

Utilizando Docker, despliega un contenedor que ejecute un servidor web Apache y otro contenedor que ejecute una base de datos MariaDB. Configura MariaDB para que use un volumen para la persistencia de datos. Asegúrate de que ambos contenedores se comuniquen correctamente a través de una red personalizada. Utiliza la última versión de la imagen en ambos casos.

- Crea un volumen de Docker llamado `mi_volumen_mariadb` para almacenar los datos de la base de datos de manera persistente.

```
PS C:\Users\alcar> docker volume create mi_volumen_mariadb
mi_volumen_mariadb
PS C:\Users\alcar> |
```

Uso el comando 'docker volume create' para crear un volumen a que he llamado 'mi_volumen_mariadb'.

- Despliega un contenedor de MariaDB utilizando el volumen creado y establece las credenciales de la base de datos utilizando variables de entorno.

```
PS C:\Users\alcar> docker run -it -d --name c_mariadb -v mi_volumen_mariadb:/var/lib/mysql -e MYSQL_ROOT_PASSWORD="1234"
-e MYSQL_DATABASE="my_database_mariadb" -e MYSQL_USER="user_mariadb" -e MYSQL_PASSWORD="1234" mariadb
f24ad096a0d24af905b3038ff9306eb857b3213d7ed1d4b345f0489dda940984
PS C:\Users\alcar> |
```

Usando docker run para ejecutar un contenedor mapeando el volumen creado previamente:

-it: terminal interactiva.
-d: ejecución en segundo plano.
--name: nombre del contenedor.
-v: mapeo del volumen.
-e: variables de entorno

- Despliega un contenedor de Apache que sirva una página web de prueba, mapeando el puerto 8080 de tu máquina local al puerto 80 del contenedor.

```
PS C:\Users\alcar> docker run -it -d --name c_apache -p 8080:80 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
6bd9d3710aae: Download complete
dbde712f81fb: Download complete
4f4fb700ef54: Download complete
867b2ea3628d: Download complete
3a2663e66670: Download complete
Digest: sha256:bbea29057f25d9543e6a96a8e3cc7c7c937206d20eab2323f478fdb2469d536d
Status: Downloaded newer image for httpd:latest
800b87386f3bb62f59d2461de27cf9fedf2d9a5da9f4a3d76f98c49d9c020729
PS C:\Users\alcar> |
```

Igual que con mariadb pero mapeando un puerto con el tag '-p'

- Crea una red personalizada para que los dos contenedores puedan comunicarse entre sí.

```
PS C:\Users\alcar> docker network create --driver bridge red_exmn
1a05b9e76bf5591a0f650a9d5bf52dc5e8ed49c94c2b43a0fb054c679b802275
PS C:\Users\alcar> |
```

Con el comando 'docker network create' es posible crear una red:

—drive: tipo de driver que emplea la red, en este caso tipo bridge.

El nombre de la red en este caso es red_exmn

- Verifica que ambos contenedores están en funcionamiento y que el contenedor de Apache es accesible en <http://localhost:8080>.

```
PS C:\Users\alcar> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
800b87386f3b   httpd     "httpd-foreground"      3 minutes ago Up 3 minutes   0.0.0.0:8080->80/tcp     c_apache
f24ad096a0d2   mariadb   "docker-entrypoint.s..." 6 minutes ago Up 6 minutes   3306/tcp             c_mariadb
PS C:\Users\alcar> |
```

Uso de docker ps para ver los contenedores activos.



Conexión con mi puerto local 8080 usando el navegador web.

- Investiga como se realiza la configuración en Docker para limitar el contenedor a usar un máximo de 1 CPU y 512MB de memoria

Con el siguiente comando :

```
docker run --cpus 1 --memory 512MB <nombre-de-la-imagen>
```

—cpus para limitar el uso de cpu y —memory para el de memoria RAM

Entrega:

- Incluir un pdf donde se describa brevemente el proceso seguido, los comandos utilizados, cualquier configuración específica y las fuentes utilizadas.
- Hay que demostrar no solo la ejecución de los comandos, sino también que se comprende lo que se ha hecho.

