**Clase 11**

**Escritorios remotos**

Un escritorio remoto es una tecnología que permite a un usuario trabajar en una computadora a través de su escritorio gráfico desde otro dispositivo terminal ubicado en otro lugar.

Programas que nos permiten acceder e interactuar con una computadora a distancia, a través de una conexión a internet, permitiendo que podamos trabajar desde donde sea.

Estos programas no requieren de una conexión de red física ni un hardware adicional para vincular las computadoras.

**Lo que necesitamos es:**

* Acceso a internet
* Que ambas computadoras tengan la misma aplicación de escritorio remoto
* Que amabas permanezcan encendidas de manera simultánea.

La computadora a la que se accede de forma remota se la llama *HOST*, la computadora desde la que se trabaja físicamente la llamamos *CLIENTE*.

Varios clientes pueden acceder a un mismo host si este cuenta con la capacidad suficiente para soportar todas las conexiones simultaneas.

**Ventajas**:

* Ahorro de numerosos recursos, como espacios físicos de trabajo desde el lado de las empresas, o ahorro en tiempo y transporte de los empleados a lugares de trabajo.

**Desventajas**:

* Si el programa que brinda el servicio de escritorio remoto no cuenta con la seguridad necesaria, puede ser objeto de ciberataques.
* El rendimiento del sistema depende enteramente de la calidad de conexión a internet. Si no es confiable puede comprometer a todo el sistema.

Existen diversas aplicaciones, pagas, gratuitas, con diferentes prestaciones y distintos niveles de complejidad en la instalación.

**Máquinas virtuales:**

Software capaz de contener en su interior un sistema operativo haciéndole creer que es una computadora de verdad. Este sistema operativo, puede albergar a su vez, otro más.

Existen 2 tipos de máquinas virtuales: *de sistemas, y de procesos.*

**Máquina virtual de sistemas:**

Emula una computadora completa, un software que nos permite ejecutar otro sistema operativo en su interior. Es como si estuviese viendo otra computadora completamente.

El lugar donde la máquina virtual es creada se llama ***hipervisor*** = capa de software que se instala sobre la parte física de la computadora y su función es asignar parte de la memoria, disco rígido, CPU, y otros recursos físicos. A su vez existen dos hipervisores: *el tipo 1 y el tipo 2.*

**Hipervisor tipo 1**: más rápido y seguro. Corre directo sobre la parte física de la computadora y sobre él se crearán 1 o más máquinas virtuales.

**Hipervisor tipo 2**: corre sobre un sistema operativo y es más lento que el anterior.

Sobre el hipervisor podemos crear cuantas máquinas virtuales queramos, y cada una funciona como una computadora real.

**Máquina virtual de Procesos**: no emula una computadora completa sino solo un proceso concreto, como por ejemplo una app, permitiendo que cada una se comporte de la misma manera independientemente del sistema operativo sobre el que se ejecute, lo que puede ser de gran utilidad al momento de desarrollar app que van a ejecutarse en distintos sistemas operativos.

**Ventajas de las máquinas virtuales:**

* Podemos probar otros sistemas operativos sin tener que cambiar nada de nuestra computadora.
* Ejecutar programas antiguos.
* Usar apps disponibles para otros sistemas operativos.
* Entorno de seguridad para analizar virus y malwares.
* Mejora el aprovechamiento del equipo físico al utilizar los recursos que, de otra forma, estarían oseosos. Porque generalmente nunca se llegan a utilizar todos los recursos de un servidor físico al mismo tiempo
* Permitió llevar la virtualización a otras áreas como almacenamiento o las redes.

**Desventajas de las máquinas virtuales:**

* Son menos eficientes que las máquinas reales por que acceden al hardware de manera indirecta ya que el software se ejecuta sobre el sistema operativo de la máquina virtual.
* Tiene que solicitar acceso al hardware de la máquina física ralentizado el proceso.
* Cuando varias máquinas virtuales ejecutan la misma máquina física el rendimiento se puede ver afectado si la computadora carece de los recursos suficientes.

**Contenedores:**

Un contenedor es un concepto de empaquetación de software que incluye la aplicación y todas sus dependencias de ejecución.

**La configuración**: la mejor característica de los contenedores es que podemos configurar el sistema fácilmente y también más rápido. Es posible desplegar nuestro código en menos tiempo y esfuerzo con la ayuda de contenedores. Los requisitos de la infraestructura ya no están vinculados con el entorno de la aplicación, ya que se puede utilizar en una amplia variedad de entornos.

**El tamaño**: al proporcionar una huella más pequeña del sistema operativo a través de contenedores, un contenedor tiene la capacidad de reducir el tamaño del desarrollo.

**La productividad**: utilizar contenedores equivale a aumentar la productividad. Esto facilita la configuración técnica y el despliegue rápido de la aplicación. Además, ayuda a ejecutar la aplicación en un entorno aislado y reduce los recursos.

**Gestión múltiple**: existen herramientas de programación y clustering para contenedores. Algunos contenedores exponen una web y otros ofrecen API como su front end, que nos permite utilizar varias herramientas para controlarlo. Además, nos ayuda a controlar un clúster de hosts contenedores como un único host virtual.

**Los servicios**: La lista de tareas que nos permite especificar el estado del contenedor dentro de un cluster y los servicios. Básicamente, cada tarea representa una instancia de un contenedor que debe estar en ejecución y que puede ser programada sobre los nodos (cada instancia que lo ejecuta).

**La isolación**: los contenedores se utilizan para ejecutar aplicaciones en un entorno aislado (isolado). Lo mejor de esta característica de los contenedores es que aquí cada contenedor es independiente de otro y además, nos permite ejecutar cualquier tipo de aplicación requerida.

**La seguridad**: los contenedores proporcionan configuraciones por defecto que ofrecen una mayor protección para las aplicaciones que se ejecutan sobre ellos y a través de orquestadores. La plataforma establece valores predeterminados seguros, al tiempo que deja los controles en manos del administrador para cambiar las configuraciones y las políticas según sea necesario.

**Orquestadores:**

Los orquestadores son sistemas de automatización del despliegue, ajuste de escala y manejo de aplicaciones en *contenedores.*

Usar contenedores es muy fácil y ventajoso, pero hay entornos en los que necesitamos que no existan tiempos de inactividad, por lo que, si un contenedor se cae, otro debe iniciarse automáticamente, y lo mejor es que esto sea implementado por un orquestador.

**Características principales de los orquestadores**:

* **Auto reparación**: un orquestador puede recuperar los contenedores que fallen, ya sea que tenga que reemplazarlos, o dar de baja los que no respondan.
* **Retroceso automatizado**: capacidad de retroceder aplicado a nivel de sistema.
* **Auto escalado y auto reinicio de contenedores**: cuando se producen picos de demanda se requieren muchos mas recursos de computación.
* **Balanceo de carga**: en caso de que un contenedor reciba mucha demanda, el orquestador es capaz de distribuir el trafico de red de modo que sea estable y balanceado.
* **Mantenimiento de parámetros “secretos” y configuraciones**.
* **Intercambio de datos y networking**.

**Tipos de orquestadores:**

* **Kubernetes**: motor de orquestación de contenedores más popular que existe en el mercado. La herramienta funciona agrupando contenedores que componen una aplicación en unidades lógicas para una fácil gestión y descubrimiento.
* **Docker** **Swarm**: es la solución que propone Docker ante los problemas de los desarrolladores a la hora de orquestar y planificar contenedores a través de muchos servidores. Viene incluido junto al motor de Docker y ofrece muchas funciones avanzadas integradas -como el descubrimiento de servicios, balanceo de carga, escalado y seguridad-.
* **Mesosphere DC/OS:** el sistema operativo Mesosphere Datacenter (DC/OD) es una plataforma de código abierto, integrada para datos y contenedores desarrollados sobre el kernel de sistema distribuido Apache Mesos. Se ha diseñado para gestionar múltiples máquinas dentro de un centro de datos con uno o más clústeres, ya sea en la nube o usando software en servidores en local. DC/OS puede desplegar contenedores y gestionar tanto aplicaciones sin estado como protocolos con estado en el mismo entorno. Es capaz de funcionar con Docker Swarm y Kubernetes.
* **HashiCorp Nomad**: soportada por Linux, Mac y Windows, Nomad es una herramienta binaria única capaz de planificar todas las aplicaciones virtualizadas en contenedores o independientes. Nomad ayuda a mejorar la densidad, a la vez que reduce costos, ya que es capaz de distribuir de manera eficiente más aplicaciones en menos servidores.
* **Amazon ECS**: el servicio de AWS es un sistema de gestión muy escalable que permite a los desarrolladores ejecutar aplicaciones en contenedores. Está formado por muchos componentes integrados que permiten la fácil planificación y despliegue de clústeres, tareas y servicios del contenedor.
* **Amazon Elastic Kubernetes Service**: facilita la implementación, administración y escalado de aplicaciones en contenedores mediante Kubernetes en AWS. Ejecuta la infraestructura de administración de Kubernetes por el usuario en varias zonas de disponibilidad de AWS para disminuir errores. Las aplicaciones que se ejecutan en cualquier entorno estándar de Kubernetes son totalmente compatibles y pueden migrar fácilmente a Amazon EKS.
* **Azure Kubernetes Service (AKS):** el servicio de Azure es código abierto y está optimizado para su uso en las máquinas virtuales de Azure. Proporciona las herramientas necesarias para crear, configurar y gestionar la infraestructura de contenedores Docker abiertos. AKS ofrece desarrollo simplificado de aplicaciones basadas en contenedores y despliegue con soporte para Kubernetes, Mesosphere DC/ OS o Swarm para la orquestación.
* **Google Kubernetes Engine (GKE):** montado sobre Kubernetes, permite desplegar, gestionar y escalar aplicaciones de contenedores en la nube de Gooogle. Su objetivo es optimizar la productividad del departamento de desarrollo al mejorar la gestión de las cargas de trabajo basadas en contenedores. Oculta tanto las tareas de gestión simple como aquellas más complejas detrás de herramientas de línea de comando, usando interfaces transparentes y fáciles de usar.

**Servicios de orquestación más usados en la actualidad:**

* **Amazon Elastic Container Service (ECS):**  servicio proporcionado por Amazon Web Services (AWS) para ejecutar contenedores Docker a escala en su infraestructura.
* **Azure Container Instance (ACI):** servicio básico de orquestación de contenedores proporcionado por Microsoft Azure.
* **Azure Service Fabric:** orquestador de contenedores de código abierto proporcionado por Microsoft Azure.
* **Kubernetes:** herramienta de orquestación de código abierto, iniciada por Google, parte del proyecto Cloud Native Computing Foundation(CNCF).
* **Marathon:**  marco (framework) para ejecutar contenedores a escala en Apache Mesos.
* **Nomad:** orquestador de contenedores proporcionado por HashiCorp.
* **Docker Swarm:** orquestador de contenedores proporcionado por Docker, Inc. Es parte de Docker Engine.