

# DOCKER

[devops](#), [info](#)

## related

- [docker \(altran\)](#) [devops](#), [cursos](#), [docker](#)
- [Docker: SecDevOps](#) [devops](#), [docker](#), [swarm](#), [libros](#), [tech](#)
- [Native Docker clustering with Swarm](#) [libros](#), [tech](#), [docker](#), [swarm](#)
- [MasterClass Docker](#) [docker](#)
- [docker \(first contact\)](#) [docker](#)
- [Taller pentesting en Docker](#) [docker](#)
- [docker swarm: official guide](#) [devops](#), [docker](#), [swarm](#)

## +info

- [dockerd](#)
- [Dockerfile](#)

## casos de uso

- [docker-compose](#)
- [docker run](#)

## documentación

- [docker-cookbook.pdf](#)

## dockerhub ♥

- [nextcloud](https://hub.docker.com/_/nextcloud/) : [https://hub.docker.com/\\_/nextcloud/](https://hub.docker.com/_/nextcloud/)
- [vsftpd](https://dockerfile.readthedocs.io/en/latest/content/DockerImages/dockerfiles/vsftp.html#environment-variables) : <https://dockerfile.readthedocs.io/en/latest/content/DockerImages/dockerfiles/vsftp.html#environment-variables>
- [rtorrent + rutorrent](https://hub.docker.com/r/diameter/rtorrent-rutorrent/) : <https://hub.docker.com/r/diameter/rtorrent-rutorrent/>
- [mariadb](https://hub.docker.com/_/mariadb/) : [https://hub.docker.com/\\_/mariadb/](https://hub.docker.com/_/mariadb/)
- [arm](https://hub.docker.com/u/arm32v7/) : <https://hub.docker.com/u/arm32v7/>
  - <https://medium.freecodecamp.org/the-easy-way-to-set-up-docker-on-a-raspberry-pi-7d24ced073ef>
- [autossh](https://www.everythingcli.org/ssh-tunnelling-for-fun-and-profit-autossh/) : <https://www.everythingcli.org/ssh-tunnelling-for-fun-and-profit-autossh/>
  - <https://github.com/dreamhost/docker-autossh/blob/master/Dockerfile>
- VPN
  - <https://hub.docker.com/r/hwds12/ipsec-vpn-server/>
  - [https://hub.docker.com/r/kylemanna/opencvn/](https://hub.docker.com/r/kylemanna/opencvpn/)
    - <https://www.digitalocean.com/community/tutorials/how-to-run-openssh-in-a-docker-container-on-ubuntu-14-04>
  - [https://www.reddit.com/r/docker/comments/7e0gid/docker\\_with\\_pia\\_vpn\\_for\\_transmission\\_and/](https://www.reddit.com/r/docker/comments/7e0gid/docker_with_pia_vpn_for_transmission_and/)

# resumen comandos

## instalación

```
sudo usermod -aG docker $(whoami)
```

## informativos

- `docker info`
- `docker ps`
  - `-a`: all
  - `-q`: only ID
  - `-f <filter>`: `docker -ps -af «status=exited»`
- `docker search`
- `docker inspect <id>`: vuela JSON con información
  - `docker inspect <id> | grep IPAddress`

## contenedores

- estados de los contenedores
  - **CREATED**
    - → start → **RUNNING**
    - → pause → **PAUSED** → unpause **RUNNING**
    - → stop → **STOPPED** → restart → **RUNNING**
    - → kill → **KILLED**
- `docker ps`
  - `-a`
  - `-q`
- `docker run <imagen> [<comando>]`
  - `-i`: interactive
  - `-t`: tty
  - `--rm`: contenedor de un solo uso
  - `-d`: detach (el proceso no se queda «colgado» ejecutando el contenedor)... lanzar contenedor en background
  - `-p 80:80`: mapea el puerto host:contenedor, en todos los interfaces
  - `-p 127.0.0.1:80:80`: el contenedor solo será accesible desde 127.0.0.1
  - `-P`: mapea el puerto en el que está escuchando el contenedor a un puerto aleatorio del equipo
  - `--name <nombre_contenedor>`
  - `docker run -it <container_id>`: al salir del contenedor pasa a **STOPPED**
- `docker start <contenedor>`: arrancar contenedor parado
- `docker pause <contenedor>`
- `docker unpause <contenedor>`
- `docker stop <contenedor>`
- `docker kill <contenedor>`
- `docker exec [-it] <container> [<comando>]`
  - `bash` o `sh` serían comandos válidos si están instalados en el contenedor
- `docker rm <container_id>`
  - no se puede eliminar containers en ejecución
  - `docker rm $(docker ps -aq)`: elimina todos los contenedores
- `docker commit <id_contenedor> [REPOSITORY[:TAG]]`: crea imagen de un contenedor

## imágenes

- `docker pull debian[:<tag>]`
- `docker images`
  - `-fo-filter`
    - «`dangling=true`»: filtra lista imágenes no tageadas
    - «`label=<clave>`»: filtra por labels (a nivel de imagen, se ven siempre)
    - «`label=<clave>=<valor>`»: filtra por el contenido de las claves
  - `-format «{{.ID}}:{{.Repository}}»`: formato de salida (escrito en Go, plantilla)
- `docker rmi <imagen_id>`: borrar una imagen
- `docker tag <imagen_id> <nuevo_nombre>` asignar un nombre a una imagen sin tagear, copiar si ya estaba tageada
- `docker save -o <destino> <imagen>:<tag>` guarda en <destino> una copia «física» de la imagen
  - NO recomendado!
- `docker load -i <imagen_disco>`: importa la imagen
- `docker history <imagen>`: muestra las capas de una imagen y el tamaño de cada una

```
# mostrar (y eliminar) imágenes huérfanas <none>:<none> sin usar
docker image list -q --filter "dangling=true"
docker rmi $(docker image list -q --filter "dangling=true")
docker image rm $(docker image list -q --filter "dangling=true") ??
```

From:

<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:

<https://miguelangel.torresegea.es/wiki/tech:docker:start?rev=1581629484>

Last update: **13/02/2020 13:31**

