



# RFID TRAFFIC ANALYST



**DOCKER**

**– PREGUNTAS FRECUENTES –**

**Xavi Conde**  
**Gerard Soteras**



## Índice

¿Qué son los contenedores de Docker?.....	3
¿Qué diferencias hay entre los contenedores de Docker y los LXC?.....	3
¿Cuál es la diferencia entre una imagen y un contenedor en Docker?.....	3
¿Qué sucede con los datos cuando un contenedor se elimina?.....	4
¿Cuáles son las ventajas de utilizar contenedores de docker?.....	4
Portabilidad.....	4
Ligereza y eficiencia.....	4
Facilidad de despliegue y gestión.....	4
Escalabilidad y modularidad.....	4
Seguridad y aislamiento.....	5
¿Qué tipo de aplicaciones y servicios se pueden desplegar con docker?.....	5
¿Qué otros tipos de contenedores existen además de docker?.....	5



---

## ¿Qué son los contenedores de Docker?

Los contenedores de Docker son entornos ligeros y portátiles que permiten ejecutar aplicaciones de manera aislada junto con todas sus dependencias. Funcionan como una especie de "máquina virtual", pero sin la sobrecarga de un sistema operativo completo, ya que comparten el kernel del sistema anfitrión.

## ¿Qué diferencias hay entre los contenedores de Docker y los LXC?

Los contenedores de Docker y LXC (Linux Containers) difieren principalmente en su enfoque y nivel de abstracción. Docker está diseñado para empaquetar y ejecutar aplicaciones junto con sus dependencias, proporcionando un entorno estandarizado y fácil de gestionar mediante herramientas como docker-compose. En cambio, LXC ofrece una virtualización más ligera de sistemas completos, permitiendo mayor flexibilidad y personalización, aunque requiere más configuración manual. Mientras que Docker se basa en imágenes preconfiguradas y ofrece un mayor aislamiento mediante namespaces y cgroups, LXC se asemeja más a un sistema host, brindando menos aislamiento. En resumen, Docker es ideal para la portabilidad y despliegue de aplicaciones, mientras que LXC es más adecuado para entornos que requieren una estructura similar a una máquina virtual.

## ¿Cuál es la diferencia entre una imagen y un contenedor en Docker?

**Imagen:** Es un conjunto inmutable de capas que contiene el sistema base y las aplicaciones necesarias para ejecutarse. Es como una plantilla.

**Contenedor:** Es una instancia en ejecución de una imagen. Puede tener cambios en el tiempo de ejecución, pero estos no afectan la imagen base.

### Ejemplo:

- nginx:latest es una imagen.
- Un servidor en ejecución basado en nginx:latest es un contenedor.



---

## ¿Qué sucede con los datos cuando un contenedor se elimina?

Cuando un contenedor de Docker se elimina, por defecto todos los datos almacenados dentro de él desaparecen, ya que los contenedores están diseñados para ser efímeros. Sin embargo, se puede preservar la información ya sea configurando un volumen o un bind mount.

## ¿Cuáles son las ventajas de utilizar contenedores de docker?

### Portabilidad

- Funciona igual en cualquier máquina que tenga Docker instalado, sin importar el sistema operativo.
- Se pueden compartir imágenes de Docker

### Ligereza y eficiencia

- A diferencia de las máquinas virtuales, los contenedores tienen menos consumo de recursos.
- A diferencia de las máquinas virtuales, los contenedores Docker se pueden iniciar de forma rápida y sencilla

### Facilidad de despliegue y gestión

- Gracias a Docker Compose, se puede definir y ejecutar múltiples contenedores con un solo archivo.

### Escalabilidad y modularidad

- Con Docker se puede dividir aplicaciones en múltiples servicios independientes, facilitando el escalado y mantenimiento.
- Se pueden desplegar múltiples instancias de una aplicación y balancear la carga de manera sencilla.



---

## Seguridad y aislamiento

- Al trabajar con contenedores independientes, si surgiera algún imprevisto en un contenedor, el resto de ellos no se verían afectados al igual que el sistema host.
- Se pueden definir permisos y redes personalizadas para cada contenedor y así limitar el acceso a ellos.

## ¿Qué tipo de aplicaciones y servicios se pueden desplegar con docker?

Docker permite desplegar una amplia variedad de aplicaciones y servicios en contenedores, incluyendo:

- Servidores web como Nginx o Apache
- Aplicaciones backend con Node.js, Django
- Bases de datos como MySQL, PostgreSQL o MongoDB
- Servicios de mensajería como Redis.
- También facilita la ejecución de microservicios, entornos de desarrollo aislados y aplicaciones basadas en inteligencia artificial con herramientas como TensorFlow.

## ¿Qué otros tipos de contenedores existen además de docker?

Además de Docker, existen otros tipos de contenedores y tecnologías de contenedorización como:

**Podman:** Similar a Docker, pero sin necesidad de un demonio en segundo plano y con una arquitectura más segura basada en rootless containers.

**LXC (Linux Containers):** Un enfoque más ligero que Docker, proporcionando entornos de sistema operativo completos sin el overhead de una máquina virtual.

**Kubernetes Containers:** Aunque Kubernetes no es un motor de contenedores en sí, usa contenedores como Docker o containerd para orquestar y gestionar aplicaciones a gran escala.



---

**containerd:** Un runtime ligero para la gestión de contenedores que forma la base de Docker y Kubernetes.

**CRI-O:** Un runtime diseñado específicamente para Kubernetes, optimizado para la ejecución eficiente de contenedores en clústeres.

**Singularity:** Enfocado en entornos de alto rendimiento y computación científica, permitiendo ejecutar contenedores sin necesidad de privilegios de root.