

Actividad 6

Docker compose MySQL y PostgreSQL

Cristóbal Suárez Abad

Optativa – 2º ASIR

1. *Usando las imágenes de MySQL y PostgreSQL*
2. *Queremos instalar MySQL y PostgreSQL, y configurarlos.*
3. *PostgreSQL no levantará su contenedor hasta que no esté levantado MySQL*
4. *elegir un usuario y contraseña que quieras crear para las dos base de datos.*
5. *Cambiar los puertos por defecto de las base de datos*
6. *Probar conexión con DBeaver*

Tenéis que entregar el fichero docker compose y un documento explicando cada uno de los pasos pedidos.

1) Servicio de MySQL:

mysql:

image: mysql:latest

container_name: mysql_server

restart: always

“mysql”: Nombre del servicio. Este nombre es descriptivo. Si no se usa

“container_name”, Docker lo usará para ponerle nombre al container.

“image”: imagen que se va a usar.

“container_name”: nombre del container.

“restart: always” permite que el contenedor se reinicie si se para ante cualquier error.

Si lo paramos manualmente se reiniciará si reiniciamos el Daemon Docker¹.

ports:

- "3307:3306"

Establece los puertos por los que escuchará la aplicación: el primero es el del equipo host y el segundo el del contenedor.

expose:

- "3306"

Establece o más bien documenta, cual es el puerto por el que se comunica con otros contenedores dentro del mismo bridge².

environment:

MYSQL_ROOT_PASSWORD: 12345

MYSQL_USER: cristobal

MYSQL_PASSWORD: 12345

MYSQL_DATABASE: optativadb01

Son las variables que nos van a permitir establecer contraseña del ROOT, usuario, contraseña de usuario y crear y nombrar una base de datos. De manera automática el usuario creado tiene todos los privilegios en esa base de datos³.

volumes:

- mysql_data:/var/lib/mysql

Volumen de persistencia para evitar que la información de la base de datos se borre⁴.

networks:

- red_bd

Se ha establecido una red propia para ambos contenedores, para evitar futuras interferencias con anteriores y futuros contenedores.

¹ <https://stackoverflow.com/questions/61725195/difference-in-docker-restart-policy-between-on-failure-and-unless-stopped>

² <https://docs.docker.com/reference/dockerfile/>

³ https://hub.docker.com/_/mysql

⁴ https://hub.docker.com/_/mysql

2) Servicio PostgreSQL.

postgres:

image: postgres:latest

container_name: postgres_server

restart: always

depends_on:

- mysql

La configuración es parecida a la anterior, excepto por “depends_on” que hace que el contenedor tenga que esperar a que esté levantado primero el que se indica, en este caso el que se genera con el servicio “mysql”⁵.

ports:

- "5433:5432"

expose:

- "5432"

environment:

POSTGRES_USER: cristobal

POSTGRES_PASSWORD: 12345

POSTGRES_DB: optativadb02

Variables propias de PostgreSQL. Hay que tener en cuenta que cada imagen tiene sus propias variables⁶.

volumes:

- postgres_data:/var/lib/postgresql

networks:

- red_bd

⁵ https://docs.docker.com/reference/compose-file/services/#depends_on

⁶ https://hub.docker.com/_/postgres

3) Declaración de “networks” y “volumes”.

Tanto las redes como los volúmenes se declaran en un apartado aparte.

networks:

red_bd:

driver: bridge

volumes:

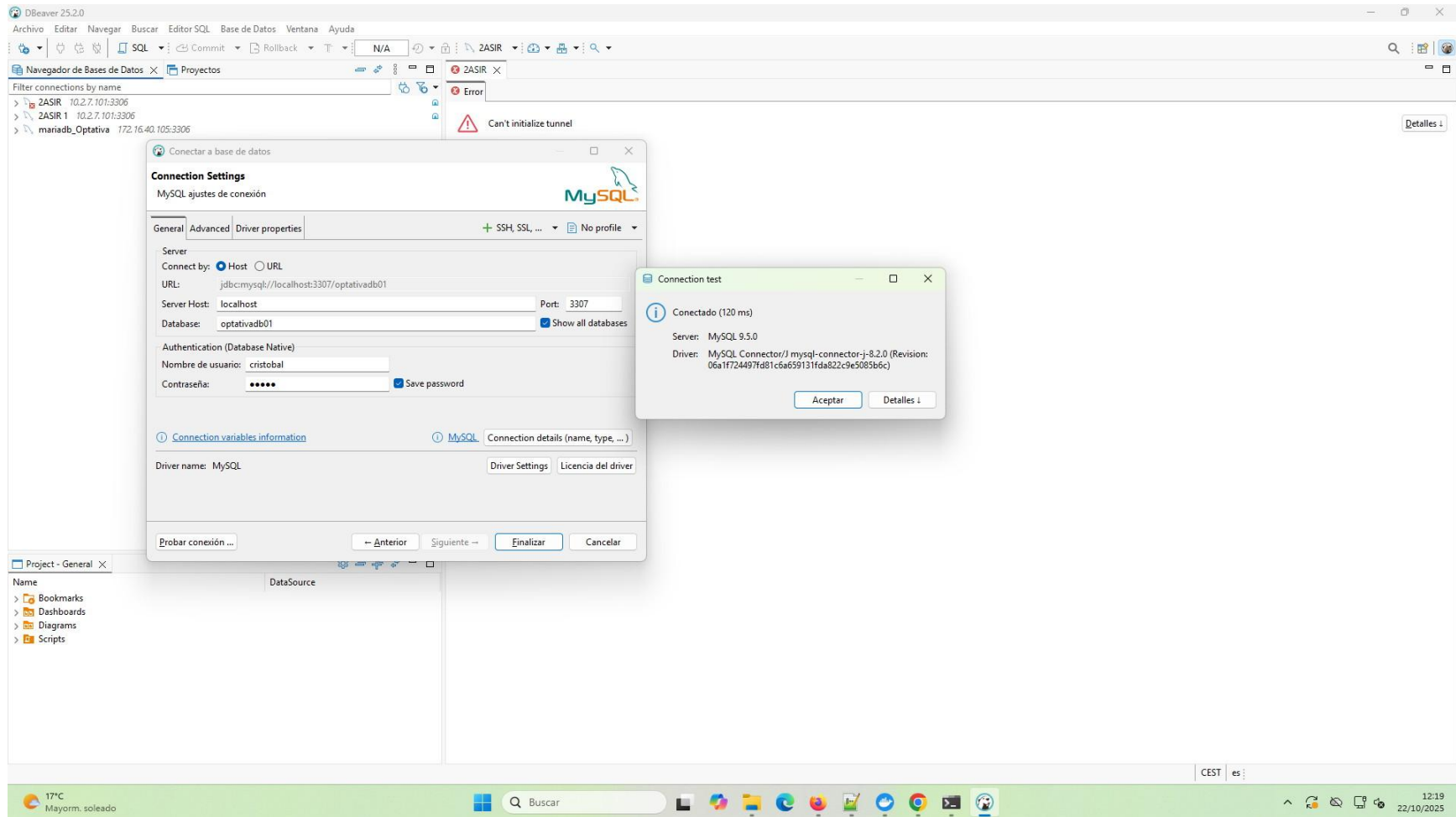
mysql_data:

postgres_data:

En ambos casos se han dejado los valores por defecto. La “network” tiene los valores por defecto de un bridge, pero estableciendo uno nuevo a parte. Y los volúmenes en esta ocasión solo hay que nombrarlos en el archivo .yaml, no hay necesidad de crearlos antes como con “docker run”.

4) Conexión con DBeaver.

a) MySQL.



b) PostgreSQL.

