Cloud Computing – Architecture and Services

Big Data and Cloud Computing (CC4093)

Eduardo R. B. Marques, DCC/FCUP

From last class ...

NIST definition of Cloud Computing as:

> ***"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."***

It also mentions:

> *This cloud model is composed of five* ***essential characteristics****, three* ***service models****, and* ***four deployment models****.*

Let us cover the general aspects regarding cloud computing architecture and services.

Essential characteristics

On-demand self-service: A consumer can provision computing resources without human intervention.

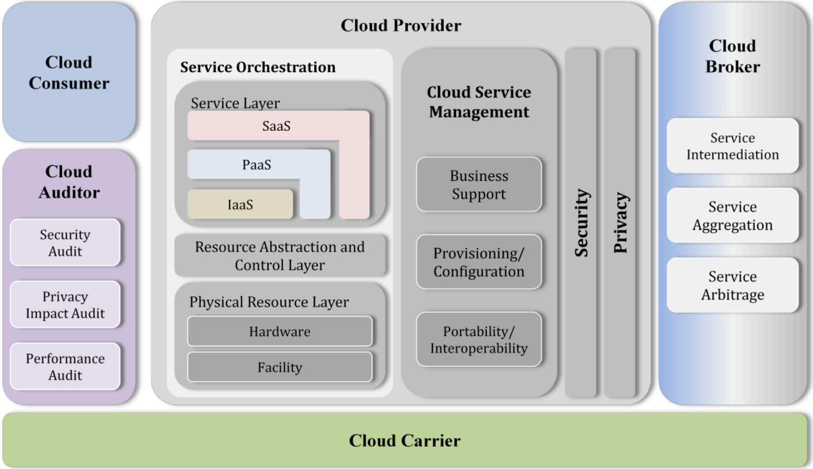
Broad network access: The cloud services are available over the network through standard means (e.g. Internet connection, thin clients such as smartphones or PCs).

Resource pooling: Resources are pooled by the cloud provider to serve multiple consumers using a multi-tenant model.

Rapid elasticity: The amount of computing resources can be adjusted/scaled dynamically at the consumer's request or automatically in some configurable manner by the consumer.

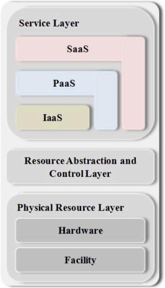
Measured service: Resource usage can be tracked and measured precisely by both the cloud provider and consumer, for instance to allow for a clear billing analysis.

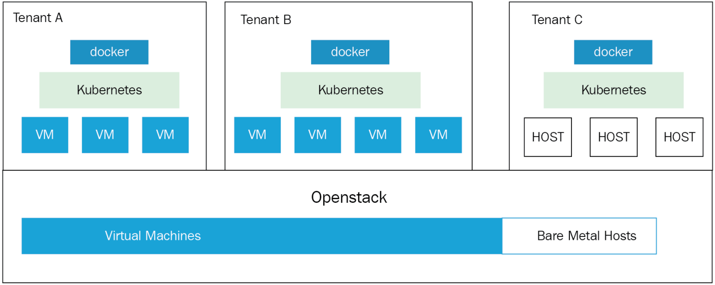
NIST Cloud Computing Reference Architecture



The NIST Cloud Computing Reference Architecture provides a logical view of the organization, concerns and involved parties in a cloud. **Our focus** will be the use of cloud services (Cloud Consumer → Cloud Provider).

Service orchestration





Resource abstraction and control layer: mediates the use of physical resources by services through techniques such as virtualization, dynamic provisioning, access control, etc. It usually exposes **virtual resources** on top of the **physical resource layer**, that manages the the actual pool of concrete **physical resources** available for processing, networking, storage, etc. In some cases (for higher quality of service) there is a direct mapping between virtual and (a priori reserved) physical (aka “bare metal”) resources.

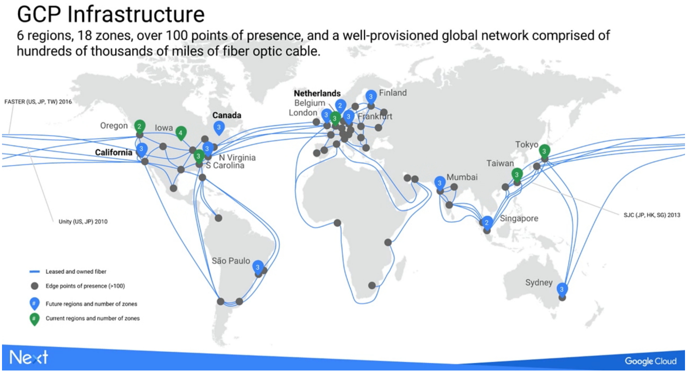
Virtualization

A **virtual resource** provides the illusion of a real resource (computer, storage, network) through diverse context-specific techniques.

General strategies include:

- multiplexing the same physical resource such that it actually provisions several virtual resources, e.g. a physical computer may host several VMs;
- provisioning a virtual resource by several different physical resources at different times or even at the same time, e.g. storage service using several physical disks;
- dynamic provision of virtual resources by physical resources, e.g. through reconfiguration, replication, auto-scaling, migration, etc

Regions & availability zones

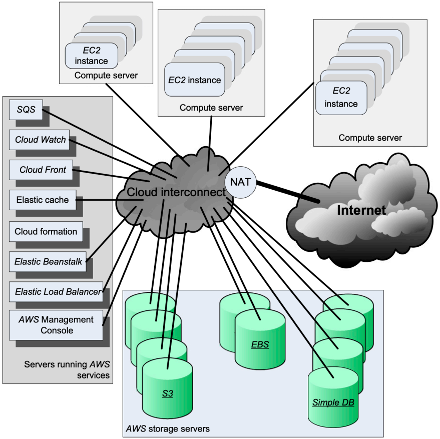


GCP Infrastructure

6 regions, 18 zones, over 100 points of presence, and a well-provisioned global network comprised of hundreds of thousands of miles of fiber optic cable.

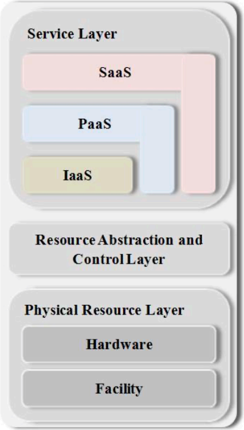
Cloud infrastructure is geographically distributed in **regions**, each with a set of **availability zones** (locations of data centers). The set of service and their cost may be different from zone to zone. Network access speed by a customer depends the geographical location; it is usually faster for data centers that are nearby geographically.

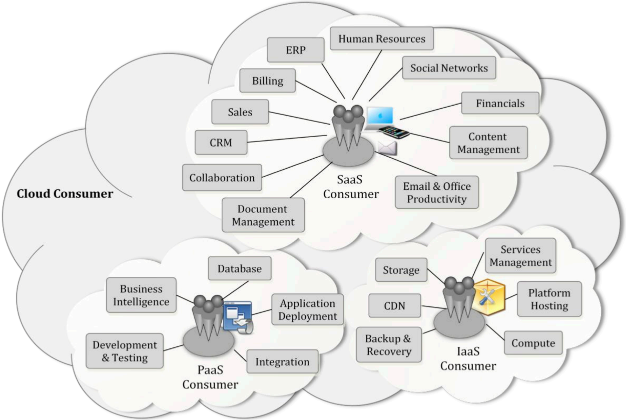
Regions & availability zones (cont.)



Overview of the organization of an AWS availability zone (image from Marinescu, “Cloud Computing - Theory and Practice”).

Service models





Images from: NIST Cloud Computing Reference Architecture.

Service models - IaaS

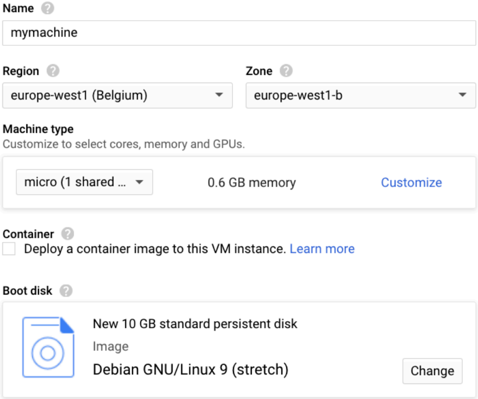
Infrastructure-as-a-Service (IaaS)

Customers get actual infrastructure in the form of computing resources (e.g., for processing, storage, or networking purposes).

The service provider hosts, maintains, and updates the backend infrastructure, such as compute, storage, networking, and virtualization.

Customers manage everything else including the operating system, middleware, data, and applications.

IaaS example - Google Compute Engine



Name

mymachine

Region

europe-west1 (Belgium)

Zone

europe-west1-b

Machine type

Customize to select cores, memory and GPUs.

micro (1 shared ...

0.6 GB memory

Container

□ Deploy a container image to this VM instance. Learn more

Boot disk

New 10 GB standard persistent disk

Image

Debian GNU/Linux 9 (stretch)

Main parameters: **Name**, **Region and Zone** (defines the hosting data center), **Machine Type** (base CPU + memory configuration), and **Boot disk** (initial OS image + disk size).

Servide models - PaaS

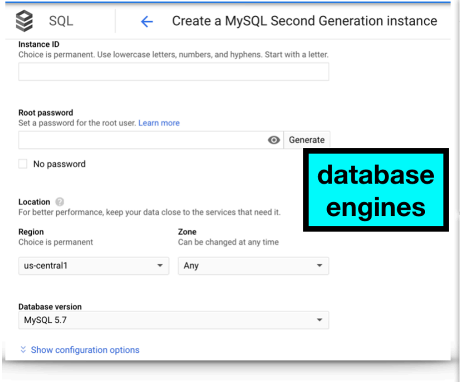
Platform-as-a-service (PaaS)

Customers get a platform that supports the development and/or deployment of cloud applications (e.g., database engines, data workflows, programming environments).

As with IaaS models, in PaaS models, the service provider delivers and manages the backend infrastructure. Additionally, PaaS models provide all the software features and tools needed for application development.

The customer manages your applications and data but need not be concerned with managing or maintaining the software development platform.

PaaS examples



SQL ← Create a MySQL Second Generation instance

Instance ID
Choice is permanent. Use lowercase letters, numbers, and hyphens. Start with a letter.

Root password
Set a password for the root user. Learn more

□ No password

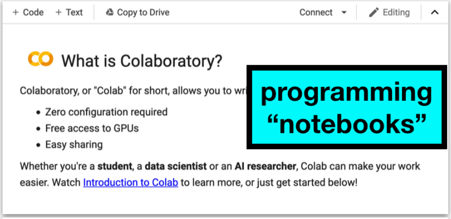
Location
For better performance, keep your data close to the services that need it.

Region
Choice is permanent
us-central1

Zone
Can be changed at any time
Any

Database version
MySQL 5.7

database engines



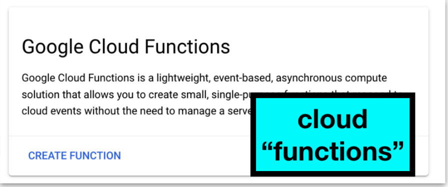
What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to wri

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

programming "notebooks"



Google Cloud Functions

Google Cloud Functions is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-p cloud events without the need to manage a serve

cloud "functions"

Services are configured for the development/deployment of cloud applications. Little infrastructure configuration is required: the infrastructure is deployed automatically according to high-level parameterisation.

Service models - SaaS

Software-as-a-Service (SaaS)

Customers get access to applications running in the cloud (e.g., Gmail, Google docs). The cloud is transparent to the user.

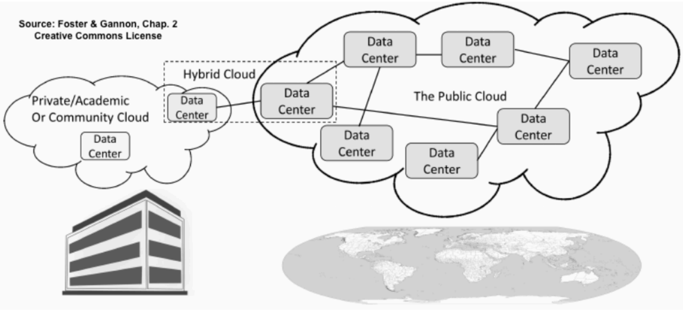
The service provides manages everything, including the application and data.

The customer simply connects to the service through the Internet.

SaaS examples

Software deployed on the cloud that is available to general users.

Deployment models


Source: Foster & Gannon, Chap. 2
Creative Commons License
Hybrid Cloud
Private/Academic Or Community Cloud
Data Center
The Public Cloud

Public cloud : open for use to the general public by a cloud provider. The cloud provider may be a business operator (e.g. Google) or a public institution.

Private / community clouds : used exclusively by of a single organization (private) or a group of organizations (community).

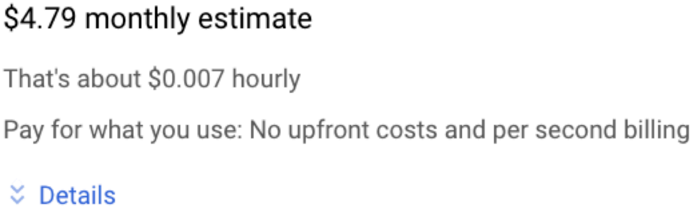
Hybrid cloud results from the combination of two or more clouds that exist separately but are interoperable in some way.

Public vs. private/community clouds

Important factors:

- Cost
  - public cloud services tend to be more expensive taken individually
  - these costs may be offset by the maintenance required by private clouds
  - public clouds also have vast amounts of pooled resources that can be easily adjusted (augmented or reduced) to customer needs
- Privacy / security
  - use of public clouds may increase surface for security and sensitive data breaches
  - enforcing security and privacy is non-trivial for small organisations / requires specialized skills

Measured service



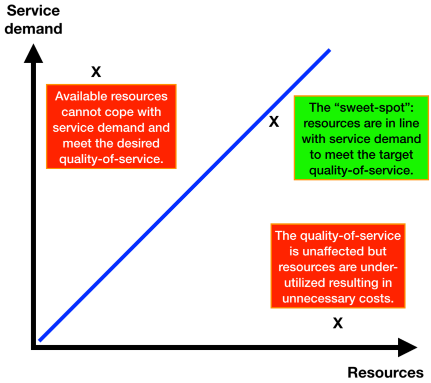
$4.79 monthly estimate

That's about $0.007 hourly

Pay for what you use: No upfront costs and per second billing

- **Basic principle**: pay for what you use.
- Examples resources and a few billing factors
  - VMs: computing time, capabilities (e.g., CPUs, RAM, disk size, GPU)
  - storage: amount of storage, traffic, availability settings (e.g. check the pricing tables for Google Cloud Storage)

Elasticity - service demand vs resource usage


Service demand
x
Available resources cannot cope with service demand and meet the desired quality-of-service.
x
The “sweet-spot”: resources are in line with service demand to meet the target quality-of-service.
The quality-of-service is unaffected but resources are under-utilized resulting in unnecessary costs.
x
Resources

Quality-of-Service (QoS): some kind of metric related to the functionality (e.g., requests-per-second, response time) or resources (e.g., computing power, network bandwidth, storage size) at stake.

Elasticity: The amount of computing resources can be adjusted/scaled dynamically at the consumer’s request or, through **auto-scaling**, automatically in some configurable manner.