Despliegue y ejecución de una mini-aplicación en un entorno “contenedorizado”

Contenidos

Objetivo 2
Comprobaciones a realizar con todo contenedor e imagen 2
Cada vez que se cree un contendor 2
Cada vez que se descargue una imagen: 3
Cada vez que se arranque una imagen en un contenedor: 3
Contenedor con imagen MySQL 3
Contenedor con imagen Apache 4
Comprobaciones finales 4
ANEXO I: Scripts de creación de tablas (DDL) e inserción de datos (DML). 5

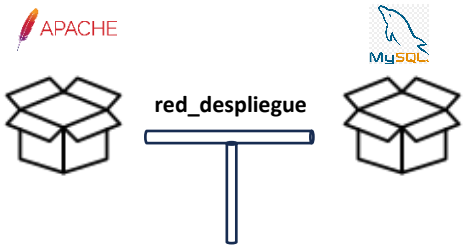
Objetivo

Se quiere generar con Docker un entorno de desarrollo "contenedorizado", en el que se ejecute una mini-aplicación en PHP sobre Apache (dentro de un contenedor) que consulte una base de datos MySQL (dentro de otro contenedor).

Para ello, se crearán dos contenedores:

- Uno que contenga MySQL, en el que se ejecutará un script para crear y rellenar unas tablas de prueba. Este script se encuentra en el ANEXO I.
- Otro que contenga Apache sobre Ubuntu, en el que se desplegará un script de PHP que accederá al contendor de MySQL antes citado. Este script se deberá codificar a partir de documentación en línea:
    - https://www.w3schools.com/php/php_mysql_select.asp
    - https://www.startertutorials.com/ajwt/database-access-in-php.html
    - …

  Y para que funcione correctamente se debe tener en consideración:
    - Host donde se encuentra el sistema gestot de bses de datos, en este ejemplo MySQL.
    - Credenciales (usuario/contraseña) con el que se accede a la base de da a esa base de datos.
    - Nombre de la base de datos donde se ubican las tablas que se va a consultar.



ANFITRIÓN

Los dos contendores deberán encontrarse en red con la máquina anfitriona. Para ello, se arrancarán formando parte de una red Docker, de nombre **red_despliegue**, que se debe haber creado previamente.

Comprobaciones a realizar con todo contenedor e imagen

Cada vez que se cree un contendor

1. Comprobar su estado (Creado, Detenido, Arrancado…), su identificador, su nombre, la imagen que se corre dentro, el comando ejecutable que hace que se mantenga encendido…
2. Inspeccionar qué dirección IP se le ha asignado.

Cada vez que se descargue una imagen:

3. Comprobar su nombre, su tamaño…

Cada vez que se arranque una imagen en un contenedor:

1. Ejecutar una Shell para “entrar” dentro del contendor.
2. Instalar todos los paquetes que se necesiten para que, desde el sistema operativo del contendor, se puedan realizar las verificaciones habituales que se hacen tras la instalación de un servicio:
    a. La versión del sistema operativo (habitualmente Linux) que se está ejecutando (distribución, versión…).
    b. Conectividad con la máquina anfitriona y los demás contendores que se encuentren en su red.
    c. Puertos abiertos.
    d. Procesos en ejecución.
    e. Servicios en ejecución
    f. Usuarios/grupos existentes.
    g. Ubicación de los servicios (ejecutable, archivos de configuración, manual…)
    h. Otras aplicaciones que se considere necesario.

Contenedor con imagen MySQL

1. En un solo comando Docker:
    a. Se comprueba, desde la línea de comandos de Docker, que existe una imagen de nombre **mysql**.
    b. Se descarga esa imagen de DockerHub. La página oficial es https://hub.docker.com/_/mysql. Se recomienda leer en contendor de esta página a que se dan indicaciones de cómo arrancar la imagen en un contendor, qué parámetros se pueden usar…
    c. Se ejecutará la imagen en un contendor, de nombre **contiene_mysql**.
        i. En el propio comando de arranque de la imagen en el contendor, y para facilitar el uso de MySQL, se creará una variable de entorno con la que se le indica a MySQL cuál es la contraseña de root, que será **admin**. Para conocer qué variables de entorno hay y cómo se pueden usar, se puede consultar la página oficial de la imagen, antes citada.
        ii. El contendor se ejecutará en segundo plano (*detached*).
2. En la máquina anfitriona, instalar el IDE *MySQL Workbench Community Edition* para conectarse al servidor de MySQL y ejecutar los scripts DDL y DML.
3. Arrancar una shell **bash** en el contenedor y:
    a. Comprobar la versión de Linux que está corriendo en el contendor.
    b. Instalar todos los paquetes necesarios.

c. Instalar un cliente de MySQL de línea de comandos (mysql-client) y comprobar que las operaciones DDL y DML realizadas con Workbench se repercutieron correctamente en la base de datos.

Contenedor con imagen Apache

Se va a construir una imagen que contenga un Apache con la capacidad de ejecutar scripts PHP. Para ello se partirá de una imagen de Ubuntu sobre la que se instalará todos los paquetes necesarios. En el siguiente enlace se detalla cómo realizar esta tarea:

Los pasos a dar son:

1. Se descargará la imagen de Ubuntu, de nombre **ubuntu**.
2. Se ejecuta la imagen en un nuevo contendor de nombre **contiene_apache**.
    a. Se "mapeará" (vinculará) el puerto 8888 de la máquina anfitriona con el puerto 80 del contendor.
3. Se comprueba desde la máquina anfitrión, usando un navegador, que se accede correctamente al Apache existente en el contendor.
4. Se ejecuta un **bash** sobre ese contendor.
    a. Se instalan todos los paquetes necesarios.
    b. Si no está ya operativo, se activa el funcionamiento de PHP en Apache.
    c. Se creará, en el directorio raíz de documentos, un script PHP de prueba de nombre **phpinfo.php** para comprobar que funciona el intérprete de PHP en Apache. El código del script es:

```
<?php
phpinfo();
¿>
```

    d. Se comprueba desde la máquina anfitrión, invocándolo con un navegador, que se ejecuta correctamente al script PHP recién creado.
5. Se copia el script PHP que accede a MySQL desde la máquina anfitriona al contendor.
6. Se comprueba desde la máquina anfitrión, usando un navegador, que se ejecuta correctamente al script PHP recién creado.
7. Una vez que la imagen esté funcionando correctamente dentro del contenedor, se exportará y se subirá a la cuenta personal de DockerHub.

Comprobaciones finales

1. Lista de contenedores con su información asociada.
2. Detener los dos contendores, volver a arrancarlos y comprobar que se sigue ejecutando correctamente la mini-aplicación.

ANEXO I: Scripts de creación de tablas (DDL) e inserción de datos (DML).

```
mysql> CREATE DATABASE SCOTT;
Query OK, 1 row affected (0,03 sec)

mysql> USE SCOTT;
Database changed


DROP TABLE dept;
CREATE TABLE dept (
    deptno decimal(2,0),
    dname varchar(14),
    loc varchar(13),
    CONSTRAINT `PK_dept` PRIMARY KEY(deptno)
);

INSERT INTO dept VALUES (10,'ACCOUNTING','NEW YORK');
INSERT INTO dept VALUES (20,'RESEARCH','DALLAS');
INSERT INTO dept VALUES (30,'SALES','CHICAGO');
INSERT INTO dept VALUES (40,'OPERATIONS','BOSTON');

DROP TABLE emp;
CREATE TABLE emp (
    empno decimal(4,0),
    ename varchar(10),
    job varchar(9),
    mgr decimal(4,0),
    hiredate date,
    sal decimal(7,2),
    comm decimal(7,2),
    deptno decimal(2,0),
    CONSTRAINT `PK_emp` PRIMARY KEY (empno),
    CONSTRAINT `FK_emp_dept_deptno` FOREIGN KEY (deptno)
        REFERENCES dept (deptno),
    CONSTRAINT `FK_emp_emp_mgr` FOREIGN KEY (mgr)
        REFERENCES emp (empno)
);


INSERT INTO emp VALUES
  ('7839','KING','PRESIDENT',NULL,'1981-11-17','5000.00',NULL,'10'),
  ('7566','JONES','MANAGER','7839','1981-04-02','2975.00',NULL,'20'),
  ('7698','BLAKE','MANAGER','7839','1981-05-01','2850.00',NULL,'30'),
  ('7782','CLARK','MANAGER','7839','1981-06-09','2450.00',NULL,'10'),
  ('7788','SCOTT','ANALYST','7566','1982-12-09','3000.00',NULL,'20'),
  ('7902','FORD','ANALYST','7566','1981-12-03','3000.00',NULL,'20'),
  ('7499','ALLEN','SALESMAN','7698','1981-02-20','1600.00','300.00','30'),
  ('7521','WARD','SALESMAN','7698','1981-02-22','1250.00','500.00','30'),
  ('7654','MARTIN','SALESMAN','7698','1981-09-28','1250.00','1400.00','30'),
  ('7844','TURNER','SALESMAN','7698','1981-09-08','1500.00','0.00','30'),
  ('7900','JAMES','CLERK','7698','1981-12-03','950.00',NULL,'30'),
  ('7934','MILLER','CLERK','7782','1982-01-23','1300.00',NULL,'10'),
  ('7876','ADAMS','CLERK','7788','1983-01-12','1100.00',NULL,'20'),
  ('7369','SMITH','CLERK','7902','1980-12-17','800.00',NULL,'20');


DROP TABLE bonus;
```

```
CREATE TABLE bonus (
    ename varchar(10),
    job varchar(9),
    sal decimal(7,2),
    comm decimal(7,2)
);

DROP TABLE salgrade;
CREATE TABLE salgrade (
    grade decimal(2,0),
    losal decimal(7,2),
    hisal decimal(7,2),
    CONSTRAINT `PK_salgrade` PRIMARY KEY (grade)
);
INSERT INTO salgrade VALUES (1,700,1200);
INSERT INTO salgrade VALUES (2,1201,1400);
INSERT INTO salgrade VALUES (3,1401,2000);
INSERT INTO salgrade VALUES (4,2001,3000);
INSERT INTO salgrade VALUES (5,3001,9999);

COMMIT;
```