

# Activitat 1

## ***Docker-compose i servidors d'aplicacions***

Version 1.01

# Taula de continguts

1. Teoria .....	1
1.1. Pàgines web PHP utilitzant contenidors Docker amb PHP, Apache i MySQL .....	1
1.1.1. Pre-requisits .....	1
1.1.2. Construcció de l'arxiu <i>docker-compose</i> .....	2
1.1.3. Configuració i execució d'una instància local d'un servidor Apache amb PHP .....	2
1.2. Configuració del contenidor amb el MySQL .....	3
1.2.1. Executant consultes SQL utilitzant scripts PHP .....	4
1.2.2. Configurant el PHPMyAdmin .....	5
1.2.3. Obtenir registres i mostrar-los en una web basada en PHP .....	5
2. Activitats .....	8
3. Entrega .....	9

# 1. Teoria

## 1.1. Pàgines web PHP utilitzant contenidors Docker amb PHP, Apache i MySQL

La tecnologia de contenidors està creixent cada dia. Aquesta tecnologia fa que desenvolupar una aplicació sigui més fàcil i ràpida. Disposen d'una arquitectura neta que garanteix que els serveis que utilitzen les aplicacions estiguin dividits en serveis més petits, les imatges. Això permet configurar cada servei de manera independent sense que això afecti als altres serveis.

En aquest cas, Docker ens proporciona l'arxiu *docker-compose* que permet configurar tots els entorns de l'aplicació i executar algunes comandes per acabar de configurar l'aplicació de manera fàcil, ràpida i elegant.

En el cas d'haver d'executar una aplicació en PHP, s'han d'instal·lar tots els entorns necessaris per tal que s'executin *scripts* PHP. Com a mínim es necessita un servidor web *Apache* instal·lat al servidor i molt possiblement un sistema gestor de BD's, com MySQL. Després, clar, s'han de configurar per tal de poder executar el lloc web basat en PHP.

Amb l'ús de Docker, les coses són més manejables. Docker permet configurar l'aplicació amb cada servei executant-se com a micro-servei. D'aquesta manera s'estableix un únic arxiu (YML) que aïllarà tots els serveis que necessita l'aplicació per tal d'executar-se. A l'arxiu s'hi configura el servidor Apache amb PHP i el SGBD's. S'hi especifiquen tots els paràmetres que es necessiten per tal que l'aplicació s'executi.

La principal avantatge que ofereixen els contenidors és un entorn escalable per executar els serveis que utilitzen les aplicacions. Assegura que les pràctiques d'integració contínua i desplegament continu (CI/CD) milloren per tot l'equip de desenvolupament i operacions (DevOps).

Compartint l'arxiu YML amb tots els membres de l'equip s'estableixen els entorns necessaris, per tot l'equip, independentment del SO en què s'executa. Així tots els membres de l'equip de desenvolupament poden sincronitzar la feina sense trencar el codi.

### Passos a seguir

- Configurar i executar una instància local d'un servidor Apache amb PHP
- Servir una pàgina web dinàmica basada en PHP
- Configuració d'una BD's MySQL per tal d'executar *scripts* SQL, obtenir registres i mostrar-los al lloc web basat en PHP

Per fer-ho s'utilitzen les imatges disponibles al *Docker-Hub*.

### 1.1.1. Pre-requisits

- Assegurar-se que el dimoni de Docker està executant-se a l'ordinador
- Coneixement bàsic de PHP i consultes SQL
- Coneixement bàsic de com crear i executar imatges Docker des d'un arxiu *DockerFile*

- Entendre com funcionen els contenidors
- Coneixements bàsics sobre com executar ordres *Docker* i *Docker-compose*

### 1.1.2. Construcció de l'arxiu *docker-compose*

El *docker-compose* permet establir els paràmetres de les imatges necessàries per tal de poder executar l'aplicació.

Com hem vist, al *docker-compose* el primer que hem de fer és indicar quines imatges farem servir.

*Exemple de docker-compose:*

```
services:
  php-apache-environment:
    container_name:
```

### 1.1.3. Configuració i execució d'una instància local d'un servidor Apache amb PHP

#### Entorns a configurar

- *container name*: nom amb el que volem anomenar el contenidor (per exemple: *container\_name: php-apache*)
- *image*: la imatge oficial de PHP que es vol utilitzar del *Docker hub*, que serà: *image: php:8.0-apache*
- *volume*: el directori on es troba el codi font de la nostra aplicació

Com per exemple:

*Exemple de definició de volums:*

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

- *ports*: defineix els ports des d'on s'executarà l'*script*.

For example:

*Exemple de definició de ports:*

```
ports:
  - 8080:80
```

Això vol dir que estem configurant un servidor *Apache* per exposar el port 80. El port 8080 arriba als *scripts* PHP i els executa en un navegador des dels contenidors *Docker*.

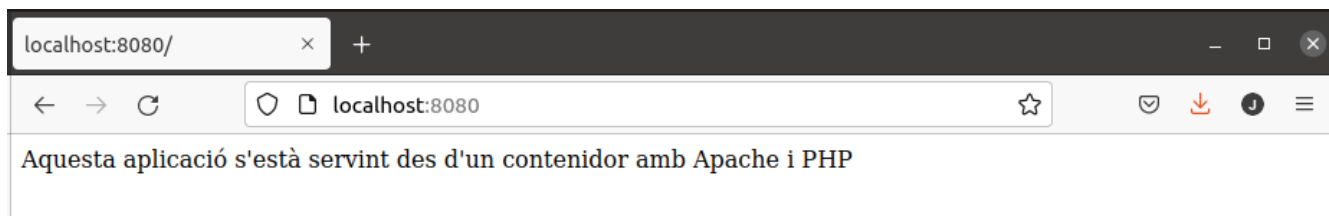


Figura 1. Exemple d'aplicació web PHP dockeritzada

Exemple d'script senzill de PHP

```
<?php
    echo "Aquesta aplicació s'està servint des d'un contenidor amb Apache i PHP";
?>
```

Per accedir-hi es fa mitjançant la següent adreça: <http://localhost:8080> (si li hem assignat aquest port).

## 1.2. Configuració del contenidor amb el MySQL

Per poder fer consultes a una base de dades des de l'aplicació haurem de crear un altre servei dins del contenidor PHP. En aquest cas un servei que ens proveeixi el servei MySQL.

S'ha d'afegir el servei MySQL a l'arxiu *docker-compose.yml*. A més a més, haurem de configurar-ne alguns paràmetres com:

- L'accés amb contrasenya al servidor MySQL, usuari i contrasenya:
  - `MYSQL_USER: USUARI_MYSQL`
  - `MYSQL_PASSWORD: PASSW_MYSQL`
- L'accés a la BD's:
  - `MYSQL_DATABASE: BD_M08`
- Una política de reinici: reiniciar-se sempre
- Els ports

Exemple de definició del servei de MySQL:

```
db:
  container_name: db
  image: mysql
  restart: always
  environment:
    MYSQL_ROOT_PASSWORD: PASSW_ROOT_MYSQL
    MYSQL_DATABASE: BD_M08
    MYSQL_USER: USUARI_MYSQL
    MYSQL_PASSWORD: PASSW_MYSQL
  ports:
    - "3306:3306"
```

S'han d'afegir, també algunes eines de suport dins del contenidor PHP per tal que tant el servei de PHP com el de MySQL funcionin correctament. Aquesta eina inclou el *mysqli*.

Per incloure aquesta eina s'ha de crear un arxiu *Dockerfile* que inclogui les configuracions.

*Exemple d'arxiu Dockerfile:*

```
FROM php:apache
RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli
RUN apt-get update && apt-get upgrade -y
```

Un cop fet això, s'ha de construir aquesta imatge dins del servei del servidor web amb PHP a l'arxiu *docker-compose.yml*. El servei del servidor web, dependrà del servei que connecta amb la BD's.

### 1.2.1. Executant consultes SQL utilitzant scripts PHP

S'ha de comprovar que el contenidor creat funciona com s'espera. Per tal de fer-ho modificarem l'*script* PHP fent que es connecti a la BD's.

*Exemple d'script 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!";
}

?>
```



*Figura 2. Exemple de connexió a BD's des d'una aplicació web PHP.*

### 1.2.2. Configurant el PHPMyAdmin

Un cop comprovat el correcte funcionament del MySQL. Haurem d'extreure dades d'una BD's i mostrar-les en una pàgina web mitjançant PHP.

Suposant que l'aplicació interactua amb la BD's, probablement es voldrà una interfície per interactuar amb les dades. Per fer-ho s'afegirà el servei *PHPMyAdmin* per proporcionar una interfície per interactuar amb la BD's MySQL.

S'ha d'afegir, doncs, al *docker-compose*.

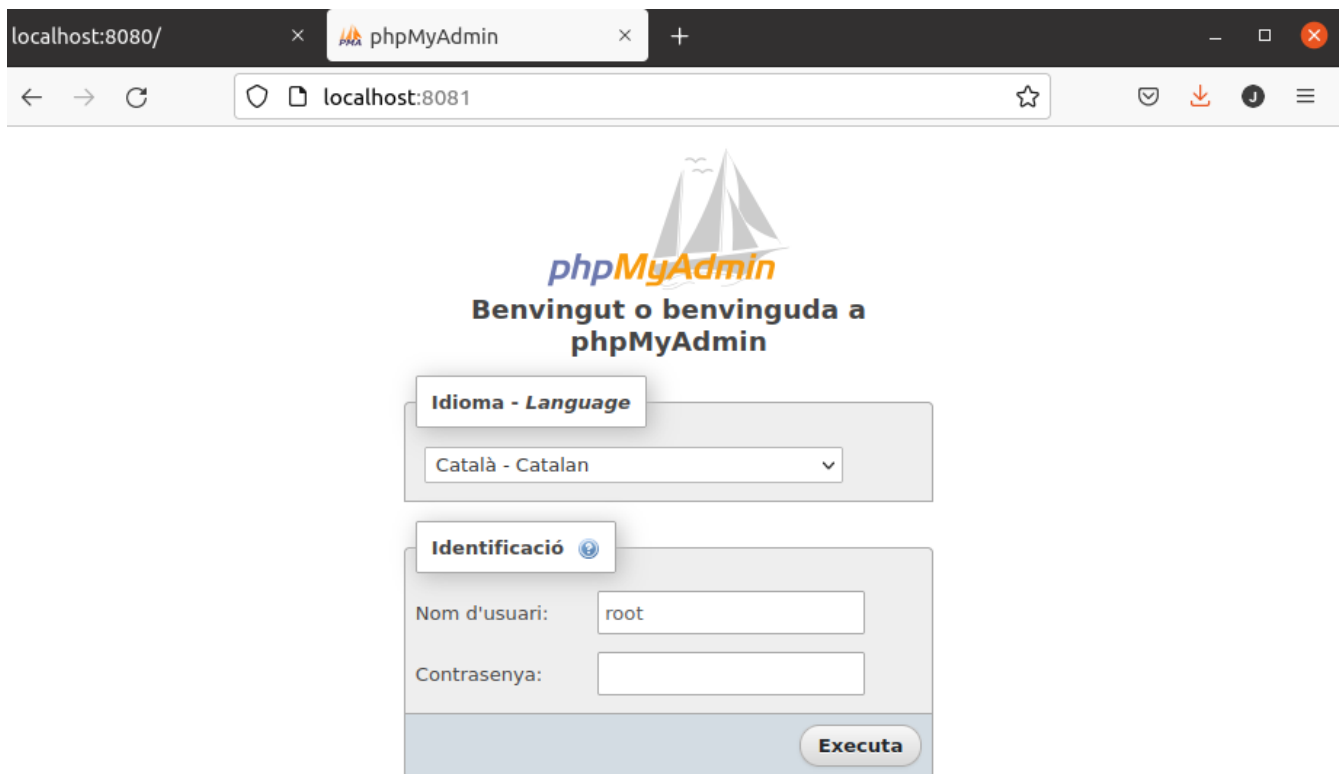


Figura 3. Accés a PHPMyAdmin

### 1.2.3. Obtenir registres i mostrar-los en una web basada en PHP

S'ha de crear una taula i omplir-la amb alguns registres. Seleccioneu la base de dades i executeu les següents consultes:

Consultes a executar:

```
drop table if exists `usuaris`;
create table `usuaris` (
  id int not null auto_increment,
  nomusuari text not null,
  password text not null,
  primary key (id)
);
insert into `usuaris` (nomusuari, password) values
  ("admin","password"),
  ("Núria","aquest és el meu password"),
```

```
("Pau","12345678");
```



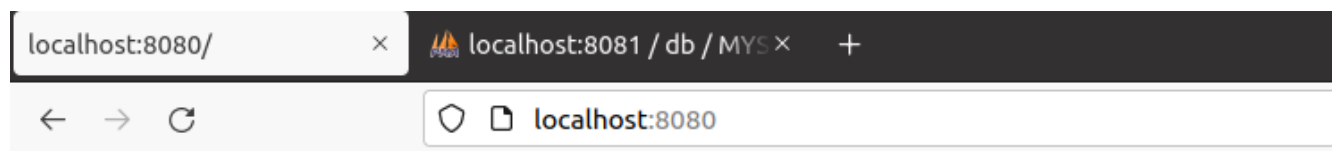
Figura 4. Consultes introduïdes al PHPMyAdmin

Fem una consulta a la base de dades amb

```
<?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>";  
}  
?>
```



Per accedir-hi es fa mitjançant la següent adreça: <http://localhost:8080>



USUARI: admin PASSW: password

USUARI: Núria PASSW: aquest és el meu password

USUARI: Pau PASSW: 12345678

*Figura 5. Resultat de l'aplicació web basada en PHP i MySQL*

## 2. Activitats

1-. Desplega un contenidor amb el servei web i php. Aquest servei contindrà un volum amb un codi en php que comprovi la connexió amb la base de dades. Desplega un altre contenidor amb amb *MySQL*, i finalment un amb *PHPMyAdmin*. Comprova el seu funcionament accedint a *PHPMyAdmin* i mostrant el web que comprova la connexió.

2-. A l'activitat anterior has creat una aplicació web que interactua amb una BD's. Aquesta BD's, però, no és capaç de mantenir les dades de manera persistent. Investiga i realitza allò que calgui per tal que les dades introduïdes a la BDs siguin persistents.

### 3. Entrega

Heu d'entregar un document ben format (portada, índex, contingut...).

Al document hi heu d'explicar els passos que realitzeu i hi heu de posar captures de pantalla on es demostrï que vosaltres sou els autors de la feina realitzada i que heu assolit els objectius de la tasca.



**IMPORTANT:**

- La correcció del document, que inclou correcció lingüística i presentació forma part de la nota que obtindreu
- L'entrega ha de ser un arxiu en format *pdf* de nom: *nom\_cognomsAct[número de l'activitat].pdf*
- També heu d'entregar els arxius de configuració utilitzats (*Dockerfile*, *docker-compose*,...)