

Práctica: Jugando con Deployments y Comandos Básicos de Docker

En esta práctica, aprenderemos a utilizar algunos de los comandos fundamentales de Docker para crear, administrar y detener contenedores. Exploraremos cómo funcionan los contenedores, cómo se ejecutan en segundo plano y cómo interactuar con ellos.

Objetivos de la Práctica

- Comprender cómo funcionan los contenedores en Docker.
- Practicar con comandos Docker para iniciar, detener y eliminar contenedores.
- Explorar cómo persisten los cambios en un contenedor.

Paso 1: Listar los Contenedores Activos

1. Para ver los contenedores que se están ejecutando actualmente, usa el comando:

```
docker ps
```

Explicación: `docker ps` muestra una lista de contenedores activos. Esta lista incluye información como el ID del contenedor, la imagen utilizada y el tiempo que lleva en ejecución.

Paso 2: Ejecutar un Contenedor Interactivo

1. Ejecuta un contenedor interactivo con la imagen `alpine` (una imagen ligera de Linux):

```
docker run -it alpine sh
```

2. En el contenedor, ejecuta el siguiente comando para listar procesos:

```
ps -elf
```

Explicación: - `docker run -it alpine sh`: Este comando inicia un contenedor con la imagen `alpine` y nos da acceso a una terminal dentro del contenedor (`-it` nos permite interactuar con él). - `ps -elf`: Dentro del contenedor, este comando muestra los procesos que se están ejecutando en ese entorno aislado.

Paso 3: Verificar los Contenedores Activos

1. Abre otra terminal y ejecuta `docker ps` para verificar los contenedores en ejecución:

```
docker ps
```

Nota: Observa que el contenedor que iniciaste en modo interactivo sigue apareciendo en la lista de contenedores activos.

Paso 4: Ejecutar un Contenedor en Segundo Plano

1. Ejecuta un contenedor en segundo plano (modo "detached") que permanece activo durante un día:

```
docker run -d alpine sleep 1d
```

2. Ahora, verifica nuevamente los contenedores activos:

```
docker ps
```

Explicación: - `docker run -d alpine sleep 1d`: Este comando inicia un contenedor en segundo plano (`-d`) usando la imagen `alpine` y ejecutando el comando `sleep 1d` (dormir durante 1 día). - Al usar `docker ps`, puedes ver que ahora hay dos contenedores en ejecución.

Paso 5: Comprobar la Imagen Usada

1. Para comprobar que ambos contenedores están usando la misma imagen, ejecuta:

```
docker images
```

Explicación: Este comando muestra todas las imágenes descargadas en el sistema. Puedes ver que tanto el contenedor interactivo como el de segundo plano están utilizando la misma imagen `alpine`.

Paso 6: Detener un Contenedor

1. Detén el contenedor que está en segundo plano:

```
docker stop <container_id>
```

(Reemplaza `<container_id>` con el ID del contenedor que aparece en `docker ps`).

2. Verifica nuevamente los contenedores activos:

```
docker ps
```

3. Para ver también los contenedores detenidos:

```
docker ps -a
```

Explicación: - `docker stop <container_id>` : Este comando detiene el contenedor especificado. Puedes ver que ahora solo el contenedor interactivo sigue activo. - `docker ps -a` muestra todos los contenedores, incluidos los detenidos.

Paso 7: Reiniciar un Contenedor Detenido

1. Inicia nuevamente el contenedor detenido:

```
docker start <container_id>
```

2. Verifica que el contenedor está activo otra vez:

```
docker ps
```

Explicación: `docker start <container_id>` permite reiniciar un contenedor que había sido detenido. Ahora el contenedor está de nuevo en ejecución.

Paso 8: Ejecutar Comandos en un Contenedor en Ejecución

1. Accede al contenedor en ejecución y abre una terminal:

```
docker exec -it <container_id> sh
```

2. Dentro del contenedor, crea un archivo llamado `newFile` :

```
echo "hola" > newFile
```

3. Sal del contenedor (con `exit`).

Explicación: `docker exec -it <container_id> sh` permite ejecutar un comando interactivo dentro de un contenedor en ejecución. Aquí, hemos creado un archivo `newFile` con contenido de texto dentro del contenedor.

Paso 9: Verificar el Archivo Creado desde el Host

1. Desde la terminal, lista los archivos en el contenedor para verificar que `newFile` existe:

```
docker exec <container_id> ls -l
```

Explicación: Este comando permite ver el contenido del directorio en el contenedor sin necesidad de acceder a él directamente.

Paso 10: Eliminar los Contenedores

1. Detén y elimina todos los contenedores en una sola línea:

```
docker rm $(docker ps -aq) -f
```

Explicación: - `docker rm $(docker ps -aq) -f` detiene (`-f` fuerza la detención) y elimina todos los contenedores, tanto activos como detenidos. - Usa `Ctrl+P+Q` para salir de un contenedor sin detenerlo.

Esta práctica te ha permitido trabajar con comandos Docker para administrar y manipular contenedores.