Clase 11: Máquinas virtuales.

Escritorios remotos.

Programas que nos permiten usar una computadora desde otro dispositivo aunque esté ubicada en otro lugar físico, a través de internet. No requieren una conexión de red física ni un hardware adicional para vincular las computadoras, deben estar conectadas a internet, poseer la app y estar prendidas.

El host es la pc a la que se accede, el cliente es el que se conecta. Varios clientes pueden acceder a un mismo host mientras este lo soporte.

TeamViewer, AnyDesk, Assit…

Puede ser inseguro dependiendo del software, y el rendimiento depende de la conexión a internet.

Máquina virtual.

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

Existen dos tipos:

* De sistema: Emula una computadora completa. Es un software que nos permite ejecutar un SO en su interior. El lugar donde la VM es creada se denomina hipervisor, este es una capa de software que se instala sobre la parte física de la pc y su función es asignar parte del hardware. Existen dos tipos de hipervisor:
  + Tipo 1: Más rápido, más seguro, más utilizado. Corre directamente sobre la parte física de la pc y sobre él se crean una o más VM.
  + Tipo 2: Corre sobre un SO y es más lento.
* De proceso: No emula la compu completa, sino un proceso concreto. Permitiendo que cada una se comporte de la misma manera sin importar el SO. Es de utilidad para desarrollar aplicaciones que se van a ejecutar en diferentes SO.

Las ventajas de utilizar máquinas virtuales son:

La virtualización fue llevada a otras áreas, como el almacenamiento o las redes.

1. Probar otros SO sin cambiar el hardware.
2. Ejecutar programas antiguos.
3. Ejecutar apps disponibles para otros sistemas.
4. Entorno de seguridad para analizar virus y malwares.
5. En servidores: Se aprovechan los recursos que de otra forma serían ociosos.

Las desventajas de las máquinas virtuales son:

1. Son menos eficientes ya que acceden al hardware de forma indirecta.
2. El rendimiento se ve afectado.

Contenedores.

Es un concepto de empaquetación de software que incluye a la app o producto que está en desarrollo y todas sus dependencias de ejecución, como las herramientas o librerías y el SO donde lo pruebo. Para que sea más fácil para el cliente probar el software.

Características:

* Configuración del sistema más sencilla y rápida.
* Se despliega el código en menos tiempo.
* Los requisitos de la infraestructura ya no están vinculados al entorno.
* **Tamaño**: tiene la capacidad de reducir el tamaño de desarrollo al proporcionar una huella más pequeña del SO.
* **Productividad**: Facilita la configuración técnica y el rápido despliegue de la app, reduce recursos ejecutando la app en un entorno aislado.
* **Gestión múltiple**: Los contenedores pueden ser agrupados, controlados por varias herramientas y se puede gestionar un grupo de host de contenedores como un único host virtual.
* **Servicios**: Lista de tareas que nos permite especificar el estado del contenedor. Cada tarea representa una instancia que debe estar en ejecución.
* **Isolación**: Los contenedores nos permiten ejecutar apps en entornos aislados,  pueden ser independientes de otros.
* **Seguridad**: La plataforma ofrece valores predeterminados seguros que ofrecen mayor protección para las apps que se ejecutan sobre ellos, estos valores pueden ser modificados.

Los contenedores funcionan de manera conjunta con el SO y no requieren de un hipervisor, por lo que son más rápidos. Trabajan en capas, cuando realizamos un cambio este se acopla arriba de los anteriores.

Para **crear un contenedor** primero debemos crear una imagen, el docker build posee las imágenes de base que pueden ser la bdd, el software en sí o el SO. Luego se realiza un docker publish para subirlo al repositorio. Docker run es para ejecutarlo.

**¿Qué es una imagen?** Es una plantilla que contiene toda la información necesaria para crear un contenedor, estas se pueden compartir. (gracias <https://teloexplicocongatitos.com/poster/tlecg16>).

Orquestadores o administradores de contenedores.

Automatiza el despliegue, administración, escalamiento, comunicación y disponibilidad de nuestro software ejecutándose en contenedores.  Ayuda a evitar tiempos de inactividad, por ejemplo si un orquestador se cae otro se inicia automáticamente.

* **Autorreparación**: Recupera los contenedores que fallan reemplazandolos o dando de baja los que no responden.
* **Retroceso automatizado**: Capacidad de retroceder, como ctrl + z.
* **Autoescalado**: Cuando se producen picos de demanda se deben ampliar los recursos.

**Balanceo de carga**: En el caso de que un contenedor reciba mucha demanda el orquestador es capaz de distribuir el tráfico de red de modo de que sea estable y balanceado.

Actividad:

