330.1 Virtualization Concepts and Theory

Weight: 8

Description: Candidates should know and understand the general concepts, theory and terminology of Virtualization. This includes Xen, KVM and libvirt terminology.

Key Knowledge Areas:

- Terminology
- Pros and Cons of Virtualization
- Variations of Virtual Machine Monitors
- Migration of Physical to Virtual Machines
- Migration of Virtual Machines between Host systems
- Cloud Computing

The following is a partial list of the used files, terms and utilities:

- Hypervisor
- Hardware Virtual Machine (HVM)
- Paravirtualization (PV)
- Container Virtualization
- Emulation and Simulation
- CPU flags
- /proc/cpuinfo
- Migration (P2V, V2V)
- IaaS, PaaS, SaaS

1 ¿Qué es la virtualización?

El significado del término virtual, significa que algo no es real. En el caso de la informática este término virtual significa "entorno hardware el cual no es real". Así, conseguimos duplicar las funciones del hardware físico y se le presenta a un sistema operativo.
La tecnología que se utiliza para crear este entorno se denomina tecnología de virtualización, resumiendo, virtualización simplemente.
El sistema físico que ejecuta el software de virtualización (hipervisor o monitor de máquina virtual) es llamada al host y las máquinas virtuales instaladas en el hipervisor se llaman invitados o guests.

2 ¿Por que motivo debería usar la virtualización?

En un primer momento la virtualización aparece en Linux en forma de modo usuario (UML, user-mode Linux) y desde este momento comienza el camino hacia la virtualización en Linux.
Hoy día, existen varias opciones disponibles en Linux para convertir un ordenador físico en varias.
Existen varias soluciones de virtualización populares en Linux, incluidas KVM, Xen, QUEMU, VirtualBox, VMWare...

Existen distintas técnicas de virtualización, las principales son las siguientes:

- **↘Virtualización completa (a nivel de hardware):** El sistema operativo anfitrión (que es el sistema operativo principal), asigna un "contenedor" al/los distintos sistemas operativos huéspedes, con lo que esto genera ventaja ya que permite ejecutar varios sistemas operativos y sin necesidad de realizar modificaciones en los mismos, en contra está que no se puede alcanzar las tasas de rendimiento que se dispondría en una máquina real.

  Usando esta técnica se puede implementar usando un **hipervisor de Tipo 1 o de Tipo 2**:
  - **Hipervisor tipo 1**: También denominado *nativo*, *unhosted* o *bare metal* (*sobre el metal desnudo*), es software que se ejecuta directamente sobre el hardware, para ofrecer la funcionalidad descrita.
    Los hipervisores tipo 1 más conocidos son los siguientes: VMware ESXi (gratis), VMware ESX (de pago), Xen (libre), Citrix XenServer (gratis), Microsoft Hyper-V Server (gratis).
  - **Hipervisor tipo 2**: También denominado *hosted*, es software que se ejecuta sobre un sistema operativo para ofrecer la funcionalidad descrita.
    Los hipervisores tipo 2 más utilizados son los siguientes: **Oracle:** VirtualBox (gratis), VirtualBox OSE (libre); **VMware:** Workstation (de pago), Server (gratis), Player (gratis); **QEMU** (libre); **Microsoft:** Virtual PC, Virtual Server.

  ↘ Virtualización a nivel de sistema operativo: Con esta técnica se divide un ordenador en varios compartimentos de forma independiente, de manera que en cada compartimento se pueda instalar un servidor. A estos compartimentos se les denomina "entornos virtuales". En realidad el sistema funciona como si realmente existieran varios servidores ejecutandose en máquinas distintas. Ejemplos: OpenVZ, Virtuozzo, FreeVPS, Linux-Vserver.

- **↘ Para-Virtualización:** En la que existe una "capa" o "Hypervisor" entre la parte hardware y los sistemas operativos ya que se consigue mediar para usar los recursos, digamos que se consigue un mini sistema operativo. La principal ventaja de uso con esta técnica es el rendimiento, y que es muy cercano al real, pero que en su defecto es necesario modificar el sistema operativo principal.

2.1 La virtualización completa

En la siguiente tabla, se presenta las capas de la arquitectura de una máquina virtual utilizando esta técnica:

Aplicación	**Aplicación**
Sistema operativo huésped	**Sistema Operativo huésped**
Software de Virtualización	
Sistema operativo anfitrión	
Hardware	

2.2 Para-Virtualización

Aplicación	**Aplicación**
Sistema Operativo huésped	Sistema Operativo huésped
Hypervisor de Soft. Virtualización integrado con el sistema operativo anfitrión	
Hardware	

3. Ventajas e inconvenientes de la virtualización

Encontramos distintas **ventajas** para utilizar la virtualización:

a) Se consigue un aumento de la disponibilidad (uptime)
b) Se consigue una mejora con las poĺíticas de backup
c) El aprovechamiento óptimo en los recursos hardware diponibles
d) Se consigue escalabilidad
e) Eficiencia energética
f) Se consigue el ahorro de costes
g) Poder crear entornos de prueba
h) Compatibilidad con aplicaciones
i) Compatibilidad con los periféricos
j) Disponibilidad de aislamiento y seguridad
k) Clonar y migrar sistemas en caliente (hotplug)
l) Se consigue un ahorra de espacio en el CPD (Centro de Proceso de Datos)
m) El poder centralizar la administración de todas las máquinas

En lo que respecta a los **inconvenientes:**

a) La importancia de uso de hardware de altas prestaciones
b) Sistemas que dependen de un sólo equipo
c) Las limitaciones con el hardware de las máquinas virtuales
d) Problemas con la emulación de algunos controladores
e) Inferior rendimiento
f) Aumento y proliferación de máquinas virtuales

4. Software de virtualización

Actualmente existe software que proporcionan distintos entornos de virtualización como WinBSD, XEN, Parallels o VirtualBox, además de VMWare.

Más información:

VMWare:

http://www.vmware.com/es/
http://es.wikipedia.org/wiki/VhMware

XEN:

https://www.xenproject.org/
https://es.wikipedia.org/wiki/Xen

PARALLELS:

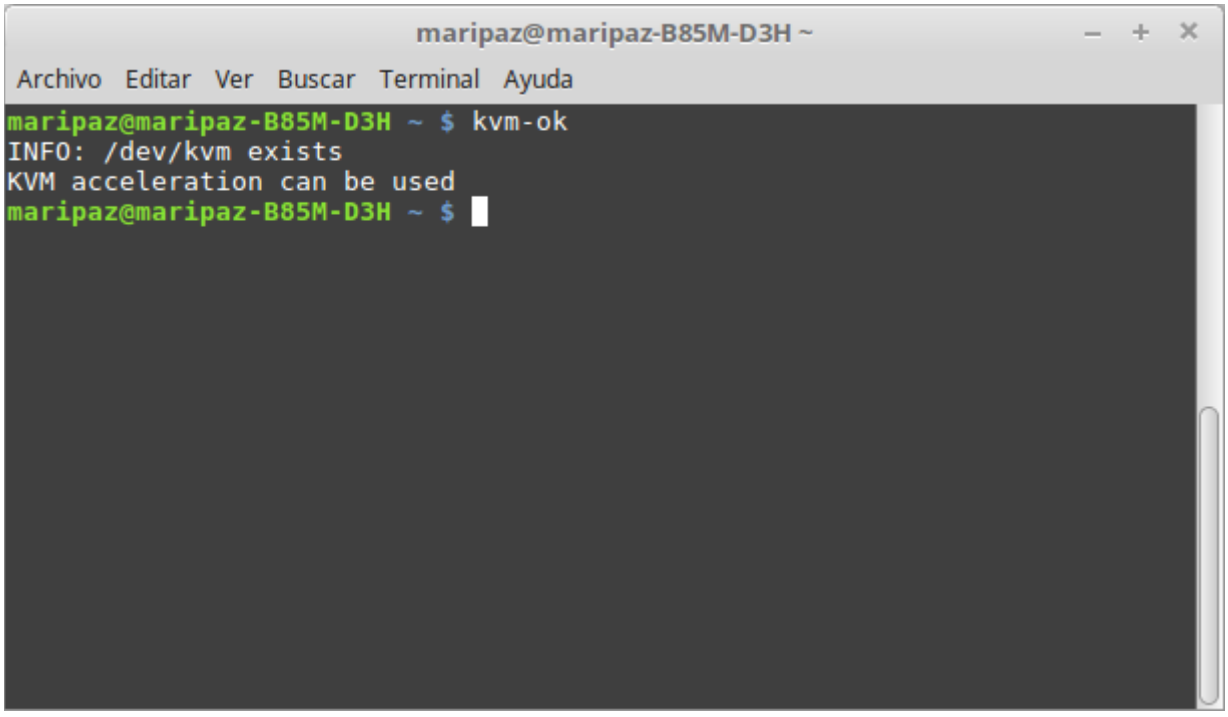
http://www.parallels.com/es/
https://es.wikipedia.org/wiki/Parallels_Desktop_para_Mac

KVM:

https://www.linux-kvm.org/page/Main_Page
https://es.wikipedia.org/wiki/Kernel-based_Virtual_Machine

LIBVIRT:

La librería libvirt es usada como interfaz con distintas tecnologías de virtualización. Antes de empezar a utilizar libvirt es importante asegurarse de la compatibilidad del hardware para KVM con el comando kvm-ok



Existen diferentes maneras para permitir a una máquina virtual acceder a una red externa. La configuración por defecto de la red incluye el puenteado y las reglas iptables implementadas, las cuales utilizan el protocolo SLIRP. El tráfico está con NAT entre la interfaz del host hacia fuera de la red.

Para activar al host directamente y acceder a los servicios en la máquina virtual se debe configurar un tipo distinto de bridge. Esto permitirá a los interfaces virtuales conectar a la red externa a través de la interfaz física, consiguiente que aparece como un host normal al resto de la red.

Instalación: **sudo apt install qemu-kvm libvirt-bin**

Una vez finalizada la instalación, añadimos al usuario al grupo libvirtd: **sudo adduser maripaz libvirtd**

Vamos a instalar **virt-install** y que está dentro de virtinst: **sudo apt install virtinst**

Ejemplo de uso con '**virt-install**':

```
sudo virt-install -n web_devel -r 256 \
--disk path=/var/lib/libvirt/images/web_devel.img,bus=virtio,size=4 -c \
ubuntu-16.04-server-i386.iso --network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

- **-n** web_devel → nombre de la nueva máquina virtual, en este ejemplo es web_devel.
- **-r** 256 → especifica la cantidad de memoria RAM de la máquina virtual en megabytes.
- **--disk path=**/var/lib/libvirt/images/web_devel.img,**size=**4 → indica la ruta al disco virtual, el cual puede ser un archivo, partición, o volumen lógico. En el ejemplo el archivo nombrado como web_devel.img en /var/lib/libvirt/images con un tamaño de 4 gigabytes y usando virtio para el bus de disco.
- **-c** ubuntu-16.04-server-i386.iso → archivo a usar como CDROM virtual. El archivo puede ser un archivo ISO o la ruta al dispositivo de CDROM del host.
- **--network** → proporciona detalles relativos al interfaz de red de la máquina virtual. La red por defecto a utilizar, y la configuración del modelo de la interfaz para virtio.
- **--graphics vnc,listen=0.0.0.0** → exporta la consola virtual del invitado usando VNC en todos los interfaces de los hosts. Tipicamente los servidores no tienen GUI, así que otro equipo en la red local puede conectar vía VNC para terminar la instalación.
- **--noautoconsole** → no conexión automatica a la consola de la máquina virtual.
- **-v** → crea un host invitado con virtualización completa.

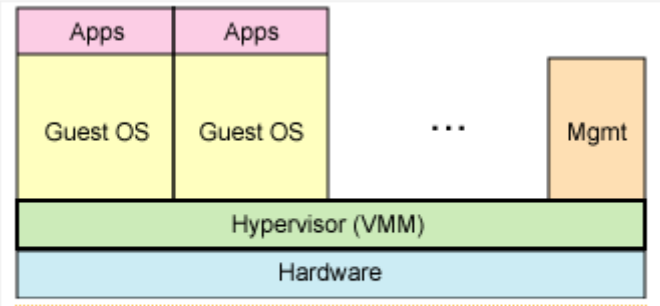
virt-clone → se utiliza para copiar una máquina virtual a otra.

Ejemplo: **sudo virt-clone -o web_devel -n database_devel -f /path/to/database_devel.img**

- **-o** → máquina virtual original
- **-n** → nombre de la nueva máquina virtual
- **-f** → ruta al archivo, volumen lógico, partición para ser usada en la nueva máquina virtual

5. Virtualización completa

Con este tipo de virtualización, las instrucciones enviadas de la máquina virtual al procesador físico tal como se muestra en la siguiente imagen:



Virtualización completa utilizado por VMWare y VirtualBox

El uso de la virtualización completa está más extendida que las dos sistemas anteriores, y sobre todo en usuarios comunes, ya que se consigue la posibilidad de ejecutar un sistema operativo sobre otro que esté instalado en el equipo real.

6. El hipervisor

El hypervisor es la capa intermedia y que está situada entre el hardware real y el sistema operativo virtual consiguiente con ello la ejecución de un sistema operativo sobre un hardware real.
Con este tipo de virtualización conseguimos mejor rendimiento si el propio procesador da soporte a las instrucciones virtuales, como el caso de tecnologías VT y PACIFICA de Intel y AMD. Esto se consigue ya que el procesador consigue interpretar las instrucciones generadas por la propia máquina virtual, por lo que no es necesaria su traducción.

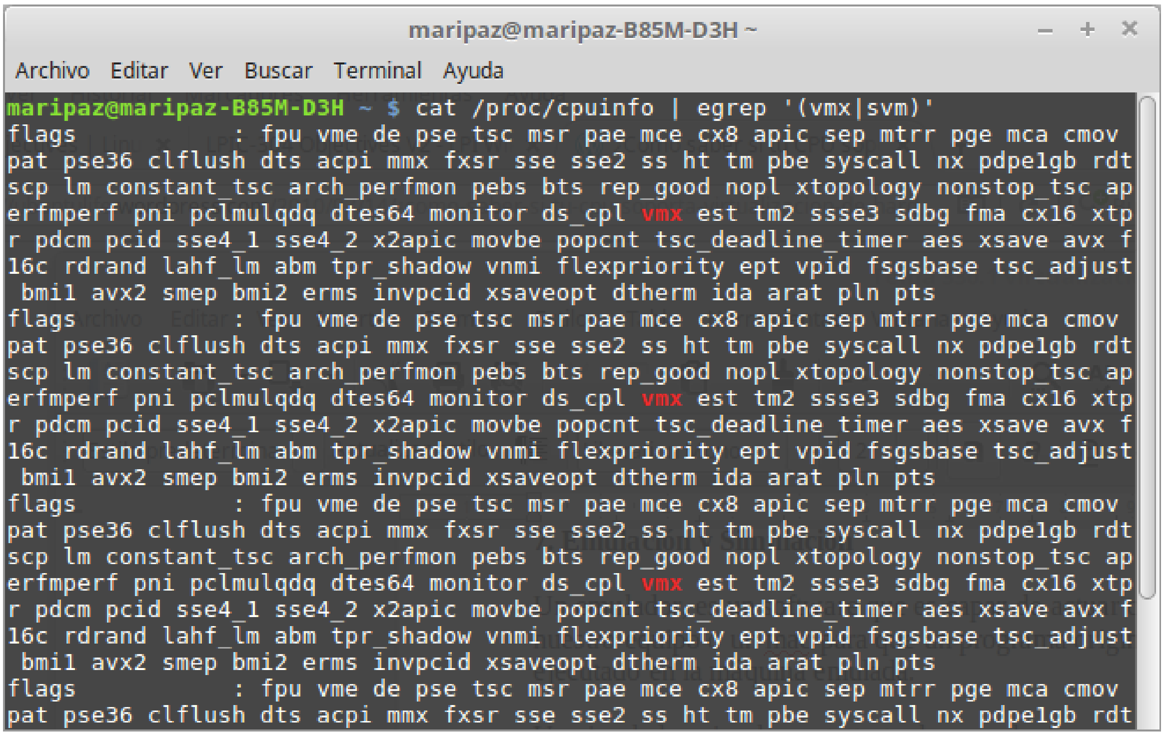
7. Emulación y Simulación

Los emuladores son programas que emulan el funcionamiento de una plataforma la cual está diseñada para otro procesador y otro chipset, es decir como ejemplo el emular juegos que estén diseñados para SNES en PC, y esta emulación se consigue completamente a base de software y del driver de la tarjeta de vídeo, por eso algunos emuladores que parezca que no corran juegos pesados para PC, a la hora de ejecutarlos en PC se pueda creer que no se tengan problemas para ejecutar esos juegos, y se puedan ver en cámara lenta o con problemas del audio.
También se disponen de algunos emuladores de otros sistemas operativos, como BlueStacks, con el que se ejecuta una emulación del sistema operativo Android, y que dispone de capacidad para instalar y ejecutar todas las aplicaciones compatibles para la versión Android emulada, este programa es similar a los virtualizadores, debido a que se crea un espacio de disco duro, y que se reserva especialmente para utilizarse como disco duro virtual para el Android emulado.

Un simulador sin embargo, es el software que ofrece la capacidad de experimentar la experiencia de una actividad, por ejemplo el volar, carreras de coches en un ambiente falso, pero con el que se simula las condiciones orginales.
Existen también simuladores de software, y que se limitan a ejecutar un programa de aspecto similar al original, pero sin la capacidad de ejecutar los programas compatibles en el caso de una simulación de OS, como ejemplo iPadian, que es un simulador que imita el aspecto de iOS del iPhone e iPad, además de disponer del acceso a internet y ejecutando las aplicaciones en el escritorio virtual de iPadian, pero sin capacidad de instalación de nuevas aplicaciones compatibles con iOS.

8. CPU flags

Estas flags, nos van a indicar si nuestro procesador soporta o no la tecnología de la virtualización, y para conseguir esta información debemos escribir el siguiente comando en una consola → **cat /proc/cpuinfo | egrep '(vmx|svm)'**

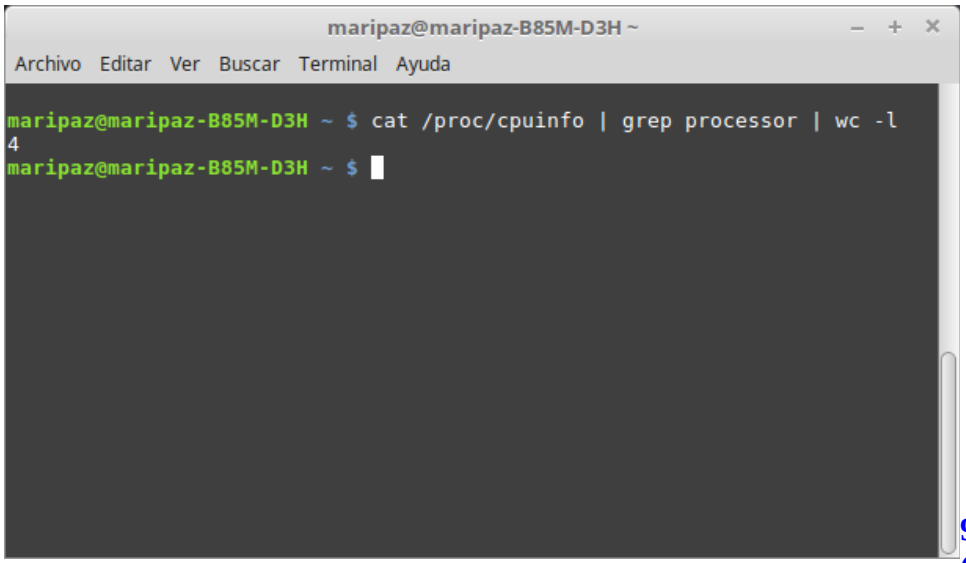


Comprobando si el procesador soporta la virtualización → **'vmx'**

Tal como vemos en la imagen, aparece vmx (procesador de intel) y por lo tanto esta CPU soporta la virtualización.

Además de esta comprobación, también es importante activarla en la BIOS ya que sino no funcionará adecuadamente.

Otro comando importante a utilizar para terminar de comprobar si el número de núcleos del que dispone nuestro procesador que debe ser mayor de 1 para disponer de un funcionamiento correcto de la virtualización y que se hace con este comando: **cat /proc/cpuinfo | grep processor | wc -l**



9. Migrar (P2V, V2V)

Existen herramientas como por ejemplo Converter Standalone incluida en Vmware vCenter que permite migrar una máquina física a una máquina virtual de una forma sencilla y con el correspondiente ahorro de tiempo.

Aparte de la opción anterior, en la que se puede migrar una máquina física a una máquina virtual, existe la opción de migrar de una máquina virtual a otra máquina virtual sin necesidad de crear una nueva VM usando una herramienta como puede ser Vmware Machine Importer.

10. IaaS, PaaS, SaaS

Cuando desarrollamos aplicaciones en el cloud (la nube) hay que especificar como lo vamos a hacer, con lo que según como se realiza la tarea se conseguirá flexibilidad o conseguir sencillez a la hora de conseguir mantenimiento de las mismas. Por lo que existen distintas formas que la nube puede adoptar:

↘ **Infraestructure-as-a-Service (IaaS)**: Si utlizamos esta forma de adoptar la nube, esta ofrece un mayor control, aunque es indispensable encargarse de la gestión de la infraestructura. El ejemplo de este tipo de nube es el que proporciona Amazon Web Service (AWS).

↘ **Platform-as-a-Service (PaaS):** Con este servicio se consigue que los desarrolladores puedan crear y modificar las aplicaciones que se ejecutan en la nube, por lo que la principal ventaja es que sólo debemos construir nuestra aplicación ya que la infraestructura es creada por la plataforma.

Con este modelo se consigue reducir bastante la complejidad a la hora de desplegar y mantener las aplicaciones ya que una solución PaaS gestiona automaticamente la escalabilidad utilizando más recursos si se considera necesario. De esta forma los desarrolladores estarán centrados en optimizar las aplicaciones para minimizar el consumo de recursos (peticiones, escrituras en disco, etc.)

↘ **Software-as-a-Service (SaaS):** Este concepto lleva presente desde hace unos años, pero en la actualidad se ha definido con mucha más claridad. Simplemente son servicios basados en la web. Ejemplos claros es Webmail de Gmail, CRM online, etc. Utilizando estos servicios web accedemos a través de un navegador y no se consigue atender al software, por lo que todo el mantenimiento, desarrollo, actualizaciones, copias de seguridad es responsabilidad del proveedor.

Ejemplos: Dropbox, SalesForce, Google Docs, Gmail….

11. Cloud Computing

El cloud computing, servicios en la nube o informática en la nube es un paradigma que permite servicios de computación a través de la red, y que habitualmente es Internet.

Sus comienzos fueron a través de los proveedores de servicios de internet a gran escala, como por ejemplo Google, Amazon AWS, Microsoft y otros que fueron construyendo sus propias infraestructuras. De ellos emergieron un sistema de recursos distribuidos de forma horizontal, y que fueron introducidos como servicios virtuales de TI escalados de forma masiva y manejados como recursos configurados y mancomunados de forma continua.

En el año 2002 fué Amazon Web Services, prevé un conjunto de servicios basados en la nube, e incluyendo almacenamiento, computación e incluso la inteligencia humana a través del Amazon Mechanical Turk. Posteriormente, Amazon lanza su Elastic Compute Cloud (EC2) como servicio comercial y que permite a las pequeñas empresas y los particulares poder alquilar equipos en los que se puedan ejecutar sus propias aplicaciones.

Características de la computación en la nube:

- **Agilidad:** Capacidad de mejora para ofrecer recursos tecnológicos al usuario por parte del proveedor.
- **Costo:** los distintos proveedores de la computación en la nube afirman que los costos se reduzcan. Un modelo pública en la nube convierte los gastos de capital en gastos de funcionamiento. Con ello se consigue reducir barreras de entrada, ya que las infraestructuras se proporcionan tipicamente por una tercera parte, y no tiene que ser adquirida por una sola vez o tareas informáticas intensivas infrecuentes.
- **Escalabilidad y elasticidad:** aprovisionamiento de recursos sobre una base de autoservicio en casi tiempo real, sin que los usuarios necesitan cargas de alta duración.
- **Independencia entre el dispositivo y la ubicación:** permite a los usuarios poder acceder a los sistemas usando un navegador web, independiente de su ubicación o del dispositivo que se utilice (ejemplo, PC, teléfono móvil).
- La **tecnología de virtualización permite compartir servidores y dispositivos de almacenamiento** y una mayor utilización. Las aplicaciones pueden ser fácilmente migradas de un servidor físico a otro.
- **Rendimiento:** Los sistemas en la nube controlan y optimizan el uso de los recursos de manera automática, dicha característica permite un seguimiento, control y notificación del mismo. Esta capacidad aporta transparencia tanto para el consumidor o el proveedor de servicio.

- Seguridad**:** puede mejorar debido a la centralización de los datos. La seguridad es a menudo tan buena o mejor que otros sistemas tradicionales, en parte porque los proveedores son capaces de dedicar recursos a la solución de los problemas de seguridad que muchos clientes no pueden permitirse el lujo de abordar. El usuario de la nube es responsable de la seguridad a nivel de aplicación. El proveedor de la nube es responsable de la seguridad física.4
- **Mantenimiento:** en el caso de las aplicaciones de computación en la nube, es más sencillo, ya que no necesitan ser instalados en el ordenador de cada usuario y se puede acceder desde diferentes lugares.

Ventajas

- Integración probada de servicios Red. Por su naturaleza, la tecnología de *cloud computing* se puede integrar con mucha mayor facilidad y rapidez con el resto de las aplicaciones empresariales (tanto software tradicional como Cloud Computing basado en infraestructuras), ya sean desarrolladas de manera interna o externa.5
- Prestación de servicios a nivel mundial. Las infraestructuras de *cloud computing* proporcionan mayor capacidad de adaptación, recuperación completa de pérdida de datos (con copias de seguridad) y reducción al mínimo de los tiempos de inactividad.
- Una infraestructura 100% de *cloud computing* permite también al proveedor de contenidos o servicios en la nube prescindir de instalar cualquier tipo de software, ya que éste es provisto por el proveedor de la infraestructura o la plataforma en la nube. Un gran beneficio del *cloud computing* es la simplicidad y el hecho de que requiera mucha menor inversión para empezar a trabajar.
- Implementación más rápida y con menos riesgos, ya que se comienza a trabajar más rápido y no es necesaria una gran inversión. Las aplicaciones del *cloud computing* suelen estar disponibles en cuestión de días u horas en lugar de semanas o meses, incluso con un nivel considerable de personalización o integración.
- Actualizaciones automáticas que no afectan negativamente a los recursos de TI. Al actualizar a la última versión de las aplicaciones, el usuario se ve obligado a dedicar tiempo y recursos para volver a personalizar e integrar la aplicación. Con el *cloud computing* no hay que decidir entre actualizar y conservar el trabajo, dado que esas personalizaciones e integraciones se conservan automáticamente durante la actualización.
- Contribuye al uso eficiente de la energía. En este caso, a la energía requerida para el funcionamiento de la infraestructura. En los datacenters tradicionales, los servidores consumen mucha más energía de la requerida realmente. En cambio, en las nubes, la energía consumida es sólo la necesaria, reduciendo notablemente el desperdicio.

Desventajas

- La centralización de las aplicaciones y el almacenamiento de los datos origina una interdependencia de los proveedores de servicios.
- La disponibilidad de las aplicaciones está sujeta a la disponibilidad de acceso a Internet.
- La confiabilidad de los servicios depende de la "salud" tecnológica y financiera de los proveedores de servicios en nube. Empresas emergentes o alianzas entre empresas podrían crear un ambiente propicio para el monopolio y el crecimiento exagerado en los servicios.6
- La disponibilidad de servicios altamente especializados podría tardar meses o incluso años para que sean factibles de ser desplegados en la red.
- La madurez funcional de las aplicaciones hace que continuamente estén modificando sus interfaces, por lo cual la curva de aprendizaje en empresas de orientación no tecnológica tenga unas pendientes significativas, así como su consumo automático por aplicaciones.
- Seguridad. La información de la empresa debe recorrer diferentes nodos para llegar a su destino, cada uno de ellos (y sus canales) son un foco de inseguridad. Si se utilizan protocolos seguros, HTTPS por ejemplo, la velocidad total disminuye debido a la sobrecarga que éstos requieren.
- Escalabilidad a largo plazo. A medida que más usuarios empiecen a compartir la infraestructura de la nube, la sobrecarga en los servidores de los proveedores aumentará, si la empresa no posee un esquema de crecimiento óptimo puede llevar a degradaciones en el servicio o altos niveles de jitter.

Software como servicio

El **software como servicio** (en inglés *software as a service*, SaaS) se encuentra en la capa más alta y caracteriza una aplicación completa ofrecida como un servicio, por-demanda, vía multitenencia —que significa una sola instancia del software que corre en la infraestructura del proveedor y sirve a múltiples organizaciones de clientes. Las aplicaciones que suministran este modelo de servicio son accesibles a través de un navegador web -o de cualquier aplicación diseñada para tal efecto- y el usuario no tiene control sobre ellas, aunque en algunos casos se le permite realizar algunas configuraciones. Esto le elimina la necesidad al cliente de instalar la aplicación en sus propios computadores, evitando asumir los costos de soporte y el mantenimiento de hardware y software.

Plataforma como servicio

La capa del medio, que es la **plataforma como servicio** (en inglés *platform as a service*, PaaS), es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal (persistencia de datos, autenticación, mensajería, etc.). De esta forma, un arquetipo de plataforma como servicio podría consistir en un entorno conteniendo una pila básica de sistemas, componentes o APIs preconfiguradas y listas para integrarse sobre una tecnología concreta de desarrollo (por

ejemplo, un sistema Linux, un servidor web, y un ambiente de programación como Perl o Ruby). Las ofertas de PaaS pueden dar servicio a todas las fases del ciclo de desarrollo y pruebas del software, o pueden estar especializadas en cualquier área en particular, tal como la administración del contenido.

Ejemplos comerciales son Google App Engine, que sirve aplicaciones de la infraestructura Google; Microsoft Azure, una plataforma en la nube que permite el desarrollo y ejecución de aplicaciones codificadas en varios lenguajes y tecnologías como .NET, Java y PHP o la Plataforma G, desarrollada en Perl. Servicios PaaS como éstos permiten gran flexibilidad, pero puede ser restringida por las capacidades disponibles a través del proveedor.

En este modelo de servicio al usuario se le ofrece la plataforma de desarrollo y las herramientas de programación por lo que puede desarrollar aplicaciones propias y controlar la aplicación, pero no controla la infraestructura.

Ventajas y desventajas

Las ventajas de los PaaS son que permite niveles más altos de programación con una complejidad extremadamente reducida; el desarrollo general de la aplicación puede ser más eficaz, ya que se tiene una infraestructura *built-in*; y el mantenimiento y mejora de la aplicación es más sencillo[7] También puede ser útil en situaciones en las que varios desarrolladores están trabajando en un mismo proyecto y que implican a partes que no están ubicadas cerca unas de otras. [8]

Una desventaja de PaaS es que es posible que los desarrolladores no pueden utilizar todas las herramientas convencionales (bases de datos relacionales, con joins irrestrictos, por ejemplo). Otra posible desventaja es estar cerrado en una cierta plataforma. Sin embargo, la mayoría de los PaaS están relativamente libres[9]

Tipos

Públicos, privados e híbridos

Existen varios tipos de PaaS, incluyendo públicos, privados e híbridos. [10] Paas fue originalmente pensado para las nubes públicas, antes de expandirse a las privadas e híbridas. [10] Los PaaS públicos son derivados de los Software como servicio (SaaS) y está situado entre SaaS y infraestructura como servicio (IaaS) [11]

Los PaaS privados son comúnmente descargados e instalados desde una infraestructura local de una empresa, o desde una nube pública. Una vez que el software se instala en una o más máquinas, el PaaS privado organiza la aplicación y los componentes de la base de datos en una sola plataforma para el alojamiento. [12] Entre los proveedores de PaaS se encuentran Apprenda, que comenzó en la plataforma Microsoft .NET; OpenShift, de Red Hat y su Cloud Foundry Pivotal.[13] Apprenda y Microsoft eran consideradas las dos únicas PaaS que proveían soporte .NET superior. [10] Ahora acompañadas por la anunciada [14] asociación entre Microsoft e IBM [15]

PaaS híbrido es típicamente un despliegue consistente en una mezcla de despliegues públicos y privados. Un ejemplo aquí es IBM Bluemix 16 que se entrega como una sola plataforma de nube integrada a través de modelos de despliegue público, dedicado y local.

Mobile PaaS

Iniciado en 2012, mobile PaaS (mPaaS) proporciona capacidades de desarrollo para diseñadores y desarrolladores de aplicaciones móviles.17 El Yankee Group identificó a mPaas como uno de sus temas para 2014, nombrando a varios proveedores incluyendo Kinvey, CloudMine, AnyPresence, FeedHenry, FatFractal y Point.io. 18 19

PaaS Abierto

PaaS abierto no incluye alojamiento, sino que proporciona software de código abierto que permite a un proveedor PaaS ejecutar aplicaciones en un entorno de código abierto. Por ejemplo, AppScale permite a un usuario desplegar algunas aplicaciones escritas para Google App Engine a sus propios servidores, proporcionando acceso a almacén de datos desde una base de datos SQL o NoSQL estándar. Algunas plataformas abiertas permiten al desarrollador utilizar cualquier lenguaje de programación, base de datos, sistema operativo o servidor para implementar sus aplicaciones.20 21

PaaS para el Desarrollo Rápido

En 2014, Forrester Research definió Plataformas empresariales públicas para desarrolladores rápidos como una tendencia emergente, nombrando a varios proveedores incluyendo a Mendix, Salesforce.com, OutSystems y Acquia Acquia.22

Tipos de Sistemas

PaaS se encuentra en los siguientes tipos de sistemas:

Instalaciones de desarrollo complementario
: Estas instalaciones permiten la personalización de aplicaciones SaaS existentes, a menudo requieren que los desarrolladores de PaaS y sus usuarios compren suscripciones a la aplicación SaaS complementaria. 23

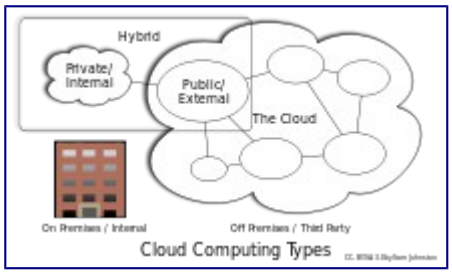
Entornos independientes
: Los entornos PaaS independientes no incluyen dependencias técnicas, de licencias o financieras de aplicaciones o servicios web específicos de SaaS, y están diseñados para proporcionar un entorno de desarrollo generalizado.9

Entornos de únicamente entrega de aplicaciones
: Los PaaS de unicamente entrega generalmente se enfocan en servicios de hosting, tales como seguridad y escalabilidad bajo demanda. El servicio no incluye capacidades de desarrollo, depuración y prueba, aunque pueden ser suministrados sin conexión (a través de un complemento de Eclipse, por ejemplo). 23

Infraestructura como servicio

La **infraestructura como servicio** (*infrastructure as a service*, IaaS) -también llamada en algunos casos *hardware as a service*, HaaS)24 se encuentra en la capa inferior y es un medio de entregar almacenamiento básico y capacidades de cómputo como servicios estandarizados en la red. Servidores, sistemas de almacenamiento, conexiones, enrutadores, y otros sistemas se concentran (por ejemplo a través de la tecnología de virtualización) para manejar tipos específicos de cargas de trabajo —desde procesamiento en lotes ("batch") hasta aumento de servidor/almacenamiento durante las cargas pico. El ejemplo comercial mejor conocido es Amazon Web Services, cuyos servicios EC2 y S3 ofrecen cómputo y servicios de almacenamiento esenciales (respectivamente). Otro ejemplo es Joyent, cuyo producto principal es una línea de servidores virtualizados, que proveen una infraestructura en demanda altamente escalable para manejar sitios web, incluidas aplicaciones web complejas escritas en Python, Ruby, PHP y Java.

Tipos de nubes



Tipos de computación en la nube

- Una nube pública es una nube computacional mantenida y gestionada por terceras personas no vinculadas con la organización. En este tipo de nubes tanto los datos como los procesos de varios clientes se mezclan en los servidores, sistemas de almacenamiento y otras infraestructuras de la nube. Los usuarios finales de la nube no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor, red, sistemas de almacenamiento, etc.25 Aplicaciones, almacenamiento y otros recursos están disponibles al público a través de el proveedor de servicios, que es propietario de toda la infraestructura en sus centros de datos; el acceso a los servicios sólo se ofrece de manera remota, normalmente a través de internet.

Dentro de estas nubes podemos encontrar englobadas las Cloud Privado virtual, 26 consideradas como nubes de dominio público pero que mejoran la seguridad de los datos. Estos datos se encriptan a través de la implantación de una VPN. Algunas grandes empresas como Amazon ya permiten aprovisionar una sección de su AWS con esta posibilidad VPC.

- Las nubes privadas son una buena opción para las compañías que necesitan alta protección de datos y ediciones a nivel de servicio. Las nubes privadas están en una infraestructura bajo demanda, gestionada para un solo cliente que controla qué aplicaciones debe ejecutarse y

dónde. Son propietarios del servidor, red, y disco y pueden decidir qué usuarios están autorizados a utilizar la infraestructura. Al administrar internamente estos servicios, las empresas tienen la ventaja de mantener la privacidad de su información y permitir unificar el acceso a las aplicaciones corporativas de sus usuarios.

- Las nubes híbridas combinan los modelos de nubes públicas y privadas. Un usuario es propietario de unas partes y comparte otras, aunque de una manera controlada. Las nubes híbridas ofrecen la promesa del escalado, aprovisionada externamente, a demanda, pero añaden la complejidad de determinar cómo distribuir las aplicaciones a través de estos ambientes diferentes. Las empresas pueden sentir cierta atracción por la promesa de una nube híbrida, pero esta opción, al menos inicialmente, estará probablemente reservada a aplicaciones simples sin condicionantes, que no requieran de ninguna sincronización o necesiten bases de datos complejas. Se unen mediante la tecnología, pues permiten enviar datos o aplicaciones entre ellas. Un ejemplo son los sistemas de correo electrónico empresarial.27

- Nube comunitaria. De acuerdo con Joyanes Aguilar en 2012,28 el Instituto Nacional de Estándares y Tecnología (NITS, por sus siglas en inglés) define este modelo como aquel que se organiza con la finalidad de servir a una función o propósito común (seguridad, política…), las cuales son administradas por las organizaciones constituyentes o terceras partes.

12. Contenedores (Docker)

Docker es un proyecto de código abierto y que automatiza el despliegue de aplicaciones dentro de contenedores de software, y proporciona una capa adicional de abstracción y automatización de Virtualización a nivel de sistema operativo en Linux. Docker usa unas características de aislamiento de recursos del kernel de Linux, como por ejemplo cgroups y espacios de nombres (namespaces) para poder permitir que “contenedores” independientes se ejecutan dentro de una sola instancia de Linux, y evitando la sobrecarga de iniciar y mantener máquinas virtuales.

El soporte del kernel de Linux para los espacios de nombres aísla de vista, en su mayoría, una aplicación del entorno operativo, incluyendo árboles de proceso, red, ID de usuario y sistemas de archivos montados, mientras que los cgroups del kernel proporcionan aislamiento de recursos, incluyendo la CPU, la memoria, el bloque de E / S y de la red. Desde la versión 0.9, Docker incluye la biblioteca libcontainer como su propia manera de utilizar directamente las facilidades de virtualización que ofrece el kernel de Linux, además de utilizar las interfaces abstraídas de virtualización mediante libvirt, LXC (Linux Containers) y systemd-nspawn.
Docker implementa una API de alto nivel para poder proporcionar contenedores livianos que ejecutan procesos de manera aislada.

Docker se puede integrar con diferentes herramientas de infraestructura, como Amazon Web Services, Ansible, Cfengine, Chef, Google Cloud Platform, DigitalOcean, IBM Bluemix, Jelastic, Jenkins, Microsoft Azure, OpenStack Nova, OpenSVC, Puppet, Salt, y Vagrant.

BIBLIOGRAFÍA:

→ Humble Devassy Chirammal, Prasad Mukhedkar, Anil Vettathu (2016). Mastering KVM Virtualization. *Packt Publishing Ltd.*

→ Reingeniería, Tecnología y Comunicaciones, S.L. Soft. De virtualización y sus aplicaciones: Xen sobre Linux, Vmware server – ESXi, Virtual PC

FUENTES WEB:

https://www.jonathanecheverria.com/2009/07/07/herramientas-de-virtualizacion-introduccion

https://help.ubuntu.com/lts/serverguide/libvirt.html

https://www.genbetadev.com/programacion-en-la-nube/entendiendo-la-nube-el-significado-de-saas-paas-y-iaas

http://www.vmware.com/pdf/vm_importer_manual.pdf

http://www.obasoft.es/CF/SIINF/SIINF_02_Contenidos/9_introduccin_a_las_mquinas_virtuales.html

https://antispyware.gratis/diferencias-entre-un-emulador-virtualizador-y-simulador-de-sistema-operativo/

https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube#Comienzos

https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube

https://es.wikipedia.org/wiki/Docker_(software)