## Resumen de comandos Docker

#systemctl status/start/stop/restart/stop docker SERVICIOS

# /var/lib/docker FICHEROS

#docker version versión

#docker help ayuda de comandos docker

#docker network –help comandos del network.

#docker run --name nombreContenedor hello-world crear un contendor

#docker run –d nginx crear un contenedor en segundo plano

#docker run –name -it Ubuntu /bin/bash crear un contendor interactivo

#docker rm $(docker ps -aq) –f forzar borrar todos los contenedores

#docker ps muestra los contenedores

#docker ps -a muestra los contenedores que están ejecutandose

#docker run –p 81:80 httpd:latest crear un contenedor al que se le asigna el puerto 80 y el 81 en el host

#netstat –Intu (o nte-tools) para ver los puertos abiertos

#docker inspect obtener información

#docker inspect ID|grep –i tcp para ver puertos abiertos en el contenedor

#docker exec –it nombreContenedor /bin/bash acceder a un contenedor

#docker attach nombreContenedor

#docker start/stop/restart/ iniciar, parar, resetear

# cd /var/lib/docker/containers ver el tamaño de los contenedores

#docker stats estadísticas

#docker ps –s tamaño

#docker network create red1 crear una red

# docker network create --help

#docker network create –subnet 192.168.3.0/24 red2

#docker network ls muestra las redes

#docker network inspect bridge inspeccionar una red

#ip a información ip

#nmcli con conexiones

#docker run -it --name mi-debian --network red1 debian Asociar a un contenedor una red en la creación

#docker network connect red2 mi-debian Asociar a un contenedor una red en CALIENTE

#docker run -it --name mi-busybox-3 --link mi-busybox-1:busy-1 busybox Crear un enlace

#/vae/lib/docker/volumes/ID/\_data VOLUMENES

#docker volumen ls mostrar volúmenes

#docker volume inspect ID Más información del volumen

#docker volume rm ID Borrar un volumen

#docker volume prune Borrar todos los volúmenes que no se usen

#docker run -it -v /datos --name mi-ubuntu ubuntu bash crear un contenedor con un volumen en la carpeta /datos

#docker volume create volumen-1 Crear un volumen independiente

#docker run -it --name mi-ubuntu -v volumen-1:/datos ubuntu bash Asociamos el volumen a un contenedor

#docker run -it --name mi-ubuntu-4 -v volumen-1:/datos:ro ubuntu bash Crear un volumen en modo solo lectura

#docker diff mi-debian Para ver los cambios en un contenedor

#docker commit mi-debian debian-manolo Creamos la imagen de forma manual

#docker images mostrar imágenes

#docker build -t mi-imagen:v1 . Construimos la imagen a partir del Dockerfile

#docker image history mi-imagen:v1 Comprobamos las capas de esa imagen

#docker run -it mi-imagen:v1 bash Construimos un contenedor con esa imagen

#docker build -t mftienda/cowsay:v1 . Creamos la imagen con el nombre de la cuenta de github (mftienda)

#docker image tag Nombre-Imagen:tag mftienda/Nombre-Imagen:tag Si ya tenemos la imagen creada, podemos cambiar el tag

#docker login

#docker push mftienda/cowsay:v1 subir imagen a github

# docker pull mftienda/cowsay:v1 bajar imagen a github

#curl -L "https://github.com/docker/compose/releases/download/1.27.4/dockercompose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose Instalar docker compose

#chmod +x /usr/local/bin/docker-compose le damos permiso al archivo

#docker-compose --version Comprobamos que está instalado

# vi docker-compose.yml Creamos el fichero

#docker-compose up -d Segundo plano

#docker compose ENTER Vemos todos los comandos

#docker-compose ps Vemos los procesos

#docker-compose stop pararlo

#docker-compose down eliminarlo