

# Chuleta Comandos Docker

NOTA: algunos de estos comandos requieren ser root o usar sudo

**1) Crear un contenedor bajando una imagen (OJO, si se hace varias veces, crea varios contenedores independientes)** y que se pueda acceder de forma interactiva -it (acceder a la shell)

```
"docker run ubuntu:latest -it"
```

**2) Obtener ids de los contenedores existentes**

```
"docker ps -a"
```

**3) Obtener ids de los contenedores en ejecución**

```
"docker ps"
```

**4) Lanzar un contenedor ya creado. Obtener id del contenedor (docker ps -a) y con el id ejecutar**

```
"docker container start 09d6ea310f75 -i"
```

Donde -i es para acceder en modo interactivo (acceder a la shell) y 09d6ea310f75 es la id.

**5) Acceder a la consola de un contenedor que está en funcionamiento donde 665b4a1e17b6 es la id del contenedor.**

```
"docker exec -i -t 665b4a1e17b6 /bin/bash"
```

**6) Parar un contenedor Docker.**

```
"docker stop 665b4a1e17b6"
```

**7) Listado de imágenes docker**

```
"docker images"
```

**8) Borrar contenedor docker**

```
"docker rm 665b4a1e17b6"
```

Borrar todos los contenedores en estado Exit

```
"docker rm $(sudo docker ps -a | grep Exit | cut -d ' ' -f 1)"
```

Borrar todos los contenedores

```
"docker rm $(sudo docker ps -a -q)"
```

**9) Obtener imagen docker sin crear contenedor, solo la imagen**

```
"docker pull ubuntu:15.04"
```

## 10) Borrar imagen docker

```
"docker rmi 665b4a1e17b6"
```

Borrar todas las imagenes docker

```
"docker rmi $(docker images -a -q)"
```

## 11) Exportar imagen docker a fichero

```
docker save <imagen> > <archivo>.tar
```

## 12) Cargar imagen docker desde fichero

```
docker load < <archivo>.tar
```

**13)** Docker commit, para hacer commit de un contenedor a una imagen. Importante la de destino tenga nuestro nombre propio de la cuenta "docker hub" para que podamos subirla con docker push.

Ejemplo: creo una máquina con etiqueta "sergarb1/bda-ceedcv-oracle-xe-11g" a partir del contenedor "c3f279d17e0a"

```
"sudo docker commit c3f279d17e0a sergarb1/bda-ceedcv-oracle-xe-11g"
```

**14)** Docker push, para subir una imagen a Docker hub. Previamente hay que logearse con el comando

```
"docker login"
```

Aquí un ejemplo de docker push

```
"docker push sergarb1/bda-ceedcv-oracle-xe-11g"
```

donde "sergarb1/bda-ceedcv-oracle-xe-11g" es una cuenta y una imagen nuestra dentro de docker hub

## 15) Crear enlaces entre dockers. Más info <https://docs.docker.com/network/links/>

Por ejemplo, creamos un contenedor docker con nombre "db" de una base de datos Postgres

```
"docker run -d --name db training/postgres"
```

donde "--name db" indica que se creara el contenedor de la imagen "training/postgres" con el nombre "db"

Entonces si quieres crear por ejemplo un contenedor web que esté enlazado a esa base de datos pones:

```
"docker run --name web --link db training/webapp"
```

Que se enlaza con "db" y refiriendose al host "db" accedes al otro contenedor.

**16)** Copiar elementos de la máquina real a un contenedor y viceversa

"sudo docker cp /nuestraPagina.zip cordova:/tmp/"

donde "cordova" es el nombre o identificador del contenedor.

**17)** Mapear una carpeta de la máquina real a una carpeta de la máquina docker

Creamos una máquina con un volumen mapeado a nuestra máquina

docker run -i -t -v /home/usuario/MiProyecto:/proyecto ubuntu:16.04 /bin/bash

Con esta orden: bajas imagen ubuntu (etiqueta 16.04) y creas un contenedor (Ojo si lo haces varias veces creas varios contenedores, para ir luego a un contenedor recuerda usar comandos anteriores).

También en ese momento mapea tu carpeta "/home/usuario/MiProyecto" a una carpeta "/proyecto" (si no existe la crea) en la máquina Docker.

#### **Webs interesantes:**

<https://tech.mybuilder.com/virtual-hosts-with-docker/>

<https://www.katacoda.com/courses/docker>