

# COMANDOS DOCKER

## **docker pull nombre\_imagen:version**

Este comando sirve para descargar la imagen desde el repositorio con la versión indicada o la última versión si no la indicamos.

## **docker run nombre\_imagen**

Este comando sirve para ejecutar contenedores.

### **-d o --detach**

Para ejecutar un contenedor (normalmente porque tenga un servicio) en background.

### **-e o --env**

Para establecer variables de entorno en la ejecución del contenedor.

### **-h o --hostname**

Para establecer el nombre de red para el contenedor.

### **--help**

Para obtener ayuda de las opciones de docker.

### **--interactive o -i**

Para mantener la STDIN abierta en el contenedor.

### **--ip**

Si quiero darle una ip concreta al contenedor.

### **--name**

Para darle nombre al contenedor.

### **--net o --network**

Para conectar el contenedor a una red determinada.

### **-p o --publish**

Para conectar puertos del contenedor con los de nuestro host.

### **--restart**

Que permite reiniciar un contenedor si este se "cae" por cualquier motivo.

### **--rm**

Que destruye el contenedor al pararlo.

### **--tty o -t**

Para que el contenedor que vamos a ejecutar nos permita un acceso a un terminal para poder ejecutar órdenes en él.

### **--user o -u**

Para establecer el usuario con el que vamos a ejecutar el contenedor.

**--volume o -v**

Para montar un bind mount o un volumen en nuestro contenedor.

**--workdir o -w**

Para establecer el directorio de trabajo en un contenedor.

**-it**

Sirve para abrir la entrada estándar del contenedor que estamos ejecutando y permitir la posibilidad de abrir un terminal en el contenedor.

**-d**

Para que el servicio se ejecute en modo background o dettach. Si no lo hacemos se bloqueará el terminal mostrando el log del servicio.

**-p PUERTO\_EN\_HOST:PUERTO\_EN\_CONTENEDOR**

Si queremos que el servicio que vamos a lanzar sea accesible desde el exterior.

**-e NOMBRE\_VARIABLE=VALOR**

Comprobar y definir si es necesario las variables de entorno que puede tener el contenedor.

**--name**

De esta manera, si usamos nombre elegidos por nosotros serán más fáciles de recordar que los asignados por defecto.

**docker exec**

Ejecutar ordenes en los contenedores en ejecución.

**OPCIONES DEL DOCKER EXEC****-it**

(-i y -t juntos) si vamos a querer tener interactividad con el contenedor ejecutando un shell (/bin/bash normalmente). Una vez tenemos el terminal ya podremos trabajar desde dentro del propio sistema.

**-u o --user**

Si quiero ejecutar la orden como si fuera un usuario distinto del de root.

**-w o --workdir**

Si quiero ejecutar la orden desde un directorio concreto.

**docker cp**

Me permite mover ficheros desde mi sistema al contenedor y desde el contenedor a mi sistema.

**docker ps**

Mostrar los contenedores que están en ejecución.

**OPCIONES DEL DOCKER PS****docker ps -a**

Mostrar todos los contenedores, estén parados o en ejecución (-a o --all).

**docker ps -a -s**

Añadir la información del tamaño del contenedor a la información por defecto (-s o --size).

**docker ps -l**

Mostrar información del último contenedor que se ha creado (-l o --latest). Da igual el estado.

**docker ps --filter name=servidor\_web**

Filtrar los contenedores de acuerdo a algún criterio usando la opción (-f o --filter).

**docker ps --filter publish=8080**

Filtrado por puerto. Contenedores que hacen público el puerto 8080.

**docker inspect (nombreImagen o ID)**

Nos va a dar una información detallada del contenedor que seleccione.

**--format/-f**

Una descripción detallada.

**docker logs (nombreImagen o ID)**

Nos dan información de lo que está pasando en el contenedor.

**-f**

Sigue escuchando la salida que pueden dar los logs del contenedor.

**--tail 5**

Muestra las 5 últimas líneas de los logs del contenedor en cuestión.

**docker stop**

Para detener el contenedor, ya sea por nombre o por ID.

**stop (-t o --time)**

Para detener el contenedor con tiempo.

**docker rm**

Borrar un contenedor.

**docker rm (--force o -f)**

Borrar un contenedor aunque esté en ejecución.

**docker start**

Iniciar un contenedor que estaba parado previamente, ya sea por nombre o por ID.

**docker start (-i o --interactive)**

Inicio de un contenedor pero haciendo el attach de la entrada estándar para poder interactuar con él.

**docker restart**

Para reiniciar un contenedor que previamente ya estaba en ejecución.