



Gestión de imágenes docker

Hasta ahora hemos **descargado** las imágenes o bien a través de la orden ***docker pull*** o bien al ejecutar ***docker run*** cuando la imagen de base para el container no existía aún en nuestro sistema.

Veremos cómo realizar ese **borrado**, profundizaremos en las **posibilidades que tenemos al descargarnos** una imagen y descubriremos **cómo obtener información detallada** sobre las mismas.

Descarga de imágenes Docker

En lo que llevamos de curso hemos visto dos formas para poder descargarme una imagen docker:

Usando la el comando ***docker pull*** indicando el nombre de la imagen y la versión de la misma (**TAG**). Si no indicamos nada se descarga la última versión (latest).

Al hacer ***docker run*** indicando, para la ejecución del contenedor, una imagen base que no hayamos descargado previamente. En ese caso se descargará la imagen y posteriormente empezará a ejecutarse el contenedor si todos los parámetros están bien

Borrado de imágenes

Conforme vamos avanzando en el uso de Docker iremos **acumulando imágenes** en nuestro sistema. Estas imágenes, bien es cierto, no ocupan tanto espacio como una máquina virtual pero si hemos descargado varias decenas o centenas de las mismas (basta un par *docker pull* -a para eso) al final nos encontraremos con que **podemos llegar a ocupar una cantidad considerable de espacio en disco** si no tenemos cierto control sobre las mismas.

No obstante, si lo intentamos no va a suceder nada, simplemente se nos mostrará un mensaje de error como el siguiente:

Si aun así queremos borrarla **podemos forzar ese borrado**, lo cuál afectará, evidentemente, a los contenedores que tuviéramos referenciando esa imagen. Eso lo conseguimos añadiendo la opción **-f o --force**.

Este proceso de borrado, sobre todo si tenemos muchas imágenes, puede ser un proceso engorroso. **Para facilitar** esto disponemos de la orden **docker image prune** que tiene tres opciones básicas:

- **-a o --all** para borrar todas las imágenes que no están siendo usadas por contenedores
- **-f o --force** para que no nos solicite confirmación. Es una operación que puede borrar muchas imágenes de una tacada y debemos ser cuidadosos. Os recomiendo no usar esta opción.
- **--filter** para especificar ciertos filtros a las imágenes.

Obteniendo información de las imágenes

Una vez tenemos ya las imágenes descargadas es muy interesante conocerlas al máximo para poder utilizarlas. Para ello tenemos **dos fuentes principales**:

- La **página de la imagen en DockerHub** que suele recoger sobre todo información relativa a aspectos como:
 - Una descripción de la aplicación o servicio que contiene la imagen.
 - Una lista de versiones TAGs disponibles.
 - Variables de entorno interesantes.
 - Cómo ejecutar la imagen.
- La salida de las órdenes **docker image inspect / docker inspect** que nos da ya una información más detallada sobre las características, con todos los metadatos de la misma.

Esta imagen es una imagen parcial, porque se nos muestra mucha información, está en formato JSON (JavaScript Object Notation) y nos da datos sobre aspectos como:

- El **id y el checksum** de la imagen.
- Los **puertos** abiertos.
- La **arquitectura** y el **sistema operativo** de la imagen.
- El **tamaño de la imagen**.
- Los **volúmenes**.
- El **ENTRYPOINT** que es lo que se ejecuta al hacer docker run.
- Las **capas**.
- Y muchas más cosas....

Adicionalmente podemos formatear la salida usando Go Templates y el flag **--format/-f**.

Más comandos relacionados con imágenes

Además de los comandos que hemos visto en los apartados anteriores la orden ***docker image*** tiene una gran variedad de **subcomandos**, que si bien no son necesarios para poder empezar con docker si que es bueno conocer que existen, os recomiendo los siguientes:

- ***docker image build*** para construir una imagen desde un fichero Dockerfile (se verá en el apartado 6).
- ***docker image history*** para que se nos muestre por pantalla la evolución de esa imagen.
- ***docker image save / docker image load*** (o ***docker save / docker load***) para guardar imágenes en fichero y cargarlas desde fichero (se verá en el apartado 6).
- ***docker image tag (docker tag)*** para añadir TAGs (versiones) a las distintas imágenes.