TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Macroarea di Ingegneria
Dipartimento di Ingegneria Civile e Ingegneria Informatica

3/10/22

Introduction to Cloud Computing

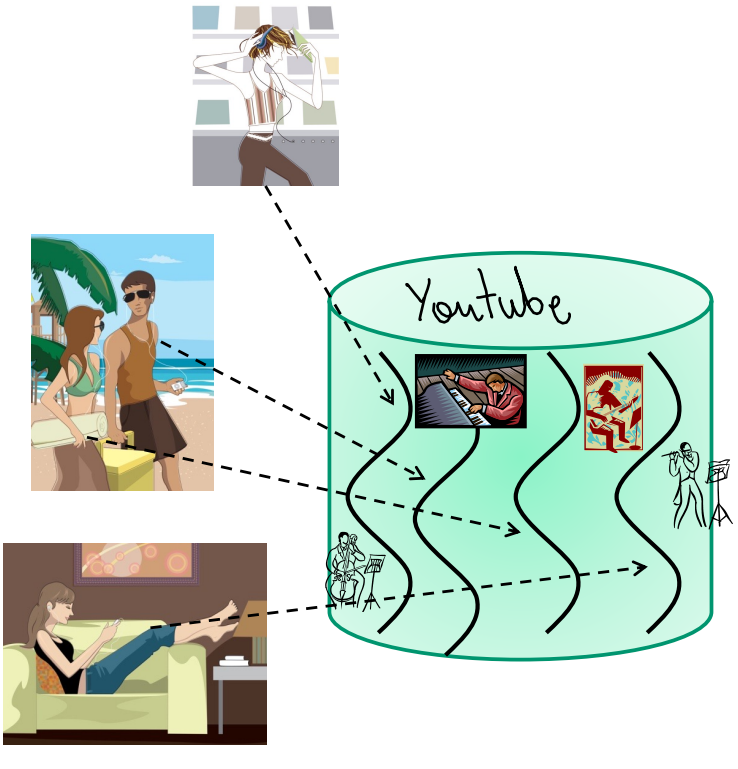
Corso di Sistemi Distribuiti e Cloud Computing
A.A. 2021/22

Valeria Cardellini

Laurea Magistrale in Ingegneria Informatica

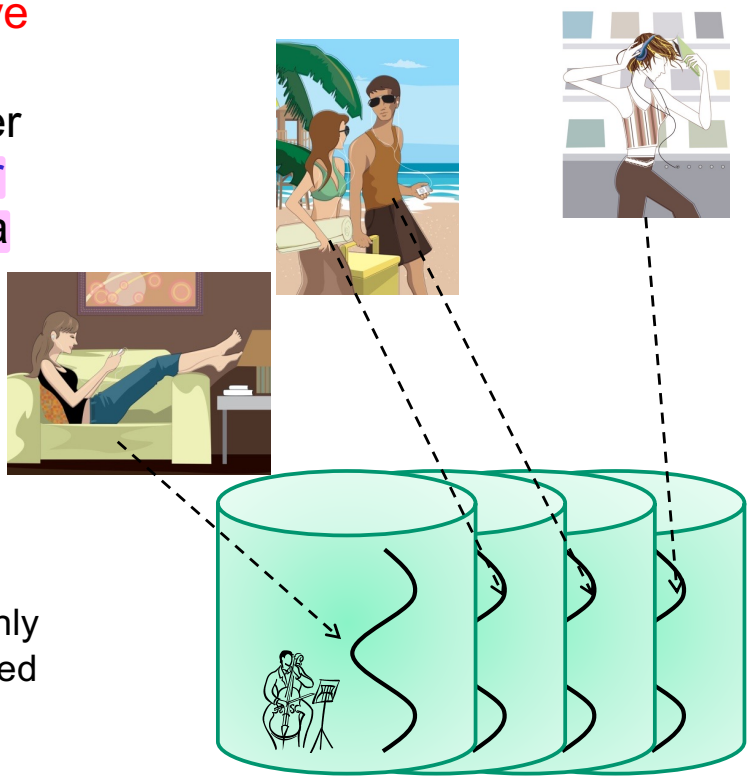
A simple problem: classic solution

- A very simple cloud computing application: video playback
- How to scale?
- "Classic" solution: multithreaded application that exploits multicore parallelism
- Cons: LIMITI SCALABILITÀ?
  - Design can be complex
  - Single failure impacts many users
  - n° thread, concorrenza



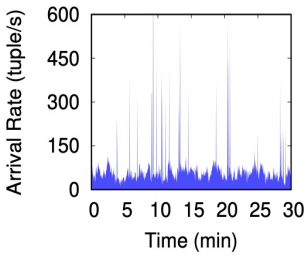
A simple problem: cloud solution

- A simpler cloud-native solution: a single-threaded video server instantiated once per user and running in a virtual machine or a container
  (∀ user ∃ 1 container)
- Pros:
  - Dramatically simpler design
  - If a player crashes, only a single user is affected



The real problem: scale and complexity

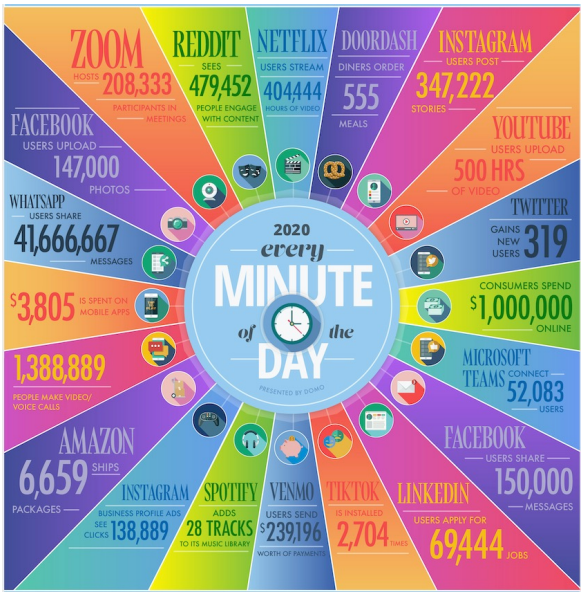
- How to realize a system/service that:
  - handles millions of requests per day
  - manages workload increase/decrease of one order of magnitude (or even more) in a quite short period
  - stores exabytes of data
    - 1 EB = 2^{60} B = 10^{18} B



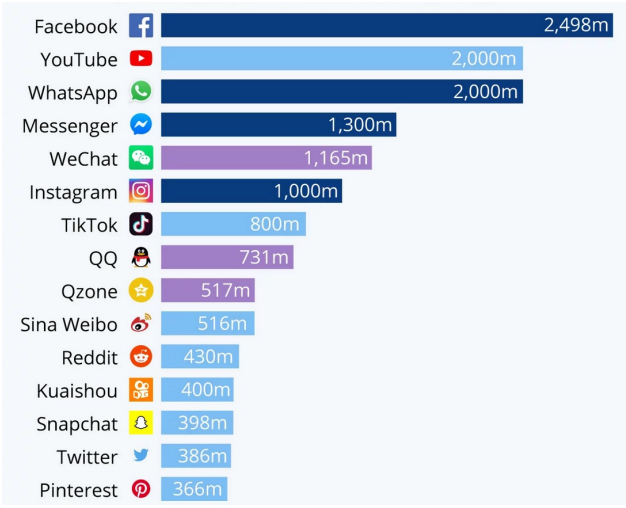
- There is a problem of scale of services!
- And scale changes every well known problem in computer research and industry

A taste of scale: scenario in 2020

- Almost 4G **smartphone users**
- More than 4G **Internet users**



Monthly active users



Impact of Covid-19 on Internet traffic

A Year in Lockdown: How the Waves of COVID-19 Impact Internet Traffic, ACM Comm., 2021

Figure 1. Traffic changes during the COVID-19 pandemic's spring and fall waves at our Internet vantage points.

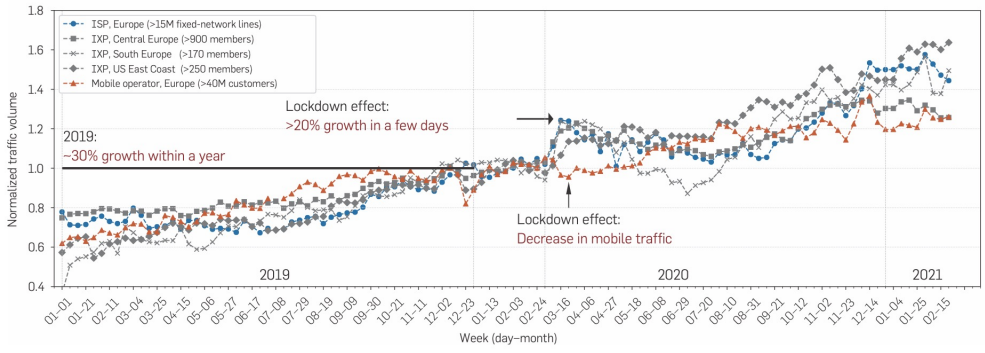
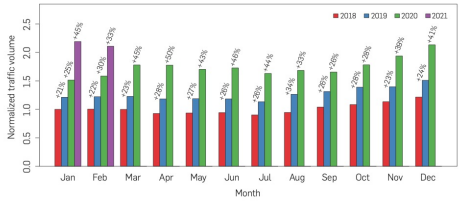


Figure 2. ISP monthly normalized downstream traffic change during the COVID-19 pandemic with percentage increase compared to the previous year.

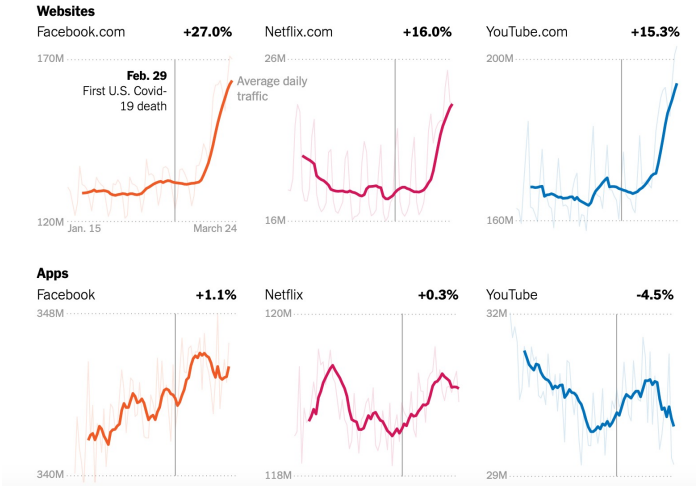


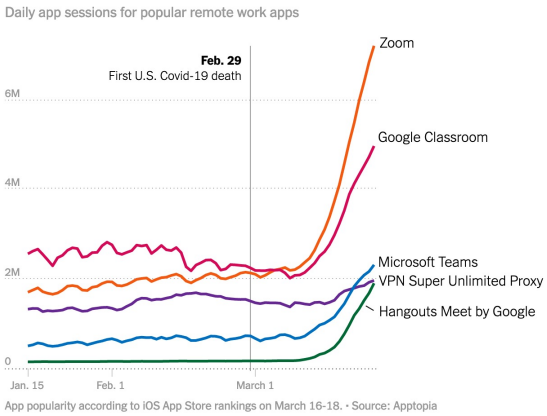
Impact of Covid-19 on Internet services

"The Virus Changed the Way We Internet", NY Times

Data from January 21 to March 24 2020

https://www.nytimes.com/interactive/2020/04/07/technology/coronavirus-internet-use.html





Some "old" and partial answers

- Utility computing
- Grid computing
- Autonomic computing
- Software as a Service (**SaaS**)
  - An "old" idea: application delivery on Internet, e non solo su server!

- … before Cloud computing (2006)
  - Cloud represents one step towards the solution of the scale problem

The origin: from 4 fundamental utilities…

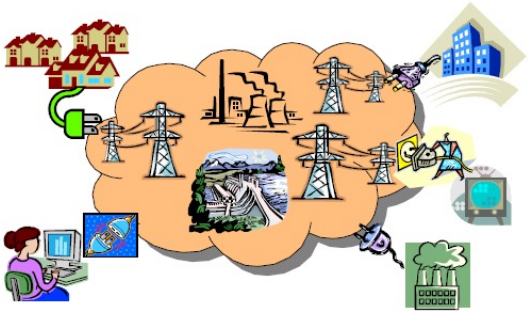
- Water

- Gas

- Electricity

- Telephony/Network

… to computing as the fifth utility



Utility computing: new idea?

- But this “computer utility” vision is not new!
- 1961: John McCarthy
  - “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.”

Utility computing: new idea?

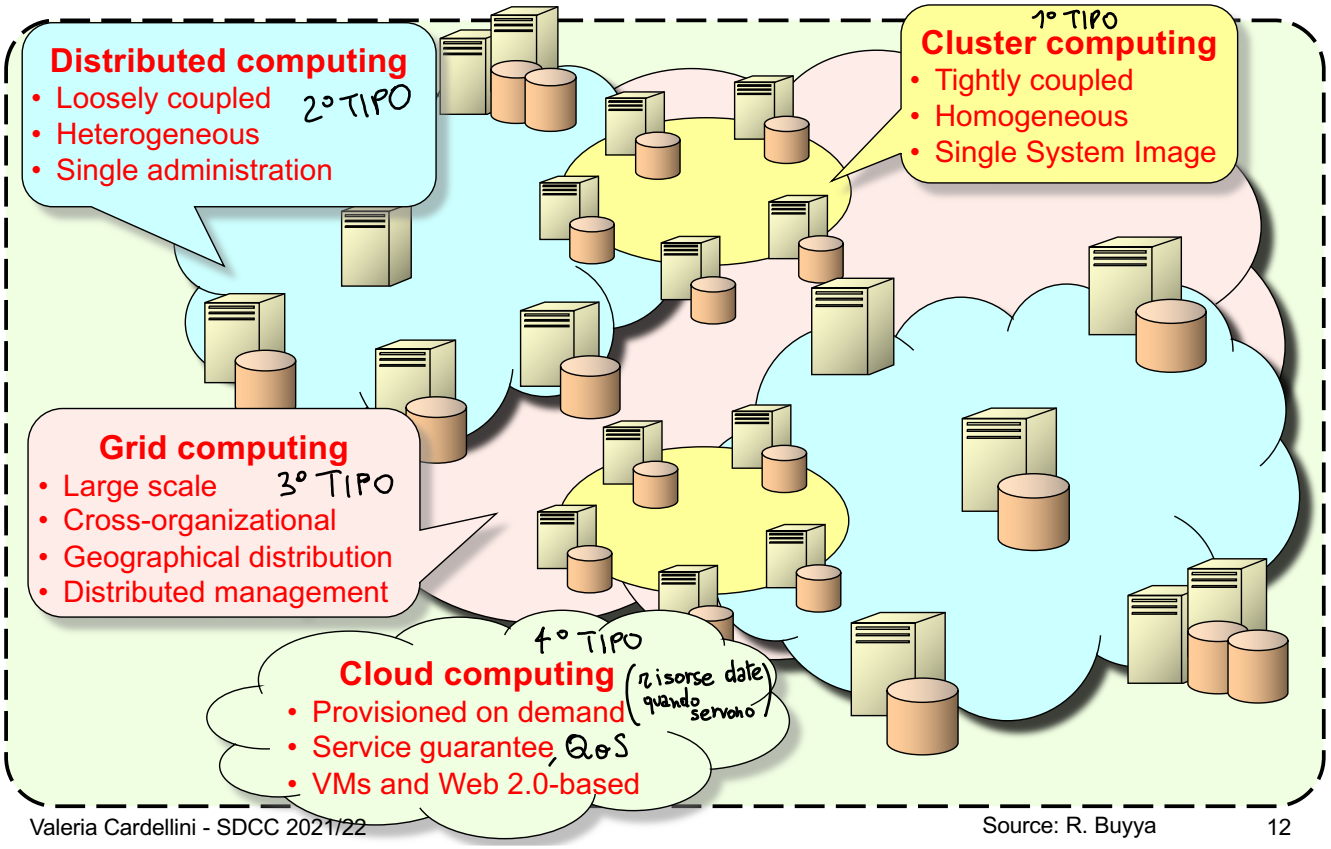
- 1969: Leonard Kleinrock, ARPANET project
  - "As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of "computer utilities", which, like present electric and telephone utilities, will service individual homes and offices across the country."
- Some computer re-definitions
  - 1984: John Gage, Sun Microsystems
    - "The network is the computer"
  - 2008: David Patterson, Univ. Berkeley
    - "The data center is the computer. There are dramatic differences between of developing software for millions to use as a service versus distributing software for millions to run their PCs"

Towards cloud computing, quando nasce?

- 2006: Jeff Bezos, Amazon: "Let us use our spare resource for making profit by offering them as services to the public"
  - In August 2006 Amazon launched Elastic Compute Cloud (EC2) and a paired online storage service called Simple Storage Service (S3) di tipo Infrastructure As a Service
  - Basic idea: let users rent data storage and computer server time from Amazon like a utility
  - Cloud computing was finally born
- 2011: "Cloud is the computer" (Rajkumar Buyya, Univ. Melbourne)

Amazon usava max il 20% della sua infrastruttura, perché non venderla quando non usata!? PAGO ∝ UTILIZZO

How do computing paradigms differ?



La scala aumenta sempre di più!

Cloud computing?

- What does it mean?
- How does it differ from other computing paradigms?
- How does it extend other computing paradigms?

Many technologies, concepts and ideas

Cloud computing

Cloud

computing

"Cloud" is the graphic symbol we use to represent Internet, è simbolo di 'Internet'

Computing includes:
- Computation
- Storage
- Coordination

spostamento della "computazione"

Cloud computing is about the shift of computing from a single server/data center to Internet

cloud è tutto stack, da infrastruttura a servizi.

A myriad of definitions…

- [Armbrust et al., 2009]: : "Cloud Computing refers to both the *applications delivered as services* over the Internet and the *hardware and software systems* in the data centers that provide those services. The services themselves have long been referred to as Software as a Service (SaaS), so we use that term. The data center hardware and software is what we will call a Cloud. … Cloud computing has the following characteristics: (1) The illusion of *infinite computing resources*… (2) The elimination of an up-front commitment by cloud users… (3) The ability to *pay for use*… as needed."

- [NIST, 2011]: 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. *(16th definition!)*

possibili interferenze!

- [Vaquero et al., 2009] 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 *SLAs*.

accordo sui livelli di servizi forniti

… that share some essential characteristics

- **On-demand self-service**
  - Cloud resources can be provisioned on-demand by the users, without requiring interactions with the cloud service provider

esempio: su AWS ho 5 VM, a seconda dell'uso. Dopo 5 VM, devo contattarlo!

- Broad **network access**
  - Cloud resources accessed over Internet using standard access mechanisms that provide platform-independent access
  - Published service interface/API

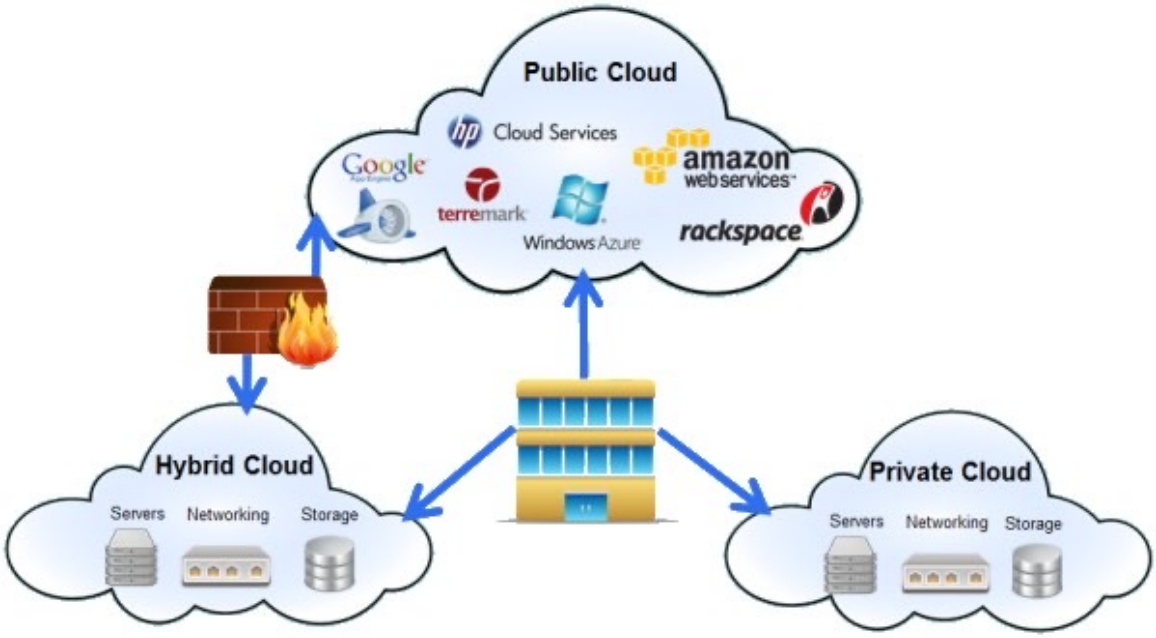
- Rapid **elasticity**
  - Elasticity: ability for customers to quickly request, receive, and later release as many resources as needed
  - Cloud resources can be provisioned rapidly and elastically. Cloud resources can be rapidly scaled out/in based on demand

richiede, riceve, rilascia nel modo più veloce possibile le risorse! Meglio se da solo, no amministratori esterni!

… that share some essential characteristics

- Resource **pooling**, condivise tra utenti!
  - Cloud resources are pooled to serve multiple users using multi-tenancy
  - Multi-tenancy: multiple users served by the same physical hardware
- Resources **virtualization**
  - Resources: storage, processing, memory, network bandwidth, and even data centers
- **Pay-per-use** pricing model / Pay as you go
  - No large up-front acquisition cost → NON PAGO "PRIMA".
- **Measured** service che implica "QUALITÀ ASSOCIATA"
  - Usage of cloud resources is measured and user is charged based on some specific metric

Cloud deployment models



Deployment models: public

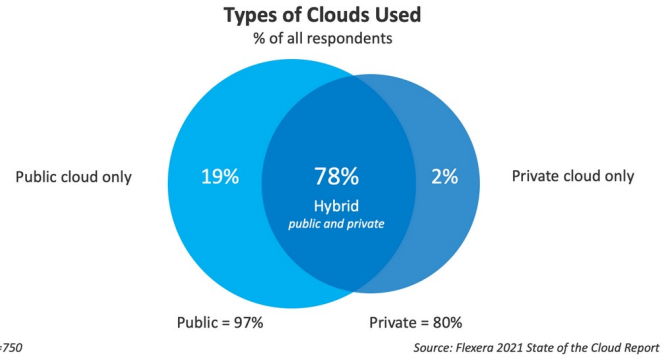
- Public cloud
  - Cloud infrastructure: provisioned for open use by the general public (multi-tenancy)
  - Owned, managed, and operated by a business, academic, or government organization, or some combination of them
  - Exists on premise of the cloud provider (servizi forniti sull'infrastruttura del cloud provider)
  - Services can be free or fee-based

Deployment models: private

- Private/enterprise cloud
  - Cloud infrastructure: provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units)
  - Owned, managed, and operated by the organization, a third party, or some combination of them
  - Exists on or off premise (magari infrastruttura non direttamente posseduta)
  - Pros: stronger security, customization
  - Cons: lower economic advantages, scalability more laborious

Deployment models: hybrid

- Hybrid cloud, sempre più diffuso!
  - Mixed use of private and public clouds
  - Cloud infrastructure: composition of two or more distinct cloud infrastructures (private or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability



5/10/22

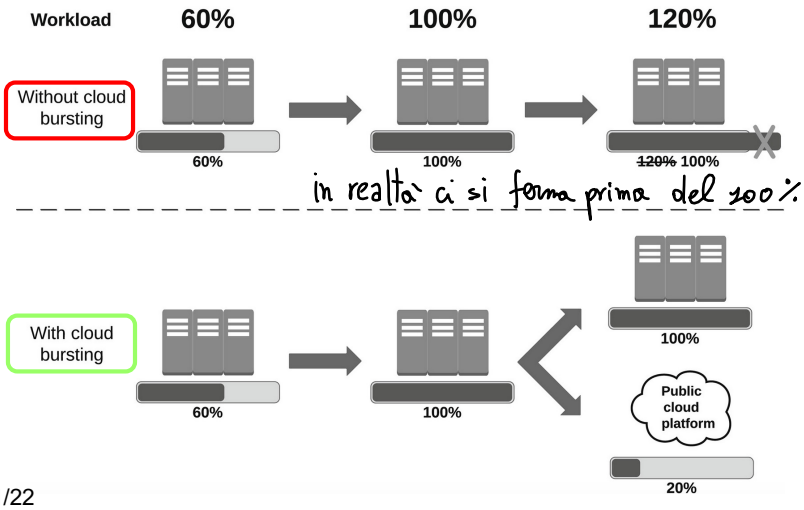
Deployment models: hybrid

- Motivations
  - Balance resources and costs
  - Differentiate privacy (keep sensitive data in private cloud)
  - Improve availability: public cloud for disaster recovery purposes in the event of an unexpected outage
  - Improve performance: cloud bursting
  - Flexibility → privacy "differenziato" mi vorrà cloud differenzia i servizi, a seconda dell'applicazione.
    → "più disponibile", uso public cloud in caso di disaster Rec. (Backup)
    → "distribuzione geografica", uso cloud pubblico per utenti in diverse regioni.
  - Reduce vendor lock-in risk
    ↳ tutti provider offrono abbastanza uguali, ma con interfacce diverse, limitando l'utente!

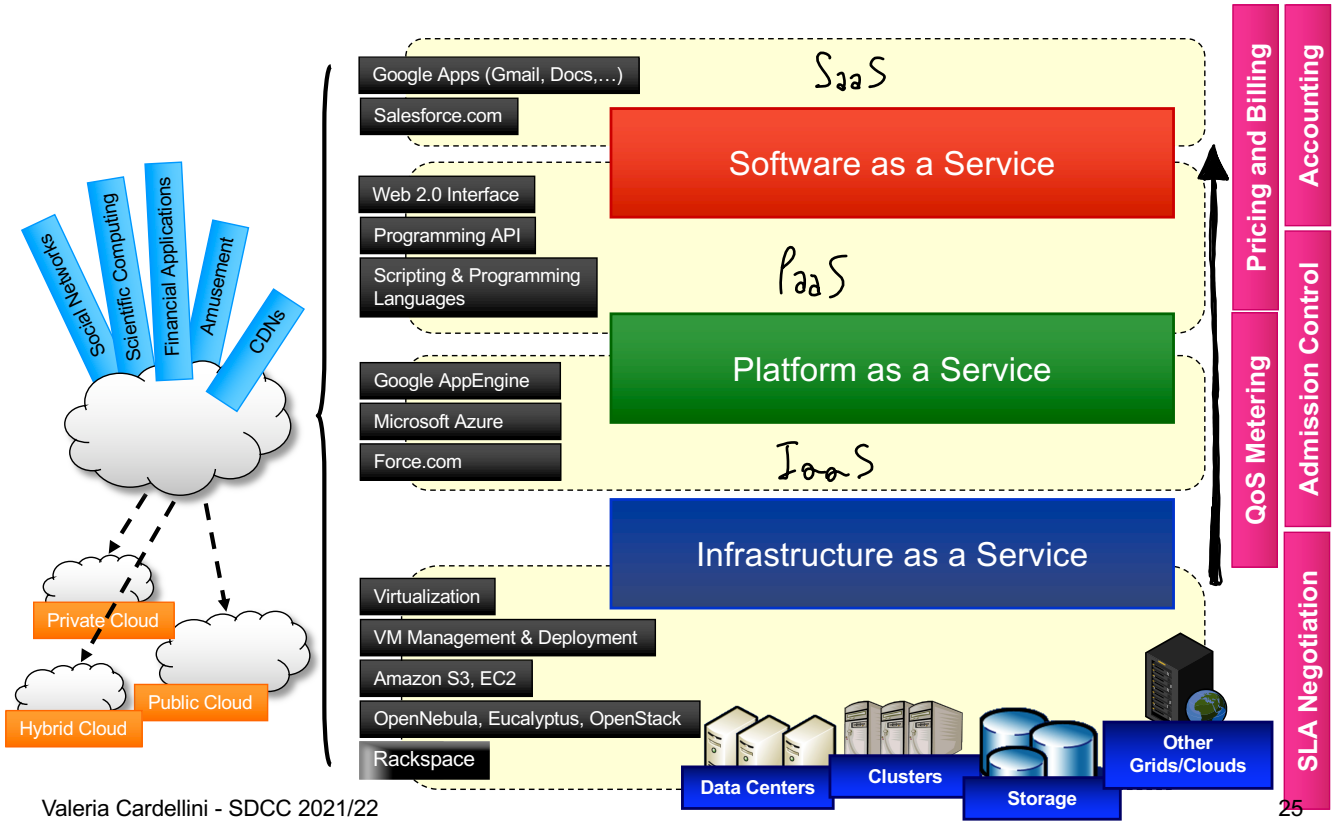
Hybrid cloud: cloud bursting

- Use a dynamic hybrid cloud (private + public clouds) to manage variable workload when private cloud capacity is insufficient
  - Private cloud is used to provide the application
  - Plus a public cloud used to manage load spikes that cannot be sustain by private cloud

parto con privato, passo al pubblico quando si "satura"



Cloud stack: service models



Service models: IaaS

- Infrastructure as a Service (IaaS)
  - Compute, storage and network resources as services
  - Customer capability
    - To provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software (including operating systems and applications), istanziare VM
  - Customer control or management
    - X No control over underlying cloud infrastructure
    - √ Control over operating systems, storage, deployed applications
    - Possibly limited control of selected networking components (e.g., host firewalls)C
    - scaling dovrei farlo io!

cosa può fare l'utente?

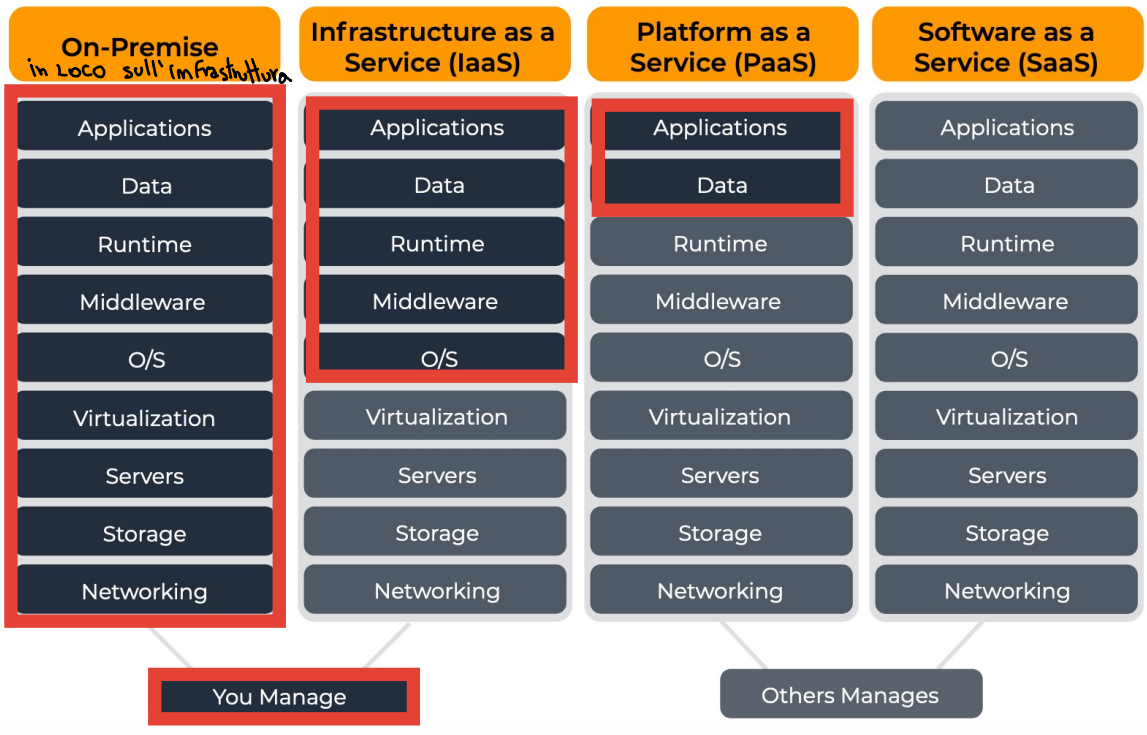
Service models: PaaS (provider fa lo scaling, NON io!)

- Platform as a Service (PaaS)
  - Platforms that allow customers to develop, run and manage scalable applications, without the complexity of building and maintaining the underlying infrastructure
  - Customer capability
    - To develop, deploy and test onto the cloud consumer-created or consumer-acquired applications realized using programming languages, application frameworks, development tools supported by PaaS provider
  - Customer control or management
    - No control over underlying cloud infrastructure (network, servers, operating systems, storage)
    - Control over deployed applications and possibly application hosting environment configurations

Service models: SaaS

- Software as a Service (SaaS), es GMAIL, non serve client
  - Applications made available to customers over Internet, still the largest market
  - Customer capability
    - To use SaaS provider's applications running on a cloud infrastructure
    - Applications accessible from various client devices through Web or provider APIs
  - Customer control or management, poco controllo!
    - No control over underlying cloud infrastructure
      - Network, servers, operating systems, storage, or even individual application capabilities
    - Possible exception: limited user-specific application configuration settings

Wrapping up the service models



IaaS: examples

HANNO INTERFACCE DIVERSE e PROPRIETARIE, NON STANDARDIZZATE

- Akamai: Intelligent Platform
- Alibaba Cloud: e.g., Elastic Compute Service
- Amazon Web Services: e.g., EC2 and S3
- Aruba Cloud
- CloudSigma
- DigitalOcean
- Google Cloud Platform: e.g., Compute Engine and Cloud Storage
- IBM Bluemix: e.g., Virtual Servers and Object Storage
- Microsoft Azure: e.g., Virtual Machines and Storage
- Flexera: e.g., Cloud Servers and Cloud Files
- Red Hat: Virtualization, Ceph Storage

IaaS example: AWS EC2

- Let us launch a virtual machine on EC2 from AWS Management Console
- Few steps in 2-3 minutes:
  1. Sign in to AWS Management Console https://signin.aws.amazon.com/
  2. Open Amazon EC2 console by choosing EC2 under Compute
  3. Choose AWS Region (e.g., EU Frankfurt)
  4. From EC2 dashboard, choose Launch Instance
  5. Choose software configuration (operating system, application server, and applications) from available Amazon Machine Images (AMIs)
     - AMI provided by AWS, user community, or AWS Marketplace
  6. Choose the instance type and configure it
     - T-shirt sizes: various combinations of CPU, memory, storage, and networking capacity
  7. If not yet configured, select the security group to open a specific network port (e.g., SSH) for the launched instance
  8. Select an existing key pair or create a new one to securely connect to the instance after it launches

PORT 22

PaaS examples

- Apprenda
- AWS Elastic Beanstalk
- Google App Engine, …
- IBM Cloud Foundry
- Microsoft Azure App Service
- Oracle Cloud Platform
- Redhat OpenShift
- Salesforce Platform, per le aziende
- More examples: https://paasfinder.org/vendors
- Also serverless computing/FaaS services (e.g., AWS Lambda, Google Cloud Functions)

Platform for web Apps

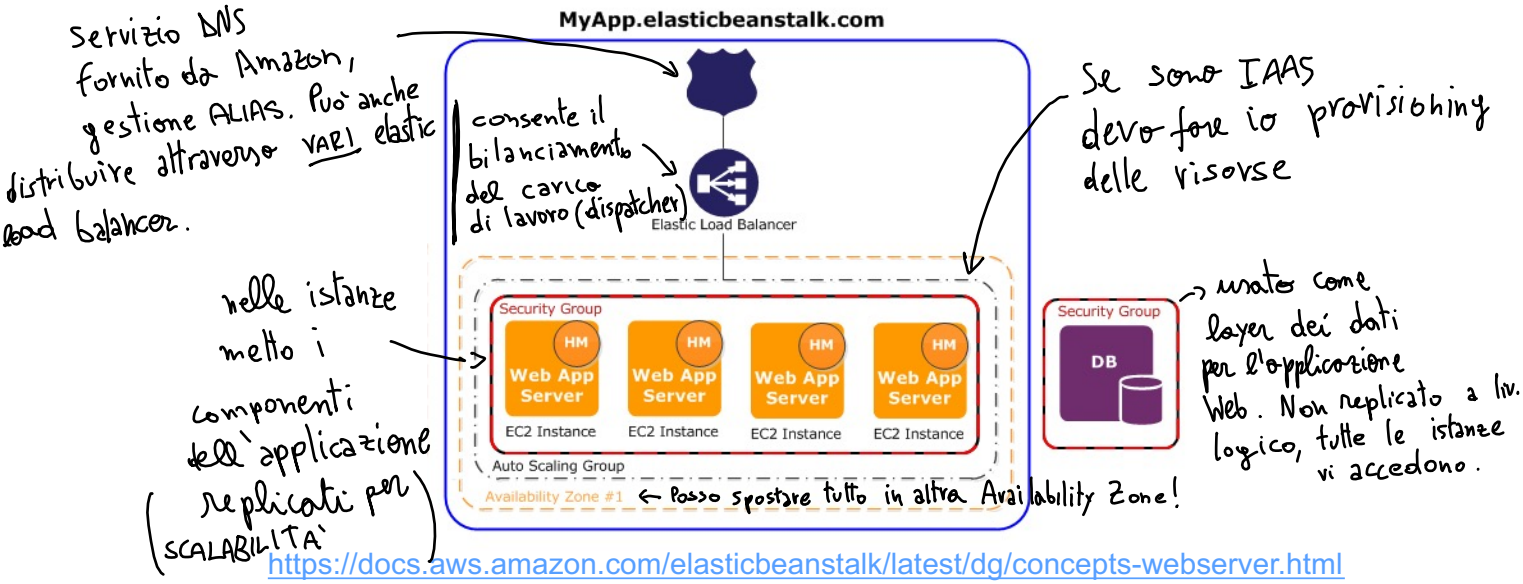
↓ Function as a Service

PaaS example: AWS Elastic Beanstalk

- Supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby
- Allows to deploy and manage applications in AWS Cloud without having to learn about the infrastructure
- Reduces management complexity
  - User uploads the application source bundle (e.g., .war file) to Elastic Beanstalk and provides some information about the application
  - Elastic Beanstalk automatically launches an environment and creates and configures the AWS resources needed to run the code, handling the details of capacity provisioning, load balancing, scaling, and application health monitoring

PaaS example: AWS Elastic Beanstalk

- Example of Elastic Beanstalk architecture for a web server environment tier
  - AWS resources (ELB, Auto Scaling group, EC2 instances) are automaticaly provisioned by Elastic Beanstalk



https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/concepts-webserver.html

Comparing IaaS and PaaS

- Using IaaS provider
  - "Pure" virtualized resources: CPU, memory, storage, network bandwidth, load balancer, ...
  - Operating system included
  - "T-shirt" size

- Using PaaS provider
  - Virtualized resources plus application framework
  - Additional services, e.g., automatic scaling without having to configure it
  - Constraints on the application and data architecture
  - Greater risk of vendor lock-in

Main IaaS and PaaS providers

Skip

- Gartner's magic quadrant for IaaS and PaaS

Figure 1: Magic Quadrant for Cloud Infrastructure and Platform Services

Source: Gartner (July 2021)

7/10/22

SaaS examples

- Communication & collaboration: Adobe Connect, Cisco WebEx, Google Mail, Zoom
- Personal productivity: Google Calendar, Google Drive, Microsoft 365
- File storage and sharing: Dropbox, OneDrive, iCloud, SugarSync
- CRM (Customer Relationship Management): salesforce.com
- Enterprise-level (accounting, ERP, HR, marketing, sales): NetSuite, SAP Business ByDesign, Workday, Zoho

Cloud pricing models

- Cloud service providers offer a number of billing models

- Pay-per-use/on-demand
  - Users are charged based on cloud resources usage
  - Pricing granularity depends on service, e.g., by hour for AWS EC2, by ms for AWS Lambda
  - Pro: maximum flexibility (no need to plan in advance)

- Fixed/reserved , usati per "servizi di calcolo"
  - Users are charged a fixed amount per month for cloud resources
  - Pro and con: significant discount compared to on-demand pricing but upfront payment
    pago meno se uso risorse in modo continuativo, ma pago in anticipo!

Cloud pricing models

- Spot $\subseteq$ Pay Per Use
  - Variable pricing for cloud resources driven by market demand (supply and demand), prezzo fluttua
  - Pro: significant savings
    - E.g., AWS EC2 spot instances can be acquired at up a 90% discount compared to on-demand pricing
      https://aws.amazon.com/ec2/spot/
  - Con: interruptible by cloud provider , possono interrompere le operazioni, le mie <=> le mie app sono tolleranti all'interruzione del servizio. (c'è un piccolo preavviso!)
    → use for fault-tolerant workloads

- dedicated
  - hardware FISICO totalmente dedicato all'user.
  - pro: HW dedicato e isolato
  - con: $

chi usa il cloud?

Which use of clouds? Some trends

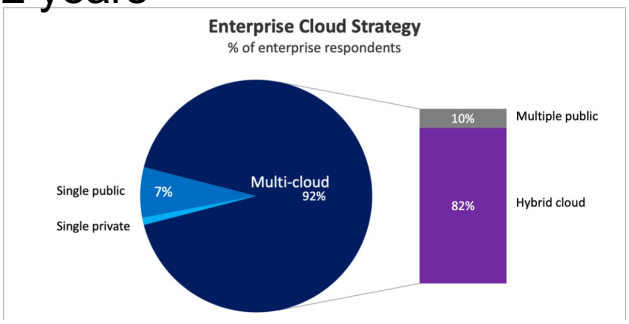
- Public cloud adoption is evolving
  - Top 3 public cloud providers remain AWS, Azure and Google
  - Use of public cloud PaaS services (e.g., relational DBaaS and IoT services) is rising
- Containers (e.g., Docker) and container orchestration engines (e.g., Kubernetes) are now mainstream
  - 65% of organizations use Docker, 58% Kubernetes
- Enterprises choose a multi-cloud strategy

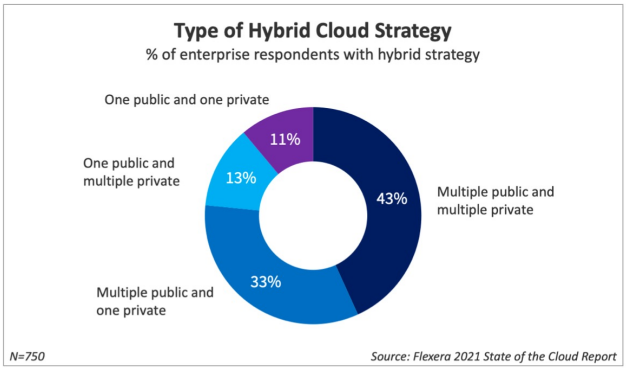
almeno ≥2 pubblici e privati

Trends: Multi-cloud

modelli di deployment

- Concurrent usage of multiple Cloud environments (hybrid, 2 or more only private, 2 or more only public)
- Increasingly popular in last 2 years
- Motivations
  - Reduce vendor lock-in risks
  - Improve cost efficiency
  - Increase flexibility
  - Satisfy data storage constraints (i.e., data physically stored in given country)
  - Improve geographic distribution
  - maggior tolleranza ai guasti





gestione dei multi cloud

Trends: Multi-cloud

- Multi-cloud management requires to:
  - Manage resources spread across several Cloud environments
  - Automate workload placement, configuration and maintenance, including identity management and data protection/encryption
  - Monitor performance of infrastructures and applications
  - Assess changes in service portfolio and pricing models of each Cloud provider
  - Summarize resource usage and costs
  - Predict usage and costs

- diverse policy sui dati (provider diversi)

- Can use
  - Purpose-built products, e.g., VMWare tools
  - Infrastructure-as-code (IaC) free tools to automate configuration and deployment: Chef, Puppet, Ansible and Terraform

→ di nostro interesse!

↳ usano un linguaggio simile a JSON.
ideali per più deployment sulla stessa applicazione.

Who use cloud?

- The back-end infrastructure used by the largest companies to provide popular and successful Internet services that are:
  - Scalable and elastic
  - Affordable
  - Reliable
  - Secure

- Some examples: Airbnb, Baidu, Dropbox, Foursquare, LinkedIn, Netflix, Shazam, Twitch and Twitter use commercial clouds

Capacity planning and elasticity

- Multi-tier applications (e.g., e-commerce, social networking, B2B) can experience rapid workload changes, il carico può variare rapidamente!
- Capacity planning: determines right sizing of each tier of the application deployment in terms of:
  - Number of resources
  - Capacity of each resource
  - For computing, storage, memory or network resources

  quantità/qualità/tipologia di risorse da usare!
- How? Forecast demand and scale-out(in) and/or scale-up(down) to manage workload variations

  (serie temporali o ML ↑ come? prevedo in avanti futuro ≠ passato)
- Key benefit for capacity planning provided by Cloud computing: **elasticity**

Forecast e/o scaling alla base dell'elasticità.

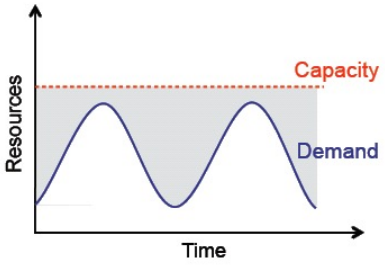
Traditional approach: over-provisioning

- Over-provisioning: resource provisioning by taking into account peak loads, mi baso sul picco di carico.
- Leads to excess capacity and under-utilization
  - Server utilization in traditional data centers
    - Typically < 20%; rarely 30%

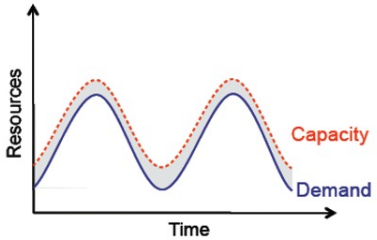
Cons → Higher costs than required, $

Cons → Increased energy requirement and consumption

We' d like to have



Static data center

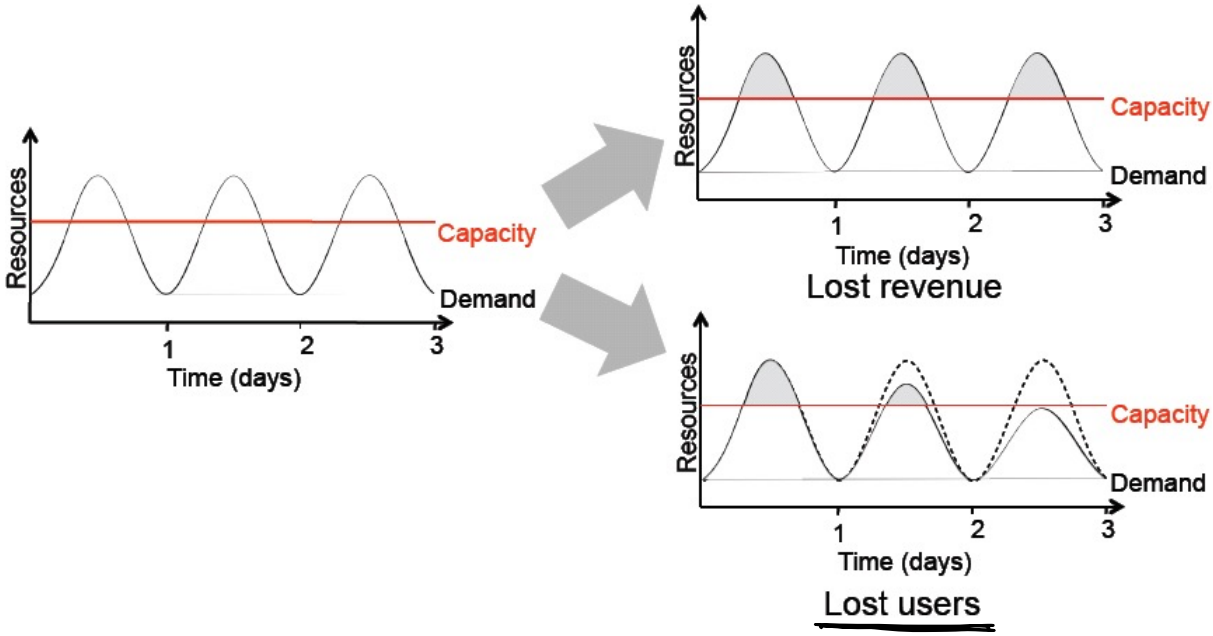


Data center in the cloud

Unused resources

Traditional approach: under-provisioning

- Under-provisioning leads to overload, poor performance and loss of opportunity to serve customers

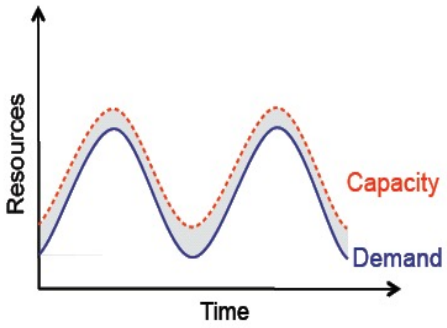


Elasticity

- Elasticity is the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible

Non c'è un amministratore!

Herbst et al., Elasticity in Cloud Computing: What It Is, and What It Is Not, ICAC 2013

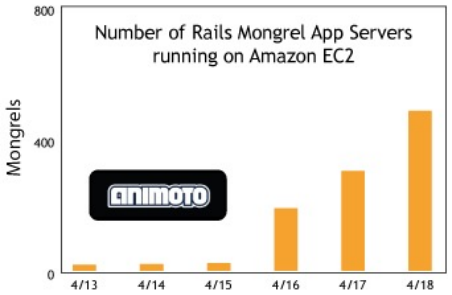


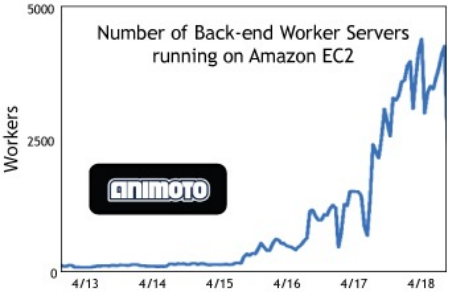
Data center in the cloud

Unused resources

Elasticity: an "old" example SKIP

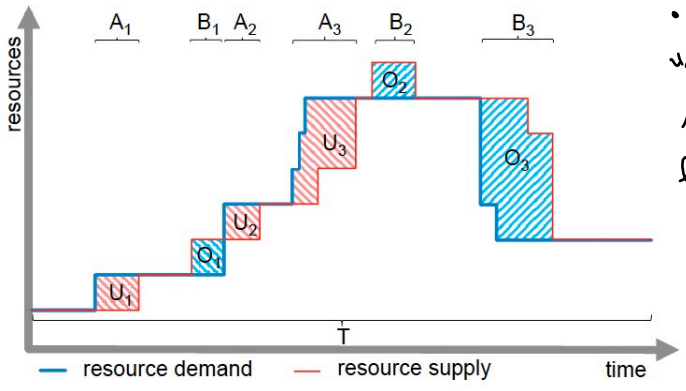
- Animoto app on Facebook (April 2008)
  - From 25K to 750K users just in 3 days
  - A peak of 20K new users in one hour





How to measure elasticity

- Consider over-provisioning and under-provisioning



• quando alloco nuova "macchina" passa sempre un Δt di tempo, in cui l'applicazione è in underprovisioning. Quando non serve più e dealloco passa un altro Δt, in cui sono in overprovisioning

- Various elasticity metrics, two examples:
  - Accuracy: sum of areas of over-provisioning (O) and under-provisioning (U) for the duration of the measurement period T
  - Timing: total amount of time spent in the over-provisioning (B) and under-provisioning (A) over the measurement period T

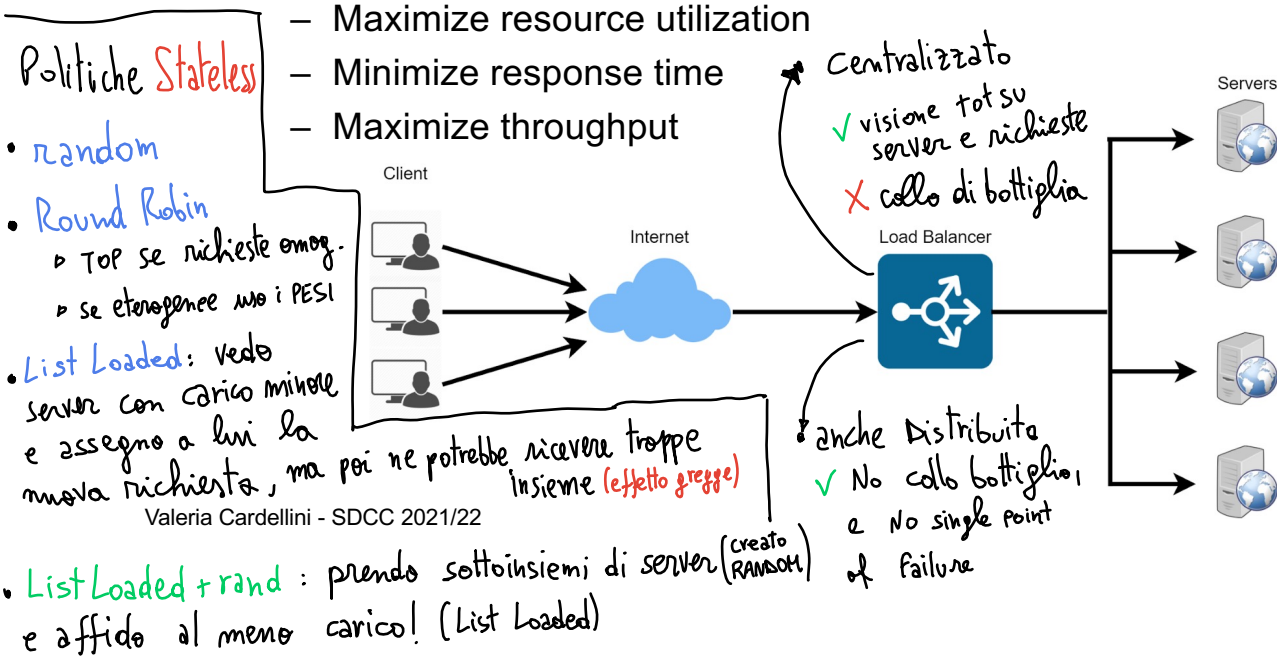
metriche

Σ(area • + •), ottimale se $\Sigma = \emptyset$

$\sum A_i + B_i$, ideale se $\to \emptyset$

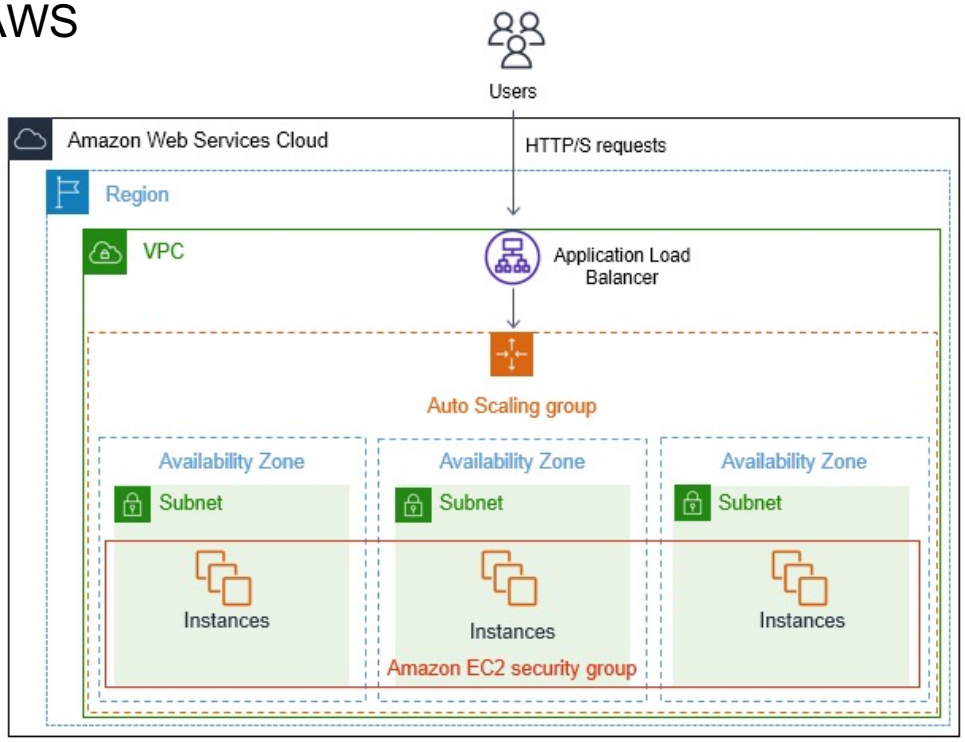
Elasticity: architectural point of view

- How to manage requests distribution among multiple replicas?
- Load balancer
  - Can be centralized or distributed
- Possible goals of load balancing techniques:
  - Maximize resource utilization
  - Minimize response time
  - Maximize throughput



Elasticity: architectural point of view

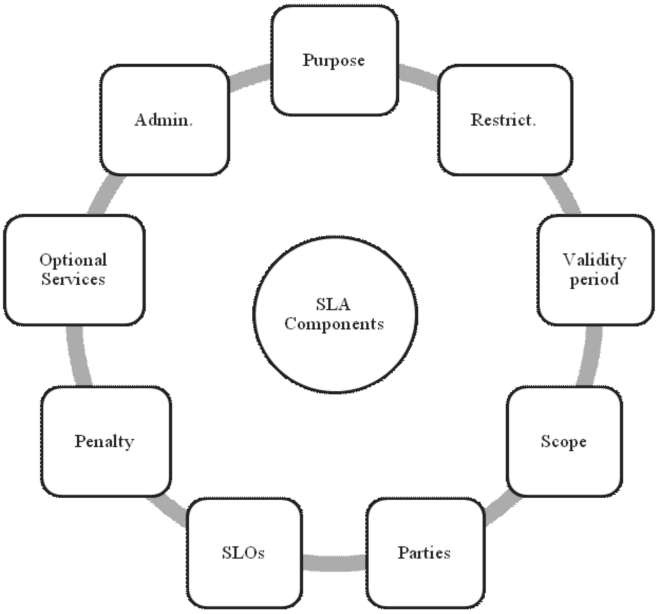
- Example of load balancing and auto-scaling using AWS



"contratto" per def. livelli di servizio

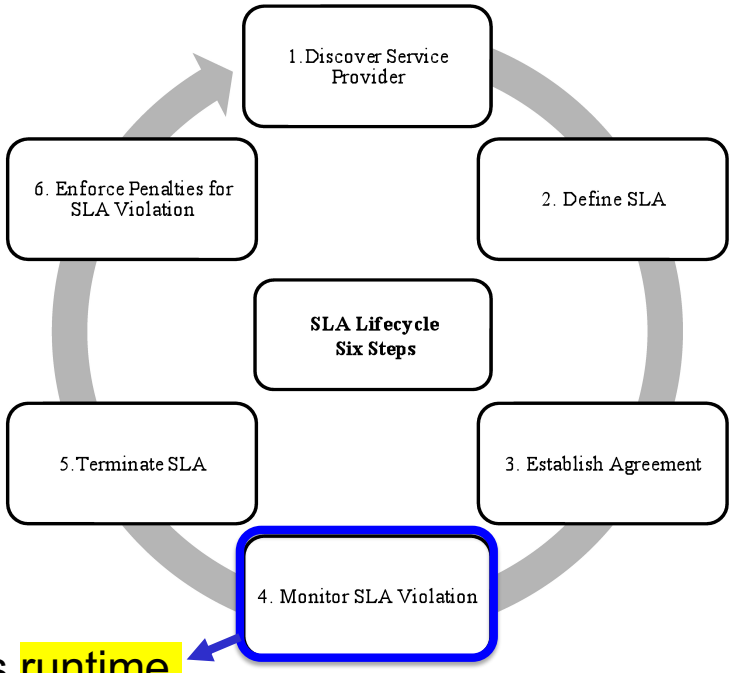
Service Level Agreement (SLA)

- Formal agreement (contract) between a provider and a consumer of a service
- Composed of one or more Service Level Objectives (SLOs):
  - SLO: condition on a measure of a specific QoS parameter (e.g., maximum response time, mean response time, availability)
- Includes penalty and/or compensation in case of SLA violation



Source: Sun Microsystems Internet Data Center Group

SLA life cycle



Requires runtime monitoring of the service, se ci sono, è l'utente che deve segnalarlo!

Si parla di Service Level Indicator, MISURABILI. Nei CLOUD NO response time, ma UPTIME (disponibilità)

Example: SLA for AWS Compute Service

- **Service Commitment** https://aws.amazon.com/compute/sla/

COSA OFFRE

AWS will use commercially reasonable efforts to make each Included Service available for each AWS region with a Monthly Uptime Percentage of at least 99.99% during any monthly billing cycle. In the event any of the Included Services do not meet the Service Commitment, you will be eligible to receive a Service Credit as described below.

  - Monthly Uptime Percentage 99.99% = 4 min 23 sec of downtime per month

regola dei '9'.

- **Service Commitment and Service Credits** Uptime? https://uptime.is/ → Prob che contattando il servizio, esso risponde!

Gli danno crediti futuri, non soldi indietro! e devo fare Io richiesta

Service Credits are calculated as a percentage of the monthly bill (excluding one-time payments such as upfront payments made for Reserved Instances)

Monthly Uptime Percentage	Service Credit Percentage
Less than 99.99% but equal to or greater than 99.0%	10%
Less than 99.0% but equal to or greater than 95.0%	30%
Less than 95.0%	100%

We will apply any Service Credits only against future payments…

To receive a Service Credit, you must submit a claim…

Service Level Objective: predicato su set di SLI: uptime mensile (SLI) è del 99.99% (SLO)

• Preferiamo storage in cloud rispetto al classico poiché ho maggior disponibilità (dati in più aree)

12/10/22

Example: SLA for Amazon S3

• Saper leggere le SLA è utile per scegliere i servizi! https://aws.amazon.com/s3/sla/

- **Service Commitment**

AWS will use commercially reasonable efforts to make Amazon S3 available with a Monthly Uptime Percentage during any monthly billing cycle. In the event Amazon S3 does not meet the Service Commitment, you will be eligible to receive a Service Credit as described below.

Monthly Uptime Percentage 99.9% = 43 min 49 sec of downtime per month

- **Definitions**
  - "Error Rate" means: (i) the total number of internal server errors returned by Amazon S3 as error status "InternalError" or "ServiceUnavailable" divided by (ii) the total number of requests during that 5-minute period. We will calculate the Error Rate for each Amazon S3 account as a percentage for each 5-minute period in the monthly billing cycle. The calculation of the number of internal server errors will not include errors that arise directly or indirectly as a result of any of the Amazon S3 SLA Exclusions.
  - "Monthly Uptime Percentage" is calculated by subtracting from 100% the average of the Error Rates from each 5-minute period in the monthly billing cycle. If you did not make any requests in a given 5-minute interval, that interval is assumed to have a 0% Error Rate.

Se in 5 min ho 0,001% error rate

Monthly Uptime Percentage = $100\% - \sum_{5\,min}^{mese}$ error rate

Issues with Cloud SLAs

- Another example:
  https://cloud.google.com/compute/sla
- Some issues on Cloud SLAs:
  - SLA jargon
  - Lack of guarantees in terms of service performance
    - No end-to-end response time (network not controlled by Cloud provider), NO GARANZIE per questo Δt. tempo risposta > tempo di latenza
  - Service credits only for future payments
  - Burden of detecting SLA violation on Cloud customers
  - Non-negotiable or customizable SLA for most users
  - SLA violation reporting time period
  - Difficult to compare SLAs of different providers, usando metriche diverse la comparazione è difficile!

S. Basat, Cloud SLAs: Present and Future, *ACM SIGOPS Oper. Syst. Rev. 46(2),* 2012.

Cloud monitoring

- Goal: to keep track of the health of applications and services deployed in the cloud
- Monitoring service: allows cloud users to collect and analyze data on various system-oriented and application-oriented metrics from cloud resources

  monitoraggio fornito dai provider stessi o da enti terzi.
  - By Cloud provider: e.g., Amazon CloudWatch, Google Cloud Monitoring (a pagamento)
  - By third party: e.g., Dynatrace, Prometheus

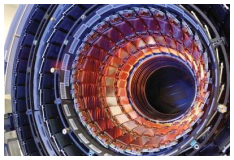
Some system-oriented monitoring metrics

Type	Metrics
CPU	CPU utilization
Disk	Disk utilization, throughput (MB/sec for read/write, operations/sec)
Memory	Memory-Used, Memory-Free, Page-Cache
Interface	Throughput (Mbps incoming/outgoing)

Cloud applications

- Scientific/technical apps
- Healthcare apps
- Business apps
- Consumer/social apps
- …

Healthcare, scientific and technical apps

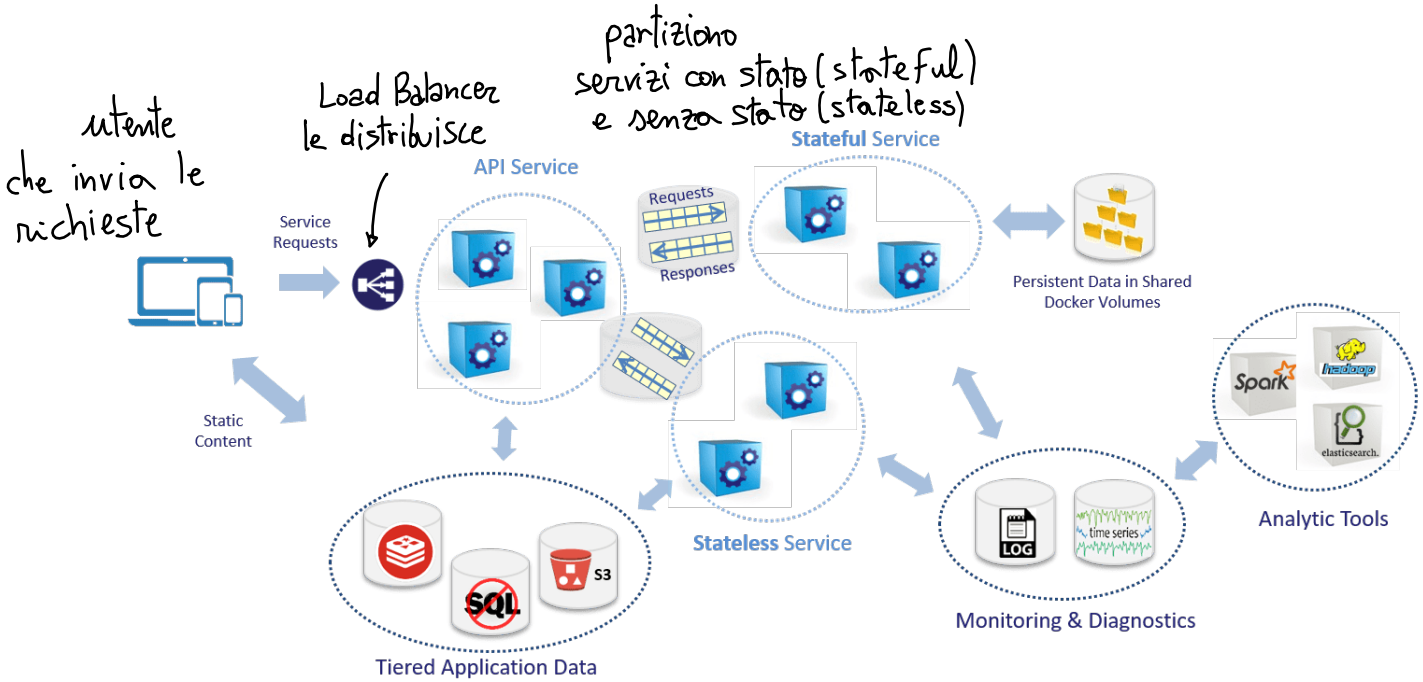




Business apps

Consumer/social apps



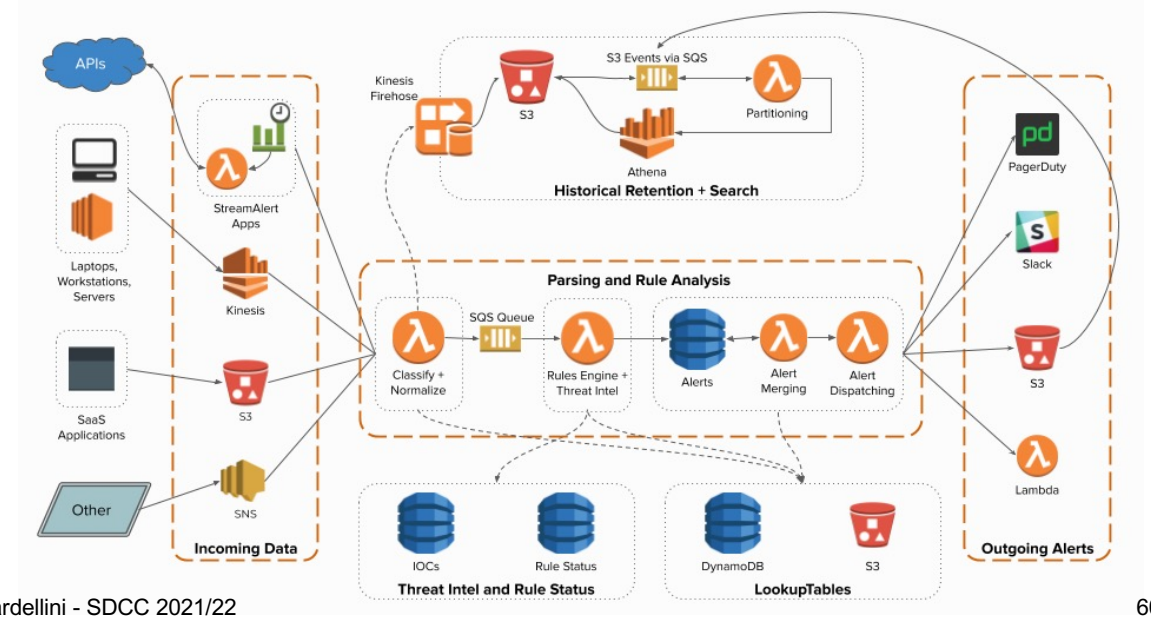

A high-level example of Cloud-native app



StreamAlert non si preoccupa di istanziare il server, bensì specificare la funzione da eseguire + dettagli tecnici per il corretto funzionamento.

A real example of cloud-native app

- StreamAlert is a serverless, real-time data analysis framework which allows you to ingest, analyze, and alert on data from any environment, using data sources and alerting logic you define. Computer security teams use StreamAlert to scan TB of log data every day for incident detection and response https://streamalert.io



Cloud application deployment

≐ processo iterativo formato dalle seguenti fasi:

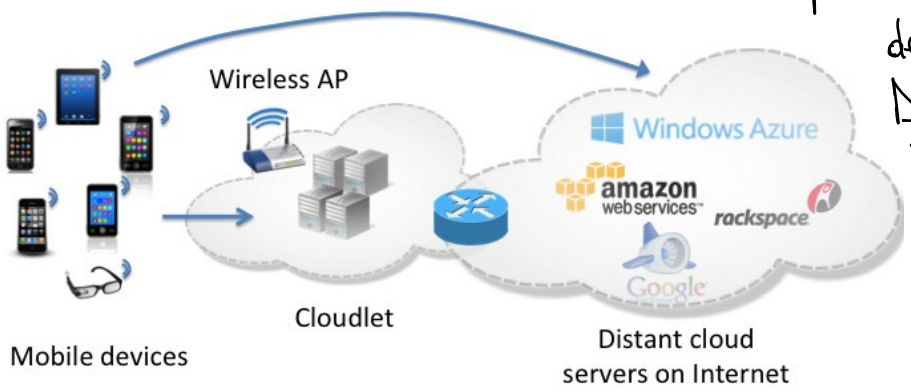
- Iterative process that involves:
  - Deployment design: scelgo servizi, uso Load Balancing? come replico?
    - Type and capacity of cloud resources in each tier (computing, memory and storage), interconnection, load balancing and replication strategies
  - Performance evaluation: come si comporta la nostra applicazione?
    - Verify whether app meets performance requirements
    - Involve monitoring app workload, performance parameters (e.g., response time, throughput) and resource utilization (CPU, memory, disk, I/O, etc.)
  - Deployment refinement: uso le valutazioni come feedback!
    - Consider various alternatives
      - Scaling (horizontal and vertical)
      - App components interconnections
      - Load balancing and replication strategies

Next-generation Cloud applications

- Convergence of different technologies, all in the Cloud:
  - Internet of Things
  - Big Data
  - Artificial Intelligence

Next generation computing scenario: Fog/edge computing

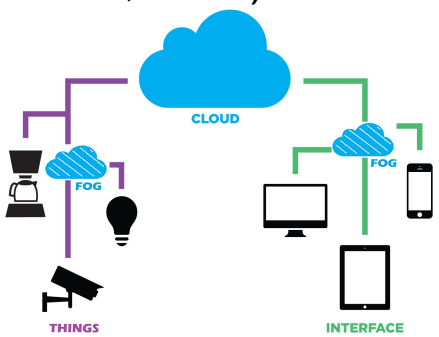
- How to manage computing- and data-intensive apps (e.g., speech recognition, assisted vision, augmented reality) with data originated from mobile and IoT devices?
  - Cloud-only solutions can be impractical due to latency
- Idea: **offload** computation and storage to nearby, one-hop resources in a micro data center (*cloudlet*) located at the network edge , cioè faccio computare agli estremi della rete! Diminuisco anche i tempi di latenza.



Con il FOG computing ho architettura a più livelli composti da nodi FOG e il cloud!

Fog computing: definitions

- "Fog Computing is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud computing data centers, typically, but not exclusively located at the edge of network." (Bonomi et al., 2012)
- "A horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum." (OpenFog consortium, 2017)



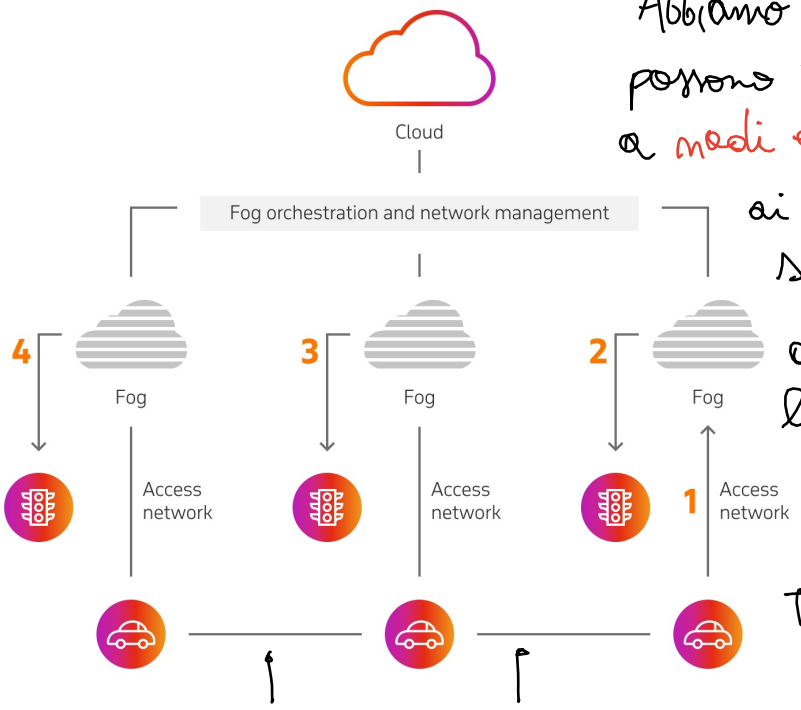
Fog vs. edge computing

- Both have their roots in content delivery networks and peer-to-peer trends of the early millennium
- Both have a decentralized IT architecture
- Edge computing: similar to Fog computing, but Fog is more tightly integrated with Cloud

esempio alto livello di app che sfrutta il Fog computing.

Putting all together: example of next-generation applications

- Road traffic regulation



Abbiamo dei "veicoli" che possono trasmettere info a nodi edge, siti "vicini" ai semafori. Lo scopo è regolare dinamicamente la durata dei semafori in funzione del traffico effettivo

Anche i nodi scambiano info reciproche, per definire i comportamenti corretti.

Summing up: main Cloud benefits

IT benefits

- Scalability, elasticity, flexibility
  - "Infinite" amount of resources
- Anytime and anywhere accessibility
  - Cloud services accessible through Internet/Web
- Organizational and operation agility
- Simple management
  - E.g., software updates and versioning

Business benefits

- No advanced payment
  - Costs scale with use
  - CapEx to OpEx (costi capitale trasformati in costi operativi)
- Rapid business innovation
- Increase in productivity

Environment benefits

- Lower environmental impact due to ICT
  (Bitcoin non scala per motivi energetici)

Issues for Cloud customers

- Privacy, security and legal issues
  - Where is cloud data physically located?
    - E.g., many European businesses want to retain sovereignty over their data
  - Who can access cloud data?
  - Is data encrypted, also in transit?
  - What about data integrity, tracing and recovery?
  - Is the Cloud provider GDPR compliant?

- In the past, stormy clouds
  - "You have zero privacy anyway. Get over it." (Scott McNealy, co-founder of SUN, 1999)
  - "If you have something that you don't want anyone to know, maybe you shouldn't be doing it in the first place.... The reality is that search engines do retain information... It could become available later..." (Eric Schmidt, Google CEO, 2009)

Issues for Cloud customers

- Now?
  - In Europe CISPE Code of Conduct https://www.codeofconduct.cloud
  - Be careful when you use Cloud storage and other free services
    - "We use the information we collect in existing services to help us develop new ones. For example, understanding how people organized their photos in Picasa, Google's first photos app, helped us design and launch Google Photos." (Google, 2021)

    https://policies.google.com/privacy?hl=en-US
    https://www.nytimes.com/interactive/2019/07/10/opinion/google-privacy-policy.html
    - "Microsoft uses the data we collect to provide you with rich, interactive experiences. (…) We also use the data to operate our business, which includes analyzing our performance, meeting our legal obligations, developing our workforce, and doing research." (Microsoft, 2021)

    https://privacy.microsoft.com/en-gb/privacystatement

Issues for Cloud customers

- Communication latency ← Se voglio "bassa latenza" NON uso cloud computing, bensì il Fog/edge model
  - ms latency from users and devices to cloud providers
  - Real-time and low-latency apps?

- Cloud portability
  - Customers ability to move and suitably adapt their apps and data between their own systems and cloud services, and between cloud services of different providers and potentially different cloud deployment models
  - Risks of vendor lock-in
    - E.g., specific features offered by PaaS provider that complicate switching to another provider
    - E.g., difficulty in getting data out and migrate to different provider
  - Towards portability
    - Containers
    - Automation tools

Issues for Cloud customers

- Cloud interoperability
  - Capability of public cloud services, private cloud services, and other diverse systems within the enterprise to understand each other's application and service interfaces, configuration, forms of authentication and authorization, data formats, etc. in order to work efficiently and cooperate effectively together
  - Need of standards for Cloud portability and interoperability
    - Open Virtualization Format (OVF), Topology and Orchestration Specification for Cloud Applications (TOSCA), Open Container Initiative (OCI), OpenShift
    - P2302 Standard for Intercloud Interoperability and Federation by IEEE and NIST

- Poor support for SLA negotiation and management
  - Lack of SLAs based on end-user experience
  - Cloud customer often in charge of SLO monitoring

Issues for Cloud customers

- The two faces of scalability and elasticity
  - Number of resources from a cloud provider is virtually infinite, but
  - Is your application able to scale similarly?
    - Rethink application design for the Cloud!
  - Is your Cloud provider able to support real elasticity?
    - That is, to *automatically* provision and de-provision resources on demand (without *any* manual reconfiguration) as workloads change, while maintaining availability and performance

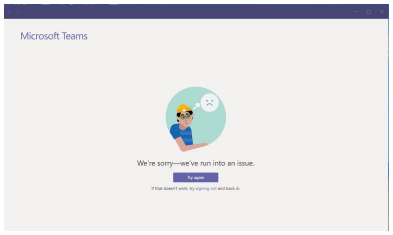
Mettere applicazione su cloud ≠ rendere applicazione scalabile.
Per la scalabilità ci vuole conoscenza & progettazione.

Issues for Cloud customers esempi di "down service"

- Cloud infrastructure outage, some examples:
  - February 28, 2017: AWS S3 went down for 4 hours, taking Slack and several media sites down with it
    - Due to human error http://amzn.to/2fJhcfg

  - On September 29, 2020: global Microsoft outage brought down Teams, Office 365 and Outlook

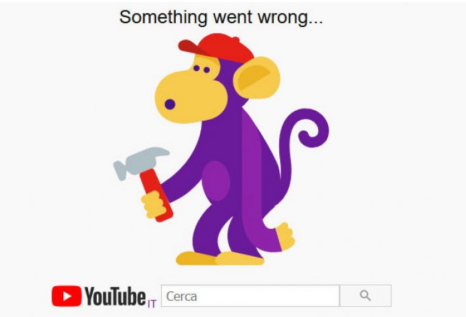
https://www.theguardian.com/technology/2020/sep/29/major-microsoft-outage-brings-down-office-365-outlook-and-teams



Issues for cloud customers: cloud infrastructure outage

- On December 14, 2020: global Google outage affected authenticated users of most Google services, including Gmail, YouTube, Google Drive, Google Docs, Google Calendar and Google Play

https://www.theguardian.com/technology/2020/dec/14/google-suffers-worldwide-outage-with-gmail-youtube-and-other-services-down



- Cloud outages in 2021: see CRN slideshows ← lista di reportage.

Issues for cloud providers

- Uncertainty and variability in service demands
  - Example of solution: spot instance market to sell unused Amazon EC2 capacity
- SLA management
  - Cascade effects
- Cloud customer requirements
- Energy management
- Cloud interoperability

Summing up

- Cloud computing is:
  - here to stay and continues to be one of the most hyped subjects in IT
  - increasingly becoming an integral concept in IT
  - a major trend in the way digital services are designed, implemented and delivered
  - a key factor for the digitalization of the whole society

- However, Cloud systems are very complex
  - Many heterogeneous resources and technologies, applications with heterogeneous workloads and complex interactions between systems and networks, i.e., all difficult aspects of large-scale distributed systems
  - Recall: “old” problems become new problems on a totally different scale and open to new solutions