

Guía Práctica de Comandos de Docker

1. Parar y eliminar contenedores e imágenes

Parar todos los contenedores

```
docker stop $(docker ps -a -q)
```

Este comando detiene todos los contenedores en ejecución. Utiliza `docker ps -a -q` para listar los IDs de todos los contenedores y pasa esa lista como entrada a `docker stop`.

Eliminar todos los contenedores

```
docker rm -f $(docker ps -a -q)
```

Elimina todos los contenedores existentes, forzando su eliminación si están en ejecución (`-f`).

Eliminar todas las imágenes

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

Elimina todas las imágenes de Docker, forzándolo a hacerlo con `-f`. Utiliza `docker images -a -q` para listar los IDs de todas las imágenes.

2. Trabajar con imágenes

Listar imágenes

```
docker images
```

Muestra una lista de todas las imágenes locales, con detalles como su ID, repositorio, etiqueta y tamaño.

Buscar imágenes en Docker Hub

```
docker search nginx
```

Busca imágenes relacionadas con "nginx" en Docker Hub.

Filtrar resultados de búsqueda

```
docker search -f is-official=true nginx
```

Muestra solo las imágenes oficiales de "nginx".

```
docker search --filter=stars=20 nginx
```

Filtra imágenes con al menos 20 estrellas.

Descargar imágenes

```
docker pull nginx
```

Descarga la última versión de la imagen **nginx** desde Docker Hub.

```
docker pull mysql:5.7
```

Descarga una versión específica (5.7) de la imagen **mysql**.

Inspeccionar imágenes

```
docker inspect nginx
```

Proporciona detalles técnicos de la imagen **nginx** en formato JSON.

Historial de una imagen

```
docker history nginx
```

Muestra las capas de la imagen **nginx**, incluyendo los comandos utilizados para construirla.

Eliminar imágenes

```
docker rmi nginx
```

Elimina la imagen **nginx** del sistema local.

3. Lanzar contenedores

Comando `docker run`

```
docker run -dti --name web1 httpd
```

Crea y ejecuta un contenedor en segundo plano (`-d`), interactivo (`-ti`), con el nombre `web1` y basado en la imagen `httpd`.

Publicar puertos

```
docker run -dti -p 8080:80 --name web2 nginx
```

Asocia el puerto 8080 del host al puerto 80 del contenedor.

Variables de entorno

```
docker run -dti --name bd -e MYSQL_ROOT_PASSWORD=123456 mysql
```

Configura una variable de entorno para el contenedor.

Reinicio automático

```
docker run -dti --name web3 --restart always nginx
```

Configura el contenedor para que se reinicie automáticamente.

Eliminar al detener

```
docker run -dti --rm --name web4 httpd
```

El contenedor se elimina automáticamente cuando se detiene.

4. Inspección e interacción con contenedores

Listar contenedores

```
docker ps -a
```

Muestra todos los contenedores, incluidos los detenidos.

Inspeccionar contenedores

```
docker inspect web1
```

Proporciona información detallada sobre el contenedor `web1`.

Ejecutar comandos en contenedores

```
docker exec -ti web1 /bin/bash
```

Inicia una sesión interactiva en el contenedor `web1`.

Ver registros de contenedores

```
docker logs web1
```

Muestra los registros del contenedor `web1`.

```
docker logs -f web1
```

Sigue en tiempo real los registros del contenedor `web1`.

Modificar configuración de contenedores

```
docker update -m 512m --memory-swap -1 web1
```

Actualiza el límite de memoria del contenedor `web1` a 512 MB.

5. Estadísticas y gestión de recursos

Ver estadísticas

```
docker stats
```

Muestra las estadísticas de uso de recursos de los contenedores activos.

Configuración de recursos

```
docker run -dti --name web5 -m 256m --memory-swap -1 nginx
```

Limita el uso de memoria del contenedor `web5` a 256 MB.

6. Redes y puertos

Redirección de puertos

```
docker run -dti -p 8080:80 --name web nginx
```

Asocia el puerto 8080 del host al puerto 80 del contenedor.

Asignación automática de puertos

```
docker run -dtiP --name web httpd
```

Docker elige puertos disponibles del host y los asigna al contenedor.

Ver puertos asignados

```
docker port web
```

Muestra los puertos expuestos y asignados del contenedor `web`.

7. Comandos adicionales

Pausar y reanudar contenedores

```
docker pause web1
```

Pausa todos los procesos en el contenedor `web1`.

```
docker unpause web1
```

Reanuda los procesos pausados en el contenedor `web1`.

Reiniciar contenedores

```
docker restart web1
```

Detiene y reinicia el contenedor **web1**.