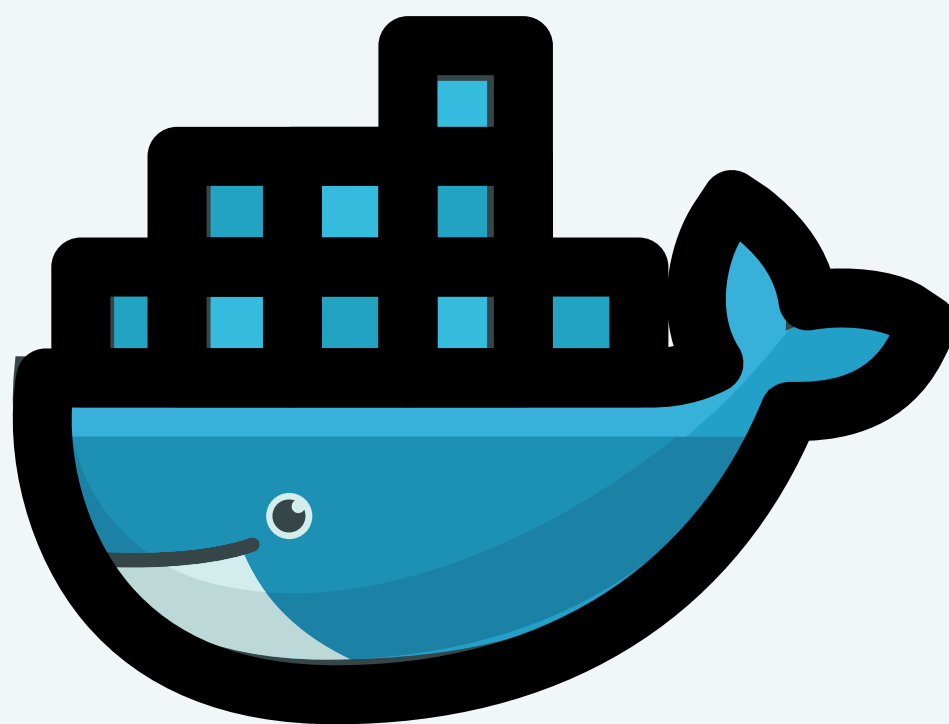




Guía Completa Comandos Docker



PABLO DEL ÁLAMO



1

Comandos Docker Básicos

- **docker run [imagen]:** Ejecuta un contenedor a partir de una imagen. Si la imagen no está presente localmente, Docker la descargará. Ejemplo: `docker run ubuntu`
- **docker ps:** Lista los contenedores en ejecución. Opciones: `docker ps -a` (muestra todos los contenedores, incluidos los detenidos).
- **docker stop [ID o nombre del contenedor]:** Detiene un contenedor en ejecución. Ejemplo: `docker stop my-container`





- **docker start [ID o nombre del contenedor]:**
Inicia un contenedor detenido previamente.
Ejemplo: docker start my-container
- **docker restart [ID o nombre del contenedor]:**
Reinicia un contenedor en ejecución o detenido.
Ejemplo: docker restart my-container
- **docker rm [ID o nombre del contenedor]:**
Elimina un contenedor detenido. Opciones:
docker rm -f [ID] (fuerza la eliminación de un contenedor en ejecución).
- **docker rmi [ID o nombre de la imagen]:** Elimina una imagen de Docker local. Ejemplo: docker rmi ubuntu
- **docker pull [imagen]:** Descarga una imagen desde Docker Hub o un registro de imágenes. Ejemplo: docker pull nginx





- **docker images:** Lista todas las imágenes locales.
- **docker exec -it [ID o nombre del contenedor] [comando]:** Ejecuta un comando dentro de un contenedor en ejecución. Ejemplo: docker exec -it my-container bash
- **docker logs [ID o nombre del contenedor]:** Muestra los logs de un contenedor. Opciones: docker logs -f [ID] (sigue los logs en tiempo real).
- **docker inspect [ID o nombre del contenedor]:** Devuelve información detallada de un contenedor o imagen, en formato JSON. Ejemplo: docker inspect my-container
- **docker top [ID o nombre del contenedor]:** Muestra los procesos que están corriendo dentro de un contenedor.



PABLO DEL ÁLAMO



2 Comandos de Imágenes Docker

- **docker build -t [nombre_imagen] [directorio]:**
Construye una imagen Docker a partir de un Dockerfile. Ejemplo: docker build -t mi-aplicacion .
- **docker tag [imagen_original] [nuevo_nombre]:**
Etiqueta una imagen existente con un nuevo nombre. Ejemplo: docker tag mi-aplicacion:latest mi-aplicacion:v1
- **docker history [ID o nombre de la imagen]:**
Muestra el historial de capas de una imagen. Ejemplo: docker history nginx





- **docker save -o [archivo.tar] [imagen]:** Guarda una imagen Docker en un archivo .tar. Ejemplo: `docker save -o mi-imagen.tar mi-aplicacion`
- **docker load -i [archivo.tar]:** Carga una imagen Docker desde un archivo .tar. Ejemplo: `docker load -i mi-imagen.tar`
- **docker push [nombre_imagen]:** Sube una imagen a Docker Hub o a un registro privado. Ejemplo: `docker push usuario/mi-aplicacion`



3

Comandos de Contenedores

- **docker rename [ID o nombre del contenedor] [nuevo_nombre]:** Renombra un contenedor. Ejemplo: docker rename my-container mi-nuevo-contenedor
- **docker pause [ID o nombre del contenedor]:** Pausa todos los procesos dentro de un contenedor. Ejemplo: docker pause my-container
- **docker unpause [ID o nombre del contenedor]:** Reanuda los procesos de un contenedor que fue pausado. Ejemplo: docker unpause my-container





- **docker attach [ID o nombre del contenedor]:**
Adjunta tu terminal al flujo de salida de un contenedor. Ejemplo: docker attach my-container
- **docker cp [origen] [destino]:** Copia archivos entre tu máquina local y un contenedor. Ejemplo: docker cp mi-archivo.txt my-container:/app/
- **docker stats [ID o nombre del contenedor]:**
Muestra el uso de recursos (CPU, memoria, etc.) de un contenedor en tiempo real.



PABLO DEL ÁLAMO



4

Comandos de Redes Docker

- **docker network ls:** Lista todas las redes de Docker. Ejemplo: `docker network ls`
- **docker network create [nombre_red]:** Crea una nueva red de Docker. Ejemplo: `docker network create mi-red`
- **docker network rm [nombre_red]:** Elimina una red de Docker. Ejemplo: `docker network rm mi-red`





- **docker network inspect [nombre_red]:** Muestra información detallada sobre una red de Docker.
- **docker network connect [nombre_red] [ID o nombre del contenedor]:** Conecta un contenedor a una red existente. Ejemplo: `docker network connect mi-red my-container`
- **docker network disconnect [nombre_red] [ID o nombre del contenedor]:** Desconecta un contenedor de una red. Ejemplo: `docker network disconnect mi-red my-container`



PABLO DEL ÁLAMO



5 Comandos de Volúmenes Docker

- **docker volume ls:** Lista todos los volúmenes de Docker. Ejemplo: `docker volume ls`
- **docker volume create [nombre_volumen]:** Crea un nuevo volumen de Docker. Ejemplo: `docker volume create mi-volumen`
- **docker volume rm [nombre_volumen]:** Elimina un volumen de Docker. Ejemplo: `docker volume rm mi-volumen`
- **docker volume inspect [nombre_volumen]:** Muestra detalles de un volumen de Docker.
- **docker run -v [nombre_volumen] [ruta_en_contenedor] [imagen]:** Monta un volumen en un contenedor al iniciarlo. Ejemplo: `docker run -v mi-volumen:/app/data nginx`





6

Comandos Avanzados y Misceláneos

- **docker system df:** Muestra el uso del disco por imágenes, contenedores, volúmenes y redes.
- **docker system prune:** Elimina todos los recursos no utilizados, incluidos contenedores detenidos, imágenes sin etiquetar, y redes no usadas.
Ejemplo: `docker system prune -a` (elimina todo lo que no esté en uso).
- **docker update --restart=always [ID o nombre del contenedor]:** Cambia la política de reinicio de un contenedor para que se reinicie automáticamente.



PABLO DEL ÁLAMO



- **docker events:** Muestra eventos en tiempo real relacionados con los objetos Docker en el sistema.
- **docker commit [ID o nombre del contenedor] [nombre_nueva_imagen]:** Crea una nueva imagen Docker a partir de los cambios realizados en un contenedor. Ejemplo: `docker commit my-container mi-nueva-imagen`
- **docker export [ID o nombre del contenedor] -o [archivo.tar]:** Exporta el sistema de archivos de un contenedor a un archivo .tar.
- **docker import [archivo.tar]:** Importa un archivo .tar como una imagen Docker.



PABLO DEL ÁLAMO



7 Comandos de Configuración y Optimización

- **docker build --no-cache -t [nombre_imagen] [directorio]:** Construye una imagen sin utilizar la caché de capas, asegurando una construcción limpia.
- **docker build --build-arg [argumento]=[valor] -t [nombre_imagen] [directorio]:** Construye una imagen con argumentos de construcción personalizados. Ejemplo: `docker build --build-arg ENV=production -t mi-app .`





- **docker run --env [clave]=[valor] [imagen]:**
Ejecuta un contenedor con variables de entorno.
Ejemplo: `docker run --env APP_ENV=production nginx`
- **docker run --name [nombre] [imagen]:** Ejecuta un contenedor y le asigna un nombre específico.
Ejemplo: `docker run --name mi-contenedor nginx`
- **docker run --link [contenedor] [imagen]:**
Conecta dos contenedores para que puedan comunicarse entre ellos. Ejemplo: `docker run --link db-container:db web-app`



PABLO DEL ÁLAMO



¿Te ha resultado útil?



- Comparte esta guía con tu equipo o amigos desarrolladores.
- Guárdala para tener siempre a mano los comandos.
- ¡Dale un like o comenta si tienes preguntas!

