

Cloud Computing

Sistemas Distribuidos

2022-2023

Concepto

IDEA CLAVE

- Suministrar **computación** como un **servicio**
- Analogía: Lavadora vs. lavandería
- ¿Vuelta a los orígenes?
 - Recursos de computación en máquinas remotas
 - El usuario utiliza *terminales tontas*
 - La nueva terminal puede ser el *navegador*
- Pero ahora el usuario puede ser también una gran empresa.

Escenario:

Las empresas necesitan recursos computacionales. Tanto si la empresa es de *software* como si no.

Cómo suplir esa necesidad:

- Comprarlos y contratar personal para mantenerlos
- Alquilarlos y subcontratar personal

Problemas:

- Dimensión inicial. Coste inicial. Coste de mantenimiento.
- Crecimiento (planificación en avance). Negociaciones, plazos.

Definición (NIST)

Cloud Computing es un modelo que permite acceso **vía red**, ubicuo, cómodo y bajo demanda, a un pool **compartido** de recursos computacionales **configurables** (por ejemplo, redes, máquinas servidoras, almacenamiento, aplicaciones y servicios) que pueden ser aprovisionados y liberados **rápidamente**, con el mínimo de esfuerzo y de gestiones o de interacción con el proveedor del servicio.

Propiedades básicas

- **Autoservicio bajo demanda.** A través de herramientas.
- **Acceso vía red de banda ancha.** Hace irrelevante la localización.
- **Servicio medido.** Y **pago-por-uso.**
- **Pool de recursos.** Menos margen para la configuración a medida.
- **Elasticidad rápida.** Los recursos se consiguen y liberan en muy poco tiempo.

Para sacar partido de estas propiedades el software debe ser diseñado apropiadamente.

- Diseño de sistemas distribuidos
- Aplicaciones *cloud ready*

Propiedades IDEALES de las aplicaciones

- Arquitectura **distribuída**
- Diseño **elástico** (que pueda escalar mediante la adición de recursos, no mediante aumentar su potencia). Escalabilidad *horizontal*.
- Minimizar el estado que requiere guardar cada recurso. Promover estado **aislado**.
- **Automatización** de la gestión (herramientas deciden cuando proveer o liberar recursos)
- Acoplamiento **débil** entre los recursos, ya que pueden no estar disponibles (conceptos de SOA)

El acrónimo IDEAL hace referencia a estas propiedades: **I**solated state, **D**istribution, **E**lasticity, **A**utomated management, **L**oose coupling

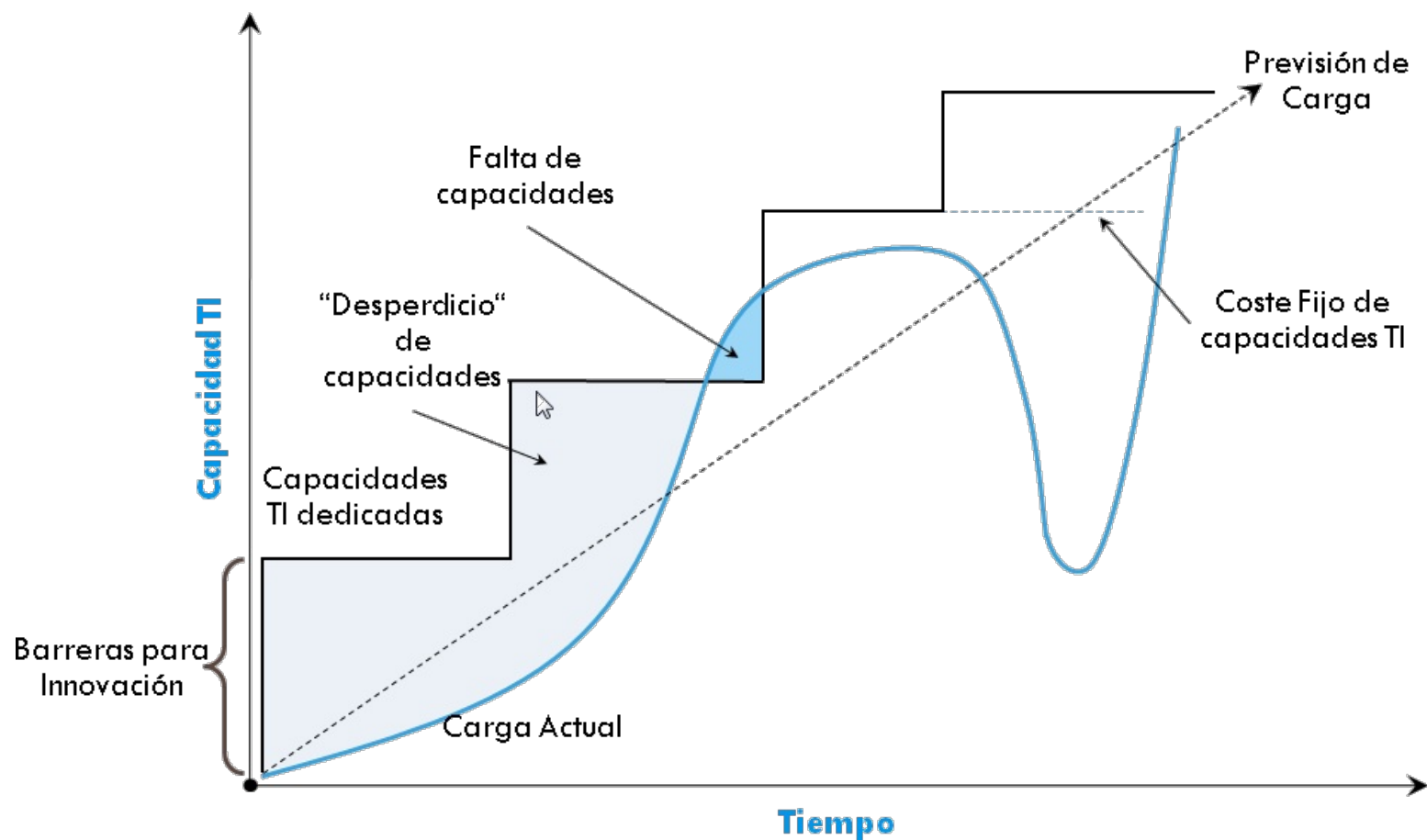
Recursos y carga

Pensemos en el dimensionamiento de recursos para dar servicio a una carga computacional.

Cuando esos recursos son nuestros:

1. Hay que planificar los recursos necesarios y hacer una inversión inicial
2. Vigilar el uso de recursos y si se acerca a su máximo, comprar más
3. Los incrementos rápidos de carga no serán atendidos correctamente
4. Si la carga disminuye otra vez, tendremos recursos sobredimensionados

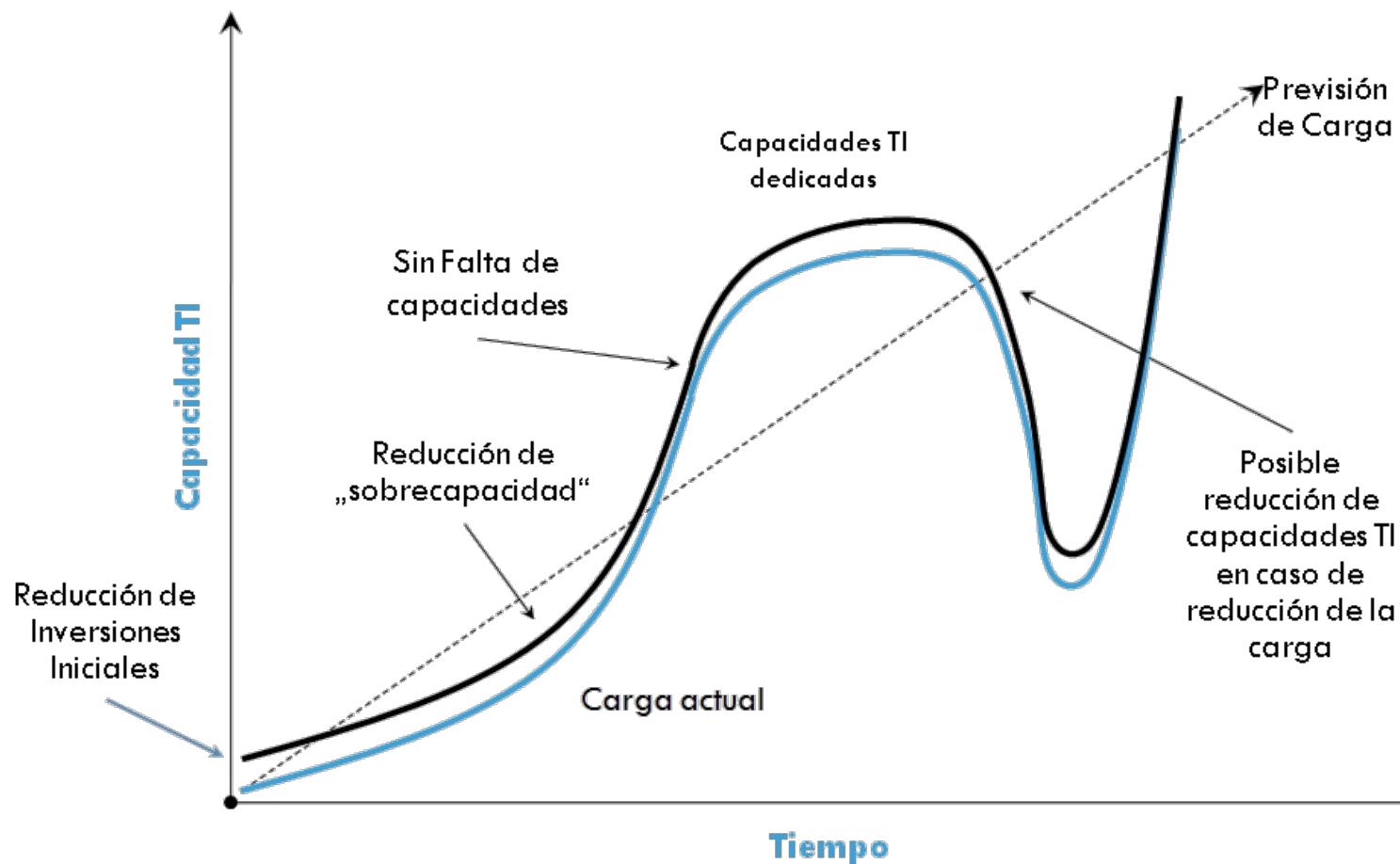
Gráficamente:



Usando la *elasticidad* del cloud

1. No hay costes de entrada
2. La provisión de recursos puede automatizarse
3. La provisión es rápida, la carga se asume
4. Si mengua la carga, se liberan recursos
5. Se optimizan los costes, pues se paga por lo que se usa.

Gráficamente



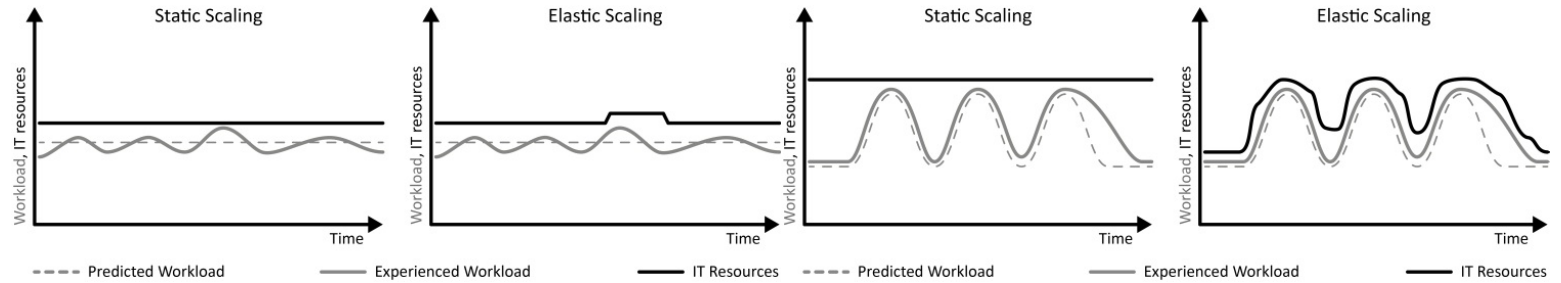
Diferentes tipos de carga

- **Estática.** Es constante o con variaciones mínimas
- **Periódica.** Picos a intervalos regulares (muy común)
- **Una vez en la vida.** Un gran pico que no se repite más (eventos mundiales)
- **Impredecible.** Picos a intervalos irregulares
- **Gradual.** Crece (o decrece) a ritmo constante

En todos ellos la elasticidad que proporciona *cloud* es deseable.

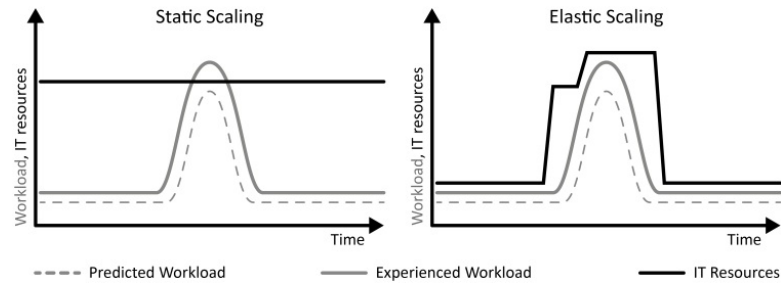
Tipos de carga

Estable



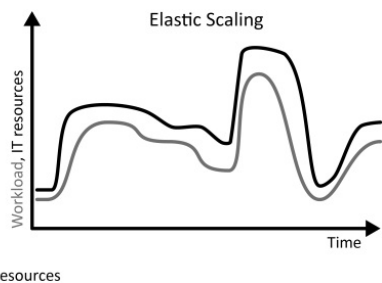
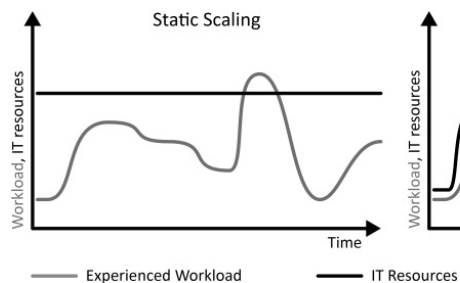
Periódica

"Una vez en la vida"

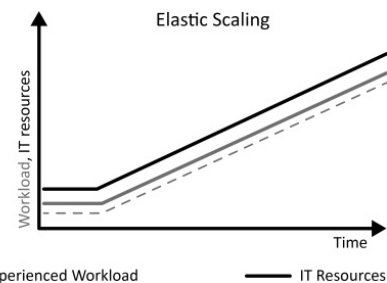
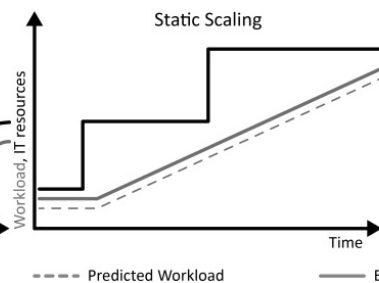


Tipos de carga

Impredecible



Creciente (o decreciente)



Todos los casos se benefician de la elasticidad.

Especialmente el "una vez en la vida" y el "periódica"

Modelos de servicio

- **IaaS**

- Infraestructura (CPU, disco, red)
- Hardware *virtualizado*
- Libertad de instalación de software

- **PaaS**

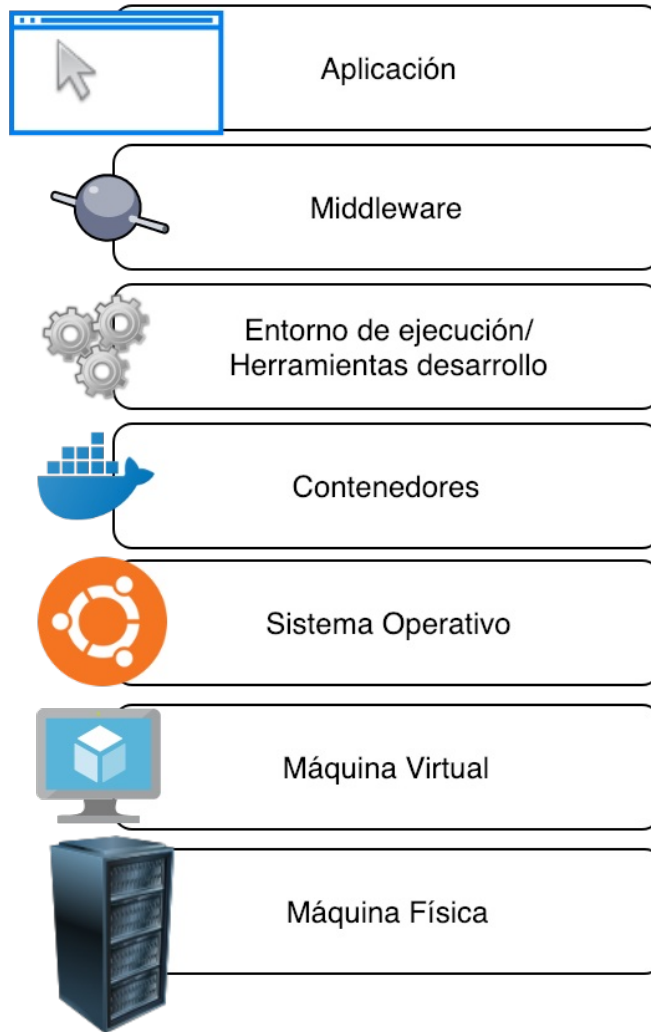
- **Plataforma** (entorno de ejecución)
- Incluye su propio operativo, servidor, *framework*, etc.
- Todo ya instalado y configurado

- **SaaS**

- **Software** (aplicaciones)
- Aplicaciones "ya hechas" que se ejecutan "en la nube"
- Típicamente aplicaciones genéricas, de uso común (ofimática, email,...)

Modelos de servicio

Un servicio se compone de capas



Modelo IaaS

Infrastructure as a Service



Modelo PaaS

Platform as a Service



Modelo SaaS

Software as a Service



Modelos de despliegue

Se refiere a la pregunta ¿dónde se aloja la nube?

- **Nube pública.** En infraestructura de un proveedor externo (Amazon, Google, ...)
- **Nube privada.** En infraestructura propia.
- **Nube comunitaria.** En infraestructura compartida por un conjunto de empresas que colaboran.
- **Híbrida.** Varias nubes, cada una con alguna de las tecnologías anteriores, unidas por alguna tecnología de integración.

Ejemplos

- **IaaS**

- Amazon Elastic Compute Cloud [EC2] (nube pública)
- Google Cloud Engine (nube pública)
- Microsoft Azure Virtual Machines
- Rackspace (nube pública)
- vCloud, OpenStack (nube privada)

- **PaaS**

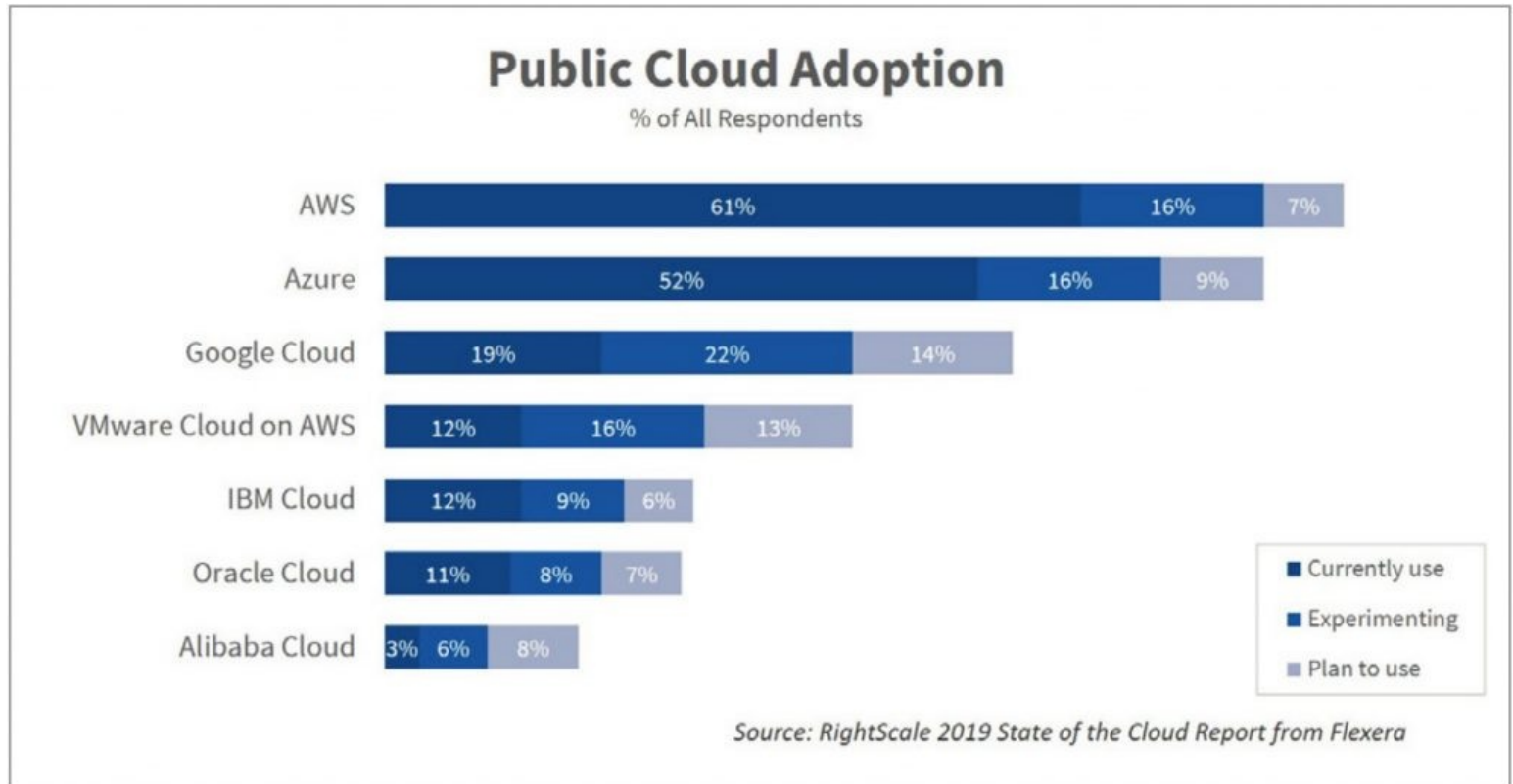
- Google App Engine (pública)
- Amazon Elastic Beanstalk (pública)
- Microsoft Windows Azure (pública)
- Plataforma Force [Salesforce] (pública)
- T-System's Dynamic Services for SAP (privada)

Ejemplos

- **SaaS**
 - Salesforce (CRM)
 - Microsoft Office365 (ofimática)
 - IBM SmartCloud (ofimática)
 - Google Apps (ofimática)
 - Dropbox, Drive, iCloud, SkyDrive
 - etc.

Catálogo de servicios

Los tres grandes: Amazon, Microsoft y Google



Servicios de computación

- **IaaS:** Máquinas virtuales. Precio según características y tiempo de uso.
- **PaaS:** Plataforma de desarrollo y de ejecución. Precio según utilización de cada recurso.
- **FaaS:** Funciones (*serverless*). Pago por número de ejecuciones, tiempo de ejecución y tamaño. Muy barato para pocas llamadas.
- **CaaS:** Contenedores. Proporcionas la imagen y el proveedor se ocupa de ejecutarla en algún sitio. Pago por el IaaS necesario para ello.
- **Batch:** ejecución *por lotes* (desatendida, largos tiempos de procesamiento). Pago por recursos utilizados.

Servicios de almacenamiento

- **Disco local** (formateable)

Está en la misma máquina física que la máquina virtual. Es efímero ante *reboots*

- **Almacén de bloques** (formateable)

Análogo a un "disco USB virtual". Pueden conectarse varios a una misma máquina, pero no varias máquinas a un disco. Sigue existiendo (y tiene coste) con máquina apagada.

- **Almacén de objetos** (*buckets* para información genérica)

Accesibles por HTTP, independiente de la máquina. Más lentos pero más baratos.

- **Sistemas de Archivado** de acceso poco frecuente

Aún más baratos (y lentos). Para *backups*

- **Sistemas de ficheros compartidos**

Protocolo específico de acceso (NFS), se puede "montar" en varias máquinas a la vez.

Database As A Service

Soluciones típicamente ofertadas (hay multitud):

- **Bases de datos relacionales.**

Son las más comunes (Oracle, Postgres, etc.) o implementaciones propias del proveedor (Aurora)

- **Almacén clave-valor.**

Accesibles vía REST. Frecuentemente usadas para guardar metainformación de los almacenes de objetos.

- **Base de datos documental.**

Alta escalabilidad y disponibilidad.

- **Cachés.**

Datos en RAM para acceso ultra-rápido.

- **Big Data.**

Bases de datos especializadas en enormes volúmenes de datos.

Y más...

- Servicios para crear y gestionar **redes** virtuales (cortafuegos, DNS, balanceadores de carga, CDNs)
- Servicios de **middleware** (mensajería de colas)
- Servicios de monitorización, *logging* y alarmas
- **Automatización** de despliegues *Infrastructure as Code*
- **Orquestación** de múltiples contenedores (Kubernetes)
- PaaS para **analítica** (big-data: Hadoop, Spark)
- PaaS para **IoT**

La oferta de servicios crece día a día.