.:: COMANDOS-DOCKER-BASICOS ::.

Autor

Freddy Alcarazo | @alcarazolabs | @surflaweb

Descargar una imagen del Docker hub:

$ docker image pull fernando93d/hello

Ejecutar imagen:

$ docker container run fernando93d/hello

Listar contenedores

$ docker ps

Obtener información

$ docker ps ls --help

Ver contenedores que se han ejecutado o están ejecutándose:

$ docker container ls -a

Eliminar a uno o más contenedores

$ docker container rm <ID/nombre>

Crear a un contenedor

$ docker container create fernando93d/hello

Ahora el contenedor tendrá estado “creado” pero aún no se ha ejecutado.

Proceder a iniciar el contenedor:

$ docker container start <ID/name>

Modo Interactivo

- **Descargando imagen de Ubuntu del docker-hub:**

- `18.04`, `bionic-20220401`, `bionic`
- `20.04`, `focal-20220404`, `focal`, `latest`
- `21.10`, `impish-20220404`, `impish`, `rolling`
- `22.04`, `jammy-20220404`, `jammy`, `devel`
- `14.04`, `trusty-20191217`, `trusty`
- `16.04`, `xenial-20210804`, `xenial`

$ docker image pull ubuntu:focal

*Descargará la imagen de Ubuntu 20.04

Inspeccionar imágenes:

$ docker image ls

```
root@ubuntu-server:~# docker image ls
REPOSITORY          TAG       IMAGE ID       CREATED       SIZE
ubuntu              focal     825d55fb6340   5 days ago    72.8MB
fernando93d/hello   latest    7fe934464560   3 years ago   150MB
root@ubuntu-server:~#
```

Ejecutar imagen de Ubuntu:

$ docker container run ubuntu:focal

El contenedor se iniciará y se apagará automáticamente debido a que no tiene ningún proceso ejecutándose.

Probar con imagen de nginx:

$ docker image pull nginx

Ejecutar imagen:

$ docker container run nginx

Eliminar una image

$ docker rmi <ID/name>

Eliminar todas las imágenes

docker system prune -a --volumes

Modo Interactivo con terminal:

- **Ejecutar contenedor Ubuntu creando una terminal**

$ docker container run -it ubuntu:focal

```
root@ubuntu-server:~# docker container run -it ubuntu:focal
root@5f289a11d4f8:/# pwd
/
root@5f289a11d4f8:/#
```

Ejecutar contenedor en segundo plano con 'd':

$ docker container run -itd ubuntu:focal

Detener contenedor:

$ docker stop <ID/name>

Uso de attach para ejecutar comandos: El contenedor debe de estar ejecutandose.

$ docker container attach <ID>

Sin embargo cuando salgamos el contenedor se detendrá. Cuando salgamos de la terminal al hacer el exit se elimina el proceso y el contenedor se apaga automáticamente.

La alternativa es usar "exec"

$ docker container exec 407266fd1cbd ls

Nota: Al final se agrega el comando.

Para evitar estar agregando comandos al final se usa exec con terminal interactiva y se ejecuta el proceso bash para que abra una terminal:

$ docker container exec -it <ID/name> bash

De esta manera se evita que estar repitiendo el comando con diferente comando.

Además, si salimos de la terminal el contenedor no se detiene.

-Mostrar procesos ejecutándose dentro de un contenedor:

$ docker container top <ID>

Mostrará el proceso -itd ejecutandose.

Puertos

Esto permite exponer los puertos de los servicios que se estén ejecutándose en el contenedor hacia afuera.

-Obtener lista de contenedores solo activos:

$ docker container ls -q

-Detener a todos los contenedores activos con un solo comando:

$ docker container stop $(docker container ls -q)

-Remover a todos los contenedores detenidos:

$ docker container prune

-Ejecutar contenedor y exponer puerto, ejemplo usando nginx que usa el puerto 80:

$ docker container run -d -p 80:80 nginx

-Ver que puertos está exponiendo el contenedor:

$ docker container port <ID>

-Dejar que docker solo se encargue de asignar un puerto al servicio: usar P Mayuscula.

$ docker container run -d -P nginx

-Exponer múltiples puertos de un contenedor, solo hay que concatenar -p varias veces para mapear los puertos.

$ docker container run -d -p 80:80 -p 3006:30090 -p 8081:2121 nginx

-Obtener logs de los contenedores:

$ docker container logs <ID/name>

-Obtener imagen de MySQL para observar los logs:

$ docker image pull mysql

-Ejecutar imagen en segundo plano:

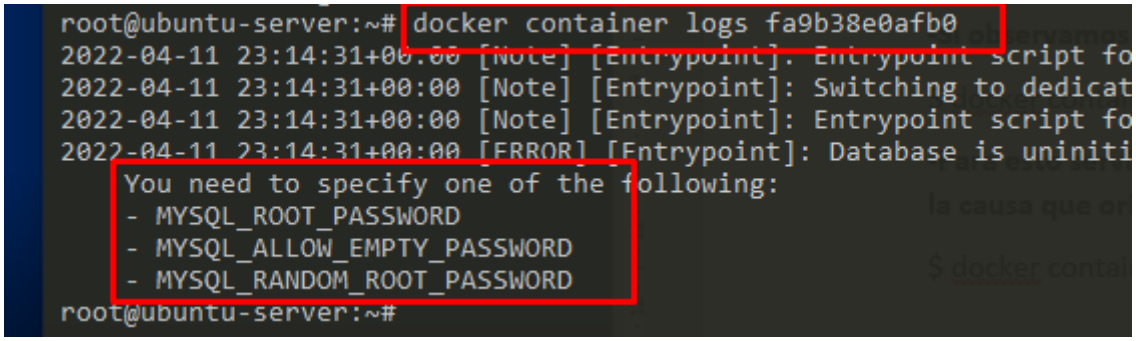
$ docker container run -d mysql

-Si observamos los contenedores veremos que se ha detenido automáticamente:

$ docker container ls -a

-Para esto servirán los logs, se procede a ver los logs del contenedor para ver cual fue la causa que origino que el container se detuviera:

$ docker container logs <ID_CONTAINER/name>



-Commits:

La idea de los commits es coger un contenedor con todos sus recursos y convertirlo a una imagen.

-Ejecutar contenedor de Ubuntu:

$ docker container run -dit ubuntu:focal

-Abrir una terminal en el contenedor:

$ docker container exec -it <ID > bash

-Luego crear archivos/instalar programas para hacer el commit y obtener la imagen.

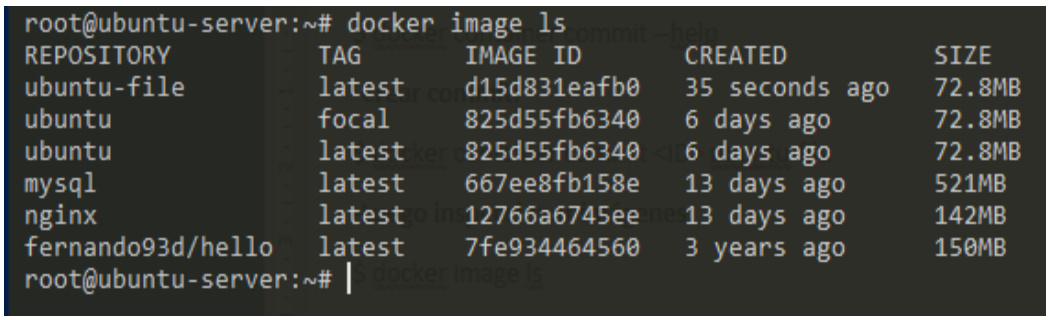
$ docker container commit --help

-Crear commit:

$ docker container commit <ID> ubuntu-file

-Luego inspeccionar imágenes:

$ docker image ls



-Ejecutar contenedor para ver que el commit tiene el archivo creado:

$ docker container run -it ubuntu-file