

11 - Máquinas Virtuales y Contenedores

domingo, 18 de abril de 2021 11:43

ESCRITORIOS REMOTOS: (sirve para el trabajo desde casa)

Son programas que nos permiten acceder e interactuar con una computadora a distancia mediante una conexión a internet (no necesitan hardware, solo que tengan la misma aplicación de escritorio remoto y estén encendidas simultáneamente). Se ahorran recursos gracias a que ya no se necesitaría viajar a la oficina y no se necesitan oficinas tan grandes.

Host: es la computadora que se accede de forma remota.

Cliente: es la computadora desde la cual se va a trabajar físicamente. Varios clientes pueden acceder a un mismo host.

DESVENTAJAS:

- Si el host no tiene suficiente seguridad, puede tener ciberataques cuyo objetivo son robar las credenciales de las cuentas para utilizar el host.
- El rendimiento del sistema depende enteramente de la calidad de la conexión a internet.

Ejemplos: TeamViewer, AnyDesk, Assist, Chrome Remote Desktop, Windows Remote Desktop.

MAQUINAS VIRTUALES: software capaz de contener en su interior un sistema operativo, haciéndole creer que es una computadora de verdad. Ese SO puede a su vez albergar otro SO más.

DE SISTEMAS: emula una computadora completa y nos permite ejecutar otro SO en su interior. La máquina virtual es creada en el Hipervisor, una capa de software que se instala sobre el hardware de la computadora, y su función es asignar recursos. Hay 2 tipos de Hipervisor:

Tipo 1: es el más utilizado por ser rápido y seguro, que corre directamente sobre el hardware, y sobre el se crean una o más máquinas virtuales, las cuales funcionan como una computadora real.

Tipo 2: corre sobre un sistema operativo, y es más lento que el anterior.

DE PROCESOS: emula un proceso concreto (como una aplicación), permitiendo que se comporte de la misma manera independientemente del SO en el que se ejecute (sirve para desarrollar aplicaciones que vayan a funcionar en muchos SO).

Las máquinas virtuales sirven para:

- Ejecutar programas antiguos.
- Ejecutar aplicaciones en otros SO sin cambiar el hardware.
- Ofrecen un entorno de seguridad para analizar malware.
- Mejora el aprovechamiento del hardware.
- Permiten la virtualización del almacenamiento y redes.

DESVENTAJAS:

- Son menos eficientes porque acceden al hardware indirectamente.

CONTENEDORES: es un concepto de empaquetación de software que incluye la aplicación y todas sus dependencias de ejecución. El software, las librerías y el sistema operativo se lo "empaqueta" en un espacio virtual denominado contenedor, para que cualquier cliente con solo abrir el contenedor en cualquier SO puede ejecutar el software.

Funcionan de manera conjunta con el SO ya que no requieren de un hypervisor. Se utilizan para ejecutar aplicaciones en un entorno aislado.

Proporcionan una mayor protección para las aplicaciones que se ejecutan sobre los contenedores.

Trabajan en capas, y cada vez que modifiquemos dependencias se genera una nueva versión del contenedor con una nueva capa por encima de la anterior.

Permiten configurar el sistema fácil y rápidamente, desplegando nuestro código en menos tiempo ya que los requisitos de la infraestructura ya no están vinculados con el entorno de la aplicación.

Al proporcionar una huella más pequeña del sistema operativo, se reduce el tamaño del desarrollo.

Aumenta la productividad.

1. Crear una imagen.
2. docker build.
3. docker publish.
4. docker pull.
5. docker run.

ORQUESTADORES: herramienta que automatiza el despliegue, administración, escalamiento, comunicación y disponibilidad de nuestro software que se ejecuta en contenedores.

- Autorreparación: se pueden recuperar los contenedores que fallen, o asignar nuevos contenedores.
- Retroceso automatizado: es la capacidad que tiene el sistema de retroceder dado un error.
- Autoescalado: cuando hay picos de demanda del software, el orquestador puede habilitar más recursos.
- Balanceo de carga: si un contenedor recibe mucha demanda, el orquestador puede distribuir el tráfico de red de forma estable y balanceada.

Ejemplos: Kubernetes (de Google), Docker Swarm, Mesosphere Datacenter (DC/OS), Nomad, Amazon ECS y EKS, Azure Kubernetes Service, Google Kubernetes Engine.

- Tienen que solicitar acceso al hardware.
- Si muchas máquinas virtuales se ejecutan al mismo tiempo, el rendimiento puede verse afectado.