

Sistemas de gestión empresarial

Dockers. Prácticas iniciales con contenedores

En esta práctica se pretende que el alumno se familiarice con las órdenes básicas de creación, ejecución y gestión de contenedores.

1. Lanzar Ubuntu accediendo a un terminal

Creamos un contenedor del sistema operativo **ubuntu** con la siguiente orden

```
docker run -it --name miubuntu1 ubuntu /bin/bash
```

Parámetros:

- **-i**: el proceso se lanza en modo interactivo, la entrada estándar se asigna a un proceso en un terminal.
- **-t**: asigna al proceso lanzado, al arrancar el contenedor, un terminal.
- **--name**: nos permite establecer un nombre a nuestro contenedor.

Por último, el comando ejecutado al lanzarse el contenedor es */bin/bash*.

Vamos a probar varias órdenes para ver qué ocurre al parar y volver a arrancar el contenedor

```
alumno@linuxserver2324:~/dockers$ docker run -it --name miubuntu1 ubuntu /bin/bash
```

```
root@c1572a757ed4:/# pwd
/
root@c1572a757ed4:/# whoami
root
root@c1572a757ed4:/# cd home
root@c1572a757ed4:/home# ls
ubuntu
root@c1572a757ed4:/home# cd ubuntu
root@c1572a757ed4:/home/ubuntu# mkdir practical
root@c1572a757ed4:/home/ubuntu# touch practical/mifile1.txt
root@c1572a757ed4:/home/ubuntu# echo "Hola, mundo!" >> practical/mifile1.txt
root@c1572a757ed4:/home/ubuntu# exit
exit
```

Con **exit** salimos del terminal y paramos la ejecución del contenedor.

Comprobamos que al volver a arrancarlo, la carpeta y el archivo creado permanecen

```
alumno@linuxserver2324:~/dockers$ docker start -ai miubuntu1
```

```
root@c1572a757ed4:/# cd /home/ubuntu
root@c1572a757ed4:/home/ubuntu# ls
practical
root@c1572a757ed4:/home/ubuntu# ls practical/
mifile1.txt
root@c1572a757ed4:/home/ubuntu# cat practical/mifile1.txt
Hola, mundo!
...
exit
```

Como el contenedor ya está creado, se usa **docker start**. Para poder interactuar con el terminal se usan las opciones **-a** (attach) y **-i** (interactivo).

Por último, podemos arrancar el contenedor y abrir una consola en el mismo, o en otro terminal, con **docker attach**

```
alumno@linuxserver2324:~/dockers$ docker start miubuntu1
miubuntu1
alumno@linuxserver2324:~/dockers$ docker attach miubuntu1
root@c1572a757ed4:/#
```

2. Lanzar un servidor web en background asociando puertos

En este apartado de ejemplo, vamos a arrancar un contenedor de una imagen de **nginx** al que vamos a poner nuestra propia página *index.html*

Se crea el contenedor con el comando

```
docker run -d -p 1200:80 nginx
```

- La imagen utilizada es **nginx:latest**
- **-d** lanza el contenedor en segundo plano. Con esta opción, no se muestra ninguna información de la entrada/salida del contenedor.
- **-p** al utilizar como valor *pAnfitrión:pContenedor* (1200:80), el puerto de la máquina anfitriona se asocia al del contenedor.
 - Otra opción es usar un valor tipo *-p 80* donde el anfitrión recoge ese puerto del contenedor y le asocia uno aleatorio del anfitrión. Se puede ver el puerto asignado con *docker ps*.

El mapeo de puertos solo puede realizarse en el momento de crear el contenedor. No se puede modificar el mapeo de puertos con el contenedor ya creado.

Podemos ver lo que está pasando en el servidor web mirando sus logs

```
docker ps (nos permite ver el nombre asignado o el identificador del contenedor)
```

```
docker logs -f a3a66
```

(salimos con Ctrl+C)

Cambiar la página index.html

En el servidor web *nginx* las páginas que se sirven están en el directorio **/usr/share/nginx/html**.

Para modificar nuestra página *index.html*, vamos a utilizar dos opciones: una, editarla en el contenedor y dos, copiarla desde nuestro anfitrión.

Editar en el contenedor

Necesitamos ejecutar un shell en el contenedor para poder abrir un editor y así modificar la página html. En este contenedor no hay editores instalados por lo que previamente hay que instalar *nano*.

```
docker exec -it festive_brattain /bin/bash
(festive_brattain es el nombre asignado al contenedor en el ejemplo)
```

```
# Una vez dentro del contenedor
cd /usr/share/nginx/html
:/usr/share/nginx/html# ls
50x.html  index.html
:/usr/share/nginx/html# apt update
:/usr/share/nginx/html# apt install nano
:/usr/share/nginx/html# cp index.html index.html.bak
:/usr/share/nginx/html# nano index.html
```

(Añade tu nombre a la página `index.html` por ejemplo)

Y verifica en el navegador que la página ha cambiado

Copiar una página ya creada en el directorio `html`

Para copiar un archivo desde el anfitrión al contenedor se usa el comando **`docker cp`**.

```
docker cp index2.html festive_brattain:/usr/share/nginx/html
```

En este caso el primer parámetro (origen) es la ruta a nuestra página `html` y el segundo parámetro se forma con el nombre del contenedor:ruta de destino.

Para probar que ha funcionado hay que poner en el navegador **`http://localhost:1200/index2.html`**

3. Crear un contenedor de XAMPP

A partir de la imagen del servidor de XAMPP alojada en **`https://hub.docker.com/r/davidsal/xampp-8.0.0`**, repite el ejercicio práctico del punto 2 con este servidor.