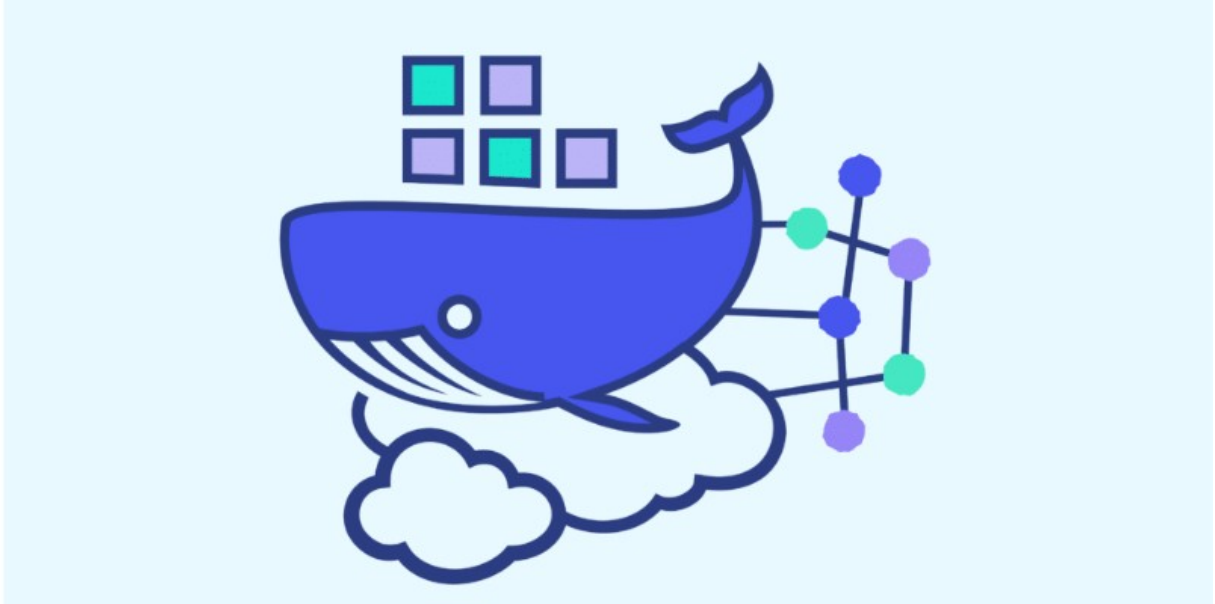


# Docker



## ¿qué es y cómo se usa?

Docker es la plataforma de contenedorización más utilizada. Los contenedores y los microservicios se utilizan cada vez más para desarrollar y ejecutar aplicaciones. Es lo que se conoce como desarrollo “cloud-native”. En este contexto, Docker se ha convertido en una solución explotada de forma masiva en el sector empresarial.

### ¿Qué es un contenedor?

Se trata de un entorno de ejecución ligero y una alternativa a los métodos tradicionales de virtualización basados en máquinas virtuales. Una de las prácticas clave en el desarrollo de software moderno es **aislar las aplicaciones** instaladas en el mismo servidor o clúster. Así se evita que interfieran entre sí.

Sin embargo, para ejecutar aplicaciones es necesario explotar paquetes, bibliotecas y otros componentes informáticos. **Las máquinas virtuales** se utilizan desde hace tiempo para explotar estos recursos y aislar a la vez una aplicación.

Estas permiten separar las aplicaciones entre sí en el mismo sistema, lo que reduce los conflictos entre componentes informáticos y la competencia por los recursos. Sin embargo, ahora ha surgido una alternativa: **los contenedores**.

Una máquina virtual es como un sistema operativo completo, de varios gigabytes de tamaño, que permite hacer particiones de los recursos de una infraestructura en lugar del contenedor que proporciona únicamente los recursos que necesita una aplicación.

El **contenedor comparte el núcleo** de su sistema operativo con otros contenedores. Esto difiere de una máquina virtual, que utiliza un hipervisor para distribuir los recursos de hardware.

Este método permite reducir la huella de las aplicaciones en la infraestructura. El contenedor alberga todos los componentes del sistema necesarios para ejecutar el código, **sin ser tan pesado como un sistema operativo completo**.

Del mismo modo, **un contenedor es más ligero y sencillo** que una máquina virtual, por lo que puede arrancar y apagarse más rápidamente. Por tanto, es más sensible y adaptable a las necesidades fluctuantes asociadas al escalado de aplicaciones.

Un último punto fuerte: a diferencia de un hipervisor, un motor de contenedor no necesita emular un sistema operativo completo. Como resultado, el contenedor ofrece **un mejor rendimiento** que la implementación en una máquina virtual tradicional.

## ¿Qué es Docker?

Docker es **una plataforma de contenedores** lanzada en 2013 que ha contribuido enormemente a la democratización de la contenedorización. Facilita la creación de contenedores y aplicaciones basadas en contenedores. Hay otras, pero esta es la más utilizada. También es más fácil de instalar y utilizar que sus competidoras.

Es **una solución de código abierto**, segura y asequible. Muchas personas y empresas contribuyen al desarrollo de este proyecto. Esta amplia comunidad está desarrollando un gran ecosistema de productos, servicios y recursos.

Inicialmente diseñada para Linux, Docker **también admite contenedores en Windows o Mac** gracias a una capa de virtualización Linux entre el sistema operativo Windows/macOS y el entorno de ejecución Docker. Por tanto, es posible ejecutar contenedores nativos de Windows en entornos de contenedores de Windows o Linux.

## ¿Cuáles son los diferentes elementos de Docker?

La plataforma Docker se basa en **varias tecnologías y componentes**. Estos son los elementos principales.

### *Docker Engine*

El Docker Engine es la **aplicación que se instala** en la máquina para crear, ejecutar y gestionar los contenedores Docker. Como su nombre indica, es el motor del sistema Docker.

Este motor es el que agrupa y enlaza los distintos componentes. Se trata de **la tecnología cliente-servidor** utilizada para crear y ejecutar contenedores, y el término Docker se utiliza a menudo para referirse al motor Docker.

Se distingue entre **Docker Engine Enterprise y Docker Engine Community**. La Docker Community Edition es la versión original, ofrecida en código abierto de forma gratuita.

**La versión Enterprise, lanzada en 2017**, añade funciones de gestión como el control de clústeres, la gestión de imágenes y la detección de vulnerabilidades.

### *Docker Daemon*

El Docker Daemon **procesa las peticiones de la API** para gestionar diversos aspectos de la instalación, como imágenes, contenedores o volúmenes de almacenamiento.

### *Docker Client*

El cliente Docker es **la interfaz principal** para comunicarse con el sistema Docker. Recibe comandos a través de la interfaz de línea de comandos y los reenvía al Docker Daemon.

### *Dockerfile*

Cada contenedor Docker comienza con un «Dockerfile». Se trata de un **archivo de texto** escrito con una sintaxis comprensible y que contiene las instrucciones para crear una imagen Docker.

Un Dockerfile **especifica qué sistema operativo** se utilizará como base para el contenedor, así como los idiomas, variables de entorno, ubicación de archivos, puertos de red y otros componentes necesarios.

### *Imágenes Docker*

Una imagen Docker es un **modelo de «solo lectura»** que se utiliza para crear contenedores Docker. Se compone de varias capas que agrupan todas las instalaciones, dependencias, bibliotecas, procesos y código de aplicación necesarios para un entorno de contenedores totalmente operativo.

Una vez escrito el archivo Dockerfile, se ejecuta **la utilidad «build»** para crear una imagen basada en este archivo. Esta imagen se presenta como un archivo extraíble que indica qué componentes de software ejecutará el contenedor y cómo lo hará.

## Contenedores Docker

Un contenedor Docker o Docker Container es **una instancia de imagen** Docker que se ejecuta en un microservicio individual o en un stack de aplicaciones completo. Cuando se lanza un contenedor, se añade una capa modificable a la imagen. Esto se utiliza para **almacenar cualquier cambio** realizado en el contenedor durante el tiempo de ejecución.

### *Docker run*

La utilidad «run» de Docker es **el comando utilizado para lanzar** un contenedor. Cada contenedor es una instancia de una imagen.

Los contenedores están diseñados para ser temporales, pero pueden **detenerse y reiniciarse en el mismo momento**. Se pueden ejecutar simultáneamente varias instancias de la misma imagen.

### *Registro Docker*

El registro Docker es **un sistema de catalogación** que permite el alojamiento y el “push and pull» de imágenes Docker. Puedes utilizar tu propio registro local o uno de los múltiples servicios de registro alojados por terceros como Red Hat Quay, Amazon ECR, Google Container Registry.

El **Docker Hub es el registro oficial** de Docker. Es un directorio SaaS para gestionar y compartir contenedores. En él se pueden encontrar imágenes Docker de proyectos de código abierto y proveedores de software. Puedes descargar estas imágenes y compartir las tuyas propias.

Un registro Docker organiza las imágenes en **diferentes directorios de almacenamiento**. Cada uno de ellos contiene diferentes versiones de una imagen Docker que comparten el mismo nombre de imagen.

## La historia de Docker

Docker Inc fue fundada por Solomon Hykes, Kamel Founadi y Sebastien Pahl durante el grupo de incubación de startups Y Combinator Summer 2010. La empresa se puso en marcha en 2011.

También fue una de las **12 startups de la primera promoción de Founder's Den**. El proyecto fue iniciado por Solomon Hykes en Francia, como proyecto interno de la empresa para una plataforma como **servicio dotCloud**.

En 2013, Docker se presentó al público en Santa Clara como parte de **PyCon**. El software se lanzó con código abierto en marzo de 2013. En aquel momento, se utilizaba LXC como entorno de ejecución predeterminado, antes de ser sustituido un año más tarde con la **versión 0.9 de Docker** por su propio componente libcontainer escrito en lenguaje Go.

A lo largo de los años, Docker ha forjado una **serie de alianzas estratégicas** con los gigantes del Cloud y las TI: Red Hat en 2013, Microsoft, IBM y Amazon Web Services en 2014, Oracle en 2015, así como Cisco, Google y Huawei.

Desde 2016, Docker se puede utilizar de forma nativa en Windows 10. Ese mismo año, un análisis de LinkedIn reveló que el número de menciones del software en los perfiles de los usuarios había aumentado un 160 %.

## ¿Cómo funciona Docker?

Docker se basa en el **núcleo de Linux** y en funciones del núcleo como cgroups y namespaces. Estas funciones permiten separar los procesos para que puedan **ejecutarse de forma independiente**.

De hecho, el propósito de los contenedores es ejecutar varios procesos y aplicaciones por separado. Esto optimiza el uso de la infraestructura **sin reducir el nivel de seguridad** en comparación con los sistemas separados.

Todas las herramientas de contenedores, como Docker, están asociadas a un **modelo de despliegue** basado en imágenes. Con este modelo es más fácil compartir una aplicación o un **conjunto de servicios** entre varios entornos.

Además, Docker permite automatizar el desarrollo de aplicaciones en un entorno de contenedores. Gracias a estas distintas herramientas, los usuarios pueden acceder a todas las aplicaciones, acelerar el despliegue, controlar las versiones y asignarlas.

## ¿Qué es la orquestación de contenedores?

Docker facilita la coordinación del comportamiento entre contenedores y la conexión entre ellos para crear stacks de aplicaciones. Para simplificar el proceso de desarrollo y testeado de aplicaciones multicontenedor, **Docker ha creado Docker Compose**.

Se trata de una **herramienta de línea de comandos**, similar al cliente Docker, que utiliza un archivo de descripción con un formato específico para agrupar aplicaciones de varios contenedores y ejecutarlas en un único host.

Cuando una aplicación está lista para instalarse en Docker, es **necesario poder abastecer**, configurar, ampliar y supervisar los contenedores en la arquitectura de microservicios.

Esto se consigue utilizando sistemas de orquestación de contenedores de código abierto como **Kubernetes, Mesos o Docker Swarm**. Estos sistemas proporcionan las herramientas necesarias para gestionar clústeres de contenedores.

En concreto, estas soluciones permiten **distribuir recursos entre contenedores**, añadir o eliminar contenedores, gestionar interacciones entre contenedores, supervisar su estado y equilibrar la carga entre microservicios.

## ¿Qué es Docker Desktop?

Docker Desktop es la aplicación nativa para ordenador diseñada por Docker para Windows y Mac. Es la forma más sencilla de ejecutar, construir, depurar y probar aplicaciones dockerizadas.

Incorpora características clave como ciclos de prueba rápidos, **notificaciones de cambios** en los archivos, compatibilidad con la red de la empresa y flexibilidad total a la hora de elegir proxies y VPN.

La aplicación **Docker Desktop** incluye herramientas para desarrolladores, Docker App, Kubernetes y sincronización de versiones. Permite crear imágenes y plantillas eligiendo idiomas y herramientas.

Las principales ventajas son la velocidad, la seguridad y la flexibilidad. Existe una **edición Community gratuita** y una **edición Enterprise** de pago con funciones adicionales de seguridad, gestión, orquestación y administración.

Existen dos versiones diferentes de Docker Desktop. La versión Stable ha sido probada exhaustivamente y puede utilizarse para desarrollar aplicaciones fiables. Las actualizaciones se publican en paralelo a las del Docker Engine.

Sin embargo, la versión Edge incluye **nuevas funciones experimentales** del Docker Engine. Por tanto, existe el riesgo de que se produzcan bugs, fallos del sistema y otros **problemas técnicos**. Sin embargo, esta versión permite probar las nuevas características por adelantado.

### *Instalar un servidor web en un contenedor Docker*

También es posible instalar un servidor web Apache dentro de un contenedor Docker. Como recordatorio, el **servidor web Apache** es una herramienta de **código abierto** para crear, desplegar y gestionar servidores web.



Sus muchas características incluyen un mecanismo de autenticación, soporte de base de datos, scripting server-side y compatibilidad con **diferentes lenguajes de programación**.

La capacidad de soportar grandes volúmenes de tráfico con una configuración mínima es una de las principales ventajas de Apache. Es compatible con Linux, macOS y Windows. Las empresas lo utilizan como alojamiento virtual o compartido.

## Las ventajas de Docker

Docker ofrece **una serie de ventajas** que permiten desarrollar aplicaciones fáciles de montar, mantener y trasladar. Los contenedores permiten aislar las aplicaciones entre sí y del sistema subyacente.

**También permiten la portabilidad**, ya que las aplicaciones no tienen que estar atadas al sistema operativo del host. Las aplicaciones en contenedores, por ejemplo, pueden transferirse fácilmente de sistemas locales a entornos en la nube.

Además, la contenedorización con Docker permite intercambiar los componentes de un stack de aplicaciones. Por último, los contenedores **simplifican la orquestación y el escalado**.

## *¿Quién utiliza Docker?*

Docker es una herramienta que **beneficia tanto a desarrolladores como a administradores de sistemas**. A menudo se encuentra en el centro de los procesos DevOps.

Los desarrolladores pueden **centrarse en el código**, sin tener que preocuparse por el sistema en el que se ejecutará. Además, pueden ahorrar tiempo incorporando programas prediseñados para sus aplicaciones.



Fuente: DataScientest "Docker: ¿qué es y cómo se usa?" 9/08/2023

<https://datascientest.com>