

Contenedores, ¿qué es un contenedor?

Artículo

[Ricardo Paiva](#) · Sep 4, 2019



Lectura de 4 min

Contenedores, ¿qué es un contenedor?

¡Hola a tod@s!

Con el [lanzamiento de la plataforma de datos InterSystems IRIS](#), podemos ofrecer nuestro producto incluso en un contenedor Docker. Pero... ¿qué es un contenedor?

La principal definición de un contenedor es que se trata de un entorno protegido para un proceso.

Los contenedores son paquetes definidos por un software, que tienen algunas similitudes con las máquinas virtuales (MV), como por ejemplo, que pueden ejecutarse.

Los contenedores proporcionan aislamiento sin la necesidad de emular completamente al sistema operativo (SO). Por lo tanto, los contenedores son mucho más ligeros que una MV.

En esencia, los contenedores son una respuesta a la cuestión de cómo trasladar de forma fiable una aplicación desde un sistema/entorno a otro y garantizar que funcione. Cuando todas las dependencias de la aplicación se encierran dentro de un contenedor y crean un espacio para aislar un proceso, hay más posibilidades de garantizar que la solución de la aplicación se ejecutará al moverse entre las plataformas.

El sistema operativo nos permite ejecutar procesos. Estos procesos comparten espacios de direcciones, namespaces, grupos de control (cgroups)... y, en general, tienen acceso a todo el entorno del sistema operativo, a su programación y también los administran. Todo esto es positivo. Sin embargo, ¿qué pasaría si quisiéramos aislar un proceso en particular o un número de procesos para ejecutar una tarea, operación o servicio específicos? Los contenedores nos brindan la capacidad de aislar un proceso. Por lo tanto, podríamos definir un contenedor como el entorno protegido para un proceso.

Entonces ¿qué es un entorno protegido? Es un grado de aislamiento en el cual un contenedor guarda sus procesos. Esta característica se implementa mediante la función central de Linux llamada namespaces (https://en.wikipedia.org/wiki/Linux_namespaces), la cual también permite que otras partes importantes del sistema se aislen, como la interfaz de red, los puntos de montaje, comunicaciones entre procesos (IPC) y tiempo compartido universal (uts).

El contenedor o entorno protegido también puede dirigirse o controlarse mediante otra función central llamada grupos de control o cgroups (<https://en.wikipedia.org/wiki/Cgroups>). Las reglas que damos a los contenedores sirven para garantizar que el contenedor sea un buen vecino cuando comparta recursos con otros contenedores y con el servidor.

Para entender cuál es la diferencia entre un contenedor y una MV, podríamos utilizar la analogía de que una MV es como una casa, mientras que el contenedor es como un *piso*.



Las máquinas virtuales son autónomas e independientes como una casa. Todas las casas tienen su propia infraestructura: tuberías, calefacción, electricidad, etc. Además, una casa tiene requerimientos mínimos (al menos 1 dormitorio, 1 techo, etc.)

En cambio, los contenedores se construyeron para aprovechar una infraestructura compartida y por ello podemos compararlos con un piso. El complejo de pisos comparte las tuberías, la calefacción, el sistema eléctrico, la entrada principal, los ascensores, etc. De la misma forma, los contenedores aprovechan los recursos disponibles del servidor mediante el kernel de Linux. También, hay que considerar que los pisos son de diferentes tamaños y formas.

Debido a que los contenedores no tienen un sistema operativo completo, sino únicamente las características mínimas para acoplarse a Linux, como algunos ejecutables en `/bin`, algunos archivos de configuración y definición en `/etc` y algunos otros archivos, estos pueden ser de tamaño pequeño, lo cual los hace más rápidos al momento de moverlos o ejecutarlos en segundo plano. Esto se traduce en mayor velocidad a partir del momento en que se construyen, desde el canal de suministro de la fábrica de software hasta la ejecución final en producción. Por cierto, los contenedores se ajustan perfectamente al contexto de la arquitectura de microservicios CI/CD, pero esa es otra historia.

Los procesos en el contenedor están estrechamente relacionados con el ciclo de vida del contenedor. Cuando *inicializo* un contenedor, normalmente quiero que todos los servicios de mi aplicación se ejecuten y funcionen correctamente (por ejemplo, piense en el puerto 80 para el contenedor de un servidor web y en los puertos 57772 y 1972 para un contenedor de InterSystems IRIS). Cuando *detengo* el contenedor, también se detienen todos los procesos.

Lo que describí en esta publicación es la noción fundamental del tiempo de ejecución de un contenedor, es decir, un entorno protegido que aísla sus procesos del servidor y de otros contenedores.

Contenedores, ¿qué es un contenedor?

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Hay otra parte para entender mejor el trabajo que realizan los contenedores, el cual hace referencia a sus [imágenes](#). Pero esto lo veremos en una siguiente publicación.

[#Administración del sistema](#) [#Contenedorización](#) [#Docker](#) [#Nube](#) [#InterSystems IRIS](#)

10 1 0 0 84

Mensajes relacionados

- Contenedores, ¿qué es un contenedor?
- [Contenedores, ¿qué es un contenedor basado en IMÁGENES?](#)

Log in or sign up to continue

Añade la respuesta

URL de fuente: <https://es.community.intersystems.com/post/contenedores-%C2%BFqu%C3%A9-es-un-contenedor>