



Jose Antonio Castillejo Lobato

Comando	Descripción
<b>docker pull nombre_imagen:version</b>	Descarga la imagen
<b>docker run ↓ contenedor</b>	

- -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.

docker ps	Mostrar los contenedores en ejecución
docker ps -a	Mostrar todos los contenedores creados ya estén en ejecución o parados
docker run -it ubuntu:18.04 /bin/bash	Crear un contenedor de ubuntu:18.04 y tener acceso a un shell en él.

<code>docker start servidorWeb</code>	Arrancar el contenedor con nombre servidorWeb
<code>docker start 511eed5992d</code>	Arrancar el contenedor con ID 511eed5992d
<code>docker run httpd</code>	Ejecuto un servidor Apache sin el flag -d ni redirección de puertos. Se bloquea el terminal mostrando los logs y tendré que salir con Ctrl+C
<code>docker run -d -p 8888:80 httpd</code>	Ejecuto un servidor Apache en background y accediendo desde el exterior a través del puerto 8888 de mi máquina.
<code>docker run -it -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root mariadb</code>	Creación de un servidor de base de datos mariadb accediendo desde el exterior a través del puerto 3306 y estableciendo una contraseña de root mediante una variable de entorno
<code>docker run -d --name servidorBD -p 3306:3306 mysql:8.0.22</code>	Damos el nombre de servidorBD a un contenedor de la imagen mysql:8.0.22
<code>docker run -d --name servidorWeb -p 80:80 httpd</code>	Damos el nombre de servidorWeb a un contenedor de la imagen httpd:latest (Apache)
<code>docker exec [opciones] nombre_contenedor orden [argumentos]</code>	
<p>Algunas de las opciones más importantes son:</p> <ul style="list-style-type: none"> <li>• -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.</li> <li>• -u o --user si quiero ejecutar la orden como si fuera un usuario distinto del de root.</li> <li>• -w o --workdir si quiero ejecutar la orden desde un directorio concreto.</li> </ul>	
<code>docker exec -it web /bin/bash</code> <code>root@5d96ce1f7374:/usr/local/</code>	Obtener un terminal en un contenedor que ejecutar un

apache2#	servidor Apache (httpd) y que se llama web
docker exec web ls /usr/local/apache2/htdocs	Mostrar el contenido de la carpeta /usr/local/apache2/htdocs del contenedor web. Como no hace falta interactividad no es necesario -it
docker exec -it web sh -c "echo 'HOLA MUNDO' > /usr/local/apache2/htdocs/index.html"	Crear directamente un fichero "HOLA MUNDO" en el directorio raíz del servidor apache. Utilizo sh -c para ordenes compuestas o complejas
docker cp prueba.html web:/usr/local/apache2/htdocs/index.html	Copiar mi fichero prueba.html al fichero /usr/local/apache2/htdocs/index.html de mi contenedor llamado web que es un servidor Apache (httpd)
docker cp web:/usr/local/apache2/htdocs/index.html \$HOME/test.html	Copiar el fichero index.html que se encuentra en /usr/local/apache2/htdocs/index.html de mi contenedor llamado web un fichero llamado test.html en mi directorio HOME
docker ps -a -s	Añadir la información del tamaño del contenedor a la información por defecto
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
docker ps --filter publish=8080	Filtrado por puerto. Contenedores que hacen público el puerto 8080
docker inspect jenkins	Mostrar información detallada del

	contenedor cuyo nombre es jenkins
docker inspect 5e5adf6815bc	Mostrar información detallada por id
docker inspect --format 'La ip es {{.NetworkSettings.Networks.bridge.IPAddress}}' jenkins	Mostrar la ip del contenedor
docker inspect --format 'Las redirecciones de puertos son {{.NetworkSettings.Ports}}' jenkins	Mostrar las redirecciones de puertos del contenedor
docker logs jenkins	Mostrar los logs del contenedor cuyo nombre es jenkins
docker logs -f jenkins	Sigue escuchando la salida que pueden dar los logs del contenedor
docker logs --tail 5 jenkins	Muestra las 5 últimas líneas de los logs del contenedor en cuestión
docker stop servidorWeb	Para un contenedor en ejecución que se llame servidorWeb
docker stop -t 10 ea9b922190d8	Para un contenedor en ejecución cuyo ID es ea9b922190d8 pero esperando 10 segundo
docker rm servidorBD	Borrar un contenedor que se llama servidorBD
docker rm -f jenkins	Borrado un contenedor que se llame jenkins aunque esté en ejecución
docker start jenkins	Inicio de un contenedor con nombre jenkins
docker start -i jenkins	Inicio de un contenedor con nombre jenkins pero haciendo el attach de la entrada estándar para poder interactuar con él

<code>docker restart ea9b922190d8</code>	Reinicio de un contenedor con ID ea9b922190d8
<code>docker restart -t 10 ea9b922190d8</code>	Reinicio de un contenedor con ID ea9b922190d8 pero esperando 10 segundo