



Objetivo: correr una aplicación siempre en el mismo entorno

¿Qué hace?: corre un **CONTENEDOR** a partir de una **IMAGEN**

Es descartable, no es normal resetearlo

se genera con

tiene

- Sistema operativo (Distribución)
- Software (Que permite correr la app)
- Aplicación (Código)

Dockerfile

Archivo con instrucciones sobre cómo crear la imagen

Dockerfile

build

Imagen

run

Contenedor

Cuando muere, se pierden los datos ingresados. Para hacerlos persistentes se pueden usar **VOLÚMENES**.

Es posible persistir una carpeta, por ejemplo, del contenedor en una carpeta local, incluso se pueden mapear archivos de código y modificar el contenido del contenedor ad-hoc. Si se modifica la imagen con el contenedor prendido es necesario volver a construirla.



Es el registro de docker. [Acó](#) se hostean las imágenes. Es posible armar una imagen propia o descargar una que se adapte a las necesidades de la aplicación. Al encontrar la imagen en el registro, ahí mismo está el comando para descargarla, si no es necesario descargar una versión particular al hacer **docker run <nombre-imagen>** se descargara la ultima versión.

También se pueden subir imágenes propias con **docker push**, la única condición es que sean imágenes taggeadas incluyendo el nombre de usuario.

¿Qué pasa si necesitamos más de un contenedor y que estén conectados a través de una misma red?

Para no hacer varios comandos separados y tener que recordarlos todos, existe **docker-compose** que es una herramienta para definir y correr aplicaciones Docker multi container usando un archivo **YAML** para definir todos los contenedores que necesite para mi aplicación.

YAML

COMANDOS ÚTILES

Sintaxis	Uso
<code>docker ps</code>	Lista los contenedores corriendo en el momento
<code>docker ps -a</code>	Lista los contenedores que corrieron hace un tiempo
<code>docker pull <nombre-imagen>:<version></code>	Descarga una versión particular de una imagen
<code>docker images</code>	Lista las imágenes locales
<code>docker logs <container-id/container-name></code>	Muestra los logs de un contenedor. El id del contenedor lo podemos obtener desde <code>docker ps</code>
<code>docker logs -f <container-id/container-name></code>	Muestra los logs en vivo de un contenedor
<code>docker build <-t nombre-imagen> .</code>	Construye la imagen y le asigna el nombre. El segundo parámetro es el directorio donde está el Dockerfile
<code>docker run <nombre-imagen></code>	Ejecuta un contenedor a partir de una imagen
<code>docker exec -it <container-id> sh</code>	Ejecuta un comando en un contenedor que está corriendo (i: interactive t: terminal)
<code>docker stop <container-id></code>	Detiene un contenedor
<code>docker start <container-id></code>	Permite reiniciar un contenedor que se había apagado (No es normal hacer esto porque los contenedores son descartables, pero por si acaso)
<code>docker run -v <directorio-local:directorio-container-que-quiero-persistir> -p 3000:3000 <nombre-imagen></code>	Ejecuta un contenedor a partir de una imagen exponiendo a la red local el puerto 3000 del contenedor en el 3000 local y asignando un volumen en una carpeta local para persistir una carpeta del container.
<code>docker tag <id-imagen> <usuario/nombre-imagen:version></code>	Taggea la imagen
<code>docker push <usuario/nombre-imagen:version></code>	Sube la imagen al registro de docker
<code>docker-compose up</code>	Corre todos los contenedores que indique el archivo .yaml
<code>docker-compose down</code>	Baja todos los contenedores y borra la red