

1. VIRTUALIZACIÓN

La virtualización es un conjunto de técnicas de hardware / software que permiten abstraer el hardware y software real y simular recursos físicos, sistemas operativos,...

La virtualización suele implementarse en instituciones, empresas, escuelas,... porque es fácil de implementar y tiene muchas ventajas.

Un ejemplo de virtualización son las máquinas virtuales. También hay otros ejemplos de virtualización, como la emulación de videojuegos.

2. DEFINICIÓN DE MÁQUINA VIRTUAL

A veces, necesitamos probar un nuevo sistema operativo, una configuración o un programa en un entorno limpio. ¿Cómo podemos hacerlo fácilmente? Usando máquinas virtuales.

Una máquina virtual permite simular una computadora (con su propio SO) y ejecutar programas, configuraciones, etc.

Las máquinas virtuales se crean utilizando un programa de virtualización que se ejecuta en la parte superior del sistema operativo de una máquina real.

Hay muchos ejemplos de máquinas virtuales, cada una con sus propias características: VirtualBox, VMware, VirtualPC, Parallels, JavaVM, .NET,...

3. TIPOS DE MÁQUINAS VIRTUALES

Según su funcionalidad, podemos clasificar las máquinas virtuales en dos tipos

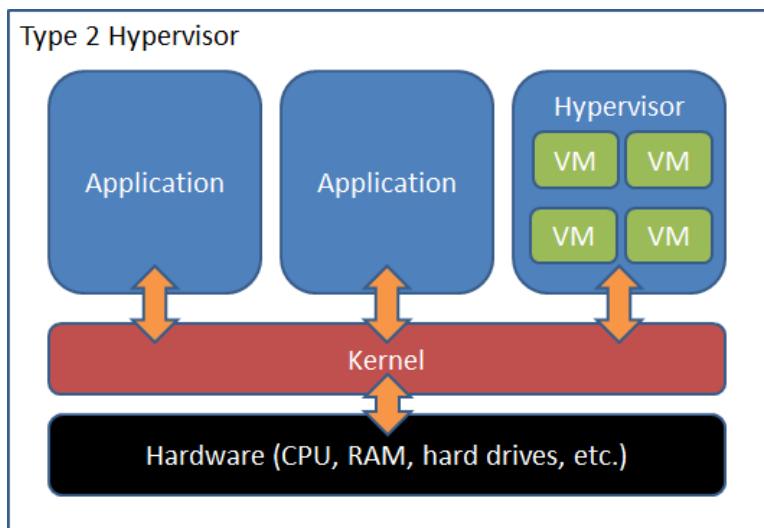
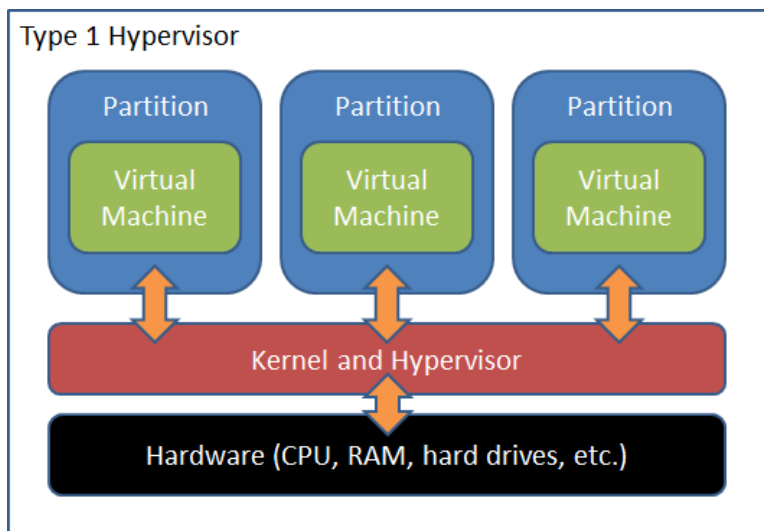
- Máquinas virtuales del sistema (SVM)
- Procesar máquinas virtuales (PVM)

3.1 Máquinas virtuales del sistema (SVM)

Permitieron replicar la máquina real en varias máquinas virtuales, cada una con su propio SO. El software que realiza la virtualización se llama "hipervisor".

Hay dos tipos de hipervisor:

- Hipervisor de tipo 1 (el hipervisor se ejecuta directamente en el hardware)
- Ejemplos de hipervisor de tipo 1: VMware ESXi (gratis), VMware ESX, Xen (libre), Citrix XenServer (gratis), Microsoft Hyper-V Server (gratis).
- Hipervisor de tipo 2 (el hipervisor se ejecuta como una aplicación en un sistema operativo host)



Hay dos tipos de hipervisor:

- Hipervisor de tipo 1 (el hipervisor se ejecuta directamente en el hardware)
- Ejemplos de hipervisor de tipo 1: VMware ESXi (gratis), VMware ESX, Xen (libre), Citrix XenServer (gratis), Microsoft Hyper-V Server (gratis).

- Hipervisor de tipo 2 (el hipervisor se ejecuta como una aplicación en un sistema operativo host)
- El hipervisor se ejecuta en un sistema operativo "host" y crea máquinas virtuales con un sistema operativo "invitado".

- Desventaja: el sistema operativo host comparte recursos con el sistema operativo invitado.
- Ejemplos de hipervisor tipo 2: VirtualBox (gratis), VMware Workstation, VMware Player (gratis), QEMU (gratis).

3.2 Procesar máquinas virtuales (PVM)

Este tipo de máquinas virtuales se ejecuta como una aplicación normal dentro de un sistema operativo host y admite un solo proceso. Se crea cuando se inicia ese proceso y se destruye cuando sale. Su propósito es proporcionar una programación independiente de la plataforma.

Algunos ejemplos:

- Máquina virtual Java: el compilador Java genera códigos JavaBytecodes y la máquina virtual Java ejecuta esos códigos JavaByte en cada sistema operativo donde existe.
- Wine: ejecuta aplicaciones de Windows en un sistema operativo que no sea Windows.

4. VENTAJAS DE LAS MÁQUINAS VIRTUALES

Ventajas de las máquinas virtuales:

- Utilice varios sistemas operativos al mismo tiempo.
- Pruebe un sistema operativo antes de instalarlo en una máquina real.
- Utilice aplicaciones que no estén disponibles en su sistema operativo host.
- Emular un tipo diferente de computadora (con otro conjunto de instrucciones).
- Cree entornos de prueba.
- Guarde el estado actual y restáurelo más tarde.
- Ahorre energía, recursos, espacio,... emulando equipos antiguos.
- Fácil de clonar o realizar copias de seguridad.
- Ecológico (evite construir y destruir computadoras, componentes, ...)

5. DESVENTAJAS DE LAS MÁQUINAS VIRTUALES

Desventajas de las máquinas virtuales:

- Comparta recursos con otras máquinas virtuales y con el sistema operativo host.
- El rendimiento es inferior al de las máquinas reales

6. CÓMO CONFIGURAR UNA MÁQUINA VIRTUAL: VIRTUALBOX

VirtualBox es un hipervisor de tipo 2 gratuito disponible para el sistema operativo más popular. Está disponible en <https://www.virtualbox.org/>

En nuestro sitio web Moodle, proporcionamos videos que explican cómo instalar y configurar VirtualBox y sus máquinas virtuales.

También hay varios tutoriales disponibles en Internet que explican cómo hacerlo:

- Cómo instalar VirtualBox en Ubuntu: <https://www.youtube.com/watch?v=QkJmahizwO4>
- Cómo instalar VirtualBox en Windows 7: <https://www.youtube.com/watch?v=7WdjSBZ794Q>
- ¿Cómo instalar VirtualBox en Windows 10: <https://www.youtube.com/watch?v=OjBQC81oXqc>
- Configure una máquina virtual en VirtualBox:
https://www.youtube.com/watch?v=H_ustCy4Ks8

Para mejorar su rendimiento, VirtualBox le permite instalar en su sistema operativo invitado una aplicación llamada "Adiciones de invitado". Es muy útil y recomiendo instalarlo.

- Instale "Adiciones de invitados" en Ubuntu:
<https://www.youtube.com/watch?v=Q84boOmiPW8>
- Instale "Adiciones de invitados" en Windows 7:
<https://www.youtube.com/watch?v=vi53zKcvgXU>
- Instale "Adiciones para invitados" en Windows 10:
https://www.youtube.com/watch?v=Bb_kJd3lSxQ

7. MÁQUINAS VIRTUALES VS CONTENEDORES

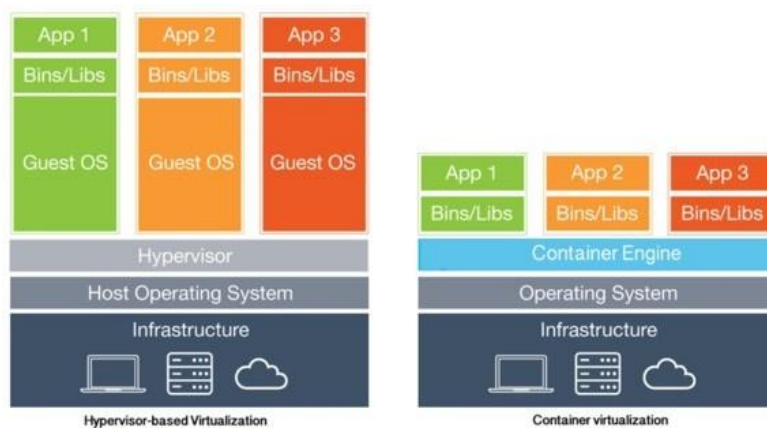
Las máquinas virtuales consisten en un hipervisor que funciona en un hardware físico simulando uno o varios "hardware falso". Cada uno de esos "hardware falso" permite emular una máquina virtual (cada una con su propio sistema operativo).

La virtualización ha tenido mucho éxito en los últimos años en el despliegue de aplicaciones y aprovisionamiento de infraestructura, pero ahora el paradigma que triunfa son los “Contenedores Ligeros” o simplemente “Contenedores”.

Los “contenedores” son una tecnología similar a las máquinas virtuales pero con la gran diferencia de que en lugar de usar un hipervisor, existe un único sistema operativo de host compartido para los contenedores.

Cada contenedor es realmente un "entorno privado" del sistema operativo host. Cada contenedor comparte recursos con el sistema operativo host, sin virtualizar el hardware.

Gracias a ello, el sistema es más rápido (no pasa por la capa de virtualización) y el tamaño de los contenedores es mucho menor, haciéndolos eficientes, más fáciles de migrar, arrancar, recuperar, pasar a la nube, ...



7.1 ¿Contenedores o máquinas virtuales?

En resumen, es útil ejecutar un contenedor si queremos:

- Obtenga un mejor rendimiento que una máquina virtual clásica.
- Desarrollar una aplicación que se pueda distribuir sin problemas de configuración ni dependencia en el mismo sistema operativo que nuestro host.
- Desarrolle una aplicación fácil de portar a la nube con casi cualquier cambio.
- Ejecute varias copias de la misma aplicación (conjunto de aplicaciones que funcionan juntas).

Si queremos flexibilidad (por ejemplo, usar un sistema operativo diferente al sistema operativo host) o ejecutar múltiples aplicaciones diferentes en diferentes sistemas operativos, debemos usar máquinas virtuales.

8. CONTENEDORES: DOCKER

Existen tecnologías que nos permiten utilizar contenedores Linux como LXC o LXD. Son populares, pero el sistema de contenedores más popular es Docker.

Más información en <https://unix.stackexchange.com/questions/254956/what-is-the-difference-between-docker-lxd-and-lxc>

Docker es un proyecto de código abierto que automatiza la implementación de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de la virtualización a nivel del sistema operativo en Linux.

Docker usa características de aislamiento de recursos del kernel de Linux:

- cgroups: grupos de control, es una característica que limita las cuentas y aísla el uso de recursos como CPU, memoria, E / S de disco, red, etc.) de una colección de procesos.
- espacios de nombres: limita los recursos que un conjunto de procesos puede ver.

Esto permite que "contenedores" independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

Podemos comunicarnos con contenedores Docker vinculando entrada / salida desde una consola al contenedor.

(comandos "docker attach" o "docker run -it") y con comandos como "docker cp" para copiar archivos. Por lo tanto, Docker no es compatible con la interfaz gráfica. Sin embargo, es posible manejarlos con interfaz gráfica con algunas de estas soluciones:

- Instalar un servidor X y conectarse con un cliente XWindows al contenedor Docker.
- Utilizando un software de administrador remoto como VNC (recomendamos instalar NoVNC en nuestro contenedor Docker. Con esta opción no requerimos un cliente especial y podemos operar directamente desde el navegador).

Más información: [https://es.wikipedia.org/wiki/Docker_\(software\)](https://es.wikipedia.org/wiki/Docker_(software))

8.1 Ventajas de Docker

Algunas de las ventajas de Docker:

- Independencia de la plataforma: permite el uso de contenedores en cualquier sistema compatible: puede ser Windows, Mac, Linux, ...
- En los sistemas Windows, si tienen capacidades Linux, funciona en modo nativo. En los que no los tienen, instale una máquina virtual (en Virtual Box) y allí lanza el Kernel de Linux.
- Es muy fácil crear e iniciar un contenedor.
- Cada contenedor tiene su propio entorno de red, configurable, compartido con otros contenedores si es necesario.
- Hay imágenes de Docker. Son como "plantillas". Podemos crear tantos contenedores como queramos bajo una misma imagen. También podemos descargar imágenes de terceros y utilizarlas para crear contenedores.
- Controlamos las versiones de todo el software dentro del contenedor: sistema operativo, versión para nuestra base de datos, servidor de aplicaciones,... Esto elimina problemas de configuración al portar el sistema de una máquina a otra.
- Existen soluciones de clustering y alta disponibilidad como Kubernetes o Docker swarm que se utilizan en entornos de producción.

- Cuenta con un potente buscador de imágenes ya pregenerado en “Docker Hub” donde podemos encontrar tanto imágenes oficiales como imágenes personales compartidas por la comunidad.

Tiene soporte en los principales sistemas de la nube: Azure, AWS, Google Cloud, OVH. De hecho, cuando contratas un VPS suele ser un contenedor.

8.3 Comandos útiles de Docker

La referencia oficial de Docker es <https://docs.docker.com/reference/> Algunos comandos útiles de Docker son:

- “docker run -it image”: crea un contenedor con la imagen dada. Si esa imagen no está en la máquina, la descarga automáticamente desde Docker Hub y crea el contenedor. El parámetro “-it” vincula la entrada y la salida del contenedor a la consola actual.
- CUIDADO: si ejecutas este comando dos o tres veces... ¡¡crearás dos o tres contenedores !! Para comenzar de nuevo un contenedor, use “docker start container”.
- “docker start -i container”: inicia un contenedor creado. Si ha creado un contenedor y desea ejecutarlo nuevamente, debe usar este comando. -i parámetro vinculará la entrada del contenedor a la consola actual.
- “docker ps”: le permite ver las máquinas Docker en ejecución.
- “docker ps -a”: le permite ver todas las máquinas Docker actualmente en ejecución o no.
- “docker image ls”: te permite ver las imágenes de Docker (no confundir con contenedores) que tienes descargadas en tu máquina.
- “docker cp origen destino”: permite copiar archivo entre máquina real y Docker.
- “Docker login”: comando para iniciar sesión en “Docker hub” usando la consola.
- “docker pull urlmaquina: etiqueta”: comando para descargar una imagen de Docker.
- “Docker commit etiqueta”: comando para confirmar una etiqueta con los cambios realizados en un contenedor Docker.
- “docker push urlmaquina: etiqueta”: comando para cargar cambios en una imagen de Docker a “Docker hub”

También existe una Cheat Sheet de Docker en español con varios comandos y sus ejemplos asociados. Puede descargarlo desde <https://github.com/sergarb1/CheatSheetsCastellano>

8.4 GUI para Docker: Kitematic

Kitematic es un software para crear y administrar contenedores Docker desde una interfaz gráfica. Tiene algunas ventajas. Por ejemplo, está completamente integrado con “Docker Hub”, lo que facilita la búsqueda de una imagen de Docker y la creación de un contenedor con ella.

Está disponible en <https://github.com/docker/kitematic/releases>