Introducción a Docker y Virtualización

Ana Valladares González

Fecha: 1 nov 2024

Definición

Docker es una plataforma que permite desarrollar, desplegar y ejecutar aplicaciones en **contenedores**. Los contenedores se diferencian de las máquinas virtuales en que no emulan un sistema operativo completo, sino que se ejecutan de manera aislada compartiendo el núcleo del sistema del anfitrión.

Comandos para la Gestión de Imágenes

Descargar imágenes

Descargar una imagen desde un repositorio.

Este comando descarga la imagen dede un repositorio sin arrancarla. **docker pull** es útil cuando solo queremos tener la imagen lista sin ejecutarla. Si no se especifica **<tag>**, Docker descarga la última versión de la imagen.

```
docker pull <nombre_imagen>:<tag>
```

Listar imágenes locales

Listar todas las imágenes descargadas.

Muestra un listado de todas las imágenes almacenadas localmente, incluyendo detalles como el nombre, la etiqueta (**tag**) y el tamaño.

```
docker images
```

Comandos para la Gestión de Contenedores

Crear y ejecutar contenedores

Crear un contenedor sin arrancarlo.

Este comando prepara un contenedor basado en la imagen especificada, pero no lo inicia. Permite personalizar la configuración del contenedor antes de ejecutarlo.

```
docker create <nombre_imagen>
```

Iniciar y ejecutar un contenedor.

Este es el comando más utilizado para crear e iniciar un contenedor en una sola acción. Opciones importantes:

- **- d**: Ejecuta el contenedor en segundo plano.

- **- it**: Inicia el contenedor en modo interactivo, enlazando la entrada estándar (stdin) y permitiendo acceso a la terminal.
- **--name <nombre>**: Asigna un nombre al contenedor, facilitando su identificación.

```
docker run <opciones> <nombre_imagen> <comando>
```

```
docker run -it --name contenedor_ejemplo alpine /bin/sh
```

Este último comando, inicia un contenedor Alpine con acceso a la terminal interactiva (/bin/sh).

Listar y verificar contenedores

Listar contenedores.

Este comando es útil para ver el estado y la información de los contenedores activos y apagados.

```
docker ps # Lista solo contenedores activos
```

```
docker ps -a # Lista todos los contenedores, incluyendo los detenidos
```

Ver detalles de un contenedor.

Proporciona información detallada del contenedor, como configuraciones de red, volúmenes y uso de recursos.

```
docker inspect <nombre_o_id_contenedor>
```

Iniciar, detener y eliminar contenedores

Iniciar o detener un contenedor existente.

Start y **stop** permiten iniciar o detener un contenedor previamente creado.

```
docker start <nombre_o_id_contenedor>
```

```
docker stop <nombre_o_id_contenedor>
```

Eliminar un contenedor.

Este comando elimina un contenedor detenido.

```
docker rm <nombre o id contenedor>
```

Eliminar todos los contenedores detenidos.

```
docker container prune
```

Comandos para Redes y Conectividad

Inspección de Red y Direcciones IP

Obtener la IP de un contenedor.

Este comando extrae la dirección IP asignada a un contenedor, útil para verificar la conectividad.

```
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' dam_alp1
```

Verificar conectividad entre contenedores.

En contenedores en la misma red, puedes usar ping **<IP_destino>** para comprobar la comunicación entre ellos.

Comandos de Monitorización y Uso de Recursos

Monitoreo de uso de disco

Ver el espacio en disco utilizado.

Muestra el espacio utilizado por imágenes, contenedores y volúmenes.

```
docker system df
```

Monitoreo de CPU y RAM

docker stats ofrece una visión en tiempo real del consumo de recursos de cada contenedor en ejecución, incluyendo CPU, memoria y otros recursos.

```
docker stats
```

Configuración de Volúmenes y Bind Mounts

Creación de volúmenes y bind mounts

Configurar bind mounts para compartir directorios entre el host y el contenedor.

-v enlaza un directorio de tu máquina (host) con un directorio en el contenedor, permitiendo la sincronización entre ambos.

```
docker run -d -v <directorio_local>:<directorio_contenedor> <nombre_imagen>
```

```
docker run -d -v $(pwd)/mi_sitio:/usr/local/apache2/htdocs httpd:2.4
```

Uso de Docker Compose para Gestión de Aplicaciones de Múltiples Contenedores

Docker Compose permite definir y ejecutar aplicaciones multi-contenedor mediante un archivo de configuración YAML.

```
nano docker-compose.yml
```

Configuración del Archivo docker-compose.yml

Un archivo de configuración docker-compose.yml define servicios, redes y volúmenes para múltiples contenedores. Ejemplo:

```
version: '3'
services:
  web:
    image: nginx
```

```
    ports:
      - "8080:80"
  db:
    image: mysql
    environment:
      - MYSQL_ROOT_PASSWORD=password
```

Iniciar y detener servicios

Iniciar todos los servicios:

Ejecuta los servicios definidos en **docker-compose.yml**. La opción -d permite ejecutarlos en segundo plano.

```
docker-compose up -d
```

Detener y eliminar servicios:

down detiene y elimina los contenedores y redes creados por Docker Compose.

```
docker-compose down
```

Resumen Rápido de Comandos

Gestión de Imágenes

- Descargar una imagen: `docker pull <nombre_imagen>:<tag>`
- Listar imágenes locales: `docker images`
- Eliminar una imagen: `docker rmi <nombre_imagen>:<tag>`

Gestión de Contenedores

- Crear un contenedor sin iniciarlo: `docker create <nombre_imagen>`
- Iniciar y ejecutar un contenedor: `docker run <opciones> <nombre_imagen> <comando>`
- Listar contenedores: `docker ps`
- Ver detalles de un contenedor: `docker inspect <nombre_o_id_contenedor>`
- Iniciar un contenedor: `docker start <nombre_o_id_contenedor>`
- Detener un contenedor: `docker stop <nombre_o_id_contenedor>`
- Eliminar un contenedor: `docker rm <nombre_o_id_contenedor>`
- Eliminar todos los contenedores detenidos: `docker container prune`
- Ejecutar un comando dentro de un contenedor en ejecución: `docker exec -it <nombre_o_id_contenedor> <comando>`

Redes y Conectividad

- Obtener IP de un contenedor: `docker inspect -f '{{range.NetworkSettings.Networks}} {{.IPAddress}}{{end}}' <nombre_o_id_contenedor>`
- Hacer ping entre contenedores (misma red): `ping <IP_destino>`
- Conectar un contenedor a una red existente: `docker network connect <nombre_red> <nombre_o_id_contenedor>`
- Desconectar un contenedor de una red: `docker network disconnect <nombre_red> <nombre_o_id_contenedor>`

Monitorización y Uso de Recursos

- Ver espacio en disco utilizado: `docker system df`
- Monitorización en tiempo real de CPU y RAM: `docker stats`

Configuración de Volúmenes y Bind Mounts

- Configurar un bind mount: `docker run -d -v <directorio_local>:<directorio_contenedor> <nombre_imagen>`
- Listar volúmenes existentes: `docker volume ls`
- Eliminar un volumen: `docker volume rm <nombre_volumen>`

Docker Compose

- Iniciar servicios definidos en `docker-compose.yml`: `docker-compose up -d`
- Detener y eliminar servicios: `docker-compose down`