

Repaso - Docker Compose

Docker Compose es un orquestador de servicios.

Para la definición de los servicios se usa un fichero con extensión **.yml** y se llama `docker-compose.yml`.

YAML es un lenguaje basado en **tabulaciones** y que permite definir valores en formato clave valor y listas de valores para una clave dada.

Para el caso de un fichero `docker-compose.yml`, encontraremos los siguientes valores principales:

version: define la versión del orquestados a usar

services: servicios a definir para que nuestro entorno de trabajo funcione, como por ejemplo: `web`, `mysql`, `redis`, etc... Es el nombre que nosotros les queremos dar.

volumes: **volúmenes docker** para asociarlos a los volúmenes de los contenedores de los servicios

network: declaración de redes para asociarlas a los contenedores de los servicios

Dentro de cada servicio, se define la configuración individual del mismo:

version: "2"

services:

db:

image: ...

volumes:

- ...:...

restart: ...

environment:

...=...

...=...

...=...

...=...

Las configuraciones de cada servicio vienen condicionadas por cada atributo que éstas pueden tener:

- **image**: imagen a partir de la cual se construye el contenedor del servicio
- **build**: imagen construida a partir de un Dockerfile
- **volumes**: volúmenes (docker o binding map)
- **restart**: cuando debe reiniciarse
- **environment**: variables de entorno que se deben crear al crear el contenedor
- **network**: red a la que estará asociado el contenedor del servicio
- **depends_on**: si depende de otro servicio para levantarse
- **tty**: mantener el contenedor activo
- **ports**: mapeo de puertos
- **command**: comando a ejecutar al lanzar el contenedor
- **network**: red a la que se le asocia

De esta forma, dentro de un fichero docker-compose.yml se pueden definir multitudes de servicios interconectados entre sí.

Vamos a realizar un ejercicio sobre docker-compose entregable y evaluable para nota.

Servicio web con bases de datos.

Crea un fichero docker-compose.yml con la siguiente orquestación de servicios:

Versión del fichero: 2

Crea dos servicios: web y mysql

Para crear el servicio **web**, tenemos la siguiente información:

- Parte de última versión de la imagen de “httpd”
- Siempre se va a iniciar de forma automática
- Crearemos una lista de dos volúmenes:
 - el propio directorio sobre el que trabajamos se mapeará dentro del contenedor como /app

- el fichero de la carpeta **provisioning/etc/apache/sites-enabled/vhost.conf** se copiará dentro del contenedor en la carpeta **/etc/apache/sites-enabled/**
- el fichero de la carpeta **provisioning/var/www/html/index.html** se copiará dentro del contenedor en la carpeta **/var/www/html/**
- El servicio depende para iniciarse del servicio **mysql**
- El puerto de acceso externo al servicio web es: 8006

Para crear el servicio **mysql**, tenemos la siguiente información.

- Parte de la última versión de la imagen “mariadb”
- Siempre se va a iniciar de forma automática
- Crearemos una lista de un volumen:
 - Se debe **crear un volumen docker** y asociarlo a este servicio. El volumen se llamará dbdata y se asociará al contenedor **/var/lib/mysql** de dentro del contenedor
- Se configurarán las siguientes variables de entorno:
 - MYSQL_ROOT_PASSWORD: dbrootpass
 - MYSQL_DATABASE: mydb
 - MYSQL_USER: root
 - MYSQL_PASSWORD: dbpass
- El puerto por defecto sobre el que trabaja mariadb es el 3306, igual que mysql, a parte de esto, nosotros queremos que el puerto de acceso externo sea 33006.

Deberemos crear la **red** general para la orquestación: mired_shared y deberemos incluirla dentro de cada uno de los servicios.

Ejecutad la construcción de los contenedores.