

# M08-UF2 DOCKER COMPOSE

## Table of Contents

Apuntes .....	1
Diferencias entre "Dockerfile" y "docker-compose.yml" .....	1
Análisis de un YAML "docker-compose.yml" .....	1
web .....	2
php .....	2
db .....	3
phpmyadmin .....	3
Activitats .....	5
Activitat 1 - PHP/APACHE/MySQL .....	5
Apache y PHP .....	5
Añadir MySQL .....	7
Añadir PhpMyAdmin .....	9
Activitat 2 - Persistencia .....	12

## Apuntes

### Diferencias entre "Dockerfile" y "docker-compose.yml"

Los **Dockerfile** son la herramienta fundamental para construir imágenes de Docker. Estos archivos contienen una serie de instrucciones necesarias para construir una imagen personalizada, como la instalación de paquetes, la copia de archivos y la configuración de variables de entorno.

**Docker Compose** es una herramienta que permite ejecutar varios contenedores de Docker como si fueran una sola aplicación. Cada contenedor se define como un servicio con un nombre específico, y se pueden vincular entre sí para proporcionar una funcionalidad más compleja. Docker Compose facilita el despliegue y la gestión de aplicaciones que requieren varios contenedores, y se puede utilizar para definir y configurar toda la pila de aplicaciones en un solo archivo YAML.

### Análisis de un YAML "docker-compose.yml"

- Indicamos la versión de la sintaxis del archivo "Dockerfile". En este caso la version 3.

```
version: '3'
```

- Indicamos que van a definirse una lista de servicios (web, php, db, phpmyadmin en el ejemplo)

```
services:
```

## web

- Se le asigna un nombre al servicio. Este servicio será el servidor HTTP.

```
web:
```

- Se utiliza una imagen de *Apache HTTP*, en este caso es el contenedor `httpd:2.4`.

```
image: httpd:2.4
```

- Se configura un volumen, es decir **"un mecanismo para compartir dataos entre contenedores y el host"**. en este caso, se mapea la carpeta local `"/web"` al directorio `"/usr/local/apache2/htdocs/"`, permitiendo que el contenido de la carpeta local se muestre en el servidor web.

```
volumes:  
  - ./web:/usr/local/apache2/htdocs/
```

- Se mapea el puerto 80 del contenedor al puerto 80 del host local, para que el sitio web que se sirve, sea accesible desde el navegador.

```
ports:  
  - "80:80"
```

- Definimos las dependencias entre servicios. En este caso, el servicio **web** depende de **php** y de **db**, por lo tanto, antes de que se ejecute el contenedor **web**, se garantizará que los contenedores de los que depende, estén en ejecución. **"depends\_on"** es utilizado para establecer un orden de arranque.

```
depends_on:  
  - php  
  - db
```

## php

- Se le asigna un nombre al servicio. Este servicio será el language e interprete de PHP.

```
php:
```

- En este caso, en lugar de una imagen ya definida, se construirá una personalizada en base al Dockerfile ubicado en el directorio "./php".

```
build: ./php/
```

- Mapeamos "./web" del host a "/var/www/html/" del contenedor, ya que este servicio es responsable de procesar los archivos PHP del site y enviar la respuesta procesada (HTML) al servicio **web**, que entonces, envia al cliente que realiza la petición HTTP. Entonces, los archivos PHP del directorio "./web" son accesibles tanto por los servicios **web** como por **php**

```
volumes:  
  - ./web:/var/www/html/
```

- Se establece una dependencia de "db"

```
depends_on:  
  - db
```

## db

- Usamos una imagen de Docker "mysql:5.7" para ejecutar una instancia de MySQL.

```
image: mysql:5.7
```

- Definimos las variables de entorno de este servicio. En este caso, configuramos las credenciales de inicio de sesión de MySQL. Esta información se puede obtener en la documentación de cada imagen disponible en **Docker Hub**

```
environment:  
  MYSQL_ROOT_PASSWORD: root  
  MYSQL_DATABASE: test_db  
  MYSQL_USER: test_user  
  MYSQL_PASSWORD: test_password
```

- Asociamos el puerto 3306 del contenedores a el del sistema host

```
ports:  
  - "3306:3306"
```

## phpmyadmin

- Usamos la imagen oficial de "phpmyadmin"

```
image: mysql:5.7
```

- Definimos variables de entorno para este servicio, en este caso con **PMA\_HOST** especificamos que se debe conectar al servicio **db** definido en este docker-compose, tambien indicamos la password de root

```
environment:  
  PMA_HOST: db  
  MYSQL_ROOT_PASSWORD: root
```

- Indicamos que este servicio depende de db para funcionar correctamente

```
depends_on:  
  - db
```

```
version: '3'  
  
services:  
  web:  
    image: httpd:2.4  
    volumes:  
      - ./web:/usr/local/apache2/htdocs/  
    ports:  
      - "80:80"  
    depends_on:  
      - php  
      - db  
  
  php:  
    build: ./php/  
    volumes:  
      - ./web:/var/www/html/  
    depends_on:  
      - db  
  
  db:  
    image: mysql:5.7  
    environment:  
      MYSQL_ROOT_PASSWORD: root  
      MYSQL_DATABASE: test_db  
      MYSQL_USER: test_user  
      MYSQL_PASSWORD: test_password  
    ports:  
      - "3306:3306"  
  
  phpmyadmin:
```

```
image: phpmyadmin/phpmyadmin
environment:
  PMA_HOST: db
  MYSQL_ROOT_PASSWORD: root
ports:
  - "8080:80"
depends_on:
  - db
```

# Activitats

## Activitat 1 - PHP/APACHE/MySQL

### Apache y PHP

```
version: '3'

services:
  web:
    container_name: php-apache
    image: php:apache
    ports:
      - "8080:80"
    volumes:
      - ./site:/var/www/html
```

- Creamos un directorio donde situar el archivo compose y otro donde guardaremos los archivos del site que vamos a servir

```
Last login: Sat Apr 15 14:27:34 2023
ubuntu@ubuntu:~$ cd /home/ubuntu
ubuntu@ubuntu:~$ mkdir docker
ubuntu@ubuntu:~$ cd docker/
ubuntu@ubuntu:~/docker$ touch docker-compose.yml
ubuntu@ubuntu:~/docker$ mkdir site
ubuntu@ubuntu:~/docker$ ls -l
total 4
-rw-rw-r-- 1 ubuntu ubuntu    0 abr 15 14:29 docker-compose.yml
drwxrwxr-x 2 ubuntu ubuntu 4096 abr 15 14:29 site
ubuntu@ubuntu:~/docker$
```

Xavi Quiñones M08 UF2 15-ABRIL-23|

- Paramos Apache2 si lo tenemos instalado en la maquina para evitar conflictos

```
ubuntu@ubuntu:~/docker$ systemctl stop apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units =
Authentication is required to stop 'apache2.service'.
Authenticating as: ubuntu
Password:
==== AUTHENTICATION COMPLETE ====
ubuntu@ubuntu:~/docker$
```

Xavi Quiñones M08 UF2 15-ABRIL-23

- Configuramos el servicio **web**

```
version: '3'

services:
  web:
    container_name: php-apache
    image: php:apache
    ports:
      - "8080:80"
    volumes:
      - ./site:/var/www/html
```

Xavi Quiñones M08 UF2 15-ABRIL-23

- Nos cambiamos de directorio a `"/site"` y creamos el archivo `index.php` que vamos a usar para la prueba

```
GNU nano 6.2 index.php *
<?php
echo "Aquesta aplicació s'està servint des d'aquí"
?>
```

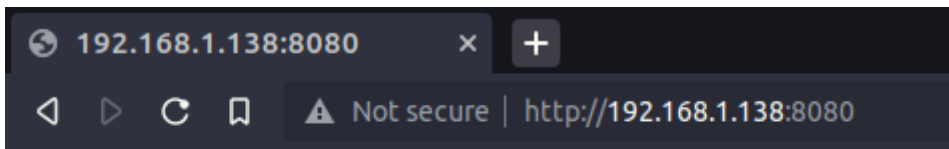
Xavi Quiñones M08 UF2 15-ABRIL-23

- Lanzamos el `docker-compose`

```
ubuntu@ubuntu:~/docker$ sudo docker-compose up -d
Creating network "docker_default" with the default driver
Pulling web (php:apache)...
apache: Pulling from library/php
26c5c85e47da: Pull complete
39c8021d1258: Pull complete
dff43c2de684: Extracting [=====]
383987c505e8: Download complete
3fd742e8a904: Download complete
ccf9807e8362: Download complete
11cc7ce10028: Download complete
7c9a93edd83f: Download complete
60001876b7f0: Download complete
02e828362dd6: Download complete
61967aaed772: Download complete
ac804962968c: Download complete
b9dda7bc4893: Download complete
```

Xavi Quiñones M08 UF2 15-ABRIL-23

- Probamos desde el navegador



Aquesta aplicació s'està servint des d'un contenidor amb Apache i PHP

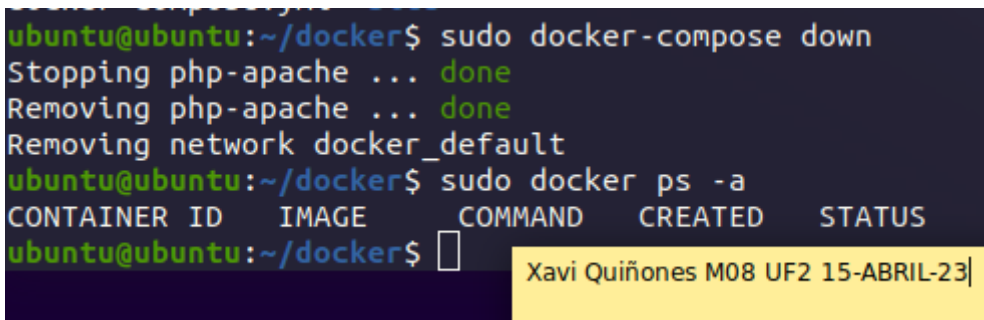
Xavi Quiñones M08 UF2 15-ABRIL-23

## Añadir MySQL

```
version: '3'

services:
  web:
    container_name: php-apache
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "8080:80"
    volumes:
      - ./site:/var/www/html
  db:
    container_name: db
    image: mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: PASSW_ROOT_MYSQL
      MYSQL_DATABASE: DB_08
      MYSQL_USER: USUARI_MYSQL
      MYSQL_PASSWORD: PASSW_MYSQL
    ports:
      - "3306:3306"
```

- Paramos docker y borramos contenedores **docker-compose down**



- Creamos el **dockerfile** personalizado con la librería mysql

```
GNU nano 6.2 Dockerfile
FROM php:apache
RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli
RUN apt-get update && apt-get upgrade -y
Xavi Quiñones M08 UF2 15-ABRIL-23
```

- Editamos el archivo `yaml`. Añadimos un nuevo servicio, en este caso **db** y lo configuramos con la imagen de apache personalizada

```
GNU nano 6.2 docker-compose
version: '3'

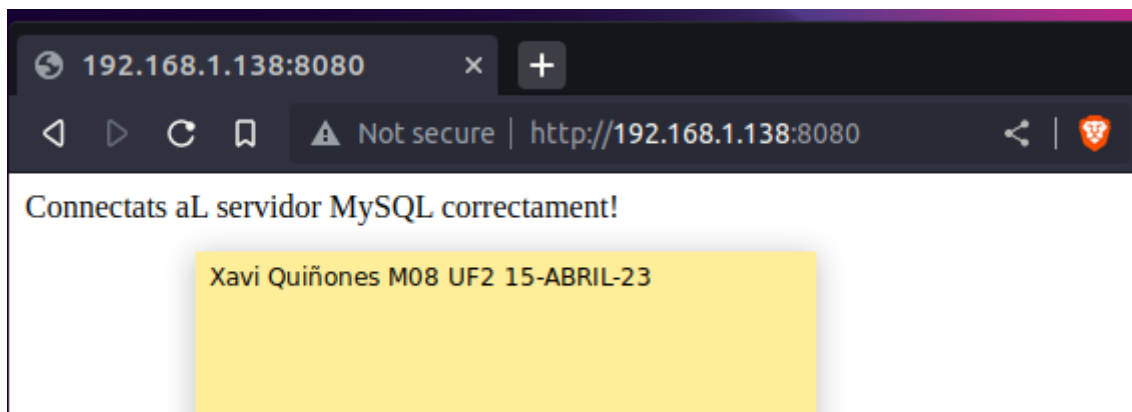
services:
  web:
    container_name: php-apache
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "8080:80"
    volumes:
      - ./site:/var/www/html
  db:
    container_name: db
    image: mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: PASSW_ROOT_MYSQL
      MYSQL_DATABASE: DB_08
      MYSQL_USER: USUARI_MYSQL
      MYSQL_PASSWORD: PASSW_MYSQL
    ports:
      - "3306:3306"
Xavi Quiñones M08 UF2 15-ABRIL-23
```

- Modificamos el archivo `index.php` para probar los nuevos cambios



```
GNU nano 6.2                                index.php *
<?php
// El servei que hem inclòs al docker-compose.yml.
$host = 'db';
// L'usuari de la BD's
$user = 'USUARI_MYSQL';
// El pwd de l'usuari de la BD's
$pass = 'PASSW_MYSQL';
// Comprovació de l'estat de la connexió a MySQL
$conn = new mysqli($host, $user, $pass);
if ($conn->connect_error) {
    die("La connexió a fallat: " . $conn->connect_error);
} else {
    echo "Connectats al servidor MySQL correctament!";
}
?>
```

- Ejecutamos de nuevo **docker-compose up -d** y probamos desde el navegador



## Añadir PhpMyAdmin

```
version: '3'

services:
  web:
    container_name: php-apache
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "8080:80"
    volumes:
      - ./site:/var/www/html
  db:
    container_name: db
    image: mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: PASSW_ROOT_MYSQL
      MYSQL_DATABASE: DB_M08
```

```

MYSQL_USER: USUARI_MYSQL
MYSQL_PASSWORD: PASSW_MYSQL
ports:
  - "3306:3306"
phpmyadmin:
  container_name: phpmyadmin
  image: phpmyadmin
  ports:
    - "80:80"
  environment:
    PMA_HOST: db
    MYSQL_ROOT_PASSWORD: PASSW_ROOT_MYSQL
  depends_on:
    - db

```

- Paramos y eliminamos

```

Stopping php-apache ... done
root@ubuntu:/home/ubuntu/docker# docker-compose down
Removing db ... done
Removing php-apache ... done
Removing network docker_default
root@ubuntu:/home/ubuntu/docker#

```

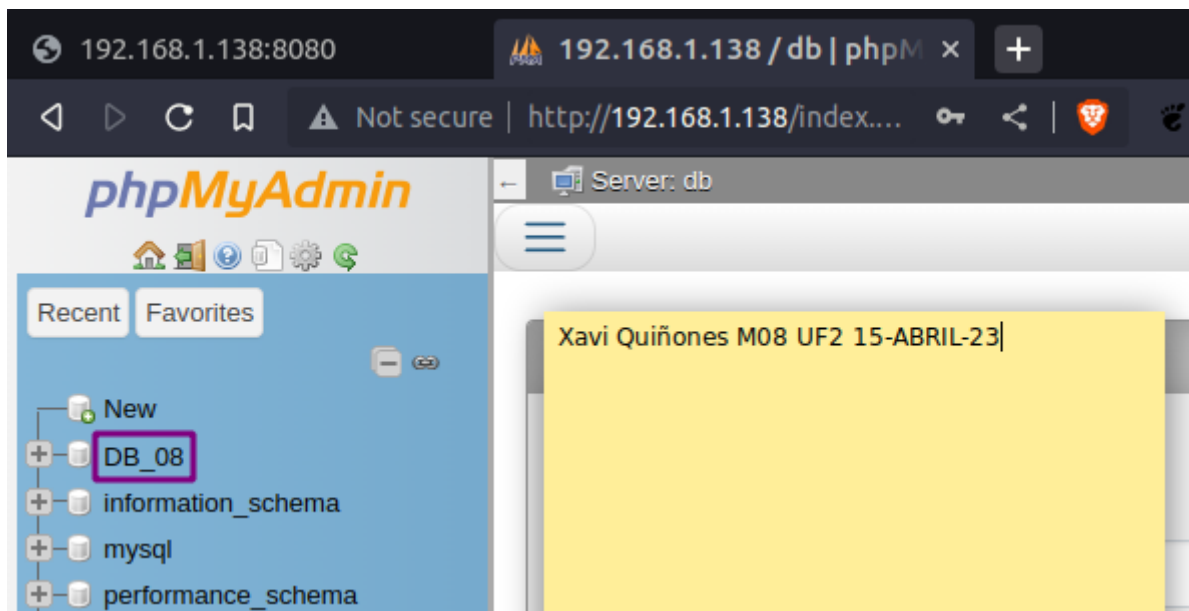
- Editamos el archivo yaml

```

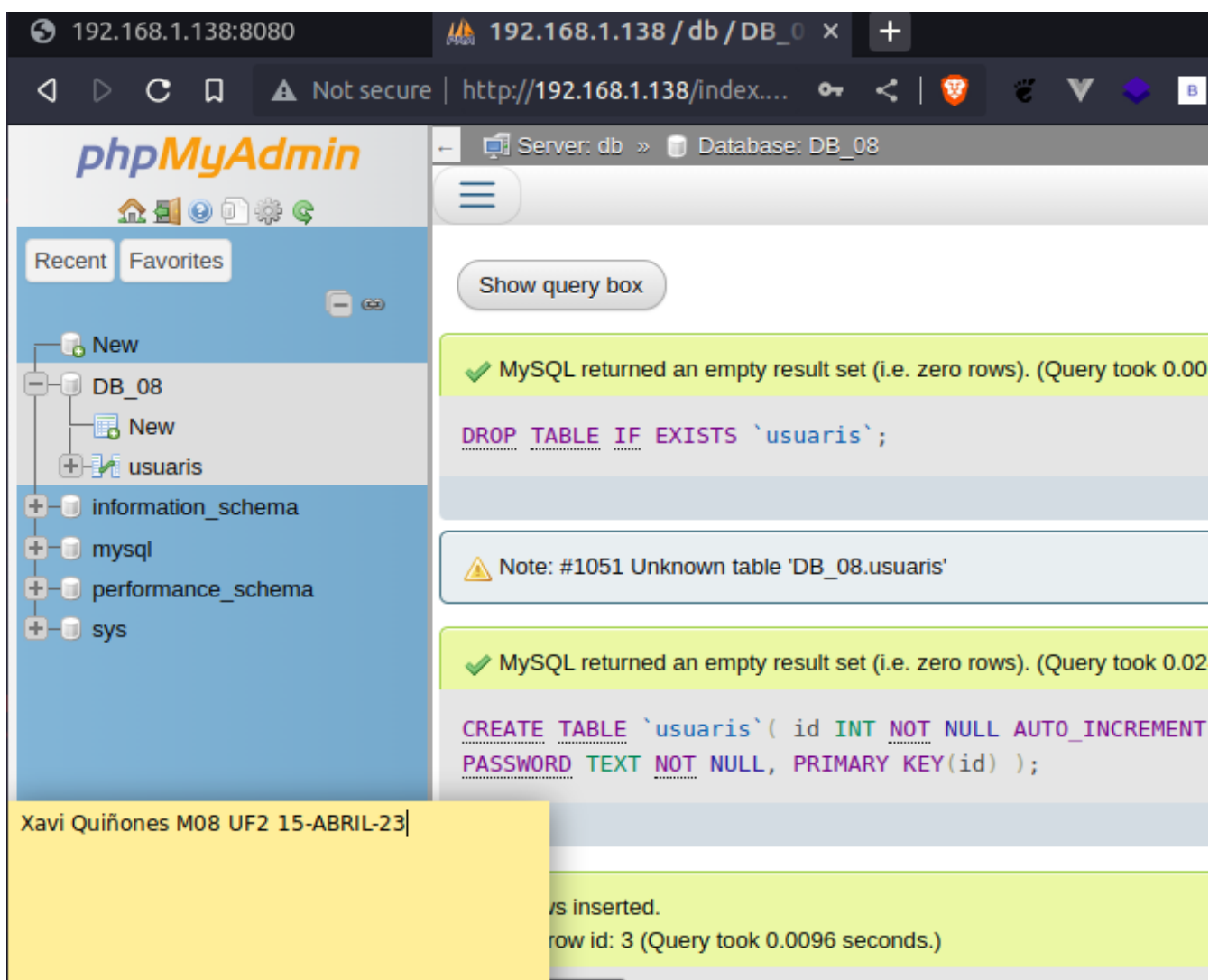
phpmyadmin:
  container_name: phpmyadmin
  image: phpmyadmin
  ports:
    - "80:80"
  envirnoment:
    PMA_HOST: db
    MYSQL_ROOT_PASSWORD: PASSW_ROOT_MYSQL
  depends_on:
    - db

```

- Ejecutamos de nuevo el docker-compose y entramos desde el navegador por el puerto 80 (asignado a PhpMyAdmin)



- Ejecutamos las líneas SQL del pdf para crear una tabla 'usuaris' y añadimos algunos registros



- Editamos index.php para iterar sobre los registros añadidos en la tabla 'usuaris'

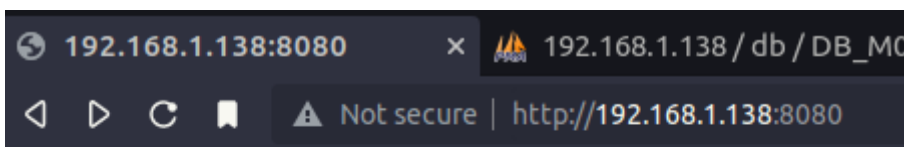
```

<?php
// El servei MySQL al docker-compose.yml.
$host = 'db';
// Nom d'usuari a la BD's
$user = 'USUARI_MYSQL';
// Contrasenya de l'usuari per accedir a la BDs
$pass = 'PASSW_MYSQL';
// Nom de la BDs
$mydatabase = 'BD_M08';
// Comprova l'estat de la connexió a la BD's
$conn = new mysqli($host, $user, $pass, $mydatabase);
// Consulta a la BD's
$sql = 'SELECT * FROM usuaris';
if ($result = $conn->query($sql)) {
    while ($data = $result->fetch_object()) {
        $users[] = $data;
    }
}
foreach ($users as $user) {
    echo "<br>";
    echo "USUARI: " . $user->nomusuari . " PASSW: " . $user->password;
    echo "<br>";
}
?>

```

Xavi Quiñones M08 UF2 15-ABRIL-23

- Comprobamos que funciona



USUARI: admin PASSW: password

USUARI: Núria PASSW: aquest és el meu password

USUARI: Pau PASSW: 12345678

Xavi Quiñones M08 UF2 15-ABRIL-23

He tenido que realizar algun cambio ya que en el *docker-compose.yml* tenía puesto DB\_M08 y en el *index.php* BD\_M08. Es posible que en alguna captura eso esté mal.

## Activitat 2 - Persistencia

- Creamos un volumen para montar `./data` en `/var/lib/mysql`, permitiendo persistir la información de la DB MySQL

```
MYSQL_DATABASE: DB_M08
MYSQL_USER: USUARI_MYSQL
MYSQL_PASSWORD: PASSW_MYSQL
ports:
  - "3306:3306"
volumes:
  - ./data:/var/lib/mysql/
phpmyadmin:
  container_name: phpmyadmin
  image: phpmyadmin
  ports:
    - "80:80"
  environment:
```