1 Creación de un contenedor Docker con MariaDB

1.1 Requisitos

Para poder ejecutar contenedores Docker es necesario tener instalado Docker Community Edition (CE) en nuestra máquina virtual. En concreto la instalaremos en un sistema operativo Debian 12.7

1.2 Descargar la imagen de mariadb

Par poder descargar la imagen de mariadb de DockerHub usaremos el comando:

```
docker pull mariadb
```

Si usamos el comando `docker images` comprobamos las imágenes que tenemos descargadas en nuestro repositorio local

1.3 Cómo crear un contenedor

Para crear un contenddo Docker utilzaremos `docker run`. Si no decimos nada el contenedor que creemos no tendrá persistencia de datos. Un contenedor Docker que no tiene persistencia de datos quiere decir que cuando finalice la ejecución perderá todo el contenido que hayamos creado durante la ejecución. Es decir, si durante la ejecución del contenedor hemos creado varias bases de datos en MariaDB, éstas se perderán cuando el contenedor se detenga.

El comando que podríamos usar para lanzar nuestro contenedor Docker con MariaDB sin persistencia de datos podría ser el siguiente:

```
docker run -d --rm --name migbd -e MARIADB_ROOT_PASSWORD=root -p 3306:3306 mariadb
```

- `docker run` es el comando que nos permite crear un contenedor a partir de una imagen Docker.
- El parámetro `-d` nos permite ejecutar el contenedor en modo detached, es decir, ejecutándose en segundo plano.
- El parámetro `--rm` hace que cuando salgamos del contenedor, éste se elimine y no ocupe espacio en nuestro disco.
- El parámetro `--name` nos permite asignarle un nombre a nuestro contenedor. Si no le asignamos un nombre Docker nos asignará un nombre automáticamente.
- El parámetro `-e` es para pasarle al contenedor una variable de entorno. En este caso le estamos pasando la variable de entorno MYSQL_ROOT_PASSWORD con el valor de la contraseña que tendrá el usuario root para MySQL Server.

- El parámetro `-p` nos permite mapear los puertos entre nuestra máquina local y el contenedor. En este caso, estamos mapeando el puerto 3306 de nuestra máquina local con el puerto 3306 del contenedor.
- `mariadb` es el nombre de la imagen. Si no indicamos la versión utilizará la última disponible. Si no se indica lo contrario buscará las imágenes en el repositorio oficial Docker Hub.

Una vez realizado el comando podemos ver todos los contenedores que tenemos en ejecución utilizando:

```
docker ps -a
```

En este momento el sistema gestor de base de datos está enlazado y ejecutándose y ya podríamos empezar a crear bases de datos.

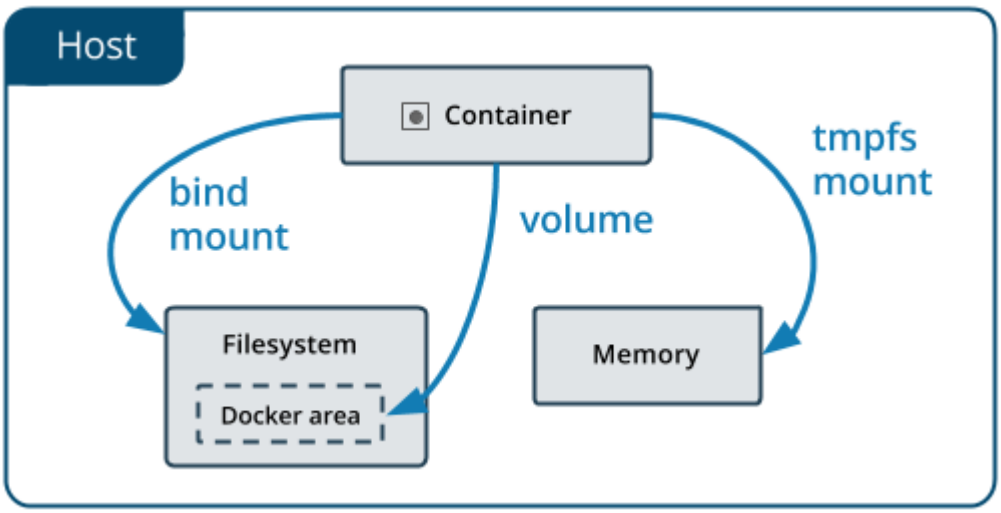
1.4 Cómo crear un contenedor con persistencia de datos

Si queremos que los datos del contenedor sean persistentes tenemos que crear un volumen donde vamos a indicar el directorio de nuestra máquina local donde vamos a almacenar el directorio `/var/lib/mysql`, que es el directorio que utiliza MariaDB para almacenar las bases de datos.

Para crear un volumen utilizamos el parámetro `-v`.

Docker nos ofrece dos posibilidades para implementar persistencia de datos en los contenedores:

1. `bind mount`: pueden estar almacenados en cualquier directorio del sistema de archivos de la máquina host. Estos archivos pueden ser consultados o modificados por otros procesos de la máquina host o incluso por otros contenedores Docker.
2. `volume`: se almacenan en la máquina host dentro del área del sistema de archivos que gestiona Docker. Otros procesos de la máquina host no deberían modificar estos archivos, sólo deberían ser modificados por contenedores Docker.



Ejemplo de uso del parámetro `-v` para crear un volumen de tipo `bind_mount` :

```
-v /home/alumno/bbdd:/var/lib/mysql
```

En este caso el directorio `/home/alumno/bbdd` de nuestra máquina local estará sincronizado con el directorio `/var/lib/mysql` del contenedor con MariaDB. Cuidado!!! el directorio bbdd debe de existir en nuestra

máquina virtual. Si no existe créala utilizando el comando `mkdir /home/alumno/bbdd`

El comando que podríamos usar para lanzar nuestro contenedor Docker con MariaDB con persistencia de datos en un volumen podría ser el siguiente:

```
docker run -d --rm --name migbd -e MARIADB_ROOT_PASSWORD=root -p 3306:3306 -v
/home/alumno/bbdd:/var/lib/mysql mariadb
```

Ejemplo de uso del parámetro `-v` para crear un volumen de tipo `volume`:

1. Primero tenemos que crear un volumen docker:

```
 docker volume create datos_mariadb
```

2. Uso del parámetro `-v` con un volumen de tipo `volume`:

`-v datos_mariadb:/var/lib/mysql`

3. El comando que podríamos usar para lanzar nuestro contenedor Docker con MariaDB con persistencia de datos en un volumen podría ser el siguiente:

```
docker run -d --rm --name migbd -e MARIADB_ROOT_PASSWORD=root -p 3306:3306 -v
mariadb_data:/var/lib/mysql mariadb
```

- `docker run` es el comando que nos permite crear un contenedor a partir de una imagen Docker.
- El parámetro `-d` nos permite ejecutar el contenedor en modo detached, es decir, ejecutándose en segundo plano.
- El parámetro `--rm` hace que cuando salgamos del contenedor, éste se elimine y no ocupe espacio en nuestro disco.
- El parámetro `--name` nos permite asignarle un nombre a nuestro contenedor. Si no le asignamos un nombre Docker nos asignará un nombre automáticamente.
- El parámetro `e` es para pasarle al contenedor una variable de entorno. En este caso le estamos pasando la variable de entorno `MARIADB_ROOT_PASSWORD` con el valor de la contraseña que tendrá el usuario root para MARIADB Server.
- El parámetro `-p` nos permite mapear los puertos entre nuestra máquina local y el contenedor. En este caso, estamos mapeando el puerto `3306` de nuestra máquina local con el puerto `3306` del contenedor.
- El parámetro `-v` nos permite crear un volumen para tener persistencia de datos al finalizar el contenedor.

- `mariadb` es el nombre de la imagen. Si no indicamos la versión utilizará la última versión disponible que está etiquetada como `latest`. Si no se indica lo contrario buscará las imágenes en el repositorio oficial Docker Hub.

1.5 Cómo comprobar que el contenedor está en ejecución

Una vez que hemos iniciado el contenedor podemos comprobar que se está ejecutando con el siguiente comando:

```
docker ps
```

Deberíamos obtener una salida similar a esta.

```
|CONTAINER ID   | IMAGE      | COMMAND                  | CREATED       | STATUS
| PORTS                    |  NAMES
|52bcaee9a157   | mariadb    | "docker-entrypoint.s…"   | 4 seconds ago | Up 2
seconds    | 0.0.0.0:3306->3306/tcp |  migbd
```

1.6 Cómo conectar con MariaDB

Una vez que MariaDB está en ejecución podemos conectarnos con cualquier cliente: MySQL Workbench, PHPMyAdmin, Adminer, etc.

Los datos de conexión serán:

```
Host: 127.0.0.1
Puerto: 3306
Usuario: root
Password: root
```

1.6.1 Acceder mediante una consola interactiva

1. Usamos el comando docker exec para acceder a la consola de un contenedor en ejecución de forma interactiva. La sintaxis general es la siguiente:

```
docker exec -it <nombre_o_id_del_contenedor> bash
```

En nuestro caso como el contenedor se llama migbd:

```
docker exec -it migbd bash
```

Este comando te llevará a una terminal interactiva dentro del contenedor.

2. Ejecutar el cliente de MariaDB: Una vez dentro de la terminal del contenedor escribimos

```
mariadb -u root -proot
```

Utilizaremos el parámetro `-p` para especificar la ocontraseña. No debemos dejar ningún espacio en blanco. Esto abrirá el cliente de MariaDB.

3. Comprobar las bases de datos creadas

```
show databases;
```

4. Salir del cliente y del contenedor: Para salir del cliente de MariaDB, escribe el comando `exit` o `quit` o presiona `Ctrl + D`. Para salir de la consola del contenedor, también puedes usar `exit` o `Ctrl + D`.

1.6.2 Utilizando un cliente de mariadb

1. Instalar el cliente de mariadb: Para instalar el cliente de mariadb en nuestra máquina virtual debian,buscaremos usando el comando `apt search` los paquetes que podemos instalar. Cuando ya lo hemos encontrado utilizamos el comando:

```
sudo apt install mariadb-client
```

2. Verificamos la instalación: Una vez que se haya instalado, puedes verificarlo ejecutando el cliente de MariaDB con el siguiente comando

```
mysql --version
```

3. Abrir el cliente de MariaDB: Una vez instalado, nos conectarte al servidor MariaDB local usando el comando mysql:

```
mysql -u nombre_de_usuario -p -h nombre_del_host
```

En nuestro caso especificaremos:

```
mysql -u root -proot -h 127.0.0.1
```

Y como podemos comprobar hemos accedido al servidor de bases de datos MariaDB que está corriendo en el contenedor docker.

4. Comprobar las bases de datos creadas

```
show databases;
```

5. Salir del cliente y del contenedor: Para salir del cliente de MariaDB, escribe el comando `exit` o `quit` o presiona `Ctrl + D`.

1.7 Cómo detener el contenedor

Para detener el contenedor en primer lugar tenemos que conocer cuál es su ID. Para obtener el ID del contenedor podemos hacer uso del comando `docker ps`.

```
docker ps
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS                    NAMES
52bcaee9a157        mariadb             "docker-entrypoint.s…"   4 seconds ago
Up 2 seconds        0.0.0.0:3306->3306/tcp   migbd
```

En la primera columna podemos ver cuál es el CONTAINER ID. Una vez localizado el identificador ejecutamos el comando docker stop y le pasamos como parámetro el identificador del contenedor que queremos detener.

Para el caso anterior deberíamos ejecutar:

```
docker stop migbd
```