

XAMP CON DOCKER PARA JAVA

PROGRAMACIÓN
CFGS DAW

Autor: Francisco Aldarias Raya

paco.aldarias@ceedcv.es

2020/2021

Fecha 14/04/21

Versión:210414.1103


Licencia




Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

código

Revisión:

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN.....	3
2. REPOSITORIO.....	3
3. DOCKER COMPOSE.....	4
4. BASE DE DATOS.....	6
5. COMPROBAR LA IP DE LOS SERVIDORES.....	6
6. APLICACIÓN java.....	6
7. APLICACIÓN WEB.....	8
8. MYSQL.....	9
9. APACHE.....	10
10. OPERACIONES SOBRE DOCKER-COMPOSE.....	11
11. BIBLIOGRAFÍA.....	11

UD013. XAMP CON DOCKER PARA JAVA

1. INTRODUCCIÓN

La versión que vamos a explicar es como instalar XAMP con docker. El cual es un sistema de Linux Apache2 Mysql y Php para Java con Maven y jdbc. . Requiere tener instalado en el ordenador docker, docker-compose y git. Utiliza docker composer para descarga de ficheros de instalación. Los ficheros del repositorio son para usar con un proyecto de Netbeans.

2. REPOSITORIO

Nos situamos dentro de la carpeta Netbeansproject.

```
cd ~/Netbeansproject
```

Clonar repositorio en la carpeta de tu usuario. Abrimos la terminal y escribimos:

```
git clone https://bitbucket.org/pacoaldarias/pacoaldariasprg2021xamp
```

Nos situamos en la carpeta clonada:

```
cd pacoaldariasprg2021lamp/
```

Nos mostrara los siguientes archivos:

```
paco@lauritahost:~/ownCloud$ cd NetBeansProject/
paco@lauritahost:~/ownCloud/NetBeansProject$ ls
dockerlamp          pacoaldariasprg2021lamp
pacoaldariasprg2021a pacoaldariasprg2021old
paco@lauritahost:~/ownCloud/NetBeansProject$ cd pacoaldariasprg2021lamp/
paco@lauritahost:~/ownCloud/NetBeansProject/pacoaldariasprg2021lamp$ ls -la
total 56
drwxrwxr-x 8 paco paco 4096 abr 10 20:28 .
drwxrwxr-x 6 paco paco 4096 abr 10 20:25 ..
drwxrwxr-x 2 paco paco 4096 abr  8 13:26 conf
-rw-rw-r-- 1 paco paco 1055 abr  8 13:26 docker-compose.yml
-rw-rw-r-- 1 paco paco  225 abr  8 13:26 Dockerfile
drwxrwxr-x 2 paco paco 4096 abr  8 13:26 dump
drwxrwxr-x 8 paco paco 4096 abr 10 20:50 .git
-rw-rw-r-- 1 paco paco 1210 abr  8 13:26 LICENSE
-rw-rw-r-- 1 paco paco   70 abr  8 13:26 php.Dockerfile
-rw-rw-r-- 1 paco paco  885 abr  8 03:37 pom.xml
-rw-rw-r-- 1 paco paco  683 abr 10 20:26 README.md
drwxrwxr-x 3 paco paco 4096 abr  8 03:29 src
drwxrwxr-x 7 paco paco 4096 abr  8 03:33 target
drwxrwxr-x 2 paco paco 4096 abr  8 13:26 www
paco@lauritahost:~/ownCloud/NetBeansProject/pacoaldariasprg2021lamp$
```

3. DOCKER COMPOSE

Docker Compose permite poner un guion para descarga e instalación del software que se quiere instalar.

```
paco@lauritahost:~/ownCloud/NetBeansProject/pacoaldariasprg2021lamp$ sudo apt install docker-compose
[sudo] contraseña para paco:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
docker-compose ya está en su versión más reciente (1.25.0-1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 4 no actualizados.
```

Ejecutamos compose:

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

La configuración de usuario y password de la base de datos se encuentra en el fichero docker-compose.yml

```
version: "3.1"
services:
  www:
    build: .
    ports:
      - "8001:80"
    volumes:
      - ./www:/var/www/html/
    links:
      - db
    networks:
      - default
  db:
    image: mysql:8.0
    ports:
      - "3306:3306"
    command: --default-authentication-plugin=mysql_native_password
    environment:
      MYSQL_DATABASE: myDb
      MYSQL_USER: user
      MYSQL_PASSWORD: test
      MYSQL_ROOT_PASSWORD: test
    volumes:
      - ./dump:/docker-entrypoint-initdb.d
      - ./conf:/etc/mysql/conf.d
      - persistent:/var/lib/mysql
    networks:
      - default
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    links:
      - db:db
    ports:
      - 8000:80
    environment:
      MYSQL_USER: user
      MYSQL_PASSWORD: test
      MYSQL_ROOT_PASSWORD: test
volumes:
  persistent:
```

Comprobamos que está todo instalado con :

```
docker ps -a
```

Paramos docker:

```
docker-compose down
```

4. BASE DE DATOS

Se encuentra en dump/mydb.sql

Es el fichero que se ejecuta al inicio. Contiene la base de datos y los datos.

```
CREATE TABLE `Person` (  
  `id` int(11) NOT NULL,  
  `name` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
INSERT INTO `Person` (`id`, `name`) VALUES  
(1, 'William'),  
(2, 'Marc'),  
(3, 'John');
```

5. COMPROBAR LA IP DE LOS SERVIDORES.

Usaremos el comando:

```
docker inspect -f '{{.Name}} - {{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $(docker ps -aq)
```

```
paco@lauritahost:~/ownCloud/NetBeansProject/pacoaldariasprg2021lamp$ docker inspect -f '{{.Name}} - {{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' $(docker ps -aq)  
/pacoaldariasprg2021lamp_phpmyadmin_1 - 172.20.0.4  
/pacoaldariasprg2021lamp_www_1 -  
/pacoaldariasprg2021lamp_db_1 - 172.20.0.2  
/pacoaldariasprg2021_www_1 - 172.19.0.2  
/pacoaldariasprg2021_phpmyadmin_1 - 172.19.0.3  
/pacoaldariasprg2021_db_1 - 172.19.0.2
```

Se puede ver que el servidor mysql tiene ip 172.20.0.2.

6. APLICACIÓN JAVA

Podemos ejecutar la aplicación java adaptando los al servidor de base de

datos. El fichero Main.java seria:

```
package com.ceedcv.pacoaldariasprg2021a;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 *
 * @author Paco Aldarias <paco.aldarias@ceedcv.es>
 */
public class Main {

    public static void main(String[] args) {

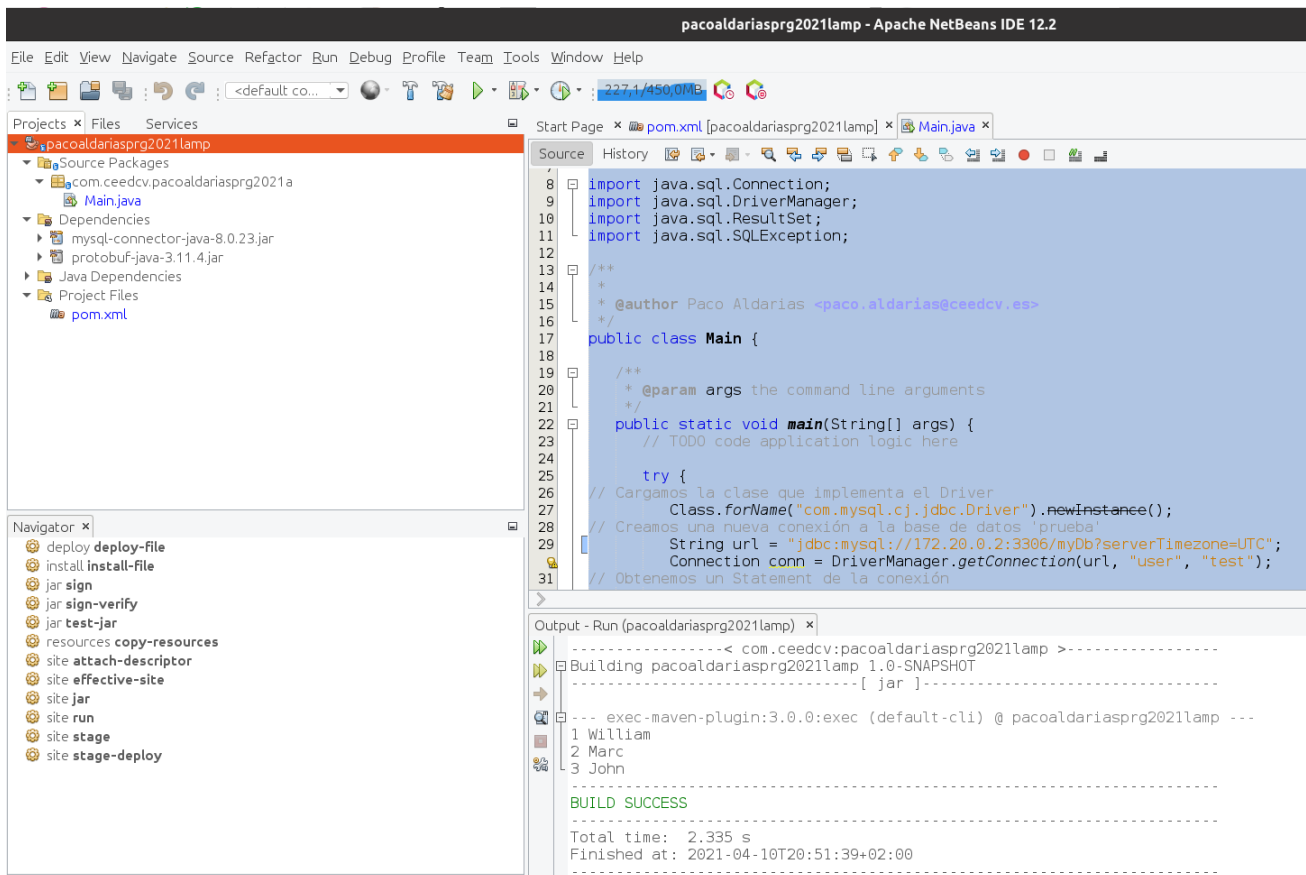
        try {
            // Cargamos la clase que implementa el Driver
            Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
            // Creamos una nueva conexión a la base de datos 'prueba'
            String url = "jdbc:mysql://172.20.0.2:3306/myDb?serverTimezone=UTC";
            Connection conn = DriverManager.getConnection(url, "user", "test");
            // Obtenemos un Statement de la conexión
            java.sql.Statement st = conn.createStatement();
            // Ejecutamos una consulta SELECT para obtener la tabla vendedores
            String sql = "SELECT * FROM Person";
            ResultSet rs = st.executeQuery(sql);
            // Recorremos todo el ResultSet y mostramos sus datos
            while (rs.next()) {
                int id = rs.getInt("id");
                String nombre = rs.getString("name");

                System.out.println(id + " " + nombre);
            }
            // Cerramos el statement y la conexión
            st.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException | IllegalAccessException |
InstantiationException ex) {
            ex.printStackTrace();
        }
    }
}
```

```
}

```

El resultado permite ver como se conecta a la base de datos y muestra los datos:



7. APLICACIÓN WEB

La aplicación web se deja en la carpeta www en el fichero index.php

```
<html>
<body>
  <?php echo "<h1>Hi! I'm happy</h1>"; ?>

  <?php

  // Connexion et sélection de la base
  $conn = mysqli_connect('db', 'user', 'test', "myDb");
```



```
$query = 'SELECT * From Person';
$result = mysqli_query($conn, $query);

echo '<table class="table table-striped">';
echo '<thead><tr><th></th><th>id</th><th>name</th></tr></thead>';
while($value = $result->fetch_array(MYSQLI_ASSOC)){
    echo '<tr>';
    echo '<td><a href="#"><span class="glyphicon glyphicon-search"></span></a></td>';
    foreach($value as $element){
        echo '<td>' . $element . '</td>';
    }

    echo '</tr>';
}
echo '</table>';

/* Libération du jeu de résultats */
$result->close();

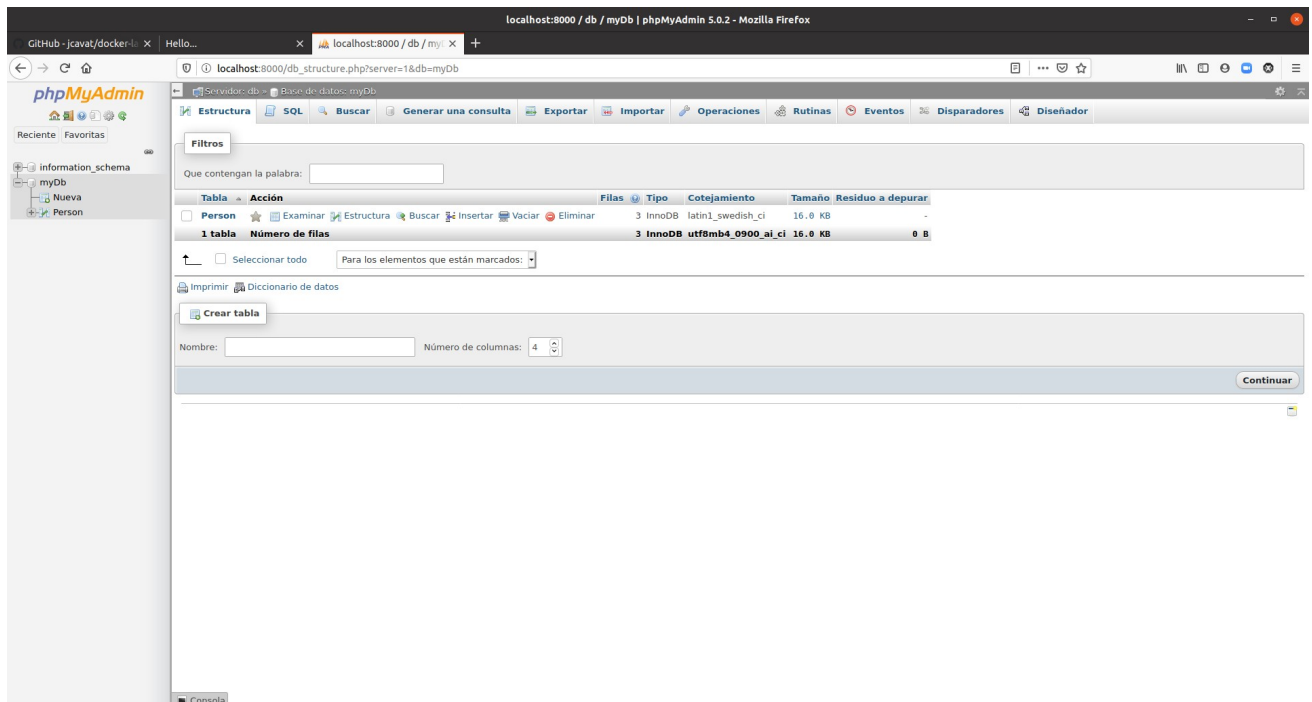
mysqli_close($conn);

?>
</div>
</body>
</html>
```

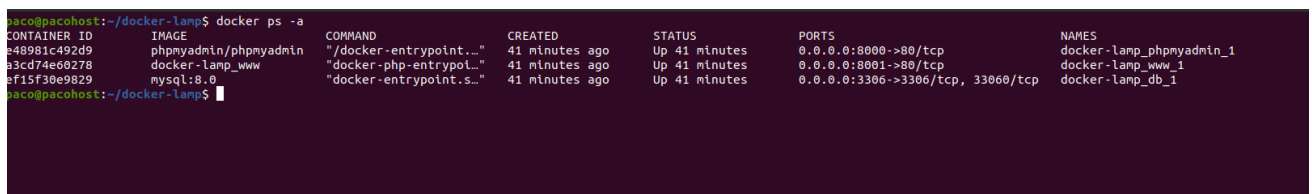
8. MYSQL

Accedemos al servidor mysql con phpmyadmin:

<http://localhost:8000>

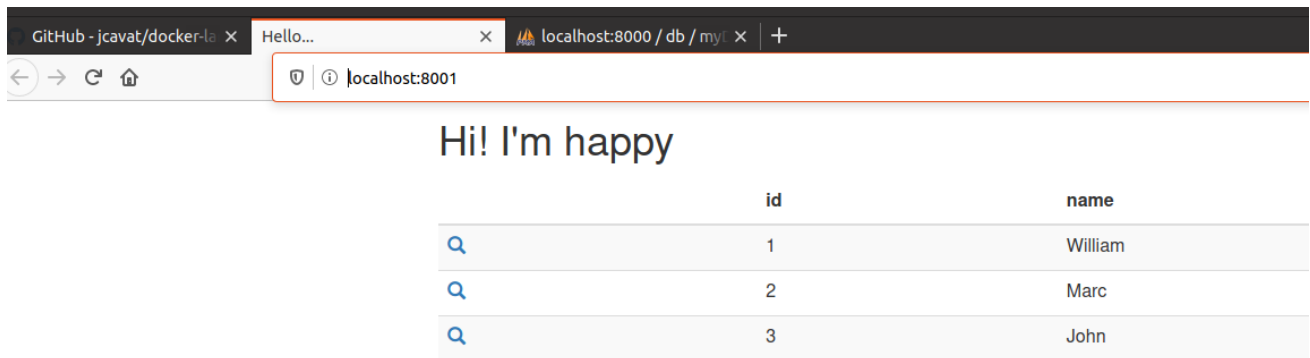


Para acceder al servidor mysql con la terminal:
`docker-compose exec db mysql -u root -p`



9. APACHE

Accedemos al servidor web con:
<http://localhost:8001/>



10. OPERACIONES SOBRE DOCKER-COMPOSE

Las operaciones básicas sobre docker composer son:

- *docker-compose up*: da instrucciones a Docker para crear el contenedor y ejecutarlo.
- *docker-compose down*: apaga todo los servicios que levantó con docker-compose up.
- *docker-compose ps*: permite ver los contenedores funcionando.
- *docker-compose exec*: permite ejecutar un comando a uno de los servicios levantados de Docker-compose.

11. BIBLIOGRAFÍA

- Set up a LAMP server with Docker
https://linuxhint.com/lamp_server_docker/