

Docker para MySQL

Una introducción práctica para
Acceso a Datos (2ºDAM)



¿Qué es Docker?

— — —

- **Docker:** Una herramienta que empaqueta aplicaciones en contenedores.
- **Contenedor:** Una caja estandarizada que contiene todo lo necesario para que una aplicación funcione.
- **Beneficio:** Permite que una aplicación se ejecute de la misma manera, sin importar el entorno.
- Diferencia con una VM:
 - VM → necesita sistema operativo completo.
 - Contenedor → comparte el kernel (más ligero).



Conceptos clave: Imagen vs. Contenedor

- **Imagen (Image):** La plantilla inmutable de la aplicación. Piensa en ella como una "clase" en programación.
- **Contenedor (Container):** Una instancia ejecutable de una imagen. Es el "objeto" vivo que se ejecuta.
- **Docker Hub:** Un repositorio público de imágenes, como el "GitHub" para contenedores.
 - Volumen: almacena datos de forma persistente.
 - Puerto: expone un servicio al host (ej: 3306).

Comandos básicos

- ``docker ps`` → contenedores activos.
- ``docker ps -a`` → todos los contenedores.
- ``docker logs <nombre>`` → ver registros.
- ``docker exec -it <nombre> bash`` → entrar en el contenedor.

¿Qué es docker-compose?

— — —

- Una herramienta para definir y ejecutar aplicaciones multi-contenedor.
- Simplifica la gestión de servicios interconectados (ej. una base de datos y un servidor web).
- Todo se define en un único archivo: `docker-compose.yml`.
 - Permite levantar/installar todo con un solo comando.



Estructura básica

— — —

- **services:** definición de servicios (ej: mysql).
- **environment:** variables de entorno (usuario, contraseña...).
- **ports:** exposición de puertos.
- **volumes:** persistencia de datos.

Comandos clave

- ``docker-compose up -d`` → levantar servicio.
- ``docker-compose down`` → parar y borrar contenedores.
- ``docker-compose ps`` → estado de servicios.
- ``docker exec -it <nombre> <comando>``

Importar / Exportar SQL

- Importar: `docker exec -i <nombre> mysql -u root -p < script.sql`
- Exportar: `docker exec -i <nombre> mysqldump -u root -p basedatos > backup.sql`

Trabajar con MySQL en Docker

1. Ejecutar un script .sql dentro del contenedor

Cuando queremos cargar tablas o datos desde un archivo, usamos:

`docker exec -i mysql-dev mysql -uroot -pasmysql acda < alumno.sql`

- `mysql-dev` → nombre del contenedor.
- `-uroot -pasmysql` → usuario y contraseña.
- `acda` → base de datos destino.
- `< alumno.sql` → redirige el contenido del archivo para que se ejecute dentro de MySQL.

Útil para cargar estructuras completas o datasets de prueba.

Trabajar con MySQL en Docker

2. Ejecutar una sentencia SQL directamente

Cuando queremos lanzar una consulta rápida, usamos la opción -e:

```
docker exec -it mysql-dev mysql -uroot -pmsmysql acda -e "SELECT * FROM Alumnos;"
```

- -e "... " → indica que vamos a ejecutar una sentencia directamente.
- La salida se muestra en la consola.

Útil para comprobar que los datos se cargaron bien.

Trabajar con MySQL en Docker

— — —

3. Acceder al cliente interactivo de MySQL: Para entrar en la consola de MySQL dentro del contenedor y poder escribir varias sentencias:

`docker exec -it mysql-dev mysql -uroot -pmaasmysql acda`

- Aparecerá el prompt de MySQL (mysql>).
- Desde ahí se pueden ejecutar múltiples sentencias:

SHOW DATABASES;

USE acda;

*SELECT * FROM Alumnos;*

Útil para practicar consultas o explorar la BD.