

Prácticas Docker

1. Enlazar contenedores con redes personalizadas

- Vamos a montar un enlace entre un Joomla y una Base de datos MySQL usando enlaces automáticos basados en redes personalizadas. Es muy similar al anterior, pero Docker se encarga de gestionar las conexiones, los nombres de las máquinas y el resto de componentes



- Comprobamos las redes que tenemos en este momento y que vienen de las prácticas anteriores

docker network ls

NETWORK ID	NAME	DRIVER	SCOPE
3d8689b8a3ea	bridge	bridge	local
81ce05a3ba16	host	host	local
8e83268b846d	net1	bridge	local
31ed5d426215	net2	bridge	local
17052d6bd175	none	null	local

- En este caso vamos a usar la red net1 para realizar el ejemplo.
- Descargamos el contenedor de joomla

docker pull joomla

```
Digest:
sha256:2930b6a641c639f231c0ec5f39b5191ee7885781074a672e230543b388b4492e
Status: Downloaded newer image for joomla:latest
```

- Descargamos ahora la de MySQL.
- NOTA IMPORTANTE.** En la versión 8 de MYSQL han cambiado el tratamiento de las password y aplicaciones que no están preparadas no funcionan. Por tanto, es necesario descargarse la versión 5.7 o anterior. También se puede usar MariaDB que es la versión open source de MYSQL

docker pull mysql:5.7

```
Digest:
sha256:691c55aabb3c4e3b89b953dd2f022f7ea845e5443954767d321d5f5fa394e28c
Status: Downloaded newer image for mysql:latest
```

- Vamos en primer lugar a arrancar el contenedor de MySQL y crear la Base de datos. Le tenemos que indicar un nombre y luego una variable de entorno para la password, en este caso he puesto “secret”, como en el ejemplo anterior. También es muy importante asociarlo a la red “net1”

```
docker run -d --name mysql1 -e MYSQL_ROOT_PASSWORD=secret --
network=net1 mysql
18759a903ff01e3ea16404ac444b0d6ea480cd4b3ff1d09c7e7f603973918a86
```

- Debemos comprobar que existe

docker ps

CONTAINER ID CREATED	IMAGE STATUS	COMMAND PORTS	NAMES
18759a903ff0 a minute ago	mysql Up About a minute	"docker-entrypoint.s..." 3306/tcp	About mysql1

- Podemos acceder con una bash para ver que tenemos funcionando la base de datos

```
docker exec -it mysql1 bash
```

- Nos conectamos a la base de datos (la password es la que hemos puesto al crear el contenedor, “secret” en mi caso). Usamos el comando “psql”. Si todo es correcto tenemos base de datos

docker exec -it mysql1 bash

```
root@18759a903ff0:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
```

```
mysql> show databases
-> ;
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0.00 sec)
```

- Nos quedamos dentro de la Base de Datos
- Desde otro terminal arrancamos el contenedor de Joomla. No hace falta poner --link
 - En este caso, a través de variables de entorno indicamos el contenedor mysql y su password.
 - Debemos poner también la red a la que pertenece, en este caso "net1"
 - También le indicamos que para acceder al Joomla vamos a usar el puerto 8080 en la máquina host

```
docker run -d --name joomla1 -e JOOMLA_DB_HOST=mysql1 -e
JOOMLA_DB_PASSWORD=secret --network=net1 -p 8080:80 joomla
Joomla not found in /var/www/html - copying now...
Complete! Joomla has been successfully copied to /var/www/html

Ensuring Joomla database is present

MySQL Database Created
=====

This server is now configured to run Joomla!
```

You will need the following database information to install Joomla:

Host Name: mysql1

Database Name: joomla

Database Username: root

Database Password: secret

=====

AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally to suppress this message

AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally to suppress this message

[Sat Mar 24 00:08:25.175142 2018] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.10 (Debian) PHP/7.1.15 configured -- resuming normal operations

[Sat Mar 24 00:08:25.175232 2018] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'

- Si volvemos a la bash que tenemos contra el contenedor mysql, vemos que ha creado una base de datos llamada Joomla

```
mysql> show databases;
```

```
+-----+
```

```
| Database |
```

```
+-----+
```

```
| information_schema |
```

```
| joomla |
```

```
| mysql |
```

```
| performance_schema |
```

```
| sys |
```

```
+-----+
```

```
5 rows in set (0.00 sec)
```

- Si ahora accedemos con el Navegador al puerto 8080 en el host principal, debe aparecer el proceso de instalación de Joomla.



The screenshot shows the Joomla! Web Installer interface in a web browser. The browser's address bar shows 'localhost:8080/installation/index.php'. The Joomla! logo is at the top, followed by the text 'Joomla! is free software released under the GNU General Public License.' Below this, there are three tabs: '1 Configuration' (active), '2 Database', and '3 Overview'. Under the 'Configuration' tab, there is a 'Select Language' dropdown menu set to 'English (United States)' and a 'Next' button. The 'Main Configuration' section contains two fields: 'Site Name *' with a text input field and a hint 'Enter the name of your Joomla! site.', and 'Super User Account Details' with an 'Email *' field and a text input field.

- Además y al contrario que cuando enlazamos con `-link`, podemos comprobar si abrimos una bash en cualquiera de los dos contenedores que podemos acceder al otro.