

TEMA 2: VIRTUALIZACIÓN Y CLOUD COMPUTING

Índice

1. Introducción. Sistemas Informáticos.....	2
2. Virtualización.....	4
3. Conceptos en Virtualización	6
4. Arquitecturas. Tipos de Máquinas Virtuales.....	7
5. Ventajas y desventajas de la virtualización.....	13
6. Software de virtualización.....	15
7. Infraestructura de Escritorio Virtual (VDI).....	18
8. Virtual Private Servers (VPS)	19
9. Cloud Computing	21
10. Modelos de Cloud Computing.....	24
11. Ejemplos de soluciones de Cloud Computing.....	28
Amazon Web Services.....	29
OpenStack.....	29

1. Introducción. Sistemas Informáticos.

En este tema vamos a estudiar dos conceptos distintos entre ellos pero que están relacionados. La primera y mayor parte del tema la vamos a dedicar a la **virtualización** de sistemas. La segunda parte veremos qué es la computación en la nube (**cloud computing**) y cómo la podemos encontrar en un entorno real.

Hoy en día, tanto en las empresas (grandes y pequeñas), en las organizaciones y también a nivel de usuario particular, se necesitan ejecutar muchas aplicaciones SW que quizás sólo estén disponibles en un determinado sistema operativo (distinto al del resto de aplicaciones), necesitando tener más de un equipo con su SO correspondiente arrancado para poder ejecutar dichas aplicaciones.

A nivel doméstico, este problema lo podemos solucionar instalando 2 sistemas operativos en distintas particiones del disco y seleccionando al arrancar qué SO vamos a utilizar para trabajar. El principal inconveniente de este método es que hay que reiniciar la máquina cada vez que deseemos cambiar de sistema, y además las aplicaciones que estaban trabajando con el anterior sistema dejan de correr.

Esta solución sería impensable en un servidor de cualquier empresa u organización que no se puede permitir el parar. Lo normal es que estos servidores estén encendidos las 24 horas del día ininterrumpidamente. En este caso la solución debería pasar forzosamente por tener 2 máquinas arrancadas con sus sistemas operativos distintos (por ejemplo Windows y Linux).

El problema de tener dos máquinas distintas arrancadas es el consiguiente aumento de costes y de mantenimiento, además de desperdiciar recursos con máquinas que apenas tienen carga de trabajo pero que necesitan estar arrancadas continuamente.

La solución óptima pasaría por disponer de varios servidores de características distintas pero ubicados físicamente **en una misma máquina**, ahorrando costes y optimizando el mantenimiento entre otras ventajas, al tener una única máquina física. Esto lo conseguiremos mediante la **virtualización**.

La virtualización está cada vez más extendida en las organizaciones, pues de una manera rápida y sencilla se pueden crear servidores virtuales que se eliminan cuando dejan de ser necesarios, sin la necesidad de comprar nuevas máquinas.

Además de la virtualización, en las empresas, también está extendido otro modelo que permite aún mayor flexibilidad y mayor optimización de recursos y costes que es el de pagar sólo por aquellos recursos informáticos que se usan según las necesidades de la empresa.

Imaginemos que en un momento determinado una organización crece en el número de empleados de manera puntual debido a una carga de trabajo o cualquier otro motivo. Para atender ese pico de trabajo se necesitan servidores más potentes donde se alojan las aplicaciones, mejores infraestructuras de red para que no se colapse el sistema, más licencias de uso de los distintos programas, etc.

Si se decide invertir en mejorar el sistema informático puede ser que pasado un tiempo la empresa vuelva a la carga habitual de trabajo y se estén desperdiciando recursos.

Aparece entonces una solución válida tanto para pequeñas como grandes empresas que ayudan a reducir costes y a pagar sólo por aquello que se necesita, ahorrando por ejemplo en licencias, costes de mantenimiento, adquisición de equipos, etc.

Este modelo de tecnología es el **cloud computing** o computación en la nube que de manera sencilla se puede decir que es un conjunto de programas y servicios que se alojan en un servidor conectado a Internet y que es accesible desde cualquier ordenador.

De este modo una empresa puede contratar un potente servidor externo, añadir más capacidad de procesamiento o disco duro en un momento determinado, y pagar *por el alquiler* de este servicio. También podría pagar por tener un servidor web contratado y aumentar o disminuir las características de dicho servidor en base a las necesidades. O incluso pagar por ejecutar una aplicación instalada en un servidor externo.

Algo así es lo que hacemos cuando accedemos a nuestro correo de yahoo, gmail, etc o compramos almacenamiento en discos externos como Dropbox, Drive o iCloud.

2. Virtualización

En muchos momentos las capacidades de los ordenadores no se utilizan al máximo, de manera que dejan un margen para el aprovechamiento de estos recursos. Pero la cuestión es cómo se pueden utilizar estos recursos en determinados ámbitos si en la máquina ya tenemos instalado el sistema operativo y una serie de software.

Así de entrada, parecería que solo podemos optimizar los recursos de la máquina a base de instalar y ejecutar más software. La virtualización abre la puerta a aprovechar estos recursos de la máquina en todo tipo de situaciones, incluso a tener varios sistemas operativos dentro de la misma máquina y trabajando a la vez. La manera en que eso se puede materializar es lo que se conoce como virtualización.

La virtualización consiste en la abstracción de los recursos de la máquina para poder utilizar los que sobran y crear máquinas virtuales que utilizan este hardware como si estuviese perfectamente libre.

Entonces, a partir de la virtualización se puede compartir una máquina física para ejecutar varias máquinas virtuales, de manera que estas comparten los recursos libres de la máquina como la CPU, la memoria, el disco y la conexión de red que de otra manera estarían libres esperando un pico de trabajo.

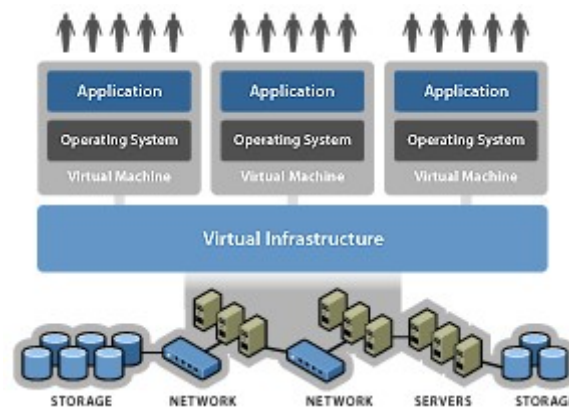
Además, la virtualización permite utilizar estas máquinas virtuales con independencia del hardware. Eso se consigue ocultando los detalles técnicos mediante el método de encapsulación.

En informática, una **máquina virtual** es un software que emula un ordenador y puede ejecutar programas como si fuera una computadora real.

Una máquina virtual permite tener varios ordenadores virtuales ejecutándose sobre el mismo ordenador físico. Eso se consigue mediante software, una capa de software que crea una capa de abstracción entre el hardware de la máquina física o host y el sistema de la máquina virtual. Este es el método para crear una versión virtual de un dispositivo

o recurso tanto si es todo un servidor o como si es simplemente un disco duro, una red o un sistema operativo.

La **virtualización** es un nivel de abstracción que oculta el hardware real de la máquina al sistema operativo. Permite la creación de múltiples máquinas virtuales donde se ejecutan diversos sistemas operativos de forma aislada en la misma máquina real. Cada una de las máquinas virtuales dispone de su propio hardware virtual en el cual se ejecutan las aplicaciones y el sistema operativo.



Una característica de las máquinas virtuales es que los procesos que ejecutan están limitados por los recursos y abstracciones proporcionados por ellas.

Una de las utilizaciones más extendidas de las máquinas virtuales consiste en ejecutar sistemas operativos para probarlos. De esta manera se puede ejecutar un sistema operativo que se quiera probar, por ejemplo, un Linux, desde el sistema operativo habitual, por ejemplo un Mac OS X, sin necesidad de instalarlo directamente en nuestra máquina y sin miedo que se desconfigure el sistema operativo primario.

El software de virtualización, nos permite trabajar con varios sistemas operativos a la vez ejecutándose simultáneamente. Su funcionamiento es sencillo y está concebido de tal manera que el sistema operativo que se ejecuta no se entere en ningún momento de que no es el único.

Cuando arranca, es como si estuviésemos viendo el arranque de un ordenador cualquiera:

- Hace el chequeo de los dispositivos, de la RAM,
- Nos permite acceder a la BIOS y modificarla
- Podemos arrancar con un CD que tenga auto arranque

Además, cada SO se ejecuta sobre una **máquina virtual** distinta en la que a la hora de crearla podemos configurar nosotros qué características tendrá esa máquina virtual (procesador, memoria, disco, unidades E/S, etc), de este modo podremos crear máquinas obsoletas donde instalar SO antiguos o adaptar las necesidades HW de cada máquina al SO que va a alojar.

Cada máquina virtual dispondrá de su procesador, memoria, disco, pero las máquinas virtuales que creamos nunca podrán ser superiores a la máquina real en cuanto a prestaciones. Si disponemos de un ordenador con 4 Gb de memoria RAM no podremos crear una máquina virtual que disponga de 8 Gb de RAM. Ni tan siquiera de 4 Gb, pues la memoria real se deberá repartir entre la máquina real y la(s) máquina(s) virtual(es). Cuanta más memoria asignemos a una máquina virtual, menos memoria libre dispondremos en la máquina real.

Una máquina virtual se compone exclusivamente de software y se **almacena** en un directorio de la máquina real (anfitrión). Esta máquina se comporta igual que lo hace una máquina física.

3. Conceptos en Virtualización

Para poder empezar a trabajar con la virtualización hemos de definir una serie de conceptos.

En primer lugar definimos el **hipervisor**. Un hipervisor, también llamado monitor de máquina virtual (*virtual machine monitor*, VMM), es una plataforma de virtualización que permite utilizar, a la vez, múltiples sistemas operativos en un equipo. Será el SW de virtualización que utilizaremos.

Cuando estemos virtualizando distinguiremos entre **máquina host** y **máquina guest**. Del mismo modo hablaremos de los SO que corren en esas máquinas como **sistema operativo host** y **sistema operativo guest**.

Máquina o sistema operativo:

HOST o anfitrión → Es la máquina real y es el sistema operativo donde instalaremos el SW de virtualización, el hipervisor.

GUEST o huésped → Cualquiera de las máquinas virtuales que creamos con su sistema operativo correspondiente.

Así, si tenemos una máquina corriendo con Windows 7 y queremos crear dentro de ella una máquina virtual con un SO Ubuntu 14.4, el SO anfitrión (o host) será el Windows 7 y el SO huésped (o guest, o invitado) será el Linux.

4. Arquitecturas. Tipos de Máquinas Virtuales

La abstracción del hardware de un ordenador como si fuera una máquina totalmente autónoma a partir del encapsulamiento proporcionado por la capa de software renombrado máquina virtual se puede producir básicamente de dos maneras diferentes. Esta clasificación se produce según la funcionalidad de la máquina virtual y según su grado de equivalencia a una verdadera máquina autónoma.

Entonces hay dos tipos de máquinas virtuales: las **máquinas virtuales de proceso** y las **máquinas virtuales de sistema**.

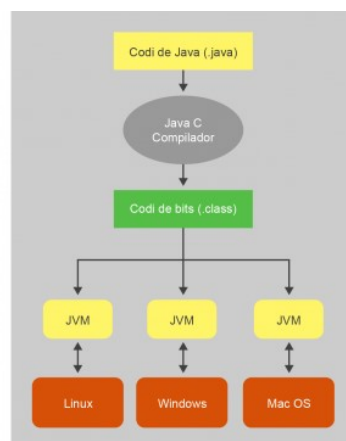
Según esta clasificación, las máquinas virtuales de proceso sirven para virtualizar sólo determinados procesos, mientras que las máquinas virtuales de sistema permiten alojar, por ejemplo, todo un sistema operativo huésped gracias al software anfitrión de la máquina virtual (el ejemplo del que estamos hablando durante todo el tema).

Máquinas virtuales de proceso

La máquina virtual de proceso, también conocida como máquina virtual de aplicación, se ejecuta como un proceso normal dentro de un sistema operativo y soporta un solo

proceso. La máquina se inicia automáticamente cuando se lanza el proceso que se quiere ejecutar y se para cuando este proceso ha finalizado. El objetivo de esta máquina es proporcionar un entorno de ejecución independiente de la plataforma de hardware y del sistema operativo, de manera que esconda los detalles de la plataforma subyacente y permita que un programa se ejecute siempre de la misma manera sobre cualquier tipo de plataforma.

Uno de los ejemplos más conocidos de este tipo de máquina virtual es la máquina virtual de **Java**. Otro ejemplo sería la del entorno .NET de Microsoft, que se llama common language runtime. Otros ejemplos de máquinas virtuales de proceso son los siguientes: Macromedia Flash Player, SWF, Perl virtual machine, Perl y un largo etcétera.



Funcionamiento de la JVM

Muchos de estos ejemplos corresponden en máquinas virtuales dedicadas a poder ejecutar varios procesos de diferentes lenguajes de programación. La finalidad siempre es abstraer el hardware del proceso a ejecutar; de esta manera, se puede programar el proceso sin tener en cuenta en qué tipo de máquina se ejecutará. En todos, la estructura de la máquina virtual es similar.

A parte de las máquinas virtuales dedicadas a ejecutar lenguajes de programación en diferentes entornos de hardware, también hay las máquinas virtuales de proceso que se dedican a simular hardware, de manera que permiten que determinadas aplicaciones pensadas para otras arquitecturas de procesador se puedan ejecutar sobre un hardware que en teoría no soportan.

Máquinas virtuales de sistema

Las **máquinas virtuales de sistema**, o también llamadas **máquinas virtuales de hardware**, permiten en la máquina física subyacente multiplexarse en varias máquinas virtuales, cada una ejecutando el propio sistema operativo.

La capa de software que permite la virtualización se denomina monitor de máquina virtual, MMV o también **Hipervisor**.

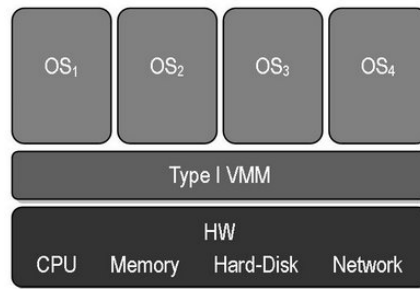
Esta capa de software MMV gestiona los cuatro recursos principales de un ordenador:

- CPU
- Memoria
- Red
- Almacenamiento

Con esta técnica de la emulación del hardware se consigue lo que se conoce como virtualización completa. Eso permite que cada máquina virtual pueda ejecutar cualquier sistema operativo soportado por el hardware de la computadora (en las técnicas más modernas ya no hace falta que el sistema operativo de la máquina virtual sea compatible con el hardware del ordenador original); de esta manera los usuarios de la máquina pueden ejecutar simultáneamente dos o más sistemas operativos diferentes en máquinas virtuales. Una máquina virtual de sistema permite tener varios ordenadores virtuales ejecutante sobre el mismo ordenador físico.

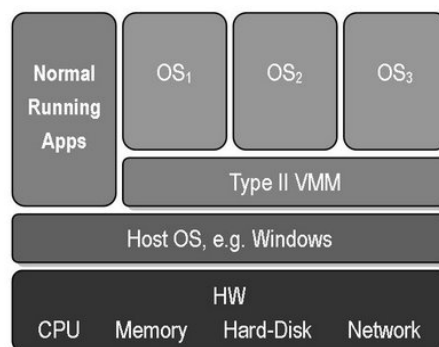
Un Monitor de Máquina Virtual (MMV) admite varios tipos de arquitectura:

Clásica. En esta arquitectura el monitor de la máquina virtual se ejecuta directamente sobre el hardware (*Bare Metal*). Entonces los sistemas operativos huéspedes corren directamente sobre la capa de software que hay encima del hardware, el MMV o el hipervisor. Este sistema se conoce como el hipervisor de tipo I (Hipervisor I).



Esta arquitectura no carga excesivamente el sistema, permite una buena optimización de los recursos de la máquina, pero requiere que los sistemas operativos que se instalan en la máquina virtual sean compatibles con el hardware de la máquina. Como ejemplo de máquina virtual del tipo hipervisor I se encuentra la versión ESX de VMWare.

Indirecta. En esta arquitectura, el monitor de la máquina virtual se instala encima del sistema operativo que hay en la máquina, en lugar de hacerlo directamente encima del hardware, y los sistemas operativos virtuales se ejecutan encima del MMV. De eso viene el nombre de indirecta. También se conoce con el nombre de Hipervisor II. Actualmente, esta arquitectura es una de las más utilizadas, y se pueden encontrar muchas máquinas virtuales que la utilizan. En este sistema, se está ejecutando el sistema operativo anfitrión y sus aplicaciones y además se ejecuta la máquina virtual como una aplicación más, y los diferentes sistemas operativos son instancias de la máquina virtual.



En esta arquitectura, igual que pasa con la de hipervisor I, sólo se pueden instalar sistemas operativos virtuales que sean compatibles con la arquitectura del procesador y otros recursos de la máquina.

Algunos ejemplos de hipervisores que utilizan este sistema son: VmWare, VirtualBox, Microsoft Virtual PC y un largo etcétera. Algunos de ellos son de pago, como VmWare, Microsoft Virtual PC y Oracle Virtual Box, y otros son de código libre, como VirtualBoxOSE. VmWare y Virtualbox ofrecen versiones gratuitas, aunque en el caso de la VMWare, con menos prestaciones. Oracle, la propietaria actual de VirtualBox ofrece versiones de código libre como la VirtualBoxOSE o bien un uso personal restringido para la versión Oracle VirtualBox. Algunas de estos hipervisores de código libre pueden correr en sistemas operativos privativos, como el Windows, o bien en los de código libre como el Linux.

Entre los inconvenientes de esta arquitectura, encontraríamos el hecho de que el hipervisor de tipo II es menos eficiente que el de tipo I, pero sobretodo que no se puede instalar un sistema operativo que no sea compatible con la arquitectura del procesador de la máquina.

Paravirtualización. El hecho de que en los sistemas de Hipervisor I y II sólo se pueden instalar sistemas operativos en la máquina virtual que sean compatibles con la arquitectura del procesador (la mayoría de veces x86), y que además comporta una disminución en el rendimiento del sistema operativo virtual de cerca del 20% en la mayoría de casos, es un inconveniente en el mundo de la virtualización.

Ante este hecho aparece una técnica que se conoce como paravirtualización, (o modelo de máquina paravirtual o PVM) que representa una solución intermedia en el camino de poder virtualizar cualquier sistema en cualquier hardware y hacerlo prácticamente sin penalización en el rendimiento del sistema operativo virtualizado, o bien con penalizaciones muy bajas de cerca del 2%-8%.

Para conseguir este funcionamiento, la máquina virtual se instala directamente sobre el hardware, haciendo lo que se conoce como una instalación de bajo nivel, muy cerca del mismo hardware. Entonces, esta técnica requiere modificar los sistemas operativos huéspedes para ser instalados en la máquina virtual. Con la paravirtualización, el sistema operativo huésped (al estar modificado) se comunica muchas veces con el hardware real directamente, quedando el hipervisor como una fina capa que apenas

interviene. El inconveniente principal de la paravirtualización es que necesita sistemas operativos modificados y aunque eso pasa desapercibido para el usuario, se reduce el abanico de posibilidades de los sistemas huéspedes.

Un ejemplo importante de la paravirtualización es Xen. Xen es una máquina virtual de código libre que proporciona virtualizaciones de alto rendimiento sobre máquinas de arquitectura x86, pero hay una limitación de sistemas operativos modificados aptos por ser instalados en la máquina virtual Xen.

Actualmente hay algunos Linux, NetBSD y una versión inicial del Windows XP, pero que no se puede lanzar al público porque necesita licencia, como sistemas operativos disponibles para la máquina Xen.

Aun así, esta máquina virtual tiene sus utilidades y aplicaciones, de manera que las compañías principales fabricantes de procesadores, Intel y AMD, ya se han apresurado a sacar versiones de sus procesadores compatibles con la máquina virtual Xen de manera que no hará falta adaptar los sistemas operativos a la máquina virtual, y eso permitirá instalar cualquier sistema operativo con un alto rendimiento.

Full virtualization. El hecho de que los hipervisores I y II requieran que el sistema operativo a virtualizar sea compatible con el hardware de la computadora, y que el rendimiento de este sistema virtual se vea penalizado, a menos que se utilice la paravirtualización con la limitación correspondiente de los sistemas a virtualizar, hace que la virtualización tenga trabas por alcanzar la situación en que se pueda virtualizar cualquier sistema operativo en cualquier arquitectura de hardware. Eso es el que se conoce como *full virtualization* o virtualización completa. La cuestión es conseguir máquinas virtuales que oculten el hardware y además lo emulen por poder instalar todo tipo de sistemas operativos en cualquier hardware, sea cual sea la arquitectura del procesador.

En esta arquitectura, la máquina virtual se instala directamente sobre el hardware (Hipervisor tipo I), los sistemas operativos virtuales se ejecutan sobre la máquina virtual, y esta emula el hardware cuando este no es compatible con el sistema operativo.

Actualmente los fabricantes de procesadores están haciendo modelos de procesadores con la máxima compatibilidad con las máquinas virtuales (AMD-V, Intel VT-x) para facilitar la virtualización total o completa.

Como ejemplos de máquinas virtuales que pueden hacer virtualización completa hay Mac-on-Linux, Parallels Workstation, algunas versiones de VmWare, como la VmWare GSX Server, Proxmox y algunas otras. En el caso de Vmware, utiliza una técnica conocida como traslación binaria para modificar automáticamente las instrucciones de software x86 que simula una virtualización completa. Proxmox utiliza dos técnicas: KVM y OpenVZ.

Existen muchas técnicas de virtualización, haciendo que esta clasificación que acabamos de ver quede un poco difusa. Por ejemplo, los hipervisores tipo I, los que se ejecutan directamente sobre el hardware, realmente llevan un kernel de sistema operativo embebido, y permiten virtualizar casi cualquier tipo de SO por lo que se les conoce a veces como virtualización completa (full virtualization).

5. Ventajas y desventajas de la virtualización

El utilizar una tecnología de virtualización hemos visto que conlleva una serie de ventajas, pero evidentemente, también de inconvenientes.

Ventajas

- Aunque tengamos múltiples máquinas sólo hemos de mantener el HW de una.
- Podemos tener varios sistemas operativos en funcionamiento simultáneamente, por ejemplo, para probar un sistema operativo nuevo sin necesidad de instalarlo directamente. De esta manera el MMV ofrece un entorno a ejecución completo, es decir, ofrece la posibilidad de instalar un nuevo sistema operativo, con sus aplicaciones, sus usuarios, su gestión del disco, de la red, etc.
- Permite restaurar la configuración en pocos segundos después de una prueba. Es decir, podemos hacer una captura (llamada snapshot o instantánea) de la configuración actual de la máquina en un momento dado, realizar pruebas y posteriormente restaurar en un instante la configuración anterior.

- Permite virtualizar HW no disponible físicamente, incluso obsoleto y de difícil obtención.
- Consolidación de servidores. Varias máquinas virtuales, cada una con su propio sistema operativo huésped o guest, se pueden utilizar para consolidar servidores. Eso permite que servicios que normalmente se deberían ejecutar en computadoras diferentes para evitar interferencias, se pueden ejecutar en la misma máquina de manera completamente aislada y compartiendo los recursos de una única computadora. Muchas veces, la consolidación de servidores contribuye a reducir el coste total de las instalaciones necesarias para mantener los servicios, ya que permiten ahorrar en hardware.
- Optimización de recursos. Hoy en día, la virtualización es una opción muy buena, ya que en la mayoría de los casos las máquinas actuales están siendo “infrautilizadas”, ya que disponen de una gran capacidad de disco duro, de memoria, etc. En muchos de los casos llegan a utilizar solo en un 30% o 60% de la capacidad de sus recursos. Virtualizar la necesidad de nuevas máquinas en una que ya existe permite un ahorro considerable de los gastos asociados como energía, mantenimiento, espacio físico, etc.

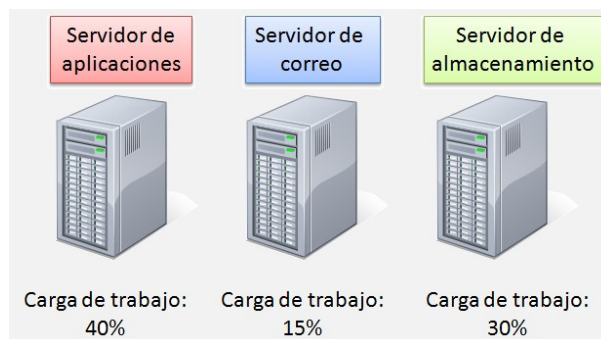


Figura 1. Tres servicios ejecutándose en tres equipos reales distintos. Sistema sobredimensionado.

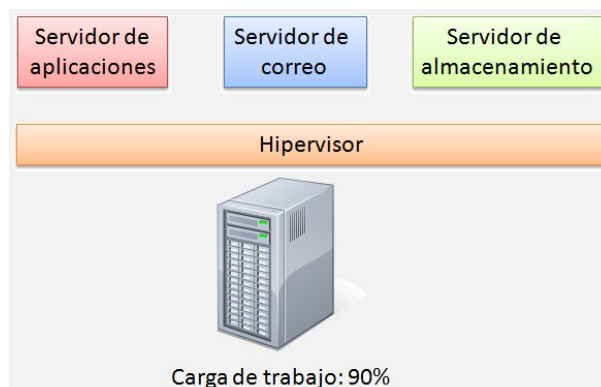


Figura 2. Tres servicios ejecutándose sobre tres servidores virtualizados distintos sobre el mismo hardware.

- Los sistemas virtualizados pueden ser 'portados' a otro equipo físico de una manera muy sencilla. De hecho, en entornos de alta disponibilidad se crean clústers de virtualización para poder ejecutar las máquinas virtuales en un equipo físico u otro en caso de que haya incidencias en el hardware.

Desventajas

- La complejidad añadida a la ejecución del sistema operativo invitado, ya que hay capas intermedias hasta llegar al hardware.
- La pérdida de prestaciones ocasionada por esas capas intermedias entre el sistema invitado y el hardware, además de las limitaciones impuestas por la compartición de los recursos hardware entre diferentes sistemas operativos. No obstante, los esquemas de virtualización bare metal 'aligeran' esas capas intermedias hasta valores que apenas alteran el rendimiento del sistema.
- Al tener varios sistemas operativos sobre una misma máquina provoca un mayor consumo de recursos en dicha máquina.
- Se necesita mayor memoria RAM y disco duro suficiente para albergar las máquinas virtuales.
- Si la máquina anfitrión cae, caen todas las máquinas virtuales.

6. Software de virtualización

Los principales fabricantes del mercado de la virtualización son VMware, Citrix, Oracle y Microsoft.

En la actualidad hay grandes alternativas en el mercado de la virtualización, siendo VMware el líder indiscutible con la mayor parte del mercado de sistemas virtualizados en producción.

VMware es software privativo, de pago, aunque dispone de herramientas gratuitas para poner en marcha máquinas virtuales (no crearlas ni configurarlas). Existen también programas libres como VirtualBox, de Oracle y programas gratuitos con Windows Virtual PC (Microsoft Virtual PC).

VMware

VMWare es un producto comercial que presenta gran variedad de productos, algunos de ellos gratuitos. Es el líder del mercado en tecnologías de virtualización.

Ofrece muchas soluciones dependiendo del entorno al que van dirigidos. Desde soluciones de escritorio con Hipervisores tipo II (Fusion, Workstation) hasta soluciones profesionales de servidores con Hipervisores tipo I de virtualización completa (VMware vSphere) o virtualización de escritorios (VMware Horizon)

Algunos de esos productos son gratuitos (vSphere Hypervisor).

VirtualBox

VirtualBox originalmente fue desarrollado por la empresa Innotek, que fue adquirida por la empresa Sun Microsystems y a su vez esta fue comprada por Oracle.

Es libre (GNU/GPL) y su instalación es sencilla y rápida. Es un Hipervisor tipo II (indirecta) que sólo funciona en arquitecturas x-86 y x86-64. Es el que utilizaremos en clase.

VirtualPC

Anteriormente llamado Microsoft VirtualPC, actualmente le han cambiado el nombre por Windows VirtualPC en Windows 7. Es gratuito.

Microsoft Windows Server 2008 ya incorporaba el programa Hyper-V, en lugar del Windows Virtual PC.

Hyper-V es un hipervisor integrado en los sistemas operativos Windows Server 2008, R2, 2012 y 2016. Este servicio de virtualización está al nivel de los ofrecidos por VMWare en lo referente a funcionalidad y capacidades, siendo además netamente superior a productos como VirtualBox.

La versión incorporada en Windows Server 2012 o 2016 permite utilizar discos duros virtuales de hasta 64TB, asignar hasta 1TB de RAM a cada máquina virtual y que cada una de estas pueda correr hasta sobre 64 microprocesadores.

Este sistema de virtualización puede hallarse en dos variantes de implementación diferenciadas:

- Como un rol a añadir a un equipo con Windows Server 2012, 2012 R2 o 2016.
- Como un sistema gratuito independiente llamado Microsoft Hyper-V Server 2012.

Son numerosos los requisitos hardware de este sistema de virtualización, especialmente a nivel de compatibilidad con la CPU.

KVM

Kernel-based Virtual Machine o KVM, (en español, Máquina virtual basada en el núcleo). Fue creado, y está mantenido, por Qumranet.

Es una solución para implementar virtualización completa (*Full Virtualization*) con Linux. Está formada por un módulo del núcleo (con el nombre `kvm.ko`) y herramientas en el espacio de usuario, siendo en su totalidad software libre. El componente KVM para el núcleo está incluido en Linux desde la versión 2.6.20.

KVM permite ejecutar máquinas virtuales utilizando imágenes de disco que contienen sistemas operativos sin modificar. Cada máquina virtual tiene su propio hardware virtualizado: una tarjeta de red, discos duros, tarjeta gráfica, etc.

PROXMOX

Proxmox es un sistema de virtualización *baremetal* (hipervisor tipo 1) que ofrece virtualización completa (*Full Virtualization*). Es una plataforma de virtualización abierta basada en Debian. Esta plataforma de virtualización consiste en un sistema operativo muy ligero instalado directamente sobre el equipo físico con un hipervisor embebido, sobre el que se crean las máquinas virtuales.

Algunas características que la hacen mejor que otras son:

- Su elevado rendimiento.
- Su sencillez de manejo.
- Es una herramienta gratuita.

Para instalar Proxmox debemos disponer de un equipo de 64 bits. La obligatoriedad de la arquitectura de 64 bits viene dada porque no existe una versión para 32 bits de Proxmox (esto va a ser una tónica cada vez más habitual en equipos profesionales ya que las limitaciones de RAM que imponen los 32 bits hacen inviable su utilización como equipos servidores). Por otra parte, necesitamos un disco duro físico dedicado, ya que Proxmox va a disponer de él completamente, por lo que si tenemos otras particiones con datos, las perderemos.

Proxmox utiliza internamente dos tecnologías: Kernel-Based Virtual Machine (la de KVM) y la de Container-based virtualization (OpenVZ), haciendo que sea más potente y flexible que otras soluciones.

XEN

Es un VMM desarrollado por la Universidad de Cambridge de código abierto.

Se utiliza en paravirtualización, esto requiere de SO guests modificados. También en algunos procesadores se puede utilizar para hacer la virtualización completa (*full virtualization*) sin necesidad de modificar los SO a instalar.

7. Infraestructura de Escritorio Virtual (VDI)

Un esquema de virtualización muy interesante, que conviene estudiar, es el conocido como Infraestructura de Escritorio Virtual (VDI) o simplemente Virtualización de Escritorio.

Cuando hablamos de virtualización pensamos automáticamente en la virtualización de servidores en nuestras máquinas del CPD (o también en máquinas externas como veremos en puntos posteriores), pero no pensamos en la posibilidad de virtualizar equipos de escritorio. El VDI consiste en virtualizar máquinas de escritorio (máquinas con Windows 10 por ejemplo) en los servidores de virtualización, y acceder a esas máquinas remotamente desde distintos dispositivos. Para poder acceder a estas máquinas virtuales se necesita tener instalado en los dispositivos de acceso un cliente de escritorio virtual.

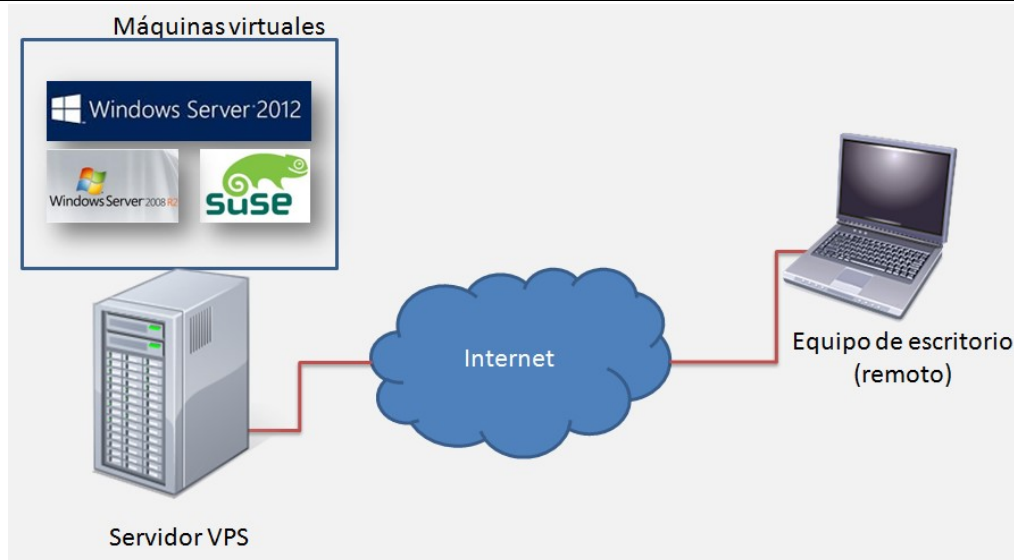
Imaginemos un usuario con mucha movilidad (un agente comercial, por ejemplo) que necesita conectarse a su puesto de trabajo (por ejemplo con Windows 10) continuamente. La solución inmediata pasaría por llevar su puesto de trabajo detrás (por ejemplo un ordenador portátil). Pero VDI nos proporciona otra solución, que sería la de virtualizar una máquina con Windows 10 en los servidores de virtualización de la empresa y acceder remotamente a esa máquina mediante un cliente de VDI instalado en los distintos dispositivos de acceso (por ejemplo un móvil, un portátil o una tableta). En todos los casos se le abriría una sesión en su máquina virtual alojada en el servidor, viendo el mismo escritorio, los mismos programas, etc, pues sólo es una máquina con múltiples accesos.



VMWare proporciona una solución VDI llamada Horizon 7 con clientes para dispositivos Windows, Mac, Linux, iOS y Android.

8. Virtual Private Servers (VPS)

Otro esquema de virtualización muy utilizado en entornos corporativos es el denominado Virtual Private Server (VPS). Este modo de funcionamiento consiste en contratar con un proveedor de servicios externo el alquiler de un servidor sobre el que pondremos en funcionamiento nuestras propias máquinas virtuales. En la siguiente figura se muestra un esquema de la estructura de un sistema de este tipo.



Una de las grandes ventajas de este tipo de implantación de nuestros servidores, consiste en que las labores de mantenimiento físico de los sistemas, así como el cumplimiento con los diferentes niveles de seguridad y redundancia, corren a cargo del proveedor del servicio, lo que por un lado nos descarga de labores de administración en nuestro entorno de trabajo, y por otro lado consigue optimizar los costes de la infraestructura, ya que es mucho más caro poner en marcha n pequeños CPDs con su doble acometida eléctrica y de datos, su equipamiento de aire acondicionado, su control de accesos, sus equipos electrógenos, etc. que poner en marcha un CPD de capacidad n veces la de los CPD pequeños.

Por otra parte, como desventaja cabe destacar que la integridad de los datos y la confidencialidad de estos atraviesa más puntos críticos, al ser otra empresa la que se dedica a albergar nuestros servidores.

La elección de un tipo u otro de implantación de los sistemas dependerá de las necesidades de la organización y de los requisitos de servicio de esta.

Existen numerosos proveedores de sistemas VPS como por ejemplo [CloudShare](#), [vps.net](#), [GoDaddy](#), [gigas](#), [dotblock](#), [Amazon Web Services](#), etc

El VPS tiene mucho que ver con la tecnología que vamos a estudiar en el siguiente punto, el cloud computing.

9. Cloud Computing

Cloud computing (computación en la nube) se refiere a las aplicaciones y servicios que se ejecutan en una red distribuida utilizando recursos virtualizados y accedidos por protocolos comunes de Internet y estándares de redes. Se distingue por la noción de que los recursos son virtuales y sin límites y que los detalles de los sistemas físicos en los que se ejecuta el software se abstraen al usuario. Cuando un usuario ejecuta una aplicación en la nube desconoce por completo el hardware utilizado, la plataforma sobre la que corre, etc. Cuando almacenamos datos en Dropbox, o leemos el correo en Gmail o editamos una hoja de cálculo en Drive desde un ordenador o una tableta, desconocemos los sistemas físicos sobre los que trabajamos, ni la plataforma, ni la ubicación de los servidores, ni nada por el estilo. Lo vemos de manera abstracta. Una nube.



La computación en la nube representa un verdadero cambio de paradigma en la forma en que los sistemas se despliegan. El aumento de los sistemas de computación en la nube ha sido posible por la popularización de Internet y el crecimiento de algunas de las grandes empresas de servicios. La computación en la nube posibilita el hecho de pagar sólo por lo que se necesita, sin necesidad de contratar más que lo estrictamente necesario (pagamos por el almacenamiento que necesitamos, por las licencias que necesitamos, por las máquinas que necesitamos, ...). Con cloud computing se puede comenzar de forma muy pequeña y convertirse rápidamente en una gran plataforma que satisfaga nuestras necesidades.

Como ejemplo, a nivel doméstico, podemos contratar almacenamiento en la nube en una plataforma como Dropbox. Podemos comenzar con una cuenta gratuita, e ir creciendo

conforme vamos necesitando almacenamiento y pagar por aquello que estrictamente necesitamos. No más. Traslademos este ejemplo a nivel corporativo. Podemos contratar una infraestructura de servidores y almacenamiento e ir aumentándola (o disminuyéndola) conforme las necesidades vayan creciendo (o decreciendo).

No todas las aplicaciones se benefician del despliegue en la nube. Problemas con latencia, control de transacciones, y en particular, la seguridad y el cumplimiento de la legislación son de particular preocupación.

Cloud computing tiene la tecnología, los servicios y las aplicaciones que son similares a los de Internet y los convierte en una herramienta de auto-servicio. El uso de la palabra "nube" hace referencia a los dos conceptos esenciales:

- **Abstracción:** La computación en la nube abstrae los detalles de la implementación del sistema a usuarios y desarrolladores. Las aplicaciones se ejecutan en los sistemas físicos que no están especificados, los datos se almacenan en lugares que se desconocen, la administración de sistemas se subcontrata a otros, y el acceso de los usuarios es ubicuo.
- **Virtualización:** La computación en la nube virtualiza los sistemas poniendo en común recursos y compartiéndolos. Los sistemas y el almacenamiento se pueden incrementar cuando sea necesario desde una infraestructura centralizada, los costos se controlan de forma dosificada y los recursos son escalable de una manera ágil.

Para ayudar a aclarar cómo la computación en nube ha cambiado la naturaleza de la implementación del sistema comercial, vamos a ver tres ejemplos:

- **Google:** En la última década, Google ha construido una red mundial de centros de datos para el servicio de su motor de búsqueda. Los ingresos por publicidad han permitido a Google ofrecer software gratuito para los usuarios en función dichas infraestructuras y ha cambiado el mercado de software de cara al usuario. Este es el claro ejemplo de Software como Servicio que veremos más adelante.
- **Plataforma Azure:** Por el contrario, Microsoft ha creado la plataforma Azure. Esta plataforma habilita que las aplicaciones que corren bajo el framework

.NET se ejecuten a través de Internet como una alternativa a la plataforma que se ejecuta en ordenadores de sobremesa.

- **Amazon Web Services (AWS):** Uno de los más exitosos negocios basados en la nube es Amazon Web Services, que es una Infraestructura como un Servicio que permite alquilar ordenadores virtuales de la infraestructura propia de Amazon.

Las nubes pueden tener muchos tipos diferentes, y los servicios y aplicaciones que se ejecutan en las nubes pueden o no ser entregados por un proveedor de servicios cloud. Estos diferentes tipos y niveles de servicios en la nube significan que es importante definir con qué tipo de sistema de computación en la nube vamos a trabajar.

Los modelos los veremos en el siguiente punto. Veamos ahora los tipos de nube.

Dependiendo de la naturaleza de la nube y de dónde está localizada, encontramos los siguientes tipos:

- **Nubes públicas:** los servicios que ofrecen se encuentran en servidores externos al usuario, pudiendo tener acceso a las aplicaciones de forma gratuita o de pago. Se manejan por terceras partes, y los trabajos de muchos clientes diferentes pueden estar mezclados en los servidores, los sistemas de almacenamiento y otras infraestructuras de la nube. Los usuarios finales no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor, red, discos como los suyos propios. La ventaja más clara de las nubes públicas es la capacidad de procesamiento y almacenamiento sin instalar máquinas localmente, por lo que no tiene una inversión inicial o gasto de mantenimiento en este sentido, si no que se paga por el uso. La carga operacional y la seguridad de los datos (backup, accesibilidad, etc.) recae íntegramente sobre el proveedor del hardware y software.
- **Nubes privadas:** las plataformas se encuentran dentro de las instalaciones del usuario de la nube y no suele ofrecer servicios a terceros. Son una buena opción para las compañías que necesitan alta protección de datos y ediciones a nivel de servicio. Como ventaja de este tipo de nubes, al contrario que las públicas, es la localización de los datos dentro de la propia empresa, lo que conlleva a una mayor seguridad de estos, corriendo a cargo del sistema de información que se

utilice. Incluso será más fácil integrar estos servicios con otros sistemas propios. Las nubes privadas están en una infraestructura local manejada por un solo cliente que controla qué aplicaciones debe correr y dónde. Son propietarios del servidor, red, y disco y pueden decidir qué usuarios están autorizados a utilizar la infraestructura. Sin embargo, como inconveniente se encuentra la inversión inicial en infraestructura física, sistemas de virtualización, ancho de banda y seguridad, lo que llevará a su vez a pérdida de escalabilidad y desescalabilidad de las plataformas, sin olvidar el gasto de mantenimiento que requiere. En la mayoría de los casos, este modelo de implementación es idéntico al de la infraestructura de TI antigua, mientras que utiliza tecnologías de virtualización y administración de aplicaciones para intentar incrementar el uso de los recursos. Se le conoce también como implementación *on-premise*.

- **Nubes híbridas:** combinan los modelos de nubes públicas y privadas. Esto permite a una empresa mantener el control de sus principales aplicaciones, al tiempo de aprovechar el cloud computing en los lugares donde tenga sentido. El usuario es propietario de unas partes y comparte otras, aunque de una manera controlada.

10. Modelos de Cloud Computing

Los modelos de servicio del cloud computing describen el tipo de servicio que el proveedor de servicios está ofreciendo. Los modelos de servicios más conocidos son el Software como Servicio (SaaS), Plataforma como Servicio (PaaS) e Infraestructura como un Servicio (IaaS). Los modelos de servicio se construyen uno sobre otro y definen lo que un proveedor debe manejar y lo que es responsabilidad del cliente.

Podemos encontrarnos otros modelos de servicio y todos toman la siguiente forma:

XaaS, “X” as a Service, (“Algo” como Servicio).

Ese “algo” puede ser un hardware, una plataforma de ejecución de aplicaciones o un software.

Es por ello que los tres modelos de servicio comúnmente aceptados son:

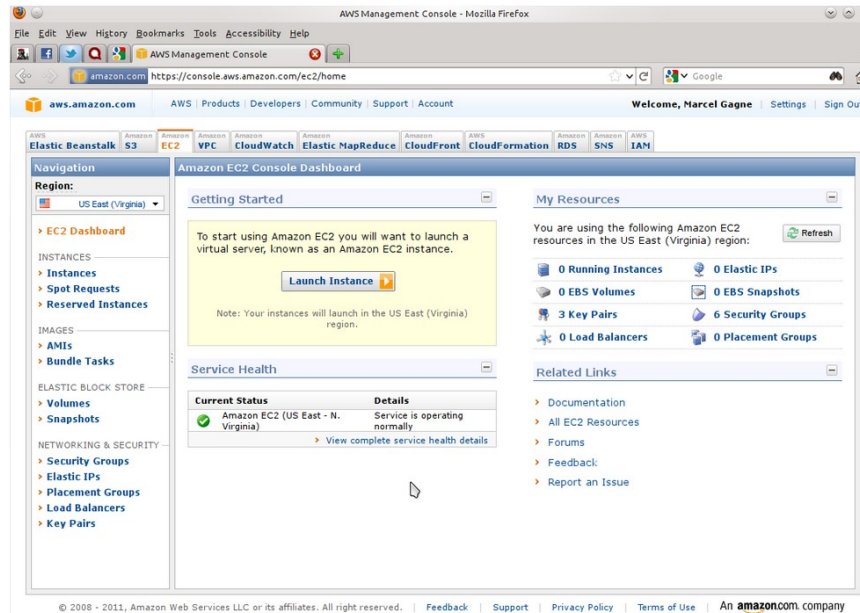
- **Infraestructura como Servicio:** IaaS ofrece máquinas virtuales, almacenamiento virtual, infraestructura virtual, y otros activos de hardware como recursos que los clientes pueden contratar. El proveedor de servicios de IaaS gestiona toda la infraestructura, mientras que el cliente es responsable de todos los demás aspectos de la implementación. Esto puede incluir el sistema operativo, las aplicaciones, y las interacciones del usuario con el sistema.
- **Plataforma como servicio:** PaaS ofrece máquinas virtuales, sistemas operativos, aplicaciones, los servicios, los marcos de desarrollo, operaciones y estructuras de control. El cliente puede desplegar sus aplicaciones en las aplicaciones de infraestructura de nube que se programaron utilizando lenguajes y herramientas que son compatibles los servicios contratados PaaS. El proveedor de servicios gestiona la infraestructura de la nube, los sistemas operativos, y el software de habilitación. El cliente es responsable de la instalación y gestión de la aplicación que está desplegando.
- **Software como Servicio:** SaaS es un entorno completo con aplicaciones, gestión, y la interfaz de usuario. En el modelo SaaS, la aplicación se proporciona al cliente a través de una interfaz de cliente ligero (un navegador, por lo general), y la responsabilidad del cliente comienza y termina con la entrada y la gestión de sus datos y la interacción con el usuario. Todo, desde la aplicación hasta la infraestructura es responsabilidad del proveedor.

Los tres modelos de servicio diferentes en su conjunto han llegado a ser conocido como el **modelo SPI** (Software, Plataforma e Infraestructura) de cloud computing. Podemos encontrar muchos otros modelos de servicio: StaaS, Almacenamiento como servicio; IdaaS, Identidad como Servicio; CmaaS, Cumplimiento como servicio; etc. Sin embargo, los servicios de SPI abarcan todas las otras posibilidades.

Ejemplos de proveedores de servicio **IaaS** son:

- Amazon Elastic Compute Cloud (EC2).
- Eucalyptus.
- GoGrid.

Todos estos proveedores ofrecen acceso directo a los recursos de hardware. En Amazon EC2, considerado el clásico ejemplo de IaaS, un cliente haría disposición de un ordenador en forma de máquina virtual con almacenamiento y, a continuación, pasaría a instalar el sistema operativo y las aplicaciones en ese sistema virtual.

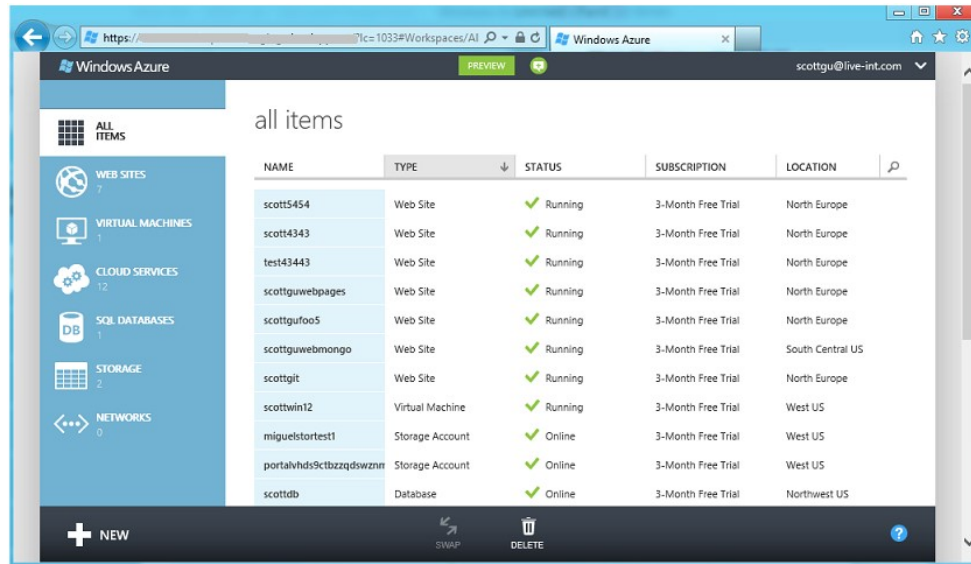


Ejemplos de **PaaS** serían:

- Force.com
- GoGrid Cloud Center
- Google AppEngine
- Microsoft Azure Plataforma.

Google App Engine es un servicio de alojamiento web que presta Google de forma gratuita hasta determinadas cuotas. Este servicio permite ejecutar aplicaciones sobre la infraestructura de Google. Si no se cuenta con un dominio propio, Google proporciona uno con la siguiente estructura, *midominio.appspot.com*. También permite implementar un dominio propio a través de Google Apps. Por el momento las cuentas gratuitas tienen un límite de 500 megabyte de almacenamiento permanente y la suficiente cantidad de ancho de banda y CPU para cinco millones de visitas mensuales, y si la aplicación supera estas cuotas, se pueden comprar cuotas adicionales. Actualmente las aplicaciones Google App Engine se implementan mediante los lenguajes de programación Python, Java, Go y PHP.

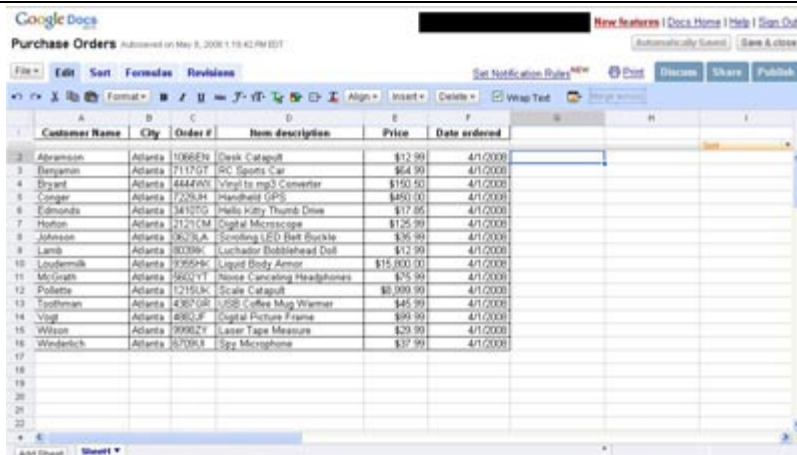
Microsoft Azure es una PaaS ofrecida por Microsoft. Ofrece todo tipo de servicios, desde máquinas virtuales en Windows o Linux, Sistemas gestores de bases de datos, entornos de ejecución .NET, entornos de creación de páginas web y aplicaciones móviles, análisis de datos, ... todo en plataformas creadas en la nube que el cliente sólo paga por utilizarlas, olvidándose de la gestión y el mantenimiento.



Por último, algunos ejemplos de **SaaS** son:

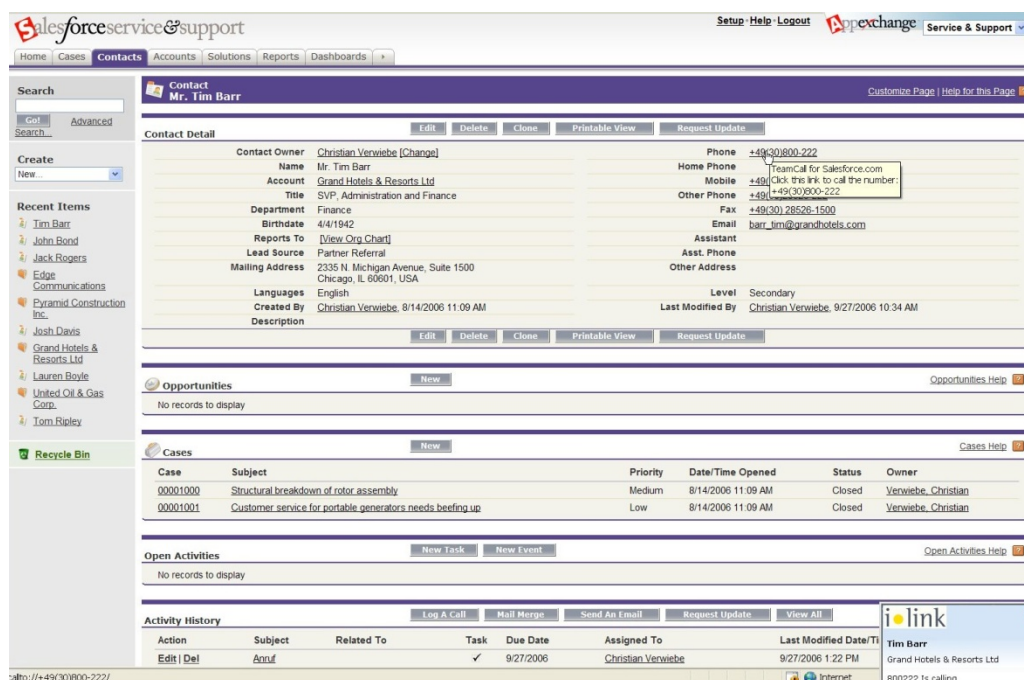
- GoogleApps
- Oracle On Demand
- SalesForce.com
- SQL Azure

Las GoogleApps son un claro ejemplo de SaaS. Contratamos un software (en el caso de Google es normalmente de manera gratuita) que corre en una nube pública. Podemos acceder a un cliente de correo electrónico (gmail), almacenamiento (drive), editores de texto y hojas de cálculo (google docs), y muchas más.



	A	B	C	D	E	F
	Customer Name	City	Order #	Item description	Price	Date ordered
3	Abramson	Atlanta	106676	Desk Catapult	\$12.99	4/1/2008
4	Bergman	Atlanta	17117-07	IRC Sports Car	\$64.99	4/1/2008
5	Brand	Atlanta	4444700	Vinyl to mp3 Converter	\$150.00	4/1/2008
6	Conger	Atlanta	77226-84	Handheld GPS	\$450.00	4/1/2008
7	Edmonds	Atlanta	34107G	Hello Kitty Thumb Drive	\$17.95	4/1/2008
8	Hoffon	Atlanta	2121CM	Digital Microscope	\$125.99	4/1/2008
9	Johnson	Atlanta	2923A	Scrolling LED Back Buckle	\$35.99	4/1/2008
10	Lewis	Atlanta	80396	Lunchbox Bobblehead Doll	\$12.99	4/1/2008
11	Louisville	Atlanta	35954K	Liquid Body Armor	\$15,000.00	4/1/2008
12	McCrath	Atlanta	760211	Noise Cancelling Headphones	\$75.99	4/1/2008
13	Pollette	Atlanta	12150K	Scale Catapult	\$8,000.99	4/1/2008
14	Troffman	Atlanta	4367G8	1.588 Coffee Mug Warmer	\$45.99	4/1/2008
15	Vogt	Atlanta	4862JF	Digital Picture Frame	\$95.99	4/1/2008
16	Wilson	Atlanta	999827	Laser Tape Measure	\$29.99	4/1/2008
17	Wunderlich	Atlanta	25700B	Spa Microphone	\$37.99	4/1/2008

SalesForce es una aplicación del tipo CRM (aplicación empresarial para la gestión de las relaciones con los clientes) que se ofrece como SaaS. El usuario sólo paga por las licencias que utiliza. Se ejecuta la aplicación desde un navegador y el usuario se despreocupa de instalaciones, mantenimientos, copias de seguridad, etc.



Contact Detail

Contact Owner	Christian Verwiebe [Change]	Phone	+49(30)800-222
Name	Mr. Tim Barr	Home Phone	TeamCall for Salesforce.com
Account	Grand Hotels & Resorts Ltd	Mobile	+49 Click the link to call the number: +49(30)800-222
Title	SVP, Administration and Finance	Other Phone	+49(30)28526-1500
Department	Finance	Fax	barr_tim@grandhotels.com
Birthdate	4/4/1942	Email	
Reports To	[View Org Chart]	Assistant	
Lead Source	Partner Referral	Asst. Phone	
Mailing Address	2335 N. Michigan Avenue, Suite 1500 Chicago, IL 60601, USA	Other Address	
Languages	English	Level	Secondary
Created By	Christian Verwiebe, 8/14/2006 11:09 AM	Last Modified By	Christian Verwiebe, 9/27/2006 10:34 AM
Description			

Opportunities

No records to display

Cases

Case	Subject	Priority	Date/Time Opened	Status	Owner
00001000	Structural breakdown of rotor assembly	Medium	8/14/2006 11:09 AM	Closed	Verwiebe, Christian
00001001	Customer service for portable generators needs beefing up	Low	8/14/2006 11:09 AM	Closed	Verwiebe, Christian

Open Activities

No records to display

Activity History

Action	Subject	Related To	Task	Due Date	Assigned To	Last Modified Date/Time
Edit Del	Annul	Annul	✓	9/27/2006	Christian Verwiebe	9/27/2006 1:22 PM

11. Ejemplos de soluciones de Cloud Computing

A nivel informativo vamos a ver 2 soluciones de cloud computing de gran importancia en el mercado.

La primera es una plataforma IaaS en una nube pública: Amazon Web Services (AWS).

La segunda no es una plataforma, sino un software de código abierto para poder crear nuestra propia nube privada (on premise) en nuestras infraestructuras y que también es utilizada en la nube pública. Se trata del software **Openstack**.

Amazon Web Services

Amazon Web Services (AWS) es una colección de servicios que en conjunto forman una plataforma de cloud computing, ofrecidas a través de Internet por Amazon.com. Es una de las ofertas internacionales más importantes de la computación en la nube y compite directamente contra servicios como Microsoft Azure y Google Cloud Platform. Es considerado como un pionero en este campo. Es usado en aplicaciones populares como Dropbox, Foursquare, HootSuite.

Ofrece soluciones tanto de computación como de networking, almacenamiento, desarrollo, bases de datos, etc ... Algunos de los productos ofrecidos en los modelos estudiados:

- **Amazon Elastic Compute Cloud (EC2)** es un servicio IaaS que ofrece servidores virtuales. Equivaldría a los servicios incluidos en Microsoft Azure o Google Compute Engine; y los equivalentes on-premises (nube privada) como OpenStack o Eucalyptus.
- **Amazon Elastic Beanstalk:** un servicio PaaS para hospedar aplicaciones. Equivaldría a los servicios de Google App Engine.
- **Amazon Simple Storage Service (S3)** ofrece objetos de almacenamiento escalables accesibles desde una servicio de interfaz de Web. Modelo SaaS.

<https://aws.amazon.com/es/getting-started/>

OpenStack

OpenStack no es una solución de Cloud Computing, sino un software para poder proporcionar Cloud Computing. Es una plataforma de software libre y de código abierto para el cloud computing, en su mayoría implementada como infraestructura como servicio (IaaS), mediante la cual los servidores virtuales y otros recursos se ponen a disposición de los clientes.

Se utiliza sobre todo en nubes privadas, para poder ofrecer en nuestra infraestructura TI soluciones de Infraestructura como Servicio (IaaS), pero también es utilizado por empresas que ofrecen nubes públicas.

Su misión es proveer una solución flexible tanto para nubes públicas como privadas, sean estas de cualquier tamaño, y para esto se consideran dos requerimientos básicos: las nubes deben ser **simples de implementar** y **masivamente escalables**.

Material elaborado a partir de:

- Introducció al programari de base i a la virtualització. Implantació de Sistemes Operatius. Josep Pons Carrió, Institut Obert Catalunya.
- Apuntes del curso del CEFIRE de Administración Centralizada de Redes con Windows 2012 Server del profesor José Ramón Ruiz Rodríguez bajo licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional:



- Cloud Computing Bible. Barrie Sosinsky. Wiley Publishing Inc. 2011.
- Recursos de Internet.