

Capítulo 1

Cloud computing Introducción

Javier Esparza Peidro - jesparza@dsic.upv.es

Contenido

1.1. Introducción	1
1.2. Cloud Computing	3
1.3. Beneficios	5
1.4. Riesgos	8
1.5. Tecnologías base	9
1.6. Modelos de servicio	11
1.7. Modelos de despliegue	14
1.8. Arquitectura de referencia	15

1.1. Introducción

Cloud computing representa un modelo de computación en el que el usuario consume los recursos de computación (cpu, memoria, almacenamiento, software, ...) del mismo modo que otros suministros públicos como el agua, la electricidad o la telefonía. Los recursos computacionales no son propiedad del usuario, sino que residen en grandes centros de datos remotos y son alquilados. Tras solicitarlos, el usuario recibe los recursos de manera instantánea, y sólo paga por lo que consume (pay-as-you-go). Se trata de una implementación específica del concepto de *utility computing*.

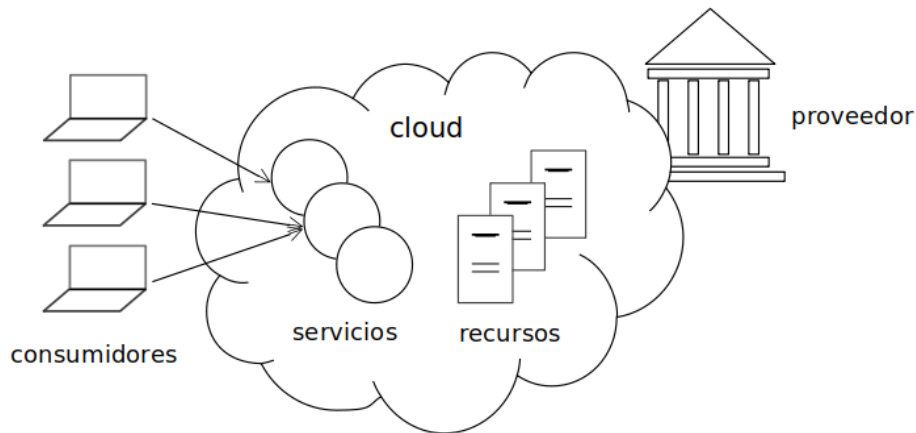


Figura 1.1: Terminología del paradigma cloud computing

En el contexto de cloud computing, un *cloud* (o plataforma de cloud computing) [Erl et al., 2013] representa un entorno tecnológico diseñado específicamente para proporcionar recursos computacionales de manera remota. Para ello, en su interior cuenta con numerosos *recursos* físicos y virtuales. El entorno es propiedad de un *proveedor* y es utilizado por múltiples *consumidores*. Los consumidores acceden a los recursos computacionales a través de *servicios*, y los consumen como una utilidad más. Los servicios ofrecen recursos computacionales a distintos niveles de abstracción y están sujetos a determinadas garantías de calidad de servicio (parámetros *QoS* - *Quality of Service*, como por ejemplo fiabilidad, rendimiento, seguridad, etc.), que se registran en un contrato entre el proveedor y el consumidor, denominado *SLA* (*Service-Level Agreement*). La figura 1.1 ilustra estos conceptos.

Este esquema contrasta con la aproximación más convencional de adquirir la infraestructura física y controlarla en el interior de una organización. En este nuevo contexto, esta última configuración queda rebautizada como infraestructura *on-premise*, o bajo las premisas de una determinada organización.

Paradójicamente, los conceptos que incorpora cloud computing no son nuevos, sino que tienen sus orígenes en los años 60.

“If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility. ... The computer utility could become the basis of a new and important industry.”

—John McCarthy. 1961

“As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread

of ‘computer utilities’

–Leonard Leinrock. 1969.

Sin embargo, los desarrollos tecnológicos necesarios para llevar a la práctica este concepto no han estado lo suficientemente maduros hasta recientemente. En 2006 Amazon lanzó su plataforma Elastic Compute Cloud (EC2), que permite alquilar capacidad de procesamiento. A Amazon siguieron otras grandes compañías como Google y Microsoft con sus productos Google App Engine y Microsoft Azure respectivamente. En la actualidad existe una gran variedad de ofertas comerciales.

1.2. Cloud Computing

En la bibliografía es posible encontrar diversas definiciones de cloud y cloud computing. A continuación se listan algunas de las más populares:

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”

–NIST¹ [[Mell et al., 2011](#)]

“Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualised computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers”

–Buyya [[Buyya et al., 2008](#)]

“Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service Level Agreements”

–Vaquero [[Vaquero et al., 2008](#)]

¹U.S. National Institute of Standards and Technology

En cualquier caso, un entorno de cloud computing generalmente reúne las siguientes características [[Mell et al., 2011](#)]:

- *Uso bajo demanda*

Los consumidores pueden solicitar, configurar, pagar y usar los servicios ofrecidos por el entorno sin intervención humana, conforme son necesarios, bajo demanda. Además, los servicios que se ofrecen suelen presentar un alto grado de configurabilidad, y pueden ser ajustados con facilidad a los requisitos específicos de los consumidores.

- *Acceso universal*

Los servicios cloud pueden ser accedidos desde cualquier lugar y en cualquier momento, a través de mecanismos estándar que favorecen el uso de distintos tipos de clientes (dispositivos móviles, tablets, ordenadores, etc.).

- *Servicio medido*

El consumidor solicita lo que necesita y únicamente paga por lo consumido (pay-as-you-go), generalmente medido en períodos cortos de tiempo (e.g. por hora). Para ello, el entorno debe habilitar mecanismos de medición y contabilización continua del consumo de recursos. Es habitual que el proveedor suministre distintos niveles de servicio, que ofrezcan diferentes garantías, a distintos precios.

- *Compartición de recursos*

Para optimizar el uso de recursos, un entorno de cloud computing habilita mecanismos de compartición de recursos (multi-tenancy), generalmente utilizando técnicas de virtualización. De este modo, los mismos recursos físicos son utilizados por múltiples consumidores concurrentemente. El consumidor generalmente desconoce la ubicación de los recursos.

- *Elasticidad*

Un entorno de cloud computing debe presentar elasticidad. La elasticidad determina la capacidad de un sistema para adaptarse dinámicamente a los cambios en la carga de trabajo adquiriendo o liberando recursos de manera autónoma. La incorporación/eliminación de recursos debe efectuarse de manera instantánea. Esta condición únicamente es viable si se efectúa por medio de procedimientos automáticos, que generalmente aplican técnicas de escalado horizontal. Además, el entorno debe ofrecer la ilusión de que no existe límite en cuanto a la cantidad de recursos que se pueden adquirir.

- *Resiliencia*

Un entorno de cloud computing utiliza diversos mecanismos para reducir al máximo el tiempo de recuperación ante fallos. Para ello es habitual disponer de diversas zonas de disponibilidad en las que se pueden distribuir los recursos adquiridos por los consumidores, incrementando de este modo la fiabilidad de la solución.

1.3. Beneficios

Utilizar un entorno de cloud computing supone importantes ventajas desde diversas perspectivas. A continuación se listan las más relevantes [Buyya et al., 2013, Erl et al., 2013]:

- *Planificación de la capacidad*

La planificación de la capacidad de un sistema es el proceso de determinar cuáles son las necesidades de recursos de dicho sistema a lo largo del tiempo con el objetivo de cubrir las necesidades de sus potenciales consumidores. La capacidad de un sistema representa la cantidad máxima de trabajo que es capaz de servir en un período de tiempo. Si la capacidad del sistema es superior a la demanda (sobre-estimación) entonces el sistema es ineficiente. Si la capacidad del sistema es inferior a la demanda (infra-estimación) entonces el sistema se satura y no es capaz de cubrir las necesidades de los consumidores. El objetivo de la planificación de la capacidad es minimizar la distancia que existe entre la capacidad y la demanda que se ejerce sobre un sistema.

Se trata de una tarea muy compleja, ya que la carga fluctúa en ocasiones de manera impredecible. Un ejemplo de esta complejidad lo representa el efecto slashdot², que sucede cuando en un sitio web con tráfico masivo se incluye un link a un sitio web pequeño, lo cual suele provocar la inmediata sobrecarga y caída del sitio web pequeño.

En un entorno de cloud computing no es necesario planificar la capacidad del sistema. En su lugar, la plataforma monitoriza de manera constante la demanda actual e inyecta o elimina recursos de manera dinámica para adecuar la capacidad del sistema.

- *Reducción de costes*

La adquisición de infraestructura tecnológica y su mantenimiento son costes muy importantes en cualquier compañía. Respecto a la adquisición,

²https://en.wikipedia.org/wiki/Slashdot_effect

requiere una inversión muy fuerte inicialmente, y atiende a una planificación de necesidades que como ya hemos visto no es nada fácil de efectuar. Esta inversión inicial suele suponer una barrera muy importante para pequeñas compañías que desean testear su idea de negocio pero que no tienen claro el retorno de su inversión a corto plazo.

Respecto al coste de mantenimiento, en general suele exceder con creces el coste de adquisición por los siguientes motivos:

- Es necesario disponer de personal técnico especializado en el mantenimiento del entorno operacional.
- Es necesario aplicar actualizaciones y correcciones en los sistemas de manera continuada.
- La inversión en electricidad y refrigeración no es nada despreciable.
- Es necesario implantar medidas y mecanismos de seguridad para proteger la infraestructura.

Externalizar la infraestructura tecnológica a una compañía especializada puede aliviar en gran medida estas problemáticas. La compañía no necesita invertir en infraestructura y personal propio, y no requiere efectuar potentes inversiones iniciales. Además, los costes en infraestructura son proporcionales a la demanda. Si no hay demanda, el coste es mínimo.

Por su parte, la compañía externa consigue mejores costes operacionales al compartir su infraestructura entre múltiples consumidores, trasladando al consumidor este ahorro de costes, y ofreciéndole precios más competitivos.

■ *Agilidad organizacional*

Con soluciones on-premise, es habitual que el departamento de IT se convierta en una carga importante para la compañía, y que constituya una de las fugas de presupuesto más significativas. Además, cualquier cambio en dicho departamento suele requerir de un gran esfuerzo económico y organizativo que impide movimientos rápidos y ágiles.

En un mercado cada vez más dinámico, en el que las organizaciones deben reaccionar rápidamente ante cualquier cambio para resultar competitivas, la infraestructura tecnológica no puede suponer un lastre que limite su velocidad de reacción y adaptación a las condiciones cambiantes del mercado. En este sentido, resulta de interés la siguiente cita, que evoca la imagen de la figura 1.2:

“In the new world, it is not the big fish which eats the small fish, it's the fast fish which eats the slow fish.”

–Klaus Schwab



Figura 1.2: El pez rápido se come al pez lento

Si se utiliza una solución basada en cloud computing, cualquier evolución de la plataforma tecnológica puede aplicarse de manera inmediata, sin inversiones iniciales y con un riesgo mínimo.

■ *Incremento en la escalabilidad*

La escalabilidad es una propiedad que determina la capacidad que tiene un sistema de atender cargas crecientes/decrecientes de trabajo sin degradarse de manera significativa. Para atender dichas cargas crecientes/decrecientes de trabajo es necesario incrementar/decrementar (escalar) la infraestructura subyacente. Tradicionalmente, existen dos procedimientos alternativos:

- Escalado horizontal, también denominado scale-out (scale-in), consiste en replicar recursos del mismo tipo, generalmente de poco coste. Los recursos están disponibles tan pronto como son incorporados. Se trata de una operación a priori únicamente limitada por el número de recursos adicionales disponibles (o de presupuesto).
- Escalado vertical, también denominado scale-up (scale-down), consiste en incrementar las prestaciones de los recursos disponibles. Generalmente se requiere hardware especializado de mayor coste. Esta operación pasa por reemplazar o actualizar recursos, y posteriormente configurarlos, por lo tanto implica un período de tiempo en el que los recursos no estarán disponibles. Además, existen límites físicos de la capacidad máxima que estos recursos pueden alcanzar.

El procedimiento de escalado que habitualmente se utiliza en cloud computing es el de escalado horizontal. Un entorno cloud puede reservar recursos de manera instantánea. Esta capacidad permite al sistema atender cargas inesperadas de manera automática, o bien liberar recursos cuando ya no son necesarios, incrementando de manera dramática la escalabilidad de la solución.

■ *Fiabilidad*

Un entorno cloud minimiza el tiempo de recuperación ante fallos utilizando diversas técnicas como replicación de componentes o distribución geográfica de componentes y reparto de la carga. Además, el proveedor se compromete a garantizar unos niveles mínimos de servicio por medio de contratos SLA. Todo ello contribuye a que los productos ofrecidos por los consumidores a sus clientes estén disponibles durante períodos más largos de tiempo, con el beneficio económico que ello conlleva.

1.4. Riesgos

Sin embargo, este nuevo modelo de computación no está exento de ciertos riesgos, que es necesario entender [Buyya et al., 2013, Erl et al., 2013]:

- *Seguridad*

Al trasladar los datos a un entorno cloud, las medidas de seguridad que se implantan están compartidas por el proveedor y el consumidor. Esta condición puede dificultar establecer una arquitectura de seguridad adecuada, en especial si los mecanismos de seguridad no son compatibles. Además, al no poseer la infraestructura, no se posee control sobre los mecanismos que se implantan para proteger los datos, lo cual puede suponer problemas de confidencialidad, si los datos no son cifrados adecuadamente. Por último, al almacenar los datos de todos los consumidores en una infraestructura común, una vulnerabilidad o ataque experimentado por un consumidor puede comprometer los datos de todos los demás.

- *Menor control*

El consumidor posee un menor grado de control sobre los recursos utilizados. Esta condición puede generar incertidumbre o comportamientos inesperados ante determinadas eventualidades. Un ejemplo sería cuando el proveedor no cumple los niveles de calidad establecidos en el SLA, una situación que puede dejar expuesto al consumidor de dichos recursos. Otro ejemplo serían los retardos que se pueden experimentar en las comunicaciones a la hora de acceder a un recurso, ya que al estar distribuido en distintas ubicaciones puede variar en función del número de saltos que dista entre el consumidor y el recurso o bien en función de las condiciones de tráfico existentes.

- *Falta de portabilidad entre proveedores*

Existe una evidente carencia de estándares en el área de cloud computing. Esta situación motiva que cada proveedor implemente sus propias

soluciones, y que las soluciones ofrecidas por un proveedor no sean interoperables con las soluciones ofrecidas por otro. Al contratar la solución de un proveedor el consumidor adquiere dependencias fuertes sobre ésta y sobre las políticas y estrategias adoptadas por dicho proveedor. Ante cualquier eventualidad, puede resultar muy complejo cambiar de proveedor. Este fenómeno es conocido como *vendor/platform lock-in*.

- *Problemas legales*

Los centros de datos que impulsan un entorno cloud pueden estar distribuidos en distintos países. Consecuentemente, sus consumidores podrían desconocer el lugar exacto en el que se almacenan sus datos, o las medidas de seguridad que se aplican sobre estos. Esta situación puede generar problemas en países donde se regulen aspectos sobre la privacidad y seguridad de los datos. Por ejemplo, si los datos son sensibles, en España se exigen una serie de medidas de protección físicas y lógicas extraordinarias que podrían no ser implementadas por un proveedor dado. Otro ejemplo sería el Reino Unido, donde la ley exige que los datos de un ciudadano del país deben residir en el mismo país.

1.5. Tecnologías base

El modelo de cloud computing es el resultado de la convergencia de diversas tecnologías que han alcanzado un grado de madurez suficiente. A continuación se efectúa una revisión de las más relevantes [Buyya et al., 2013, Erl et al., 2013].

- *Sistemas distribuidos*

Un entorno cloud es en esencia un sistema distribuido tal y como se define en [Tanenbaum and van Steen, 2007]:

“Un sistema distribuido es una colección de computadoras independientes que aparece antes los usuarios del sistema como una única computadora”

En el campo de los sistemas distribuidos tres importantes hitos han precedido, condicionado e inspirado el modelo de cloud computing:

- Los *mainframes*, computadoras muy potentes utilizadas por grandes compañías en tareas con grandes requerimientos. Estas máquinas generalmente incluían múltiples procesadores y permanecían en continua

ejecución, incluso durante las operaciones de mantenimiento (reparación, reemplazo de componentes, etc.). Brillaban especialmente en la ejecución de tareas en batch.

- Los *clústers*. Nacieron como una alternativa de bajo coste a los mainframes. Un clúster es una colección de recursos independientes (pero con características similares), de bajo coste, que se interconectan y trabajan como un único sistema. En un clúster los recursos son replicados y el sistema implementa características de detección y recuperación ante fallos, con el objetivo de incrementar la fiabilidad del conjunto. Además, resulta sencillo incrementar la capacidad del sistema incorporando nuevos recursos.
- Los *grids*. Al igual que en un clúster, un grid consiste en uno o varios pools de recursos. Sin embargo, en este caso los recursos son heterogéneos y están dispersos geográficamente, y ceden potencia computacional para trabajar en el grid. Se considera el precursor directo de cloud computing.

Además, para ofrecer los niveles de servicio prometidos por el paradigma de cloud computing, los grandes centros de datos deben ser capaces de autogestionarse [Group et al., 2005]. Son por tanto muy relevantes los avances experimentados en computación autónoma (autonomic computing). De acuerdo a IBM Autonomic Computing Initiative [Poslad, 2011], un sistema autónomo debe presentar 4 propiedades fundamentales:

- *Auto-configuración (self-configuration)*: sus componentes deben configurarse automáticamente.
- *Auto-optimización (self-optimization)*: el sistema monitoriza y controla su consumo de recursos con el objetivo de alcanzar un uso óptimo de recursos.
- *Auto-curación (self-healing)*: el sistema detecta y corrige los fallos de manera automática.
- *Auto-protección (self-protection)*: el sistema identifica y se protege de posibles ataques de manera proactiva.

Para ello, un sistema autónomo generalmente integra sensores que monitorizan el uso y salud de sus recursos, bucles de control que efectúan optimizaciones basadas en los datos monitorizados y actuadores (effectors) que llevan a cabo cambios en el sistema.

■ Virtualización

La virtualización es el proceso de crear una versión virtual (lo contrario de real) de algo. La virtualización se puede aplicar sobre diversos tipos de recursos y a distintos niveles de abstracción. La virtualización de hardware ha supuesto un elemento clave para la evolución del paradigma de cloud computing. Este tipo de virtualización convierte los recursos computacionales físicos (procesadores, memoria, disco, red, . . .) en recursos virtuales, permitiendo la compartición de la infraestructura hardware.

- **SOA**

Las arquitecturas basadas en servicios constituyen otra pieza fundamental en el modelo de cloud computing. Por un lado, los recursos computacionales son accedidos por los consumidores a través de servicios. Por otro lado, las aplicaciones cloud nativas se construyen como composición de múltiples servicios que se ejecutan en el mismo o diferentes proveedores. La evolución de las arquitecturas basadas en servicios y en particular de las tecnologías web han favorecido la integración de todo tipo de software que se ejecuta en distintas plataformas.

1.6. Modelos de servicio

Un entorno de cloud computing ofrece recursos computacionales a sus consumidores a través de servicios. Esta condición permite inducir una clasificación de distintos tipos (o modelos) de servicio con la terminología *X as a Service - XaaS*, que representa la idea de que “*X se ofrece como servicio*”. En la actualidad existen tres *modelos de servicio* que han sido universalmente aceptados y determinan el nivel de abstracción de los recursos computacionales ofrecidos por el servicio:

- **Infrastructure as a Service - IaaS**

El consumidor puede adquirir y configurar recursos computacionales de bajo nivel, como procesadores, memoria, almacenamiento o conectividad. Para ello, el entorno generalmente los virtualiza y los ofrece en “paquetes” denominados máquinas virtuales o servidores virtuales. El consumidor es responsable de instalar y configurar el sistema operativo así como el software necesario para ejecutar sus aplicaciones. Amazon Elastic Compute Cloud sería un ejemplo de proveedor IaaS.

- **Platform as a Service - PaaS**

El consumidor desarrolla, instala y ejecuta sus aplicaciones en el entorno cloud, utilizando para ello los procedimientos y las herramientas proporcionadas por el proveedor. El consumidor no tiene control de la infraestructura

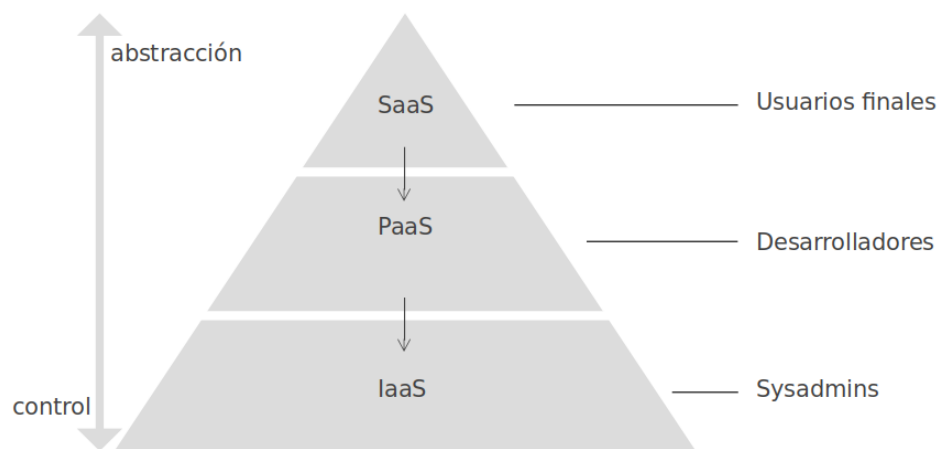


Figura 1.3: Modelos de servicio

subyacente. Se centra en diseñar sus propias soluciones y se limita a utilizar los artefactos de alto nivel que el proveedor expone. Google App Engine o Microsoft Azure serían ejemplos de proveedores PaaS.

- *Software as a Service - SaaS*

El consumidor utiliza las aplicaciones suministradas por el proveedor y que se ejecutan en el entorno cloud. El consumidor no controla la infraestructura subyacente, ni es capaz de ejecutar nuevas aplicaciones. La mayor parte de aplicaciones que se utilizan a través de un navegador son ejemplos de servicios SaaS (e.g. Google Docs, Gmail, Microsoft Office 365, ...).

Estos modelos de servicio se interrelacionan en una arquitectura en capas donde los servicios de más alto nivel hacen uso de los servicios de más bajo nivel, tal y como se ilustra en la figura 1.3.

Efectuando una comparativa con un modelo computacional tradicional, los servicios ofrecidos por una solución IaaS se corresponden con lo que se podría esperar de un sistema operativo, esto es, gestión “inteligente” de recursos de “bajo nivel”. Por su parte, los servicios ofrecidos por una solución PaaS se corresponden con entornos de ejecución y/o facilidades de desarrollo de aplicaciones. Por último, los servicios de alto nivel ofrecidos por soluciones SaaS se corresponden con las funcionalidades suministradas por las aplicaciones. La figura 1.4 ilustra esta comparativa.

Los distintos modelos de servicio están enfocados a distintos perfiles de consumidor. Los servicios IaaS están indicados para administradores de sistemas, que gestionan recursos de bajo nivel virtualizados, sobre los que deben instalar sistemas y aplicaciones. Los servicios PaaS son adecuados para desarrolladores

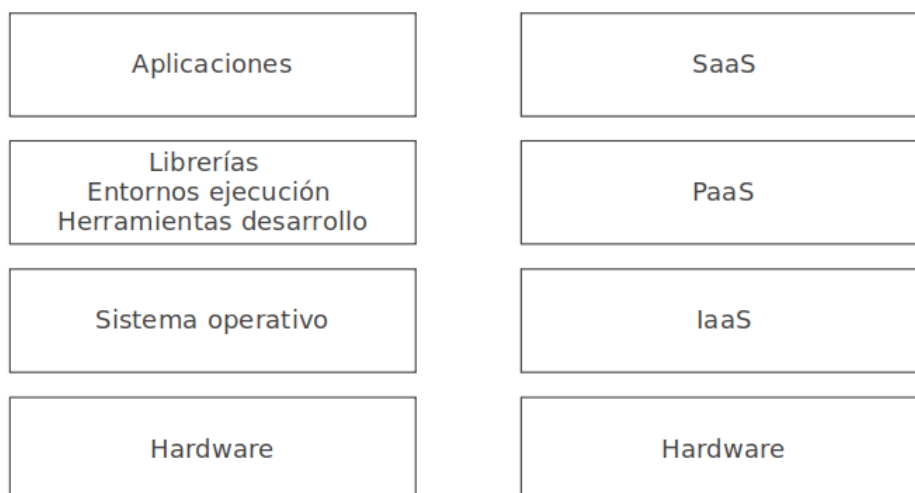


Figura 1.4: Comparativa con sistema tradicional

de aplicaciones, que cuentan con entornos, herramientas y facilidades para construir sus propias aplicaciones. Por último, los servicios SaaS están diseñados para usuarios finales, que acceden a determinadas funcionalidades bajo demanda.

En la figura 1.5 se ilustra la pila de tecnologías que es necesario gestionar para desplegar una solución. Además, se identifica la responsabilidad que el consumidor y proveedor tiene en cada uno de los modelos presentados. Desde esta perspectiva se puede pensar en el cloud como el límite que determina dónde acaban las responsabilidades del consumidor y dónde comienzan las del proveedor.

Siguiendo la corriente XaaS presentada anteriormente es posible identificar otros modelos de servicio más sofisticados, que permiten gestionar recursos de muy diversa índole. Algunos ejemplos son:

- Bare-metal as a Service - BMaaS
- Network as a Service - NaaS
- Desktop as a Service - DEaaS
- Data as a Service - DaaS
- DataBase as a Service - DBaaS
- Storage as a Service - STaaS
- Security as a Service - SECaaS
- Identity as a Service - IDaaS

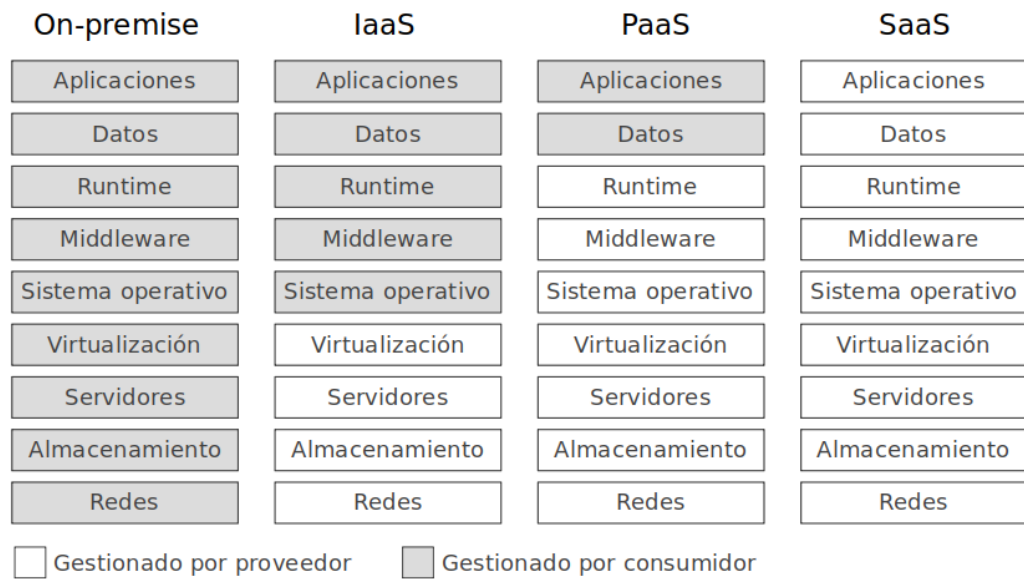


Figura 1.5: Responsabilidades en los modelos de servicio

- Hadoop as a Service - HaaS
- Backup as a Service - BaaS
- Communication as a Service - CaaS
- Load Balancers as a Service - LBaaS
- Message Queue as a Service - MQaaS
- Monitoring as a Service - MaaS

1.7. Modelos de despliegue

Atendiendo a la ubicación y distribución de los distintos componentes que conforman un entorno cloud es posible identificar cuatro modelos fundamentales [Mell et al., 2011]:

- Cloud público (public cloud): la infraestructura es propiedad de una compañía y está disponible para uso público.
- Cloud privado (private cloud): la infraestructura es propiedad de una organización, y es utilizada únicamente por ésta.

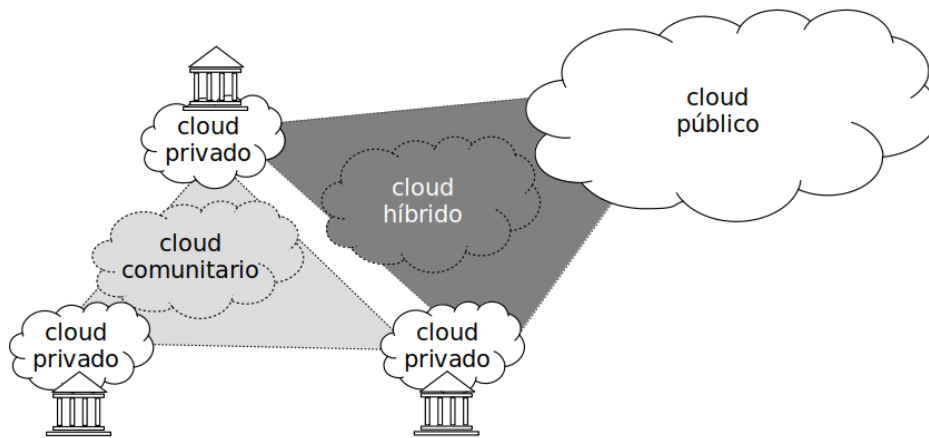


Figura 1.6: Modelos de despliegue

- Cloud comunitario (community cloud): solución diseñada por una o varias organizaciones para servir un propósito común.
- Cloud híbrido (hybrid cloud): es una solución que combina múltiples clouds (público, privado o comunitario), ofreciendo una imagen homogénea del conjunto.

La figura 1.6 ilustra esta clasificación.

1.8. Arquitectura de referencia

Una vez presentados los conceptos fundamentales de cloud computing resulta muy útil definir una arquitectura de referencia. Se trata de un modelo conceptual que ayuda a identificar los elementos fundamentales de un entorno cloud. Dicha arquitectura permite utilizar un lenguaje común, independiente de un entorno cloud específico. Para resultar útil, la arquitectura debe representar un compromiso entre abstracción y detalle. Debe ser lo suficientemente abstracta para no condicionar posibles implementaciones, pero al mismo tiempo debe incluir el detalle necesario para resultar útil a la hora de comparar y discutir distintas tecnologías. En la figura 1.7 se presenta la arquitectura de referencia propuesta por NIST [Liu et al., 2011].

Como se puede observar, la arquitectura de referencia define cinco actores principales:

- El *consumidor*: entidad que utiliza los servicios proporcionados por los proveedores. Los consumidores acuerdan mediante SLAs los requisitos de rendimiento, seguridad, etc. de los servicios consumidos.

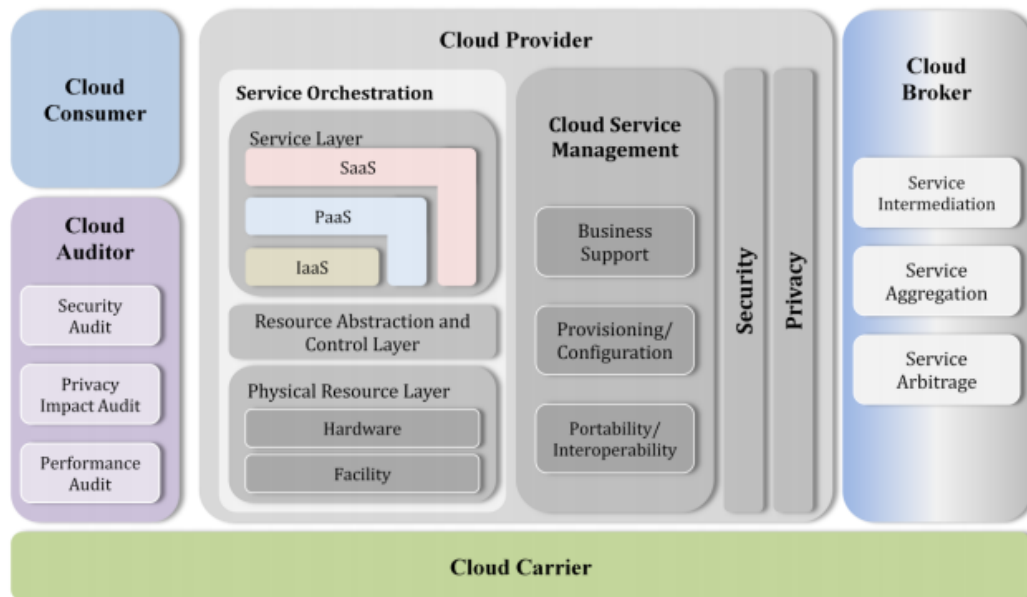


Figura 1.7: Arquitectura de referencia

- El *proveedor*: entidad responsable de proporcionar los servicios cloud a las partes interesadas. Para ello, el proveedor adquiere la infraestructura necesaria y la gestiona.
- El *auditor*: entidad que obtiene información del entorno cloud y efectúa un asesoramiento independiente sobre diversos aspectos como su rendimiento, seguridad, etc.
- El *bróker*: entidad que hace de intermediario entre el consumidor y el proveedor. El bróker ofrece servicios que se clasifican en tres categorías: intermediación (para mejorar ciertas capacidades del proveedor), agregación (permite combinar distintas soluciones) y arbitraje.
- El *facilitador* (carrier): entidad que se encarga de proporcionar la conectividad y transporte de los servicios cloud entre el consumidor y el proveedor. Esta figura la suelen encarnar las compañías de telecomunicaciones.

Respecto al proveedor, el modelo de referencia identifica las responsabilidades ilustradas en la figura 1.8, que se listan a continuación:

- Despliegue de servicios: determina cómo son de exclusivos los recursos computacionales que se hacen accesibles al consumidor. Esta condición está relacionada con los modelos de despliegue estudiados anteriormente.

En un cloud público, los recursos se hacen accesibles al público en general. En un cloud privado, los recursos únicamente pueden ser accedidos por los empleados de una determinada compañía. En un cloud comunitario, los recursos pueden ser accedidos por aquellos consumidores que comparten un mismo objetivo. Finalmente, un cloud híbrido representa la composición de dos o más clouds, y por lo tanto los consumidores podrían acceder a la unión de sus recursos.

- Orquestación de servicios: se refiere a la gestión de los recursos de computación, necesaria para poder proporcionar servicios cloud a los consumidores. En este caso, es habitual utilizar un modelo basado en tres capas, que representa los componentes que el proveedor necesita para proporcionar sus servicios. La capa de menor nivel (*Physical Resource Layer*) representa los recursos físicos adquiridos por el proveedor. Ello implica computadores, redes, sistemas de almacenamiento, etc. También incluye la infraestructura física requerida, así como los suministros necesarios (agua, electricidad, refrigeración, etc.). La capa intermedia (*Resource Abstraction and Control Layer*) pretende incrementar el nivel de abstracción de los recursos gestionados a través de abstracciones software. Para ello, es habitual utilizar técnicas de virtualización hardware (aunque no es la única forma). Además, esta capa permite adquirir un elevado nivel de control sobre dichos recursos, habilitando componentes para la reserva de recursos, control de accesos y monitorización de su uso. Finalmente, la capa superior (*Service Layer*) publica una API que posibilita el acceso a los recursos por parte del consumidor a través de servicios. Dependiendo del nivel de abstracción de los recursos accedidos, el proveedor proporcionará distintos modelos de servicio (IaaS, PaaS, SaaS). Es habitual que los servicios de más alto nivel hagan uso de los servicios de más bajo nivel. De este modo, una aplicación SaaS podría construirse por encima de componentes PaaS y estos podrían construirse por encima de componente IaaS. Sin embargo, estas dependencias no representan una obligación, y es perfectamente posible que una aplicación SaaS se implemente directamente a partir de máquinas virtuales.
- Gestión de servicios cloud: incluye la gestión de todos los servicios ofrecidos a los consumidores. Se pueden describir desde distintas perspectivas: desde la perspectiva de negocio, desde la perspectiva de aprovisionamiento y configuración, y desde la perspectiva de la portabilidad e interoperabilidad.
- Seguridad y privacidad: se trata de un asunto transversal que afecta a todos los componentes del cloud. Debe ser correctamente integrado en todas sus facetas.

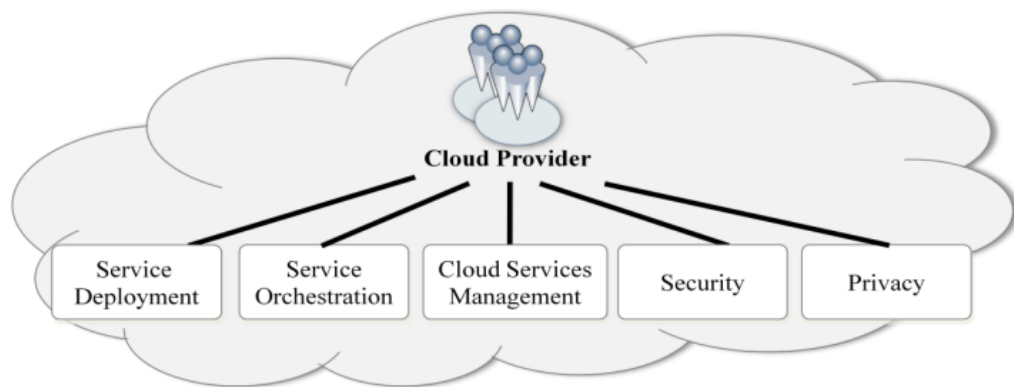


Figura 1.8: Arquitectura de referencia

Referencias

- [Buyya et al., 2013] Buyya, R., Vecchiola, C., and Selvi, S. T. (2013). *Mastering Cloud Computing: Foundations and Applications Programming*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.
- [Buyya et al., 2008] Buyya, R., Yeo, C. S., and Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *2008 10th IEEE international conference on high performance computing and communications*, pages 5–13. IEEE.
- [Erl et al., 2013] Erl, T., Puttini, R., and Mahmood, Z. (2013). *Cloud computing: concepts, technology, & architecture*. Pearson Education.
- [Group et al., 2005] Group, I. et al. (2005). An architectural blueprint for autonomic computing. *IBM White paper*.
- [Liu et al., 2011] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., Leaf, D., et al. (2011). Nist cloud computing reference architecture. *NIST special publication*, 500(2011):1–28.
- [Mell et al., 2011] Mell, P., Grance, T., et al. (2011). The nist definition of cloud computing.
- [Poslad, 2011] Poslad, S. (2011). *Ubiquitous computing: smart devices, environments and interactions*. John Wiley & Sons.
- [Tanenbaum and van Steen, 2007] Tanenbaum, A. S. and van Steen, M. (2007). *Distributed Systems: Principles and Paradigms*. Pearson Prentice Hall, Upper Saddle River, NJ, 2 edition.
- [Vaquero et al., 2008] Vaquero, L. M., Roderio-Merino, L., Caceres, J., and Lindner, M. (2008). A break in the clouds: towards a cloud definition.