

La mejor guía de contenedores: Docker, Kubernetes y más

La mejor guía de contenedores: Docker, Kubernetes y más. Los contenedores son para todos.

Aprende a comenzar con ellos. Al trabajar con software, innumerables compañías se han quejado del hecho de que las máquinas virtuales necesarias para ejecutar diferentes partes del mismo aumentan considerablemente sus costos.

Recientemente, muchos comenzaron a usar un método llamado «Containerization», que agrupa la aplicación junto con todos sus archivos de configuración, bibliotecas y dependencias relacionadas.

Entonces, ¿por qué las grandes empresas, como IBM y Google, están tan entusiasmadas con este enfoque?

En pocas palabras, la creación de contenedores ayuda a las empresas a reducir los costos generales, hace que el software sea más portátil y permite escalarlo mucho más fácilmente.

En este artículo, repasaremos:

- ¿Por qué la tecnología de contenedores se está volviendo tan popular?
- Una breve historia de la contenedorización.
- Una visión general de Docker y Kubernetes
- ¿Quién debe usar contenedores?
- Las mejores soluciones de contenedores

Contenedorización

vs.

Virtualización

Muchas personas intentan poner la Contenedorización frente a la virtualización como si fueran enfoques completamente diferentes, que no es exactamente la mejor manera de ver las cosas. La Contenedorización en realidad resolvió todas las deficiencias de la virtualización completa , por lo que se ha ganado el favor de los desarrolladores de todo el mundo.

Entonces, para comenzar, echemos un vistazo a algunas [diferencias entre las dos tecnologías](#):

Virtualización

Una vez que se instala un componente de software llamado hipervisor en el servidor, puede crear máquinas virtuales (VM), que actúan como servidores separados que ejecutan sus propios sistemas operativos. Este enfoque generalmente termina costando mucho dinero ya que el [hipervisor](#) y todas las máquinas virtuales a menudo pueden ocupar alrededor del 20% del rendimiento de un servidor.

Contenedorización

La contenedorización es la virtualización del núcleo del sistema operativo . Mientras los contenedores ejecutan procesos aislados, comparten un sistema operativo común, archivos binarios y bibliotecas . Esto no usa casi tanto espacio y memoria como la VM tradicional, lo que reduce los costos generales considerablemente.

Los contenedores son mucho más rápidos de crear y activar porque, a diferencia de las máquinas virtuales, no necesita instalar un sistema operativo cada vez que configura uno. La implementación también es mucho más fácil ya que los contenedores están aislados de su entorno, lo que hace que su implementación en un entorno diferente sea mucho más simple.

Historia de los contenedores: 1970-ahora

Muchos de nosotros comenzamos a escuchar sobre contenedores en 2013 cuando apareció un software emergente de código abierto llamado Docker que trajo contenedores a la esfera pública. Pero, la idea de los contenedores es en realidad mucho más antigua que eso.

En los años 70 , el sistema operativo UNIX salió con un comando «chroot» (cambio de raíz) , que permitía a los usuarios aislar los sistemas de archivos.

20 años después, FreeBSD modificó esta idea para crear un comando de cárcel , que permite a los administradores dividir los sistemas informáticos en sistemas más pequeños (o «cárceles»).

En 2005, el sistema operativo Solaris creó contenedores que aislaron las aplicaciones para que pudieran ejecutarse de forma independiente sin potencialmente afectarse entre sí.

Un año después, los ingenieros de Google crearon la tecnología cgroups (grupos de control) , que luego fue absorbida por el kernel de Linux, lo que llevó a la aparición de la virtualización «ligera» LXC .

Pero, todas estas soluciones aún necesitaban personas con amplios conocimientos para configurarlas manualmente, por lo que no obtuvieron mucha tracción entre los expertos.

Docker fue el primero en resolver este problema en 2013. Usaron contenedores de Linux, agregaron más funcionalidades (como imágenes de contenedores portátiles) y los hicieron más fáciles de administrar.

Y así, los contenedores de Docker se convirtieron casi en un éxito de la noche a la mañana.

¿Quiere su propio contenedor?

Docker : # 1 en tecnología de contenedores

Si bien hay algunas opciones para elegir, [Docker](#) es, sin lugar a dudas, la tecnología de contenedores más extendida hoy en día. Hoy, el 83% de todos los contenedores se ejecutan en Docker . ¿Pero qué es Docker?

Docker es un software que permite a los usuarios encapsular aplicaciones, bibliotecas, archivos de configuración, archivos binarios y otros archivos dependientes en un contenedor para que puedan ejecutarse en cualquier entorno sin problemas.

Pero Docker no se detiene allí, el software también gestiona el contenedor, desde la creación hasta el final de su ciclo de vida.

Si bien es importante mencionar que existen otras tecnologías de contenedores (LXC, Open-VZ, CoreOS, etc.), no son tan conocidas como Docker porque requieren expertos para configurarlas y mantenerlas.

Términos de Docker que debe saber

(No se preocupe, no habrá una prueba después).

Un software comprimido e independiente creado por Dockerfile. Básicamente es una plantilla (aplicación más bibliotecas y binarios requeridos) necesaria para construir un contenedor Docker en ejecución. Las imágenes se pueden usar para compartir aplicaciones en contenedores.

Dockerfile

Un archivo de texto que contiene todos los comandos e instrucciones necesarios para construir un Docker Image

Docker Run

Un comando que se utiliza para iniciar los contenedores Docker. Cada contenedor es una instancia en ejecución de una imagen.

Docker Hub

Docker Hub es un repositorio basado en la nube donde puede almacenar, administrar, descargar y compartir imágenes de contenedores (tanto de Docker como de la comunidad).

Docker Engine

La tecnología central que crea y ejecuta contenedores Docker.

Kubernetes y otros orquestadores

Administrar contenedores manualmente en un solo servidor es bastante difícil, pero ¿qué pasa si sus contenedores están distribuidos en cientos de servidores? Imposible, ¿verdad? No para un orquestador como Kubernetes.

Un orquestador es una herramienta que administra las redes de contenedores, incluido el lanzamiento, el mantenimiento y el escalado de aplicaciones en contenedores.

Los orquestadores han facilitado significativamente la vida de muchas personas, especialmente los desarrolladores de microservicios. Llevan a cabo tareas como asegurarse de que la capacidad del servidor se usa de manera eficiente, lanza servicios basados en las necesidades, detiene las versiones antiguas de contenedores y activa las nuevas, y más.

Existen bastantes herramientas de orquestador de contenedores diferentes, pero, al igual que con las tecnologías de contenedores, hay una que se destaca del paquete: Kubernetes, un orquestador de código abierto con una amplia variedad de opciones que se pueden personalizar para satisfacer sus necesidades.

Para mostrarle por qué Kubernetes es el claro ganador, echemos un vistazo a algunas alternativas.

Docker Swarm

El orquestador de Docker, Docker Swarm, es más simple que Kubernetes, pero fue diseñado principalmente para grupos de contenedores más pequeños . Si bien Kubernetes requiere un poco más de trabajo para implementar, es una solución mucho más completa aprobada por Docker que le brinda una infraestructura de aplicación resistente y fácil de administrar.

Mesos

Mesos, un proyecto de Apache, no es solo una herramienta de orquestación, es esencialmente un sistema operativo en la nube que coordina los componentes en contenedores y no en contenedores. Muchas plataformas pueden ejecutarse simultáneamente en Mesos, incluidos Kuberenetes.

Microservicios y Contenedores

L
O
S
M
I
C
R
O
S
E
R
V
I
C
I
O
S



son una forma de dividir las aplicaciones en módulos más pequeños para facilitar su desarrollo e implementación. Si bien este enfoque aún permite que estas partes divididas se comuniquen entre sí, los cambios en un módulo no afectan a otros módulos.

Si bien el concepto de microservicios se desarrolló independientemente de la idea de los contenedores, la tecnología de contenedores permitió a las personas utilizar la técnica de microservicios. Por ejemplo, el orquestador del contenedor gestiona un gran número de contenedores y realiza copias de seguridad antes de una actualización para facilitar el regreso a una versión anterior.

¿Pueden los contenedores ayudarme?

Ahora que sabemos que usar contenedores puede tener bastantes beneficios, pero es posible que se pregunte si esta tecnología puede ayudarlo.

Para simplificar la respuesta: cualquiera que trabaje con software puede beneficiarse del uso de la tecnología de contenedores.

Echemos un vistazo a algunas formas en que los contenedores pueden ayudarlo:

Los contenedores hacen que el software sea súper portátil.

Dado que el software está incluido en un contenedor, puede ejecutarse en cualquier máquina sin problemas .

Es más seguro . El software aislado no tendrá un impacto negativo en otros sistemas.

Software fácilmente escalable . Ya sea que el número de usuarios aumente o disminuya, los contenedores le permiten reaccionar de inmediato.

Facilitando el escalado

Dado que los contenedores son tan livianos, generalmente solo demoran unos segundos en iniciarse (en comparación con los minutos que se tarda en ejecutar una VM). Dado que este proceso también divide la aplicación en partes más pequeñas, los contenedores también nos permiten utilizar los recursos del servidor de manera mucho más eficiente. Los contenedores también pueden escalar microservicios varias veces en lugar de una sola, haciendo que el proceso sea mucho más rápido.

Transición a contenedores

Si bien es posible encapsular una aplicación monolítica gigante en un contenedor, no obtendrá todos los beneficios que ofrecen los contenedores. Los contenedores utilizan los recursos del servidor de manera más eficiente cuando las aplicaciones se dividen en partes más pequeñas (especialmente porque los microservicios pueden necesitar ser escalados muchas veces, y otras partes solo pueden ser escaladas una

vez).

Hay algunas maneras en que puede desglosar las aplicaciones monolíticas , pero desafortunadamente, no existe un método universal. Cuantas más dependencias de datos hay, más difícil es dividir.

¿Qué herramienta de contenedor es mejor para mí?

Dado que Docker y Kubernetes son proyectos de código abierto, puede comenzar a ejecutar contenedores por su cuenta. Pero tenga en cuenta que la transición de grandes proyectos a contenedores puede ser una tarea enorme. Si no tiene el tiempo o la experiencia para hacerlo usted mismo, le recomendamos que se comunique con una organización para configurar su solución .

El futuro de los contenedores.

Todos saben que los contenedores serán influyentes en la tecnología del futuro cercano , y ahora que los grandes jugadores como Google e IBM están apostando por las herramientas de contenedores, no vemos que esto desaparezca pronto.

En 2019, el informe de Turbomic ha demostrado que el 26% de las empresas de TI ya han comenzado a usar aplicaciones en contenedores, y se espera que ese número se duplique para 2021.

Algunas compañías han comenzado a usar contenedores fuera de los servidores. Linux lanzó recientemente Flatpak , una utilidad de software basada en contenedores para la implementación de software y la gestión de paquetes. Con Flatpak, las personas pueden usar aplicaciones que están aisladas del resto del sistema, lo que fortalece la seguridad,

facilita la actualización y desalienta el desperdicio de recursos en la recopilación de datos innecesarios.

El equipo Silverview también utilizó la tecnología de contenedores cuando crearon Fedora , un sistema operativo experimental que distribuye todo el software en contenedores.

¿Quiere asegurarse de que su empresa no se quede atrás? En HostDime , podemos trabajar con usted para determinar qué solución de contenedor se adapta mejor a sus necesidades y hacer que la transición sea lo más fluida posible.

Leer también: [¿Es bueno un alojamiento web, un servidor, con Plesk Onyx? ¿Lo recomiendan?](#); [La gestión de la nube híbrida requiere nuevas herramientas y habilidades.](#)