



Pràctica 10

Bases de dades en Docker

Lliuraments

Els resultats d'aquesta part de la pràctica s'hauran d'entregar en **format PDF** i l'entrega pot ser a través de GIT* o el Moodle.

* S'ha d'entregar l'enllaç del GIT al Moodle.

MariaDB en un Docker

1. Crea un contenidor Docker de **MariaDB** i executa'l al **port 3307**. Pots descarregar una imatge de MariaDB del DockerHub.

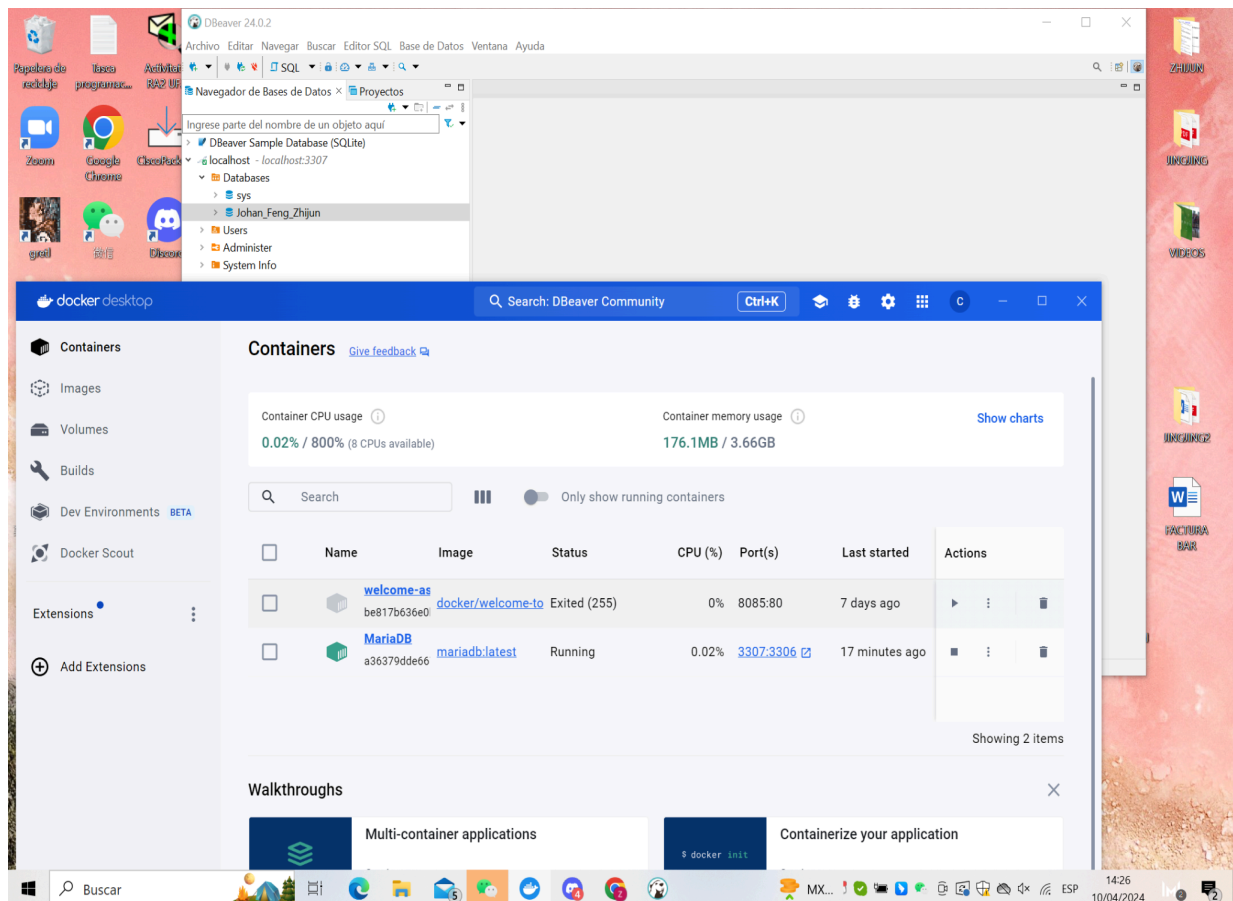
Pots consultar la documentació de la imatge de docker de MariaDB:

https://hub.docker.com/_/mariadb

A més, recorda que has d'establir la variable d'entorn **MARIADB_ROOT_PASSWORD** i escollir una contrasenya que després faràs servir per connectar-te a la base de dades.

2. Instal·la't **DBeaver Community**¹ i connecta't al port 3307 del localhost, on s'executa el servei de la base de dades.
3. Crea una base de dades amb el **nom dels integrants del grup**.
4. **Entrega una screenshot** on es vegi la base de dades i el contenidor docker.

¹DBeaver Community: <https://dbeaver.io/>



MariaDB en un Docker mitjançant un fitxer YAML

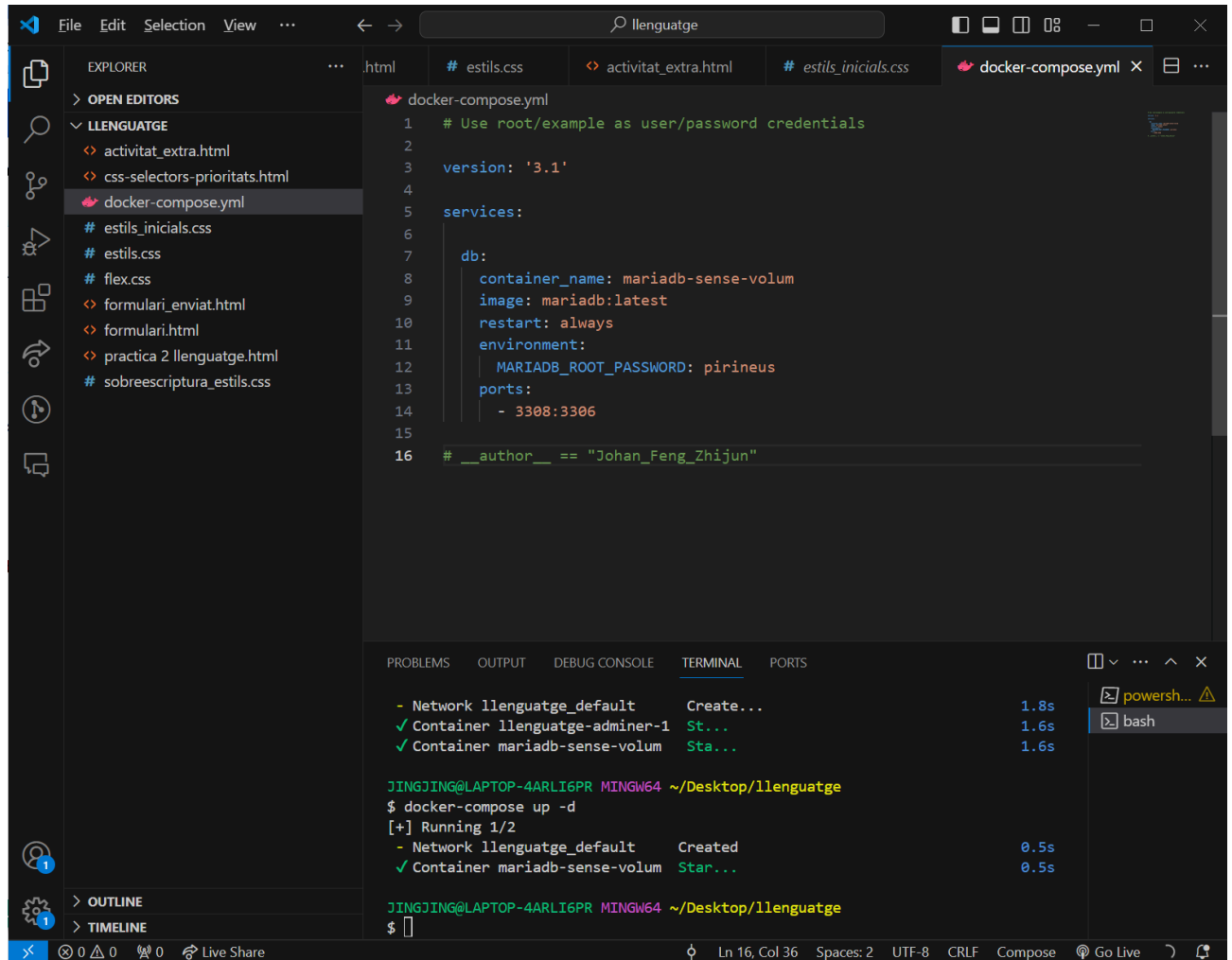
5. Crea un contenidor Docker de **MariaDB** fent servir un **fitxer de configuració YAML**.
 - a. Crea una carpeta buida al teu ordinador.
 - b. Afegeix-hi un fitxer amb el nom **docker-compose.yml**
 - c. **Modifica** el següent contingut perquè tingui la configuració per crear un contenidor a partir de la imatge de MariaDB i s'executi al **port 3308** i el nom del contenidor sigui ***mariadb-sense-volum***.

```
version: "3.9"
services:
  nom_servei:
    image: url_imatge_dockerhub
    container_name: nom_del_contenidor
    environment:
      #variables d'entorn si calen
    ports:
      - port_teva_maquina:port_servei_docker
```

- d. Executa la següent instrucció i comprova, amb DBeaver, que et pots connectar a la base de dades del port 3308.

```
docker-compose up -d
```

- e. **Entrega una captura del fitxer docker-compose.yml.** Títol de la captura: compose_mariadb



The screenshot shows the Visual Studio Code interface with the `docker-compose.yml` file open. The file content is as follows:

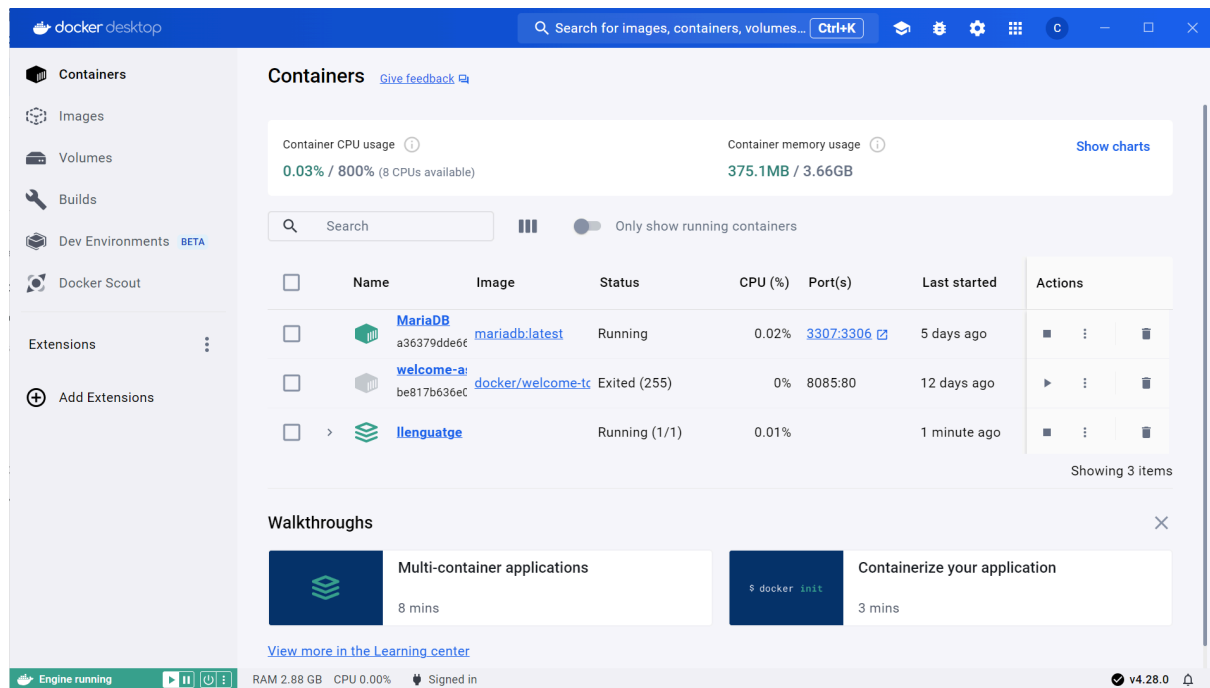
```
1 # Use root/example as user/password credentials
2
3 version: '3.1'
4
5 services:
6   db:
7     container_name: mariadb-sense-volum
8     image: mariadb:latest
9     restart: always
10    environment:
11      MARIADB_ROOT_PASSWORD: pirineus
12    ports:
13      - 3308:3306
14
15
16 # __author__ == "Johan_Feng_Zhijun"
```

The terminal output shows the execution of `docker-compose up -d` and the subsequent status of the containers:

```
JINGJING@LAPTOP-4ARLI6PR MINGW64 ~/Desktop/llenguatge
$ docker-compose up -d
[+] Running 1/2
- Network llenguatge_default Created 0.5s
✓ Container mariadb-sense-volum Star... 0.5s
```

The terminal also shows the status of the containers after the command is executed:

```
- Network llenguatge_default Create... 1.8s
✓ Container llenguatge-adminer-1 St... 1.6s
✓ Container mariadb-sense-volum Sta... 1.6s
```



MariaDB en un Docker mitjançant YML amb un volume

6. Crea un nou fitxer de configuració **docker-compose** que creï un contenidor de MariaDB amb un **volum**. Tingues en compte que la informació de MariaDB s'emmagatzema a la ruta **/var/lib/mysql**
 - a. El **nom del container** ha de ser ***mariadb-volum***
 - b. El **nom del volum** ha de ser ***data_nom_del_grup***
 - c. **Entrega una captura** del fitxer **docker-compose.yml**. Títol de la captura: ***compose_mariadb_volume***

```
🔥 docker-compose.yml ●
C: > Users > Ponnc > Documents > M4 > Docker > 🔥 docker-compose.yml
1  version: '3.1'
2
3  services:
4
5      db:
6          container_name: mariadb-volumes
7          image: mariadb:latest
8          restart: always
9          volumes:
10             - Zhijun_Feng_Johan:/var/lib/mysql
11          environment:
12             MARIADB_ROOT_PASSWORD: '1456'
13          ports:
14             - 3309:3306
15  volumes:
16      Zhijun_Feng_Johan:
17
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\Ponnc\Documents\M4\Docker> docker-compose up -d
[+] Running 1/0
✓ Container mariadb-volum Running
PS C:\Users\Ponnc\Documents\M4\Docker> docker-compose up -d
[+] Running 1/1
✓ Container mariadb-volum Running
PS C:\Users\Ponnc\Documents\M4\Docker> docker-compose up -d
[+] Running 1/2
- Container mariadb-volum Recreated
✓ Container mariadb-volumes Started
PS C:\Users\Ponnc\Documents\M4\Docker> 
```

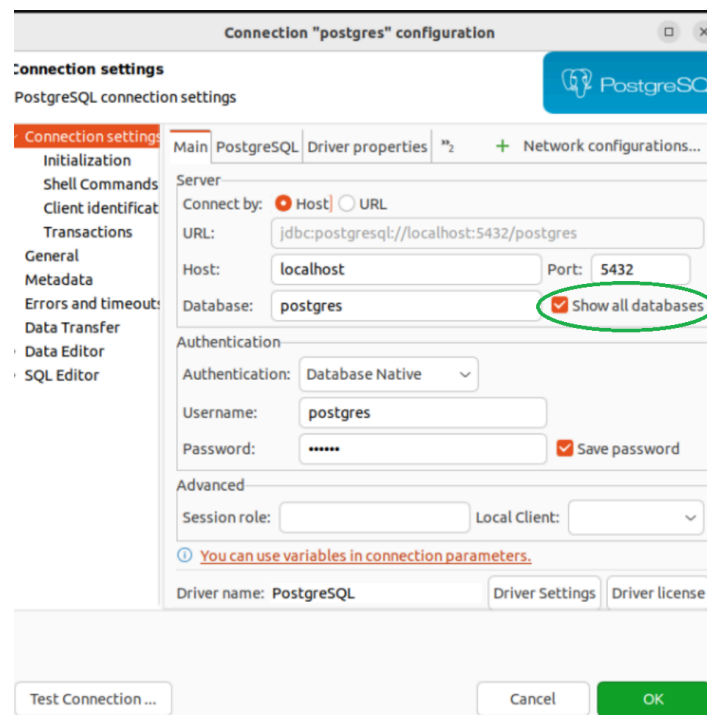
7. **Crea una base de dades** al contenidor *mariadb-sense-volum* i una altra a *mariadb-volum*.
8. Suprimeix els dos contenidors dockers i torneu-los a crear.
9. **Explica quina diferència** veus una vegada estableixes la connexió als dos dockers.

La diferència és que el docker sense volum no guarda la informació del DB una vegada borrada i creada una altre vegada el docker i el que té volum si que guarda el DB.

PostgreSQL en un Docker mitjançant YML amb un volume

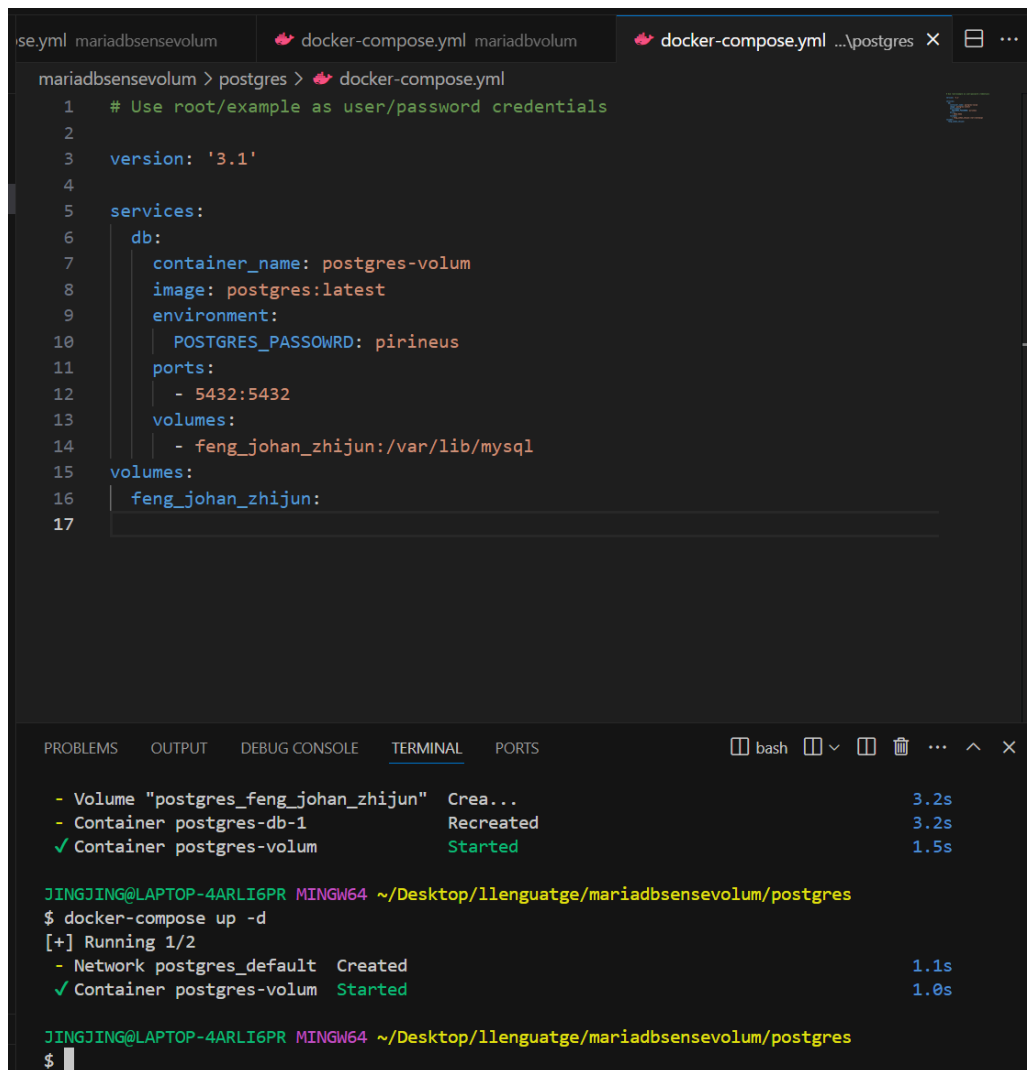
10. Crea un nou fitxer de configuració **docker-compose** que creï un contenidor de PostgreSQL amb un **volum**. Busca tota la informació en la documentació del dockerhub i fes servir un port estàndard.

https://hub.docker.com/_/postgres



Ves amb compte tenir marcada l'opció **“show all databases”** quan configuris el postgres. Sense aquesta opció, el DBeaver no mostra totes les bases de dades i sembla que s'esborrin quan es reinicia el docker.

11. Entrega una captura del fitxer **docker-compose.yml**. Títol de la captura: **compose_postgres**



The image shows a VS Code editor with a file named `docker-compose.yml` open. The file contains the following configuration:

```
1 # Use root/example as user/password credentials
2
3 version: '3.1'
4
5 services:
6   db:
7     container_name: postgres-volum
8     image: postgres:latest
9     environment:
10      POSTGRES_PASSWORD: pirineus
11     ports:
12      - 5432:5432
13     volumes:
14      - feng_johan_zhijun:/var/lib/mysql
15 volumes:
16   feng_johan_zhijun:
```

The terminal at the bottom shows the output of the command `docker-compose up -d`:

```
JINGJING@LAPTOP-4ARLI6PR MINGW64 ~/Desktop/llenguatge/mariadbsensevolum/postgres
$ docker-compose up -d
[+] Running 1/2
- Network postgres_default Created 1.1s
✓ Container postgres-volum Started 1.0s

JINGJING@LAPTOP-4ARLI6PR MINGW64 ~/Desktop/llenguatge/mariadbsensevolum/postgres
$
```

Docker-compose MariaDB + PostgreSQL

12. Crea un nou fitxer de configuració **docker-compose** que creï un contenidor de **PostgreSQL** i un contenidor **MariaDB** amb un **volum** per cada un.
13. Crea una **base de dades** a cada sistema gestor.
14. Suprimeix els contenidors dockers i torneu-los a crear i comprova que les bases de dades continuen persistint.
15. **Entrega una captura** del fitxer **docker-compose.yml**. Títol de la captura: `compose_maria_postgres`

```
mariadb-sensevolum > postgres > postgres-mariadb > docker-compose.yml
```

```
1  # Use root/example as user/password credentials
2
3  version: '3.1'
4
5  services:
6    db:
7      container_name: mariadb-volumes
8      image: mariadb:latest
9      restart: always
10     environment:
11       MARIADB_ROOT_PASSWORD: pirineus
12     ports:
13       - 3309:3306
14     volumes:
15       - johan_feng_zhijun:/var/lib/mysql
16
17     mariadb:
18       container_name: postgres-volumes
19       image: postgres:latest
20       environment:
21         POSTGRES_PASSWORD: pirineus
22       ports:
23         - 5432:5432
24       volumes:
25         - feng_johan_zhijun:/var/lib/mysql
26
27     volumes:
28       feng_johan_zhijun:
29       johan_feng_zhijun:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
$ docker-compose up -d
```

```
[+] Running 1/2
```

```
✓ Container postgres-volumes Started 1.1s
```

```
- Container mariadb-volumes Starting 0.9s
```

```
Error response from daemon: driver failed programming external connectivity on endpoint mariadb-volumes (f95369e6e483e7c486785099c0de94a62352ac90b65bf313a022ef43bf1053e3): Bind for 0.0.0.0:3309 failed: port is already allocated
```

```
JINGJING@LAPTOP-4ARLI6PR MINGW64 ~/Desktop/llenguatge/mariadb-sensevolum/postgres/postgres-mariadb
```

```
$
```