Cloud Infrastructures for Big Data Platforms

Hong-Linh Truong
Department of Computer Science
linh.truong@aalto.fi, https://rdsea.github.io

CS-E4640 Big Data Platforms, Spring 2024, Hong-Linh Truong
16/1/2024

This is not a cloud computing lecture but how cloud computing is important for big data platforms

Learning objectives

- **Understand key cloud technologies**

- **Understand how cloud technologies empower big data platforms**

- **Understand cloud technologies enable us to acquire, utilize and manage resources for big data platforms**

Service Model

- **Services offer well-defined interfaces for**
  - access resources: data, things, machines, and people
  - provide functions: ingestion, computation, sensing, analytics, inferences, etc.
  - offer diverse service level agreements (SLAs) for different types of business models (e.g.,pay-per-use and subscription)
- **Services are**
  - characterized by scalability, reliability, elasticity, etc.
  - provisioned in distributed systems of IoT, edge, cloud and HPC infrastructures, possibly a combination of different types of systems
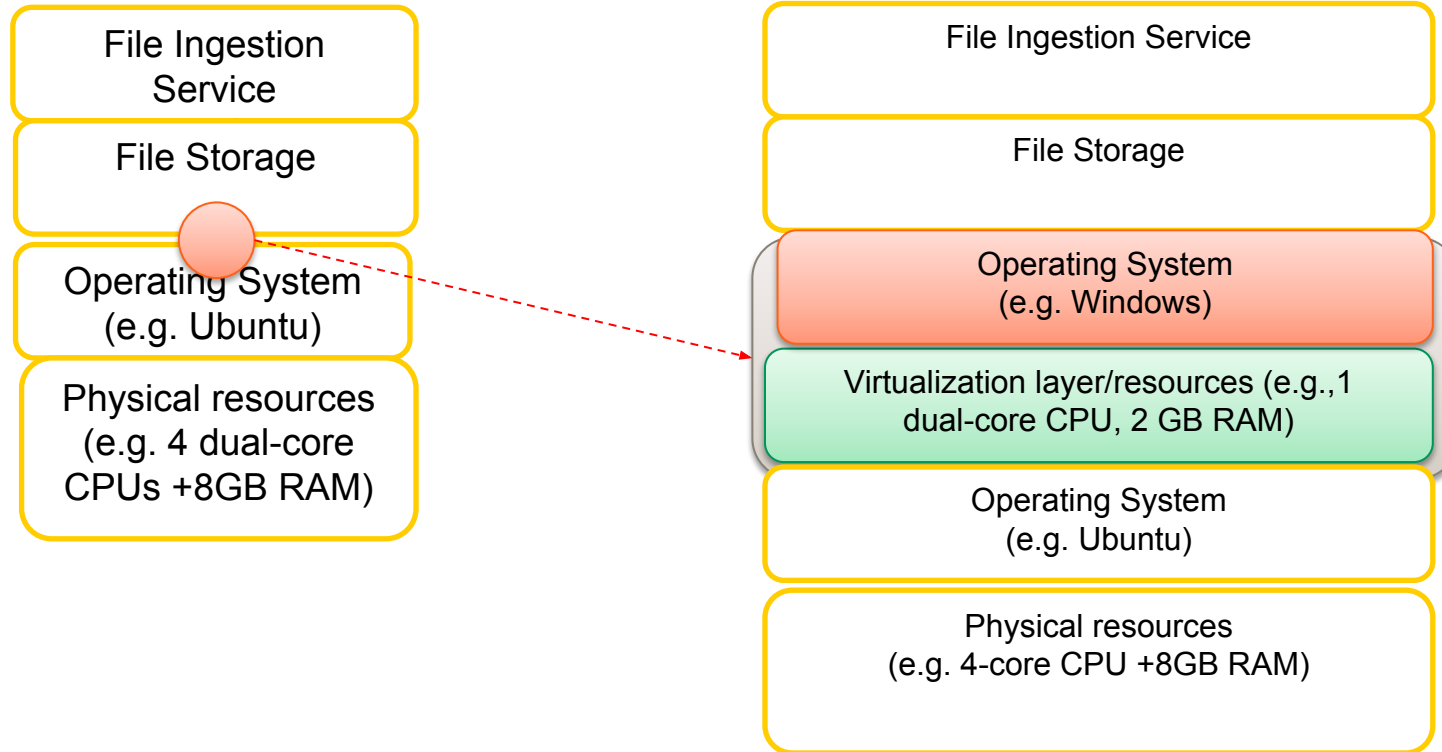
Virtualization

- **Virtualization**
  - abstracts low-level compute, data and network resources to create *virtual version* of these resources
  - virtualization software creates and manages "virtual resources" isolated from physical resources

- **Virtualization is a powerful concept**
  - we can apply virtualization techniques virtually for everything!

Virtualizing physical resources

Main types of virtualization of infrastructures

- **Compute resource virtualization**
  - compute resources: CPU/GPU, memory, I/O, disks, etc.
  - "virtual machines"/containers
- **Storage virtualization**
  - resources: storage devices, hard disks, etc.
  - for usage and management of data storage
- **Network Function virtualization**
  - network resources: network equipment & functions
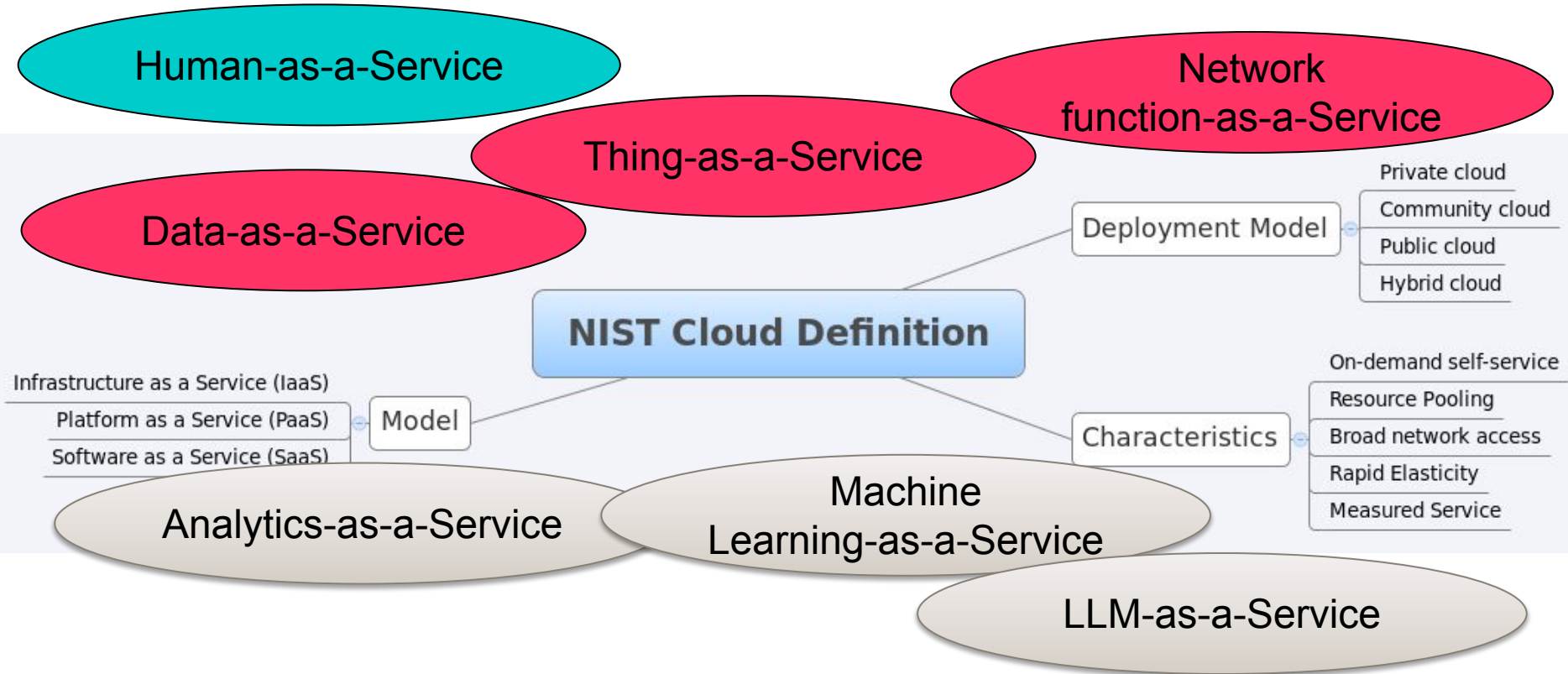 - dynamically provision and manage network functions

Cloud Computing

Original definition from NIST

"This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models."

Source: NIST Definition of Cloud Computing v15, http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc

Cloud Computing

Human-as-a-Service

Network function-as-a-Service

Thing-as-a-Service

Data-as-a-Service

Deployment Model
- Private cloud
- Community cloud
- Public cloud
- Hybrid cloud

NIST Cloud Definition

Infrastructure as a Service (IaaS)
Platform as a Service (PaaS)
Software as a Service (SaaS)

Model

Characteristics
- On-demand self-service
- Resource Pooling
- Broad network access
- Rapid Elasticity
- Measured Service

Analytics-as-a-Service

Machine Learning-as-a-Service

LLM-as-a-Service

Cloud computing principles

- **"Cloud"**
  - not just data centers or public cloud infrastructures
- **For big data platforms: we need the "cloud mindset"**
  - apply cloud principles for developing and operating big data platforms
  - big data platforms can be in on-premise infrastructures empowered with cloud technologies!
  - both **multicloud and hybrid cloud** systems can be used used

Aalto University
School of Science

Compute resource virtualization technologies

- **Physical compute resources for big data platforms**
  - individual physical hosts/servers (CPU, memory, I/O)
  - clusters, supercomputer, and data centers
- **At the low-level: two main streams**
  - hypervisor/Virtual Machine monitor
    - *Virtual machines (VirtualBox, VMWare, Zen, etc.)*
  - containterlization
    - *Containers (Linux Containers, Docker, Warden Container, OpenVZ, OCI based containers, etc.)*

Virtual infrastructural resources for big data platforms

- **For big data platforms: we leverage clusters/infrastructures of VMs/containers**
  - computing resources for core services and data
    - *e.g., data storage, data ingestion, data processing, and messaging*
- **On-demand resources for large-scale deployments**
  - compute nodes, storage, communication, etc.
  - virtual data centers work like a single distributed system
- **On-demand resources for elastic workload**
  - e.g., for data ingestion and analytics tasks

Example: OpenStack

- **A Big Data Platform can be built based on OpenStack (or similar)-based data center**
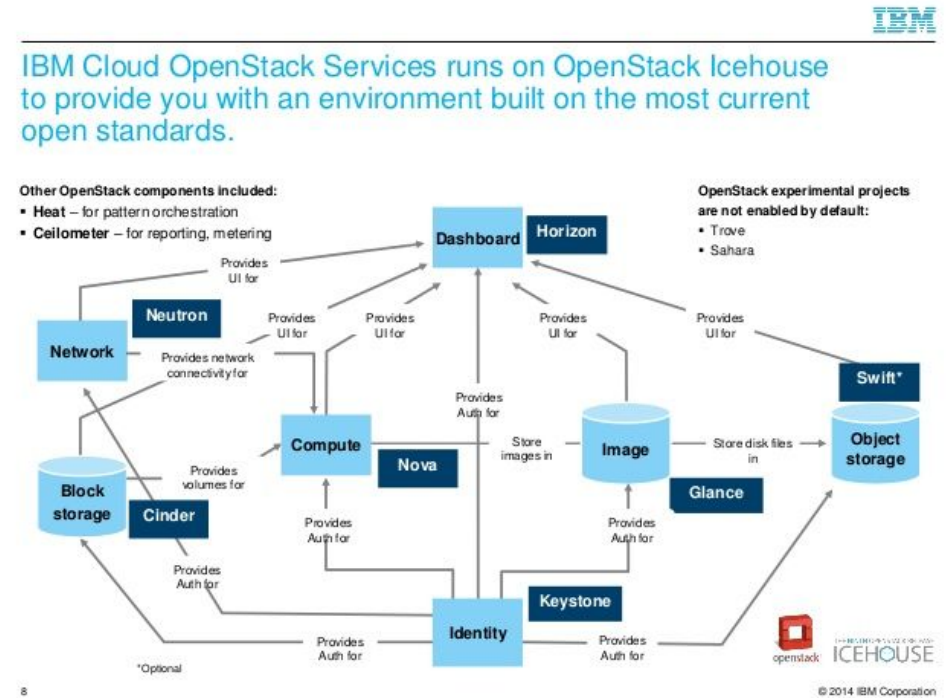
Figure source: http://www.slideshare.net/OpenStack_Online/ibm-cloud-open-stack-services

Example: Kubernetes

- **Support Docker, rkt, runc, etc.**
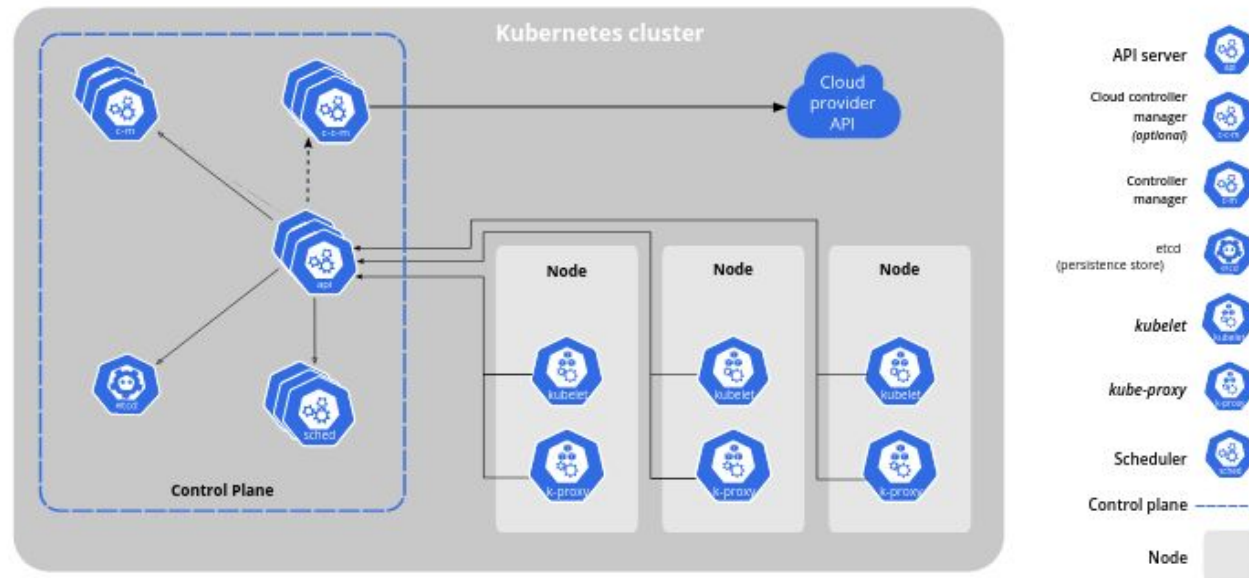- **Act as a resource orchestration and management for big data platforms**

Figure source: https://kubernetes.io/docs/concepts/architecture/cloud-controller/

Example: storage virtualization

- **Low-level storage**
  - e.g., VMware Virtual Machine File Systems
- **High-level, e.g., database**
  - MySQL Cluster + auto-sharding

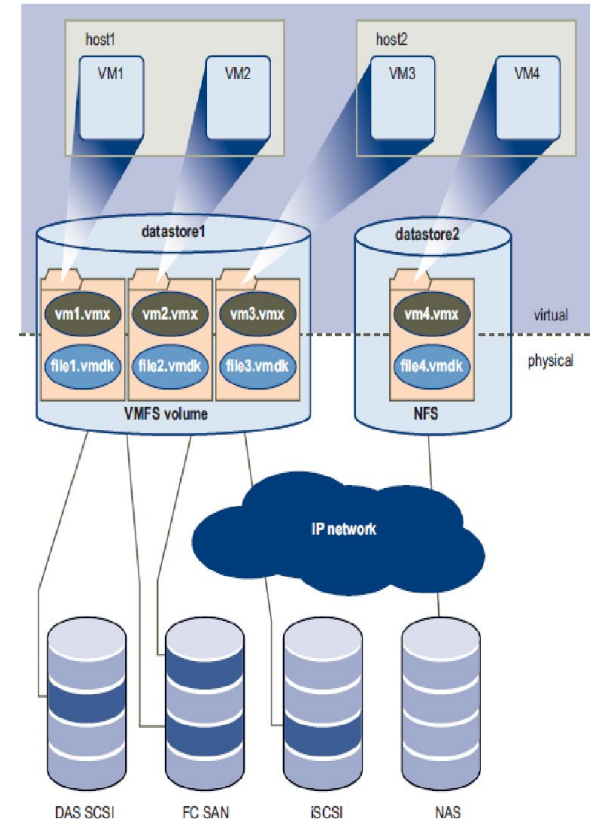**Can be used as the Data Storage layer for Data Lake or Data Services**

Figure source:
https://www.vmware.com/pdf/vi_architecture_wp.pdf

#1: Enabling managed services

Many options for resource provisioning in big data platforms

- **Managing *infrastructural resources* for big data platforms must be easier with virtualization**
  - different types of storage: block storage and file storage
  - many CPU/memory configurations: single core to many cores
  - suitable for different workloads
- **Different SLA offerings**
  - reliability, security, performance, maintainability ...
- **Elasticity**
- **Globalization support: important for many businesses**
  - design with multiple clouds and hybrid clouds

Aalto University
School of Science

On-demand storage provisioning

- **Big data requires big storage which can be changed on-demand!**
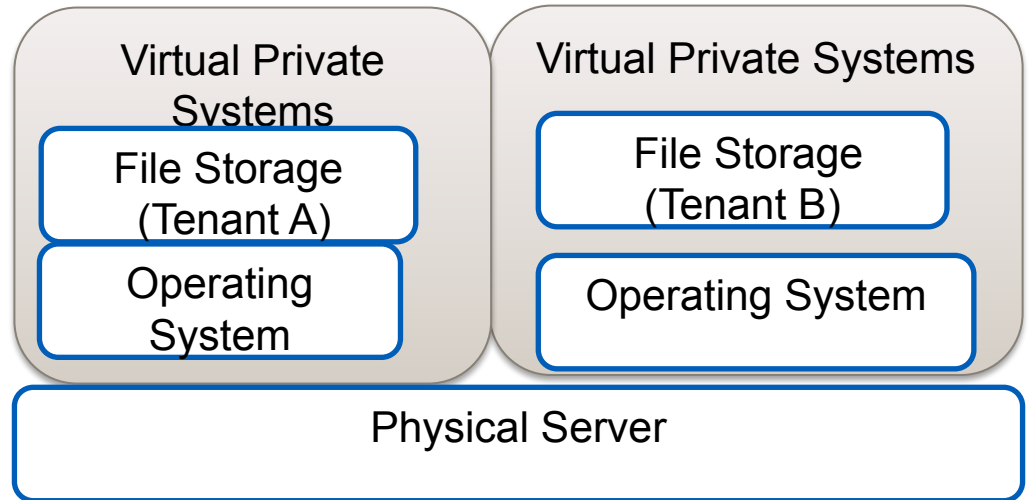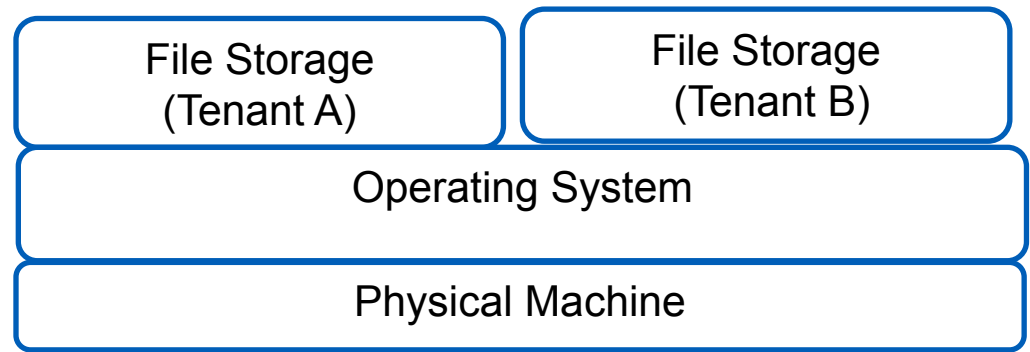

Dynamic numbers and sizes
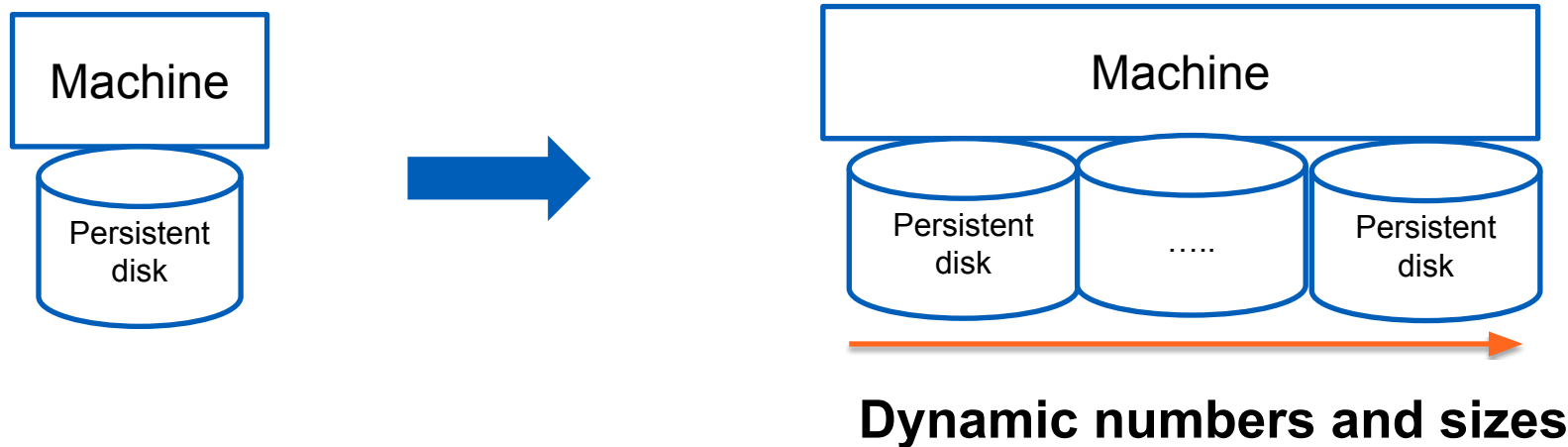
- **Examples of dynamic configurations:**
  - Google persistent disks: different types of disks, can have 128 disks and max 257 TiB for a single VM

**Aalto University
School of Science**

Flexible computing capabilities

- **Different workloads and programming models in big data platforms need flexible computing resource provisioning**
  - Storage, Data Ingestion, and Analysis
- **Examples**
  - small data ingestion/analysis jobs
    - *with pandas, scikit-learn (few cores)*
  - large-scale MapReduce/Spark ⇒ clusters of VMs/containers
  - Machine Learning with TensorFlow ⇒ TPU (Tensor Processing Unit)
- **Cloud technologies easily enable different computing capability configurations**

Security improvement

- **Tenant's service isolation while platforms support multiple tenants**

- **Virtual private instances for security and performance of tenant's data!**

File Storage (Tenant A) | File Storage (Tenant B)

Operating System

Physical Machine

Virtual Private Systems
- File Storage (Tenant A)
- Operating System

Virtual Private Systems
- File Storage (Tenant B)
- Operating System

Physical Server

#2: Achieving fault tolerance, performance and elasticity

**Aalto University
School of Science**

Easing Replication Management

Passive (Primary backup) model:

- FE (Front-end) can interface to a Replication Manager (RM) to serve requests from clients.
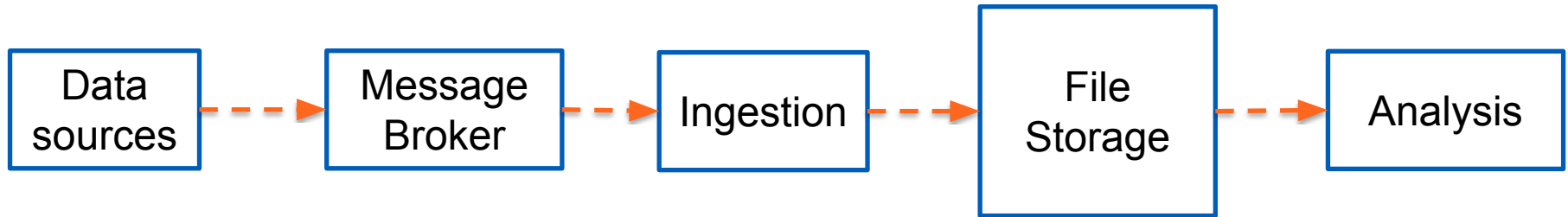- E.g., in MongoDB
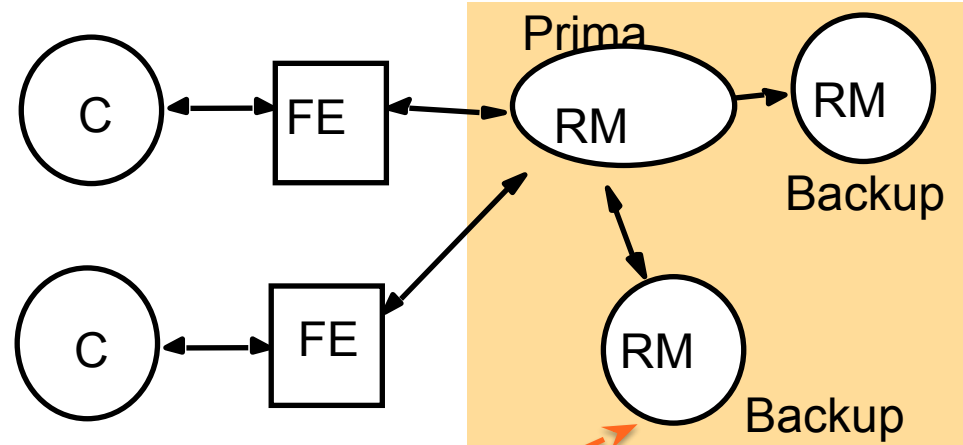
Easy to deploy, globalize, manage and replace RM using cloud resources

Figure source: Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design Edn. 5

Aalto University
School of Science

High availability and performance

- **Cost/optimization**
  - elasticity, hot deployment, etc.
  - cloud bursting with hybrid cloud (combining private/on-premise + public resources)
- **Improving service performance in incident management**
  - e.g., spend time to fix a machine or just quickly relaunch a new one (and fix the old one later) ?

Scaling in every place of the data pipelines

```
Data
sources  ┄┄▶  Message   ┄┄▶  Ingestion  ┄┄▶  File      ┄┄▶  Analysis
              Broker                           Storage
```
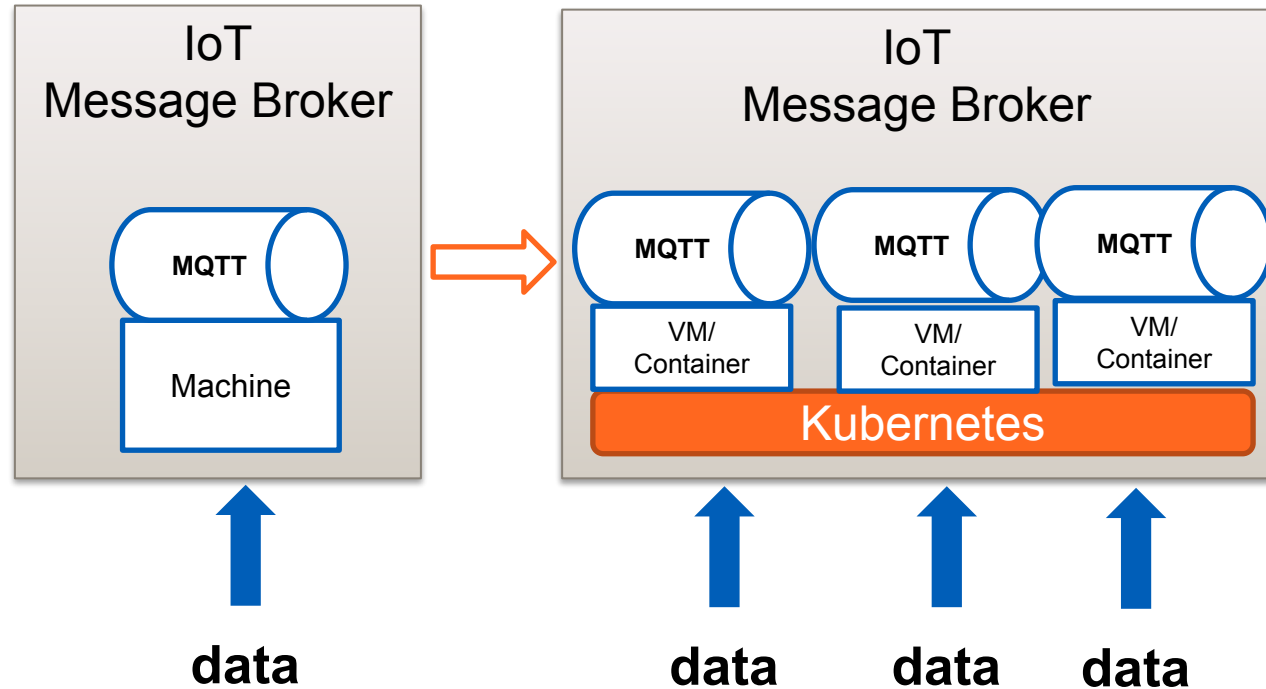
- **Scaling**
  - disk spaces for file storage
  - resources for data ingestion
  - resources for data analysis

Happen at different times and locations

**Aalto University
School of Science**

Scaling middleware nodes

- **Increase the number of brokers when more data arrive**

- **Provide dedicated brokers on-demand**

Aalto University
School of Science

Example: scaling compute nodes for data analysis

On-demand change

4 nodes → **6 nodes**

Aalto University
School of Science

#3: Living in the world of Microservices & DevOps

Aalto University
School of Science

Microservices

- **Many components for data storage, data processing and ingestion**
  - microservices can be used to design components of big data platforms
  - in particular: services serving data requests and services for storing data in the platform
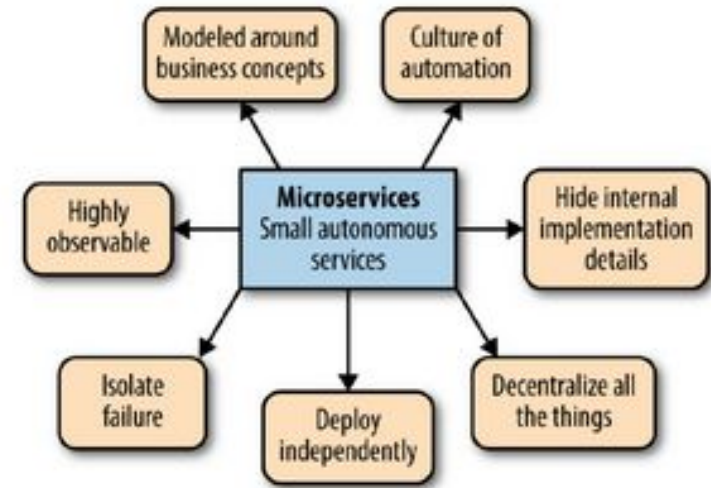- **Big data platforms provide features for other microservices**

Figure source: Sam Newman, Building Microservices, 2015

Aalto University
School of Science

Examples

Microservices for both services using big data platforms and components of data platforms

eShopOnContainers reference application
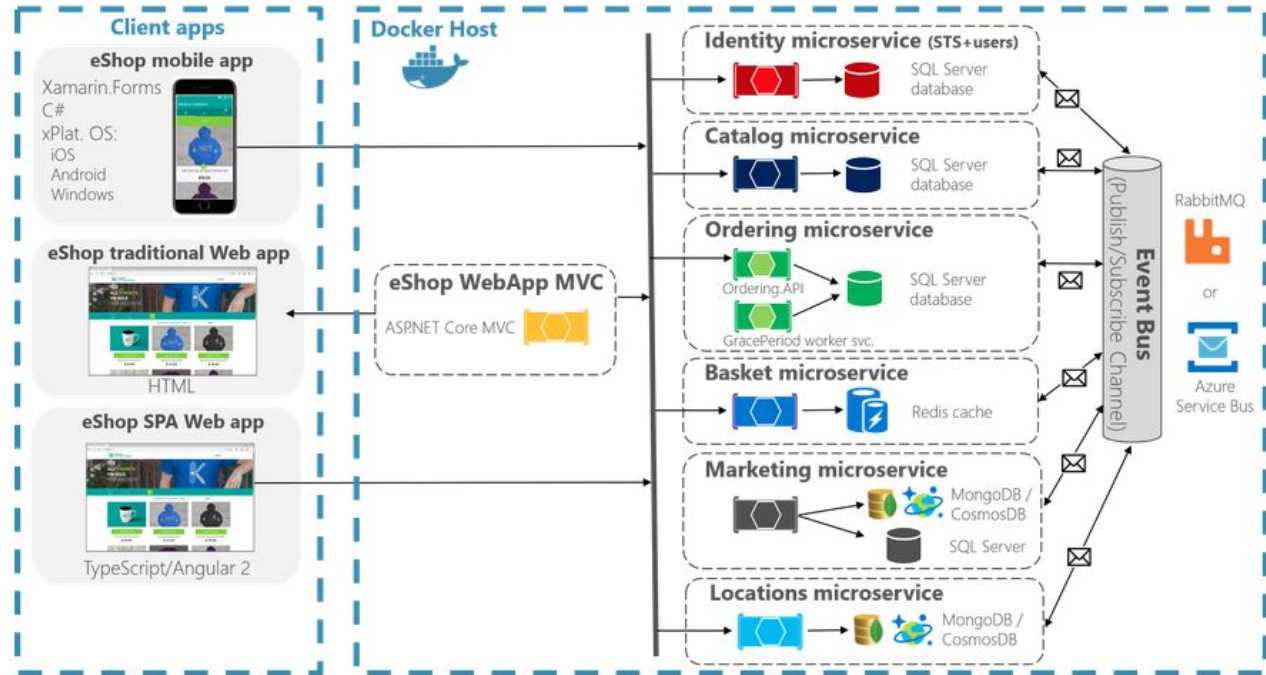(Development environment architecture)

Figure source:
https://blogs.msdn.microsoft.com/dotnet/2017/08/02/microservices-and-docker-containers-architecture-patterns-and-development-guidance/

**Aalto University
School of Science**

DevOps

- **Close the gap between development/test environment and real/production environments**

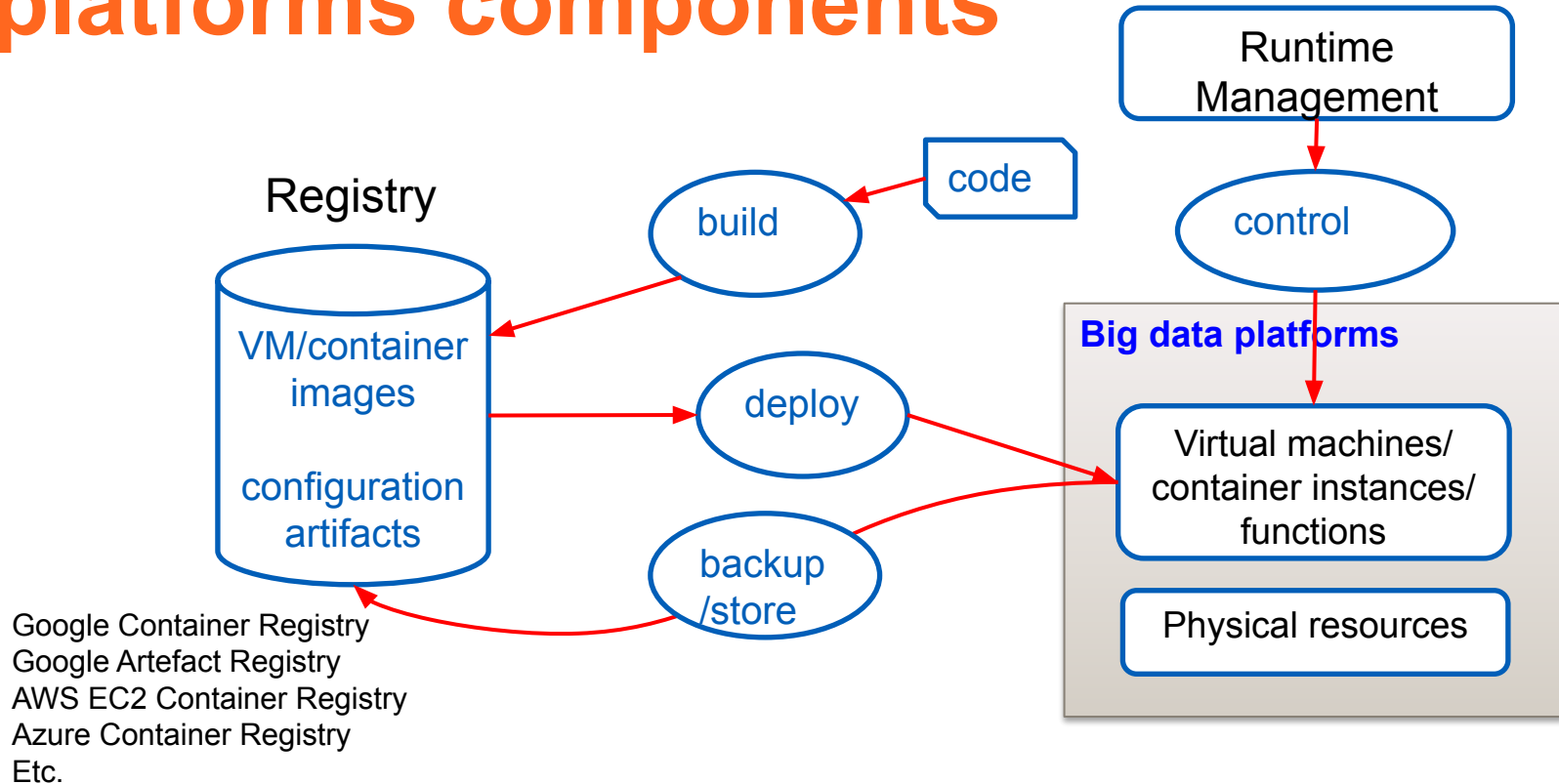- **Simplify testing, emulating real environments, etc.**

DevOps for big data platforms are part of software systems under DevOps. However, many tasks have to deal with requirements from big data analytics and services!

For example: testing a data pipeline (data intensive test cases) != testing an integrated set of software components

Tools and frameworks and providers for infrastructural resources provisioning and service deployment:

Chef, Vagrant, Terraform, Amazon, Google, Microsoft, OpenStack, OpenShift, …

Aalto University
School of Science

Provisioning and management of platforms components

Runtime Management

code

build

Registry

VM/container images

configuration artifacts

deploy

backup /store

control

Big data platforms

Virtual machines/ container instances/ functions

Physical resources

Google Container Registry
Google Artefact Registry
AWS EC2 Container Registry
Azure Container Registry
Etc.

Aalto University
School of Science

Thanks!

Hong-Linh Truong
Department of Computer Science

rdsea.github.io