VIRTUALIZACIO´N Y CONTENEDORES

Laura Atencio, Nilson Felix (2015053846)
Velasco Sucapuca, Andree Ludwerd(2016055286)

Universidad Privada de Tacna \Facultad de Ingenieria \Escuela Profesional de Ingenieria de Sistemas

Resumen

Las tendencias tecnológicas siguen evolucionando y cada día van apareciendo nuevos conceptos que debemos aprender. Hoy en día casi no se puede tener una discusión respecto a **Cloud Computing** sin llegar al concepto de “Contenedores” (Containers). Organizaciones de todos los segmentos de negocio hoy quieren entender que son los contenedores, que significan para las aplicaciones en la nube y como pueden usarlos.
Antes de ahondar en el concepto de contenedores, volvamos unos años atrás para recordar el nacimiento de la **virtualización**. A medida que el **hardware** se hacía más poderoso nos encontramos con que el software no ocupaba todas las capacidades de la maquina física donde se encontraba siendo ejecutada (en muchos casos ni siquiera una fracción de estos recursos). Dado lo anterior se crearon recursos “virtuales” para simular el hardware base sobre el cual se ejecuta el software, permitiendo que múltiples aplicaciones puedan ser ejecutadas al mismo tiempo, cada una usando una fracción de los recursos del hardware físico disponible. A a esta “simulación” que permite de compartir recursos la denominamos comúnmente “virtualización”.

Abstract

Technological trends continue to evolve and new concepts are emerging every day that we must learn. Nowadays you can hardly have a discussion about cloud computing to get to the concept of "Containers" (Containers). Organizations from all segments of the business.
Before delving into the container concept, we went back a few years ago to remember the birth of virtualization. To the extent that the hardware becomes more powerful we find software that will not occupy all the capabilities of the physical machine where it is being executed (in many cases not even a part of these resources). Given the above, virtual resources were created to simulate the base hardware on which the software is located, the multiple applications can be used to run at the same time. A "simulation" that allows sharing the resources of virtualization.

I. INTRODUCCIO´N

Mucha gente, cuando oye hablar de Docker y de lo que se puede hacer con él, lo primero que piensa es en máquinas virtuales. Al fin y al cabo, una máquina virtual es un software que permite aislarse del sistema operativo subyacente y compartirlo entre varias aplicaciones.
Sin embargo las diferencias entre las tecnologías de contenedores como Docker y las máquinas virtuales son enormes, tanto conceptualmente como en la práctica. En este artículo vamos a repasar brevemente ambas tecnologías para ver cómo trabajan y entender bien sus diferencias. No volverás a tener dudas al respecto
Como su propio nombre indica, una máquina virtual (o VM a partir de ahora, de sus siglas en inglés: *Virtual Machine*) es un sistema operativo completo funcionando de manera aislada sobre otro sistema operativo completo.
La tecnología de VMs permite compartir el *hardware* de modo que lo puedan utilizar varios sistemas operativos al mismo tiempo.
La filosofía de los contenedores es totalmente diferente a la de las VMs. Si bien tratan también de aislar a las aplicaciones y de generar un entorno replicable y estable para que funcionen, en lugar de albergar un sistema operativo completo lo que hacen es compartir los recursos del propio sistema operativo "host" sobre el que se ejecutan.
Docker Engine se encarga de lanzar y gestionar los contenedores con nuestras aplicaciones, pero en lugar de exponer los diferentes recursos de hardware de la máquina de manera discreta (es decir, 1 procesador y "x" GB de RAM... para cada aplicación), lo que hace es compartirlos entre todos los contenedores optimizando su uso y eliminando la necesidad de tener sistemas operativos separados para conseguir el aislamiento.

II. OBJETIVOS

A. General:

- Determinar las caracteriısticas diferenciales las m´aquinas virtuales y los contenedores.

B. Espec´ıf tcos:

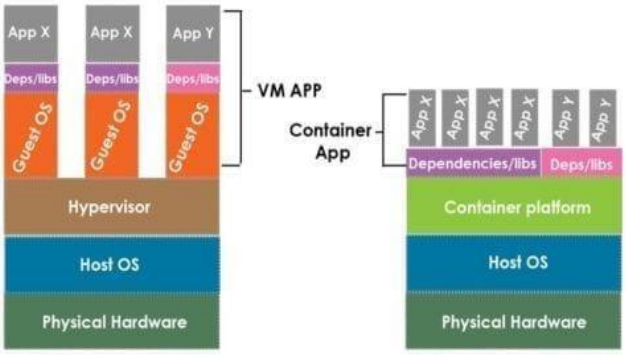
- Definir los conceptos de maquinas virtuales y con- tenedores. - Comparar las definiciones.

III. MARCO TEO´RICO

A. Contenedores vs virtualización: ¿cuál es superior?

La virtualización ha sido la base de muchas tecnologías modernas, y no puede faltar en la implementación de la red de próxima generación: Virtualización de funciones de red y redes definidas por software (NFV/SDN). La virtualización se ha utilizado sistemáticamente para simplificar la implementación, la gestión, la orquestación y la elasticidad de los proveedores de servicios de comunicaciones (CSP), y en la actualidad los principales CSP han virtualizado más de la mitad de sus redes. Aún así, Kubernetes, una plataforma de contenedores diseñada por google para cargas de trabajo de red, está ganando terreno en la implantación de redes y las soluciones de infraestructura virtual podrían enfrentarse pronto a una competencia feroz. Las máquinas virtuales son hoy en día el estándar de facto para el despliegue de software, pero no es la única tecnología capaz de satisfacer este nicho. Los contenedores, una tecnología que esencialmente aísla las aplicaciones de los sistemas operativos del host (muy similar a una máquina virtual) se está convirtiendo rápidamente en una alternativa viable para muchos escenarios de implementación de software. Aunque las tecnologías comparten muchas similitudes en cuanto a la funcionalidad final, los contenedores ofrecen algunas ventajas y desventajas a las máquinas virtuales.

Contenedores y máquinas virtuales: ¿cuál es la diferencia?



Se muestra: comparación de la arquitectura de un contenedor con la de una máquina virtual que ejecuta aplicaciones X e Y.

Aunque ambas tecnologías tienen el mismo objetivo: aislar una aplicación de otros procesos y aplicaciones en el sistema host, ambas tienen enfoques bastante diferentes.

Máquinas virtuales: Como su nombre indica, este enfoque está mucho más involucrado en el alcance. Se basa en un hipervisor (por ejemplo, KVM, XEN) que emula una máquina física completa, asigna una cantidad deseada de memoria del sistema, núcleos de procesador y otros recursos como almacenamiento en disco, redes, complementos PCI, etc. **Contenedores:**Las tecnologías existen desde hace mucho tiempo, aunque con diferentes nombres: jaulas, areneros, etc. Es el único hada reciente que la tecnología ha madurado lo suficiente y se ha introducido en los entornos de producción. Los contenedores aíslan esencialmente una aplicación del host a través de varias técnicas, pero utilizan el mismo núcleo de sistemas host, procesos (por ejemplo, pila de red) para ejecutar la aplicación o VNFs.

¿Qué significa esto para el rendimiento, la seguridad y la portabilidad?

Tecnologías y técnicas de **Virtualización** han recorrido un largo camino tanto para el software como para el hardware. La mayoría de los procesadores x86 fabricados a partir de 2013 incluyen optimizaciones específicas de virtualización (Intel VTx, AMD-V), lo que supone una penalización de los gastos generales de virtualización en el procesador de alrededor del 2%, una compensación más que justa por la funcionalidad que aporta la virtualización. Lo mismo no puede decirse de otros recursos como la memoria del sistema y el almacenamiento. Dado que una máquina virtual ejecuta todo un sistema operativo sobre el sistema operativo del host, es inherentemente más ineficiente en términos de tamaño de aplicación y uso de la memoria del sistema. La virtualización no sólo consume más memoria del sistema, sino que requiere que se asigne una cantidad fija a la VM, incluso si la aplicación no está consumiendo esos recursos. Teniendo todo esto en cuenta, el uso de la memoria del sistema podría acabar siendo la diferencia más importante entre la virtualización y los contenedores.
Una de las verdaderas ventajas de las VM sobre los contenedores es su portabilidad. Aunque los contenedores docker ofrecen un cierto grado de portabilidad entre el sistema operativo del host al empaquetar las dependencias con la aplicación, no hay garantía de que el sistema operativo subyacente del host sea compatible con la aplicación de contenedores XYZ. Otra ventaja es la madurez de las soluciones de gestión de máquinas virtuales, aunque Kubernetes está cerrando esta brecha de manera constante.

Contenedores se consideran a menudo más eficientes que la virtualización por diseño, porque en lugar de duplicar los procesos y servicios disponibles en el sistema operativo del host dentro del sistema operativo del host, las aplicaciones se ejecutan en entornos de caja de arena dentro del sistema operativo del host, eliminando las capas de abstracción, esencialmente ejecutando aplicaciones en modo totalmente metálico. Aunque no es falso, Docker (el principal proyecto de contenedores) no llega sin sus propios éxitos de rendimiento. Por ejemplo: La traducción de acceso a la red de los Dockers introduce una sobrecarga que puede afectar al rendimiento en cargas de trabajo elevadas.
Considerando la baja sobrecarga de los hipervisores modernos, la eficiencia real en los contenedores proviene de la reducción del uso de memoria debido a la eliminación del SO huésped, la subsiguiente desduplicación de los procesos que consumen recursos adicionales y la reducción del tamaño de la aplicación debido a las reducciones mencionadas anteriormente. Combinando esto con la capacidad de gestionar recursos como la memoria del sistema sobre la marcha y de forma dinámica, se podría conseguir una opción de implementación mucho más eficiente.

Otra característica prometedora del contenedor es el tiempo de arranque. Dado que la aplicación no tiene que iniciar todo un SO huésped antes de iniciarse, se trata de una plataforma de implementación mucho más ágil que podría impulsar potencialmente la adopción en áreas como la división de redes 5G.
Cuando se trata de seguridad, ambas tecnologías pueden sufrir las mismas vulnerabilidades del sistema operativo del host, de la biblioteca o de las aplicaciones, aunque la superficie de ataque se reduce bastante para los contenedores, ya que no es necesario un sistema operativo huésped adicional. Al mismo tiempo, los hipervisores son más maduros y, como tales,

ofrecen actualmente una visión más transparente de los procesos en ejecución. Sólo el tiempo dirá qué tecnología puede proporcionar el sistema más seguro.

Problemas de los contenedores

Los contenedores pueden parecer el entorno de virtualización ideal y, sin embargo, no están exentos de problemas. El principal de ellos es la **seguridad**. El contenedor deja la seguridad en manos del sistema operativo y, por tanto, se debe encapsular al usuario. Una política de permisos incorrecta puede derivar en un sistema vulnerable.

Una forma de **encapsular** al usuario consiste en definir accesos de "solo lectura" o de "escritura" en ubicaciones definidas, pero esta configuración es manual y requiere bastante trabajo del Administrador del sistema.

La **regla básica** es "*tratar un contenedor como cualquier otra aplicación*", siguiendo tres directrices:

1. Quitar todos los privilegios lo más pronto posible.
2. Ejecutar los servicios como un usuario regular (no Administrador) siempre que se pueda.
3. Tratar al Administrador *dentro* del contenedor como al Administrador *fuera* del contenedor.

Otro problema es que cualquiera puede crear contenedores y el **origen** de los mismos puede ser **desconocido**. Es muy recomendable verificar su contenido en sistemas independientes del entorno productivo, ya que su contenido puede ser inesperado y pueden hacer que un sistema se convierta en vulnerable.

Un contenedor soluciona la parte descendente del problema, ya que se sabe lo que tiene, pero no soluciona la parte ascendente de las **dependencias**. Es fácil implantar una aplicación en un contenedor, pero si no es la adecuada la implantación se transforma en una pérdida de tiempo.

La **extensión** de contenedores puede convertirse en un quebradero de cabeza. Desmenuzar las implantaciones en partes es un método inteligente, pero implica que hay muchas más partes que gestionar. Habrá que establecer un punto de inflexión entre extensión y preocupación.

El **objetivo** de un contenedor es ejecutar una única aplicación. Cuantas más aplicaciones se adhieran a un contenedor, más necesario se hace usar una máquina virtual.

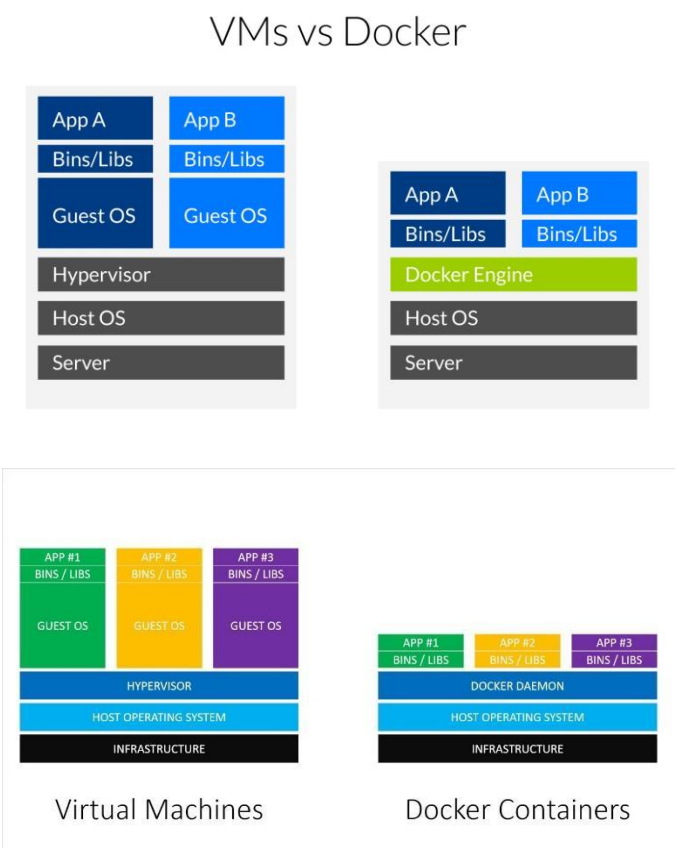
Un contenedor está diseñado para ejecutar el mismo sistema operativo donde se aloja. Es muy complejo ejecutar aplicaciones diseñadas para un sistema operativo dentro de un contenedor alojado en otro sistema operativo diferente. La razón es que el **núcleo** del sistema subyacente es **distinto**. Esta característica puede ser beneficiosa porque evita preocupaciones por las dependencias una vez que la aplicación se ejecute adecuadamente en el contenedor, pero establece límites que la máquina virtual no tiene.

En términos generales, el uso de contenedores está orientado a **una sola aplicación**, por ejemplo, varias instancias de un motor de base de datos o servicios web, mientras que las máquinas virtuales están enfocadas en implantar sistemas operativos ágilmente para ejecutar múltiples aplicaciones de manera flexible.
El enfoque de uso de estas tecnologías no debe ser exclusivo sino al contrario. La **combinación** de ambos métodos al usarlos conjuntamente aporta una complementación de virtudes muy beneficiosa.

Cuáles serían entonces las **ventajas** de la virtualización por contenedores frente a las máquinas virtuales?

1. Permiten implantar mayor número de aplicaciones en un único servidor que las máquinas virtuales en sistemas On Cloud o Data Centers.
2. Solo necesitan un sistema operativo donde alojarse, software de soporte y unos recursos mínimos. Las máquinas virtuales necesitan muchos recursos del sistema para recrear un sistema operativo independiente y todo el hardware que necesite, por lo que la carga del sistema aumenta considerablemente.
3. Los contenedores permiten instalar entre dos y tres veces más software que con una máquina virtual.
4. Son muy útiles para replicar entornos operativos estables, por ejemplo, entornos de desarrollo, test e implantación.

B. Diferencias



En primer lugar debemos tener en cuenta que, en el caso de los contenedores, el hecho de que no necesiten un sistema operativo completo sino que reutilicen el subyacente **reduce mucho la carga** que debe soportar la máquina física, **el espacio** de almacenamiento utilizado **y el tiempo** necesario para lanzar las aplicaciones. Un sistema operativo puede ocupar desde poco menos de 1GB para algunas distribuciones de Linux con lo mínimo necesario, hasta más de 10GB en el caso de un sistema Windows completo. Además, estos sistemas operativos, para funcionar requieren un mínimo de memoria RAM reservada, que puede ir desde 1 hasta varios GB, dependiendo de nuestras necesidades. Por lo tanto **los contenedores son mucho más ligeros que las máquinas virtuales**.

Cuando definimos **una máquina virtual debemos indicar de antemano cuántos recursos físicos le debemos dedicar**. Por ejemplo, podemos decir que nuestra VM va a necesitar 2 vCores (procesadores virtuales), 4GB de RAM y un espacio en disco de 100 GB. En el caso de los procesadores, es posible compartirlos entre varias máquinas virtuales (pero no conviene pasarse o irán fatal de rendimiento), y el espacio en disco se puede hacer que solo ocupe lo que de verdad se esté utilizando, de modo que crezca en función de las necesidades y no ocupe siempre tanto como habíamos reservado. Pero en el caso de la memoria y otros elementos (acceso a unidades externas o dispositivos USB) la reserva es total. Por eso, aunque nuestra aplicación no haga uso en realidad de los 4GB de RAM reservados da igual: no podrán ser utilizados por otras máquinas virtuales ni por nadie más. **En el caso de los contenedores** esto no es así. De hecho no indicamos qué recursos vamos a necesitar, sino que es Docker Engine, en función de las necesidades de cada momento, el encargado de **asignar lo que sea necesario para que los contenedores funcionen** adecuadamente.

Esto hace que los entornos de ejecución de Docker sean mucho más ligeros, y que se aproveche mucho mejor el *hardware*, además de permitir levantar muchos más contenedores que VMs en la misma máquina física. Mientras que una VM puede tardar un minuto o más en arrancar y tener disponible nuestra aplicación, **un contenedor Docker se levanta y responde en unos pocos segundos** (o menos, según la imagen). **El espacio ocupado en disco es muy inferior con Docker** al no necesitar que instalemos el sistema operativo completo.

Por otro lado, Docker no permite utilizar en un sistema operativo "host" contenedores/aplicaciones que no sean para ese mismo sistema operativo. Es decir, no podemos ejecutar un contenedor con una aplicación para Linux en Windows ni al revés. Lo cual puede suponer un impedimento en algunas ocasiones.

CONCLUSIONES

Ambas tecnologías ofrecen ventajas distintas:

La virtualización viene con una plétora de herramientas probadas a lo largo del tiempo, plataformas de gestión y orquestación, sondas virtuales, soluciones de infraestructura virtual hiperconvertidas y mucho más. La portabilidad y la interoperabilidad son las características que destacan frente a los contenedores.

Los contenedores ofrecen una mayor eficiencia de recursos y agilidad de servicio. Aunque no parezca mucho, abre la puerta a un modelo de microservicios que puede escalar más rápido y de manera más eficiente. Los contenedores de papel se ajustan más a las iniciativas de NFV/SDN y la industria se ha dado cuenta de

que Kubernetes es uno de los proyectos de código abierto de más rápido crecimiento hasta la fecha.

Hoy en día, los proveedores de servicios se encuentran en medio de una evolución de la red y buscan utilizar la mejor tecnología disponible, y para ello muchos están utilizando contenedores dentro de una máquina virtual para aprovechar las mejores herramientas y soluciones de gestión de infraestructura disponibles en la actualidad. Aunque esto elimina algunos de los beneficios de los contenedores, permite a los proveedores de servicios aprovechar la agilidad y eficiencia de memoria de los contenedores para mitigar las ineficiencias presentes en las máquinas virtuales y ofrece lo mejor de ambos mundos. Eventualmente, la interoperabilidad mejorada y las API estandarizadas pueden permitir que los contenedores y las máquinas virtuales trabajen juntos y creen la solución de implementación de software ideal para SDN/NFV.

[1] Miguel Alonso. Virtualizacion de sistemas: m´aquinas vir- tuales frente a contenedores. Accessed: 2019-05-17.

[2] Know How. Docker y otros container: mas all´a de la vir- tualizaci´on. Accessed: 2019-05-17.

[3] Campus MVP. Los beneficios de utilizar docker y contene- dores a la hora de programar. Accessed: 2019-05-17. [4] G. Wolf. *Fundamentos de sistemas operativos*. 2015.