



TAREA EVALUABLE SEGUNDO CUATRIMESTRE

DESPLIEGUE DE APLICACIONES WEB

David García Vincelle



Índice

Solución Ejercicio 2

1.1 Despliegue de contenedor en red: Adminer y MariaDB

1.1.1 Crea una red `bdnet`

1.1.2 Crea un contenedor con una imagen de `mariaDB` que estará en la red `bdnet`. Este contenedor se ejecutará en segundo plano, y será accesible a través del puerto 3306. (Es necesario definir la contraseña del usuario `root` y un volumen de datos persistente [con cualquier método])

1.1.3 Crea un contenedor con el programa `Adminer` que se pueda conectar al contenedor de la BD

1.1.4 Utiliza Adminer para conectar con el contenedor del servidor de la base de datos abriendo un navegador web y accediendo a la URL: `http://localhost:8080`

1.1.5 Crea una base de datos llamada *despliegue*

Solución Ejercicio 2

1.1 Despliegue de contenedor en red: Adminer y MariaDB

En ese ejercicio, voy a crear los contenedores via docker-compose

En cada apartado donde haya que crear un contenedor, indicaré la parte de código que debemos añadir al fichero docker-compose.yml, para finalmente, lanzar el script que cree todos los contenedores y deje los microservicio funcionando como se solicita

1.1.1 Crea una red **bdnet**

- Creamos una red de tipo **bridge** que se llame **bdnet**

```
docker network create bdnet
```

1.1.2 Crea un contenedor con una imagen de **mariadb que estará en la red **bdnet** . Este contenedor se ejecutará en segundo plano, y será accesible a través del puerto 3306. (Es necesario definir la contraseña del usuario **root** y un volumen de datos persistente [con cualquier método])**

- Creamos un directorio llamado **adminerMariadb**, donde entraremos y crearemos el fichero **docker-compose.yml**

*En mi caso me encuentro en: **/home/cliente-david***

```
mkdir adminerMariadb
cd adminerMariadb
gedit docker-compose.yml
```

- Una vez cargado el editor de textos de ubuntu, implementamos el siguiente código:

*el volumen de datos persistente, lo creamos colgando del propio directorio **adminerMariadb**, creando un subdirectorio **mariadb_data***

```
version: '3.1'
services:
  mariadb:
    container_name: mariadb
    image: mariadb
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
    volumes:
```

```

- ./mariadb_data:/var/lib/mysql
networks:
default:
  external:
    name: bdnet

```

- Una vez guardado, NO procedemos aún a lanzar el script de `docker-compose`, pues queremos completarlo en el apartado siguiente

1.1.3 Crea un contenedor con el programa `Adminer` que se pueda conectar al contenedor de la BD

- Completamos la configuración del fichero `docker-compose.yml` añadiendo el código necesario para que `docker-compose` implemente la creación del contenedor `adminer`

```

adminer:
  container_name: adminer
  image: adminer
  restart: always
  environment:
    ADMINER_DEFAULT_SERVER: mariadb
  ports:
    - 8080:8080
  depends_on:
    - mariadb

```

- lo realizamos de forma que el contenido final del código sea el siguiente:

```

version: '3.1'
services:
  mariadb:
    container_name: mariadb
    image: mariadb
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
    volumes:
      - ./mariadb_data:/var/lib/mysql
  adminer:
    container_name: adminer
    image: adminer
    restart: always
    environment:
      ADMINER_DEFAULT_SERVER: mariadb
    ports:
      - 8080:8080
    depends_on:
      - mariadb
networks:
default:
  external:

```

```
name: bdnet
```

Nos aseguramos de que el contenedor `adminer` no se cree antes de que el contenedor `mariadb` ya esté creado, y por tanto pueda conectarse al servidor de bases de datos, implementando su dependencia en el script.

- Una vez guardado, procedemos a lanzar el script de `docker-compose` y comprobamos que los contenedores están creados y en ejecución

```
docker-compose up -d
docker ps -a
```

```
cliente-david@clientedavid-VirtualBox: ~/adminerMariadb
cliente-david@clientedavid-VirtualBox:~/adminerMariadb$ docker-compose up -d
Creating mariadb ... done
Creating adminer ... done
cliente-david@clientedavid-VirtualBox:~/adminerMariadb$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
5645337df196	adminer	"entrypoint.sh php -..."	33 seconds ago	Up 31 seconds	0.0.0.0:8080->8080/tcp, ::8080->8080/tcp	adminer
e5201f821917	mariadb	"docker-entrypoint.s..."	34 seconds ago	Up 32 seconds	3306/tcp	mariadb

```
cliente-david@clientedavid-VirtualBox:~/adminerMariadb$
```

- Comprobamos que están en la red correcta

```
docker network inspect bdnet
```

```
cliente-david@clientedavid-VirtualBox: ~/adminerMariadb

{
  "Ingress": false,
  "ConfigFrom": {
    "Network": ""
  },
  "ConfigOnly": false,
  "Containers": {
    "5645337df19629cc757c5e008c56e25117583b8556fc07cd12d2b1b510014972": {
      "Name": "adminer",
      "EndpointID": "1365663bccdc15e9539b2376fafcdaedebdcb47210eb84072d70bba585c63cc3",
      "MacAddress": "02:42:ac:17:00:03",
      "IPv4Address": "172.23.0.3/16",
      "IPv6Address": ""
    },
    "e5201f8219178cc2aec6658e3e98cb95653db6dbe2657ffe31843c81dc9f7b49": {
      "Name": "mariadb",
      "EndpointID": "e5849f117835f49dc09c0288b0ae3cf1d75c49e7651de30c6c7976146c7d79a1",
      "MacAddress": "02:42:ac:17:00:02",
      "IPv4Address": "172.23.0.2/16",
      "IPv6Address": ""
    }
  },
  "Options": {},
  "Labels": {}
}
```

```
cliente-david@clientedavid-VirtualBox:~/adminerMariadb$
```

1.1.4 Utiliza Adminer para conectar con el contenedor del servidor de la base de datos abriendo un navegador web y accediendo a la URL: <http://localhost:8080>

Abrimos el navegador web Firefox y escribimos la dirección de nuestro servicio web con adminer: `localhost:8080`

Adminer 4.8.1

Login

Motor de base de datos: MySQL

Servidor: mariadb

Usuario:

Contraseña:

Base de datos:

☐ Guardar contraseña

1.1.5 Crea una base de datos llamada *despliegue*

- Hacemos login con los datos de usuario **root** y contraseña **root**
- Creamos la base de datos **despliegue** haciendo click en crear base de datos

Adminer 4.8.1

Seleccionar Base de datos

DB:

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

[Crear Base de datos](#) [Privilegios](#) [Lista de procesos](#) [Variables](#) [Estado](#)

Versión MySQL: 5.5.5-10.11.2-MariaDB-1:10.11.2+maria-ubu2204 a través de la extensión de PHP MySQLi

Logueado como: root@172.23.0.3

Base de datos - Refrescar	Colación	Tablas	Size - Compute
<input type="checkbox"/> information_schema	utf8mb3_general_ci	?	?
<input type="checkbox"/> mysql	utf8mb4_general_ci	?	?
<input type="checkbox"/> performance_schema	utf8mb3_general_ci	?	?
<input type="checkbox"/> sys	utf8mb3_general_ci	?	?

Selected (0)

- Guardamos la base de datos

Adminer 4.8.1

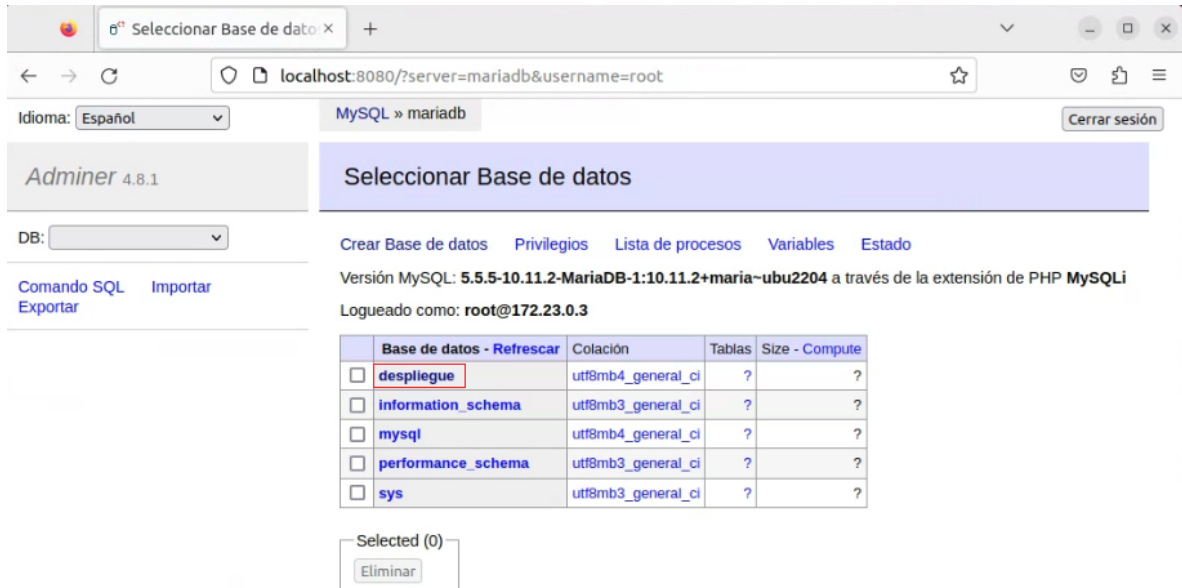
Crear Base de datos

DB:

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

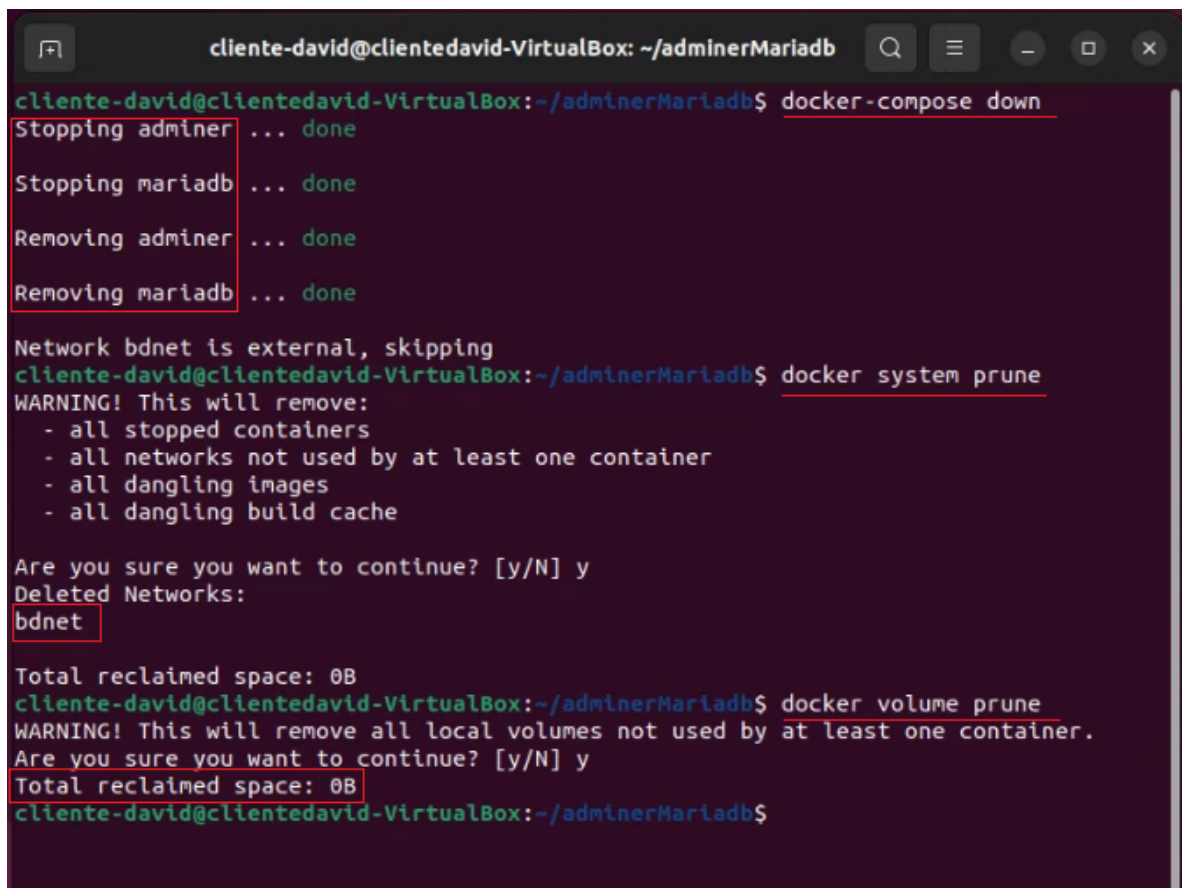
(colación)

- Vemos cómo ha sido creada



- Borramos los contenedores, la red y los volúmenes utilizados

```
docker-compose down
docker system prune
docker volume prune
```



No se libera espacio al eliminar el volumen `mariadb_data` por que hemos utilizado un bind mount

- Comprobamos que los datos generados por el contenedor del servidor de base de datos, persisten

