

IMÁGENES DOCKER

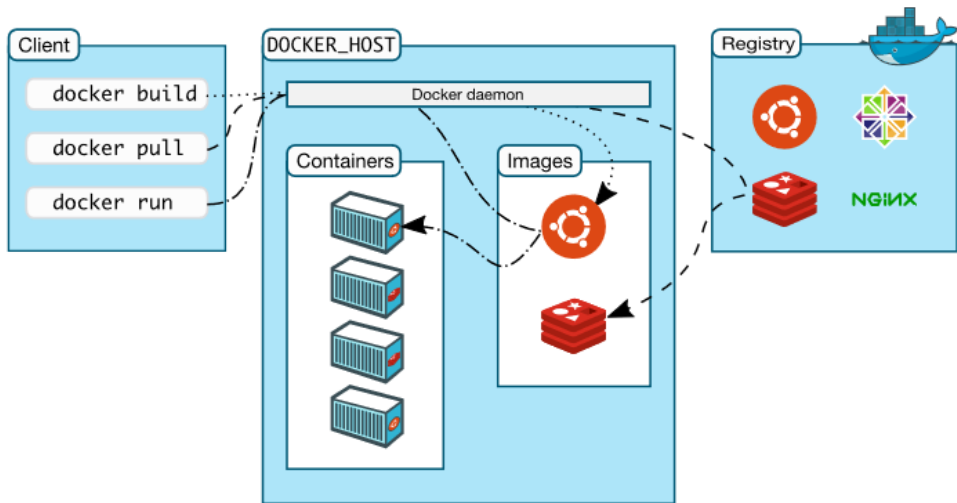
JOSÉ DOMINGO MUÑOZ

IES GONZALO NAZARENO

FEBRERO 2021



REGISTROS DE IMÁGENES



- Las **imágenes de Docker** son plantillas de solo lectura, es decir, una imagen puede contener el sistema de archivo de un sistema operativo como Debian, pero esto solo nos permitirá crear los contenedores basados en esta configuración.
- El **Registro docker** es un componente donde se almacena las imágenes generadas por el Docker Engine. Pueden ser privados o públicos. El proyecto nos ofrece **Docker Hub**.



El nombre de una imagen suele estar formado por tres partes:

usuario/nombre:etiqueta

- **usuario**: El nombre del usuario que la ha generado. Coincide con el usuario de Docker Hub. Las **imágenes oficiales** en Docker Hub no tienen nombre de usuario.
- **nombre**: Nombre significativo de la imagen.
- **etiqueta**: Nos permite versionar las imágenes. Si no indicamos etiqueta, por defecto se usa la etiqueta **latest**.



Para crear un contenedor es necesario usar una imagen que tengamos descargado en nuestro registro local. Por lo tanto al ejecutar `docker run` se comprueba si tenemos la versión indicada de la imagen y si no es así, se precede a su descarga.

Las principales instrucciones para trabajar con imágenes son:

- `docker images`: Muestra las imágenes que tenemos en el registro local.
- `docker pull`: Nos permite descargar la última versión de la imagen indicada.
- `docker rmi`: Nos permite eliminar imágenes. No podemos eliminar una imagen si tenemos algún contenedor creada a partir de ella.
- `docker search`: Busca imágenes en Docker Hub.
- `docker inspect`: nos da información sobre la imagen indicada:

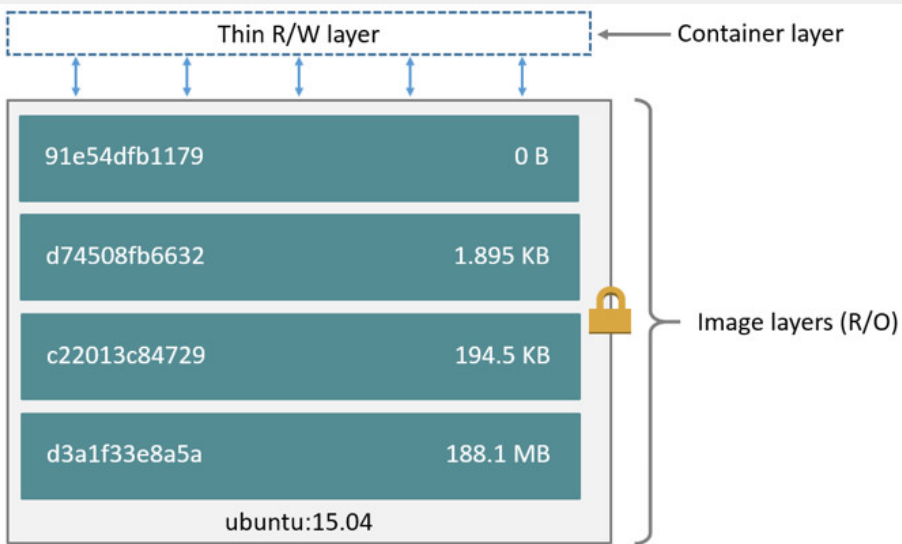


¿CÓMO SE ORGANIZAN LAS IMÁGENES?

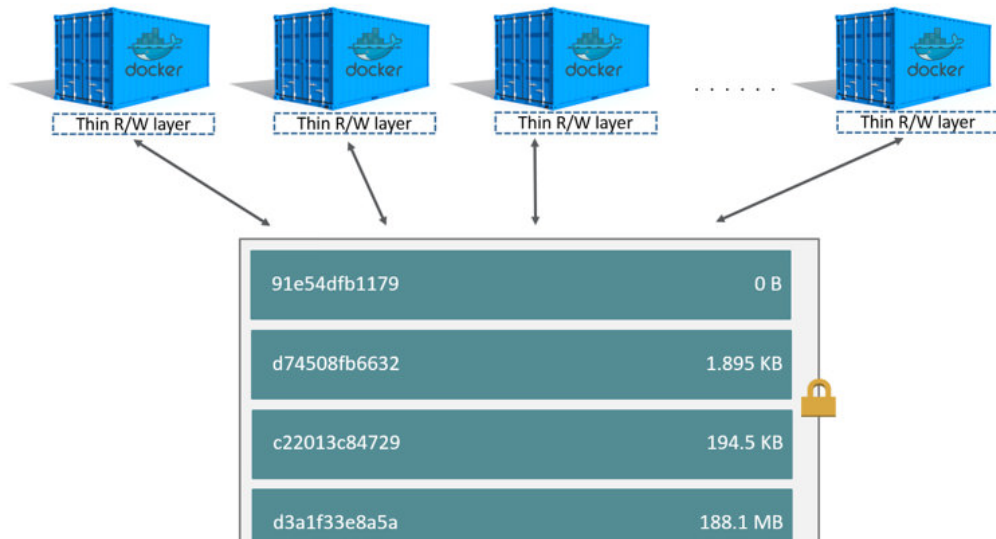
- Las imágenes están hechas de **capas ordenadas**. Puedes pensar en una capa como un conjunto de cambios en el sistema de archivos.
- Si tienes muchas imágenes basadas en capas similares, como Sistema Operativo base o paquetes comunes, entonces todas éstas capas comunes será almacenadas solo una vez.
- Cuando un nuevo contenedor es creado desde una imagen, todas las capas de la imagen son únicamente de lectura y una delgada capa lectura-escritura es agregada arriba. Todos los cambios efectuados al contenedor específico son almacenados en esa capa.
- **No podemos eliminar una imagen cuando tenemos contenedores creados a partir de ella.**



¿CÓMO SE ORGANIZAN LAS IMÁGENES?



¿CÓMO SE ORGANIZAN LAS IMÁGENES?



CREACIÓN DE INSTANCIAS DESDE IMÁGENES

Por ejemplo en la imagen ubuntu el proceso por defecto es bash, por lo tanto podemos ejecutar:

```
$ docker run -it --name contenedor1 ubuntu
```

Pero podemos indicar el comando a ejecutar en la creación del contenedor:

```
$ docker run ubuntu /bin/echo 'Hello world'
```

Otro ejemplo: la imagen httpd:2.4 ejecuta un servidor web por defecto, por lo tanto al crear el contenedor:

```
$ docker run -d --name my-apache-app -p 8080:80 httpd:2.4
```



1. Descarga las siguientes imágenes: `ubuntu:18.04`, `httpd`, `tomcat:9.0.39-jdk11`, `jenkins/jenkins:lts`, `php:7.4-apache`.
2. Muestras las imágenes que tienes descargadas.
3. Crea un contenedor demonio con la imagen `php:7.4-apache`.
4. Comprueba el tamaño del contenedor en el disco duro.
5. Con la instrucción `docker cp` podemos copiar ficheros a o desde un contenedor. Copia un fichero `info.php` al directorio `/var/www/html` del contenedor.
6. Vuelve a comprobar el espacio ocupado por el contenedor.
7. Accede al fichero `info.php` desde un navegador web.

