Mondragon Unibertsitatea

Dockers: Imágenes

Docker: Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

- Objetivos:
  - Conceptos básicos del entorno Docker
  - ¿Qué son las imágenes Docker?
  - Como llevar a cabo las operaciones básicas.

Docker: Conceptos básicos

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Componentes Principales

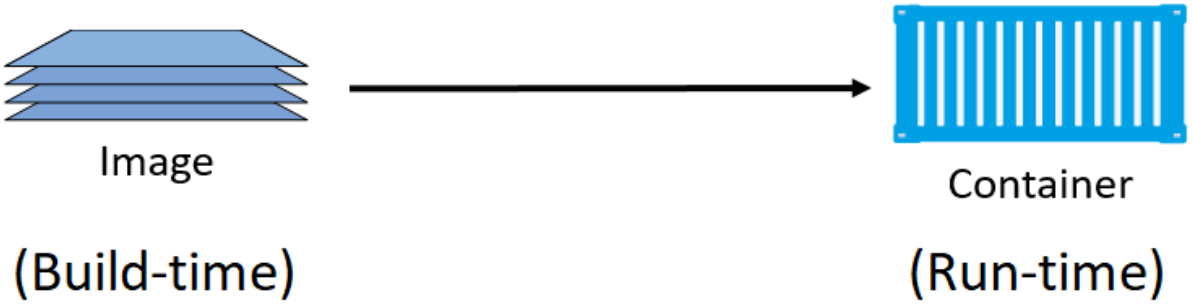
Docker tiene tres componentes principales:

- Las **imágenes** son el componente de **construcción** de Docker y son las plantillas de sólo lectura que definen un sistema operativo de aplicación.
- Los **contenedores** son el componente de **ejecución** de Docker y se crean a partir de imágenes. Los contenedores pueden ser ejecutados, iniciados, detenidos, movidos y eliminados.
- Las imágenes se almacenan, comparten y gestionan en un registro y son el componente de distribución de Docker. Docker Store es un registro de acceso público y está disponible en http://store.docker.com.

Para que estos tres componentes trabajen juntos, el **demonio Docker (o motor Docker)** se ejecuta en una máquina anfitriona y hace el trabajo pesado de construir, ejecutar y distribuir contenedores Docker. Además, el cliente es un binario Docker que acepta comandos del usuario y se comunica con el motor.

Docker: Conceptos básicos

Mondragon Unibertsitatea
Goi Eskola Politeknikoa



Docker: Conceptos básicos

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

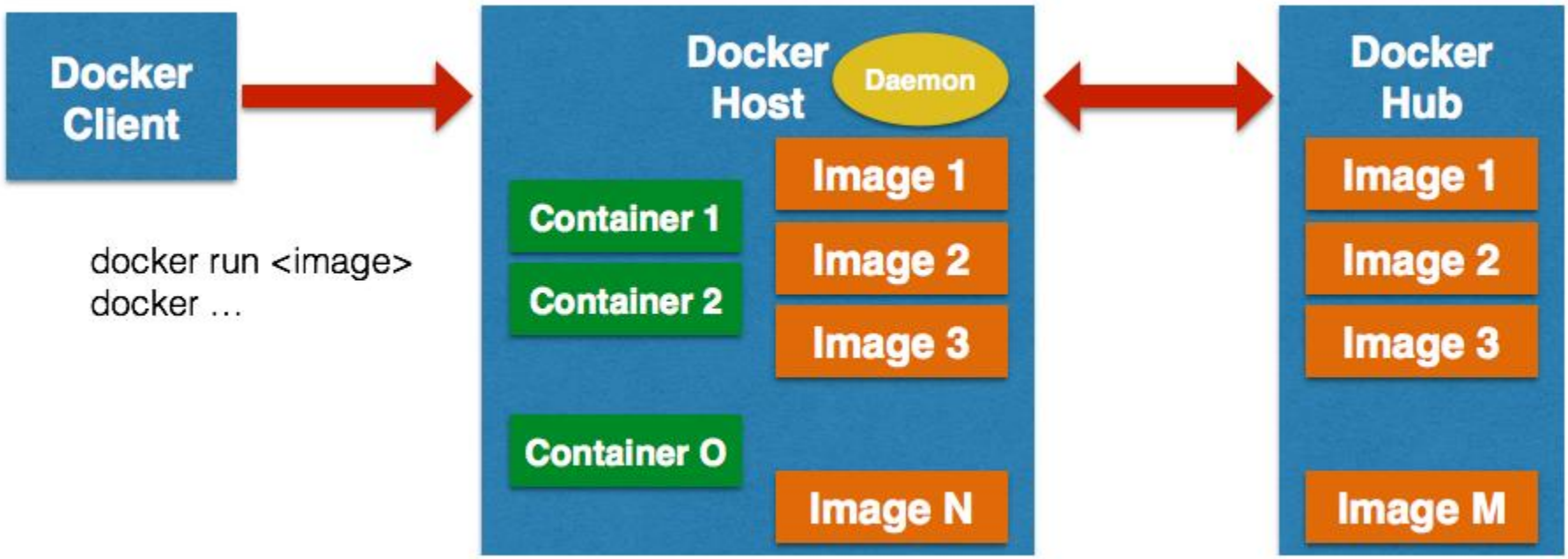
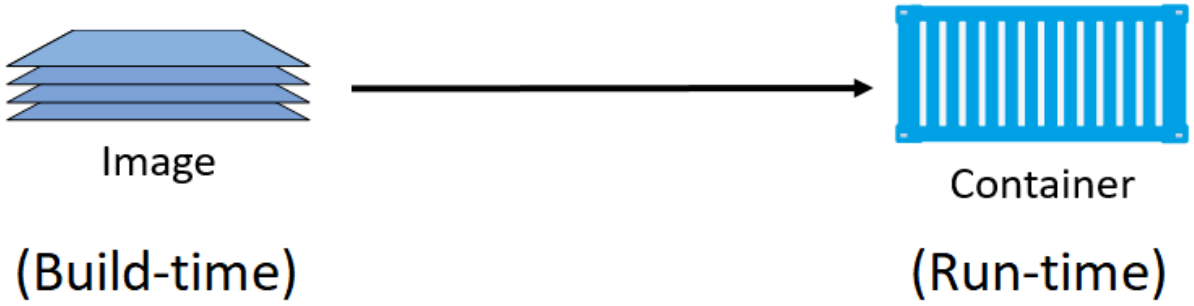


Figure 1. Docker architecture

El Cliente se comunica con el Motor que está ubicado en el mismo host o en un host diferente. El cliente utiliza el comando pull para solicitar al motor que extraiga una imagen del registro. A continuación, el motor descarga la imagen de Docker Store o del registro que se haya configurado. Se pueden descargar varias imágenes del registro e instalarlas en el motor. El cliente utiliza el comando de ejecución del contenedor.

Docker Images en detalle

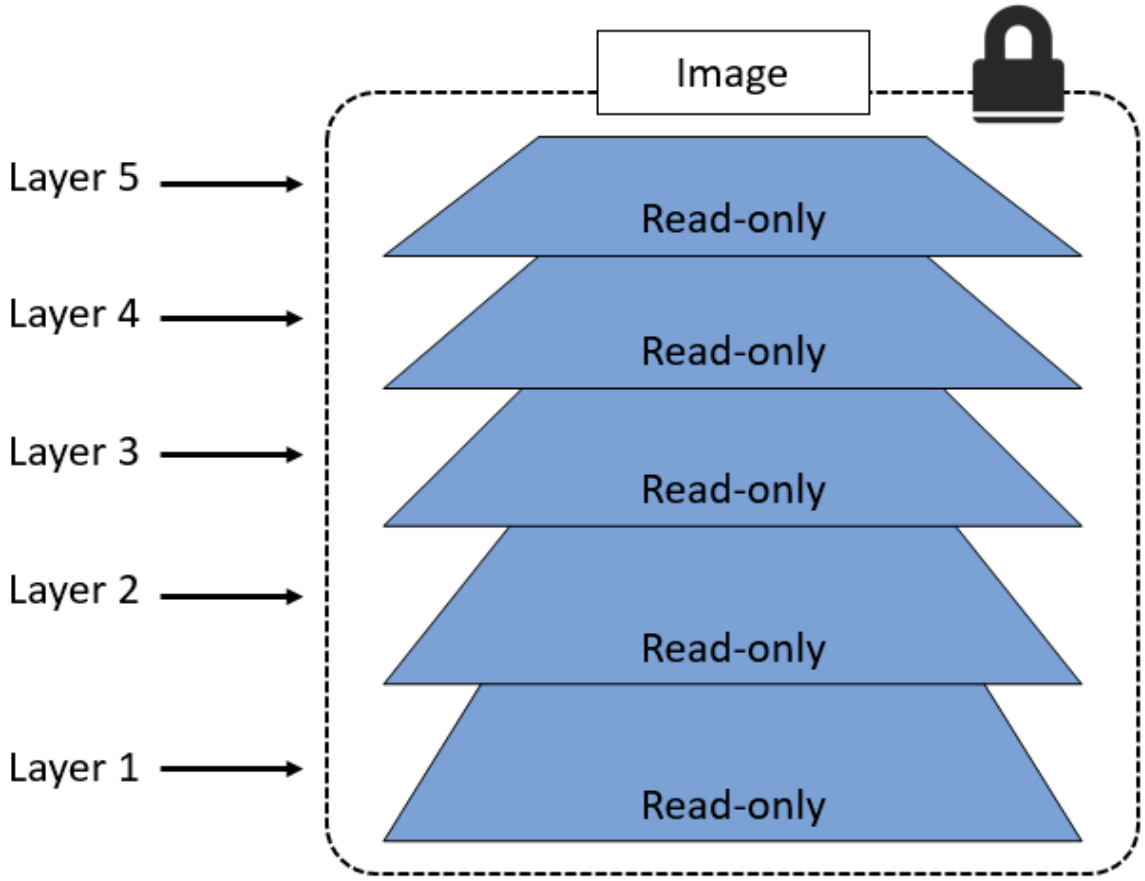
Mondragon Unibertsitatea
Goi Eskola Politeknikoa



La imagen oficial de *Alpine Linux* Docker es de unos **4-5MB**
La imagen de Ubuntu Docker actualmente es de unos **110MB**
La imagen de Microsoft .NET (microsoft/dotnet:latest) ocupa más de **1.7GB**
La imagen de Windows Server 2016 Nano Server (microsoft/nanoserver:latest) es ligeramente superior a **1GB**

Docker Images

Imagenes y layers/capas



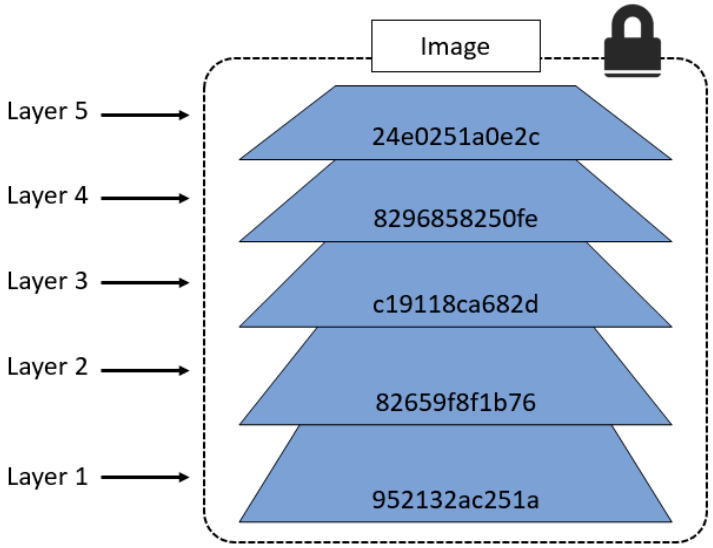
Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Imagenes y layers/capas

- Veamos la salida del commando $ docker image pull ubuntu:latest

```
$ docker image pull tomcat:latest
```



Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Imagenes y layers/capas

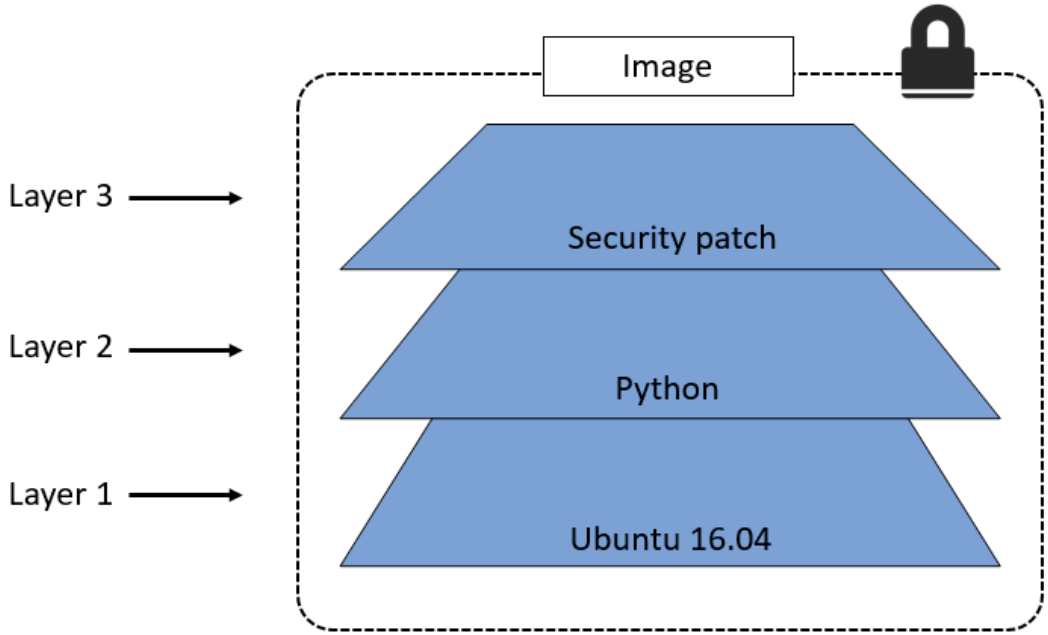
```
$ docker image inspect tomcat:latest
[
    {
        "Id": "sha256:bd3d4369ae.......fa2645f5699037d7d8c6b415a10",
        "RepoTags": [
            “tomcat:latest"
        <Snip>
        "RootFS": {
            "Type": "layers",
            "Layers": [
                "sha256:c8a75145fc...894129005e461a43875a094b93412",
                "sha256:c6f2b330b6...7214ed6aac305dd03f70b95cdc610",
                "sha256:055757a193...3a9565d78962c7f368d5ac5984998",
                "sha256:4837348061...12695f548406ea77feb5074e195e3",
                "sha256:0cad5e07ba...4bae4cfc66b376265e16c32a0aae9"
]
}
}]
```

Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Imagenes y layers/capas

- Ejemplo:

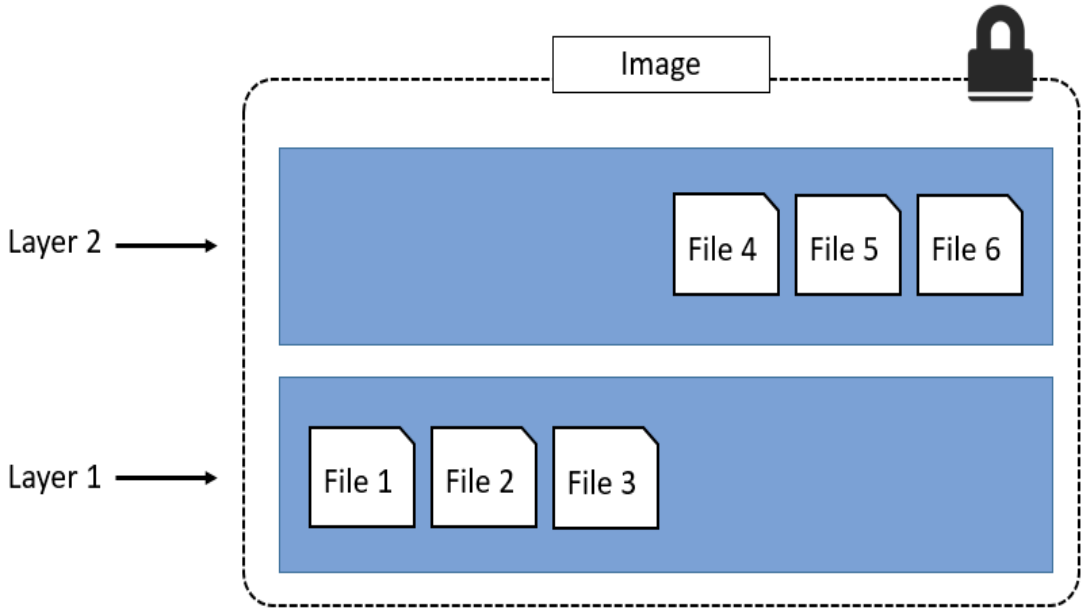


Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Imagenes y layers/capas

- Ejemplo:
  - Dos capas
  - Cada capa tiene 3 ficheros
  - Imagen final: 6 ficheros

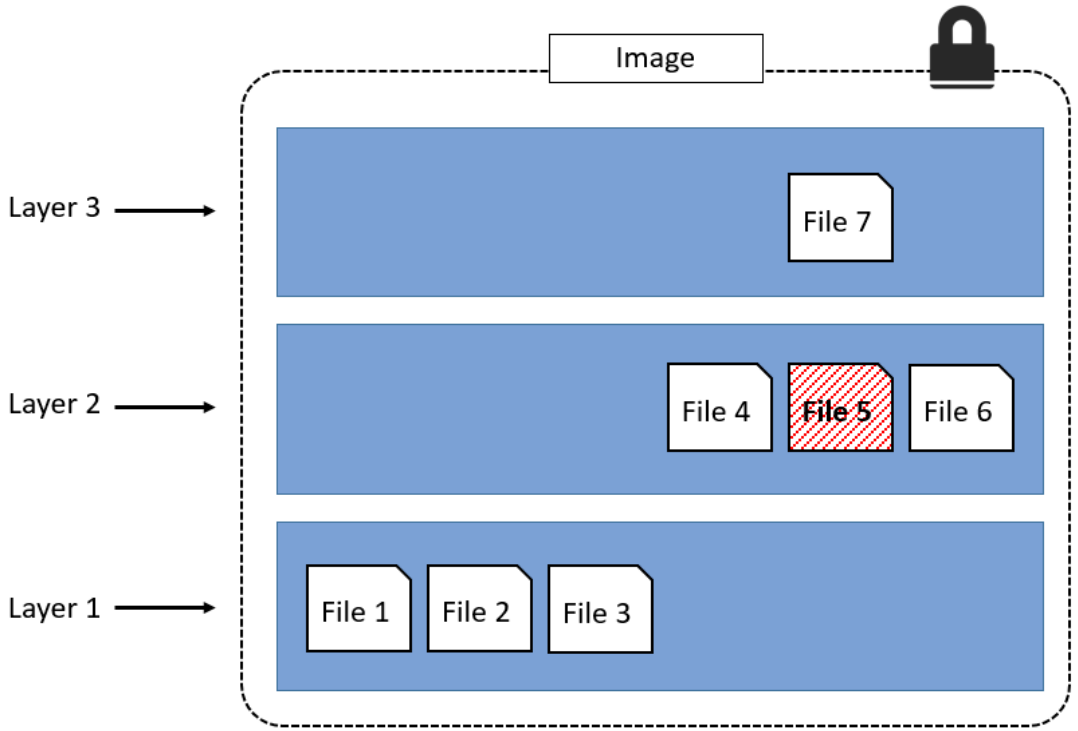


Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Imagenes y layers/capas

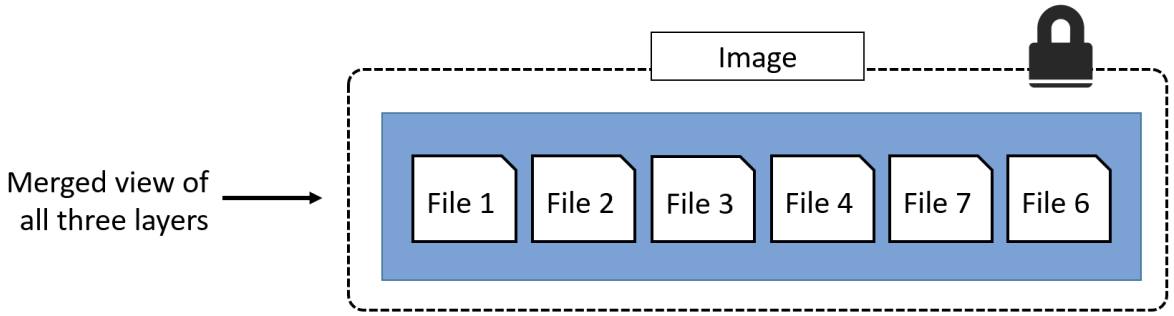
- Ejemplo
  - Tres capas
  - Imagen final: 6 ficheros

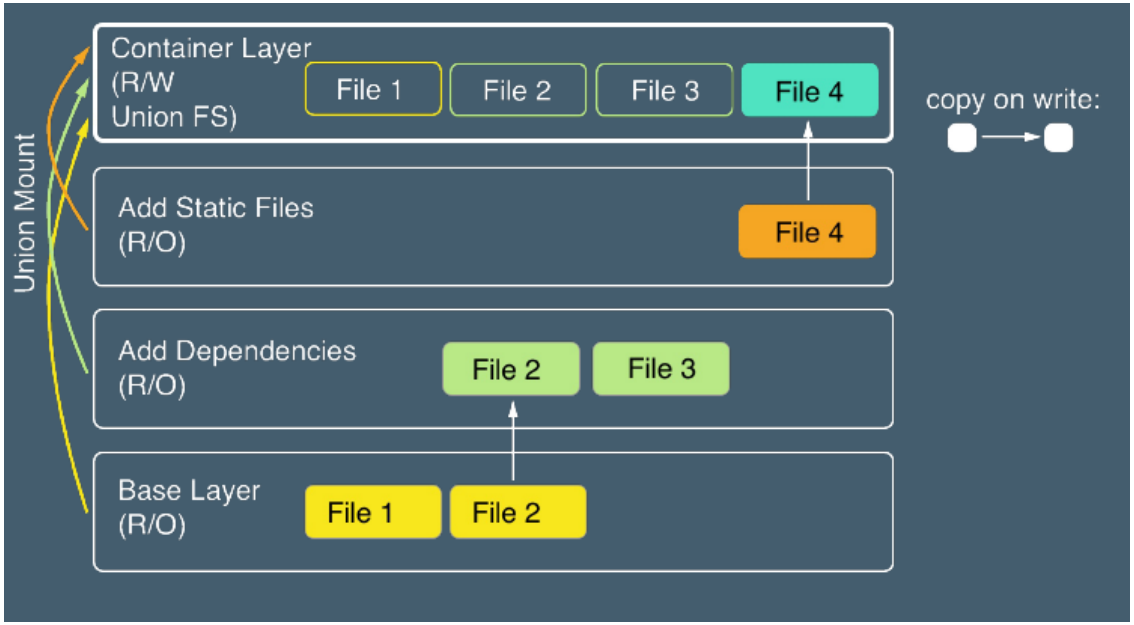


Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Imagenes y layers/capas





Docker Images en detalle

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Docker Image

Ya hemos visto que las imágenes Docker son plantillas de sólo lectura desde las que se lanzan los contenedores Docker. Cada imagen consiste en una serie de capas. Docker hace uso de sistemas de archivos de unión para combinar estas capas en una sola imagen. Los sistemas de archivos de unión permiten que los archivos y directorios de sistemas de archivos separados, conocidos como ramas, se superpongan de forma transparente, formando un único sistema de archivos coherente.

Una de las razones por las que Docker es tan ligero es por estas capas. Cuando se cambia una imagen Docker -por ejemplo, actualizar una aplicación a una nueva versión- se construye una nueva capa. Así, en lugar de reemplazar toda la imagen o reconstruirla por completo, como se puede hacer con una máquina virtual, sólo se añade o actualiza esa capa. Ahora no es necesario distribuir una imagen completamente nueva, sólo la actualización, lo que hace que la distribución de imágenes Docker sea más rápida y sencilla.

Docker Images en detalle

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Docker Image

Cada imagen parte de una imagen base, por ejemplo ubuntu, una imagen base de Ubuntu, o fedora, una imagen base de Fedora. También puedes utilizar imágenes propias como base para una nueva imagen, por ejemplo si tienes una imagen base de Apache podrías utilizarla como base de todas tus imágenes de aplicaciones web.

Las imágenes Docker se construyen a partir de estas imágenes base utilizando un conjunto simple y descriptivo de pasos que llamamos instrucciones. Cada instrucción crea una nueva capa en nuestra imagen. Las instrucciones incluyen acciones como:

- Ejecutar un comando
- Añadir un archivo o directorio
- Crear una variable de entorno
- Ejecutar un proceso al lanzar un contenedor

Por defecto, Docker obtiene estas imágenes base desde Docker Store.

Estas instrucciones se almacenan en un archivo llamado Dockerfile. Docker lee este Dockerfile cuando se solicita la construcción de una imagen, ejecuta las instrucciones y devuelve una imagen final.

Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

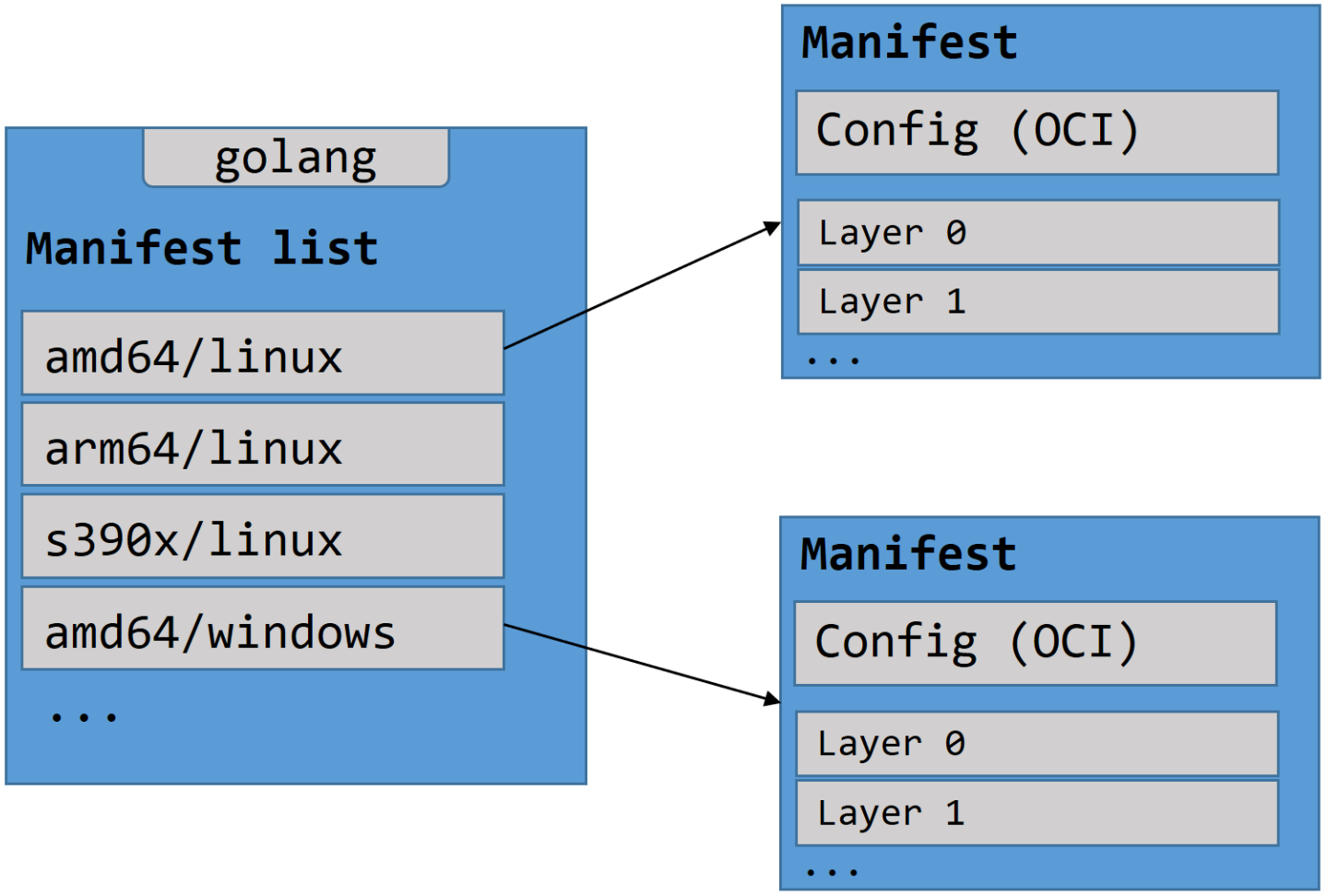
Extracción de imágenes mediante “digest”

- Cada vez que haces un pull de una imagen, el comando docker image pull incluye el digest de la imagen como parte del código resultante.
- Se puede chequear los digests de las imágenes en el repositorio local de tu Docker host añadiendo el flag --digest al comando docker image ls.

```
$ docker image pull alpine
$ docker image ls --digests alpine
REPOSITORY TAG    DIGEST              IMAGE ID       CREATED     SIZE
alpine     latest sha256:3dcd...f73a 4e38e38c8ce0 10 weeks ago 4.8 MB
```

Docker Images

Imágenes multi-arquitectura



Docker Images

Imágenes multi-arquitectura

Ejemplo Linux en x64:

```
$ docker container run --rm golang go version
```

Ejemplo Windows x64:

```
PS> docker container run --rm golang go version
```

Docker Images en detalle

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Repositorio local de imágenes

Comando para ver las imágenes que tiene el sistema:

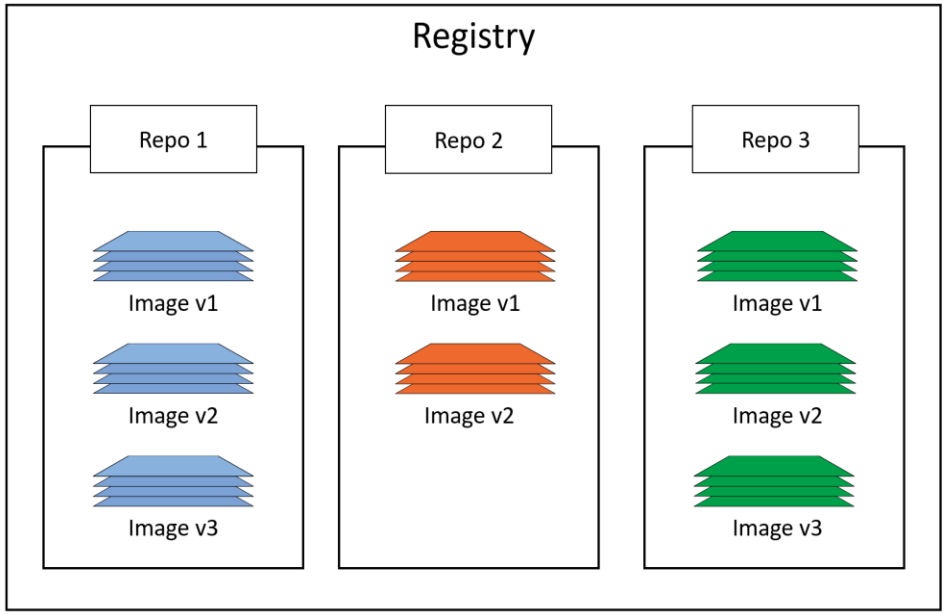
```
$ docker image ls (-xxx diferentes opciones)
```

Docker Images: Naming

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Repositorio de Imágenes

- Docker Hub (https://hub.docker.com).



Docker Images: Naming

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Repositorios Oficiales y No-oficiales

- *Repositorios Oficiales:* contienen imágenes que han sido verificados por Docker, Inc.
- *Repositorios No-Oficiales:* no son seguros, y no suelen estar documentados o construidos de acuerdo a las mejores prácticas.
- Algunos repositorios oficiales:
  - **nginx:** https://hub.docker.com/_/nginx/
  - **busybox:** https://hub.docker.com/_/busybox/
  - **redis:** https://hub.docker.com/_/redis/
  - **mongo:** https://hub.docker.com/_/mongo/
  - …

Docker Images: Naming

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Nombrado y Etiquetado de Imágenes

Cuando se trabaja con una imagen de un repositorio oficial es el formato a utilizar es:

```
$ docker image pull <repository>:<tag>
```

Ejemplo:

```
$ docker image pull alpine:latest
$ docker image pull ubuntu:latest
```

Docker Images: Naming

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Filtrar la salida de `$ docker image ls`

flag “--filter”

```
$ docker image ls --filter dangling=true
```

Dangling: <none>:<none>.

Borrar las imagenes dangling:

```
$ docker image prune
```

Añadiendo el flag -a , Docker también borrará todas las imágenes que no se usan (aquellas que no son usadas por ningún container).

Docker Images: Naming

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Filtrando la salida de $ `docker image ls`

- Actualmente Docker soporta los siguientes filtros:
  - dangling: acepta true o false, y devuelve solo las imágenes dangling (true), o no-dangling (false).
  - before: require el nombre o ID de una imagen como argumento, y devuelve todas las imágenes creadas antes de esa imagen.
  - since: igual que la anterior, pero devuelve las imágenes creadas después de la imagen especificada.
  - label: filtra imágenes basadas en la presencia del label o label y valor.

Docker Images: Naming

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Filtrando la salida de $ `docker image ls`

- flag `--format`

```
$ docker image ls --format "{{.Size}}“
```

Devuelve la propiedad de tamaño de las imagenes Docker

```
$docker image ls --format "{{.Repository}}:
{{.Tag}}: {{.Size}}“
```

¿Qué devolverá?

Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Búsqueda en Docker Hub desde el CLI

`$ docker search`

- `Busca` en todos los repositorios que contienen un cierto string en el campo “NAME”.

```
$ docker search millarramendi
$ docker search alpine
$ docker search alpine --filter "is-official=true“
$ docker search alpine --filter "is-automated=true“
```

Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Imágenes – Los Comandos

- `$ docker image pull` : es el comando para descargar imágenes. Hacemos pull de las imágenes de los repositorios dentro de los registros remotos. Por defecto, las imágenes se extraerán (pull) de los repositorios en el Docker Hub.
- `$ docker image push` : es el comando para cargar imágenes. Hacemos push de una imagen a un repositorio remoto. Por defecto, las imágenes se enviarán a los repositorios en el Docker Hub. Para ello necesitamos logearnos (`$docker login`) y utilizar el formato con el nombre de usuario.
- `$ docker image ls` : muestra todas las imágenes almacenadas en la caché local del host del Docker. Para ver los resúmenes de imágenes del SHA256, añada el flag --digests.
- `$ docker image inspect` : le da todos los detalles de una imagen: datos de capa y metadatos.
- `$ docker image rm` : es el comando para eliminar imágenes. No se puede eliminar una imagen que está asociada con un contenedor que está en ejecución/running (Up) o parado/stopped (Exited).

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Práctica: Gestión de Imágenes

Etiquetar y listar imágenes

- Descargar la imagen tomcat:lastest
- Cambiar el tag de esta imagen

$ docker image tag tomcato:latest tomcat:dev

¿Qué es lo que ocurre al ejecutar este commando? ¿Has creado una nueva imagen?

- Listar las imágenes

¿Cuántas imágenes tomcat tienes? Fíjate en los identificadores. ¿Qué valor tienen? ¿Por qué?

- *Hacer un Push de la imagen a Docker Hub:*

$ docker image push tomcat:dev

- *Debería de dar un error de autentificación*
- *Logeate ($docker login) e intenta hacer la misma operación. ¿Qué ocurre?*

Docker Images

Mondragon Unibertsitatea
Goi Eskola Politeknikoa

Compartir imágenes en Docker Hub

- *Debería de darte un error porque la imagen no tiene el formato adecuado y no consigue identificar tu usuario. El namespace de la imagen no es adecuado para Docker Hub*
- *Para compartir una imagen en Docker Hub, tienes que seguir el siguiente formato :*

  <Docker ID>/<repo name>[:<optional tag>]

  (siendo Docker ID, tu nombre de usuario)
- *Etiqueta de nuevo tu imagen siguiendo el formato definido. Hacer de nuevo el push.*

```
$ docker image tag tomcat:dev <Docker ID>/tomcat:dev
$ docker image push <Docker ID>/tomcat:dev
```

- *Busca en Docker Hub (en tu repositorio) la nueva imagen que has subido y confirma que puedes ver el tag :dev*
- *Lista las imágenes de tu nodo y todavía debe de existir la que no seguía el formato adecuado para el Docker Hub.*
- *Borralo y comprueba realmente qué es lo que se ha borrado.*

¿Siguen las capas de la imagen en el nodo?

Mondragon Unibertsitatea

Goi Eskola Politeknikoa

Eskerrik asko

Muchas gracias

Thank you