

Comandos Docker

\$ docker run [options] IMAGE [comandos] [argumentos] ejecutar un contenedor

- **-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 parar 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.

Flags:

- **-it** para que nos permita interactuar con el contenedor
- **-d** para que el servicio se ejecute en modo background
- **-p PUERTO_EN_HOST:PUERTO_EN_CONTENEDOR** para que el servicio sea accesible desde el exterior
- **-e NOMBRE_VARIABLE=VALOR** para definir variables de entorno
- **--name** para asignar un nombre al contenedor

Ejemplos de todo lo anterior:

```
docker run -it -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root mariadb
```

```
docker run -d --name servidorWeb -p 80:80 httpd
```

\$ docker exec: docker exec [opciones] nombre_contenedor orden [argumentos]

Sirve para realizar varias operaciones con los contenedores que estén en ejecución

- **-it** para interactuar con el contenedor.
- **-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.

Ejemplo:

```
docker exec -it web /bin/bash
```

docker cp mover ficheros desde mi sistema al contenedor y viceversa

Ejemplo:

```
docker cp prueba.html web:/usr/local/apache2/htdocs/index.html
```

\$ docker ps para mostrar información de los contenedores

- **docker ps -a** para mostrar todos los contenedores.
- **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
- **docker ps --filter name=servidor_web** filtrar por nombre
- **docker ps --filter publish=8080** filtrar por puerto

\$ docker --version ver la versión de docker

\$ docker pull IMAGE descargar una imagen

\$ docker inspect para obtener información detallada

\$ docker start para iniciar un contenedor

\$ docker stop para parar un contenedor

\$ docker rm para borrar un contenedor

\$ docker restart para reiniciar un contenedor

\$ docker kill para detener un contenedor inmediatamente