

# Glosario de Docker

**Para iniciar un contenedor debemos iniciarlo con Docker run. Las opciones más habituales son:**

- `-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.

**Para el uso de Docker, necesitaremos el manejo de ciertos comandos como lo son los siguientes:**

<b>docker stop ID</b>	Para un contenedor
<b>docker start ID</b>	Si queremos reiniciar-recuperar el contenedor que hemos paralizado
<b>docker run imagen</b>	Para arrancar una imagen (si no existe, la descarga también)
<b>docker run 'imagen:versión'</b>	Para arrancar una imagen con una versión concreta
<b>docker run -d imagen</b>	Para arrancar una imagen "background"
<b>docker rm ID</b>	Borra un contenedor
<b>docker rmi &lt;idimagen&gt;</b>	Borra la imagen seleccionada
<b>docker pull imagen</b>	Descarga la imagen pero no la arranca (por defecto la más reciente)
<b>docker ps -a</b>	Muestra las últimas imágenes utilizadas
<b>docker ps</b>	Muestra las imágenes que están arrancadas, activas

<b>docker log -f ID</b>	Muestra las salidas de la imagen
<b>docker images   head</b>	Muestra las primeras lineas de las imágenes descargadas.
<b>docker images</b>	Muestra las imágenes que tenemos descargadas
<b>docker exec ID</b>	Ejecuta un comando dentro de un contenedor que está arrancado
<b>docker exec -it ID sh</b>	( <b>-i</b> crea una sesión interactiva, <b>-t</b> emula una terminal)
<b>CTRL + C</b>	Frena, para una imagen arrancada

## Esta es una serie de comandos para el uso de Docker los cuáles son muy útiles:

<b>docker builder</b>	- Manage builds
<b>docker config</b>	- Manage Docker configs
<b>docker container</b>	- Manage containers
<b>docker context</b>	- Manage contexts
<b>docker image</b>	-Manage images
<b>docker network</b>	- Manage networks
<b>docker node</b>	- Manage Swarm nodes
<b>docker plugin</b>	- Manage plugins
<b>docker secret</b>	- Manage Docker secrets
<b>docker service</b>	- Manage services
<b>docker stack</b>	- Manage Docker stacks
<b>docker swarm</b>	- Manage Swarm
<b>docker system</b>	- Manage Docker
<b>docker trust</b>	- Manage trust on Docker images
<b>docker volume</b>	- Manage volumes

## Argumentos para Docker exec:

**docker exec [opciones] nombre\_contenedor orden  
[argumentos]**

### OPCIONES

<b>-it</b>	(-i y -t juntos) si vamos a querer tener interactividad con el contenedor ejecutando un shell (/bin/bash normamente). Una vez tenemos el terminal
<b>-u</b> <b>--user</b>	Si quiero ejecutar la orden como si fuera un usuario distinto del de root.
<b>-w</b> <b>--workdir</b>	Si quiero ejecutar la orden desde un directorio concreto