

Docker

Comandos	Explicaciones
docker pull nombre_imagen:version	Descarga desde el repositorio una imagen con la versión indicada o la última versión (latest) si no indicamos versión.
docker run [opción] image [comando][arg] [opciones] <ul style="list-style-type: none">• -d o --detach para ejecutar un contenedor 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	Para ejecutar contenedores si no tenemos la imagen, ésta se descargará de manera automática.

<p>ejecutar el contenedor.</p> <ul style="list-style-type: none"> • --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 : (--interactive) + -t (--tty) abre la entrada estándar del contenedor que estamos ejecutando y permite la posibilidad de abrir un terminal en el contenedor 	
docker run -d	Ejecutar en modo background o dettach. Si no lo hacemos se bloqueará el terminal mostrando el log del servicio (en ciertas ocasiones puede interesarnos) y tendremos que salir del mismo con Ctrl+C. Esto para el contenedor aunque podremos arrancarlo posteriormente.
docker run -p PUERTO_EN_HOST:PUERTO_EN_CONTENEDOR	Si queremos que sea accesible desde el exterior. Esto es una REDIRECCIÓN DE PUERTOS
docker run -e NOMBRE_VARIABLE=VALOR	# 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 > docker run -it -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root mariadb
<p>docker exec [opciones] nombre_contenedor orden [argumentos]</p> <p>[opciones]</p> <ul style="list-style-type: none"> • -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. 	<p>Ejecutar órdenes en el contenedor. es NECESARIO QUE EL CONTENEDOR ESTÉ EN EJECUCIÓN.</p> <p># Mostrar el contenido de la carpeta /usr/local/apache2/htdocs del contenedor web. Como no hace falta interactividad no es necesario -it > docker exec web ls /usr/local/apache2/htdocs</p>

<ul style="list-style-type: none"> • -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. 	
<p>docker cp</p> <p># Copiar mi fichero prueba.html al fichero /usr/local/apache2/htdocs/index.html de mi contenedor llamado web que es un servidor Apache (httpd)</p> <pre>> docker cp prueba.html web:/usr/local/apache2/htdocs/index.html</pre>	<p>Permite mover ficheros desde mi sistema al contenedor y desde el contenedor a mi sistema.</p>
<p>docker ps</p> <p># Mostrar los contenedores que están en ejecución</p> <pre>> docker ps</pre> <p># Mostrar todos los contenedores, estén parados o en ejecución (-a o --all)</p> <pre>> docker ps -a</pre> <p># Añadir la información del tamaño del contenedor a la información por defecto (-s o --size)</p> <pre>> docker ps -a -s</pre> <p># Mostrar información del último contenedor que se ha creado (-l o --latest). Da igual el estado</p> <pre>> docker ps -l</pre> <p># Filtrado por nombre</p> <pre>> docker ps --filter name=servidor_web</pre> <p># Filtrado por puerto. Contenedores que hacen público el puerto 8080</p> <pre>> docker ps --filter publish=8080</pre>	<p>Obtener información de los contenedores ya arrancados.</p>
<p>docker inspect</p> <p># Por nombre. Por ejemplo: Mostrar información detallada del contenedor cuyo nombre es jenkins</p> <pre>> docker inspect jenkins</pre>	<p>Información detallada del contenedor</p>

<p>#Por id. Por ejemplo: Mostrar información detallada del contenedor cuyo id es 5e5adf6815bc</p> <pre>> docker inspect 5e5adf6815bc</pre>	
<p>docker logs</p> <p># Por nombre. Por ejemplo: Mostrar los logs del contenedor cuyo nombre es jenkins</p> <pre>> docker logs jenkins</pre> <p># Por id. Por ejemplo: Mostrar los logs cuyo id es 5e5adf6815bc</p> <pre>> docker logs 5e5adf6815bc</pre> <p>Opción -f o --follow . Sigue escuchando la salida que pueden dar los logs del contenedor</p> <pre>> docker logs -f jenkins</pre> <p># Opción --tail 5. Muestra las 5 últimas líneas de los logs del contenedor en cuestión</p> <pre>> docker logs --tail 5 jenkins</pre>	<p>Información de lo que está pasando en el contenedor. Sirve tanto para contenedores que estén parados como para contenedores en ejecución.</p>
<p>docker start</p>	<p>Iniciar un contenedor que estaba parado previamente, ya sea por nombre o por ID.</p>
<p>docker restart</p>	<p>Para reiniciar un contenedor que previamente ya estaba en ejecución.</p>
<p>docker stop</p> <p># Para un contenedor en ejecución cuyo ID es ea9b922190d8 pero esperando 10 segundo (-t o --time)</p> <pre>> docker stop -t 10 ea9b922190d8</pre>	<p>Para detener el contenedor, ya sea por nombre o por ID.</p>
<p>docker rm</p> <p># Borrado un contenedor que se llame</p>	<p>Para borrar el contenedor, ya sea por nombre o por ID.</p>

jenkins aunque esté en ejecución (--force o -f) > docker rm -f jenkins	
docker ps	Mostrar los contenedores en ejecución
docker ps -a	Mostrar todos los contenedores creados ya estén en ejecución (Estado Up) o parados (Estado Exited)
docker start nombre_contenedor docker start ID_contenedor	Arrancar el contenedor
docker ps	Mostrar los contenedores en ejecución
docker images head	Muestra las primeras líneas de las imágenes descargadas.