Sistemas de gestión empresarial

Dockers. Casos prácticos con contenedores

1. Balanceo de carga de servidores web

En esta práctica se pretende visializar el funcionamiento de un proxy que realiza el balanceo entre varios servidores web conectados a él.

Para ésto se van a utilizar las imágenes de Apache **httpd:2.4** y del balanceador de carga de HAProxy **haproxy**.

Además, en el fichero **practica4_balanceo.zip** se proporcionan los ficheros *index.html* que servirán los servidores apache, más la configuración necesaria del proxy en el fichero *haproxy.cfg*

Descomprime este archivo en un directorio que sea tu área de trabajo. Las rutas para el montaje de volúmenes en el momento de crear los contenedores serán relativas a este directorio.

Paso 1: Crear la red

Crea un red llamada **ha_network** a la que se van a unir los tres contenedores de la práctica.

Paso 2: Servidores Apache

En este primer paso hay que crear los contenedores de Apache.

Las configuraciones necesarias para cada uno son:

- Se conectan a la red *ha_network*
- Los nombres son **apache01** y **apache02**
- Asocian el puerto **80** de cada uno a los del anfitrión **8088** (apache01) y **8089** (apache02)
- Se ejecutan en segundo plano
- Montan la carpeta **./apache01** y **./apache02** al directorio donde apache sirve los archivos **/usr/local/apache2/htdocs**
- La imagen es **httpd:2.4**

Una vez que cada contenedor está funcionando, verifica que responde abriendo un navegador en las direcciones **http://localhost:8088** y **http://localhost:8089**

En caso de que aparezca el mensaje "Forbidden", hay que revisar los permisos de los directorios *apache0x* y el de los ficheros *index.html*

Paso 3: Balanceador de carga

Una vez que los servidores web están en funcionamiento correctamente, habrá que crear el contenedor del balanceador de carga. Las configuraciones de este contenedor son:

- Se conecta a la red *ha_network*
- El nombre es **hapx**
- El puerto **80** en el que responde se asocia al **8080** en la máquina anfitrión.
- Hay que montar la carpeta **./haproxy** al directorio donde se aloja la configuración de este proxy **/usr/local/etc/haproxy**. El archivo *haproxy.cfg* está preparado para esta configuración, no hace falta editarlo.
- Se asignan valores a variables de configuración (opción -e):
  - **APACHE_1_IP** con valor *apache01*
  - **APACHE_2_IP** con valor *apache02*
  - **APACHE_EXPOSED_PORT** con valor *80*
- Se ejecuta en segundo plano

Paso 4: Comprobación del funcionamiento

Para verificar el correcto funcionamiento del balanceador, hay que abrir un navegador con la dirección **http://localhost:8080**.

Al ir actualizando el navegador, debe ir sirviendo las páginas de ambos servidores de forma alternativa. Ésto está configurado en el fichero *haproxy.cfg* en la línea *balance roundrobin*.

2. Despliegue de una aplicación en el servidor Tomcat

En este caso práctico se va a desplegar una aplicación de ejemplo en un servidor Tomcat.

Paso 1. Arrancar el contenedor Tomcat

La imagen a utilizar va a ser la oficial desarrollada por Apache que se encuentra en Dockerhub. Su nombre es el del servidor **tomcat**.

Esta imagen tiene configurado la publicación del puerto 8080 como puerto de servicio del servidor de aplicaciones. Por tanto, hay que asignarlo a otro puerto en el equipo anfitrión. Como en el caso anterior hemos utilizado el 8080 para el proxy, le asignamos el 8090. De nombre al contenedor le ponemos **tom**

```
docker run -it -d -p 8090:8080 --name tom tomcat
```

Paso 2. Despliegue de la aplicación java

La aplicación a desplegar se encuentra en *https://tomcat.apache.org/tomcat-6.0-doc/appdev/sample/*. También se incluye en los archivos de prácticas del aula virtual.

Mirando la documentación de la imagen *tomcat*, "How to use this image" vemos que el directorio de instalación es */usr/local/tomcat* y el asignado para los despliegues es */usr/local/tomcat/webapps*.

El despliegue consiste finalmente en copiar el archivo sample.war en el directorio *webapps*, configurado para las aplicaciones.

```
docker cp sample.war tom:/usr/local/tomcat/webapps
```

Para comprobar su funcionamiento escribimos en el navegador **http://localhost:8090/sample/**