

DOCKER

en el taller Iris

Nota: Trabajaremos desde la consola (Ctrl+Alt+t)

Gestión del “daemon” Docker:

Ver el estado. Inicialmente estará parado (punto negro):
systemctl status docker

Arrancarlo

systemctl start docker
Authenticating as: alumno,,, (alumno)
Password: "alumno"

Ver el estado de nuevo, estará en marcha (punto verde):
systemctl status docker

Ver contenedores:

docker ps
docker ps -a
docker ps -a -q

Ver imágenes:

docker image ls -a
docker images -a

Borrar contenedores (por limpieza):

docker rm -f f8306b233fe1
docker rm -f \$(docker ps -a -q)

Borrar imágenes (por limpieza):

docker rmi nombrelimagen
docker rmi \$(docker images -a -q)

Usando Imágenes y contenedores de **linode-lamp**

Buscar imágenes:

docker search linode

Descargar imágenes:

docker pull linode/lamp

Arrancar (previamente hay que listar o ver las imágenes disponibles para saber el nombre)

Nota: la opción “-p” es para indicar los puertos por los que se comunican nuestro ordenador o anfitrión con el contenedor. Las opciones “-t” y “-i” (también “-ti” o “-it”) nos permiten entrar dentro del contenedor y ver que es una máquina Linux. En muchos ejemplos veréis “-d”, es lo “contrario” de “-it”, es decir, lo ejecuta en segundo plano.

docker run -p 80:80 -t -i linode/lamp /bin/bash
Cambia el prompt -> root@c0bb49a663c9:/#
Esto indica que estamos dentro del contenedor

Dentro del contenedor podemos arrancar (start) y parar (stop) los servidores apache2 y mysql:

service apache2 start
service mysql start
service mysql stop
service apache2 stop

Salir del contenedor:

exit (Ctrl + d) → Cierra el contenedor.
Ctrl+p y luego Ctrl+q → Se sale y se puede volver con
docker exec -ti 228e438e7fe3 bash

Ejemplo práctico 1 con **linode-lamp**

Se parte de la imagen descargada y con sus contenedores parados o borrados.

***Arrancamos un contenedor:**

```
docker run -p 80:80 -t -i linode/lamp /bin/bash
```

***Dentro (nuevo prompt). Arrancamos Apache**

```
service apache2 start
```

Se puede probar que funciona desde el navegador → "http://localhost"

Arrancamos MySQL (seguimos dentro del contenedor)

```
service mysql start
```

```
mysql -u root -p
```

```
password → "Admin2015"
```

```
SELECT * FROM mysql.user;
```

```
SELECT User FROM mysql.user;
```

```
exit;
```

Nota: para profundiza y crear un usuario, una BD, etc, ver en la carpeta "ayuda/mysql/"

***Dos formas de trabajar con ficheros:**

1. Copiar ficheros desde el host al contenedor (comando "cp"):

```
docker cp index.php fbd9b4a38486:/var/www/example.com/public_html/index.php
```

***2.- Arrancar el contenedor con volúmenes ("-v").** Similar a una carpeta compartida.

```
docker run -v $(pwd)/public_html:/var/www/example.com/public_html -p 80:80 -t -i linode/lamp /bin/bash
```

Nota: Antes de arrancar el contenedor debo estar una carpeta de trabajo y haber creado la carpeta local del volumen. En el ejemplo es "public_html", que estará montada (compartida) con la ruta del contenedor (/var/www/example.com/public_html) donde están los ficheros PHP, HTML, etc., que usa el servidor Web Apache.

"\$(pwd)" equivale a poner el directorio actual

Ejemplo práctico 2 con **linode-lamp**

- 1.- Crea las carpetas necesarias para el proyecto y el para el volumen.
- 2.- Arranca el contenedor con un volumen, como en el ejemplo.
- 3.- Dentro del contenedor: arranca Apache.
- 4.- En tu ordenador crea los ficheros necesarios para tu Web en la carpeta del volumen.