TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Macroarea di Ingegneria
Dipartimento di Ingegneria Civile e Ingegneria Informatica

Introduction to Cloud Computing

Corso di Sistemi Distribuiti e Cloud Computing
A.A. 2022/23

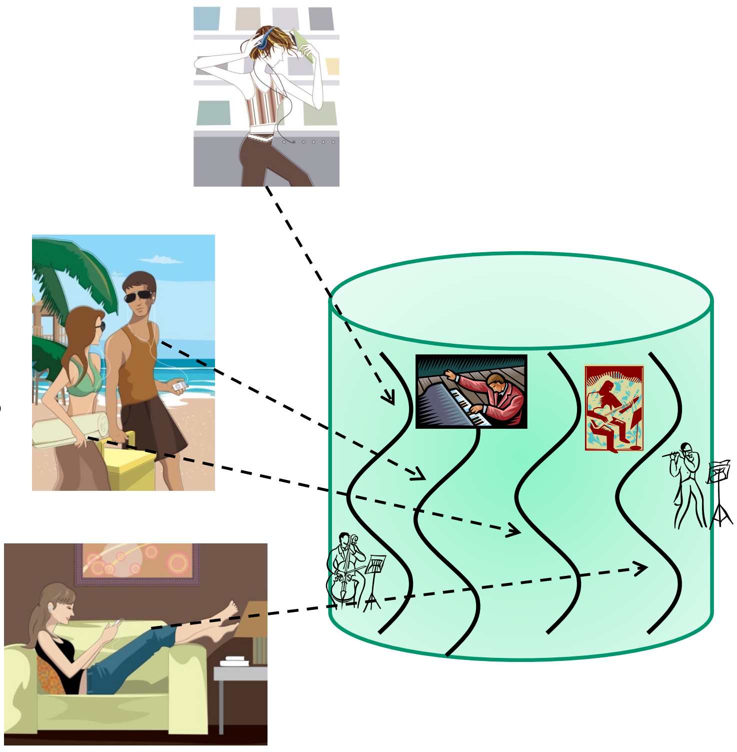
Valeria Cardellini

Laurea Magistrale in Ingegneria Informatica

A simple problem: classic solution

- A simple application: video playback
- How to scale it?

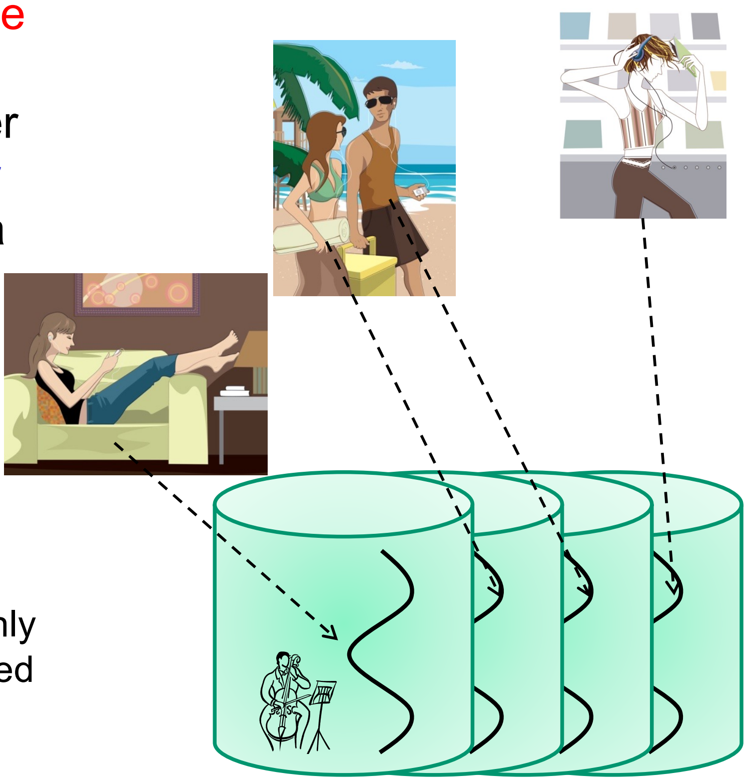
- "Classic" solution: multithreaded application that exploits multicore parallelism
- Cons:
  - Design can be complex
  - Single failure impacts many users



A simple problem: cloud solution

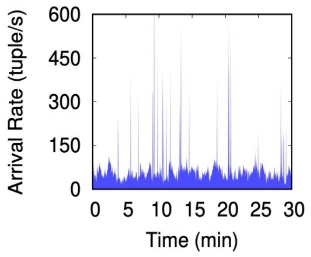
- A simpler cloud-native solution: a single-threaded video server instantiated once per user and running in a virtual machine or a container

- Pros:
  - Dramatically simpler design
  - If a player crashes, only a single user is affected



The real problem: scale and complexity

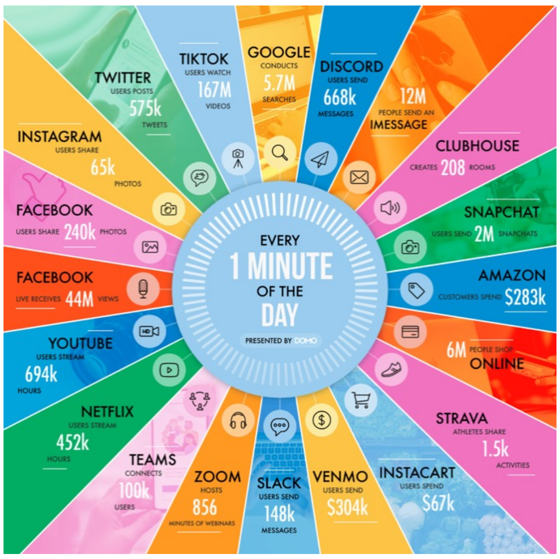
- How to realize sw services that:
  - handle millions of requests per day
  - manage workload increase/decrease of one order of magnitude (or even more) in a quite short period
  - store EBs of data
    - 1 EB = 2^{60} B = 10^{18} B

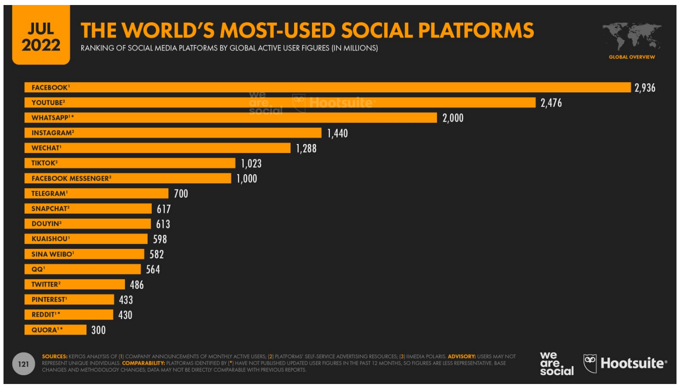


- There is a problem of scale of services!
- And scale changes every well known problem in computer research and industry

A taste of scale: scenario in 2022

- More than 6G **smartphone users**
- Almost 5G **Internet users**


EVERY 1 MINUTE OF THE DAY
PRESENTED BY DOMO
TWITTER USERS POSTS 575k TWEETS
TIKTOK USERS WATCH 167M VIDEOS
GOOGLE CONDUCTS 5.7M SEARCHES
DISCORD USERS SEND 668k MESSAGES
12M PEOPLE SEND AN IMESSAGE
CLUBHOUSE CREATES 208 ROOMS
SNAPCHAT USERS SEND 2M SNAPCHATS
AMAZON CUSTOMERS SPEND $283k
6M PEOPLE SHOP ONLINE
STRAVA ATHLETES SHARE 1.5k ACTIVITIES
INSTACART USERS SPEND $67k
VENMO USERS SEND $304k
SLACK USERS SEND 148k MESSAGES
ZOOM HOSTS 856 MINUTES OF WEBINARS
TEAMS CONNECTS 100k USERS
NETFLIX USERS STREAM 452k HOURS
YOUTUBE USERS STREAM 694k HOURS
FACEBOOK LIVE RECEIVES 44M VIEWS
FACEBOOK USERS SHARE 240k PHOTOS
INSTAGRAM USERS SHARE 65k PHOTOS



Monthly active users

https://www.domo.com/learn/infographic/data-never-sleeps-9

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.

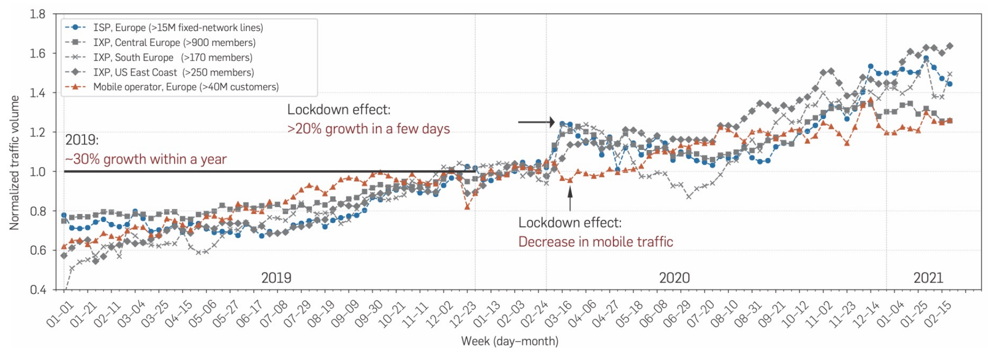
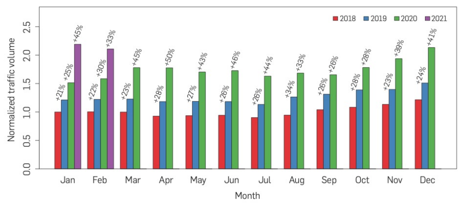


Figure 2. ISP monthly normalized downstream traffic change during the COVID-19 pandemic with percentage increase compared to the previous year.



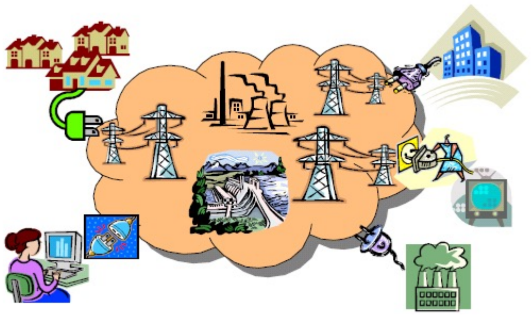
Some "old" and partial answers

- Utility computing
- Grid computing
- Autonomic computing
- Software as a Service (**SaaS**)
  - An "old" idea: application delivery on Internet
- … before Cloud computing (2006)
  - One step towards the scale problem solution

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."

John McCarthy (1927–2011) received the Turing Award in 1971 and was the inventor of Lisp and a pioneer of timesharing large computers. Clusters of commodity hardware and the spread of fast networking have helped make his vision of timeshared "utility computing" a reality.

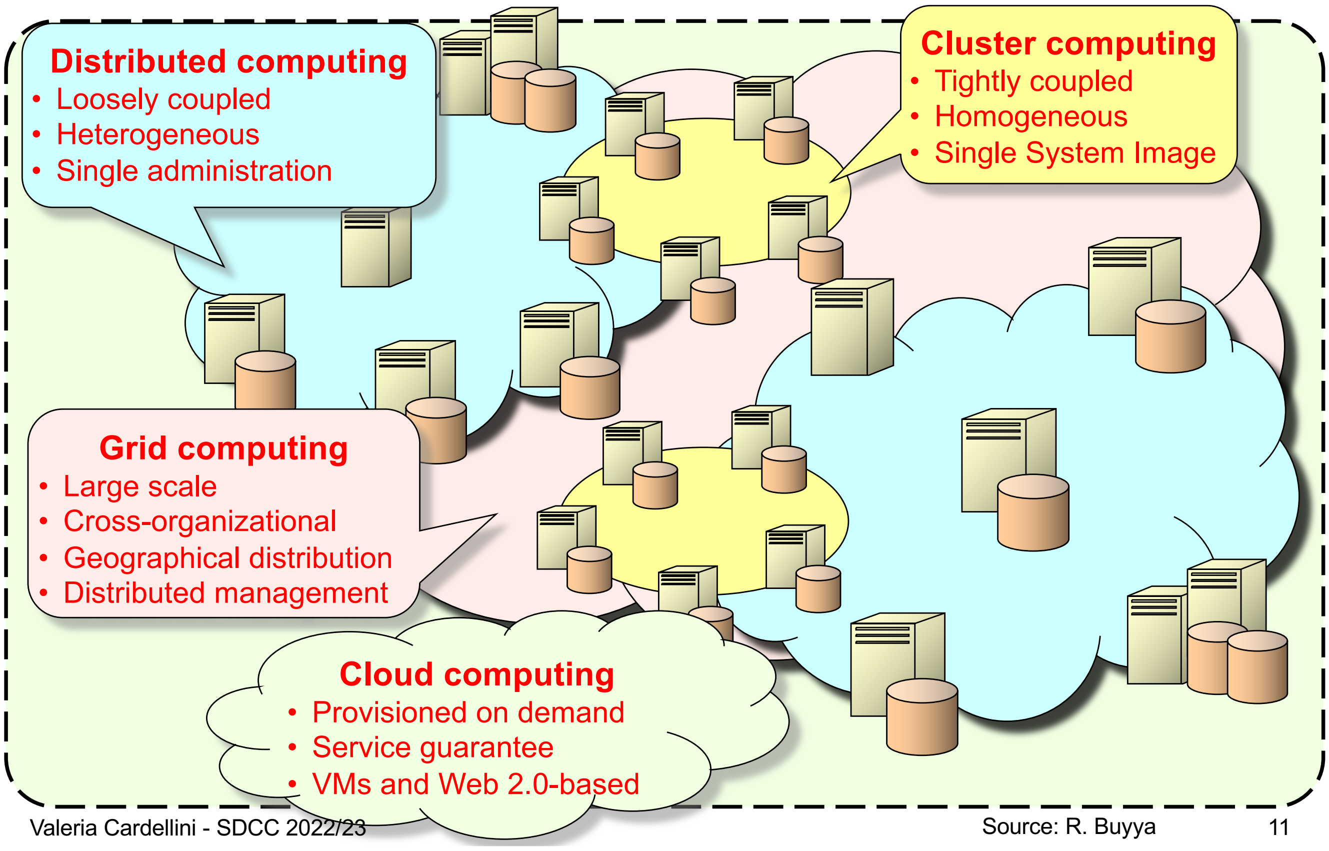
Utility computing: new idea?

- 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 re-definition of computer
  - 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

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

How do computing paradigms differ?

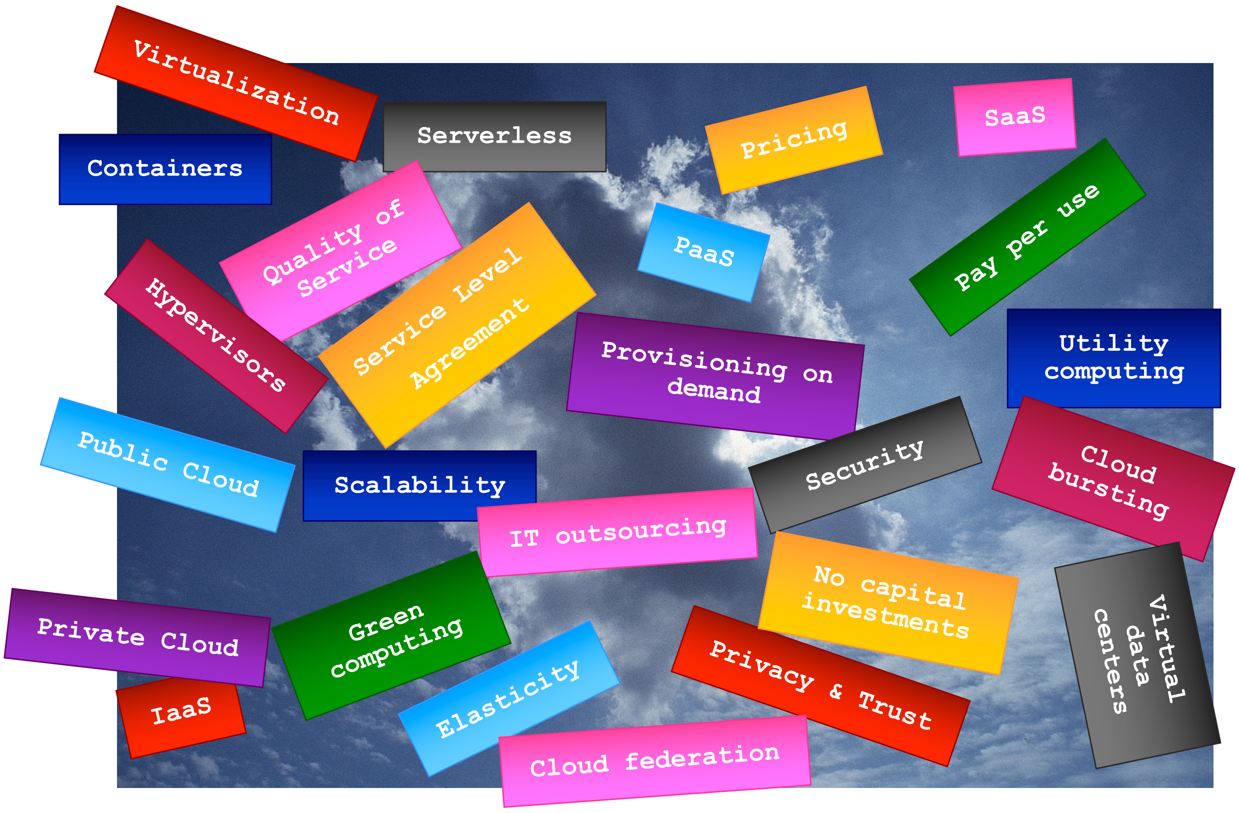


Source: R. Buyya

Cloud computing?

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

Many technologies, concepts and ideas



Cloud computing

Cloud

"Cloud" is the graphic symbol we use to represent Internet

computing

Computing includes:
- Computation
- Storage
- Coordination

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

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!)*

- [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*.

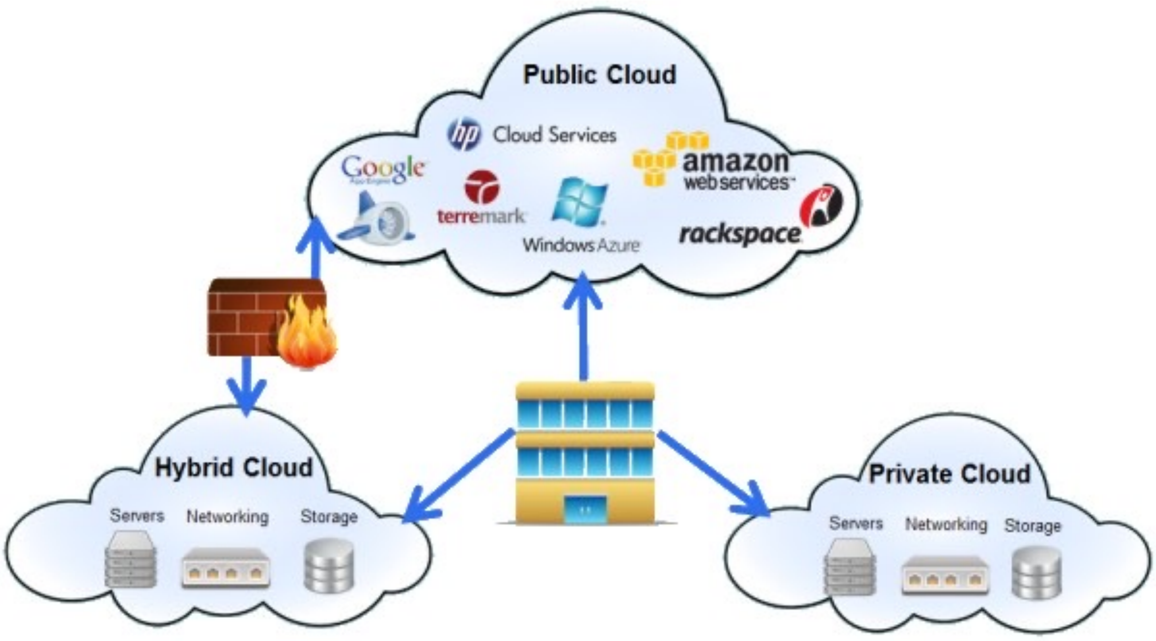
… that share some essential characteristics

- **On-demand self-service**
  - Cloud resources can be provisioned on-demand by the users, without requiring interactions with the cloud service provider
- 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: they can be rapidly scaled out/in based on demand

… that share some essential characteristics

- Resource **pooling**
  - 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
  - No large up-front acquisition cost
- **Measured** service
  - Usage of cloud resources is measured and user is charged based on some specific metric

Cloud deployment models



Deployment models: public

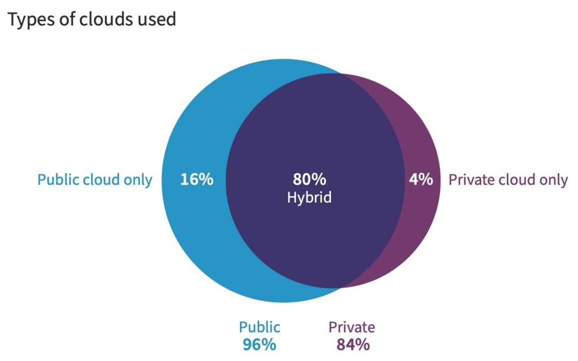
- Public cloud
  - Cloud infrastructure: provisioned for open use by the general public (multi-tenancy)
  - Owned, managed, and operated by a business, academic, or government organization, or some combination of them
  - Exists on premises of cloud provider
  - Services can be free or fee-based

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-premises of cloud provider
  - Pros: stronger security, customization
  - Cons: lower economic advantages, scalability more laborious

Deployment models: hybrid

- Hybrid cloud
  - 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
  - The most popular deployment model

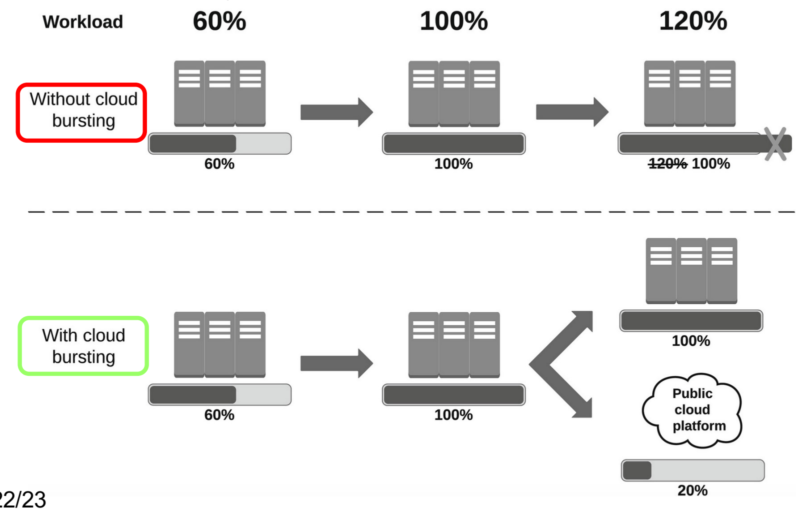


Deployment models: hybrid

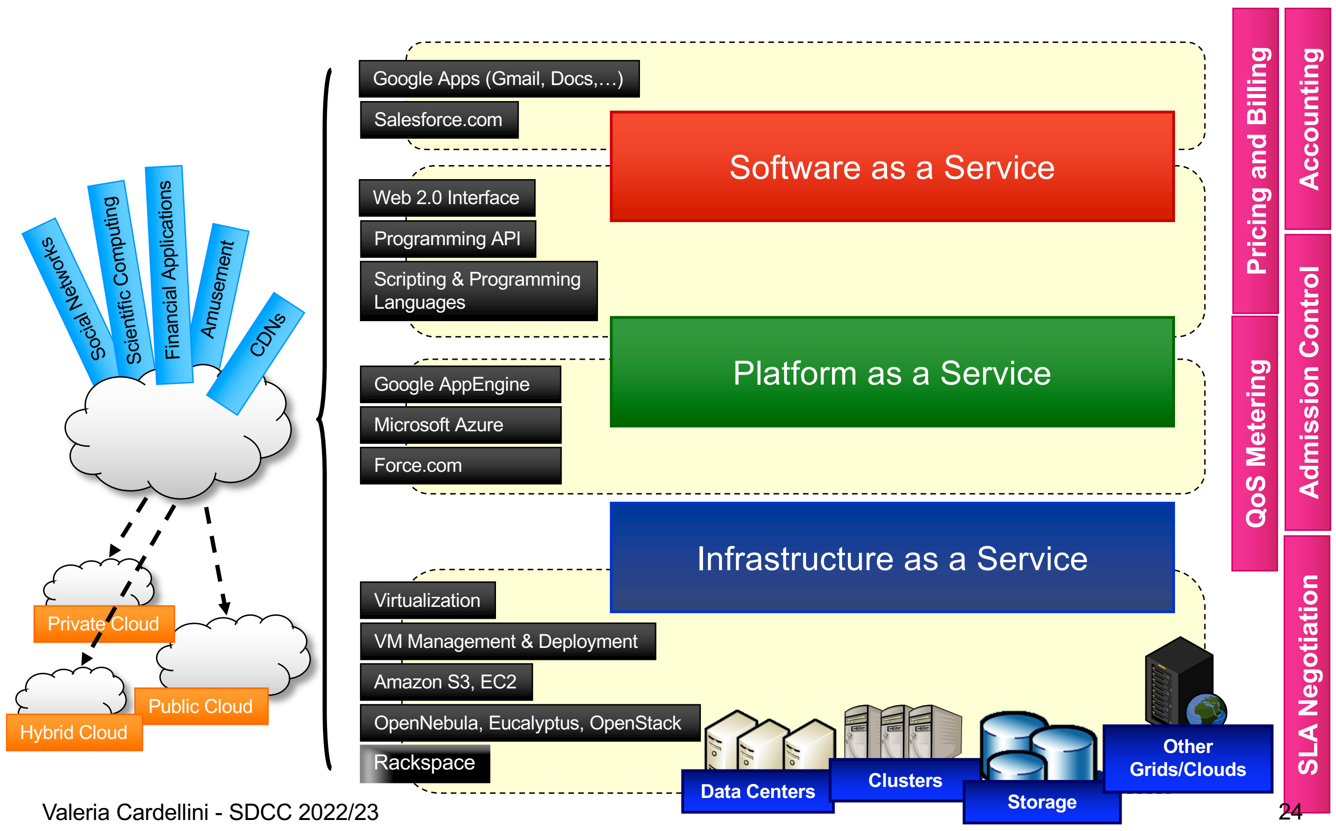
- Motivations
  - Balance resources and costs
  - Increase flexibility
    - Differentiate privacy, e.g., sensitive data is kept in private cloud
    - Differentiate services: apps are siloed on different clouds
    - Improve availability: public cloud for disaster recovery in case of unexpected outage
    - Improve geographic distribution: public cloud resource in a different geographic region
    - Improve performance: cloud bursting
  - Reduce vendor lock-in risks

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 app
  - Public cloud is used to manage load spikes unsustainable by private cloud only



Cloud stack: service models



Service models: IaaS

- Infrastructure as a Service (IaaS)
  - Compute, storage and network resources as services
  - Customer capability
    - To provision fundamental computing resources (processing, storage, networks) where consumers deploy and run arbitrary software (including OS and apps)
  - Customer control or management
    - No control over underlying cloud infrastructure
    - Control over OS, storage, deployed apps
    - Possibly limited control of selected networking components (e.g., host firewalls)

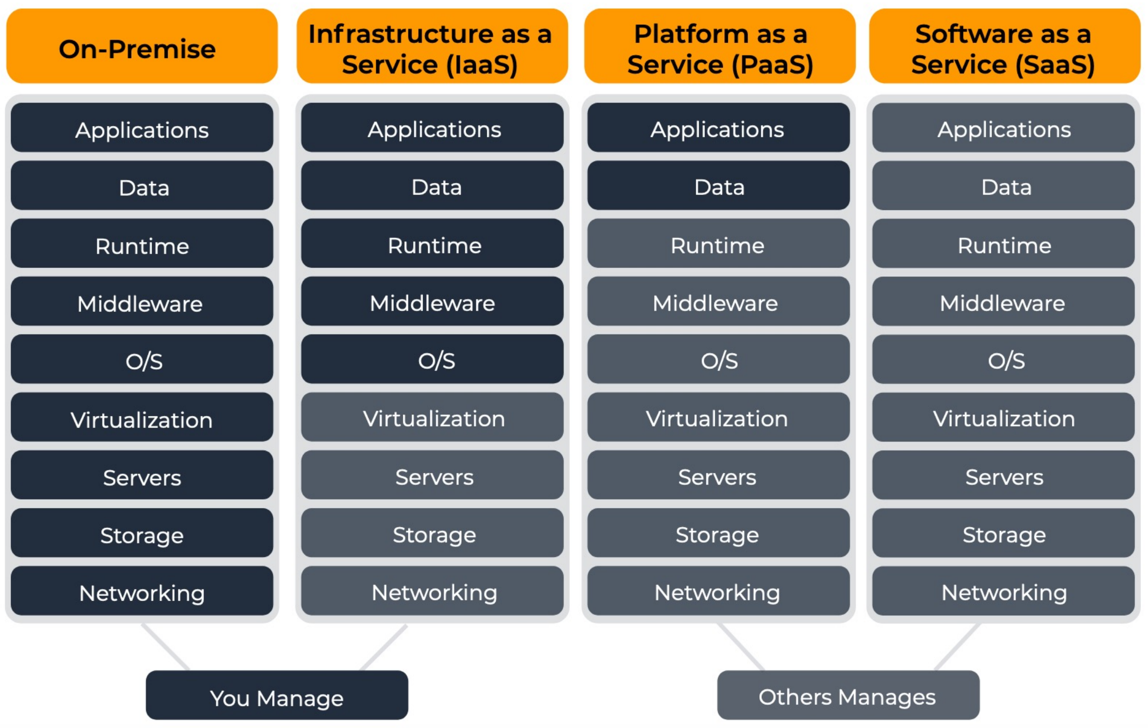
Service models: PaaS

- 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) apps realized using programming languages, application frameworks, development tools supported by PaaS provider
  – Customer control or management
    - No control over underlying cloud infrastructure (network, servers, OS, storage)
    - Control over deployed apps and possibly application hosting environment configurations

Service models: SaaS

- Software as a Service (SaaS)
  – 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
    - 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

- Alibaba Cloud: e.g., Elastic Compute Service
- Amazon Web Services: e.g., EC2, S3
- Aruba Cloud
- CloudSigma
- DigitalOcean
- Google Cloud Platform: e.g., Compute Engine, Cloud Storage
- IBM: e.g., Cloud Virtual Servers, Cloud Object Storage
- Microsoft Azure: e.g., Virtual Machines, Storage
- Flexera: e.g., Cloud Servers, 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 instance type and configure it
     - T-shirt sizes: various combinations of CPU, memory, storage, and networking capacity
  7. If not yet configured, select 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

PaaS examples

- Apprenda
- Platforms for Web apps
  – AWS Elastic Beanstalk, Google App Engine, Microsoft Azure App Service
- Kubernetes platforms
  – Google Kubernetes Engine, Amazon Elastic Kubernetes Service, Azure Kubernetes Service, Red Hat OpenShift, VMware Tanzu
- Oracle Cloud Platform
- Salesforce Platform
- Serverless computing/FaaS services
  – AWS Lambda, Google Cloud Functions, IBM Functions
- More examples, see https://paasfinder.org/vendors

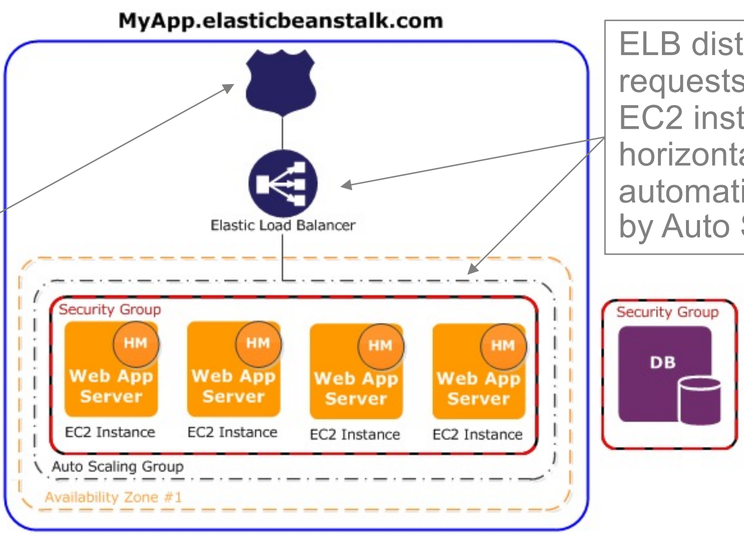
PaaS example: AWS Elastic Beanstalk

- 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 automatically provisioned by Elastic Beanstalk

Also Amazon Route 53 to map the app URL to the URL of the front-end load balancer; can also be used for geographic distribution among different app replicas



ELB distributes requests among the EC2 instances, whose horizontal scaling is automatically managed by Auto Scaling

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

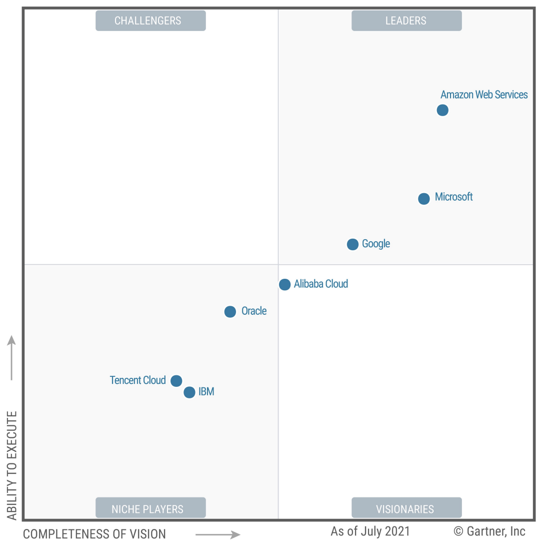
Comparing IaaS and PaaS

- Using IaaS provider
  - "Pure" virtualized resources: CPU, memory, storage, network bandwidth, load balancer, ...
  - OS 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

- Amazon Web Service (AWS), Microsoft Azure, Google Cloud Platform (GCP), IBM Cloud, Oracle, Alibaba
- Gartner's magic quadrant for IaaS and PaaS (2021)

Figure 1: Magic Quadrant for Cloud Infrastructure and Platform Services



Source: Gartner (July 2021)

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 providers offer a number of pricing 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
  – Users are charged a fixed amount per month for cloud resources
  – Pro and con: significant discount compared to on-demand pricing but upfront payment

Cloud pricing models

- Dedicated
  - Physical resources (e.g., physical server) rather than virtual ones fully dedicated to the user
  - Pro: dedicated hardware, maximum isolation
  - Con: higher cost
- Spot
  - Variable pricing for cloud resources driven by market demand (supply and demand)
  - 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

    → use for fault-tolerant workloads

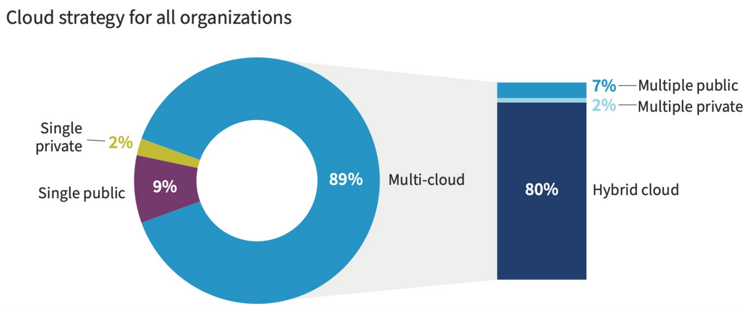
Which use of clouds? Some trends

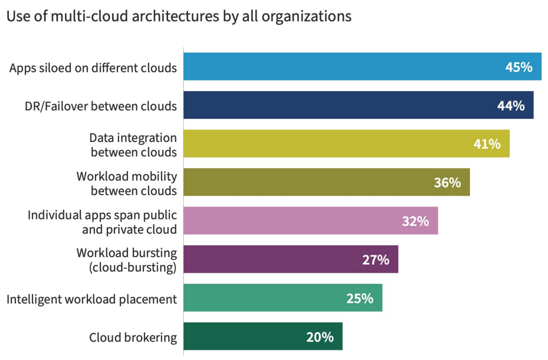
- Public cloud adoption continues to accelerate
  - ~50% of enterprise apps and data is in public cloud
  - Top-3 public cloud providers: Amazon, Google and Microsoft
  - Use of public cloud PaaS services (e.g., IoT, AI and ML services) is rising
  - COVID-19 increased cloud usage, though not as much as predicted
- Container technology (e.g., Docker , Kubernetes) is now mainstream
  - 96% of organizations are using or evaluating Kubernetes (CNCF 2021 report)
  - In 2020 use of containers in production has increased by 300% since 2016

Trends: Multi-cloud

- Most organizations choose a multi-cloud strategy: concurrent usage of multiple cloud environments
  - hybrid, 2 or more only private, 2 or more only public

- Which mix of clouds?
- Which use of multi-cloud?





Trends: Multi-cloud

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

Lab

Who use cloud?

- More than 90% of organizations
- Including the largest companies providing popular and successful Internet services that are:
  - Scalable and elastic
  - Affordable
  - Reliable
  - Secure
- Some examples: Airbnb, Baidu, Dropbox, Fabebook, 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
- 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 and network resources
- How? Forecast demand and scale-out(in) and/or scale-up(down) to manage workload variations
- Key benefit for capacity planning provided by Cloud computing: **elasticity**

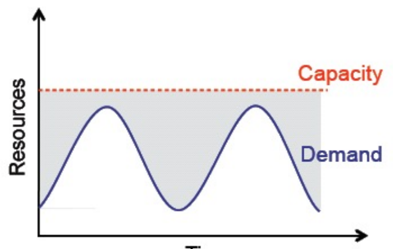
Traditional approach: over-provisioning

- Over-provisioning: resource provisioning by taking into account peak loads
- Leads to excess capacity and under-utilization
  - Server utilization in traditional data centers
    - Typically < 20%; rarely 30%

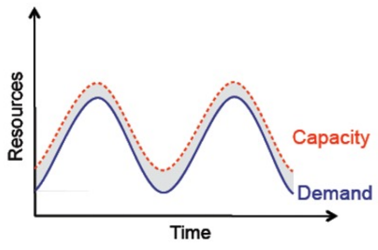
Higher costs than required

Increased energy requirement and consumption

We' d like to have



Static data center

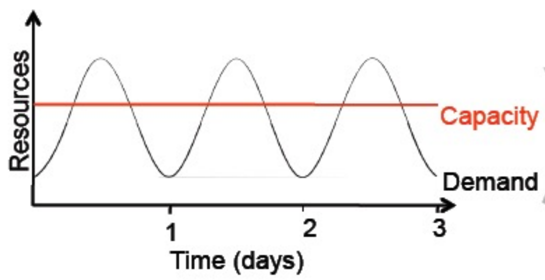


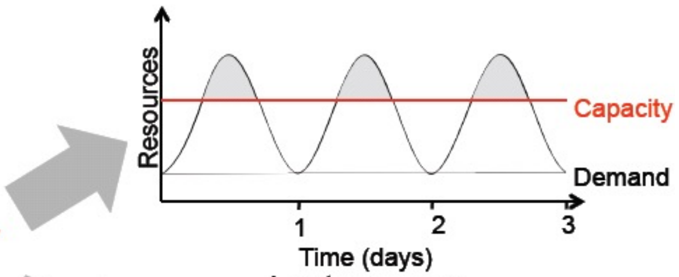
Data center in the cloud

▨ Unused resources

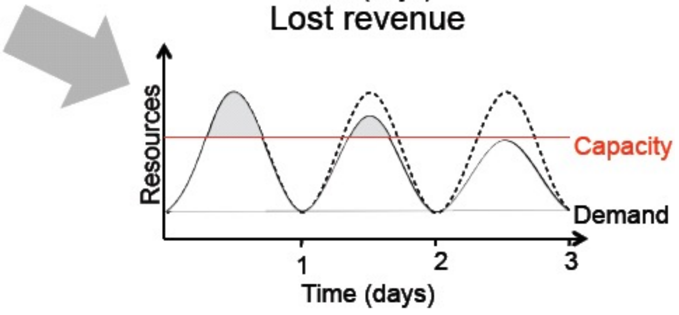
Traditional approach: under-provisioning

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





Lost revenue

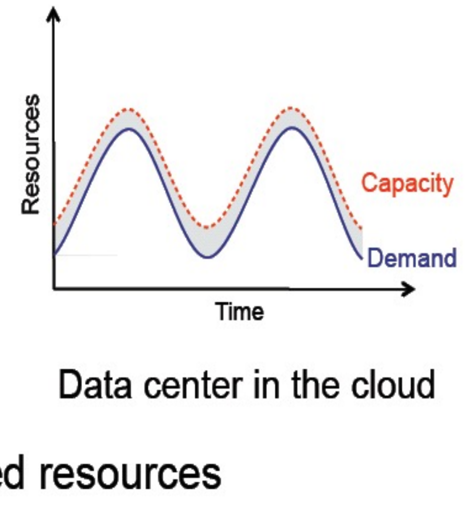


Lost users

Elasticity

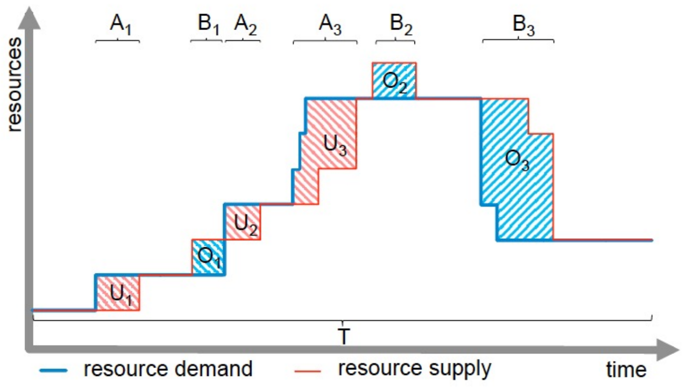
- Elasticity is the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible

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



Elasticity: metrics

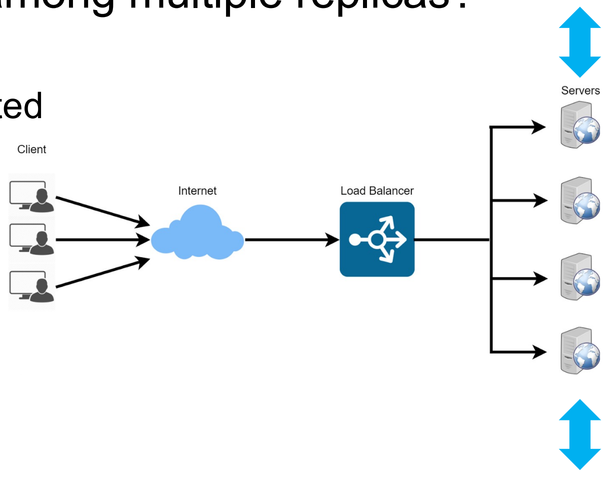
- Consider over-provisioning and under-provisioning



- 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*

Elasticity: architectural point of view

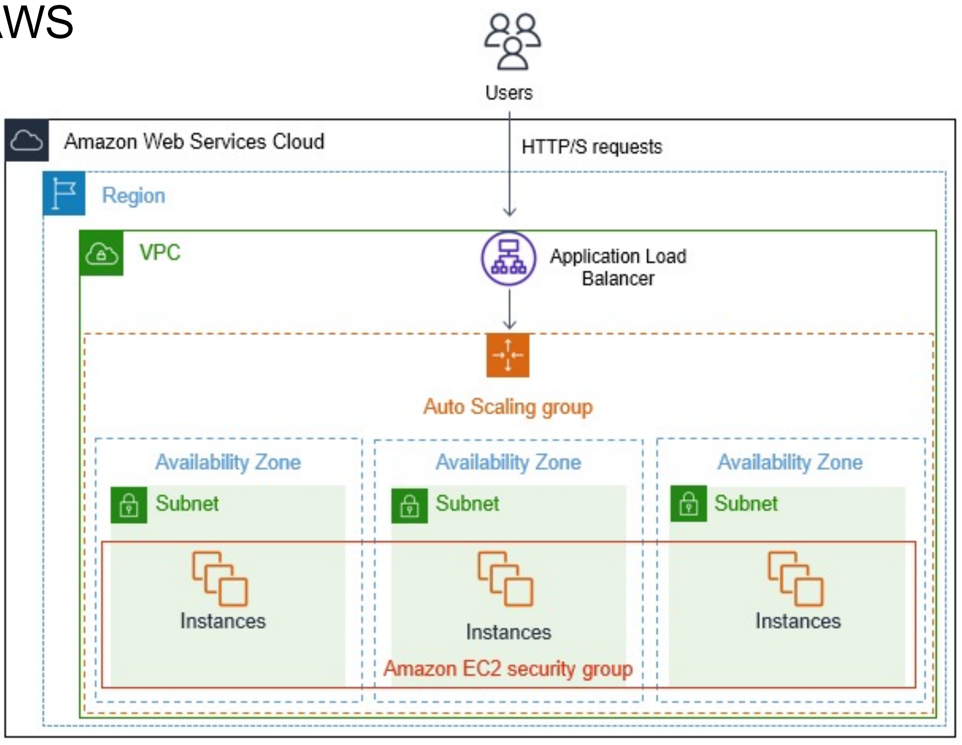
- How to distribute requests among multiple replicas? **Load balancer**
  - Can be centralized or distributed
- Load balancing goals:
  - Maximize resource utilization
  - Minimize response time
  - Maximize throughput



- Load balancing policies
  - Stateless: random, round robin
  - Server state-aware: weighted round robin, least loaded, power of two choices, …
  - Client state-aware: hashing, …
  - Client & server state-aware

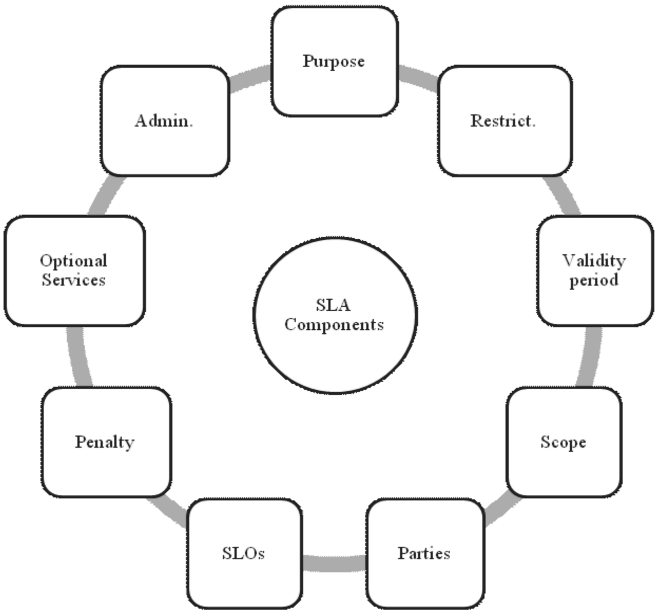
Elasticity: architectural point of view

- Example of load balancing and auto-scaling using AWS



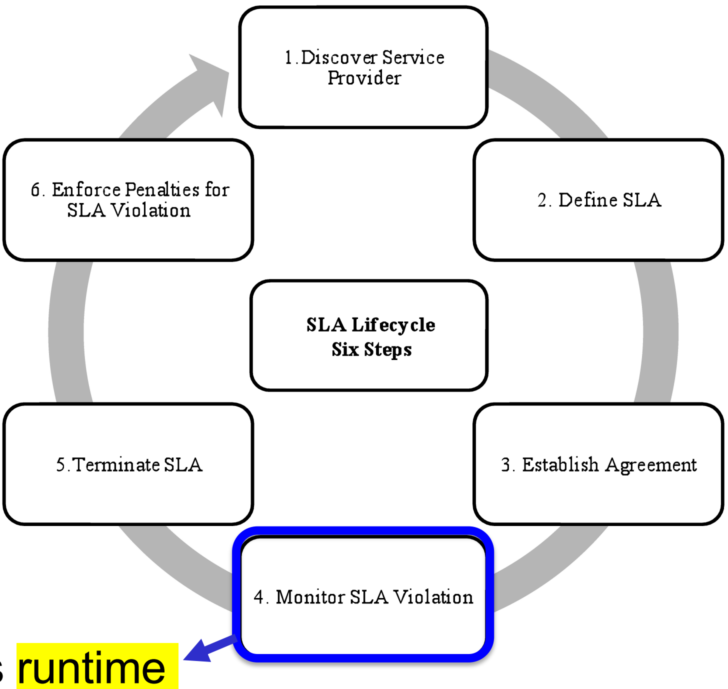
Service Level Agreement (SLA)

- 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 metric (e.g., mean response time <= 1 s)
- Includes penalty and/or compensation in case of SLA violation



Source: Sun Microsystems Internet Data Center Group

SLA life cycle



Requires runtime monitoring of service

SLI and SLO

- Service Level Indicator (SLI): something you can measure
  - E.g., response time, uptime
  - Uptime: the most common SLI for Cloud services https://uptime.is/

- Service Level Objective (SLO): a predicate over a set of SLIs
  - E.g., a fixed threshold on a single SLI

monthly uptime percentage for a VM is at least 99.99%

SLI SLO

Example: Amazon Compute SLA

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

For each individual Amazon EC2 instance ("Single EC2 Instance"), AWS will use commercially reasonable efforts to make the Single EC2 Instance available with an Instance-Level Uptime Percentage of at least 99.5%, in each case during any monthly billing cycle (the "Instance-Level SLA"). In the event any Single EC2 Instance does not meet the Instance-Level SLA, you will be eligible to receive a Service Credit

Instance-Level Uptime Percentage	Service Credit Percentage
Less than 99.5% 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%

Monthly Uptime Percentage 99.5% = 3h 39m 8s of downtime per month

Monthly Uptime Percentage 99% = 7h 18m 17s of downtime per month

- **Service Credits**

Service Credits are calculated as a percentage of the monthly bill… We will apply any Service Credits only against future payments… To receive a Service Credit, you must submit a claim…

Example: SLA for Amazon S3

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.

Issues with Cloud SLAs

- Another example: https://cloud.google.com/compute/sla
- Some issues on Cloud SLAs:
  - SLA jargon
  - Availability SLO
    - Using average over a period hides distinction between many short outages and a few long ones
  - Lack of guarantees in terms of service performance
    - No end-to-end response time, why?
  - 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

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

Cloud monitoring

- Monitoring goal
  - To keep track of the health of systems 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 and services
  - By Cloud provider: e.g., Amazon CloudWatch, Google Cloud Monitoring
  - By third party: e.g., Dynatrace, Prometheus

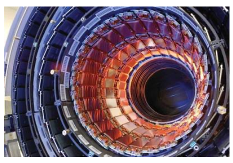
Some system-oriented monitoring metrics

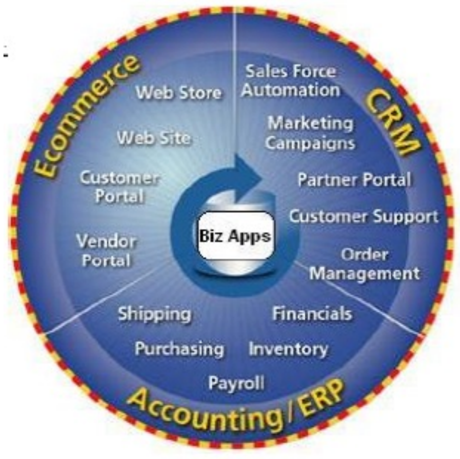
Type	Metrics
CPU	CPU utilization
Disk	Disk utilization, throughput (MB/sec for read/write, operations/sec)
Memory	Memory-Used, Memory-Free, Page-Cache
Interface	Throughput (Mbps incoming/outgoing)

Cloud applications

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

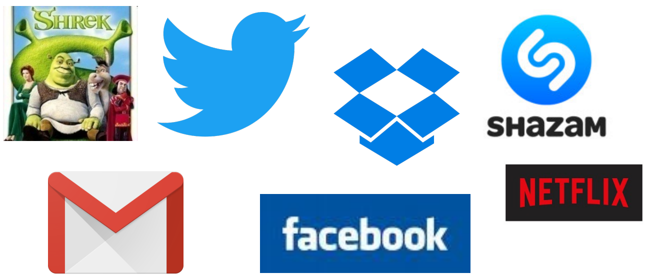
Healthcare, scientific and technical apps



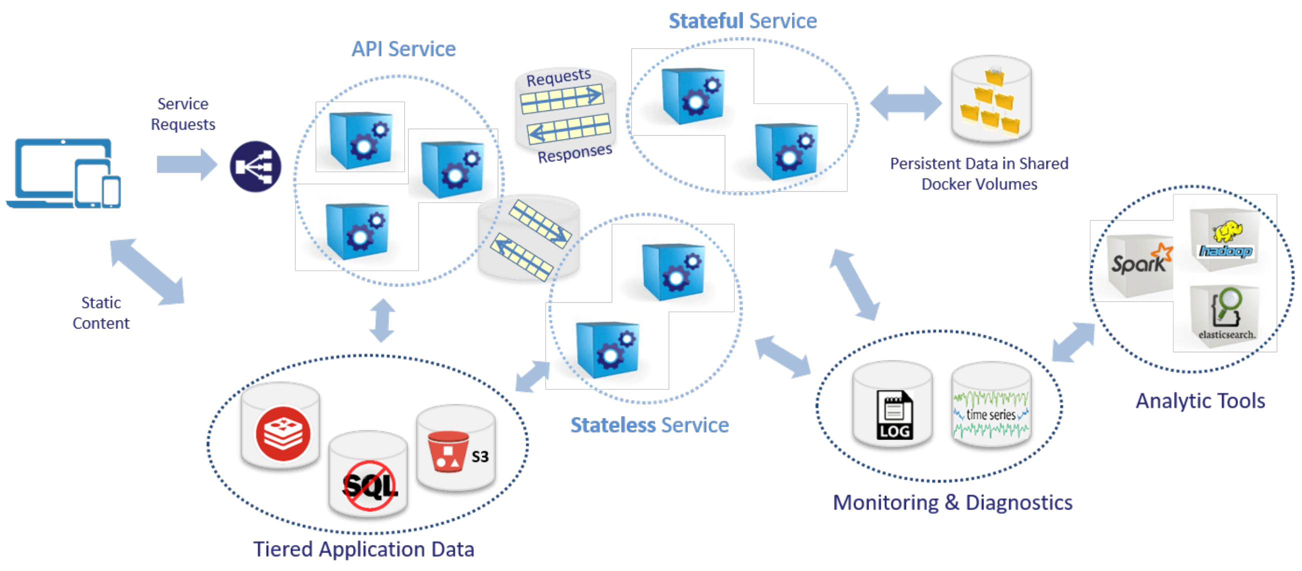


Business apps

Consumer/social apps

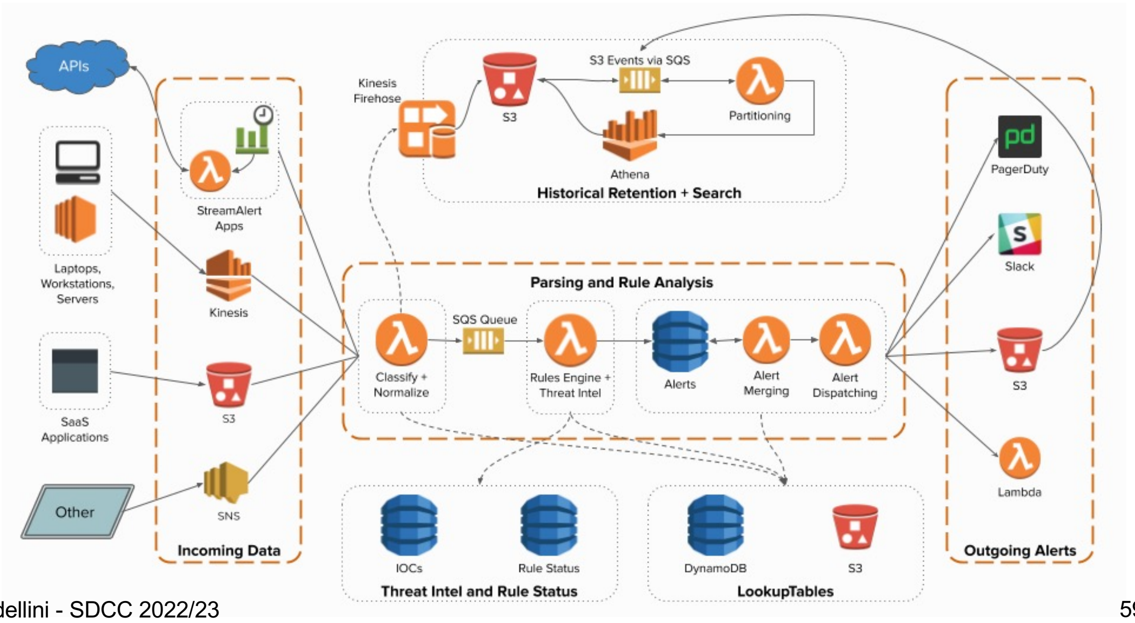


A high-level example of Cloud-native app



A real example of cloud-native app

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



Cloud application deployment

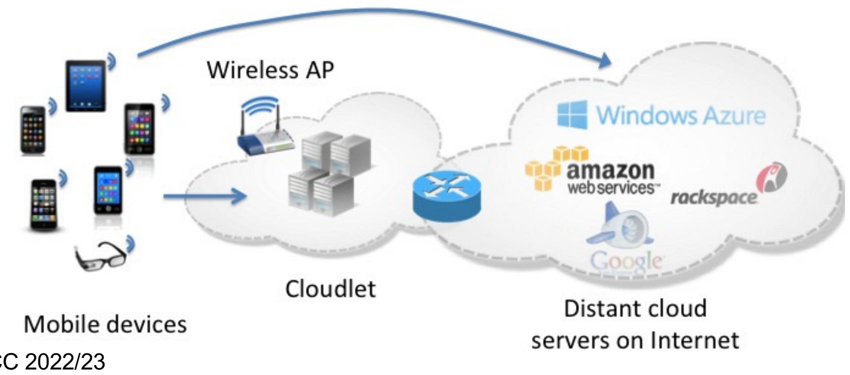
- Iterative process that involves:
  - Deployment design
    - Type and capacity of cloud resources in each tier (computing, memory and storage), interconnection, load balancing and replication strategies
  - Performance evaluation
    - Verify whether app meets performance requirements
    - Involve monitoring app workload, performance parameters (e.g., response time, throughput) and resource utilization (CPU, memory, disk, I/O, etc.)
  - Deployment refinement
    - 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

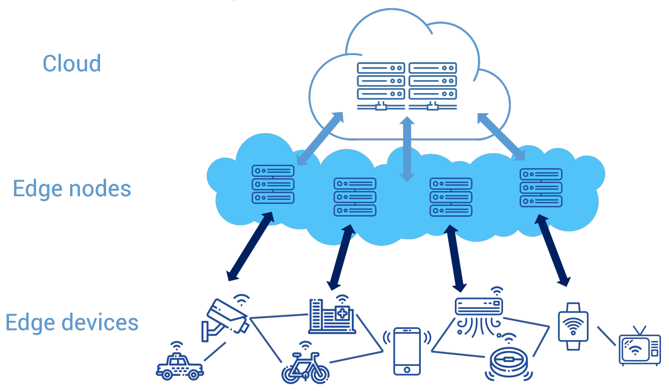
Next generation computing scenario: Fog and Edge computing

- How to handle latency-sensitive and location-aware apps (e.g., patient monitoring, real-time manufacturing, self-driving cars, flocks of drones, cognitive assistance) that use large data volumes originated from mobile and IoT devices?
  - Cloud-only solutions can be impractical due to latency
- Idea: **offload** part of computation and storage to nearby resources located at the network edges



Edge computing

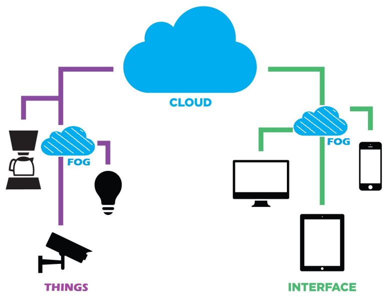
- Brings computation and storage at the proximity of data sources (e.g., IoT devices), as close as one hop to them
  - Edge nodes include routers, switches, sensors and actuators



- EU commission vision by 2030: 10,000 climate-neutral highly secure edge nodes

Fog computing

- Similarly to edge computing, fog computing is a layered model that pushes computation and storage from cloud servers down, closer to data sources
  - A fog node can be seen as a *micro-cloud* (or cloudlet), located near edge layer and IoT devices
  - Fog nodes can be organized in a multi-level hierarchy



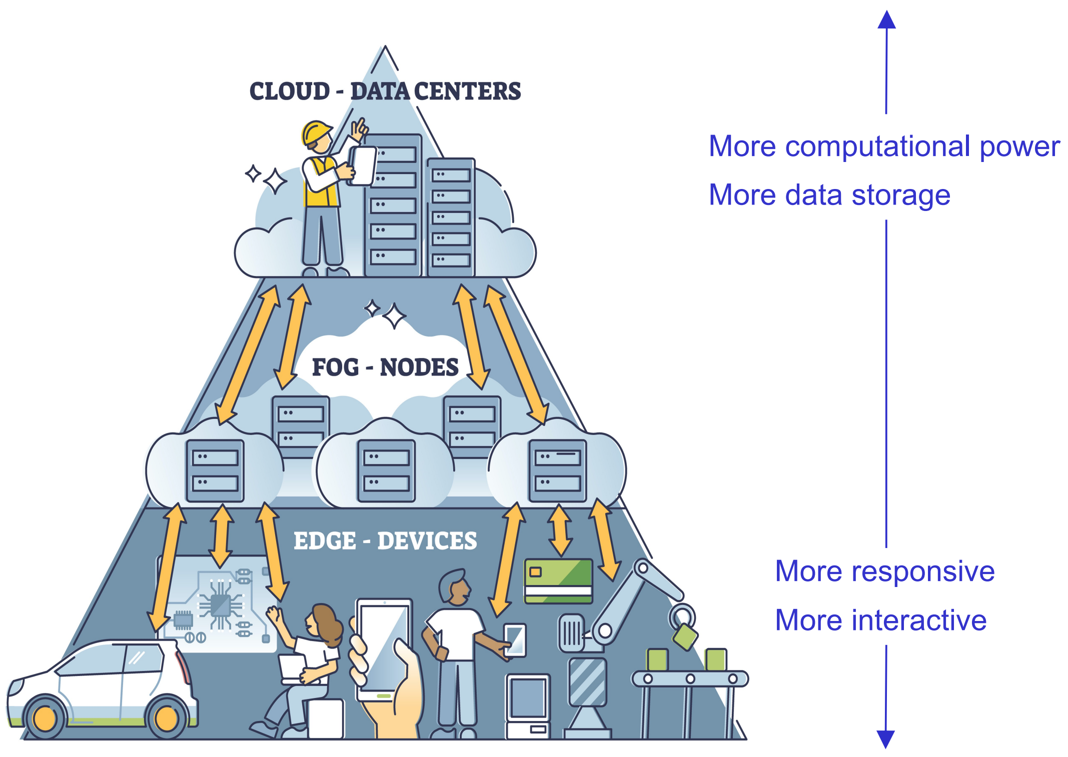
- Some definition
  - "Fog computing is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud computing data centers, typically, but not exclusively located at the edge of network." (Bonomi et al., 2012)
  - "A horizontal, system-level 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?

- Similar distributed computing paradigms at the network edge
  - Both have their roots in content delivery networks and peer-to-peer trends of the late 1990s
  - Both have a decentralized IT architecture
  - Both are beneficial for applications which require low latency (i.e., few ms) and real-time analysis
- Any difference?
  - For someone no: they are interchangeable

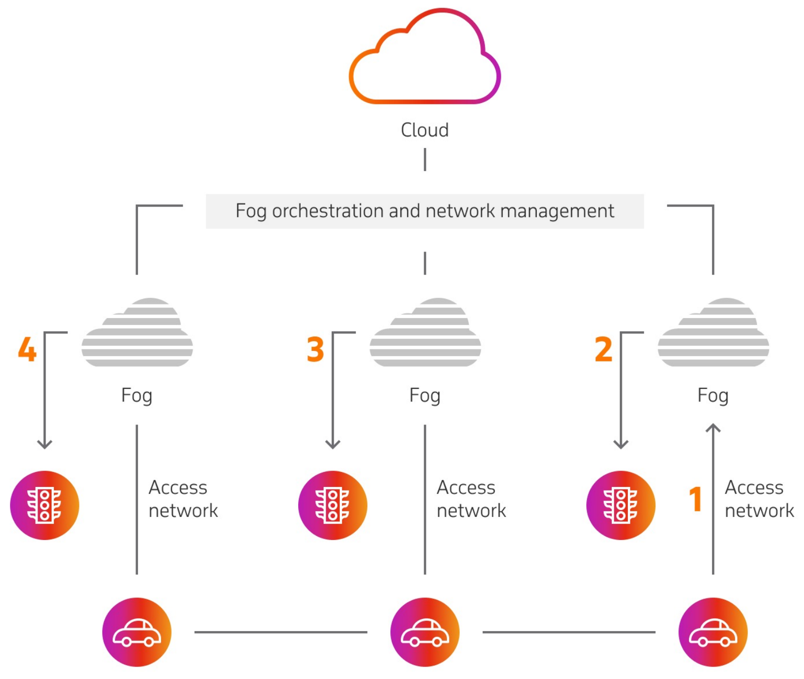
    "Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services." Edge computing: Vision and challenges
  - For someone yes: Fog is more tightly integrated with Cloud and its deployment size can be larger than edge
  - Fog is a layer in between Edge and Cloud

Compute continuum: from edge to cloud



Putting all together: example of Cloud continuum app

- Road traffic regulation



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
- Rapid business innovation
- Increase in productivity

Environment benefits

- Lower environmental impact due to ICT

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
  - 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, 2022)

    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, 2022)

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

Issues for Cloud customers

- Communication latency
  - ms latency from users and devices to cloud providers
  - Real-time and low-latency apps?
  - Compute continuum is the answer
- 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

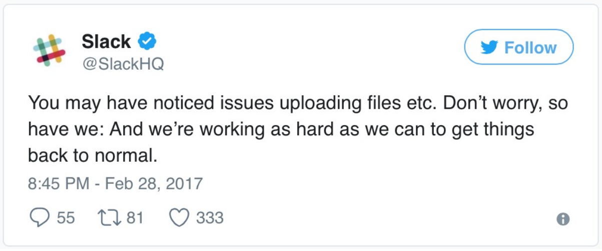
- Poor support for SLA negotiation and management
  - Lack of SLAs based on end-user experience
  - Cloud customer often in charge of SLO monitoring

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

Issues for Cloud customers

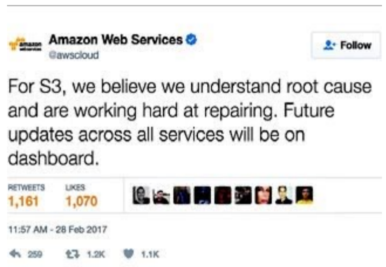
- Cloud infrastructure outage, some examples:
  - February 28, 2017: AWS S3 went down for 4 hours, taking Slack and several media sites down with it
    - Due to human error http://amzn.to/2fJhcfg


Slack @SlackHQ Follow

You may have noticed issues uploading files etc. Don't worry, so have we: And we're working as hard as we can to get things back to normal.

8:45 PM - Feb 28, 2017

55 81 333


Amazon Web Services @awscloud Follow

For S3, we believe we understand root cause and are working hard at repairing. Future updates across all services will be on dashboard.

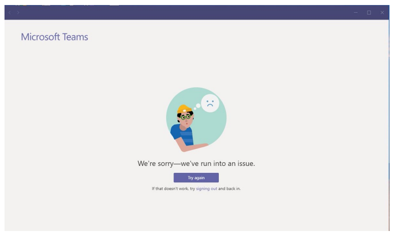
RETWEETS 1,161 LIKES 1,070

11:57 AM - 28 Feb 2017

259 1.2K 1.1K

  - 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


Microsoft Teams

We're sorry—we've run into an issue.

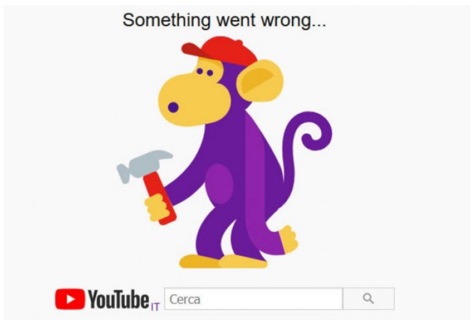
Try again

If that doesn't work, try signing out and back in.

Issues for cloud customers: cloud outages

  - 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


Something went wrong...

  - Cloud outages in 2022: see CRN slideshows

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