

# MÁQUINAS VIRTUALES VS CONTENEDORES

Estrella Palacios, Katherine Lizbeth (2016056193)), Gonzales Cave, Angel Gabriel (2017057861)), Porlles Carrillo, Diego Armando (2015050948)), Quispe Mamani, José Luis (2015053235))

*Escuela Profesional de Ingeniería de Sistemas*

*Universidad Privada de Tacna*

*Tacna, Perú*

---

## Abstract

Each virtual machine instantiated by a hypervisor loads an image of the entire operating system into memory, but the Docker containers will share a single image in memory. This makes Docker containers lightweight and can be initialized in seconds. Each container executed a Docker image that packages the application, all its dependencies and the execution environment; Docker images apart from being versionable are very portable between the different environments of the systems development cycle, creating a simple and safe process.

---

## 1. Resumen

Cada máquina virtual instanciada por un hipervisor carga en memoria una imagen de sistema operacional completa en cambio los contenedores Docker compartirán una única imagen en memoria. Esto hace que los contenedores Docker sean ligeros y puedan ser inicializados en segundos. Cada contenedor ejecuta una imagen Docker que empaqueta la aplicación, todas sus dependencias y el ambiente de ejecución; las imágenes Docker aparte de ser versionables son muy portables entre los diferentes ambientes del ciclo de desarrollo de sistemas, creando un proceso simple y seguro.

## 2. Introducción

La mayoría de empresas utilizan múltiples servidores para llevar a cabo sus operaciones, y para la continuidad de sus operaciones tienen dos o mas servidores ejecutando los mismos servicios en diferentes servidores. Esto se debe a que no se puede garantizar que un sistema operativo funcione correctamente todos los días durante el año. Por esta razón cada servicio se separa y se deja funcionando en un host distinto, pero esta solución resulta costosa y difícil de administrar por el número de máquinas y el espacio requerido para su funcionamiento. Debido a esta problema surgieron las máquinas virtuales que permiten la ejecución de múltiples sistemas operativos llamados invitados en una sola máquina física llamada anfitrión.

## 3. Marco Teórico

### 3.1. *Máquina Virtual*

Una máquina virtual (VM) es una computadora compuesta completamente de software que puede ejecutar su propio sistema operativo y aplicaciones como si fuera una computadora física. Una VM se comporta exactamente como una computadora física y contiene su propia CPU virtual (basada en software), RAM, disco duro y tarjeta de interfaz de red (NIC).[2]

### 3.2. *Hyper-V*

Hyper-V es un hipervisor que permite la virtualización de diferentes sistemas operativos, que se ejecutan al mismo tiempo sobre un sistema físico, sin que se vean interferidos entre ellos. Esta separación de entornos se consigue mediante la creación de una capa de abstracción entre el hardware de la máquina física (también conocida como host) y el sistema operativo que se ejecuta dentro del entorno virtualizado (también conocido como máquina virtual o huésped). De este modo, los diferentes recursos de la máquina física (tarjeta de red, memoria RAM, etc.) se dividen y se reparten entre uno o más entornos virtualizados.

El hipervisor, constituye una pequeña capa de software entre el hardware y los diferentes sistemas operativos instalados en el sistema. Es el encargado de ejecutar múltiples instancias de sistemas operativos de forma aislada a través del uso de múltiples instancias de ejecución conocidas como particiones.

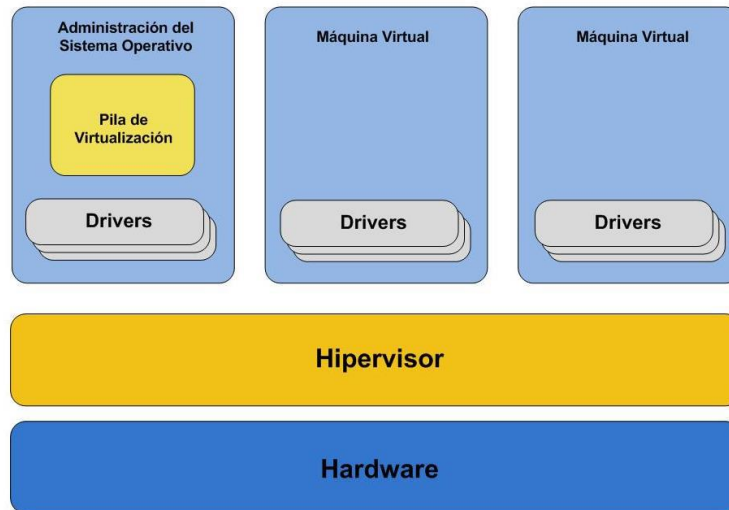


Figura 1: Hipervisor

El hipervisor se divide en dos capas diferentes. La capa inferior, corresponde a la implementación de un microkernel que soporta acceso a memoria, uso de hilos, uso de señales y un mecanismo de abstracción del hardware; mientras que la capa superior, se encarga de proporcionar los servicios de virtualización.[1]

### 3.2.1. *Beneficios*

La virtualización realizada por Hyper-V, proporciona diferentes beneficios tanto técnicos como de gestión, que se detallan a continuación:

- Mayor eficiencia de los recursos de la máquina física.
- Reducción de los costes de operación y mantenimiento de la máquina física.
- Reducción del tiempo requerido para el despliegue y configuración del hardware y software, así como la realización de las pruebas correspondientes al entorno físico sobre el que se despliega el servicio Hyper-V.
- Los recursos físicos son gestionados por Hyper-V para proporcionar entornos totalmente aislados.

- Implementación de entornos de escritorios virtuales a través de VDI.
- Posibilidad de crear entornos de testing o prueba sin que sea necesario obtener equipamiento hardware

[1]

### 3.2.2. *Limitaciones*

El servicio Hyper-V posee una serie de limitaciones en cuanto a la cantidad de recursos que puede utilizar. Del mismo modo, cada una de las máquinas virtuales gestionadas por el servicio de virtualización posee una serie de limitaciones con respecto a sus recursos de tipo físico asignados. Las limitaciones del servicio Hyper-V son las siguientes:

- Número de procesadores lógicos, 512.
- Procesadores virtuales por cada procesador lógico, sin limitación.
- Máquinas virtuales en ejecución por servidor, 1024.
- Procesadores virtuales por servidor, 2048.
- Memoria RAM, 24 TB.
- Capacidad de almacenamiento, sin limitación. Depende de la capacidad de almacenamiento del sistema operativo base.
- Redes de área de almacenamiento (SAN) virtuales, sin limitación.
- Adaptadores de red físicos, sin limitación.
- Equipos de adaptadores de red (NIC Teaming), sin limitación.
- Redes virtuales (conmutadores virtuales), sin limitación. Depende de los recursos del sistema físico.
- Puertos de los conmutadores de red virtuales por servidor, sin limitación. Depende de los recursos del sistema físico.

[1]

### 3.2.3. *Componentes Físicos Virtualizados*

El servicio Hyper-V se encarga de virtualizar los recursos físicos de los que dispone en la máquina que despliega el servicio de virtualización. Los recursos virtualizados por el servicio se indican a continuación:

- Procesadores virtuales.
- Memoria RAM.
- Discos duros virtuales.
- Adaptadores de red virtuales.
- Unidades de CD/DVD virtuales.
- Unidades de disquete virtuales.
- Adaptadores de video virtuales.
- Dispositivos de entrada de datos (ratón y teclado) virtuales.

[1]

### 3.3. *XenServer*

XenServer es la plataforma completa de virtualización de servidores de Citrix. XenServer está optimizado para servidores virtuales Windows y Linux.

XenServer se ejecuta directamente en el hardware del servidor sin requerir un sistema operativo subyacente, lo que resulta en un sistema eficiente y escalable. XenServer funciona abstrayendo elementos de la máquina física (como discos duros, recursos y puertos) y asignándolos a las máquinas virtuales que se ejecutan en él.

Usar XenServer reduce los costos :

- Consolidación de múltiples máquinas virtuales en servidores físicos

Usar XenServer aumenta la flexibilidad al:

- Aumento de la disponibilidad de máquinas virtuales mediante el uso de alta disponibilidad para configurar políticas que reinicien máquinas virtuales en otro host XenServer si falla uno.
- Mayor portabilidad de las imágenes de VM, ya que una imagen de VM funcionará en una variedad de infraestructuras de implementación

### **3.3.1. *vApps***

Una vApp es un grupo lógico de una o más máquinas virtuales (VM) relacionadas que se pueden iniciar como una sola entidad en caso de un desastre. Cuando se inicia una vApp, las máquinas virtuales contenidas en la vApp comenzarán en un orden predefinido por el usuario, para permitir que las máquinas virtuales que dependen unas de otras se secuencian automáticamente. Esto significa que un administrador ya no tiene que secuenciar manualmente el inicio de máquinas virtuales dependientes si un servicio completo requiere reiniciar. La funcionalidad de vApp es particularmente útil en la situación de recuperación de desastres donde un administrador puede elegir agrupar todas las máquinas virtuales que residen en el mismo repositorio de almacenamiento.

### **3.3.2. *Monitoreo y Gestión XenServer***

XenServer proporciona un monitoreo detallado de las métricas de rendimiento, incluida la CPU, la memoria, el disco, la red y el almacenamiento para cada máquina virtual. XenServer también proporciona alertas de sistema y rendimiento. Las alertas son notificaciones que se producen en respuesta a eventos seleccionados del sistema, o cuando la CPU, el uso de memoria, la red, el rendimiento de almacenamiento o la actividad del disco de la VM superan un umbral especificado en un host, VM o repositorio de almacenamiento administrado.[2]

### **3.4. *Contenedores***

Para las tecnologías de la información, es bien sabido que de nada sirve un Servidor de Cómputo (equipo de cómputo físico o virtual, que permite ofrecer aplicaciones específicas a Usuarios finales), sin la piedra angular de la Infraestructura. En estos equipos es posible, tras la instalación de un robusto, seguro y escalable Sistema Operativo (como Red Hat Enterprise Linux), aprovisionar aplicaciones capaces de atender las peticiones de un Usuario, para subsanar cualquier necesidad informática. Servidores de Correo Electrónico, servidores de Páginas Web (HTTP Servers), servidores de Aplicaciones (como Red Hat JBoss Application Server), etc.

Si bien las máquinas virtuales (VM) tradicionales permiten la virtualización de la infraestructura de computación, los contenedores habilitan la de las aplicaciones de software. A diferencia de las máquinas virtuales, los contenedores utilizan el sistema operativo (SO) de su host en lugar de proporcionar uno propio.

¿Qué es lo que se viene a la mente cuando mencionamos la palabra Contenedor? Obviamente que esas inmensas cajas metálicas que, apiladas por cientos sobre buques plataforma, nos permiten trasladar de manera segura cualquier mercancía o bien usando los bastos océanos de nuestro planeta. Baste decir sólo como referencia, que aproximadamente el 95 por ciento de los bienes que se producen en cualquier país del mundo, son trasladados (exportados-importados) usando contenedores, buques plataforma y los océanos.

#### 3.4.1. (*¿Por qué utilizar contenedores?*)

Dado que no incluyen sistemas operativos completos, los contenedores requieren recursos informáticos mínimos y son rápidos y fáciles de instalar. Esta eficiencia permite que se implementen en clústeres, con contenedores individuales que encapsulan componentes únicos de aplicaciones complejas. Separar los componentes de la aplicación en diferentes contenedores permite a los desarrolladores actualizar componentes individuales sin necesidad de rehacer la aplicación completa.

De manera muy semejante, los Contenedores Informáticos utilizan una pieza de software a modo de un buque plataforma (ese software casi siempre es Docker), aprovechando ese océano de recursos informáticos (procesador, memoria, red y almacenamiento) sobre el mismo Sistema Operativo.

En lugar de virtualizar el hardware como ocurre con el enfoque de las máquinas virtuales, los contenedores se virtualizan a nivel del sistema operativo, con múltiples contenedores ejecutándose directamente sobre el núcleo del sistema operativo. Esto significa que los Contenedores son mucho más livianos (necesitan menos memoria RAM), comparten el núcleo del sistema operativo, se inicializan –bootstrap– mucho más rápido y usan una fracción de la memoria en comparación con el que necesita un sistema operativo

#### 3.4.2. *¿Qué es lo que dice Docker?*

Una imagen de contenedor es un paquete ligero, independiente, ejecutable de una pieza de software que incluye todo lo necesario para ejecutarlo: código, tiempo de ejecución, herramientas del sistema, bibliotecas del sistema, configuraciones.

Vale la pena agregar que: Disponible para aplicaciones basadas en Linux, el software en contenedor siempre funcionará igual, independientemente del entorno. Como ya lo habíamos mencionado líneas arriba: "Los contenedores aíslan el software de su entorno, por ejemplo, las diferencias entre los entornos

de desarrollo y de etapas, y ayudan a reducir los conflictos entre los equipos que ejecutan software diferente en la misma infraestructura.”

Ahora bien. ¿Qué es Docker? Es proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de Virtualización a nivel de sistema operativo en Linux.

### **3.4.3. *Impacto en los Procesos TI***

Como fue el caso con la Virtualización, los Contenedores traen consigo una cantidad de impactos positivos en los procesos TI. El primero gira alrededor de la gestión. En procesos tradicionales, operaciones de TI tiene mucha responsabilidad, que a menudo no se comparte con el grupo de desarrollo u otras áreas de la organización [1]. Algunos procesos tradicionales incluyen el uso de máquinas virtuales, actualización periódica del software, nuevas implementaciones del servidor físico, sincronización entornos de prueba / desarrollo y producción, gestión de la seguridad, y mucho más.

### **3.4.4. *Características de un contenedor***

- Representan recursos virtuales
- Ocupan poco espacio
- Están basados en imágenes. (En este contexto una imagen corresponde a un archivo con todos los elementos necesarios para ejecutar el contenedor; y metadatos que describen las capacidades y necesidades del mismo.)
- No necesitan un sistema operativo propio; pues utilizan el kernel del mismo sistema operativo que los hospeda.
- Comparten el sistema operativo principal con otros containers dentro de un mismo servidor.

## **4. Análisis**

### **4.1. *Virtualización vs Contenedores***

Cuando nos referimos a virtualización pensamos inmediatamente en máquinas virtuales, pero es importante entender que este es solo un tipo de virtualización en el cual se habilita un sistema operativo el cual tiene la ilusión de



que posee recursos dedicados para operar. Entendiendo lo anterior podemos ahora definir a Contenedores como otra alternativa de virtualización. Cada tecnología tiene sus aplicaciones y sus ventajas según las necesidades y circunstancias de cada desarrollo. En la actualidad los contenedores en general y Docker en particular se están convirtiendo en una tecnología indispensable y cada vez se utilizan para más cosas, no solo para desplegar aplicaciones en producción, sino también para crear entornos de desarrollo replicables entre todos los miembros de un equipo, asegurar que las aplicaciones se van a ejecutar igual en todos los entornos (desarrollo, pruebas y producción), etc.

#### **4.2. *¿Por qué Contenedores y no Máquinas Virtuales?***

Sólo para comenzar, los Contenedores son más ligeros, requieren de menos recursos informáticos y por ende más versátiles. Al igual que las Máquinas Virtuales, ofrecen portabilidad, escalabilidad y son tan seguros como lo es el Sistema Operativo donde se están ejecutando. Lo que hace a los Contenedores, para ciertas ocasiones, no muy adecuados, es para esos escenarios en los que se requieren Sistemas Operativos de distinta versión (sobre todo versión de Kernel), pues el requisito indispensable para que puedan ejecutarse los contenedores en el mismo servidor físico, es precisamente que todas éstas puedan ejecutarse sobre la misma versión del Kernel.

Los Contenedores aíslan las aplicaciones entre sí, a menos que se les conecte explícitamente. Eso significa que no existen dependencias conflictivas o contención de recursos, se establecen límites de recursos explícitos para cada servicio. Es importante destacar que se trata de una capa adicional de seguridad ya que sus aplicaciones no se ejecutan directamente en el sistema operativo host.

#### **4.3. *¿Cómo podemos ofrecer una solución segura, confiable, escalable y completa de Contenedores con Software Red Hat?***

Infraestructura con Red Hat Enterprise Linux: Manténgase ligero y ejecute sus contenedores Linux con un sistema operativo optimizado, seguro y de mínima huella. Plataforma con Red Hat Open Shift: Desarrolle, implemente y administre sus contenedores en cualquier lugar, a cualquier escala. Red Hat® OpenShift es una plataforma de aplicaciones de contenedores que incluye Docker y Kubernetes. Independientemente de la arquitectura de sus aplicaciones, OpenShift le permite construir, desarrollar e implementar fácil y rápidamente en casi cualquier infraestructura, pública o privada.

#### 4.4. *Selección entre Contenedores y Máquinas Virtuales*

Los contenedores tienen como principal beneficio el aislamiento de aplicaciones y va a compartir los recursos entre todos los contenedores optimizando su uso mientras las máquinas virtuales permiten la administración del hardware para crear múltiples máquinas en un servidor o varios. Además los contenedores son mucho más ligeros que las máquinas virtuales con el uso de contenedores ya no es necesario instalar múltiples sistemas operativos para varias aplicaciones.

### 5. Conclusiones

- Conclusion 1 :  
Los contenedores permiten desplegar aplicaciones más rápido, arrancarlas y pararlas más rápido y aprovechar mejor los recursos de hardware. Las máquinas virtuales nos permiten crear sistemas completos totalmente aislados, con mayor control sobre el entorno y mezclando sistemas operativos host y huésped.
- Conclusion 2 :  
Si bien usamos los contenedores para desarrollar y desplegar aplicaciones de forma sencilla y rápida la mayoría de las empresas todavía mantienen el uso de máquinas virtuales por su flexibilidad en casos de desastres gracias a que si un servidor falla, la máquina virtual levantará un servidor espejo
- Conclusion 3 :
- Conclusion 4 : El uso de Contenedores para una integración e implementación continua es posible porque Docker proporciona un sistema para el control de versiones de imágenes. Puede configurarse un proceso de compilación para extraer el código de un repositorio, compilarlo, empaquetarlo en una imagen Docker y enviar la imagen recién creada a un repositorio de imágenes. A continuación, puede hacerse que el proceso de implementación extraiga la nueva imagen de su repositorio, pruebe la aplicación y la implemente en los servidores de producción. Pudiendo evitar tener una aplicación que funcione en su entorno de desarrollo pero falle en la producción porque el daemon de Docker es el mismo en todas las máquinas de desarrollo, montaje y producción.

## Referencias

- [1] Criptologico, C. (2019). *Implementacion de Seguridad en Microsoft Hyper-V sobre Windows Server 2016*. Sidertia Solutions S.L.
- [2] XenServer, C. (2018). *Citrix XenServer 7.0 Administrator's Guide*. Citrix Inc.