Aalto University
School of Science

Cloud Infrastructures for Big Data Platforms

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

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**

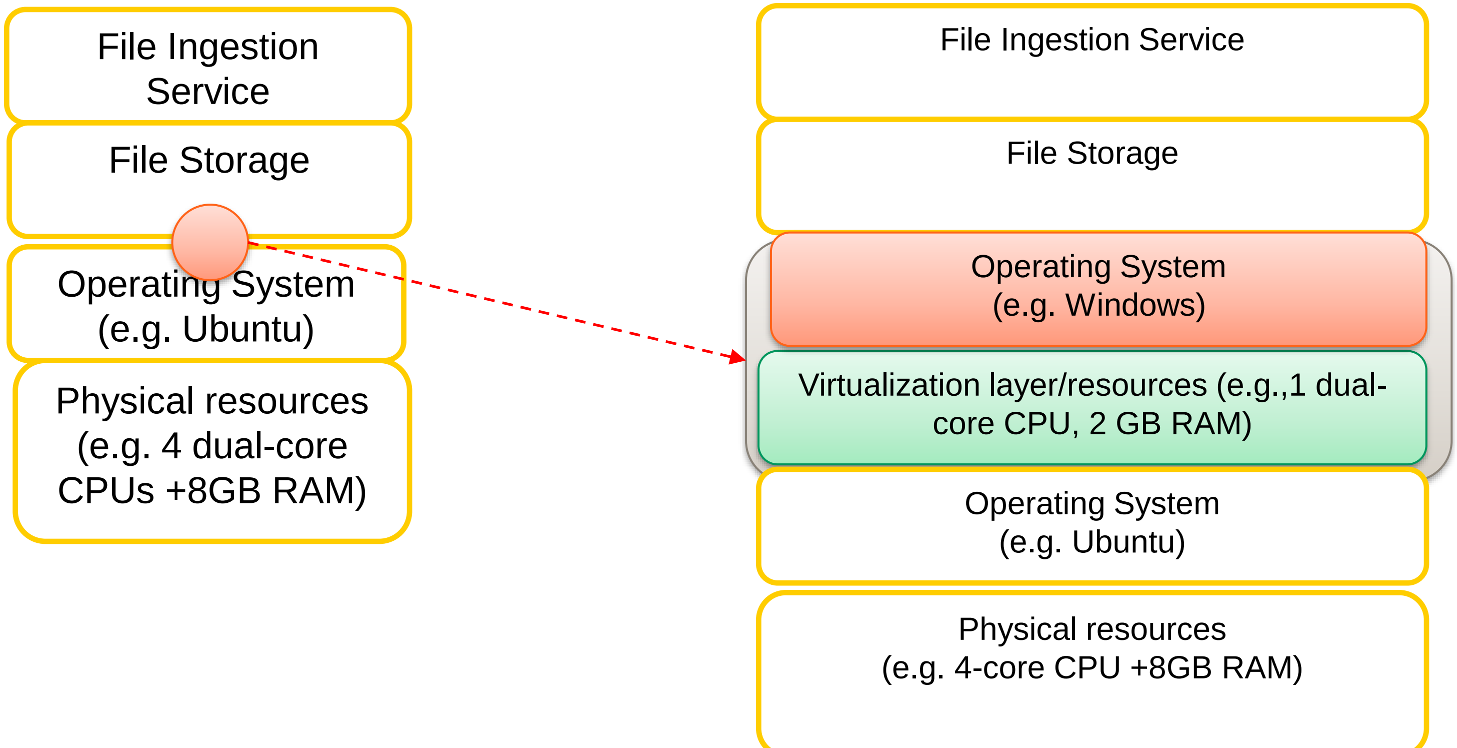
Service Model

- **Services offer well-defined interfaces for**
  - access resources: data, things, machines, and people
  - provide functions: ingestion, computation, sensing, analytics, 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 and cloud infrastructures

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, memory, I/O, 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

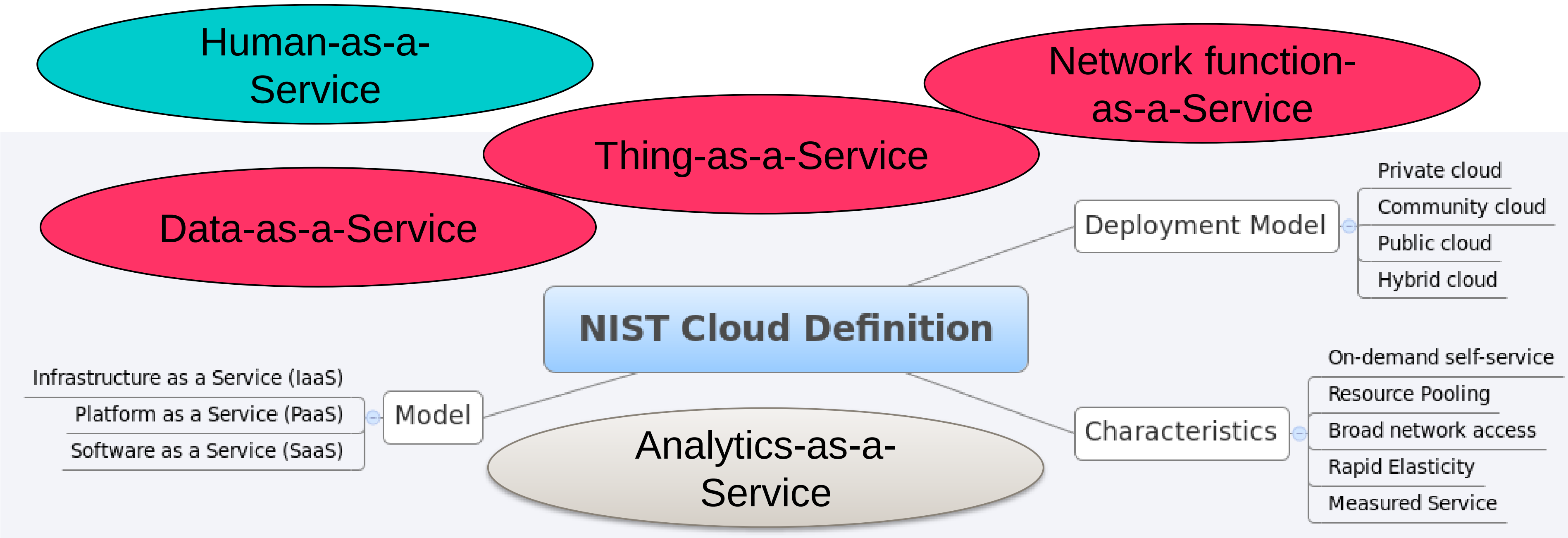
Cloud Computing

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



Cloud computing principles

- **“Cloud”**
  - not just datacenters 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!

Compute resource virtualization technologies

- **Physical compute resources for big data platforms**
  - Individual physical hosts/servers (CPU, memory, I/O)
  - Clusters 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.)*

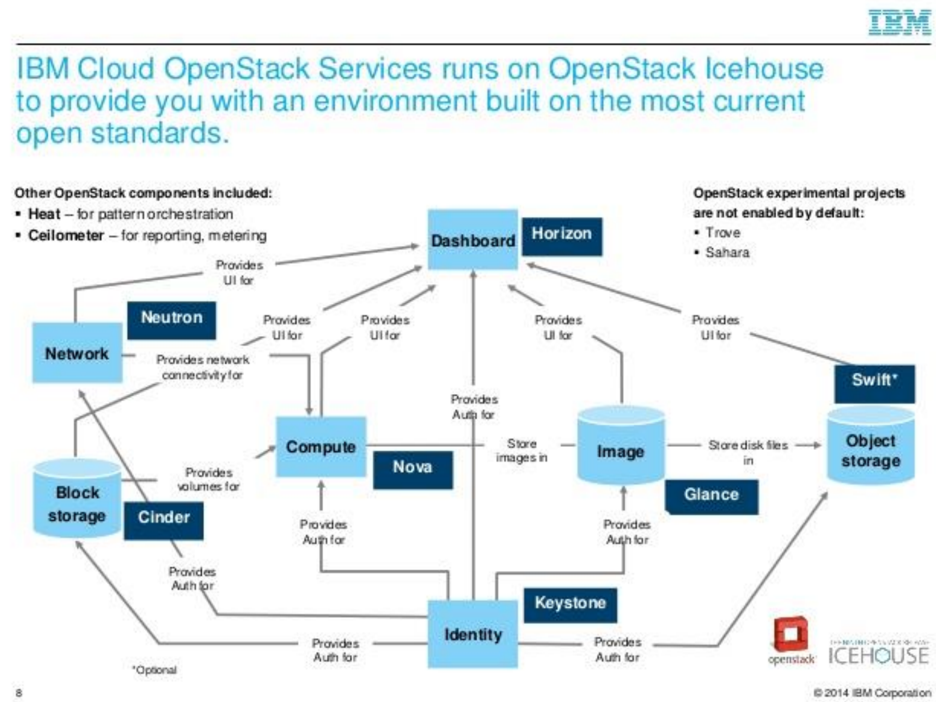
Virtual infrastructural resources for big data platforms

- **For big data platforms: we leverage clusters/infrastructures of VMs/containers**
  - 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

12

Example: OpenStack

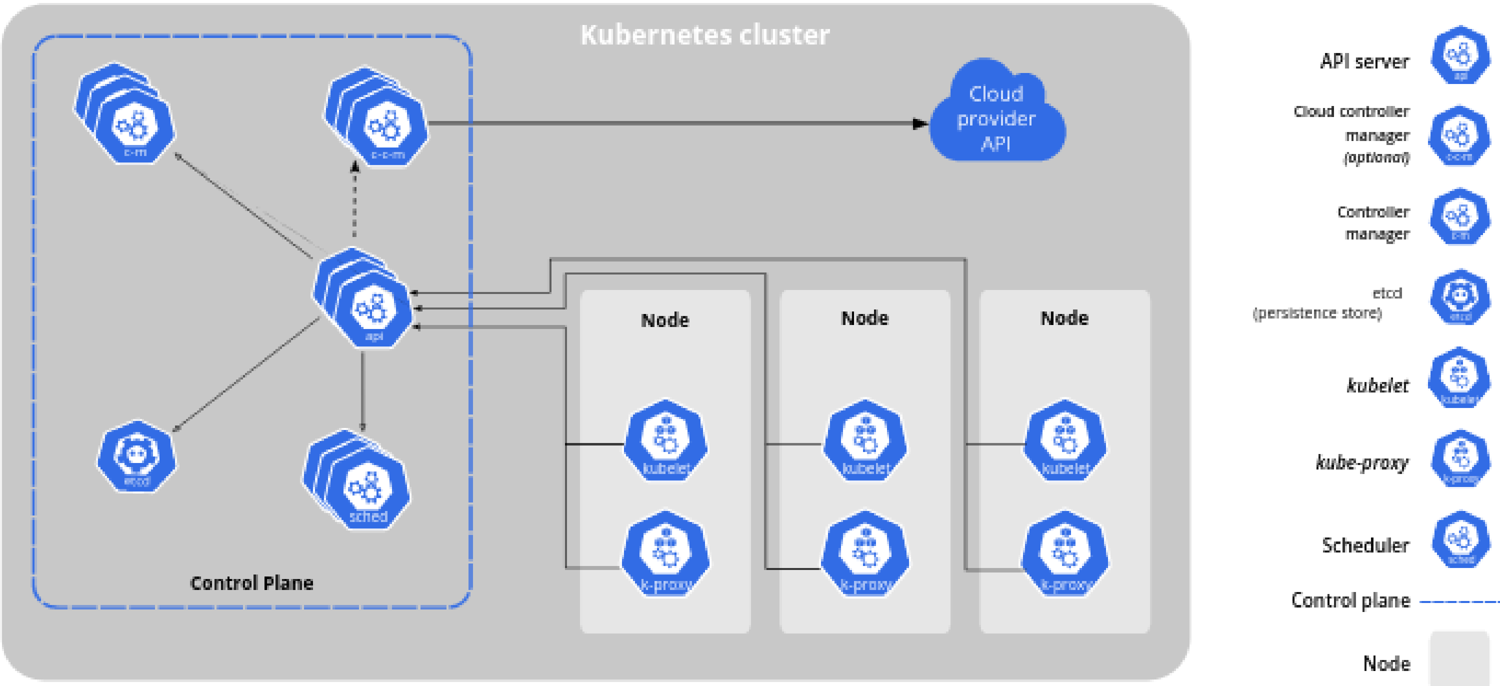
- **A Big Data Platform can be built based on OpenStack (or similar)-based data center**



Source: http://www.slideshare.net/OpenStack_Online/ibm-cloud-open-stack-services

Example: Kubernetes

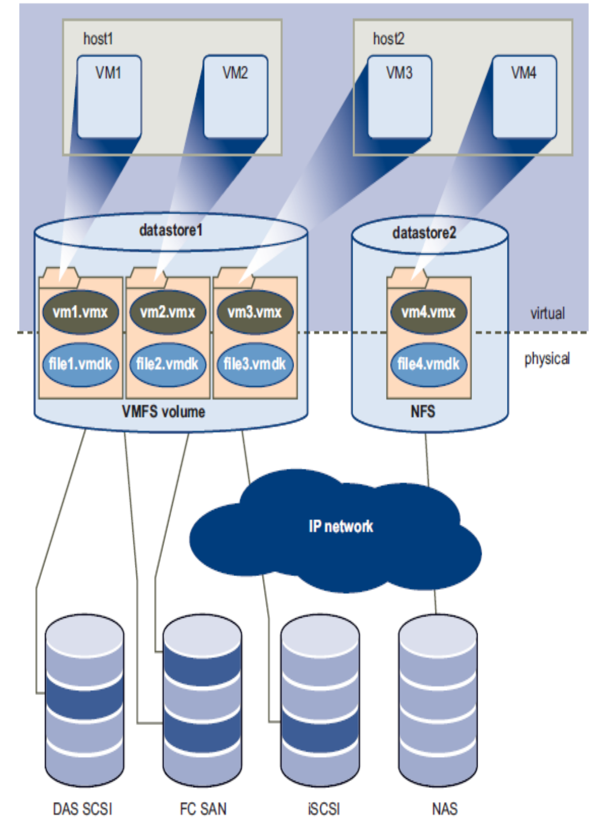
- **Support Docker, rkt, runc, etc.**



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



Source: https://www.vmware.com/pdf/vi_architecture_wp.pdf

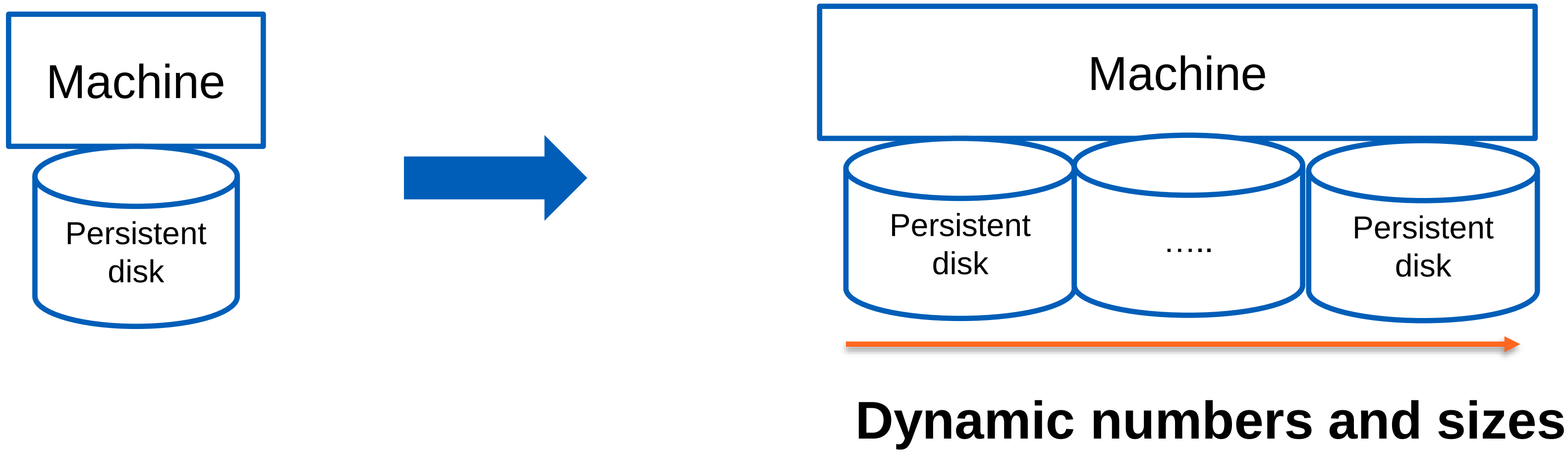
#1: Enabling managed services

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**

On-demand storage provisioning

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



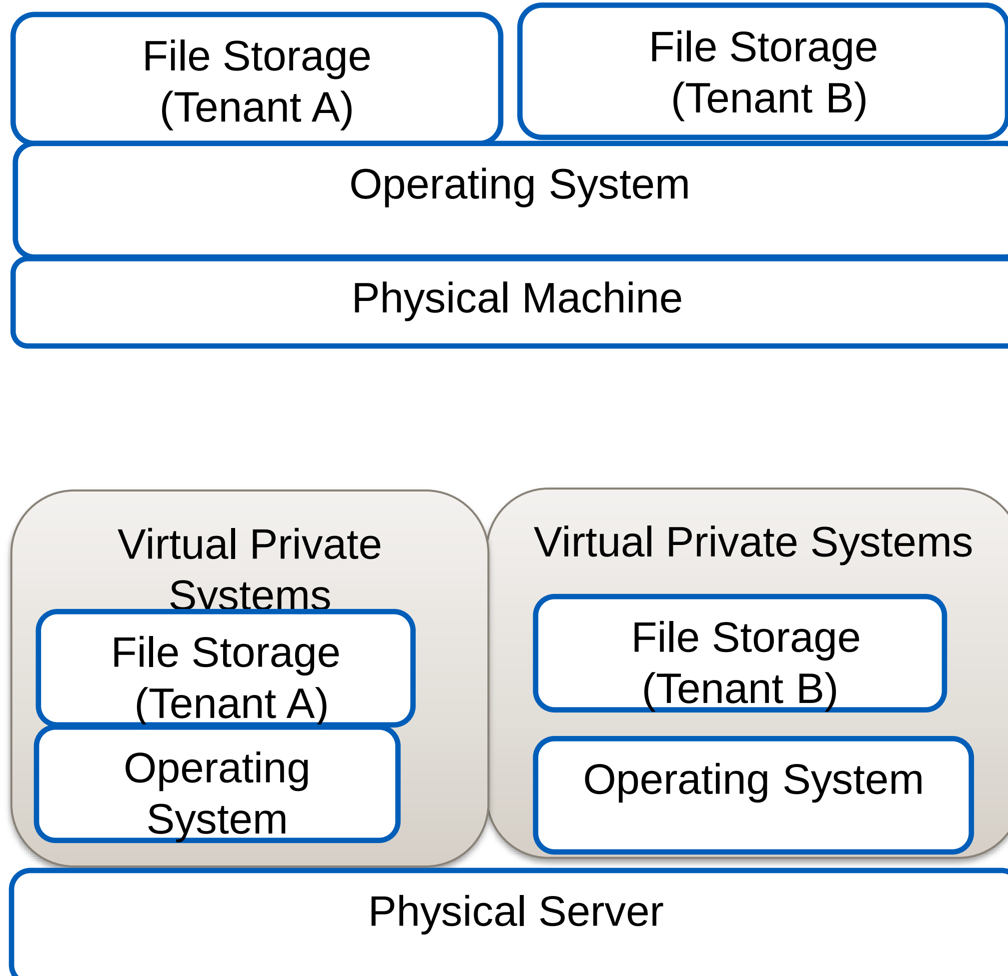
- **Examples of dynamic configurations:**
  - Google persistent disks: can have 128 disks, each persistent disk with 64GB (and with 1GB capacity increment)

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 analysis job with 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 their data!**



#2: Achieving fault tolerance, performance and elasticity

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

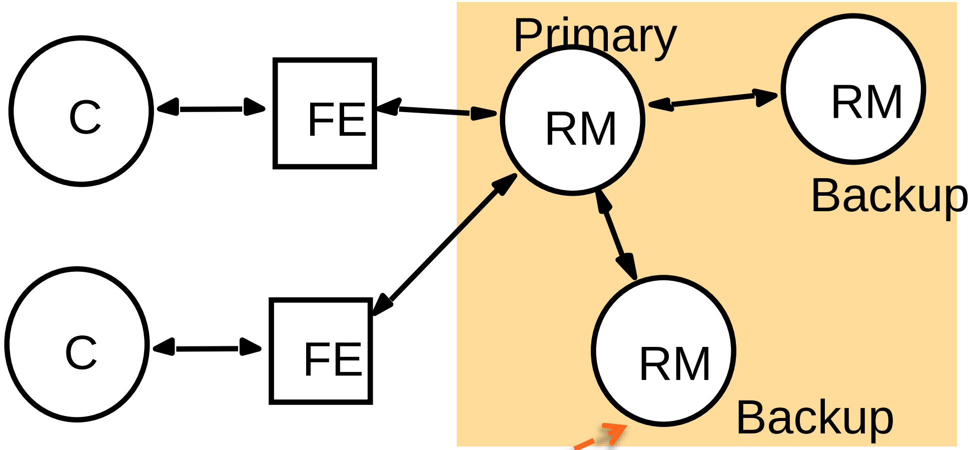


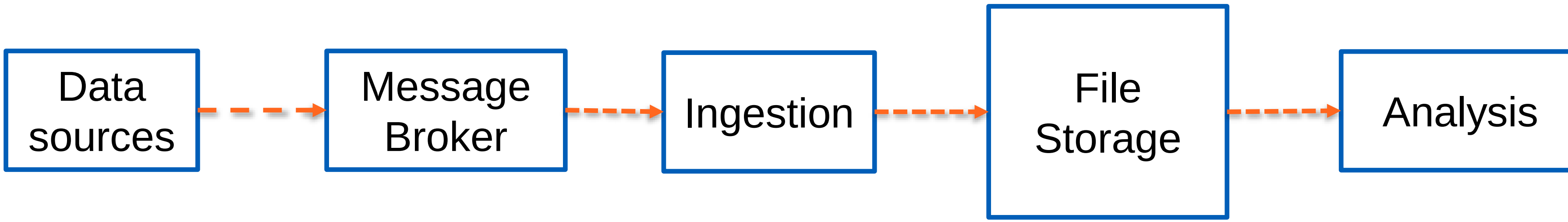
Figure source: Coulouris, Dollimore, Kindberg and Blair, Distributed Systems: Concepts and Design Edn. 5

Easy to deploy, globalize, manage and replace RM using cloud resources

High availability and performance

- **Cost/optimization**
  - elasticity, hot deployment, etc.
  - cloud bursting (combining private + 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

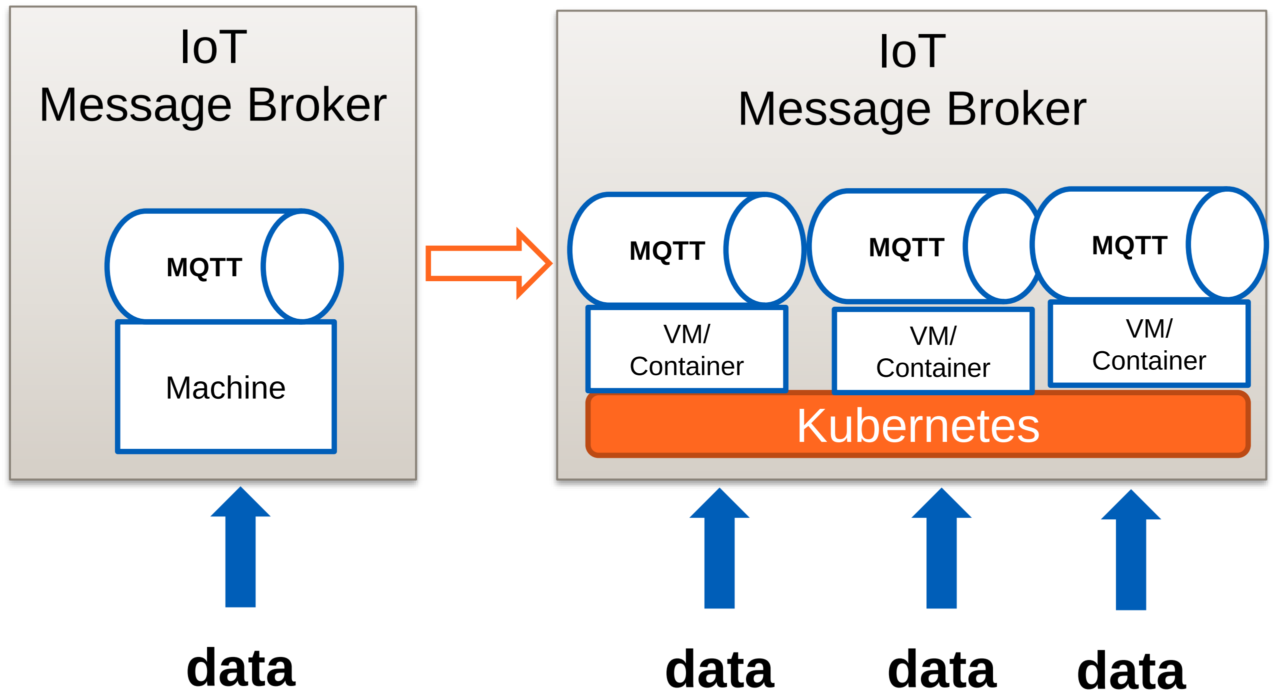


- **Scaling**
  - disk spaces for file storage
  - resources for data ingestion
  - resources for data analysis

Happen at different times and location

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

Monitoring | Jobs | VM Instances | Configuration | Web Interfaces

Name	Role	
thebasecluster-m	Master	SSH
thebasecluster-w-0	Worker	
thebasecluster-w-1	Worker	
thebasecluster-w-2	Worker	
thebasecluster-w-3	Worker	

Equivalent REST

4 nodes

On-demand change

Name	thebasecluster
Region	europe-north1
Zone	europe-north1-a
Autoscaling	Off
Scheduled deletion	Off
Enhanced flexibility mode	Off
Master node	Standard (1 master, N workers)
Machine type	n1-standard-2 (2 vCPU, 7.50 GB memory)
Primary disk type	pd-standard
Primary disk size	500 GB
Worker nodes	6
Machine type	n1-standard-1 (1 vCPU, 3.75 GB memory)

Monitoring | Jobs | VM Instances | Configuration | Web Interfaces

Name	Role	
thebasecluster-m	Master	SSH
thebasecluster-w-0	Worker	
thebasecluster-w-1	Worker	
thebasecluster-w-2	Worker	
thebasecluster-w-3	Worker	
thebasecluster-w-4	Worker	
thebasecluster-w-5	Worker	

6 nodes

#3: Living in the world of Microservices & DevOps

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**

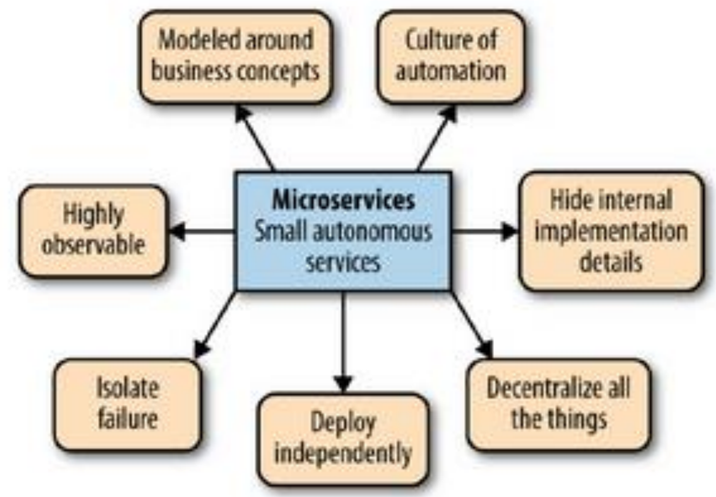


Figure source:Sam Newman, Building Microservices, 2015

Examples

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

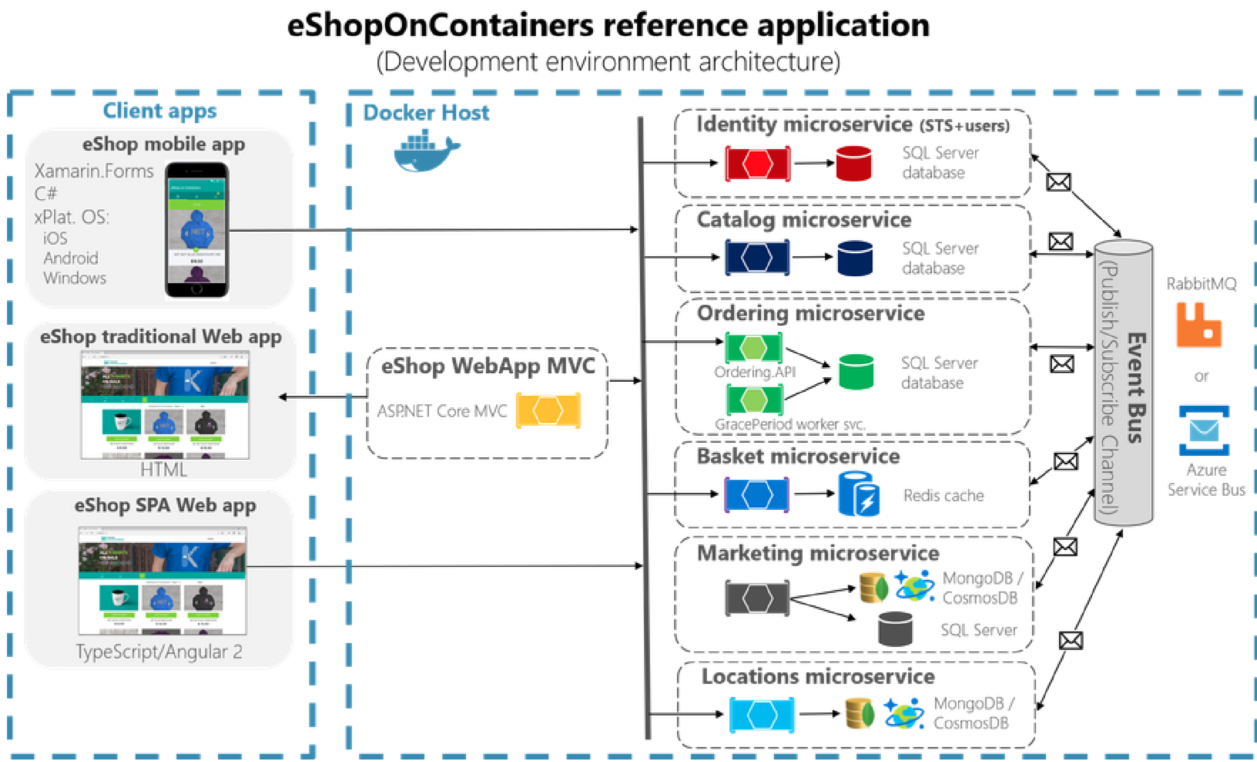


Figure source: https://blogs.msdn.microsoft.com/dotnet/2017/08/02/microservices-and-docker-containers-architecture-patterns-and-development-guidance/

DevOps

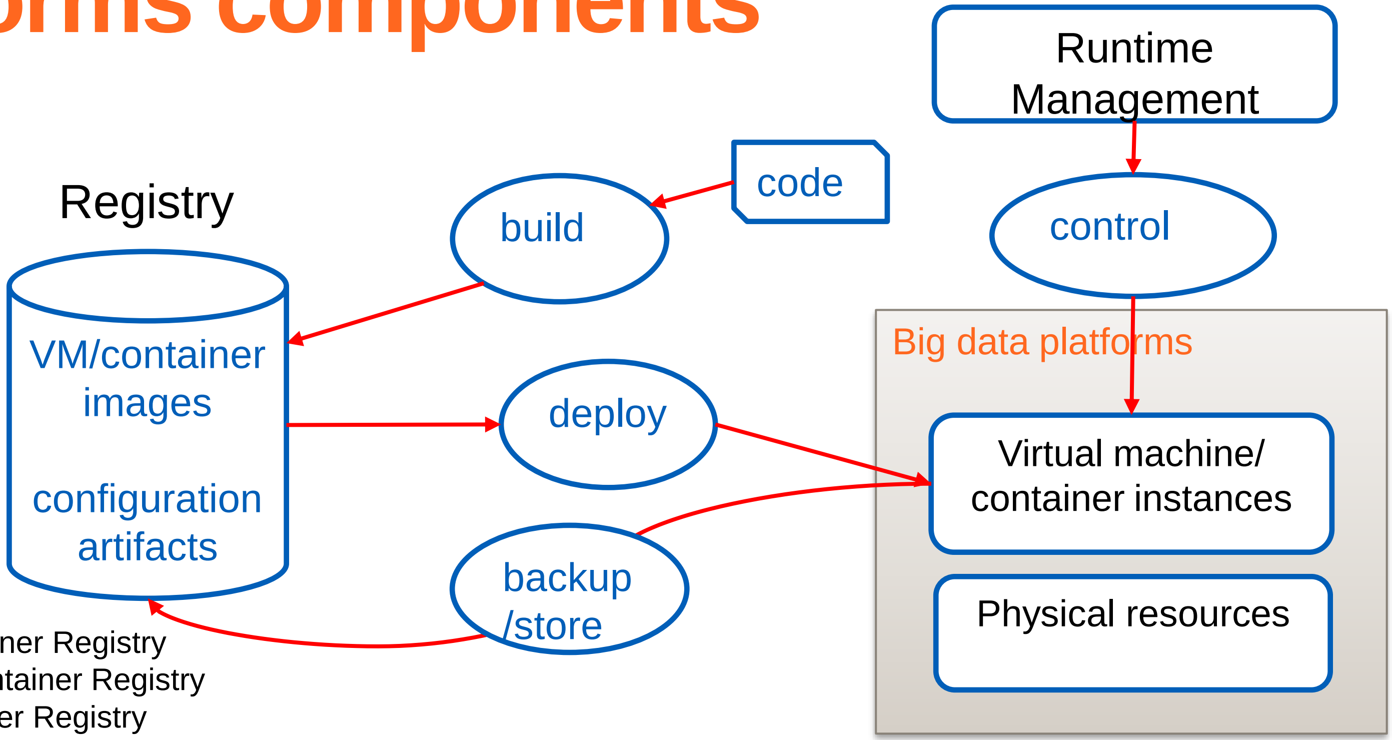
- Close the gap between development/test environment and real/production environments
- Simplify testing, emulating real environments, etc.

DevOps for big data platforms as well as big data platforms are part of software systems under DevOps

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

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

Provisioning and management of platforms components



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

But what about data?

Thanks!

Hong-Linh Truong
Department of Computer Science

rdsea.github.io