
  -v mariadb_data:/var/lib/mysql \
  mariadb:latest
```

```

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



Optional settings

Container name
mariadb-container

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

Container path

Environment variables

Variable

MYSQL_ROOT_USER

Value

root

Variable

MYSQL_ROOT_PASSWORD

Value

root

Variable

MYSQL_DATABASE

Value

base

Variable

MYSQL_USER

Value

daw

Variable

MYSQL_PASSWORD

Value

daw

Cancel

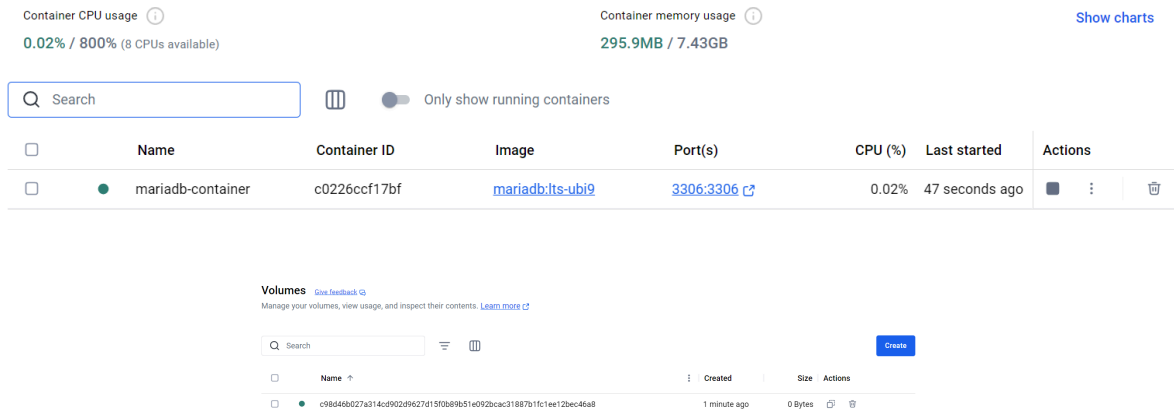
Run

```

2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Shutdown completed; log sequence number 47875; transaction id 15
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] mariadb: Shutdown complete
2023-02-12 18:48:08 2023-02-12 18:48:49-00:00 [Note] [Entrypoint]: Temporary server stopped
2023-02-12 18:48:08
2023-02-12 18:48:08 2023-02-12 18:48:49-00:00 [Note] [Entrypoint]: MariaDB Unit process done. Ready for start up.
2023-02-12 18:48:08
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] Starting MariaDB 11.4.5-MariaDB source revision 077110266ff9c426d46f12c0d5f8d7c0cf server_id 137948a0d8e
1504080000 as process 1
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Compressed tables use zlib 1.2.13
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Number of transaction pools: 1
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Using XTRX instructions
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] mariadb: 0.99924 is not supported on /tmp (disabling future attempts)
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Using libaio
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Initializing buffer pool, total size = 128.000MB, chunk size = 2.000MB
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Completed initialization of buffer pool
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: File system buffers for log disabled (flush size=4096 bytes)
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: End of log at LSN=47875
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Opened 3 undo tablespaces
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: 128 rollback segments to 3 undo tablespaces are active.
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Setting File './ibtmp1' size to 12.000MB. Physically writing the file full; Please wait ...
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: File './ibtmp1' size is now 12.000MB.
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Log sequence number 47875; transaction id 15
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Loading buffer pool(s) from /var/lib/mariadb/ib_buffer_pool.
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] Plugin 'FEDERATED' is disabled.
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] Plugin 'rpl_semi_sync_rep_provider' is disabled.
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] InnoDB: Buffer pool(s) load completed at 2023-02-12 18:48:49
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] Server socket created on IP: '0.0.0.0'
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] Server socket created on IP: '::'
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] mariadb: Event Scheduler: Loaded 0 events
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] mariadb: ready for connections.
2023-02-12 18:48:08 2023-02-12 18:48:49 # [Note] mariadb: version: '11.4.5-MariaDB' socket: '/run/mariadb/mariadb.sock' port: 3306 MariaDB Server

```

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

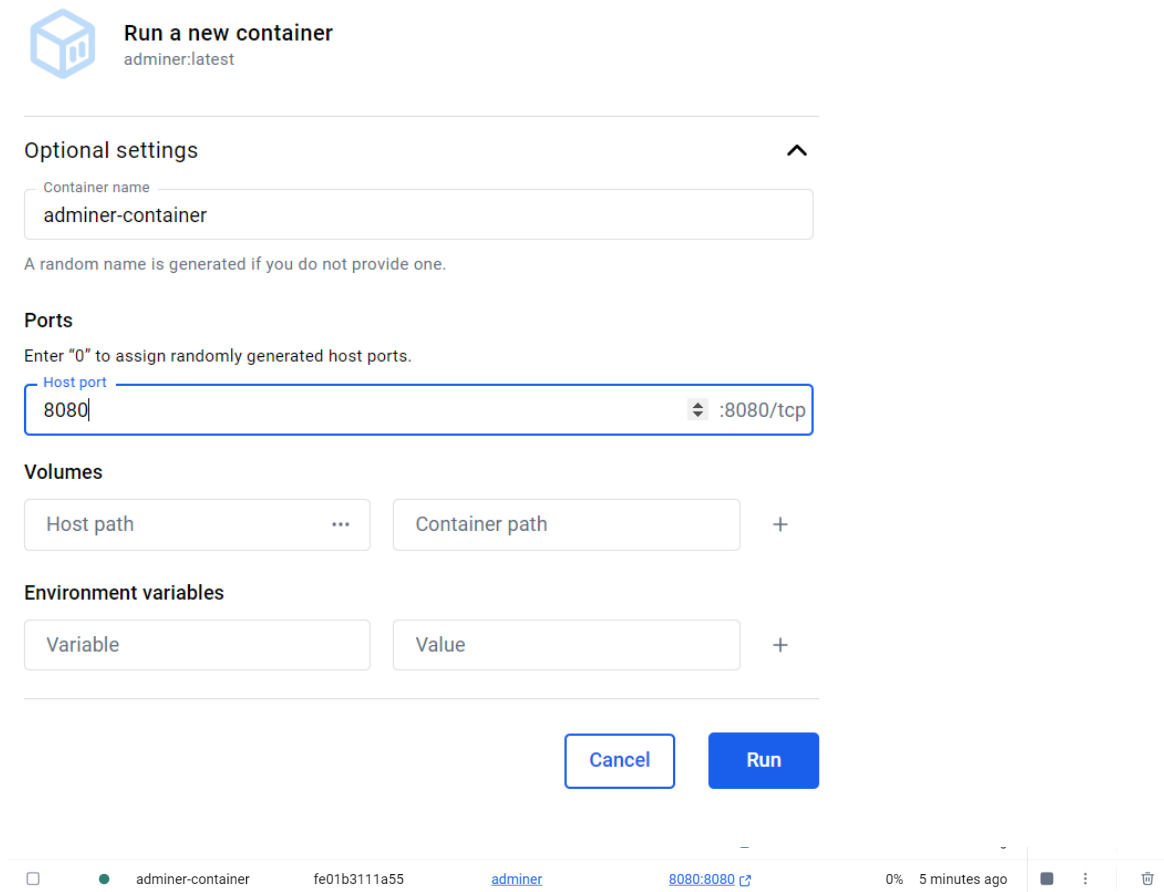
```
$docker run -d --name adminer-container \
--network redbd \
-p 8080:8080 \
adminer
```

```
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"**.



Run a new container
adminer:latest

Optional settings ^

Container name
adminer-container

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

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

Host port
8080 :8080/tcp

Volumes

Host path ... Container path +

Environment variables

Variable Value +

Cancel Run

Container	ID	Image	Ports	Status	Time	Actions
adminer-container	fe01b3111a55	adminer	8080:8080	Running	0% 5 minutes ago	Stop, Restart, Delete

Acceder a la base de datos desde la interfaz gráfica

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

← → ↻ 🏠 ⓘ localhost:8080

Idioma: Español ▼

Adminer 4.8.1

Login

Motor de base de datos	MySQL ▼
Servidor	<input type="text" value="db"/>
Usuario	<input type="text"/>
Contraseña	<input type="password"/>
Base de datos	<input type="text"/>

☐ Guardar contraseña

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

Idioma: Español ▼

Adminer 4.8.1

Login

Motor de base de datos	MySQL ▼
Servidor	<input type="text" value="mariadb-container"/>
Usuario	<input type="text" value="daw"/>
Contraseña	<input type="password" value="..."/>
Base de datos	<input type="text" value="base"/>

☒ Guardar contraseña

← → ↻ 🏠 localhost:8080/?server=mariadb-container&username=daw&db=base

Idioma: Español

MySQL » mariadb-container » Base de datos: base

Adminer 4.8.1

DB: base

[Comando SQL](#) [Importar](#)
[Exportar](#) [Crear tabla](#)

No existen tablas.

Base de datos: base

[Modificar Base de datos](#) [Esquema de base de datos](#) [Privilegios](#)

Tablas y vistas

No existen tablas.

[Crear tabla](#) [Crear vista](#)

Procedimientos

[Crear procedimiento](#) [Crear función](#)

Eventos

[Crear Evento](#)

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.

Idioma: Español

MySQL » mariadb-container » base » Crear tabla

Adminer 4.8.1

DB: base

[Comando SQL](#) [Importar](#)
[Exportar](#) [Crear tabla](#)

No existen tablas.

Crear tabla

Nombre de la tabla: ejemplo (motor) (colación) Guardar

Nombre de columna	Tipo	Longitud	Opciones	NULL	AI?	+
id	int			<input type="checkbox"/>	<input checked="" type="radio"/>	+ ↑ ↓ ✕
nombre	varchar	50	(colación)	<input type="checkbox"/>	<input type="radio"/>	+ ↑ ↓ ✕

Incremento automático: ☐ Valores predeterminados ☐ Comentario

Guardar

Particionar por

Idioma: Español

MySQL » mariadb-container » base » Tabla: ejemplo

Adminer 4.8.1

DB: base

[Comando SQL](#) [Importar](#)
[Exportar](#) [Crear tabla](#)

[registros ejemplo](#)

Tabla: ejemplo

Tabla creada, 19:28:31 [Comando SQL](#)

[Visualizar contenido](#) [Mostrar estructura](#) [Modificar tabla](#) [Nuevo Registro](#)

Columna	Tipo	Comentario
id	int(11) <i>Incremento automático</i>	
nombre	varchar(50)	

Índices

PRIMARY id

[Modificar índices](#)

Claves externas

[Agregar clave externa](#)

Disparadores

[Agregar disparador](#)

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.

Container CPU usage ⓘ











0.01% / 800% (8 CPUs available)

Container memory usage ⓘ

214.33MB / 7.43GB

Show charts










Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Stop actions
<input type="checkbox"/>	 mariadb-container	235dfa331863 	mariadb:latest		0%	12 seconds ago	  
<input type="checkbox"/>	 adminer	c5eb9aaabaf8	adminer	8080:8080 	0%	4 seconds ago	  







Y luego los borramos:

Search

Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions	Delete
<input type="checkbox"/>	 mariadb-container	235dfa331863 	mariadb:latest		N/A	1 minute ago	 	
<input type="checkbox"/>	 adminer	c5eb9aaabaf8	adminer	8080:8080	N/A	59 seconds ago	 	

Una vez borrados podemos borrar sus imagenes:

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions	Delete
<input type="checkbox"/>	 mariadb	latest	bfb1298c06cd	3 months ago	562.31 MB	 	
<input type="checkbox"/>	 adminer	latest	34d37131366c 	7 months ago	374.3 MB	