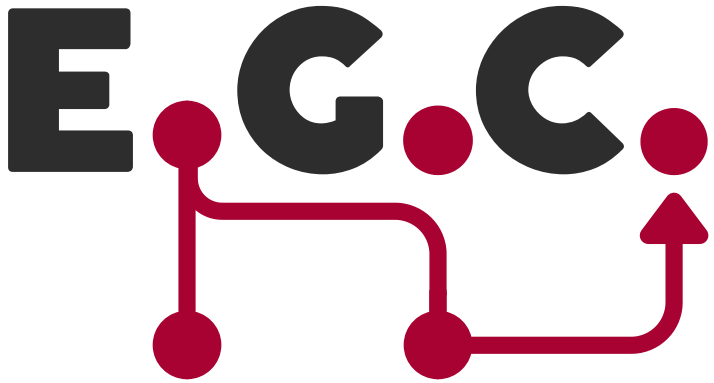


Grado en Ingeniería Informática - Ingeniería del Software

Evolución y Gestión de la Configuración

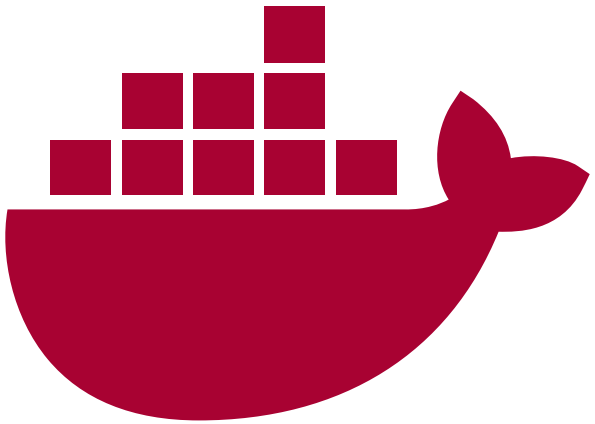




Escuela Técnica Superior de Ingeniería Informática

Práctica 4

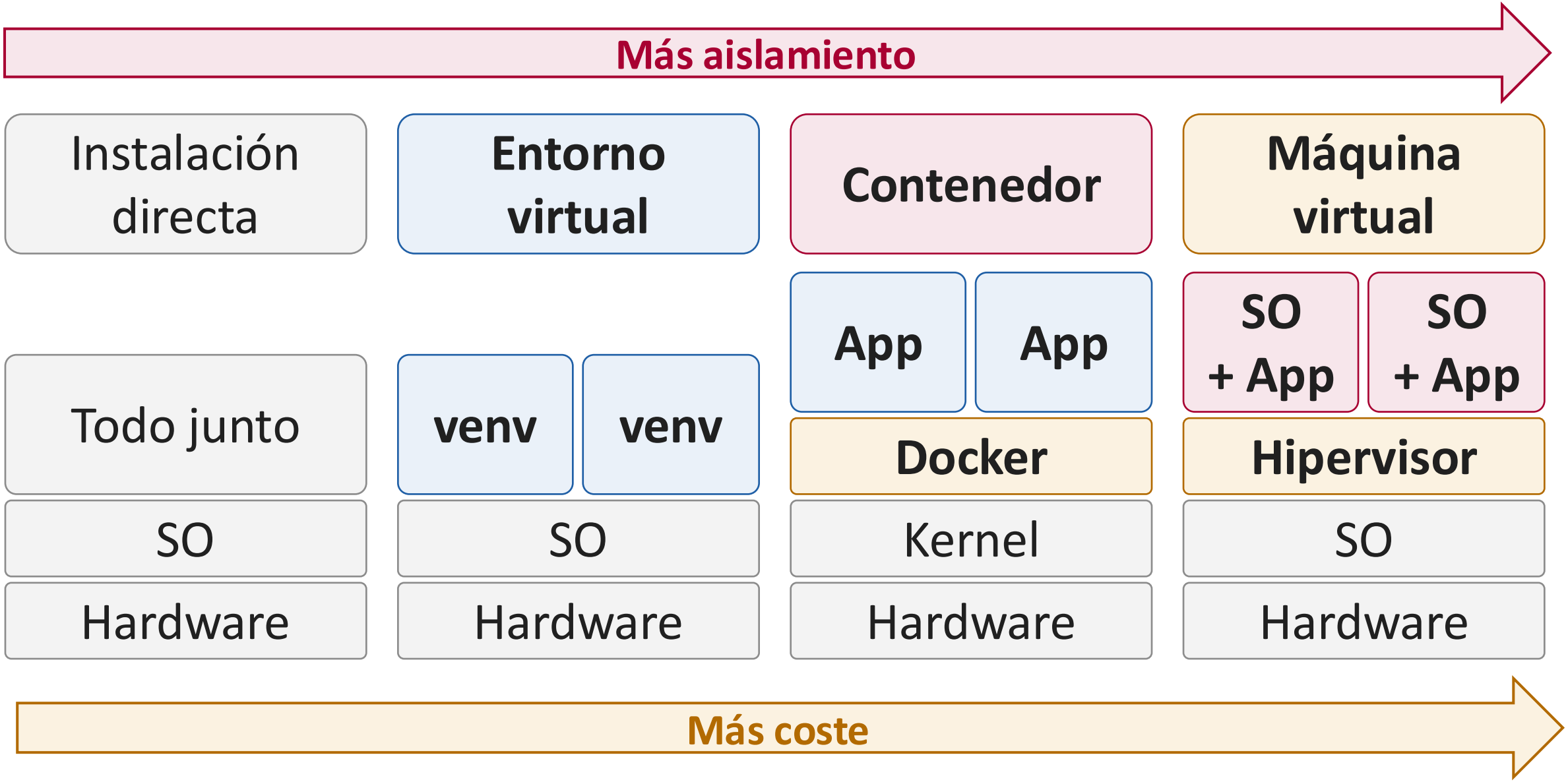
Contenedores, dev-containers aislamiento



1. Introducción a Docker y DockerHub
2. Imágenes
3. Contenedores
4. Volúmenes y redes
5. Composición de servicios: Docker Compose
6. Dev-containers
7. Y yo, ¿qué puedo hacer en mi proyecto?
8. Tutorial: *dockerizando uvlhub*

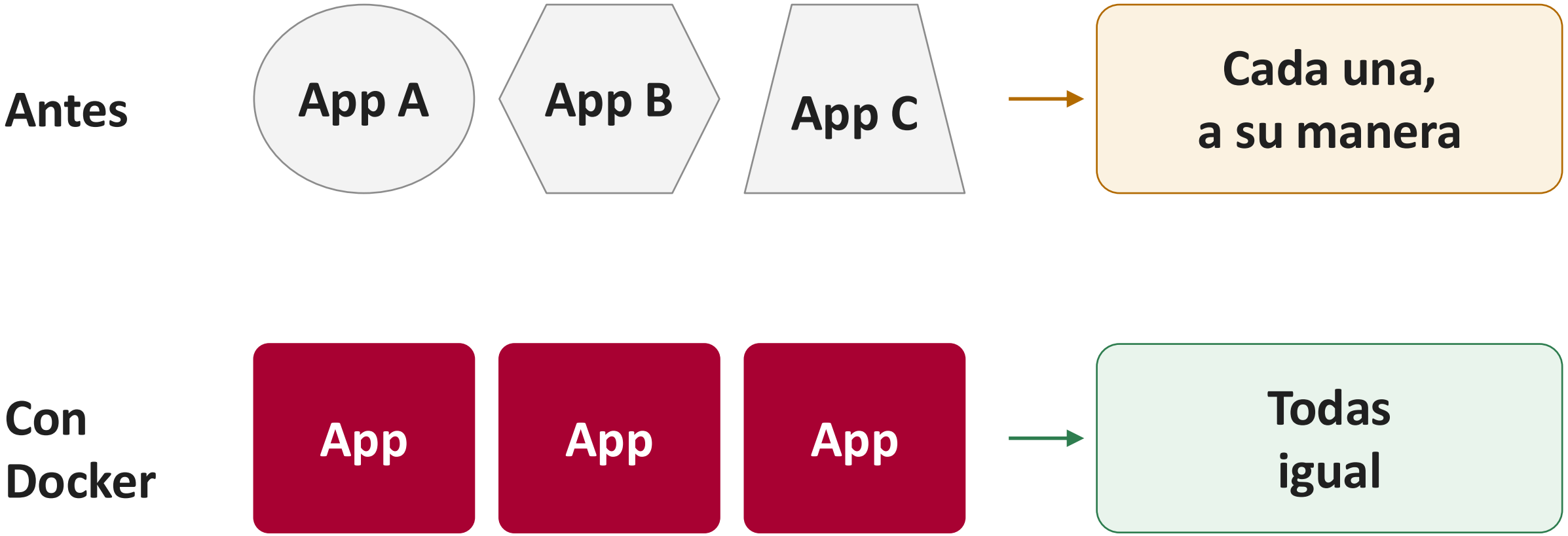
1. **Introducción a Docker y DockerHub**
2. Imágenes
3. Contenedores
4. Volúmenes y redes
5. Composición de servicios: Docker Compose
6. Dev-containers
7. Y yo, ¿qué puedo hacer en mi proyecto?
8. Tutorial: *dockerizando uvlhub*

1. Introducción a Docker



1. Introducción a Docker

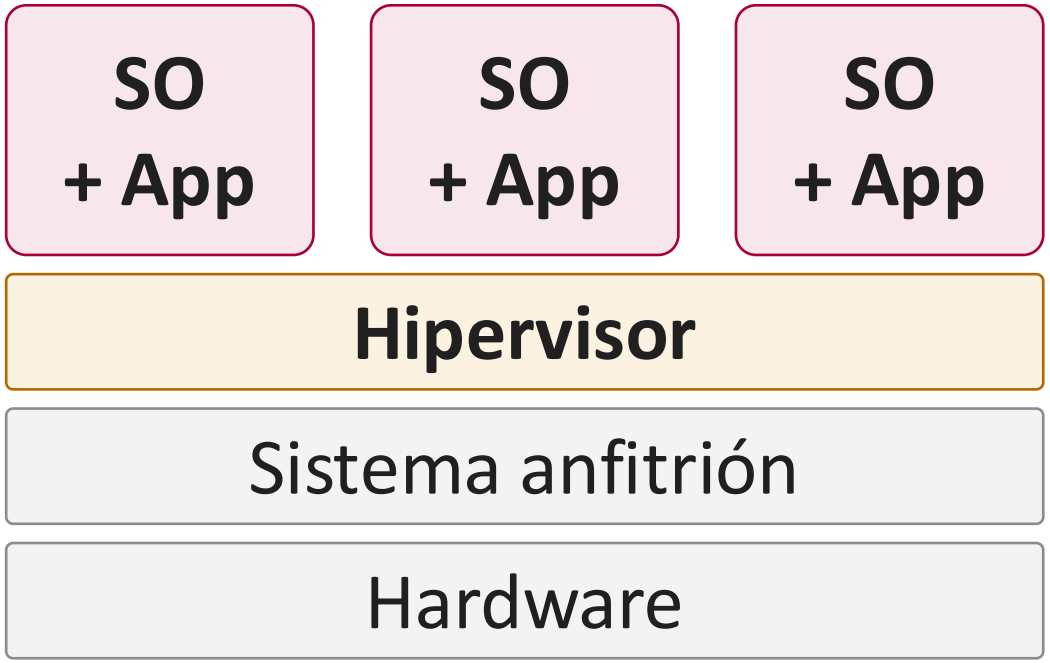
Cajas estándar para el software



1. Introducción a Docker

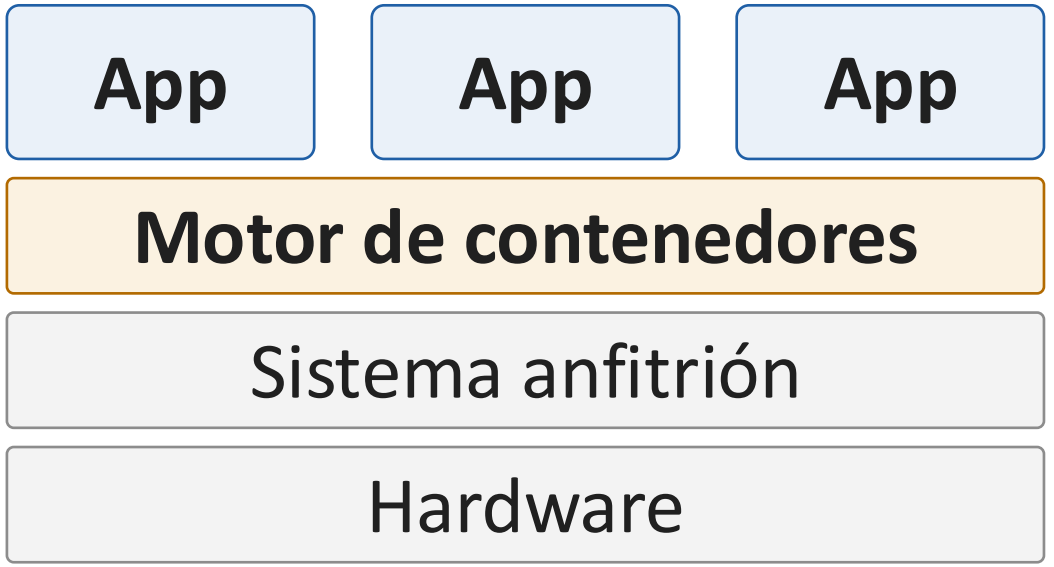
¿Por qué usar contenedores si ya tenemos máquinas virtuales?

Máquinas virtuales



Gigas y minutos

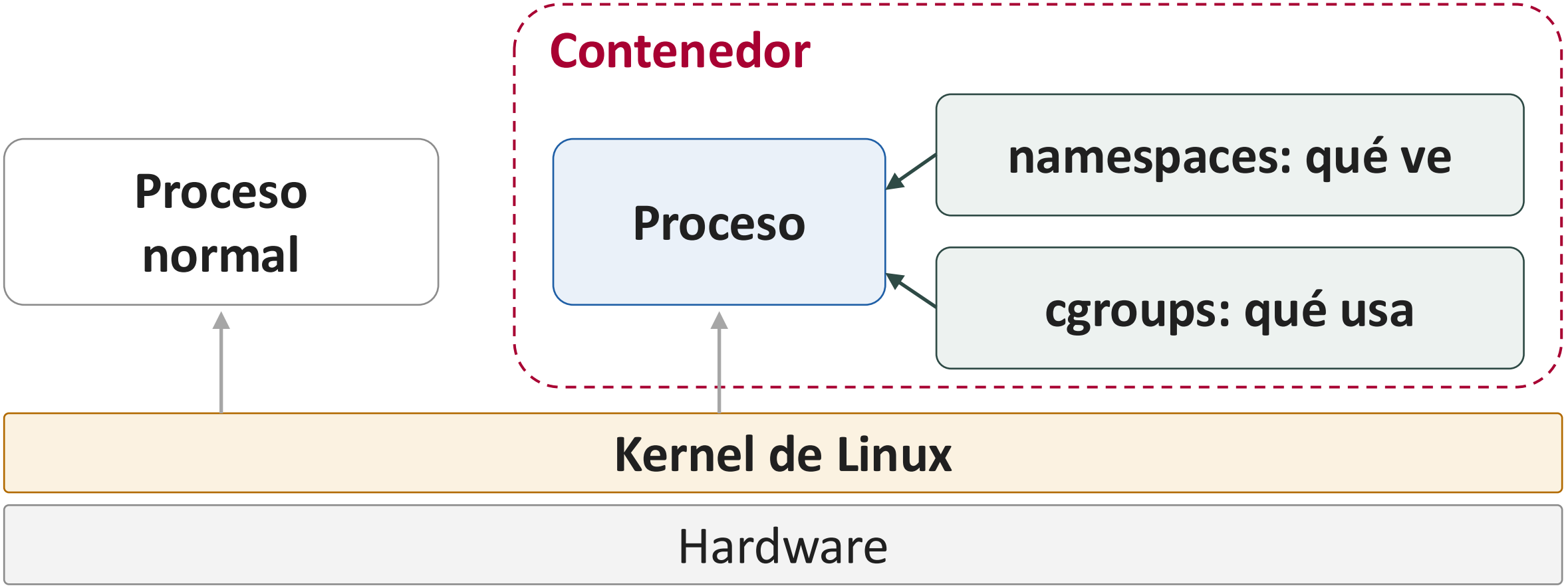
Contenedores



Megas y milisegundos

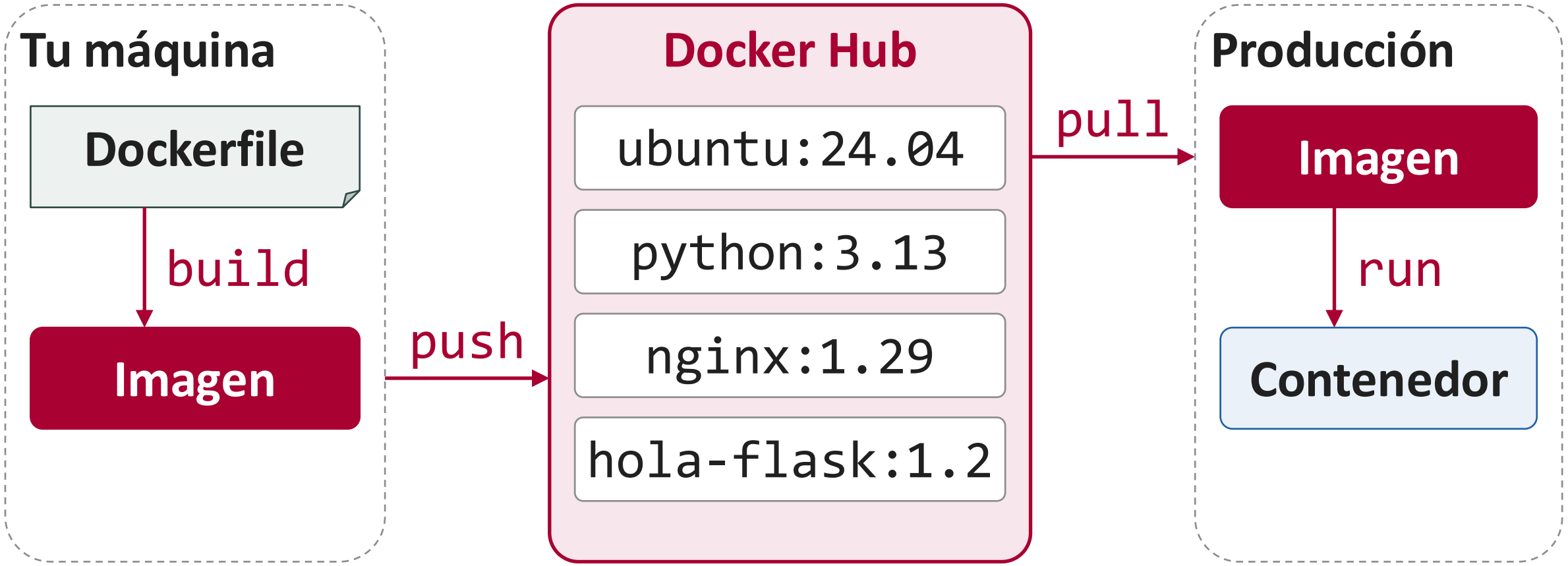
1. Introducción a Docker

Un contenedor es un proceso



Namespaces y cgroups aplicados a un proceso

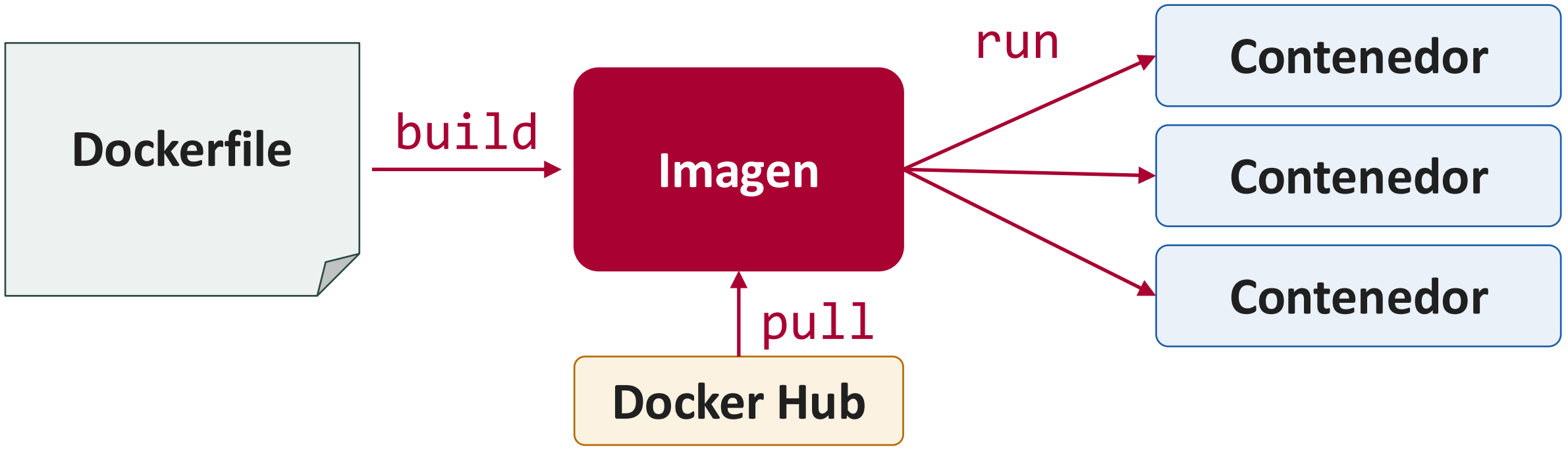
1. Introducción a DockerHub



Una imagen se versiona con su etiqueta: nunca latest

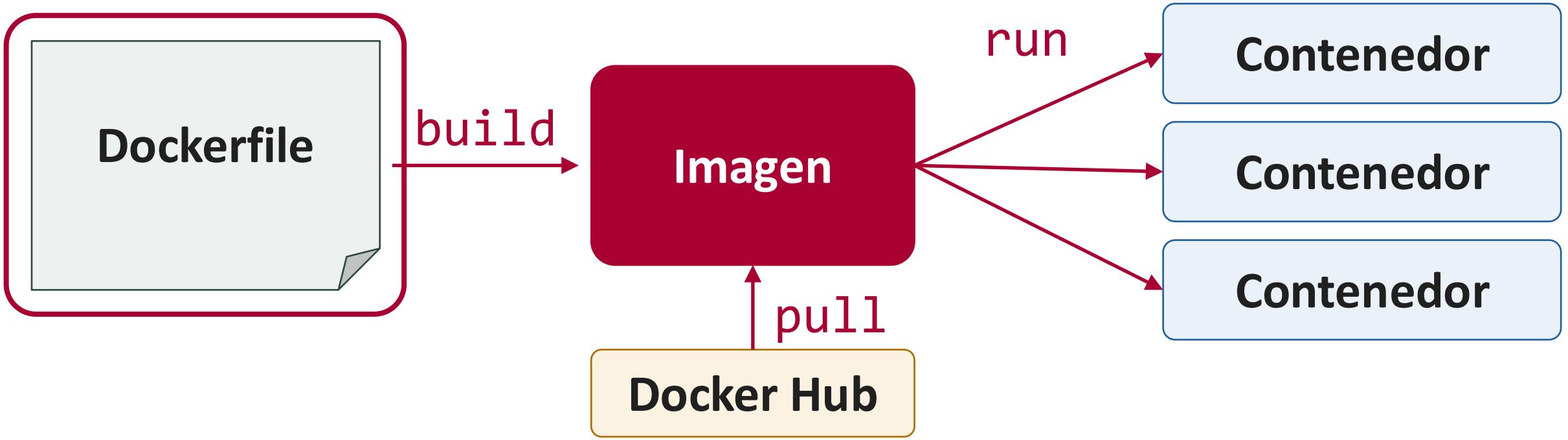
1. Introducción a Docker y DockerHub
2. **Imágenes**
3. Contenedores
4. Volúmenes y redes
5. Composición de servicios: Docker Compose
6. Dev-containers
7. Y yo, ¿qué puedo hacer en mi proyecto?
8. Tutorial: *dockerizando uvlhub*

2. Imágenes



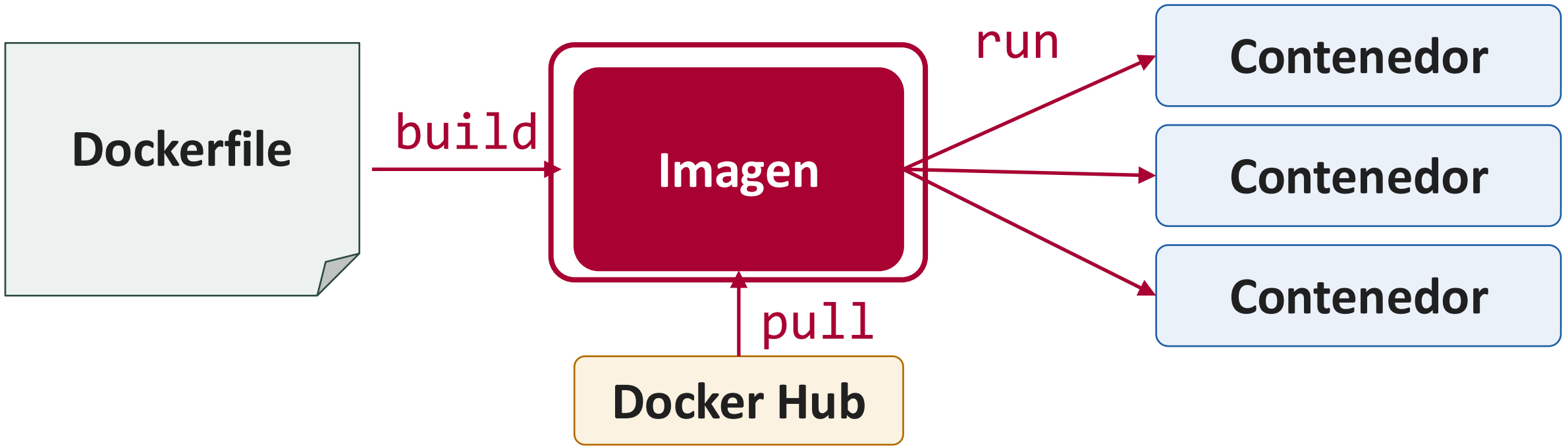
Una imagen, muchos contenedores

2. Imágenes



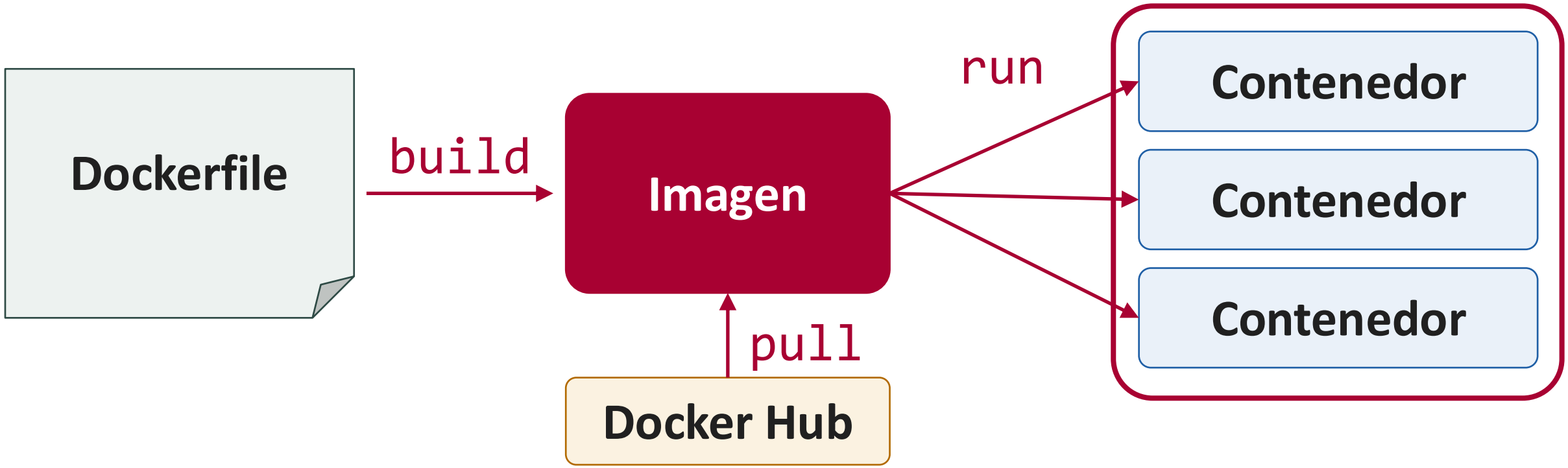
La receta para construir una imagen:
versionada y revisable

2. Imágenes



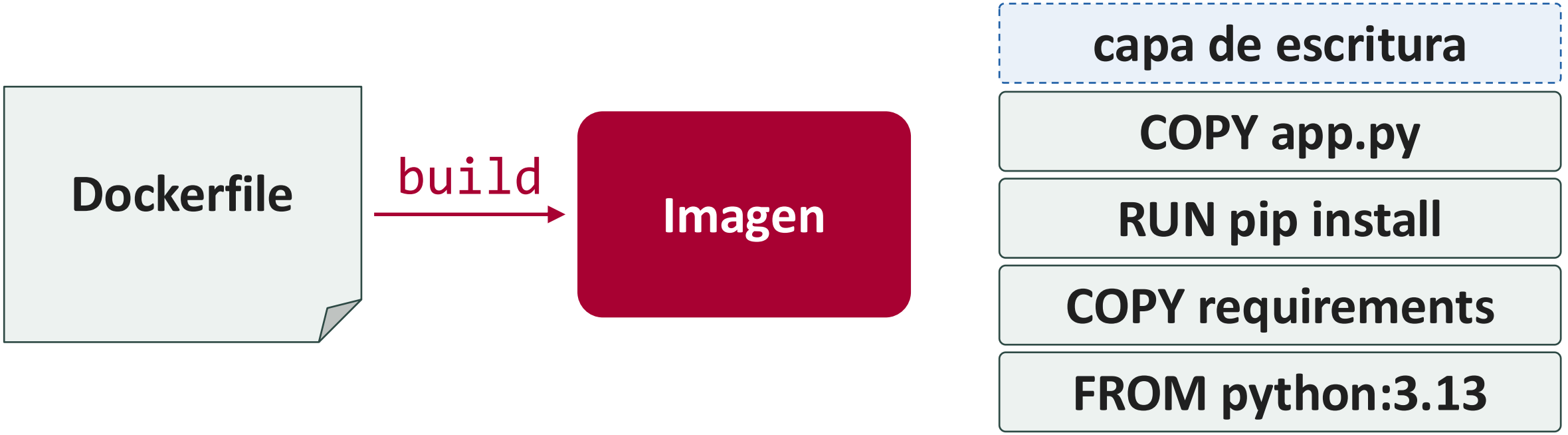
Plantilla de solo lectura con la app y sus dependencias. Un artefacto

2. Imágenes



Una imagen en ejecución: un proceso aislado.
Tantos como quieras

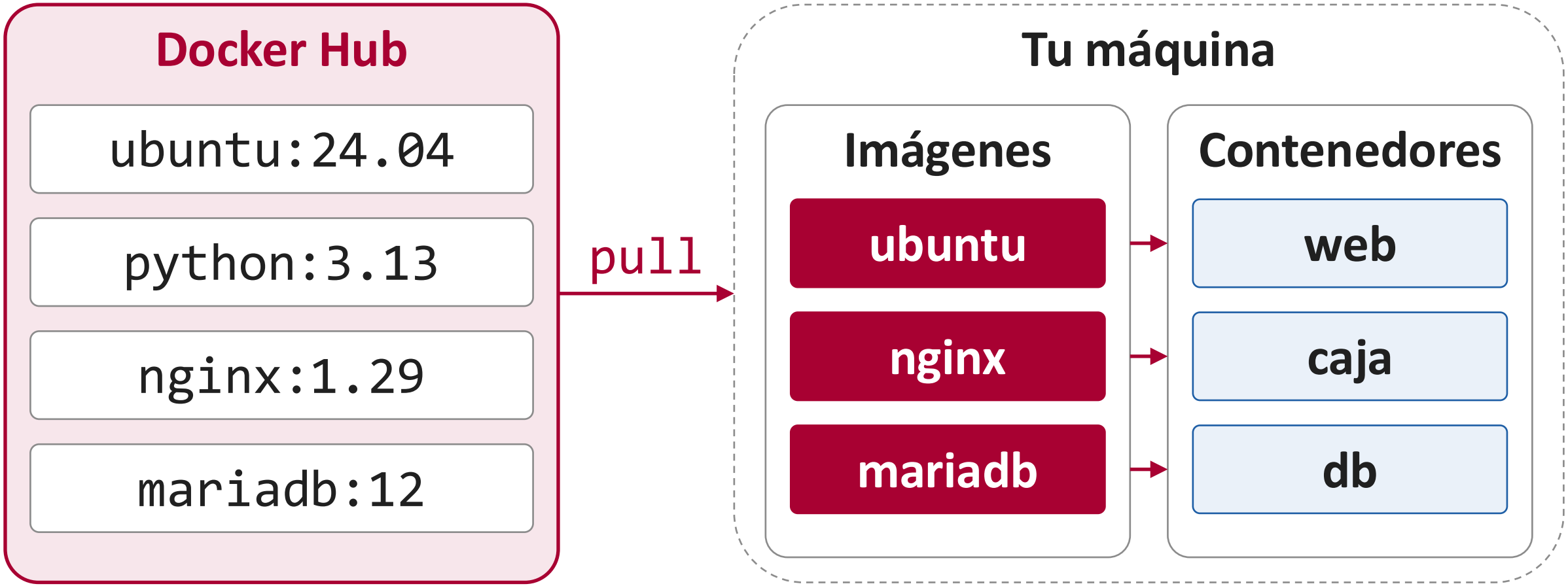
2. Imágenes



Cada instrucción es una capa.
Docker reutiliza las que no cambian

2. Imágenes

Y si no escribo un Dockerfile, ¿de dónde salen las imágenes?



Si no la tienes, docker run la descarga

2. Imágenes

Etiquetas: nombre:versión

`ubuntu` : `24.04`

nombre *etiqueta*

`mariadb:12.0.2`

fija: siempre la misma

`mariadb:latest`

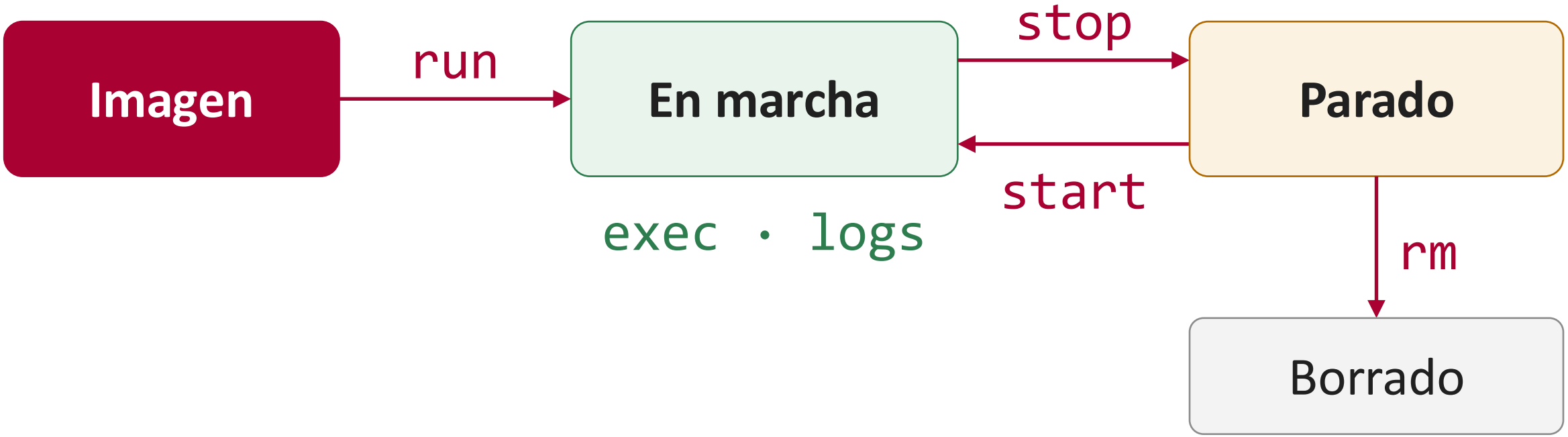
móvil: cambia sin avisar

Una imagen no se toca, se construye otra con otra etiqueta

1. Introducción a Docker y DockerHub
2. Imágenes
3. **Contenedores**
4. Volúmenes y redes
5. Composición de servicios: Docker Compose
6. Dev-containers
7. Y yo, ¿qué puedo hacer en mi proyecto?
8. Tutorial: *dockerizando uvlhub*

3. Contenedores

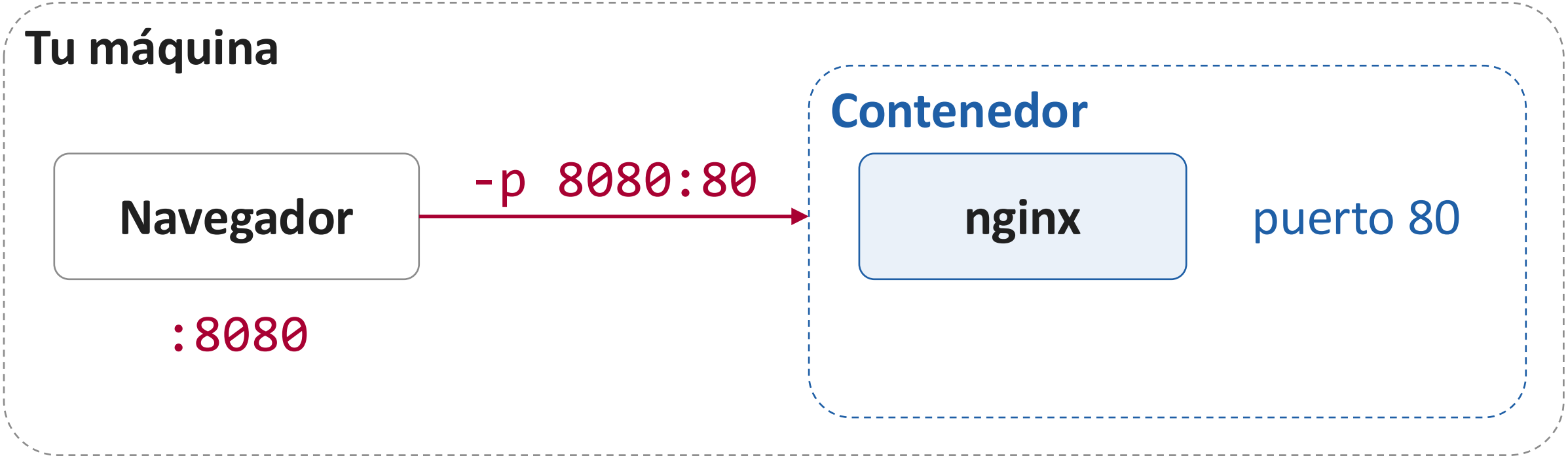
El ciclo de vida de un contenedor



Vive mientras vive su proceso principal

3. Contenedores

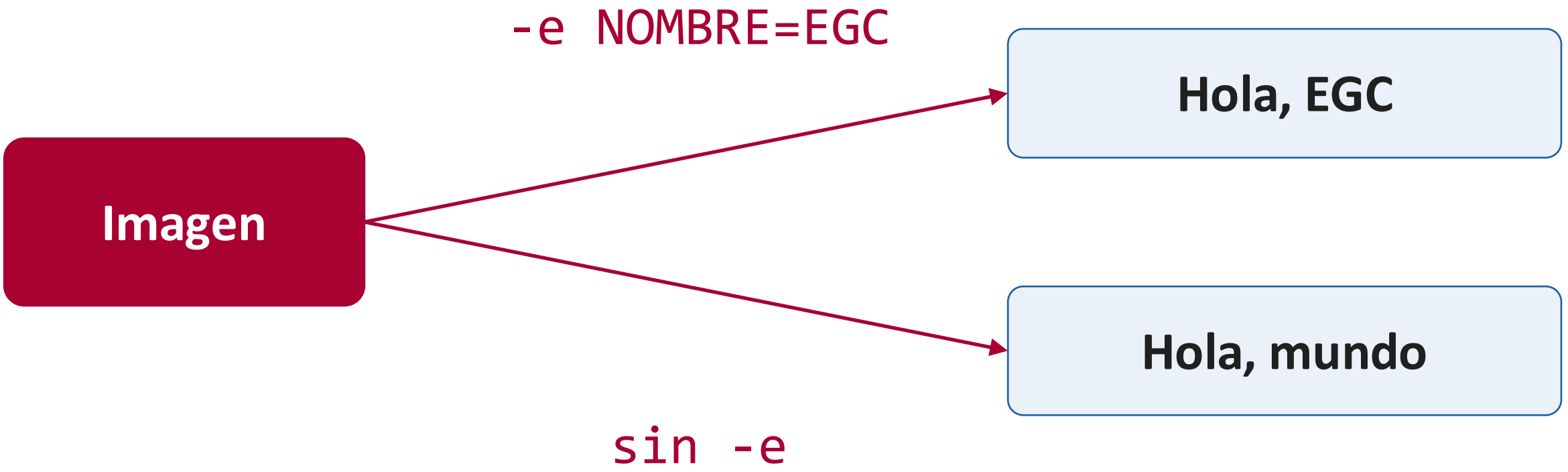
Puertos: -p anfitrión:contenedor



Sin -p, desde fuera no se llega al contenedor

3. Contenedores

Configurar desde fuera: variables de entorno



Así se le dice a MariaDB qué base y qué usuario crear

3. Contenedores

Contenedores en la nube: Kubernetes



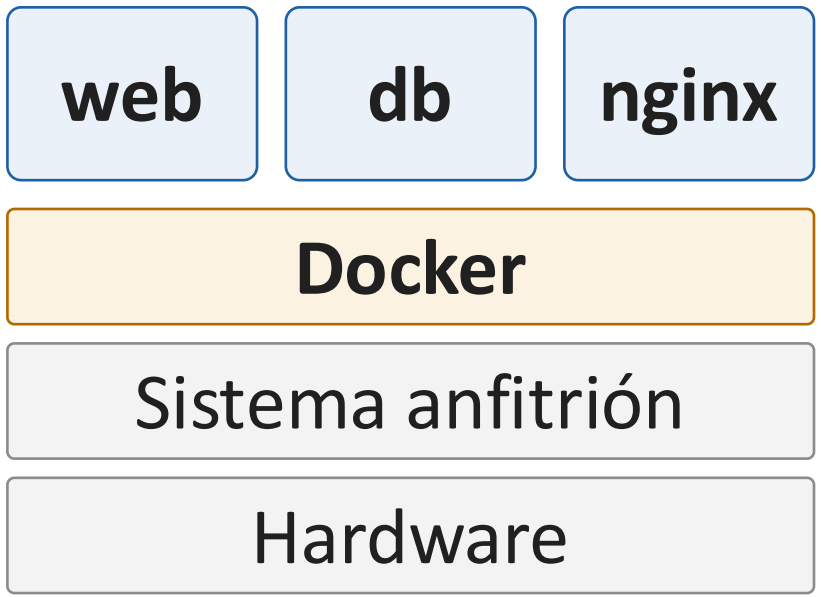
Kubernetes es una plataforma que organiza y gestiona automáticamente contenedores (como los de Docker), asegurando que las aplicaciones se ejecuten, escalen y se mantengan funcionando sin intervención manual.

Kubernetes: del griego κυβερνήτης, en español “timonel” o “gobernante”

3. Contenedores

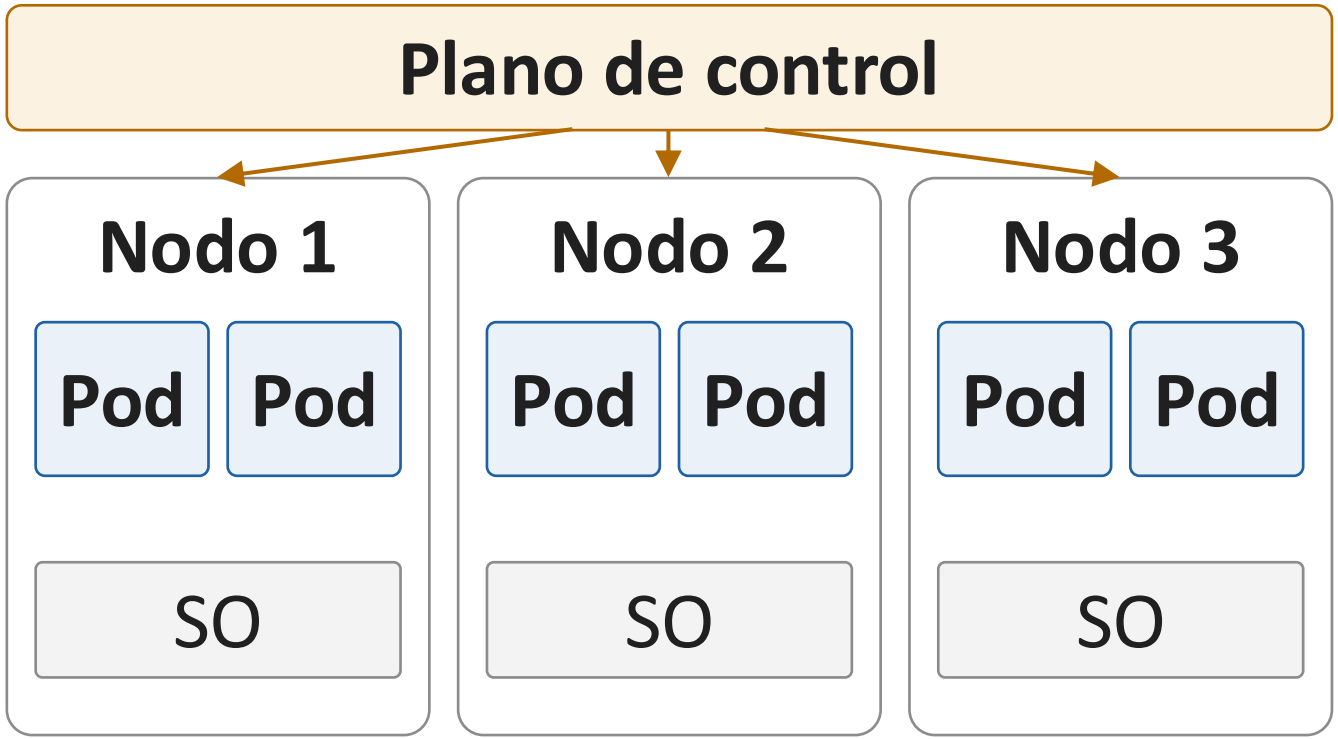
Contenedores en la nube: Kubernetes

Un host con Docker



Si cae, cae todo

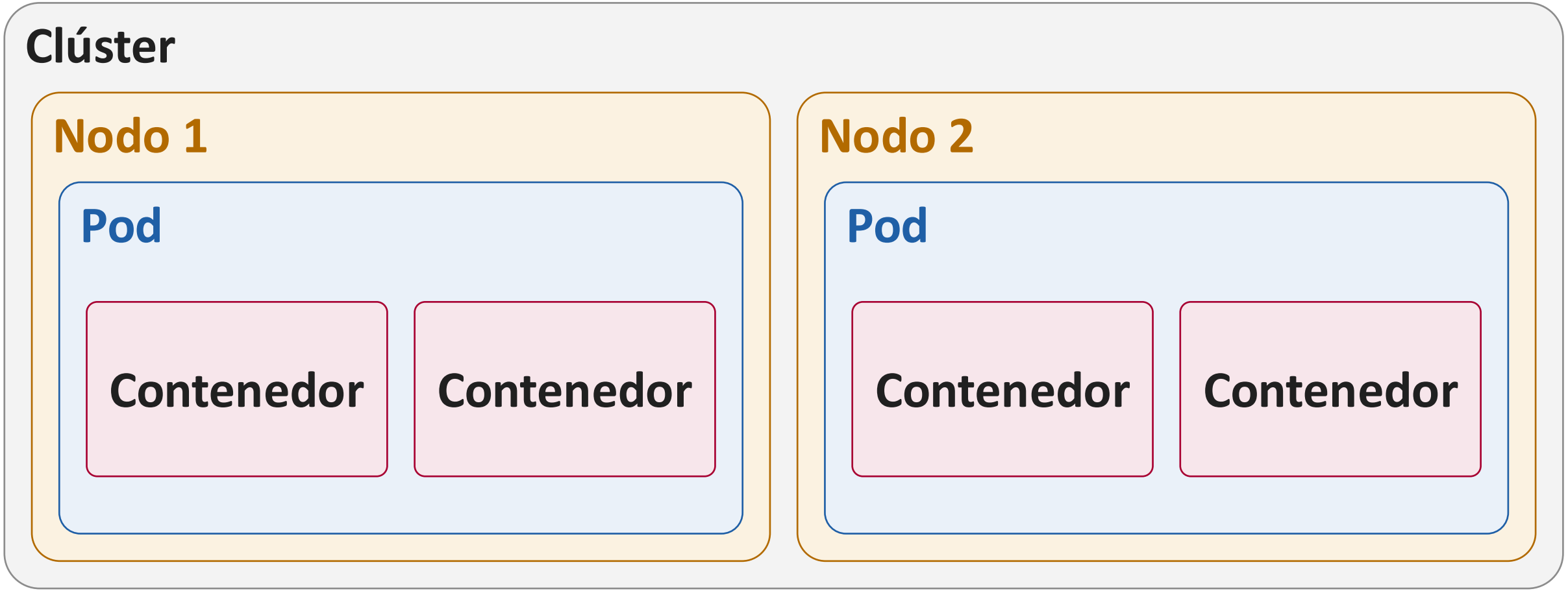
Clúster de Kubernetes



Si cae un nodo, los pods renacen

3. Contenedores

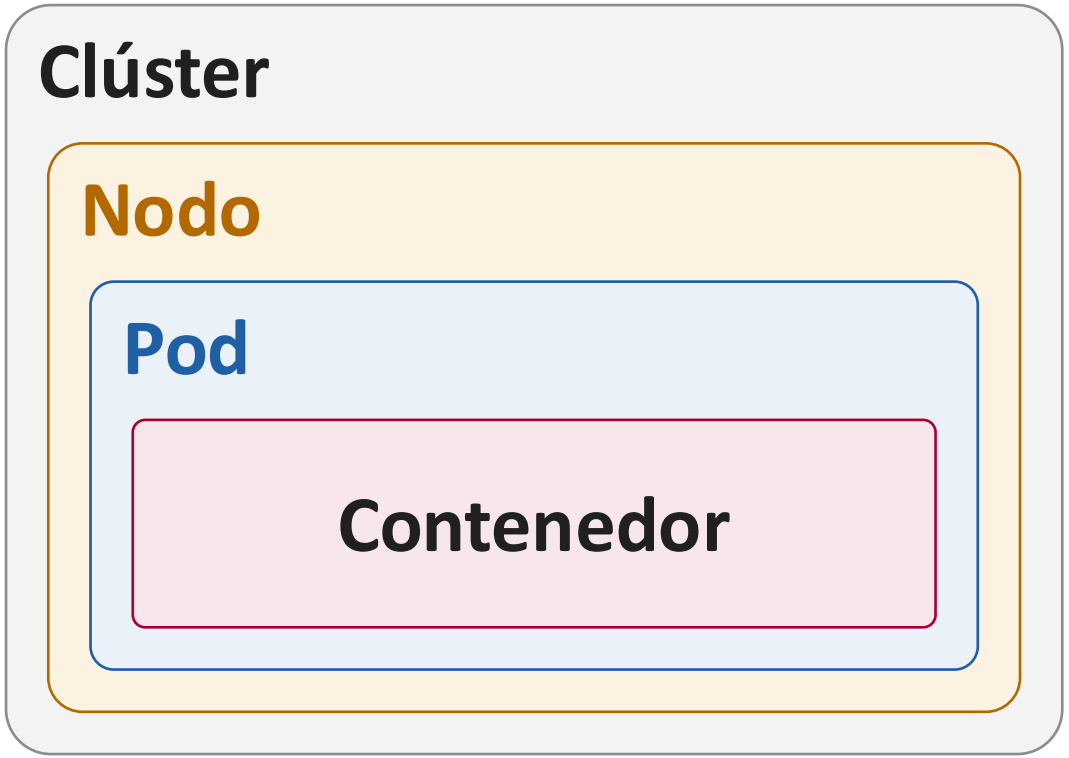
Contenedores en la nube: Kubernetes (jerarquía)



Pod: contenedores que comparten red y disco

3. Contenedores

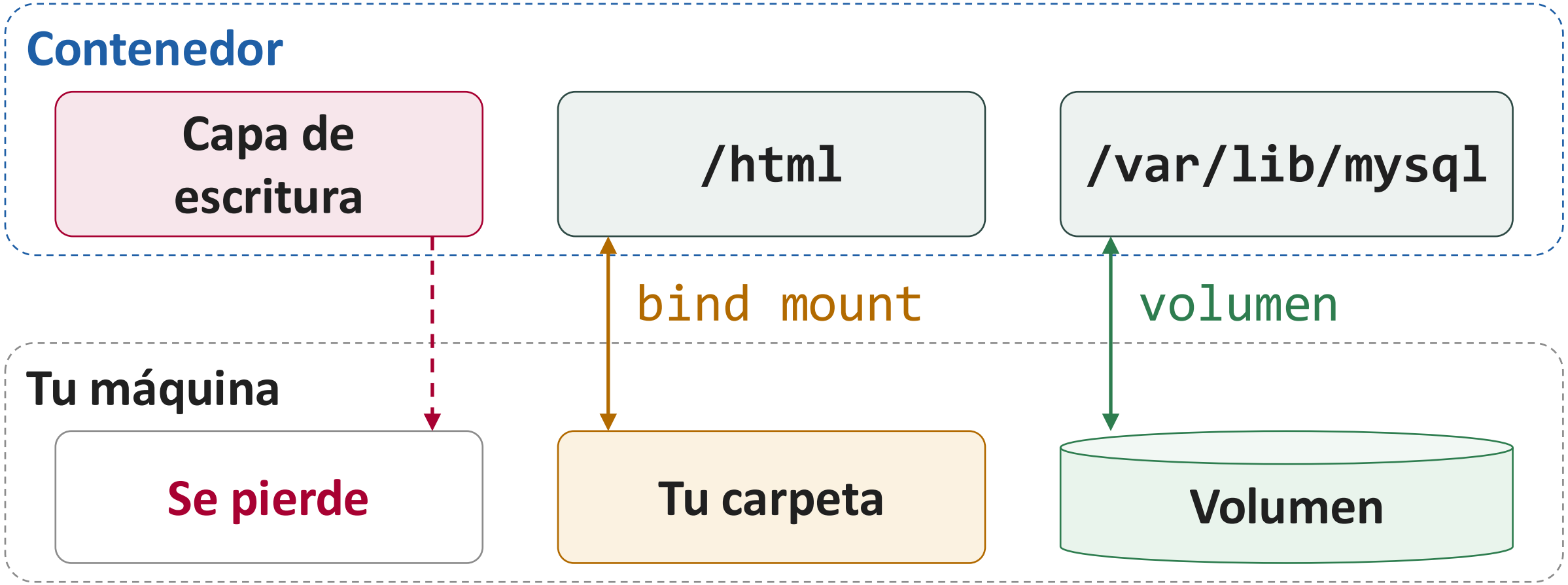
Contenedores en la nube: Kubernetes (jerarquía)



- **Contenedor**: ejecuta una aplicación o servicio concreto
- **Pod**: agrupa uno o varios contenedores que comparten red y almacenamiento
- **Nodo**: es una máquina (física o virtual) donde se ejecutan los pods
- **Cluster**: es el conjunto de nodos gestionados por Kubernetes que trabajan juntos para ejecutar todas las aplicaciones

1. Introducción a Docker y DockerHub
2. Imágenes
3. Contenedores
4. **Volúmenes y redes**
5. Composición de servicios: Docker Compose
6. Dev-containers
7. Y yo, ¿qué puedo hacer en mi proyecto?
8. Tutorial: *dockerizando uvlhub*

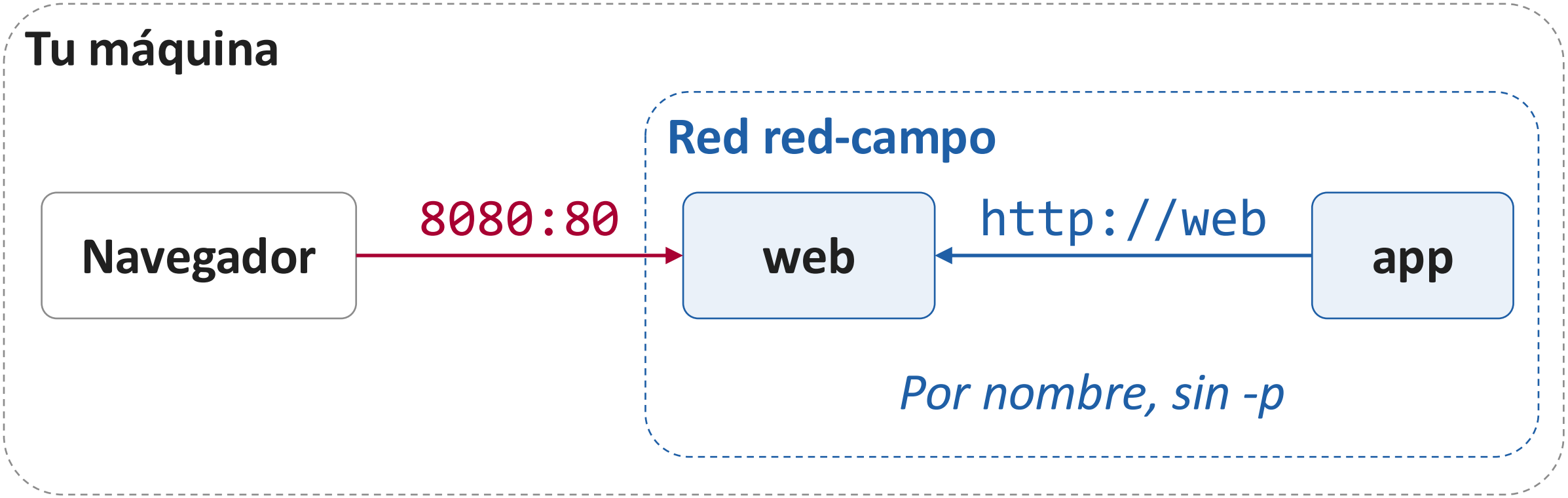
4. Volúmenes y redes



Bind mount: tu carpeta

Volumen: lo gestiona Docker y sobrevive

4. Volúmenes y redes

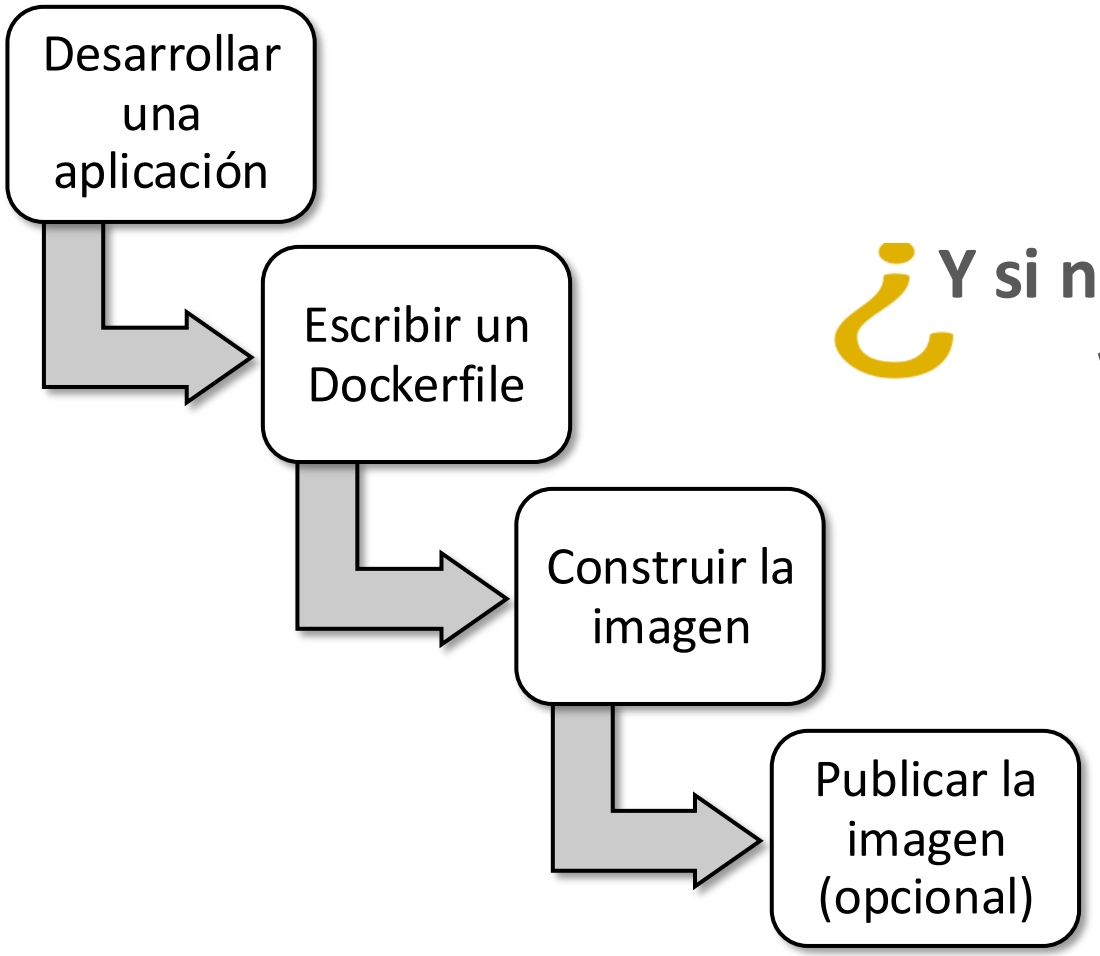


-p publica un puerto hacia fuera

Docker resuelve los nombres:
la app llega a db sin abrir el 3306.

1. Introducción a Docker y DockerHub
2. Imágenes
3. Contenedores
4. Volúmenes y redes
5. **Composición de servicios: Docker Compose**
6. Dev-containers
7. Y yo, ¿qué puedo hacer en mi proyecto?
8. Tutorial: *dockerizando uvlhub*

5. Composición de servicios: Docker compose

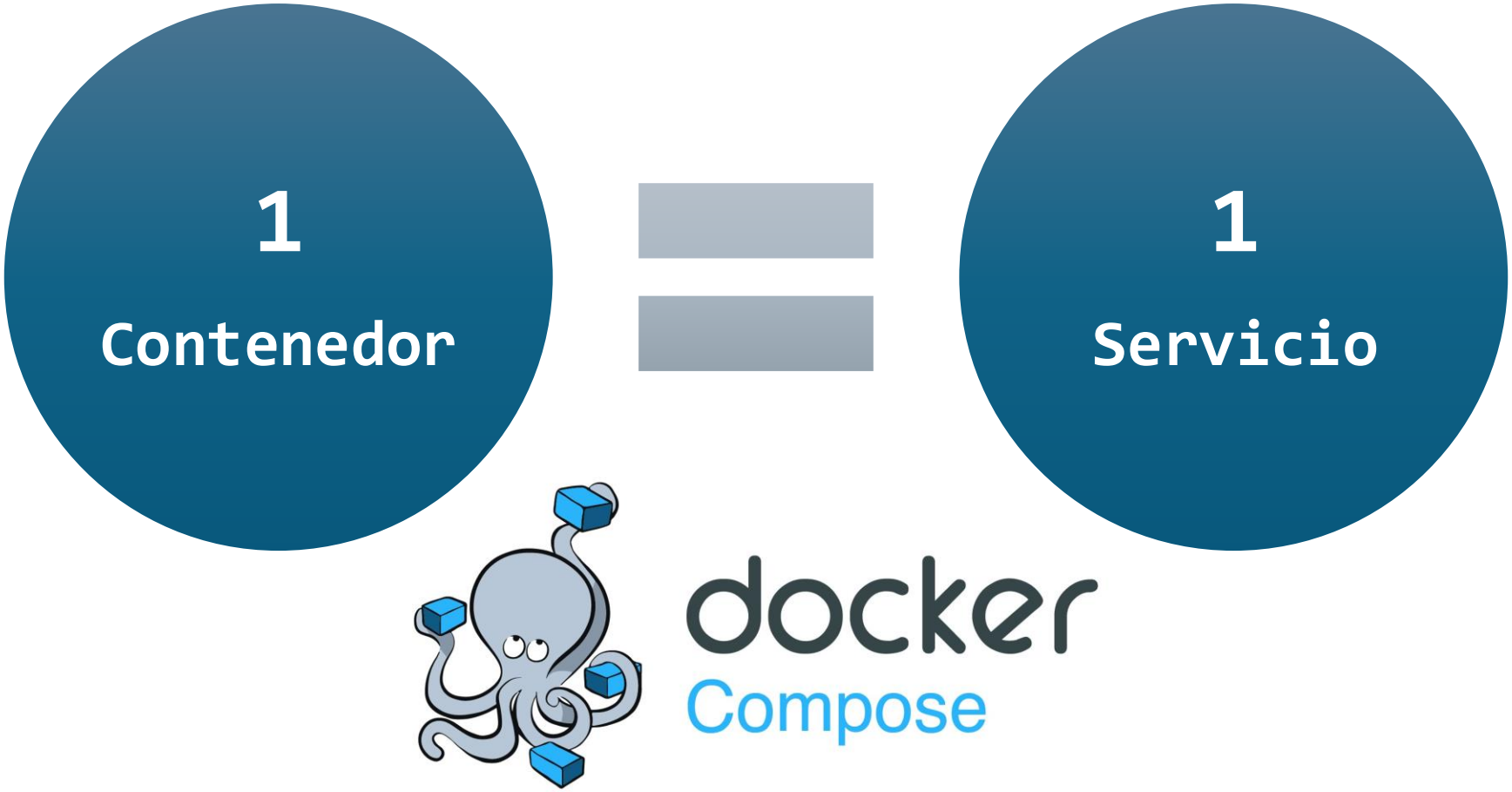


¿Y si necesito varias imágenes y varios contenedores?



5. Composición de servicios: Docker compose

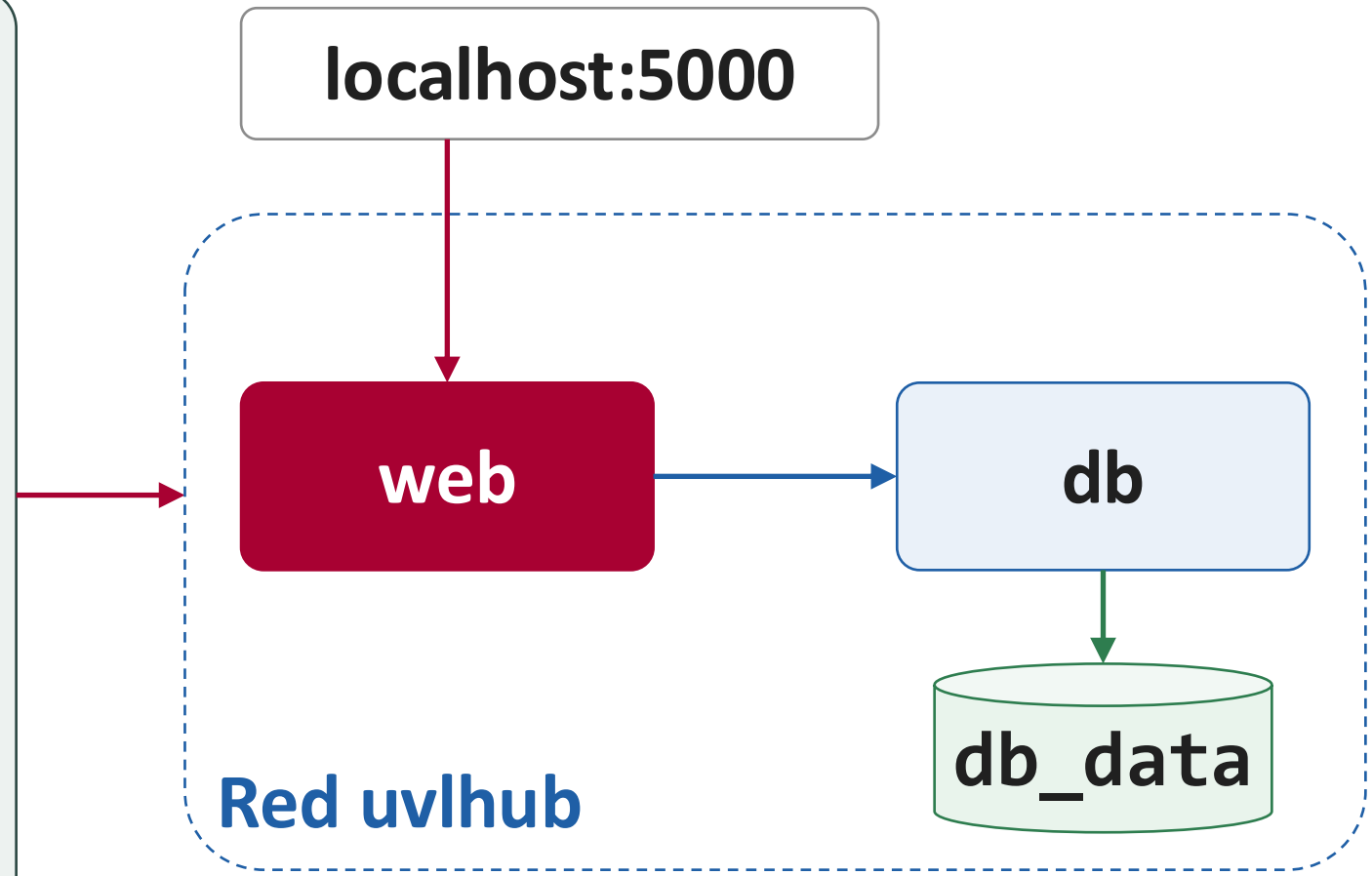
Principio de responsabilidad única



5. Composición de servicios: Docker compose

docker-compose.yml

```
services:
  web:
    build: ../
    ports: 5000
  db:
    image: mariadb
volumes: db_data
networks: red
```



Todo con una orden

1. Introducción a Docker y DockerHub
2. Imágenes
3. Contenedores
4. Volúmenes y redes
5. Composición de servicios: Docker Compose
6. **Dev-containers**
7. Y yo, ¿qué puedo hacer en mi proyecto?
8. Tutorial: *dockerizando uvlhub*

6. Dev-containers

¿Pero mi app web está en Docker y yo sigo trabajando con el código en mi host?

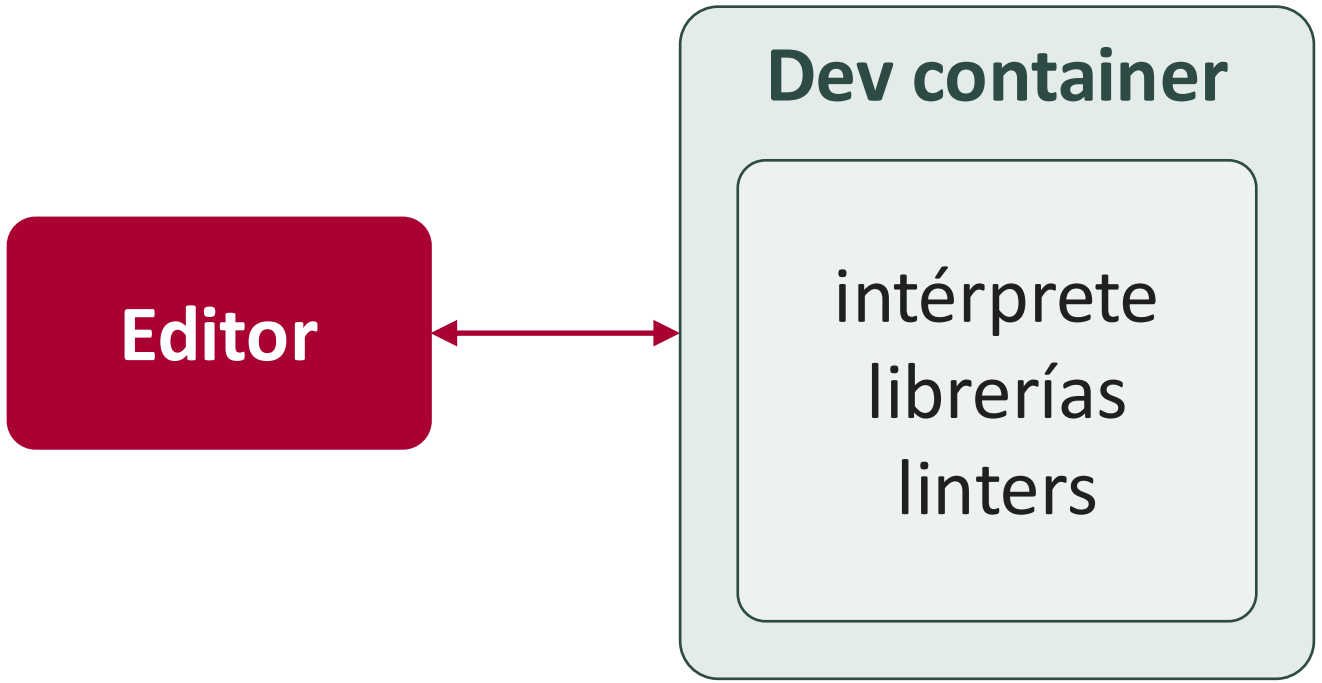
Mi VS Code no detecta las librerías al estar fuera del contenedor web…

Cada miembro de mi equipo tiene una configuración de VS Code distinta…



6. Dev-containers

Definición

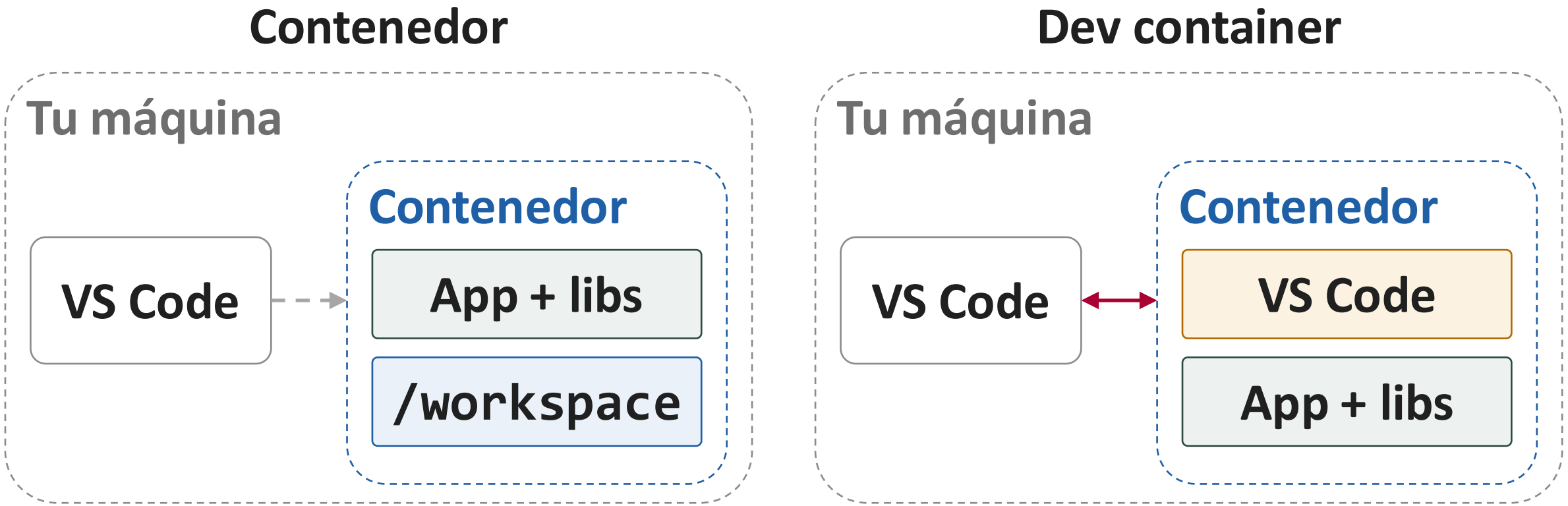


Las **herramientas de desarrollo** se ejecutan **dentro de un contenedor** que describe el propio proyecto.

El código sigue en tu máquina.

6. Dev-containers

Diferencia entre container y dev-container



Un dev-container nos “sitúa” dentro de un contenedor para aprovechar el aislamiento y la potencia de VS Code

6. Dev-containers

Ejemplo



Para cargar nuestro dev-container

Mac: ⌘ + Shift + P
Windows / Linux: Ctrl + Shift + P

Escribir en el cuadro de búsqueda:
Dev Containers: Reopen in Container
y elegimos abrir la raíz del proyecto

`.devcontainer/devcontainer.json`

1. Introducción a Docker y DockerHub
2. Imágenes
3. Contenedores
4. Volúmenes y redes
5. Composición de servicios: Docker Compose
6. Dev-containers
7. **Y yo, ¿qué puedo hacer en mi proyecto?**
8. Tutorial: *dockerizando uvlhub*

7. Y yo, ¿qué puedo hacer en mi proyecto?



¡Diseña tu propio despliegue en Docker!

Modifica y/o crea ***Dockerfiles*** propios

Añade nuevos contenedores, volúmenes y redes a tus ***docker-compose.yml***

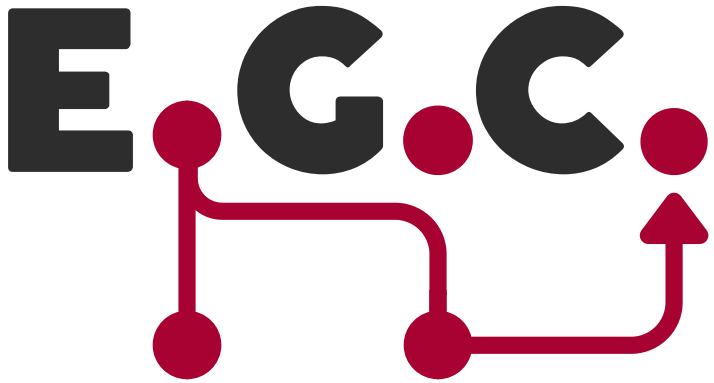
1. Introducción a Docker y DockerHub
2. Imágenes
3. Contenedores
4. Volúmenes y redes
5. Composición de servicios: Docker Compose
6. Dev-containers
7. Y yo, ¿qué puedo hacer en mi proyecto?
8. **Tutorial: *dockerizando uvlhub***

8. Tutorial: *dockerizando uvlhub*





https://egc.us.es/cursos/egc-20262027/documento/tutorial-dockerizando-uvlhub



Grado en Ingeniería Informática – Ingeniería del Software

Evolución y Gestión de la Configuración

Universidad de Sevilla 1505

Escuela Técnica Superior de Ingeniería Informática

¡Gracias!

"Pues en mi máquina funciona."
"Entonces enviaremos tu máquina."
Y así nació Docker

- Anónimo