

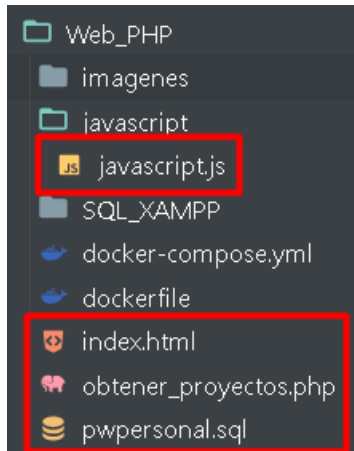
# Contenedores Docker

26/11/2024

Docker-compose/Dockerfile Web PHP sobre LAMP.....	2
Troubleshooting.....	6
Error: “phpMyAdmin no se conecta a MySQL” .....	6
Error: “Cannot access the localhost website in Docker” .....	6
La extensión <i>mysqli</i> no está habilitada en PHP.....	6
Advertencia de Apache: “Could not reliably determine the server’s fully qualified domain name” ..	7

## Docker-compose/Dockerfile Web PHP sobre LAMP

Por último tenemos la página Web con PHP y MySQL con LAMP. Para esta página debemos usar nuestra página personal realizada anteriormente por lo que cogemos los archivos y los movemos a la carpeta donde montaremos nuestro docker:



Una vez más tendremos nuestros correspondientes archivos docker-compose.yml y dockerfile. Otra vez con sus respectivas configuraciones, en el caso del docker-compose.yml tendremos:

```
version: "3.9"

services:

  web:

    image: php:8.2-apache

    container_name: php_web

    volumes:

      - ../var/www/html

    ports:

      - "8080:80"

    depends_on:

      - db

  db:
```

```
image: mysql:8.0

container_name: mysql_db

environment:

    MYSQL_ROOT_PASSWORD: root

    MYSQL_DATABASE: pwpersonal

volumes:

    - db_data:/var/lib/mysql

    - ./pwpersonal.sql:/docker-entrypoint-initdb.d/pwpersonal.sql

ports:

    - "3306:3306"

phpmyadmin:

    image: phpmyadmin:latest

    container_name: phpmyadmin

    environment:

        PMA_HOST: db

        MYSQL_ROOT_PASSWORD: root

    ports:

        - "8081:80"

    depends_on:

        - db

volumes:

    db_data:
```

Y en el caso del archivo docker-file:

```
FROM php:8.2-apache
```

```

RUN apt-get update && apt-get install -y --no-install-recommends \
    libpng-dev libjpeg-dev libfreetype6-dev zip unzip git \
    && docker-php-ext-install mysqli pdo pdo_mysql \
    && apt-get clean && rm -rf /var/lib/apt/lists/*

COPY . /var/www/html

RUN chown -R www-data:www-data /var/www/html && chmod -R 755 /var/www/html

```

Por último ejecutaremos el comando: ***docker-compose up --build -d***

Ahora comprobaremos el correcto funcionamiento de la página. Es posible que salga el siguiente error:

**"Class 'mysqli' not found"** este error nos comunica que la extensión mysqli no se ha habilitado correctamente, ya que por mucho que se haya instalado con el dockerfile no se ha habilitado. Para solucionar esto simplemente realizaremos lo siguiente, accedemos al contenedor php:

***docker exec -it php\_web bash***

Instalamos las dependencias:

***apt-get update***

***apt-get install -y libmysqlclient-dev***

Instala la extensión mysqli:

***docker-php-ext-install mysqli***

Reiniciar Apache:

***service apache2 restart***

Y volvemos a comprobar si mysqli esté habilitada:

***php -m | grep mysqli***

Reiniciamos el contenedor PHP:

***docker restart php\_web***

Reiniciar y reconstruye todo:

***docker-compose down - --rm all***

***docker-compose up - --build -d***

Una vez hecho esto ya habremos solucionado el tema del mysql por lo que nos funcionara ya la página correctamente, si no lo hacíamos los proyectos que se deberían obtener de la BBDD SQL mediante el php (obtener\_proyectos.php) no se obtendrán y veremos el apartado de “Mis proyectos” vacío ya que se cargan los datos desde la base.

## Troubleshooting

### Error: “phpMyAdmin no se conecta a MySQL”

En phpMyAdmin me aparecía este error indicando que no se puede conectar al servidor MySQL. El error ocurrió ya que el phpMyAdmin no estaba configurado correctamente para conectarse al contenedor de MySQL.

Me asegure de que el servicio **db** está configurado como host en el archivo *docker-compose.yml*:

*phpmyadmin:*

*image: phpmyadmin:latest*

*environment:*

*PMA\_HOST: db*

*MYSQL\_ROOT\_PASSWORD: root*

Reinicie el contenedor mediante:

*docker-compose restart*

### Error: “Cannot access the localhost website in Docker”

Aunque el contenedor PHP o la aplicación mediaWIKI o WordPress estaban funcionando no podía acceder a la web mediante localhost, ocurrió ya que el puerto del host no estaba correctamente mapeado al puerto del contenedor y el firewall del S.O podría estar bloqueando el acceso. Para ello:

1. Verifique puertos configurados en *docker-compose.yml*
2. Revise los contenedores en ejecución y sus puertos con: ***docker ps***
3. Cambiar el puerto de la configuración del *docker-compose* ya que otro servicio estaba ya usandolo
4. Reinicie el contenedor: ***docker-compose down / docker-compose up -d***

### La extensión *mysqli* no está habilitada en PHP

Mi página PHP no funcionaba correctamente, me cargaba bien pero no me cargaban los proyectos con el php que cogía la información de la base de datos en phpmyadmin. El error que generaba la respuesta al inspeccionar la red con el F12 en el navegador era el siguiente:

*Fatal error: Uncaught Error: Class "mysqli" not found...*

La extensión *mysqli* (necesaria para interactuar con MySQL) no estaba habilitada en el contenedor PHP,

7

tuve que acceder al contenedor PHP:

```
docker exec -it php_web bash
```

Instalar las extensiones manualmente:

```
apt-get update
```

```
apt-get install -y libmysqlclient-dev
```

```
docker -php-ext-install mysqli
```

Y reiniciar el servicio de apache dentro del contenedor:

```
service apache2 restart
```

Por último verifique que mysqli estaba ya correctamente habilitado mediante el comando:

```
php -m | grep mysqli
```

Si no está habilitado no devuelve nada al ejecutar el comando.

### **Advertencia de Apache: “Could not reliably determine the server’s fully qualified domain name”**

Me aparecia en los logs del contenedor PHP lo siguiente:

```
AH00558: Could not reliably determine the server's fully qualified domain name...
```

Esto ocurrió debido a que Apache no tenía configurado un servername, tuve que editar la configuración de apache en el dockerfile:

```
RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
```

Y reconstruir los contenedores

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

