

### imagenes

<https://hub.docker.com>

Descargar la imagen

```
docker pull nombre_imagen:tag
```

Crear un Dockerfile

```
nano Dockerfile
```

Creación imagen

punto (.) para llamar al Dockerfile

```
docker build -t
```

```
nombre_imagen_result .
```

Historial

```
docker history -H
```

```
nombre_imagen:tag
```

Eliminar imagen

```
docker rm -fv nombre_imagen:tag
```

Buscar imagenes

```
docker images | grep
```

```
nombre_imagen:tag
```

Estructura Dockerfile

```
FROM imagen:tag
```

```
LABEL descripcion
```

```
RUN comando_install -y
```

```
WORKDIR dir_trabajo
```

```
USER usuario_ejecuta_tarea
```

```
COPY carpeta_archivo ruta/...
```

```
ADD archivo_url ruta/...
```

```
ENV variable contenido
```

```
VOLUME ruta
```

```
EXPOSE puerto
```

```
CMD comando_ejecutar
```

Estructura dockerignore

```
archivo_ignorar
```

Eliminar imagenes

```
docker rmi imagen:tag
```

Eliminar imagenes dangling

```
docker images -f dangling=true -q
| xargs docker rmi
```

### Volúmenes

Volumen persistente (datos bd)

```
docker run -d parametros -v
/ruta_loc:/ruta_cont imagen
```

Volumen anónimos (datos bd)

```
docker run -d parametros -v
/ruta_loc imagen
```

Volumen docker

```
docker volume ls
```

Eliminar volumen docker (v)

```
docker rm -fv imagen
```

Volumen nombrado

```
docker volume create nombre
```

Asignar volumen

```
docker run -d -v nombre:/ruta
imagen
```

Dangling volumes

```
docker volume ls -f dangling=true
```

### Registry

Creacion

```
docker run -d -p 5000:5000 --name
registry -v
$PWD/data:/var/lib/registry
registry:2
```

Subir imagenes

```
docker tag nombre_imagen:tag
localhost:5000/nombre_imagen:tag
```

Subir imagen al registry

```
docker push
localhost:5000/nombre_imagen:tag
```

Bajar imagen al registry

```
docker pull
localhost:5000/nombre_imagen:tag
```

### Registry (cont)

Compartir imagenes desde una red

```
vi
/lib/systemd/system/docker.service
ExecStart=/usr/bin/dockerd --
insecure-registry IP:5000
sudo systemctl daemon-reload
docker stop registry
sudo systemctl restart docker
docker start registry
docker pull
localhost:5000/nombre_imagen:tag
```

### Contenedores

Listar un contenedor

```
docker ps
docker ps -a
docker ps -q
docker ps -l
```

Crear un contenedor

```
docker run --help|less
docker run -d imagen (-
d) segundo_plano
docker run -d -p ploc:pcont imagen
(-p) puerto
```

Borrar un contenedor

```
docker rm -f nombre
```

Renombrar un contenedor

```
docker rename nombre_actual
nombre_nuevo
```

Detener un contenedor

```
docker stop nombre/id
```

Iniciar un contenedor

```
docker start nombre/id
```

Reiniciar un contenedor

```
docker restart nombre/id
```

Shell un contenedor

```
docker exec -u root -ti nombre/id
bash
```

Eliminar todos los contenedores

```
docker ps -q | xargs docker rm -f
```

### Contenedores (cont)

*VARIABLES DE ENTORNO*

```
docker run -e "prueba1=1234"
imagen
```

*INSPECCIONAR CONTENEDOR*

```
docker inspect imagen
```

*LOGS EN CONTENEDOR*

```
docker logs -f imagen
```

*RECURSOS EN CONTENEDOR*

```
docker stats imagen
```

*LIMITAR RECURSOS EN CONTENEDOR*

```
docker run -d -m "500mb/gb "
imagen
```

*CUANTAS CPU SE TIENEN*

```
grep "model name" /proc/cpuinfo |
wc -l
```

*SABER CPU EN CONTENEDOR*

```
docker run --help | grep cpu
```

*RESTRINGIR CPU EN CONTENEDOR*

```
docker run -d -m "1gb" --cpuset-
cpus 0-1 imagen
```

*ENVIAR ARCHIVOS EN CONTENEDOR*

```
docker cp archivo imagen:/ruta
```

*EXTRAER ARCHIVOS EN CONTENEDOR*

```
docker cp imagen:/ruta
/ruta_local
```

*DOCKER COMMIT*

```
docker commit contenedor
imagen_resultante:tag
```

*SOBREESCRIBIR EL CMD*

```
docker run -dti imagen:tag
cmd_argumento
```

*DESTRUIR CONTENEDORES AUTO EJECUCIÓN TEMPORAL (--rm)*

```
docker run --rm -ti imagen:tag
```

*DOCUMENT ROOT*

```
var/lib/docker
```

### Docker Network

*CONSULTAR LA RED DOCKER*

```
docker network ls
Red por defecto
docker network ls | grep bridge
```

*RED DOCKER DEFINIDA*

```
docker network create nombre_red
(--help)
```

*INSPECCIONAR LA RED*

```
docker network inspect nombre_red
```

*RED*

```
docker run --network
nombre_red_a_conectar -d --name
test imagen
```

*PING*

```
docker exec contenedor bash -c
"ping IP_cont/nombre_cont"
```

*CONECTAR DIFERENTES REDES*

```
docker network connect nombre_red1
nomb_cont2
```

*DESCONECTAR DIFERENTES REDES*

```
docker network disconnect
nombre_red1 nomb_cont2
```

*ELIMINAR REDES*

```
docker network rm nombre_red
```

*ASIGNAR IP A CONTENEDOR*

```
docker network create --subnet
IP/24 --gateway IP -d bridge
nombre_red
```

*ASIGNAR IP A CONTENEDOR*

```
docker run --network nombre_red -
d --name cont1 -ti centos
```

*ASIGNAR IP A CONTENEDOR*

```
docker run --network nombre_red -
-ip IP -d --name cont1 -ti imagen
```

*LA RED DE HOST*

```
docker run --network host -d --
name cont -ti imagen
```

*RED NONE*

```
docker run --network none -d --
name cont -ti imagen
```

### Compose

*EJECUCIÓN*

```
docker-compose
```

*EJECUTAR CONTENEDOR*

```
docker-compose up -d
docker-compose -f archivo.yml up -
d
```

*ELIMINAR CONTENEDOR*

```
docker-compose down
```

*IMÁGENES COMPOSE*

```
docker-compose build
```

*PERSONALIZAR NOMBRE PROYECTO*

```
compose
docker-compose -p nombre -f
nombre.yml up -d
```

*NOMBRE DISTINTO DOCKER-COMPOSE.YML*

```
docker-compose -f
nombre_archivo.yml
```

*LOGS COMPOSE*

```
docker-compose logs -f
```

