BIG DATA [MODULE 2]

Cloud computing

whoami

Matteo Francia

- Email: m.francia@unibo.it
- Research fellow @ UniBO

Research topics

- Big data / database
- Geo-spatial analytics

Thesis proposals

- https://big.csr.unibo.it/teaching/

BUSINESS
INTELLIGENCE
GROUP

https://big.csr.unibo.it/

Exam

We will see how (big) data pipelines can be done in the cloud

- We will cover some "theory" as well as get our hands dirty
- This module is part of the oral examination
- (If agreed) The project can be extended to the cloud to get more points
  - E.g., rethink a part of the project on cloud

Roadmap

Introduction to cloud and cloud service providers

Cloud services (in AWS)

Hands-on cloud services

- Orchestrating (small) data pipelines
- Migrating a cluster to the cloud
- Migration pricing

Set up

You (will) have an AWS Educate account

- A coupon of 150$ to test AWS cloud services

The content of these slides refers to this repo

- https://github.com/w4bo/bigdata-aws/
- Options
  - Work on your pc: check the `README.md` to install the necessary software tools
    - Mainly: git, IntelliJ IDEA (Community Edition), docker, AWS CLI, AWS SAM CLI
  - Download a pre-configured (VMware) VM with the installed software

https://aws.amazon.com/education/awseducate/

Set up

Download a pre-configured (VMware) VM with the installed software

1. Connect to a PC-lab using Guacamole (click here https://csi-rlab.campusfc.unibo.it/)
    a) Enter your credentials
    b) Choose `LabCEZ`
    c) Click on the proper lab
2. Once logged in, download the VM from http://big.csr.unibo.it/downloads/ubuntu-64-bit.zip
3. Unzip the content and launch `VMware Workstation Pro` (from the desktop)
4. `File` > `Open...` > `path/to/the/unzipped/vm/*.vmx` file
5. Launch the VM (user: `bigdata`, pwd: `bigdata`)

So far

You have practiced with **on-premises** solutions

- You were given a working hardware cluster
- ... to deploy software applications on Hadoop-based stack

Let us guess, how would you start from scratch?

- How would you do that?
- How much time would it take?

So far

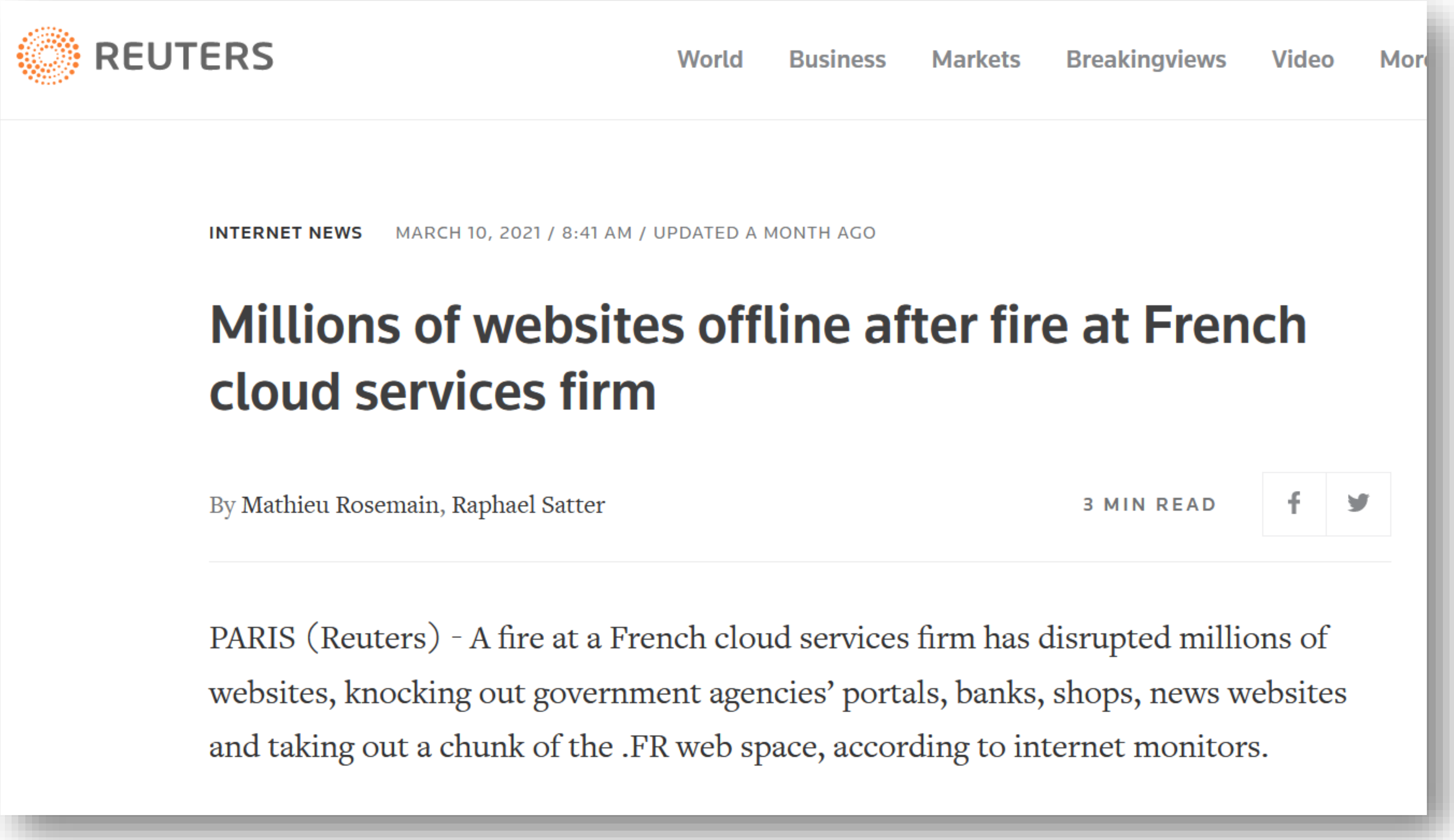
No easy answers

Big-data (distributed) architectures require a lot of skills

- **Installation**: how do I set up a new machine?
- **Networking**: how do I cable dozens of machines?
- **Management**: how do I replace a broken disk?
- **Upgrade**: how do I extend the cluster with new services/machines?
- (energy and cooling, software licenses, insurance...)

https://aws.amazon.com/compliance/data-center/data-centers/

So far



INTERNET NEWS MARCH 10, 2021 / 8:41 AM / UPDATED A MONTH AGO

Millions of websites offline after fire at French cloud services firm

By Mathieu Rosemain, Raphael Satter

3 MIN READ

PARIS (Reuters) - A fire at a French cloud services firm has disrupted millions of websites, knocking out government agencies' portals, banks, shops, news websites and taking out a chunk of the .FR web space, according to internet monitors.

https://www.reuters.com/article/us-france-ovh-fire-idUSKBN2B20NU

So far

Technological perspective

- How do we configure a distributed environment?
  - How do we set up independent services?
  - How do we integrate such services?
  - How do control resource accesses (e.g., storage)?
- How do we orchestrate data flows?
- It depends on your (team) skills (not only software engineering)

Business perspective

- No free lunch, each choice has cost/benefit
- How much time does it take to master a technology?
- How many people do I need?

So far

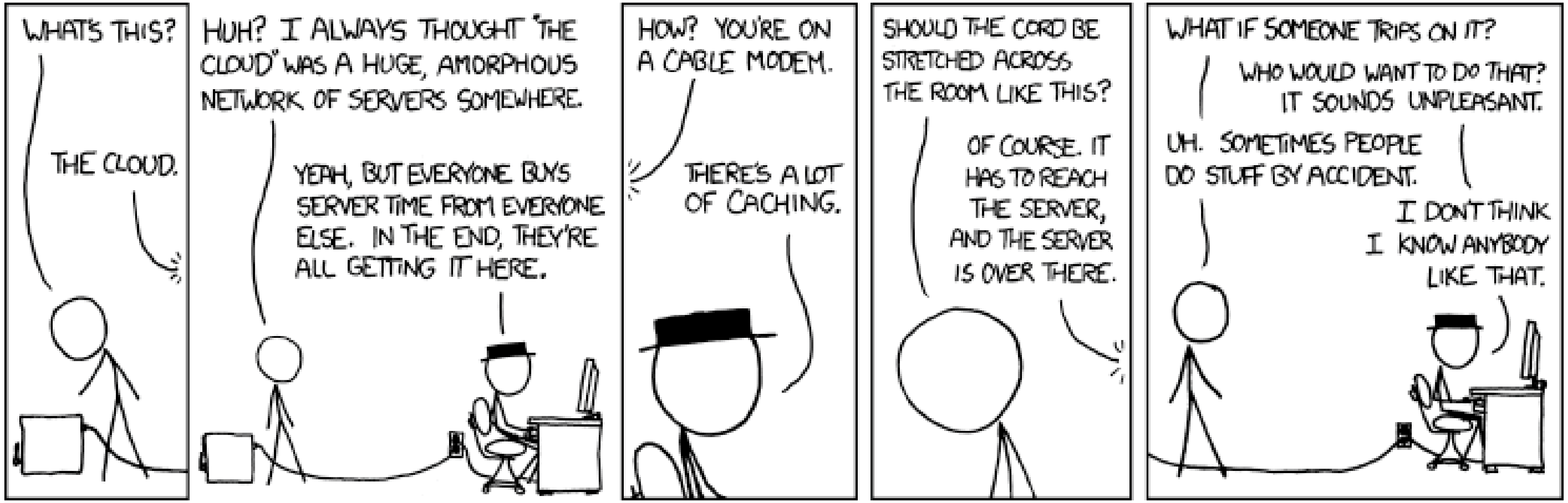
Can we afford to spend resources on tasks that are not mission oriented?

- Mission: a statement used by a company to explain its purpose(s)

How can I build a working application/data platform?

BIG DATA

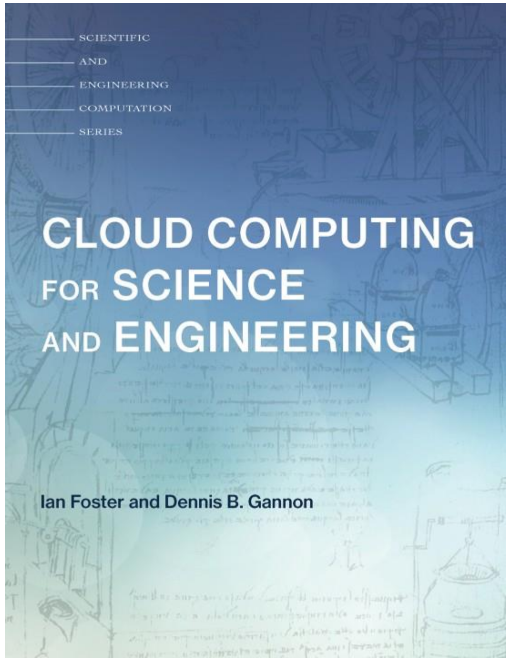
Why going cloud?


WHAT'S THIS?
THE CLOUD.
HUH? I ALWAYS THOUGHT "THE CLOUD" WAS A HUGE, AMORPHOUS NETWORK OF SERVERS SOMEWHERE.
YEAH, BUT EVERYONE BUYS SERVER TIME FROM EVERYONE ELSE. IN THE END, THEY'RE ALL GETTING IT HERE.
HOW? YOU'RE ON A CABLE MODEM.
THERE'S A LOT OF CACHING.
SHOULD THE CORD BE STRETCHED ACROSS THE ROOM LIKE THIS?
OF COURSE. IT HAS TO REACH THE SERVER, AND THE SERVER IS OVER THERE.
WHAT IF SOMEONE TRIPS ON IT?
WHO WOULD WANT TO DO THAT? IT SOUNDS UNPLEASANT.
UH. SOMETIMES PEOPLE DO STUFF BY ACCIDENT.
I DON'T THINK I KNOW ANYBODY LIKE THAT.

https://xkcd.com/908/

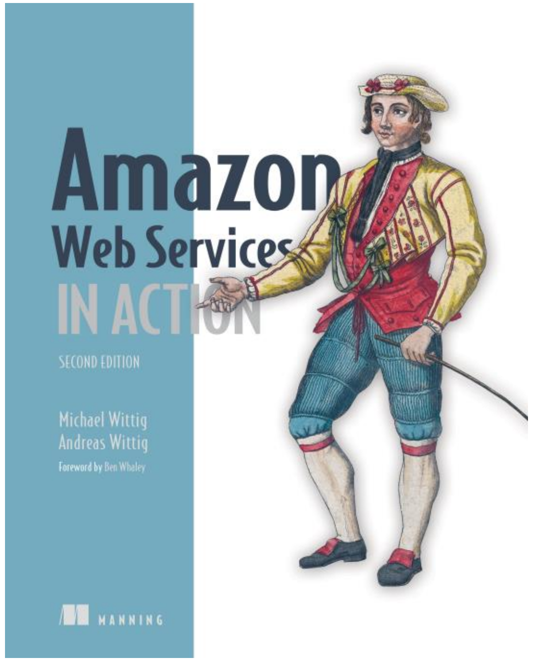
Teaching material

Books


SCIENTIFIC
AND
ENGINEERING
COMPUTATION
SERIES

CLOUD COMPUTING FOR SCIENCE AND ENGINEERING

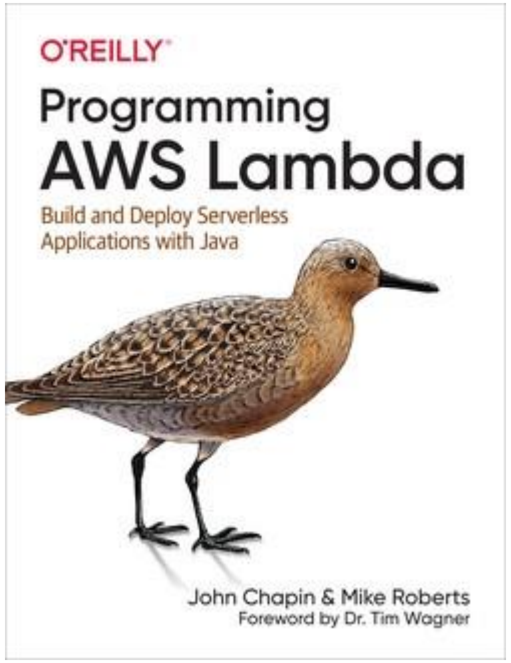
Ian Foster and Dennis B. Gannon


Amazon Web Services IN ACTION

SECOND EDITION

Michael Wittig
Andreas Wittig
Foreword by Ben Whaley

MANNING


O'REILLY

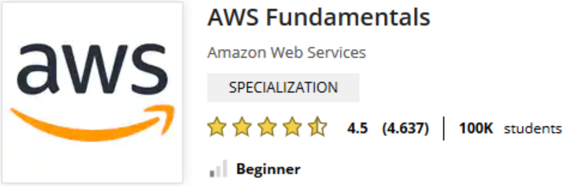
Programming AWS Lambda

Build and Deploy Serverless Applications with Java

John Chapin & Mike Roberts
Foreword by Dr. Tim Wagner

Generic → Specific

Web content


AWS Fundamentals

Amazon Web Services

SPECIALIZATION

4.5 (4.637) | 100K students

Beginner

AWS

58.870 follower

Why going cloud?

Cloud computing (National Institute of Standards and Technology)

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

- On-demand self-service (consume services when you want)
- Broad network access (consume services from anywhere)
- Resource pooling (infrastructure, virtual platforms, and applications)
- Rapid elasticity (enable horizontal scalability)
- Measured service (pay for the service you consume as you consume)

Why going cloud?

Goal: adjusts capacity to have predictable performance at the lowest cost

Scalability that is not possible on premises

- Scale from one to thousands of servers

Elasticity

- Automatically scale resources in response to run-time conditions
- Core justification for the adoption of cloud

Why going cloud

Hardware scalability

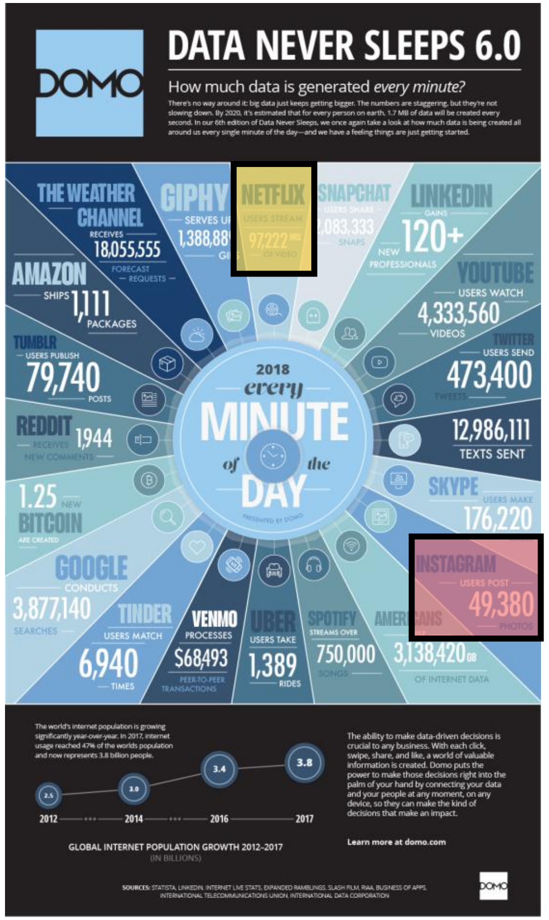
- No longer think about rack space, switches, and power supplies, etc.

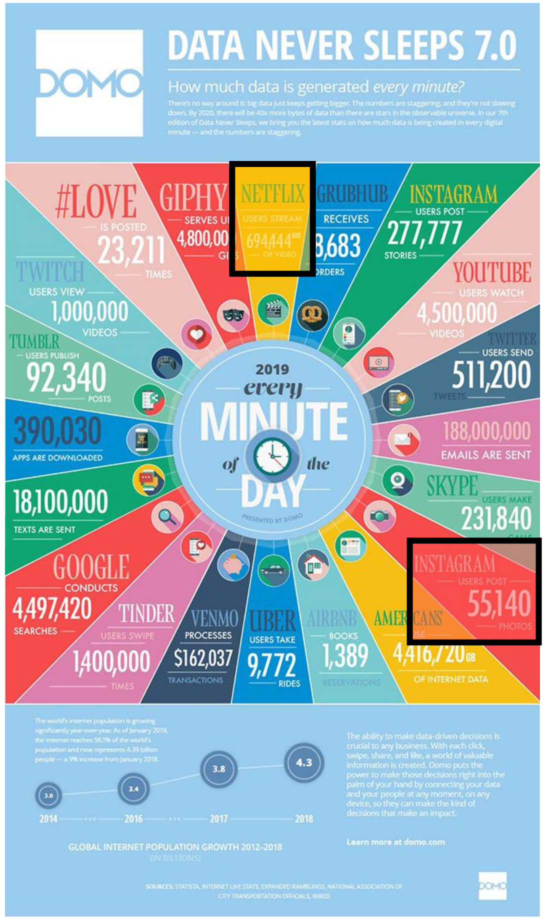
Grow storage from GBs to PBs

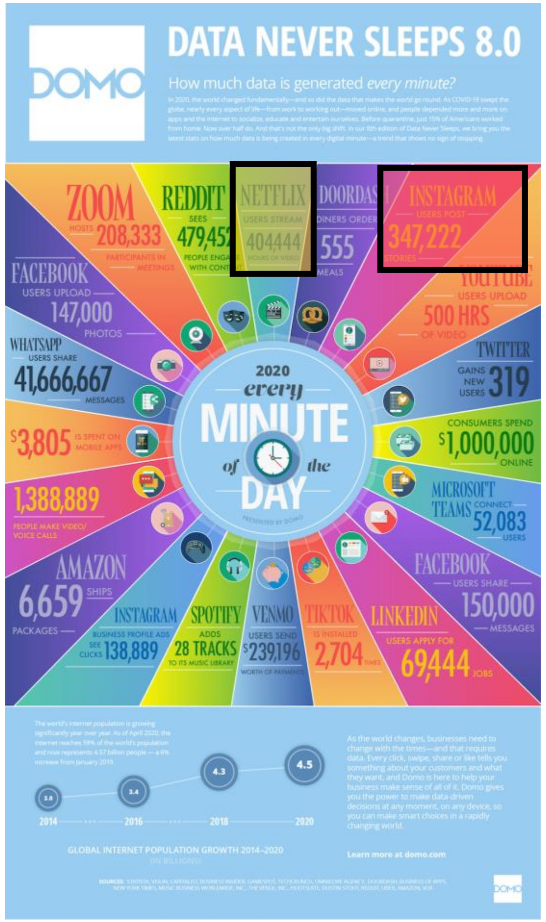
- One hundred of our 10TB Enterprise Capacity 3.5 HDD hard drives

https://blog.seagate.com/business/linus-tech-tips-want-petabyte-system/

Why going cloud?







https://www.domo.com/learn/data-never-sleeps-8

Why going cloud?

Resource pooling

- Enable a resource to serve different consumers
- Dynamically reassigned according to demands
- Economy of scale

Reliability

- Built to handle failures
- Fault-tolerant or highly available

Worldwide **deployment**

- Deploy applications as close to customers as possible
  - E.g., to reduce network latency
- Improve data locality
- Compliant to privacy regulations (e.g., GDPR)

Why going cloud?

User perspective

- Eliminate repetitive tasks to focus on strategic ones
- Abstract the underlying architecture
- Adapt infrastructure to requirements, create (test) environments on demand

Service **integration**

- Do not reinvent the wheel
- Use services that solve common problems (e.g., load balancing, queuing)

Service integration and abstraction are drivers of change

- From databases to data platforms
- From on-premises hardware to serverless architectures
- From custom to standard data pipelines

BIG DATA

From databases to data platforms

Data platform

Companies are collecting tons of data to enable advanced analytics

- Data are more and more heterogeneous and complex
- Raw data are difficult to obtain, interpret, describe, and maintain
- There is a need for describing/curating the data to make them consumable

Databases/warehouses are no longer ideal data hubs for integration/analysis

- Getting value from data is not only a matter of storage
- Need integrated and multilevel analytical skills and techniques

Data platform

Database

> *"A database is a structured and persistent collection of information about some aspect of the real world organized and stored in a way that facilitates efficient retrieval and modification. The structure of a database is determined by an abstract data model. Primarily, it is this structure that differentiates a database from a data file. The most popular data model is relational that represents data as a set of tables."*

Data Warehouse

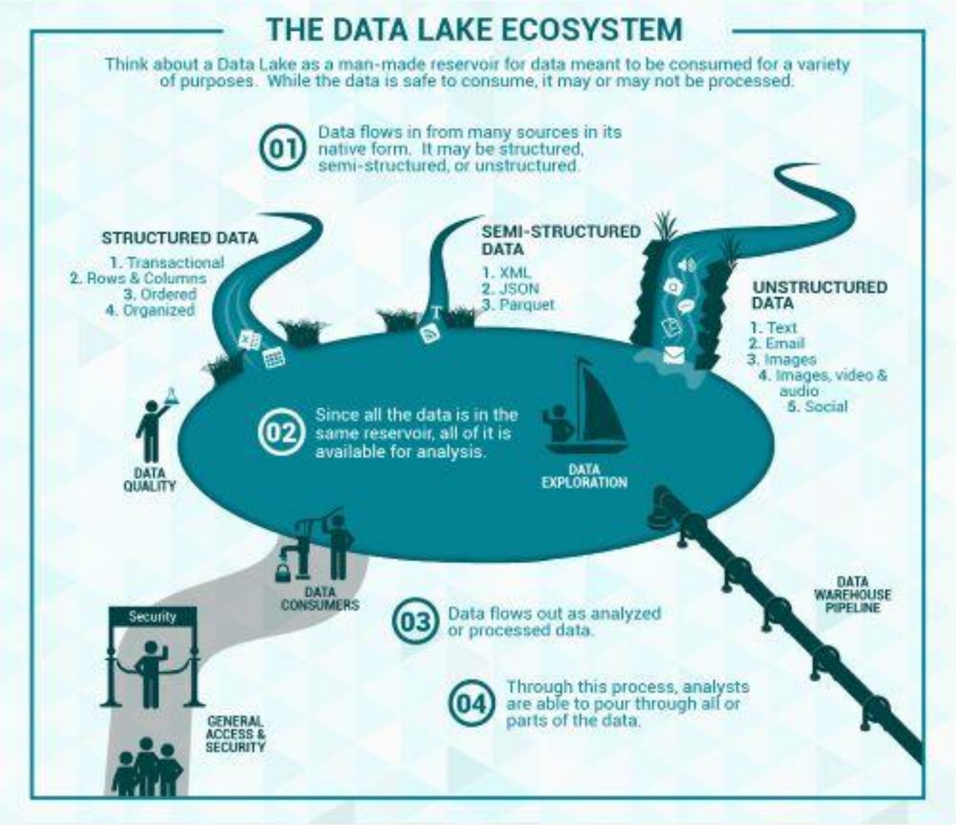
> *"A collection of data that supports decision-making processes. It provides the following features: subject-oriented, integrated and consistent, not volatile."*

Özsu M.T. (2018) Database. In: Encyclopedia of Database Systems. Springer, New York, NY. https://doi.org/10.1007/978-1-4614-8265-9_80734
Matteo Golfarelli and Stefano Rizzi. *Data warehouse design: Modern principles and methodologies*. McGraw-Hill, Inc., 2009.

Data platform

Data lake

Couto et al.: *“A DL is a central repository system for storage, processing, and analysis of raw data, in which the data is kept in its original format and is processed to be queried only when needed. It can store a varied amount of formats in big data ecosystems, from unstructured, semi-structured, to structured data sources”*



Think about a Data Lake as a man-made reservoir for data meant to be consumed for a variety of purposes. While the data is safe to consume, it may or may not be processed.

01 Data flows in from many sources in its native form. It may be structured, semi-structured, or unstructured.

02 Since all the data is in the same reservoir, all of it is available for analysis.

03 Data flows out as analyzed or processed data.

04 Through this process, analysts are able to pour through all or parts of the data.

Couto, Julia, et al. "A Mapping Study about Data Lakes: An Improved Definition and Possible Architectures." *SEKE*. 2019.
https://dunnsolutions.com/business-analytics/big-data-analytics/data-lake-consulting

Data platform

Data lakes have increasingly taken the role of data hubs

- Eliminate up-front costs of ingestion since data are stored in original format
- Once in DL, data are available for analysis by everyone in the organization

Drawing a sharp line been storage/computation/analysis is hard

- Is a database just storage?
- What about SQL/OLAP?

Blurring of the architectural borderlines

- DL is often replaced by “data platform” or “data ecosystem”
- Encompass systems supporting data-intensive storage, computation, analysis

Data platform

Data platform (e.g., on Google Cloud and Amazon AWS)

- Rationale: relieve users from complexity of administration and provision
  - Not only technological skills, but also privacy, access control, etc.
  - Only focus on functional aspects

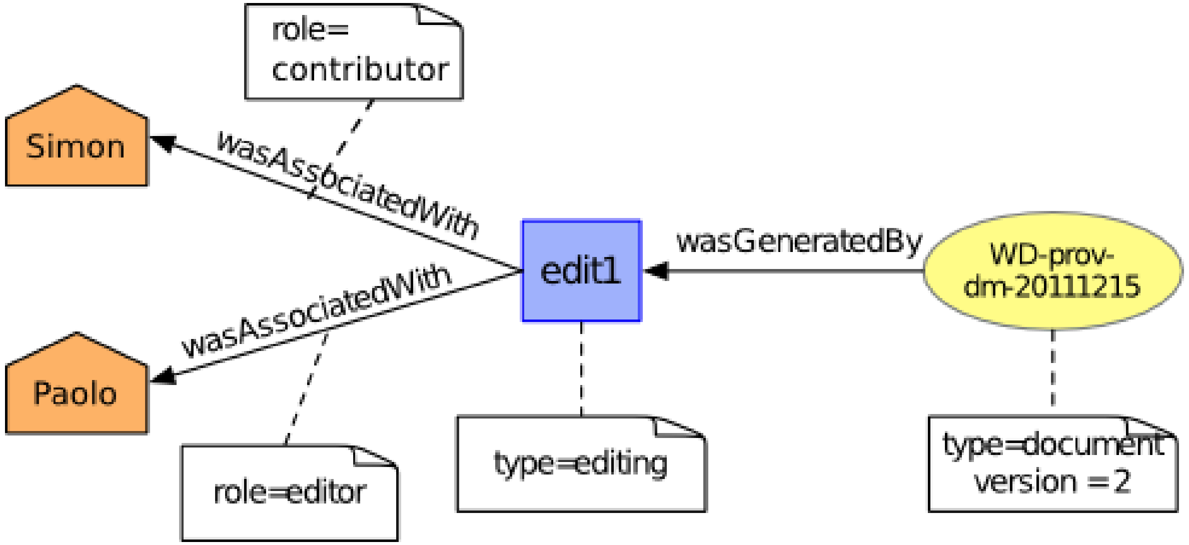
Are we done? No!

- Lacking smart support to govern the complexity of data and transformations
- Data transformations must be governed to prevent DP turns into a swamp
  - Amplified in data science, with data scientists prevailing data architects
  - Leverage descriptive metadata and maintenance to keep control over data

Data platform

Data provenance

- Metadata pertaining to the history of a data item
- Pipeline including the origin of objects and operations they are subjected to
- We have a standard: https://www.w3.org/TR/prov-dm/



https://www.w3.org/TR/prov-dm/

Data provenance

Entity

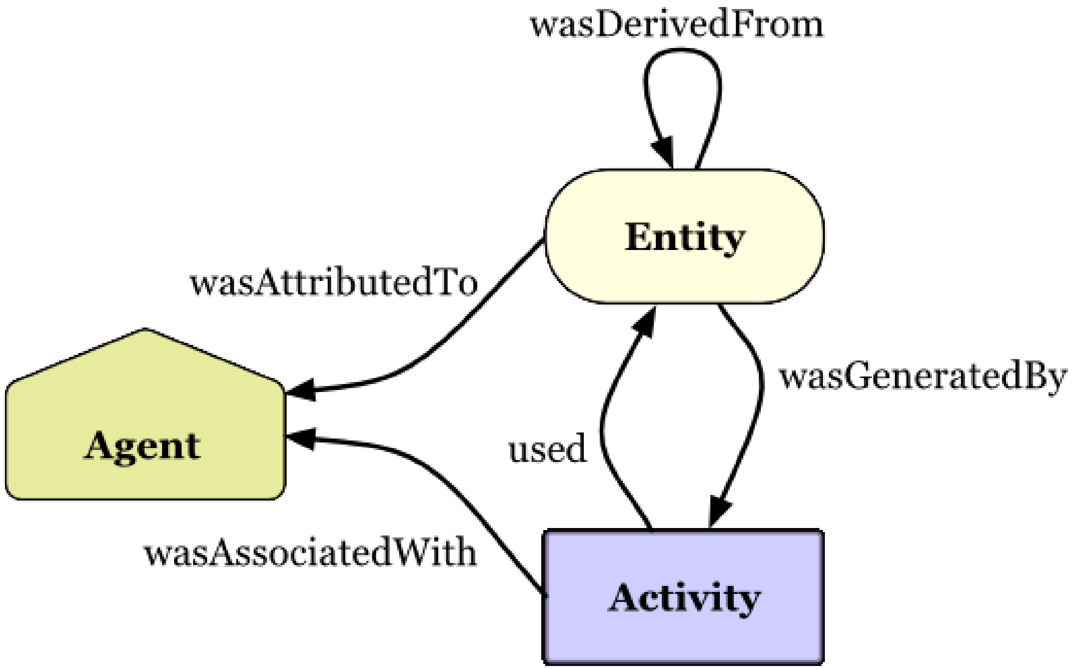
- Physical/conceptual things

Activity

- Dynamic aspects of the world, such as actions
- How entities come into existence, often making use of previously existing entities

Agent

- A person, a piece of software
- Takes a role in an activity such that the agent can be assigned some degree of responsibility for the activity taking place



https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/

Data provenance

Data quality

- Monitoring of the quality (e.g., accuracy) of the objects produced
- Notify when a transformation pipeline is not behaving as expected

Debugging

- Inferring the cause of pipeline failures is challenging
- Store inputs of each operation with versions and environmental settings (RAM, CPUs, etc.)

And so on...

Data platform

Are we done? No!

- Metadata can become bigger than data themselves

We need meta meta-data (or models)...

- ... chasing our own tails

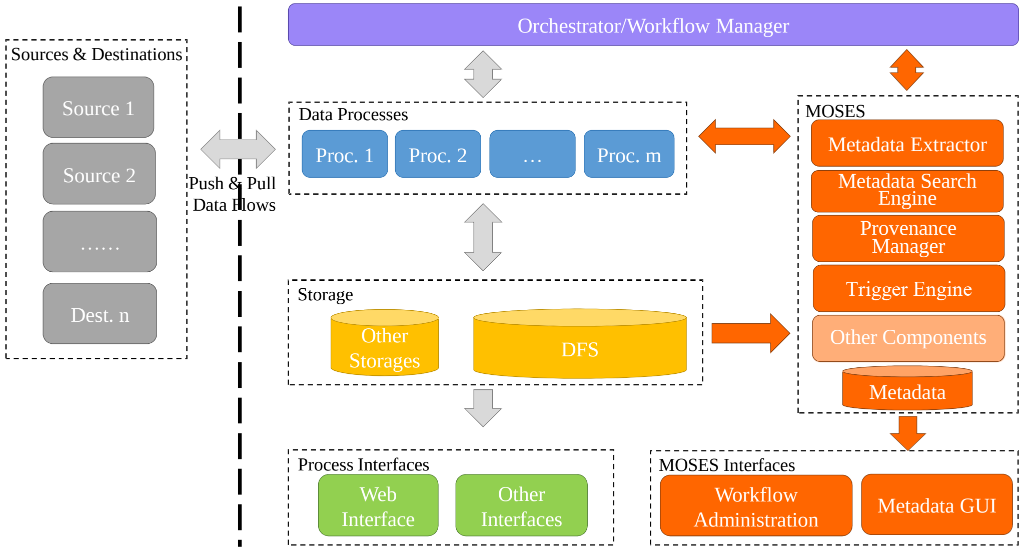
Data management is still a (research) issue in data platforms

Data platform

MOSES: making data platform smarter

Functional architecture

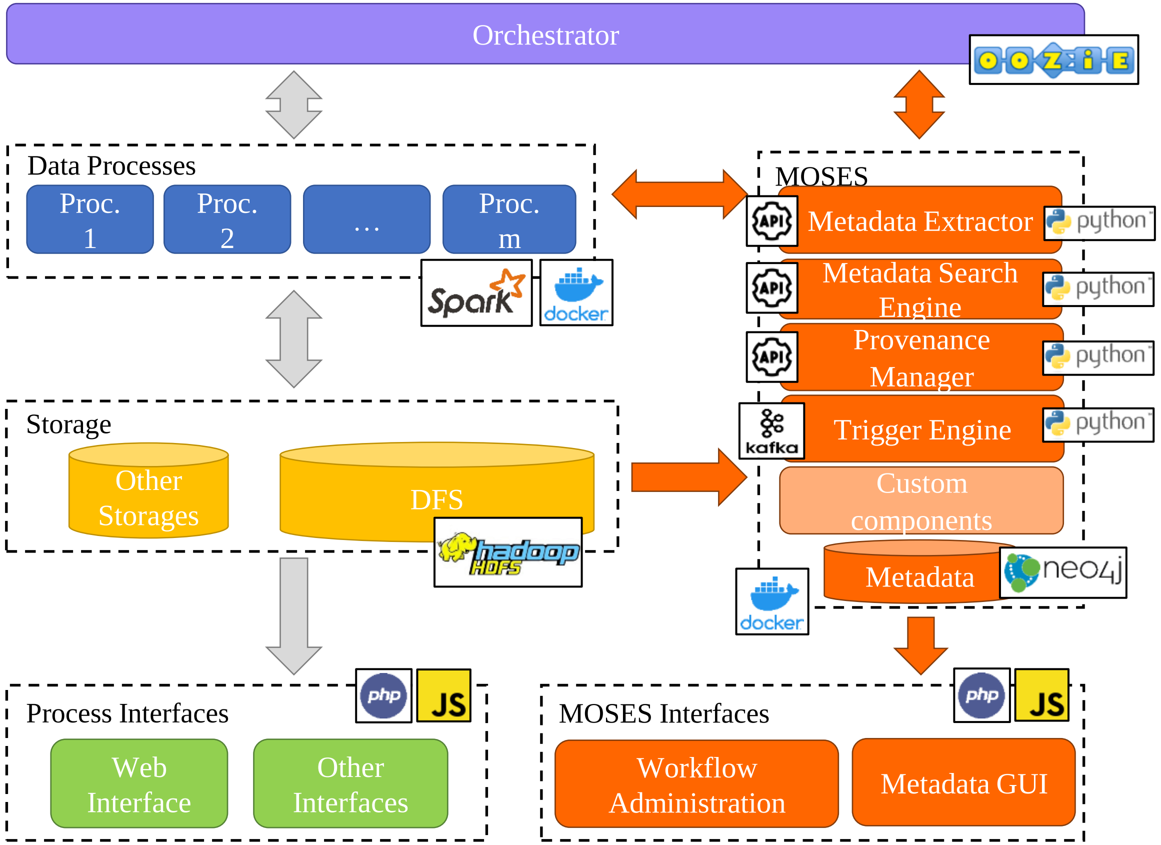
- Components of MOSES are in orange
- The others are standard components in charge of producing/consuming, processing, storing, and visualizing data
- The orchestrator (e.g., Oozie in the Hadoop ecosystem) manages (e.g., schedules) the data transformation processes



Data platform

MOSES is used within a DP that builds atop the Hadoop ecosystem

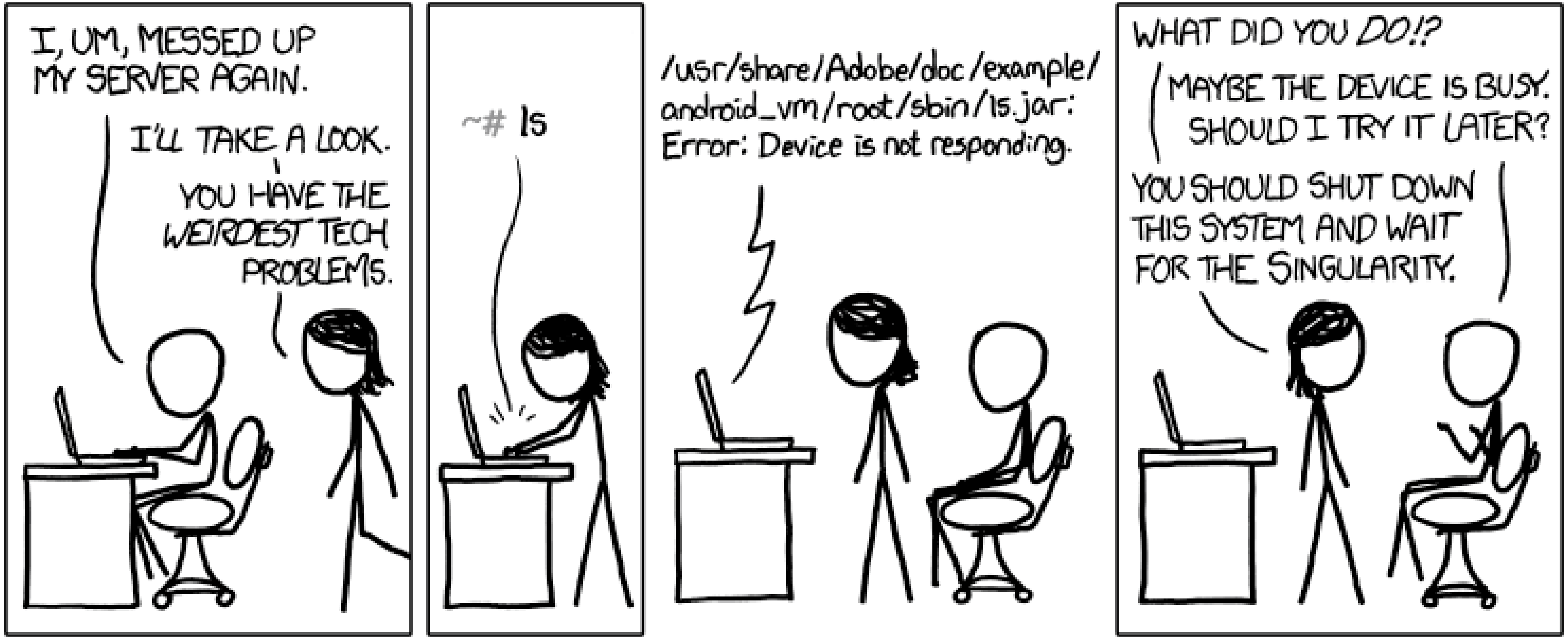
- The cluster runs the Cloudera Distribution for Apache Hadoop 6.2.0 and Docker
- The metadata collected and manipulated are stored in a graph database
  - The graph data model favors the modeling of highly interconnected data in the absence of fixed schemas
- Service decoupling: functional components are accessible using RESTful APIs



Leoni Anna Giulia. "Gestione di un data lake strutturato attraverso il riconoscimento semantico dei dati acquisiti."

BIG DATA

From PaaS to FaaS (serverless)


I, UM, MESSED UP MY SERVER AGAIN.
I'LL TAKE A LOOK.
YOU HAVE THE WEIRDEST TECH PROBLEMS.
~# ls
/usr/share/Adobe/doc/example/android_vm/root/sbin/ls.jar: Error: Device is not responding.
WHAT DID YOU DO!?
MAYBE THE DEVICE IS BUSY. SHOULD I TRY IT LATER?
YOU SHOULD SHUT DOWN THIS SYSTEM AND WAIT FOR THE SINGULARITY.

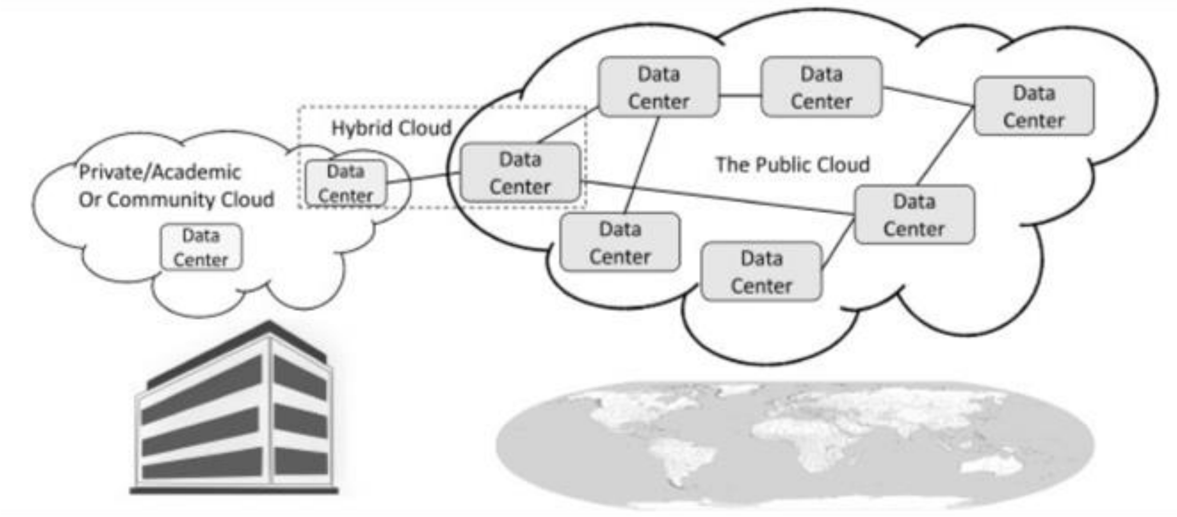
https://xkcd.com/1084/

From PaaS to FaaS

Public: accessible to anyone willing to pay (e.g., Microsoft, AWS, Google)

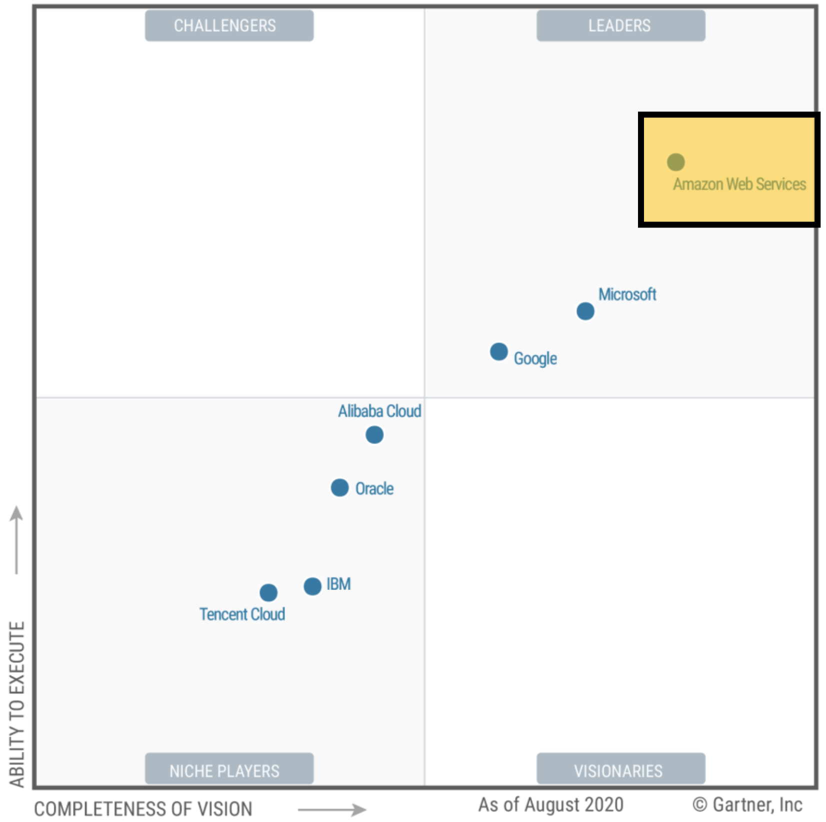
Private: accessible by individuals within an institution

Hybrid: a mix of the previous



From PaaS to FaaS

Figure 1. Magic Quadrant for Cloud Infrastructure and Platform Services



https://www.gartner.com/en/research/methodologies/magic-quadrants-research

Gartner Magic Quadrant

- Understanding the technology providers to consider for an investment
- **Leaders** execute well and are well positioned for tomorrow
- **Visionaries** understand where the market is going but do not yet execute well
- **Niche Players** focus successfully on a small segment, or are unfocused and do not out-innovate or outperform others
- **Challengers** execute well but do not demonstrate an understanding of market direction
- Focusing on leaders isn’t always the best
  - A niche player may support needs better than a market leader. It depends on how the provider aligns with business goals

From PaaS to FaaS

Cloud services are hosted in multiple locations worldwide

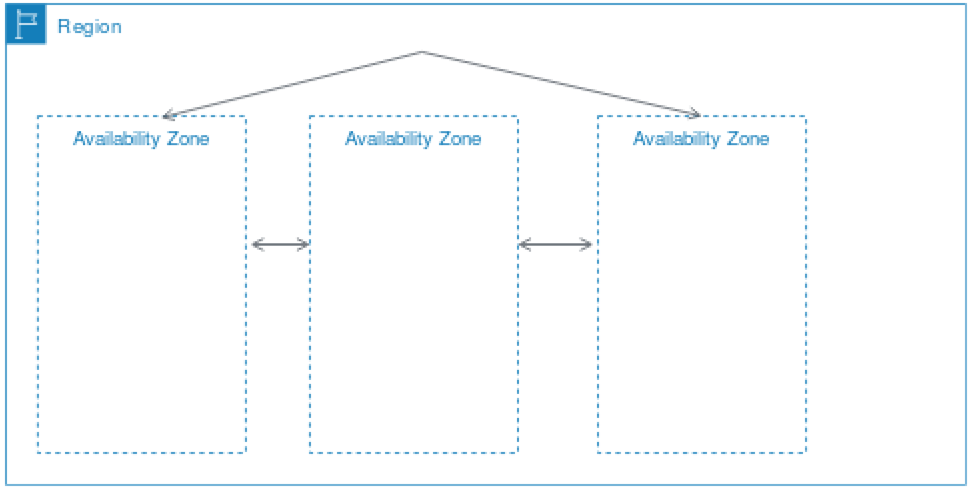
- Locations are composed of **regions** and **availability zones**

Region (e.g., us-east-1)

- Is an independent geographical area
- Has availability zones

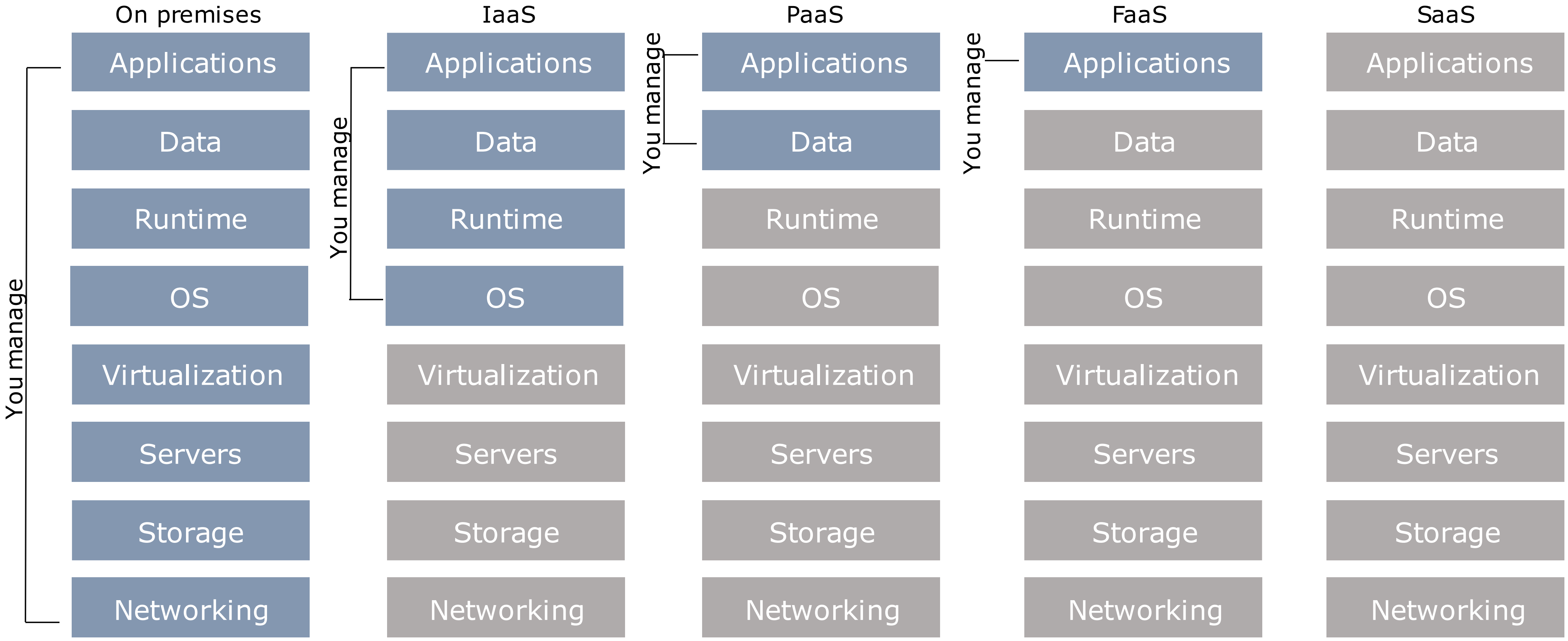
Availability zones in a region

- Are connected through low-latency links
- Resources are usually replicated across zones but not regions



https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

From PaaS to FaaS



On premises

You manage

Applications

Data

Runtime

OS

Virtualization

Servers

Storage

Networking

IaaS

You manage

Applications

Data

Runtime

OS

Virtualization

Servers

Storage

Networking

PaaS

You manage

Applications

Data

Runtime

OS

Virtualization

Servers

Storage

Networking

FaaS

You manage

Applications

Data

Runtime

OS

Virtualization

Servers

Storage

Networking

SaaS

Applications

Data

Runtime

OS

Virtualization

Servers

Storage

Networking

From PaaS to FaaS

Understanding architectures is paramount to successful software systems

- Good architectures help to scale
- Poor architectures cause issues that necessitate a costly rewrite

On-premises

- Provisioning, managing, and patching servers is time-consuming
- Require dedicated operations people
- A non-trivial environment is hard to set up and operate effectively
- Infrastructure and hardware are often a distraction from strategic tasks

From PaaS to FaaS

Infrastructure as a service (IaaS)

- A computing infrastructure provisioned and managed over the internet (e.g., AWS EC2)
- Avoid expense/complexity of buying/managing physical servers/data-centers
- IaaS overcomes issues on-premises
- Possibly requires to manage many environments

Platform as a Service (PaaS)

- A development and deployment environment in the cloud (e.g., AWS Elastic Beanstalk)
- Support complete application life-cycle: building, testing, deploying, etc.
- Avoid expense/complexity of managing licenses and application infrastructure

From PaaS to FaaS

PaaS and **containers** are potential solutions to inconsistent infrastructures

PaaS provides a platform for users to run their software

- Developers write software targeting features/capabilities of the platform

Containerization isolates an application with its own environment

- Lightweight alternative to full virtualization
- Containers are isolated but need to be deployed to (public/private) server
- Excellent solution when dependencies are in play
- Housekeeping challenges and complexities

From PaaS to FaaS

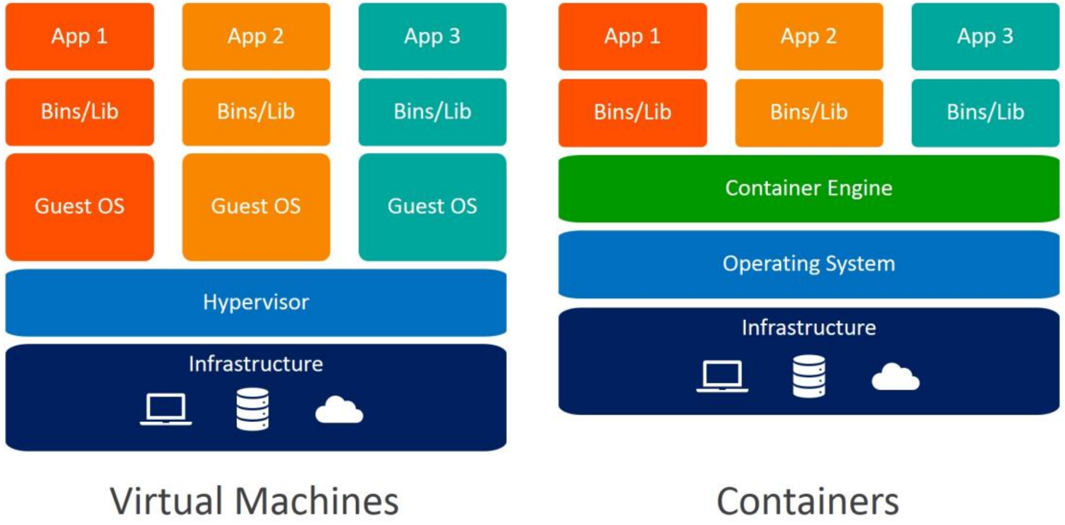
Containers and virtual machines are packaged computing environments

Containers

- On top of physical server and its host OS
- Share the host OS kernel
- Shared components are read-only
- Containers are “light” and take just seconds to start

Virtual machines

- Emulate a hardware/software system
- On top of a hypervisor (VM monitor)



From PaaS to FaaS

Function as a Service (FaaS)

- A coding environment, cloud provider provisions platform to run the code (e.g., AWS Lambda)
- Infrastructure provisioning and management are invisible to the developer

Software as a service (SaaS)

- An application environment
- Access cloud-based apps over the Internet (e.g., email, Microsoft Office 365)

From PaaS to FaaS

Serverless

- A software architecture that does not rely on direct access to a server
- Embodies principles from microservices
  - Small, standalone, fully independent services built for a specific purpose
  - Cloud provider is responsible for integration

Principles of FaaS architectures

- FaaS is based on a serverless approach
  - Use a compute service to execute code on demand (no servers/containers)
- Every function could be considered as a standalone service
- Write single-purpose stateless functions

From PaaS to FaaS

Functions react to events

- Design push-based, event-driven pipelines
- Create thicker, more powerful front ends
- Embrace third-party services (e.g., security)

FaaS is not a silver bullet

- Not appropriate for latency-sensitive applications
- Strict specific service-level agreements
- Migration costs
- Vendor lock-in can be an issue