ENVRI

FAIR

# An introduction to Cloud computing

## Cloud computing and application development for research infrastructures


**Dr. Zhiming Zhao**

**Multiscale Networked Systems, University of Amsterdam,**

**LifeWatch ERIC, vLabs & Innovation Centre**

Email: z.zhao@uva.nl



ENVRI-FAIR has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824068

# Typical questions

"I **manage** a data center, I often **get many requests** for running workflows or big data applications, sometimes I can not handle all those requests"

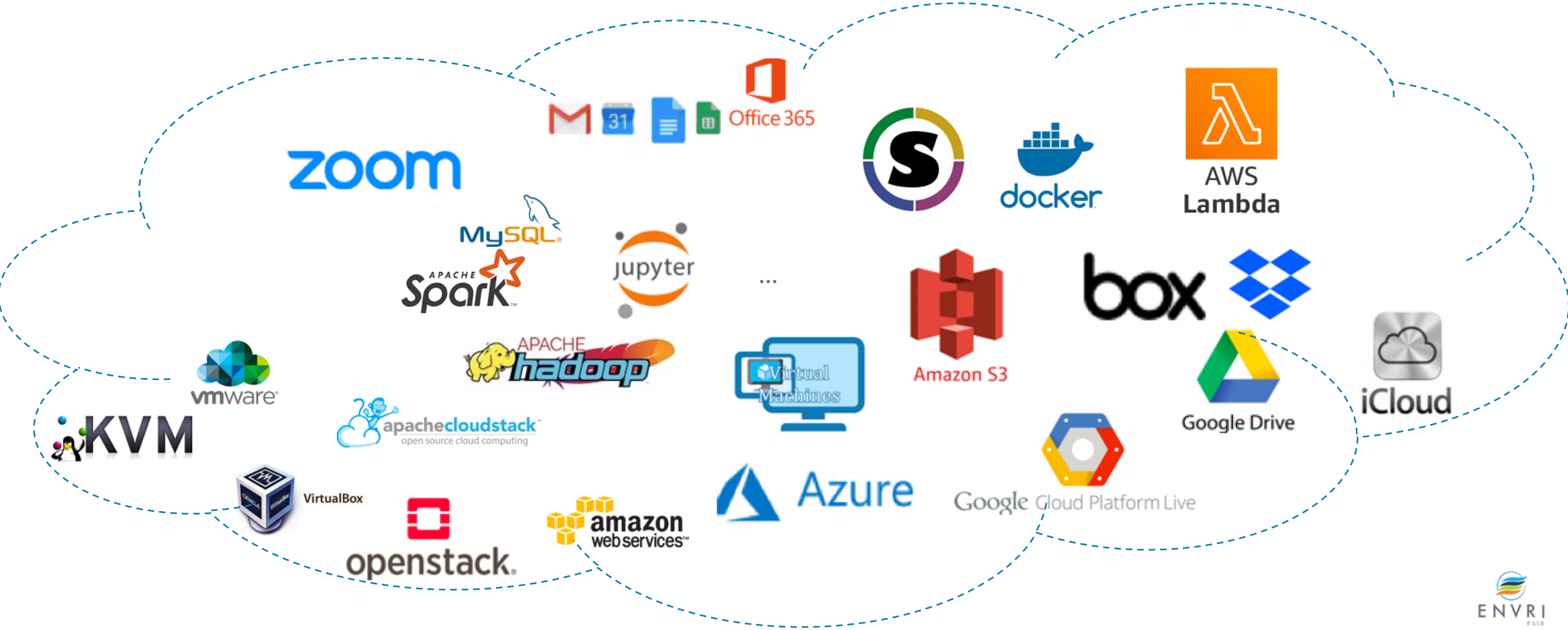
"I have problem to train **a deep neural network** using **large** data sets from **remote sources** on my own laptop"

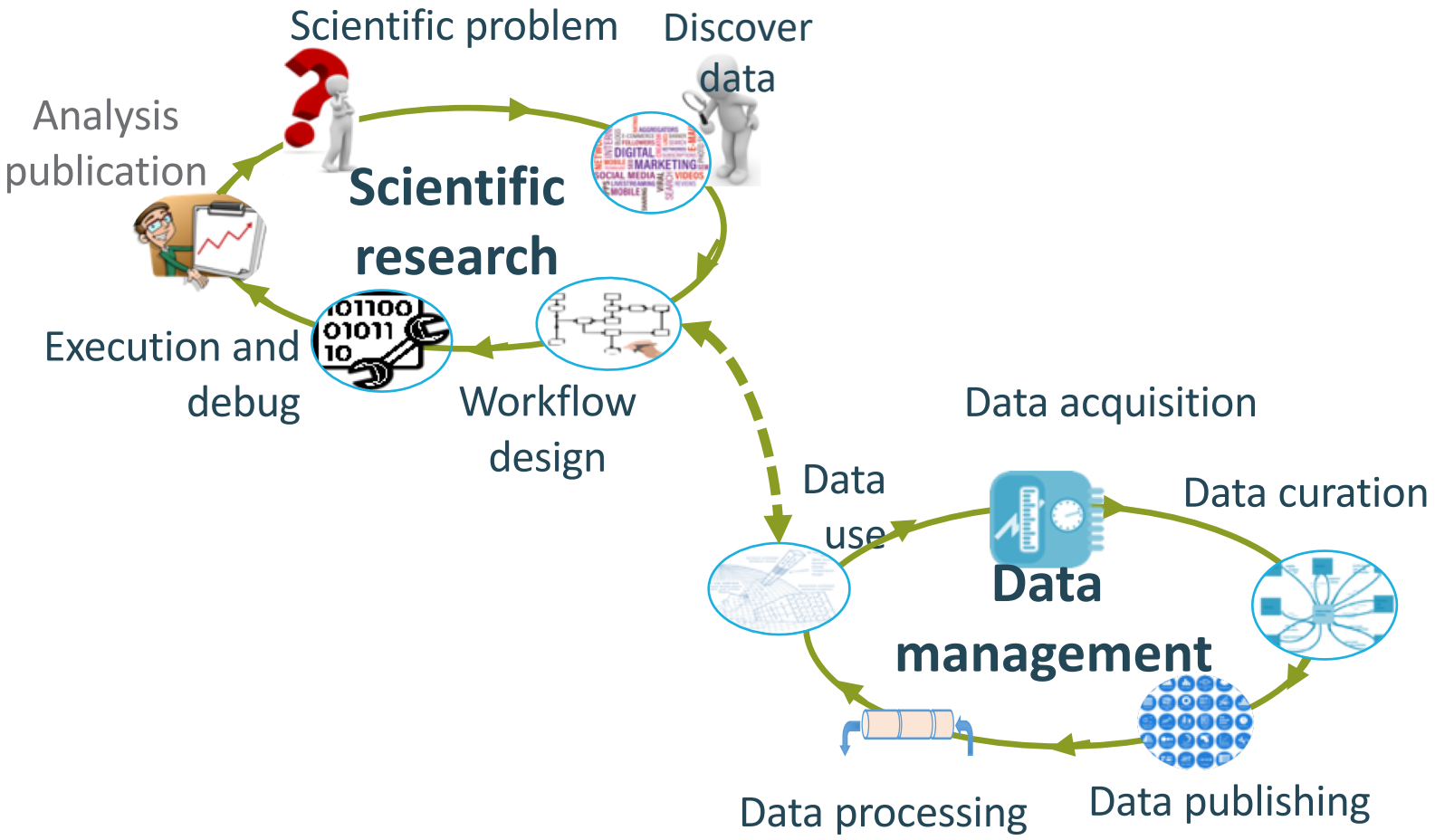
"I process the **observation** **data** in a station, sometimes the local sever is overloaded and I can not deliver data products in time"

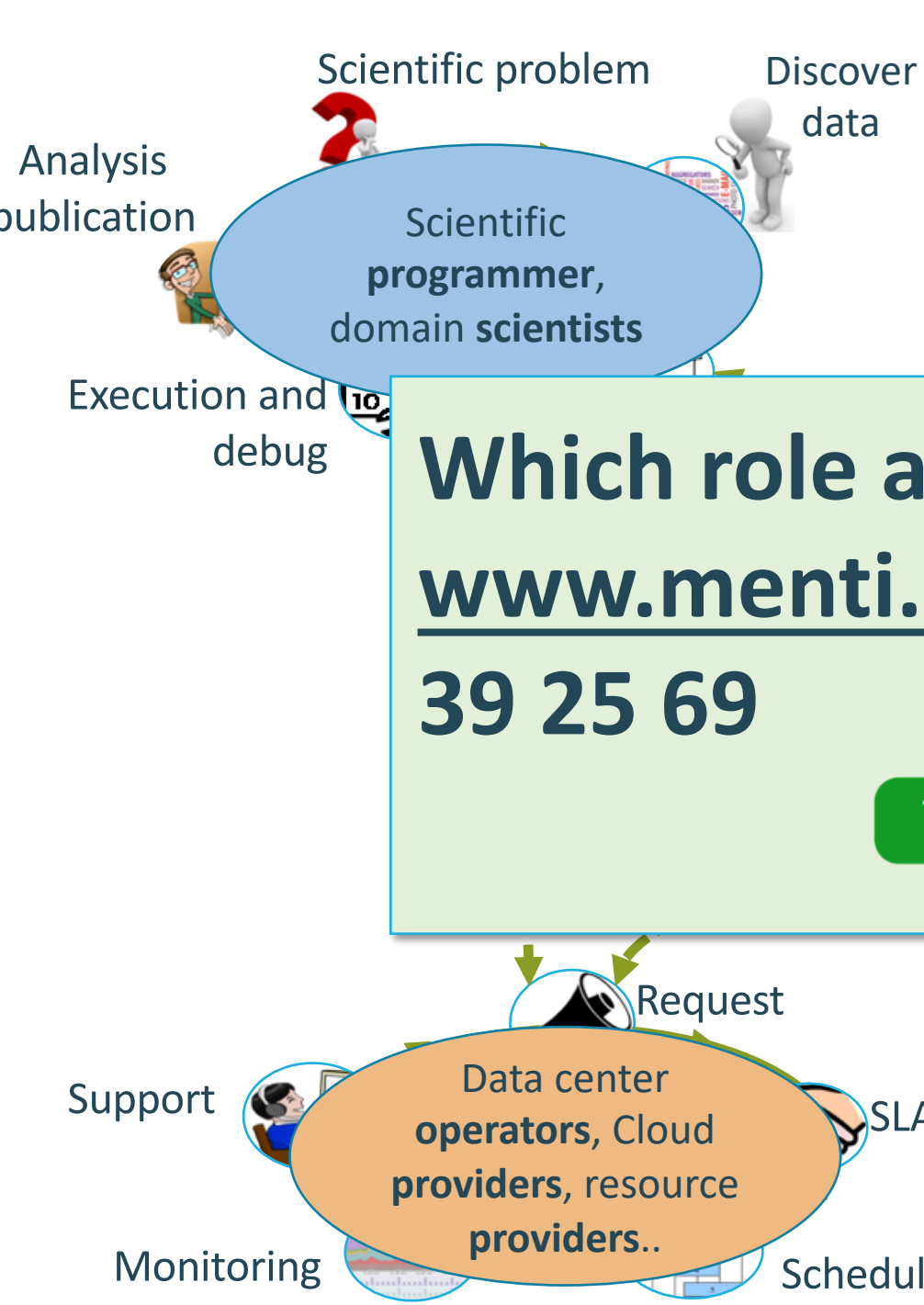
"the **scien** **workflo** colleague run on my local PC

...

# Cloud: services, technologies ....




Scientific problem
Discover data
Analysis publication
Scientific research
Execution and debug
Workflow design
Data acquisition
Data use
Data curation
Data management
Data processing
Data publishing




# Which role are you?

## www.menti.com

## 39 25 69

Virtual research environment, science gateway, virtual lab, problem solving environment, scientific workflow management systems etc.

VRE4EIC BlueBRIDGE everest ...

Research infrastructure, data infrastructure

EPOS EISCAT ICOS AnaEE SeaDataNet IAGOS SIOS ACTRiS LTER Europe is-enes Euro-Argo ...

Clouds, e-Infrastructures

amazon web services egi EUDAT ...

# Outline

1. What is Cloud computing and the Cloud?
2. How does the Cloud work?
3. How to use the Cloud?
   a) Use basic infrastructure service
   b) Run a local application in Cloud
   c) Run complex applications in Cloud
   d) Infrastructure automation for operating online services
4. Discussion
5. Summary

# Outline

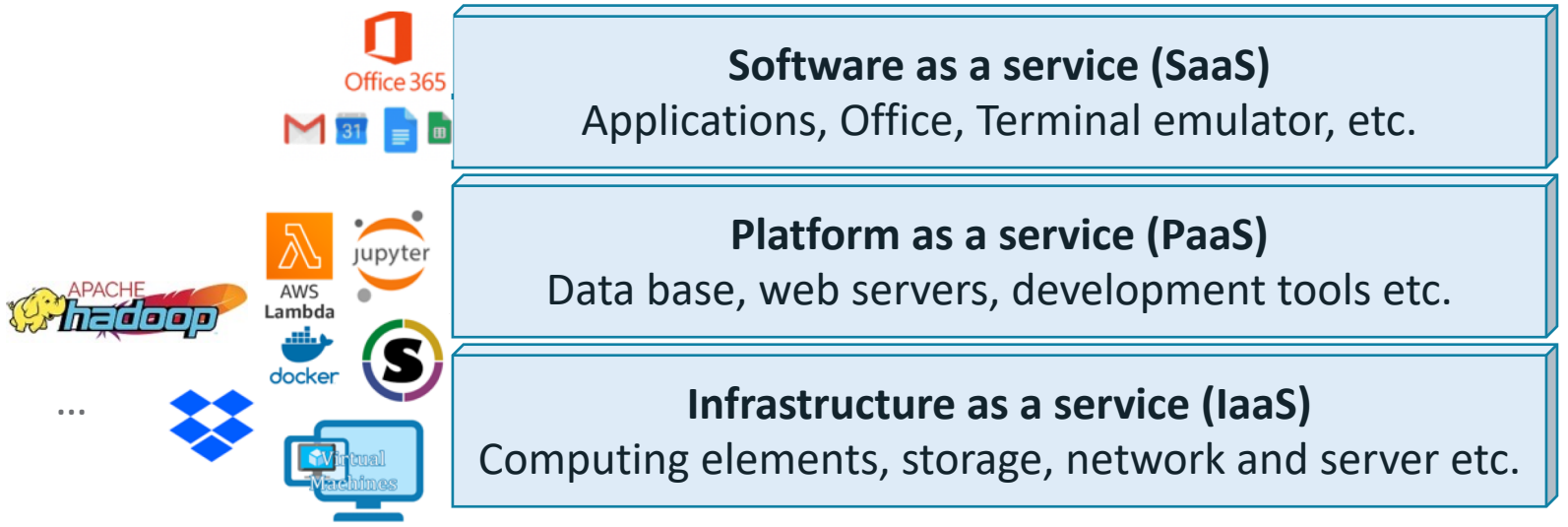
1. **What is Cloud computing and the Cloud?**
2. How does the Cloud work?
3. How to use the Cloud?
   a) Use basic infrastructure service
   b) Run a local application in Cloud
   c) Run complex applications in Cloud
   d) Infrastructure automation for operating online services
4. Discussion
5. Summary

# 1.1 What is Cloud computing?

- **on-demand availability** of **computer system resources**, especially storage and computing power, **without direct active management** by the user. [wiki]
- a model for enabling **ubiquitous, convenient, on-demand network access** to a **shared pool** of **configurable computing resources** that can be **rapidly provisioned** and **released** with minimal management effort or service provider interaction. [NIST, 2010]
- the **delivery of computing services**... You typically **pay only for cloud services you use**, helping you **lower** your operating costs, run your infrastructure more **efficiently**, and **scale** as your business needs change. [Azure]
- **the on-demand delivery of IT resources** over the Internet with **pay-as-you-go pricing,** on an **as-needed basis** from a cloud provider [AWS]

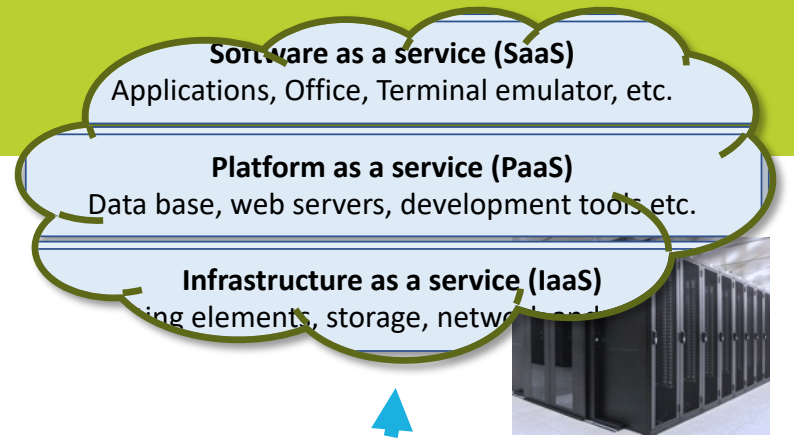
# 1.2 From a user perspective

- **Services of**
  - Computing power
  - Storage
  - Network
  - Data base
  - Server (e.g. Web server)
  - Word/Excel/PPT etc.
  - …



# 1.2 From a user perspective

- Services of resources
- **Access via Internet**
  - Using internet
  - Access from anywhere
  - Web portal

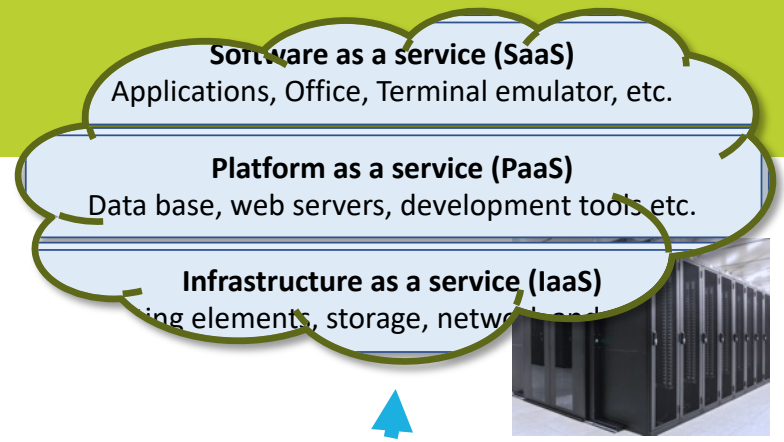


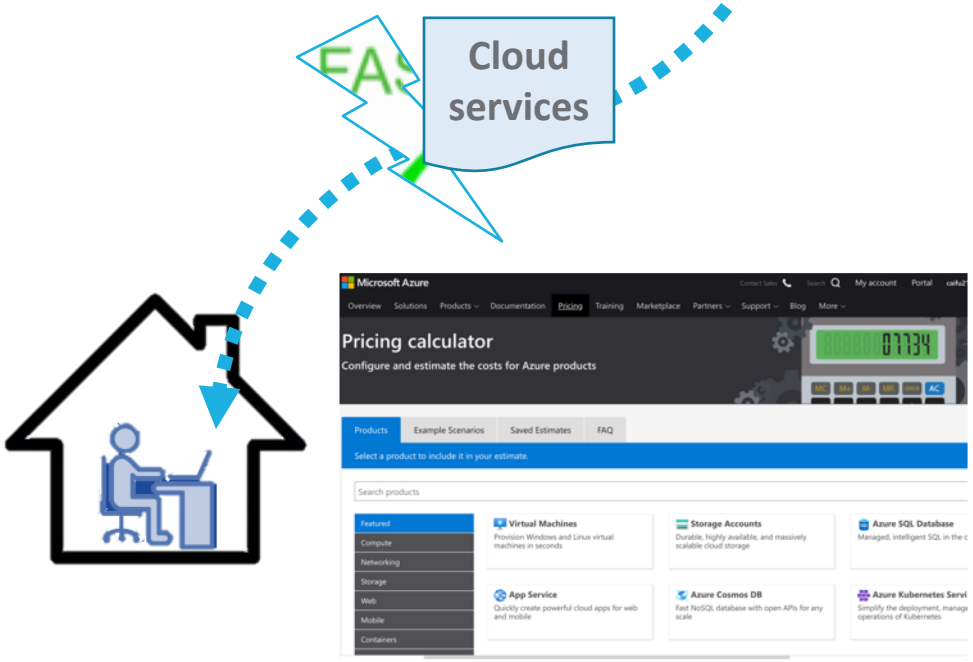

# 1.2 From a user perspective

- Services of resources
- Access via Internet
- **On demand provisioning**

# 1.2 From a user perspective

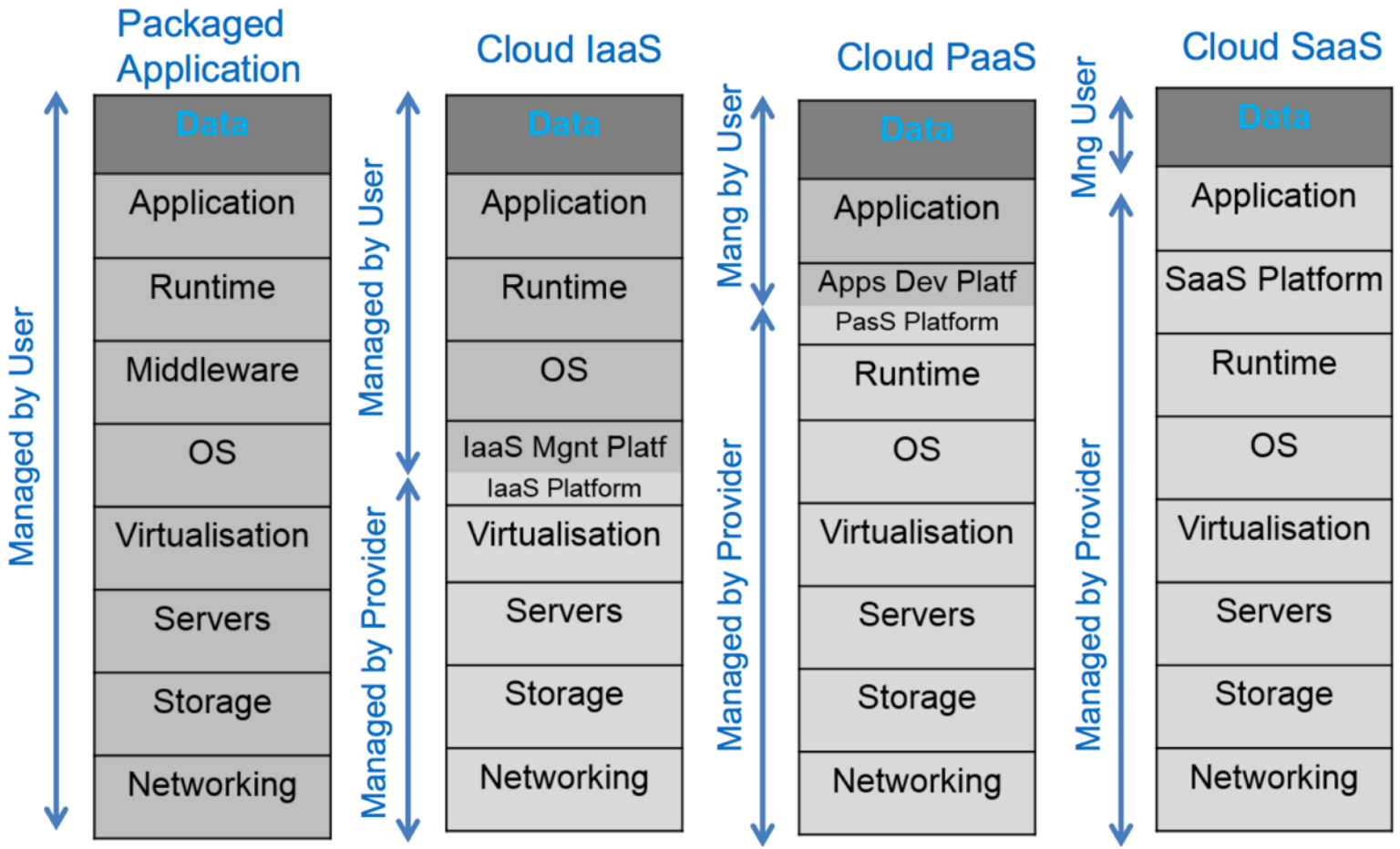
- Services of resources
- Access via Internet
- On demand provisioning
- **Flexible price model**
  - Pay per use
  - Pay as you go
  - Advanced reserved
  - Subscription based



# 1.3 From the infrastructure point of view

- Manage everything by yourself or outsource them?
  - Advantages
  - Disadvantages
- Options
  - IaaS
  - PaaS
  - SaaS
  - …

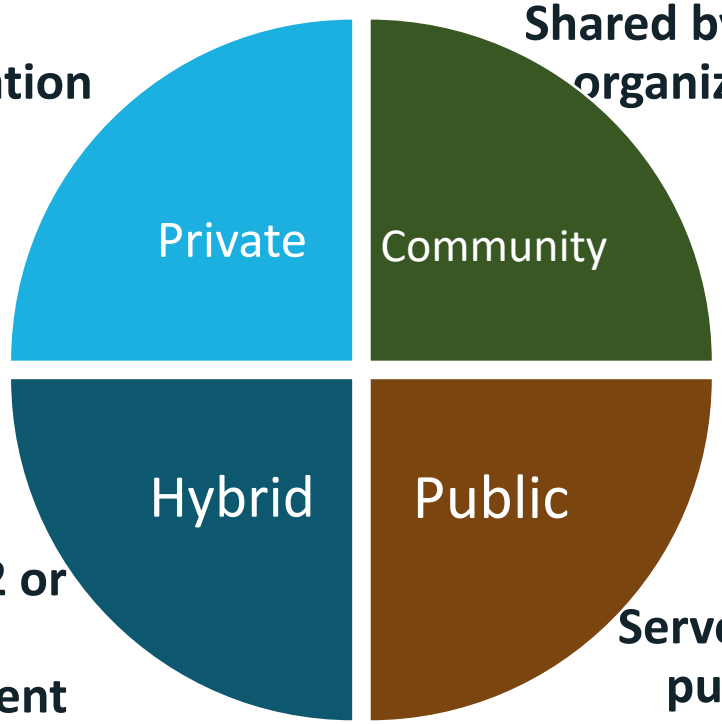


# 1.4 Different types of Cloud

- Private cloud
- Public cloud
- Hybrid cloud
- Community cloud

**Single organization**

Private used data center, e.g. for single institute




**Shared by several organizations**

EGI ExoGENI

...

**Mix of 2 or more different types.**

**Serve public use**

Google Cloud Platform aws Microsoft Azure ...

See more examples from RIs

# 1.5 e-Infrastructures

- High performance c
  - Super compute
  - Clusters
  - ..
- Clouds
  - Infrastructure
  - Containers
  - Platform as a s
  - Software as a s
  - ..
- Storage
  - Cloud storage,
- Advanced network
  - Light paths
  - Software defin
  - ..

**e-Infrastructures** address the needs of European researchers for digital services in terms of **networking, computing and data management**.





# 1.6 Discussion

- Question: What Cloud services have you ever used?

**www.menti.com**

**39 25 69**

# Outline

1. What is Cloud computing and the Cloud?
2. **How does the Cloud work?**
3. How to use the Cloud?
   a) Use basic infrastructure service
   b) Run a local application in Cloud
   c) Run complex applications in Cloud
   d) Infrastructure automation for operating online services
4. Discussion
5. Summary

# 2.1 Inside a Cloud data center

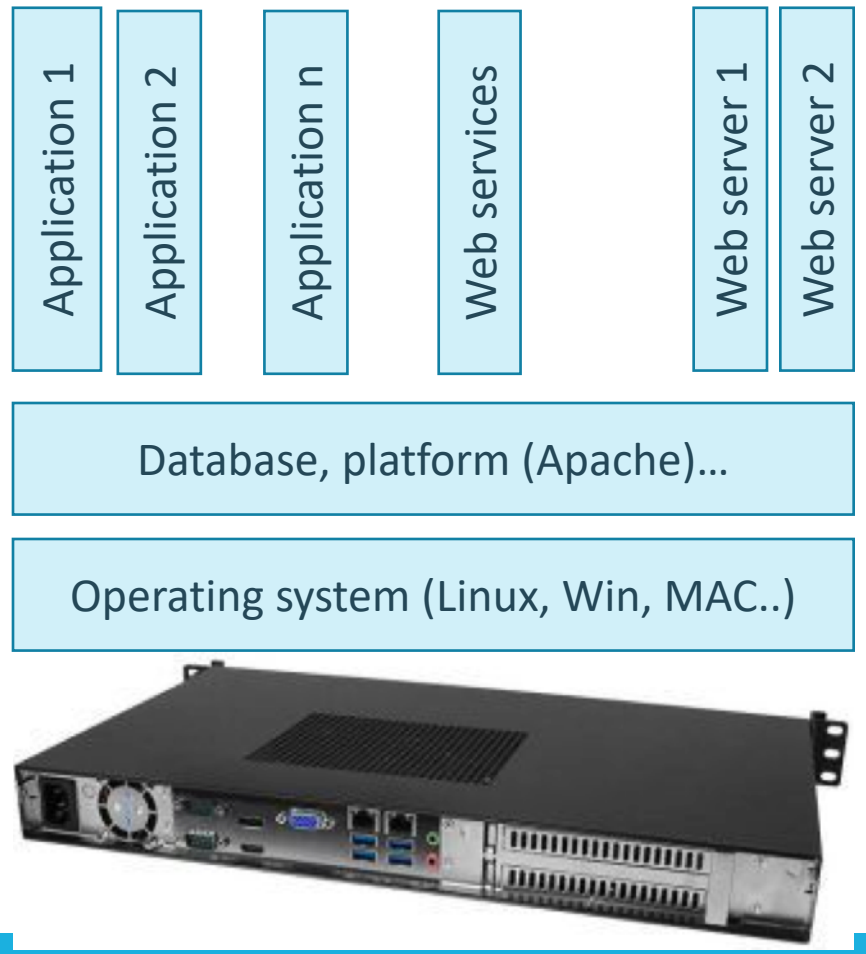
Data Center

- How to allocate different size machines from one host?
- Key concepts:
  - Virtual machines
  - Containers

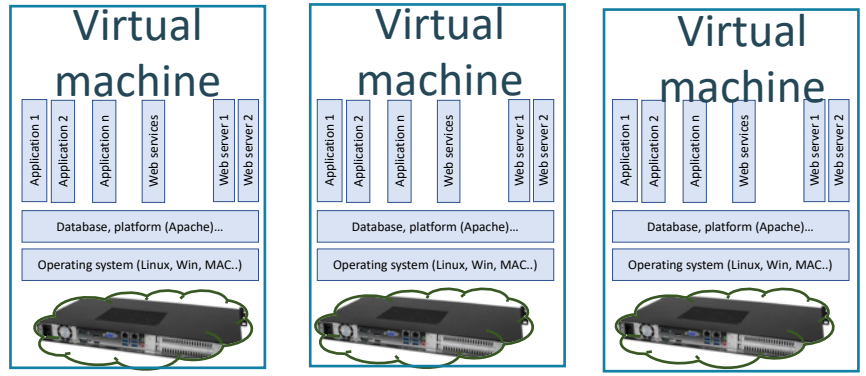
# 2.2 Virtual machine

**A virtual machine (VM):**

- A software can emulate the behavior of a real computer
- Contains hardware abstraction, OS kernel, library, file systems and etc.. The file representation is called VM **image.**

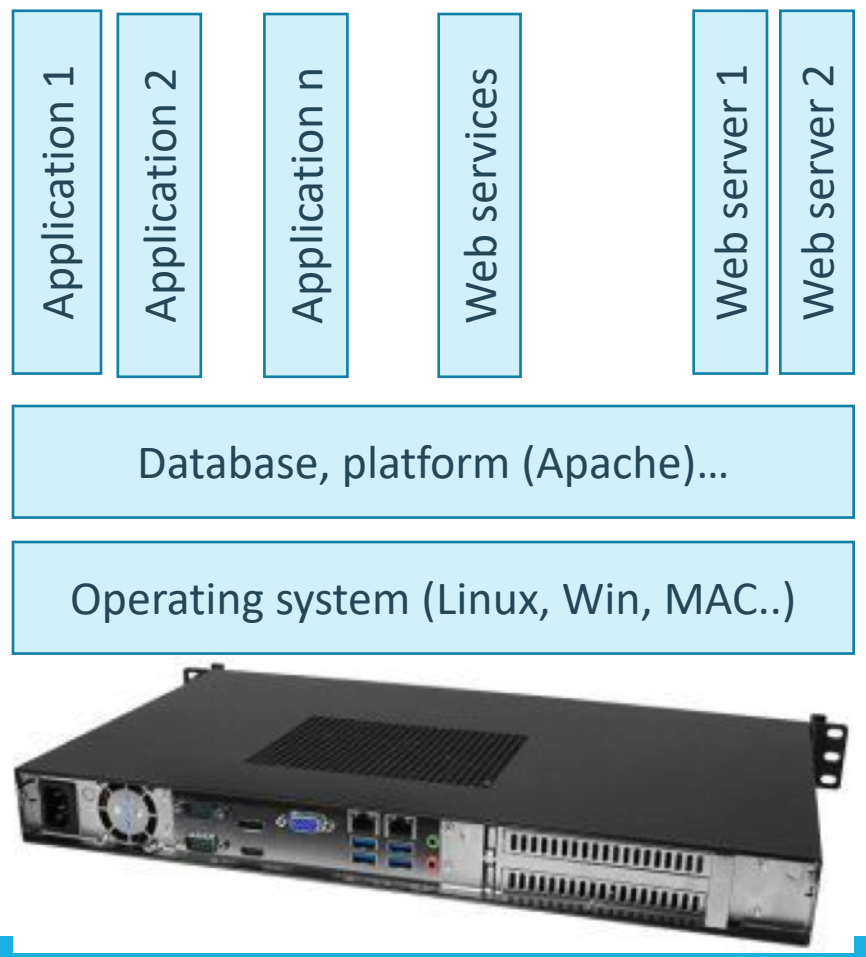



**Virtualization**

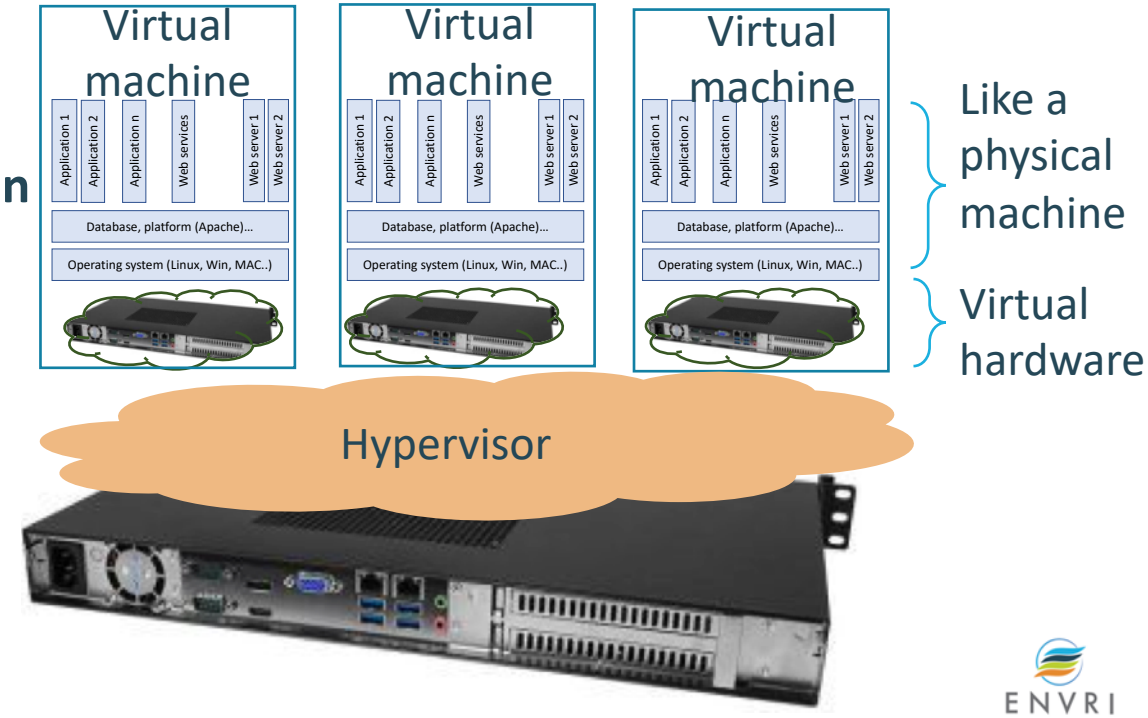



# 2.3 Hypervisor (Virtual Machine Monitor)

**Hypervisor: software platform can run Virtual Machines.**



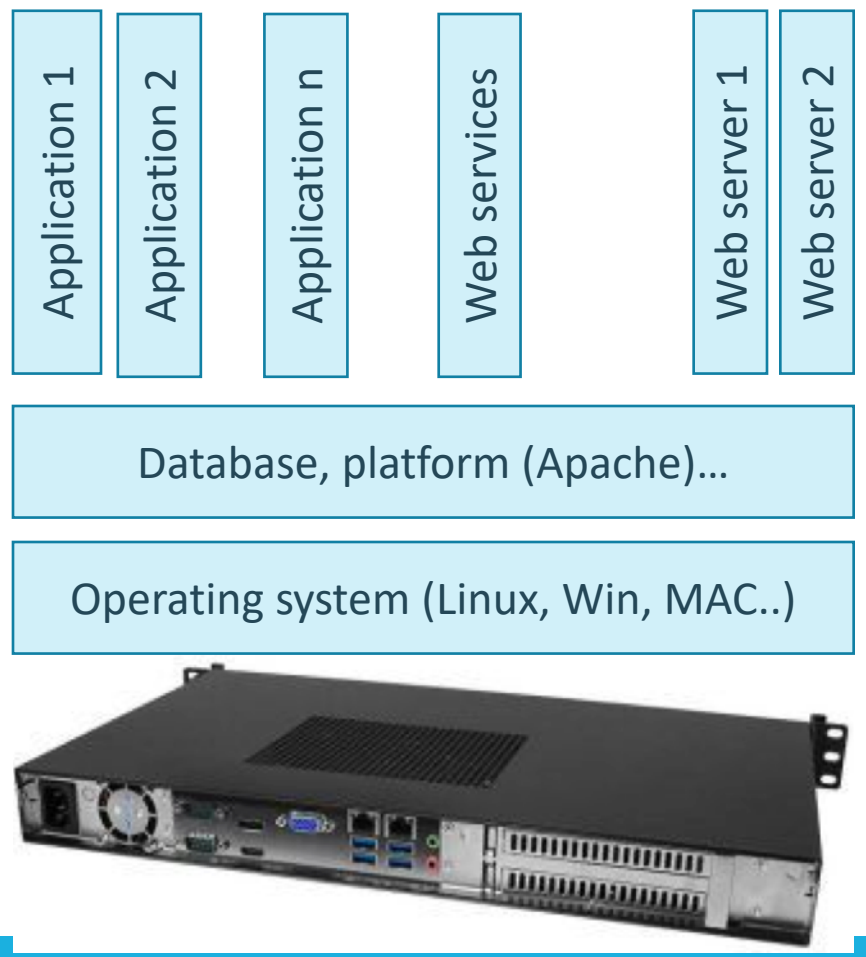

**Virtualization**



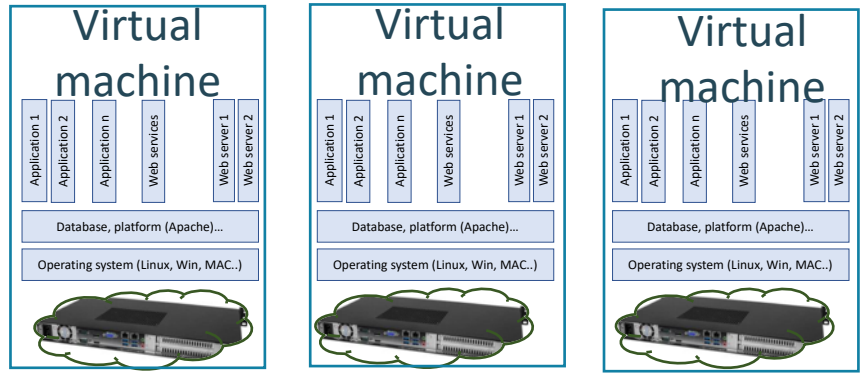
# 2.3 Hypervisor (Virtual Machine Monitor)

**Hypervisor: software platform can run Virtual Machines.**

**Type 1: directly on top of the hardware, e.g. VMware**

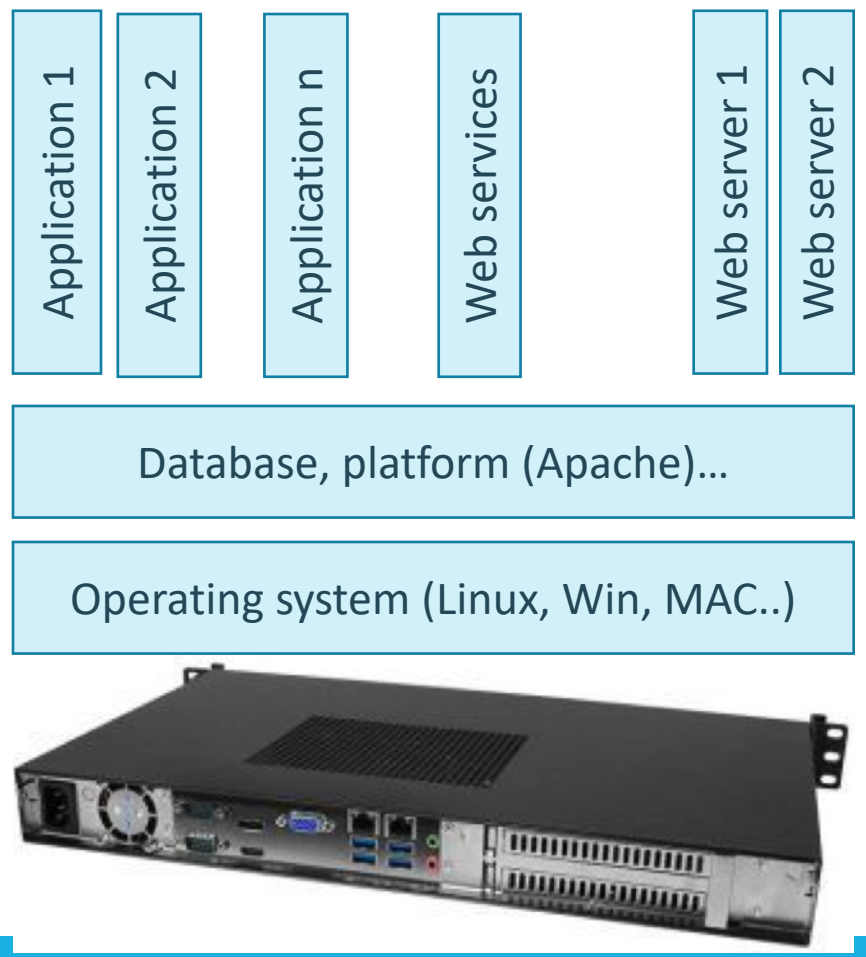



**Virtualization**

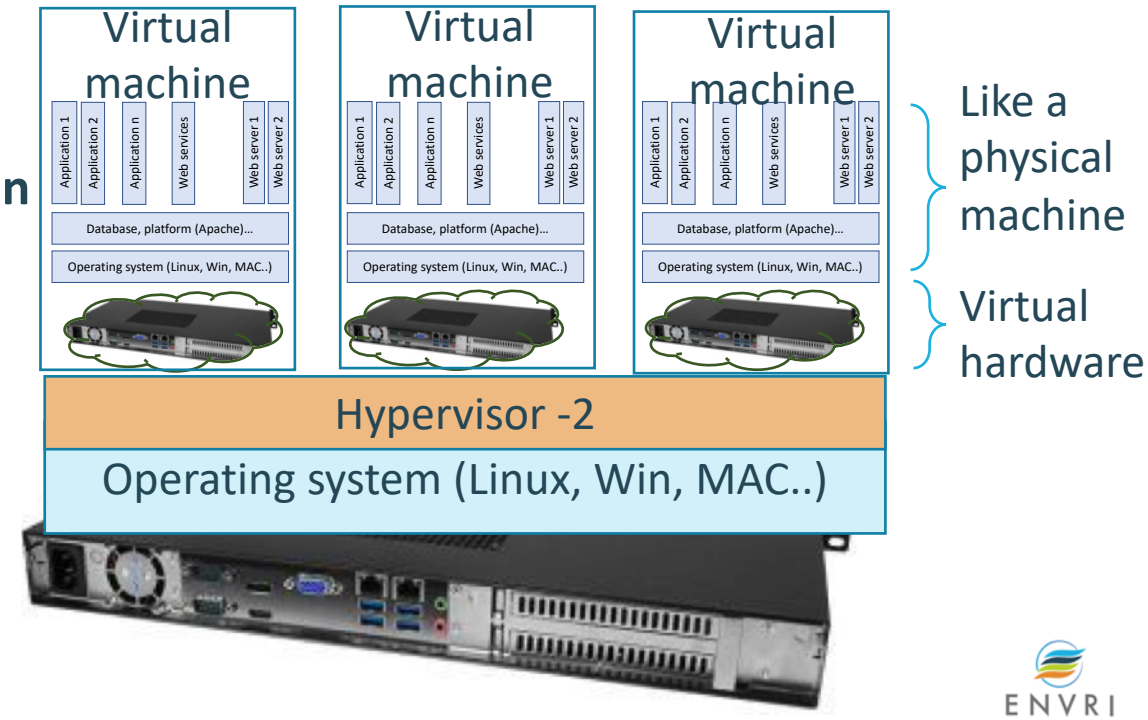


# 2.3 Hypervisor (Virtual Machine Monitor)

**Hypervisor: software platform can run Virtual Machines.**
**Type 1: directly on top of the hardware, e.g. VMware**
**Type 2: on top of the operating system, e.g. VirtualBox**



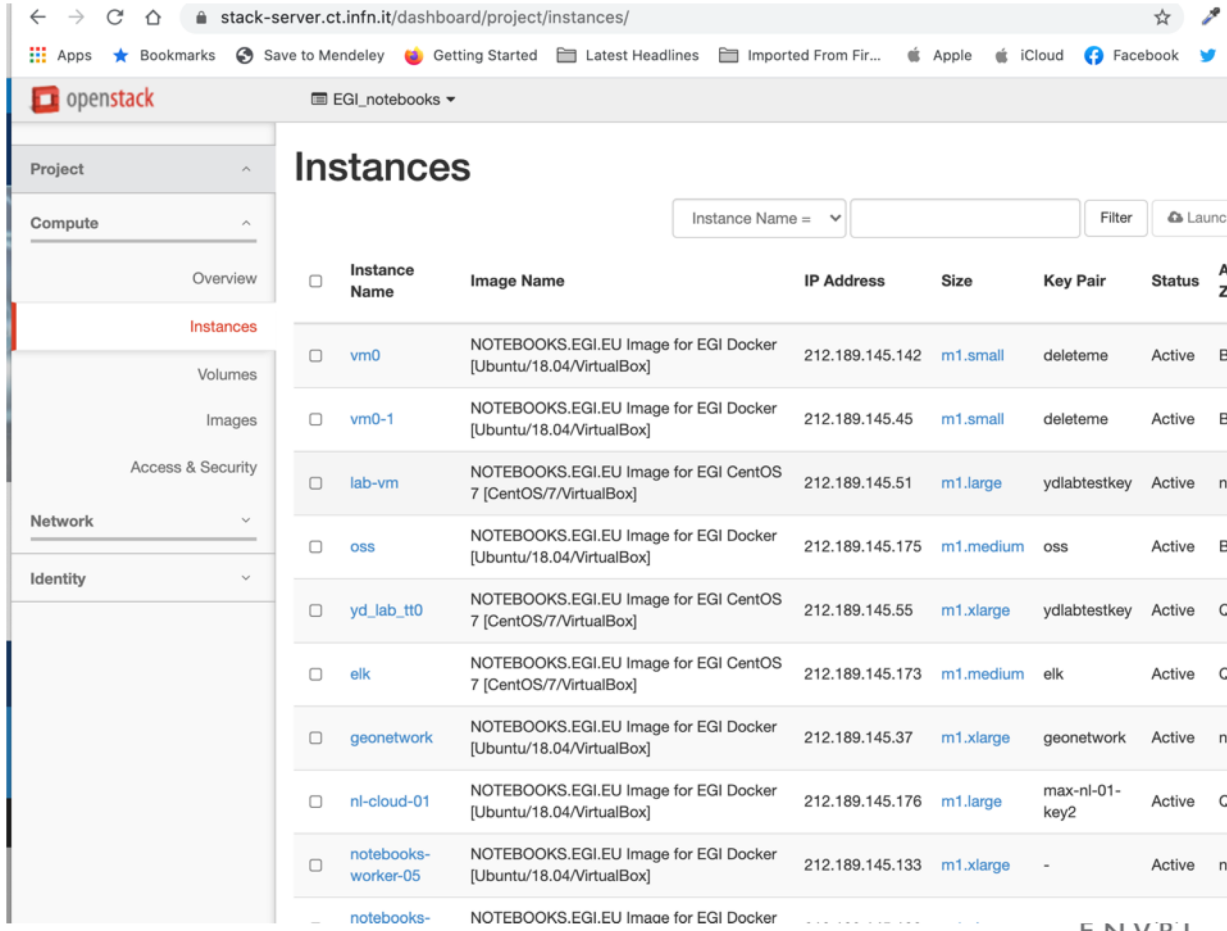

**Virtualization**



# 2.4 Cloud infrastructure service orchestrator

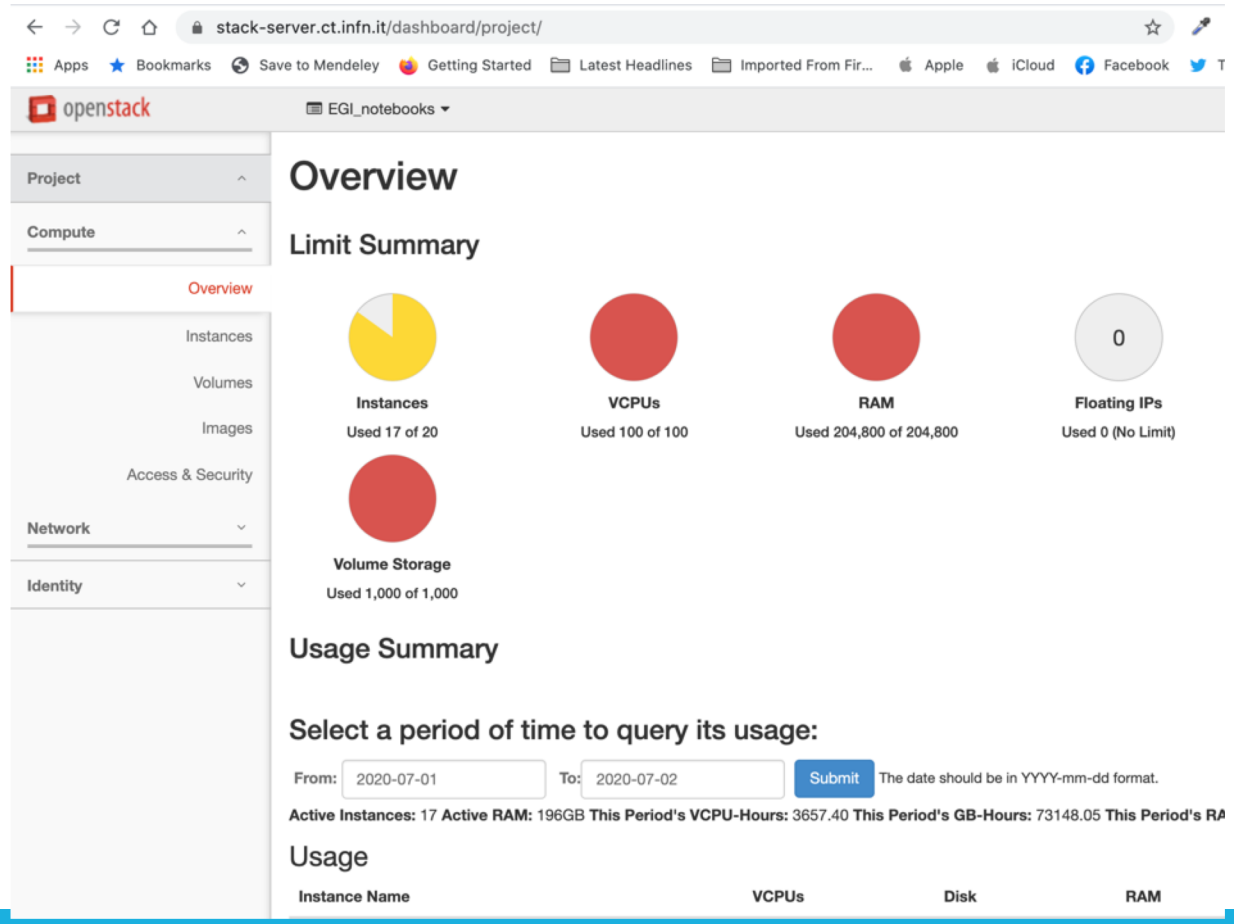
- Manages the lifecycle of resources
- User interface, e.g. dash board, resource selection/configuration/control
- Automate VM, Storage and network provisioning,
- ..

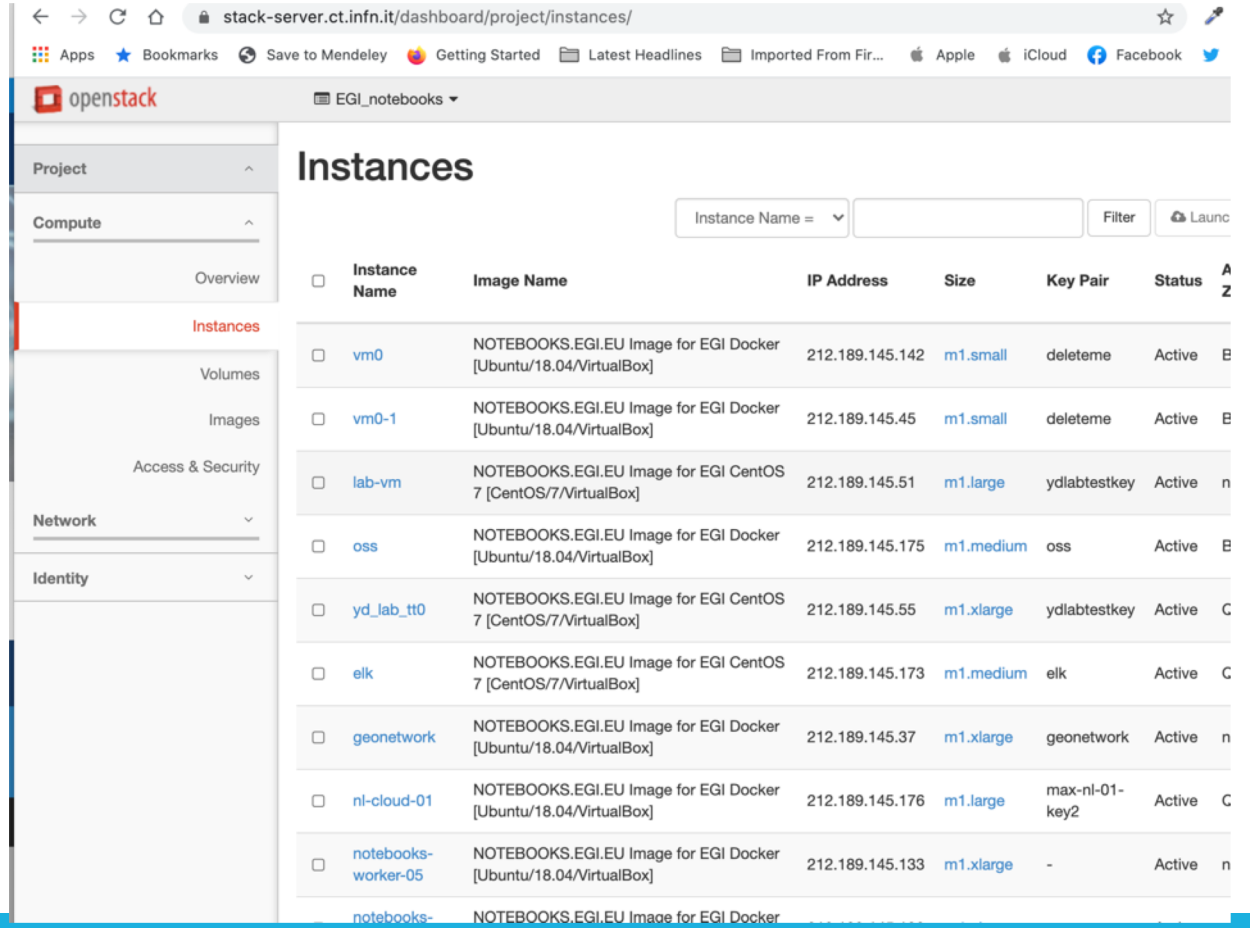
- Examples: Open Stack, Cloud Stack, Azure automation,



# 2.4 Open stack

- Example of the EGI fedcloud



# 2.5 Discussion

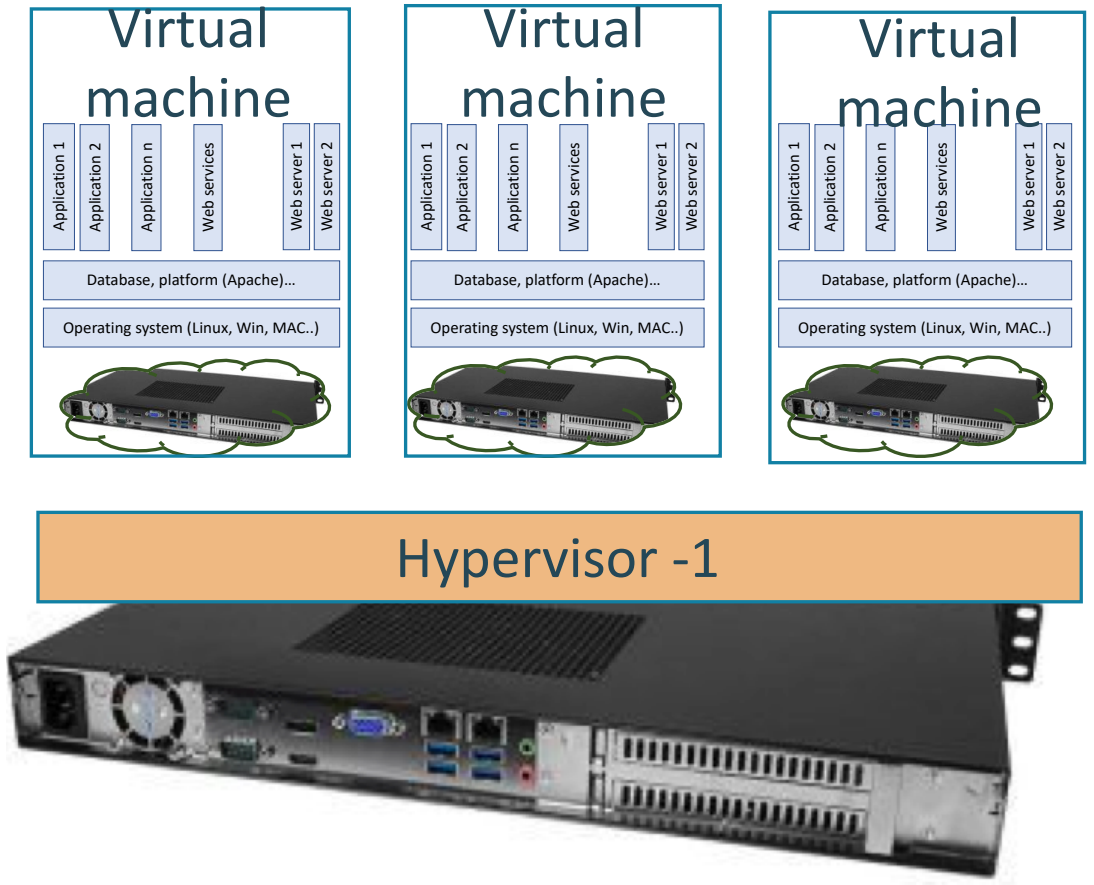
- What is the key difference between Type -1 and Type -2 hypervisor?

**www.menti.com**

**39 25 69**

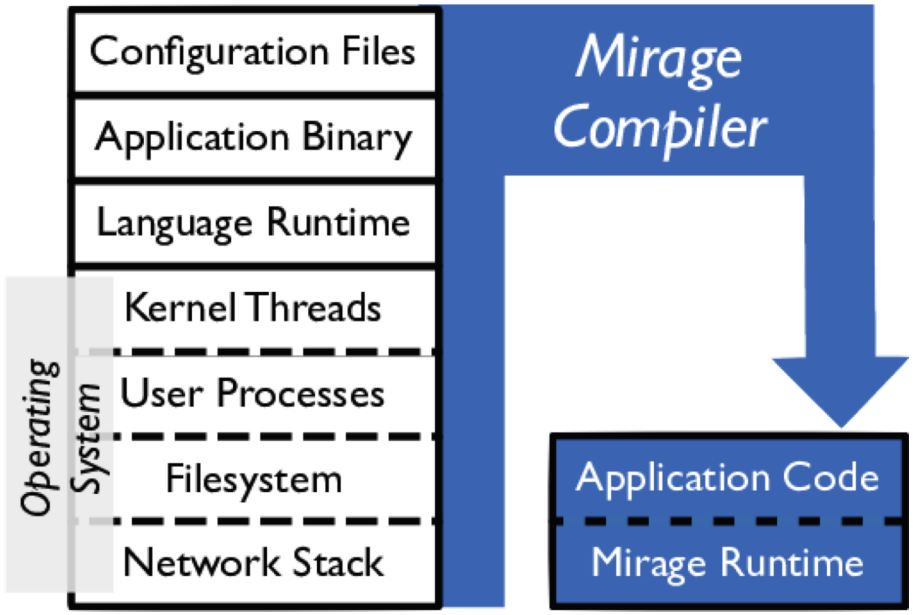
# 2.6 Limits of VM

- **Cloud IaaS**: VM (often contains a complete guest OS)
- **Large Image size (**several GB**), and large boot overhead (**from seconds to longer than a minute**)**
- Demands:
  - Improve deployment efficiency
- **Reduce the image size**



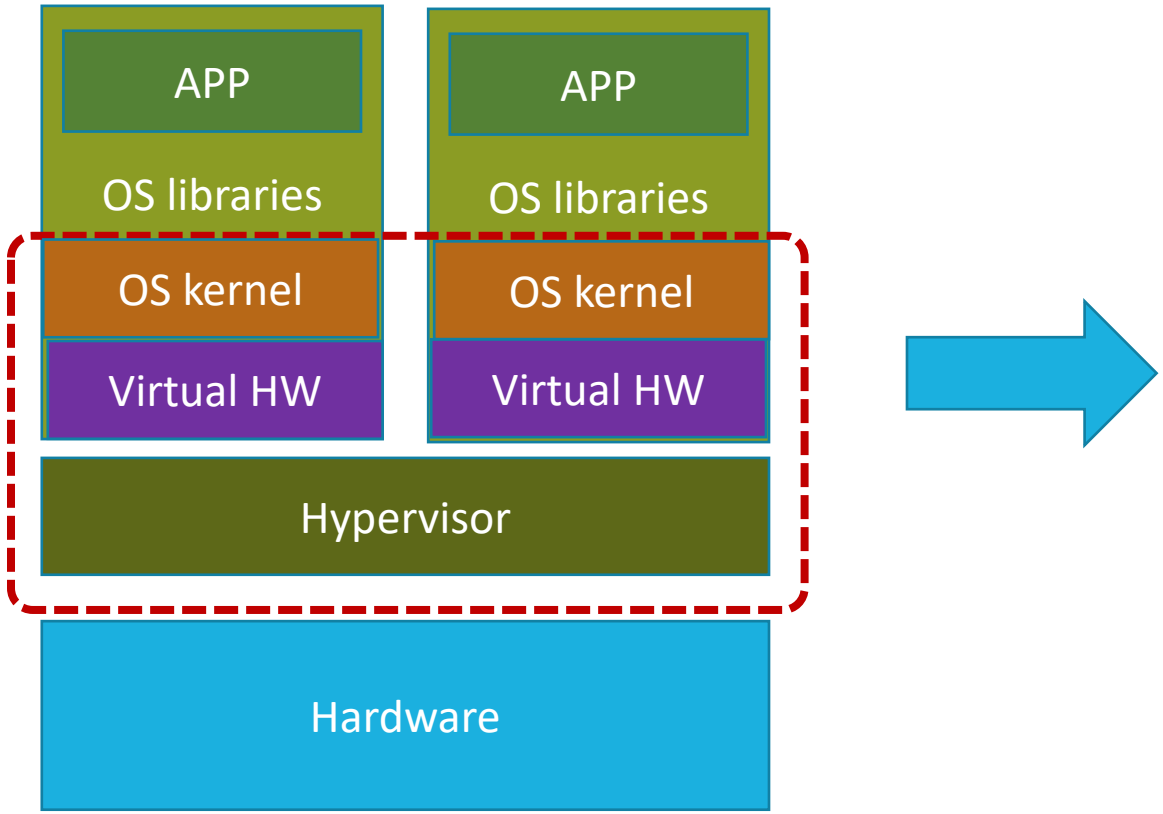
# 2.7 Approach 1: Simplifying OS

- UNIKERNEL approach
- Only keep the relevant part (**minimize the kernel**)
  - Single address space, OS as library, Directly on (virtual) hardware



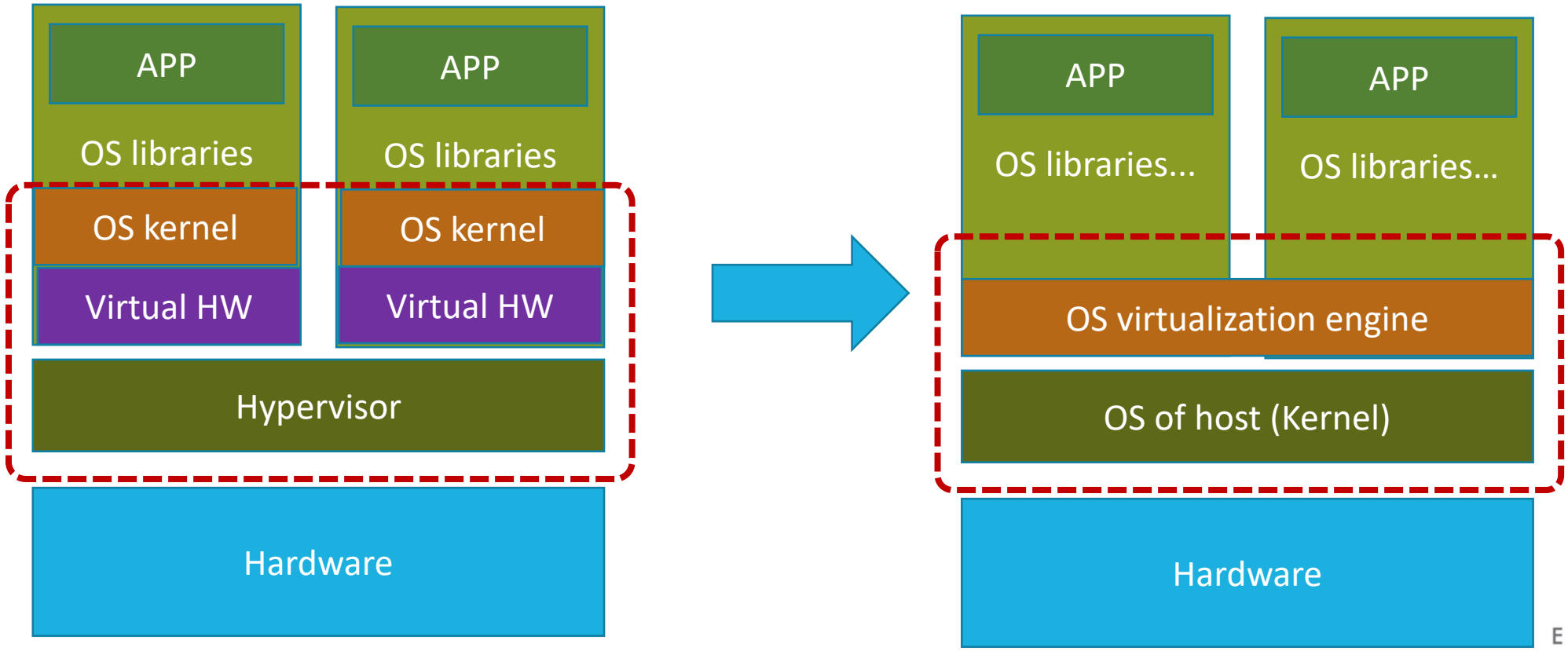

# 2.7 Approach 2: Operating system level virtualization

- Reduce full virtual hardware Shared kernel
- From full virtualization to container



# 2.8 Operating system level virtualization

- Reduce full virtual hardware Shared kernel
- From full virtualization to container



# 2.8 A history:

**1979-1982**

UNIX V7/**chroot** - Root file

**2000**

FreeBSD **Jails** Septate a system into jails, with independent IP

**2001**

**Linux V Server** (VPS), OS level

**2004**

**Solaris containers**

**2005**

**Open Virtuzzo**

OpenVZ

**2008**

**LXC Linux Container**

LXC

**2013**

**Docker** From LXC, with libcontainer

docker

**2014**

**RKT** CoreOS

rkt

**2015**

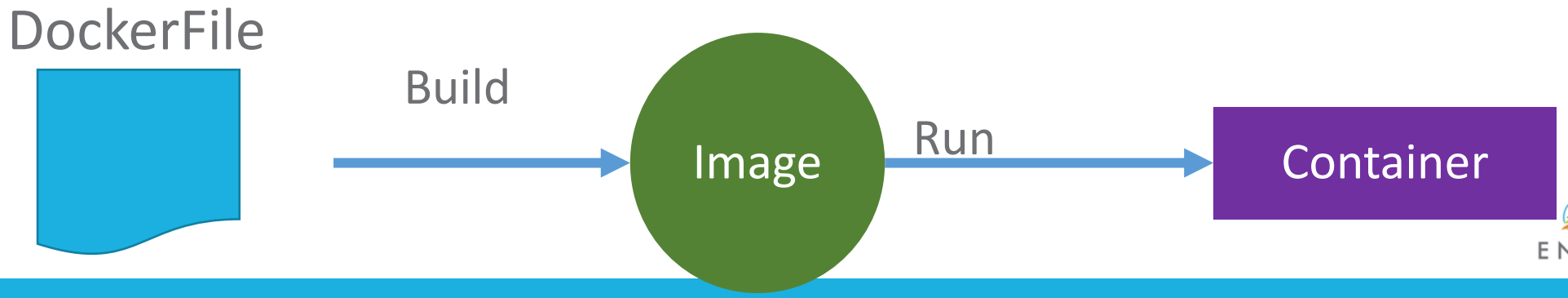
**OCI** Open Container Initiative

OPEN CONTAINER INITIATIVE

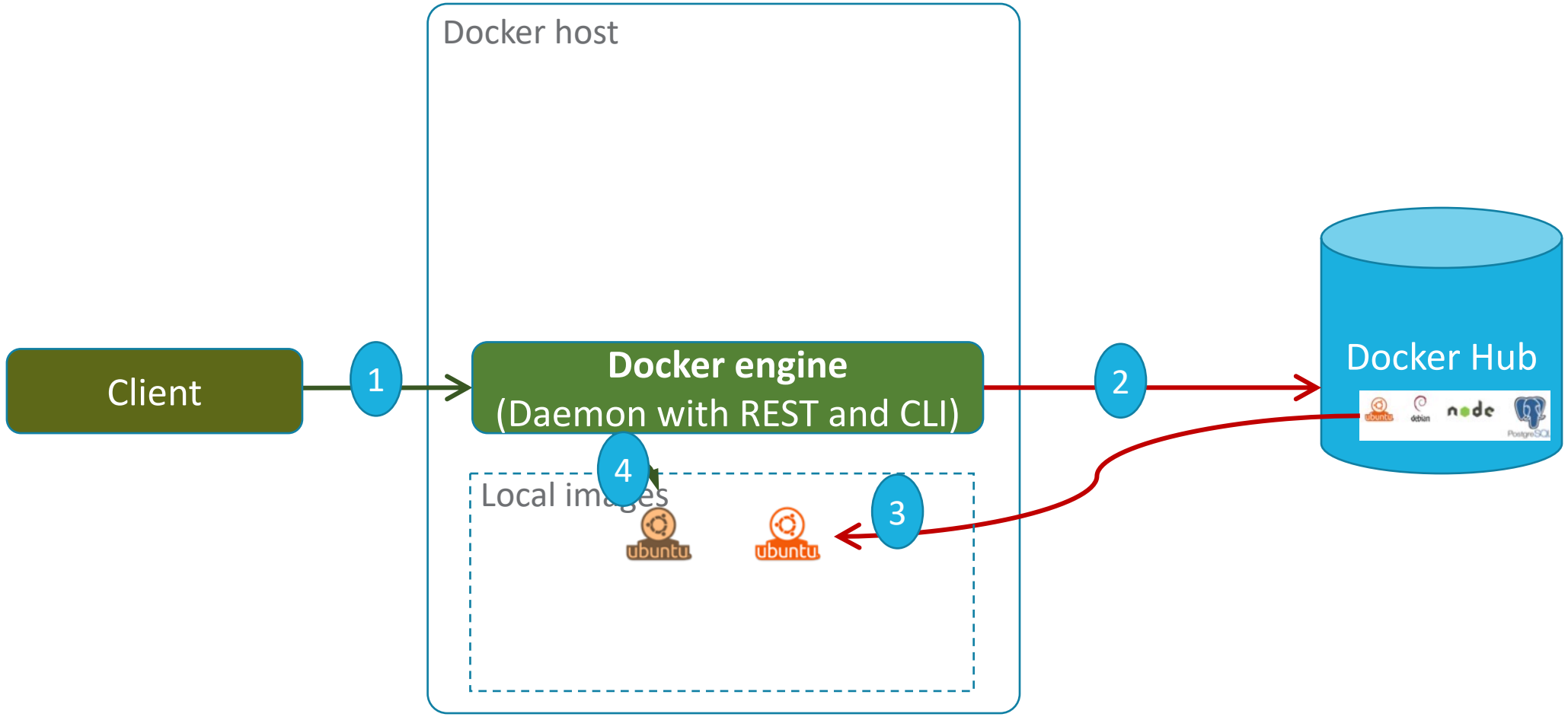
ENVRI FAIR

# 2.8 Common Terms and commands:

- **Docker Engine** – ***Docker Daemon, REST API, CLI***.
- **Image** – operating systems kernels supplied for a specific instance type / application.
- **Container** – an application running from an image.
- **DockerFile** – a text file with a list of steps to perform to create an image.
- **Docker Hub** – Docker Registry and Repository used for download and share images.

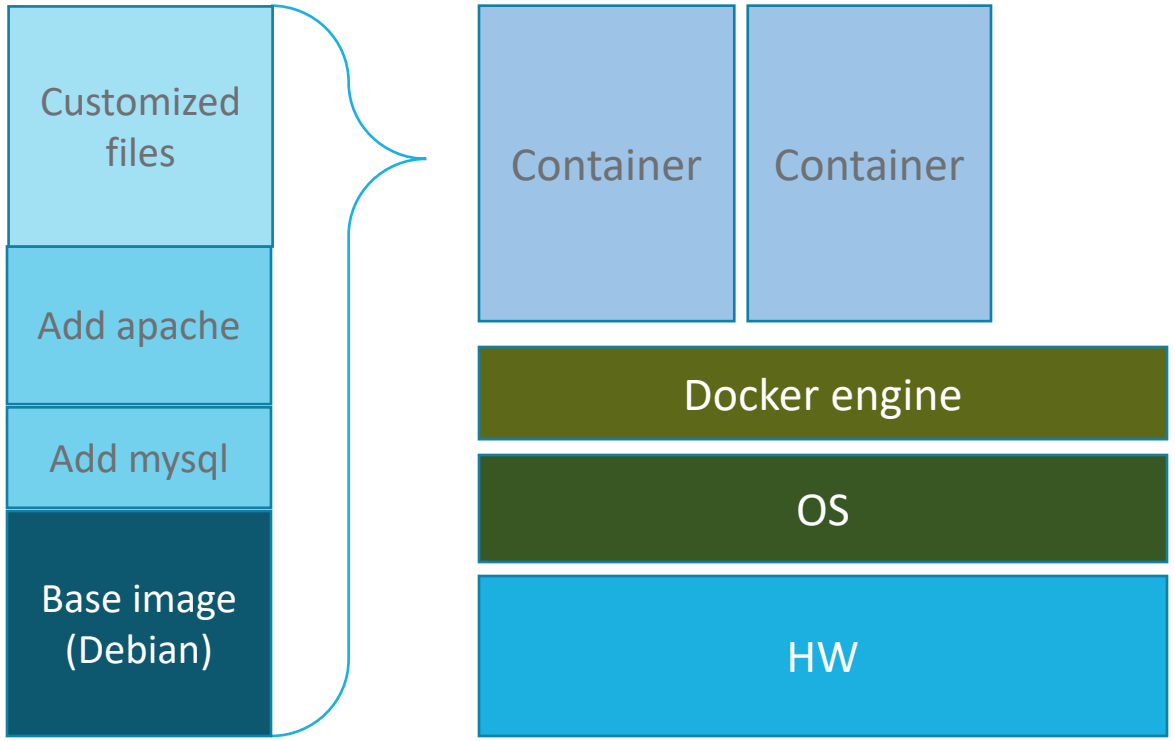


# 2.8 Docker: from image to container
# PULL, **BUILD**



# 2.8 Docker image

- Images are comprised of multiple layers,
- Every image contains a base layer
- Each layer references or is based on another image
- Each image contains software you want to run
- Basic layers are read only

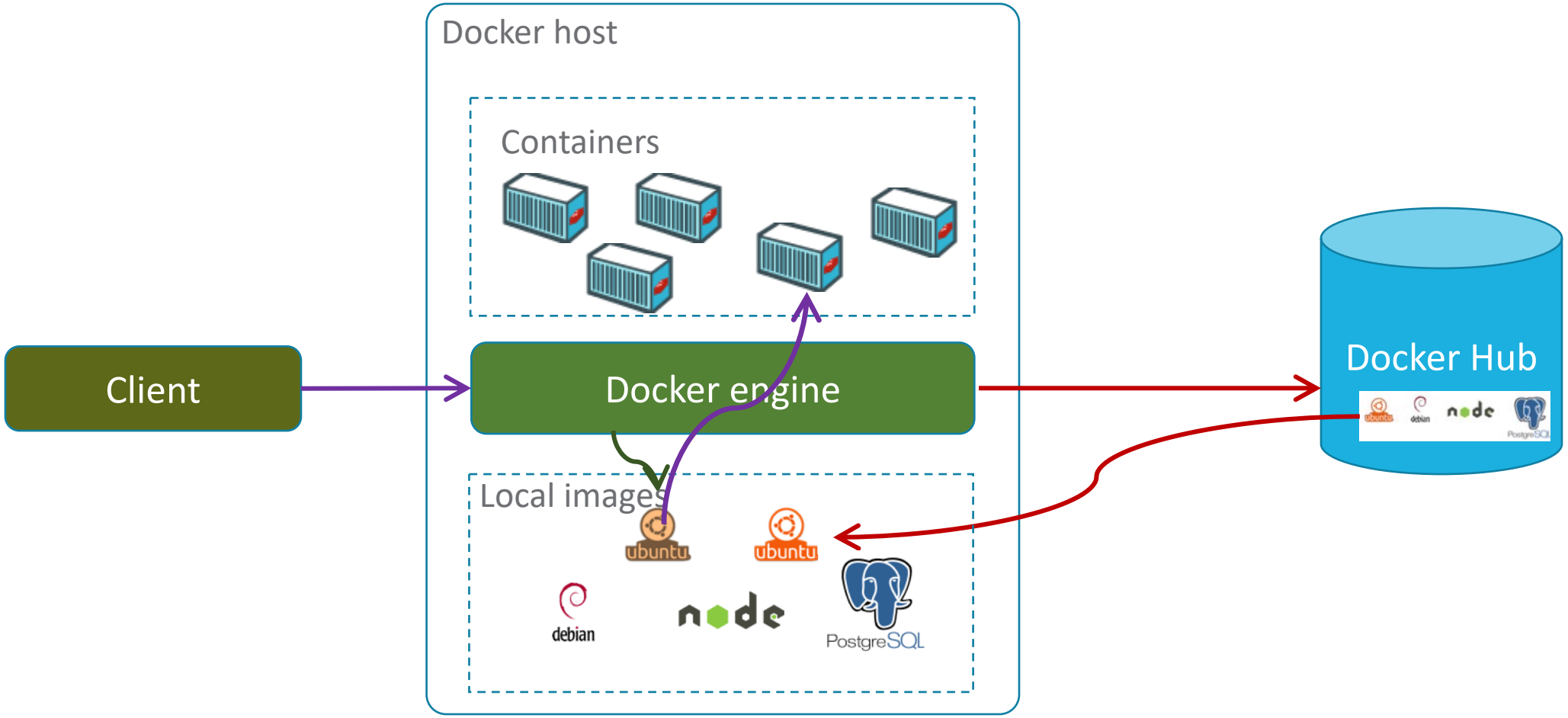


# 2.8 Docker Container Lifecycle

- Conception: **BUILD** an Image from a Dockerfile
  - docker image build [OPTIONS] PATH | URL | -
- Birth: **RUN** (create+start) a container
  - docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
- Reproduction: **COMMIT** (persist) a container to a new image
  - docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
- Sleep: **KILL** a running container
  - docker kill [OPTIONS] CONTAINER [CONTAINER...]
- Wake: **START** a stopped container
  - docker start [OPTIONS] CONTAINER [CONTAINER...]
- Death: **RM** (delete) a stopped container
  - docker rm [OPTIONS] CONTAINER [CONTAINER...]
- Extinction: **RMI** a container image (delete image)
  - docker rmi [OPTIONS] IMAGE [IMAGE...]

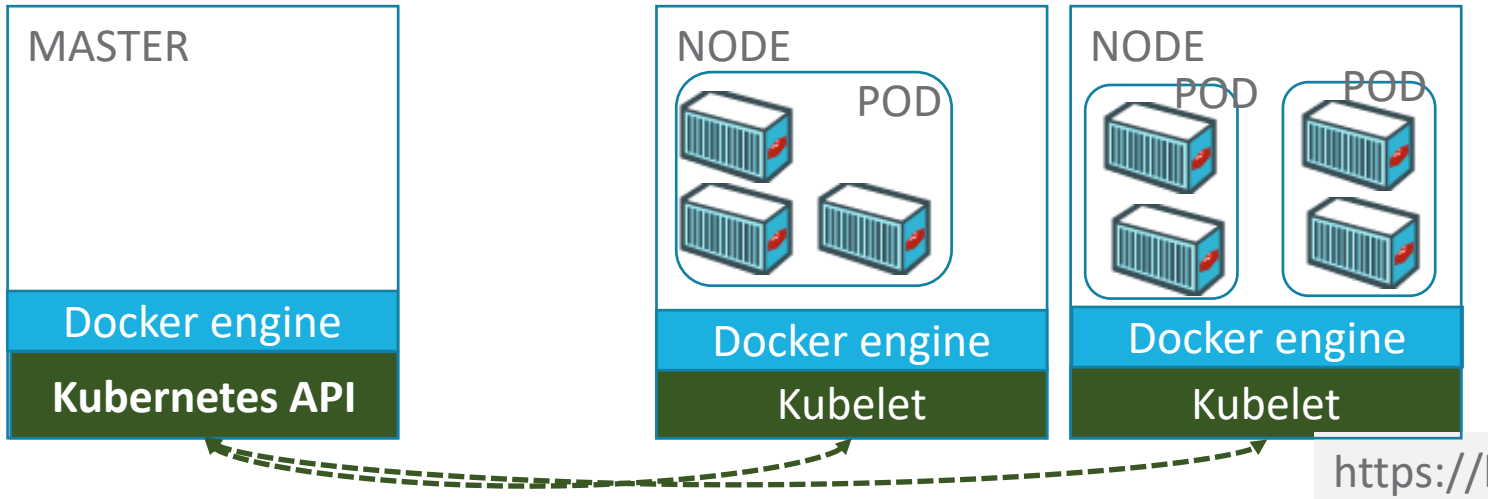
https://docs.docker.com/engine/reference/commandline/docker/

# 2.9 Docker: from image to container
# **Multiple containers**



# 2.9 Kubernetes: manage cluster for containers

- Smallest unit in Kubernetes, A structural abstraction of a group Containers
  - Some Containers are dependent, and need to be deployed in a single host, or work together. Share  IP address or port space.
  - Can also be on container per POD
- Containers in a POD share storage/network

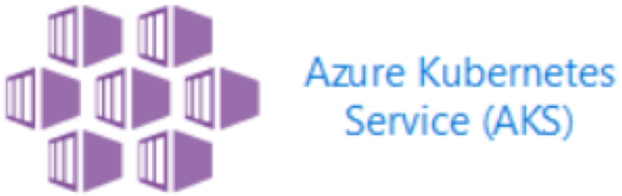



https://kubernetes.io/docs/setup/pick-right-solution/

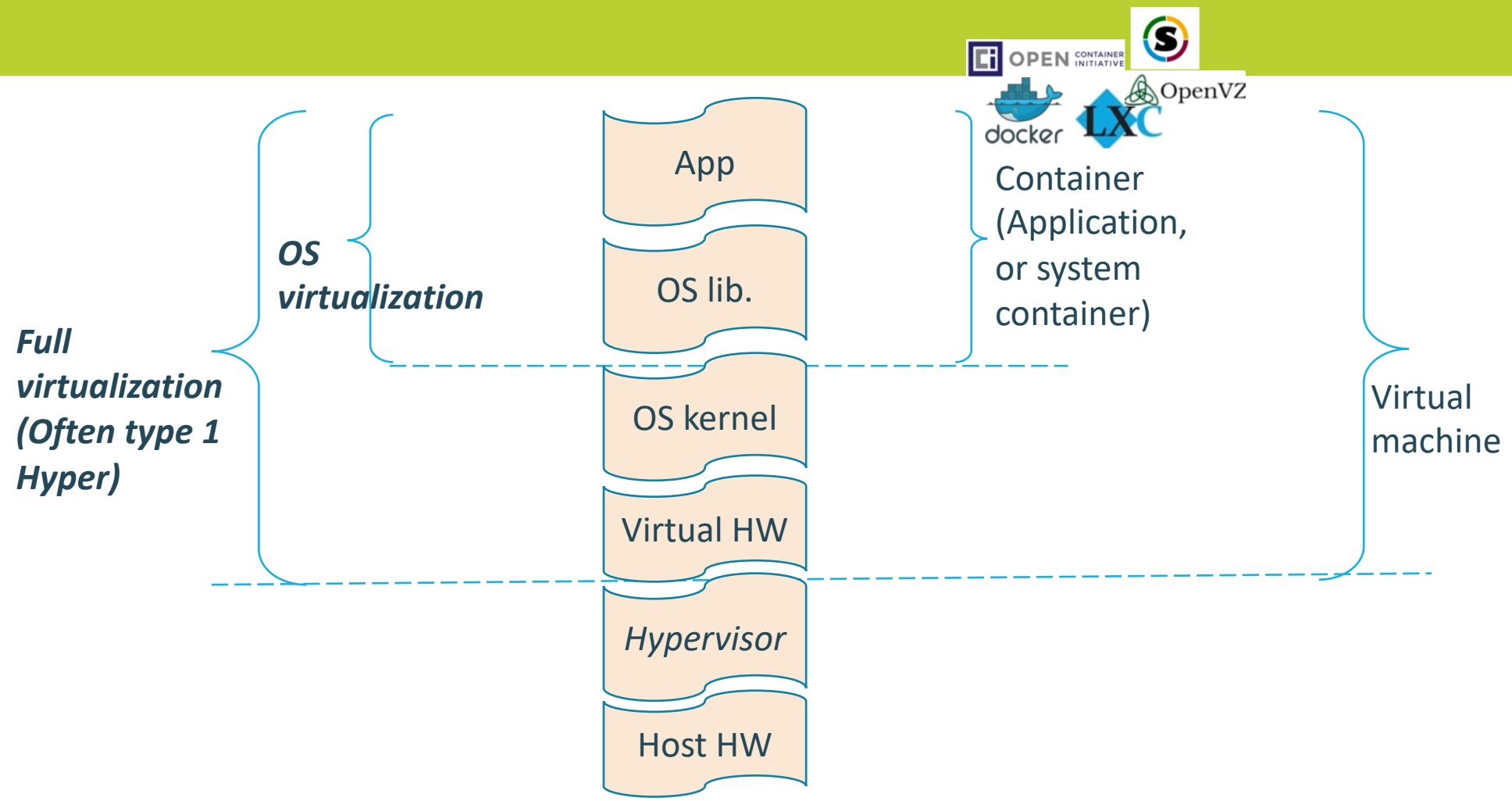
# 2.9 Kubernetes utilization

- Google Kubernetes Engine (GKE)
- Amazon Elastic Container Service for Kubernetes
- Azure Kubernetes Service (AKS)





# 2.10 Virtual machine and container



# 2.11 Discussion

- Choose correct statements

**www.menti.com**

**39 25 69**

# Outline

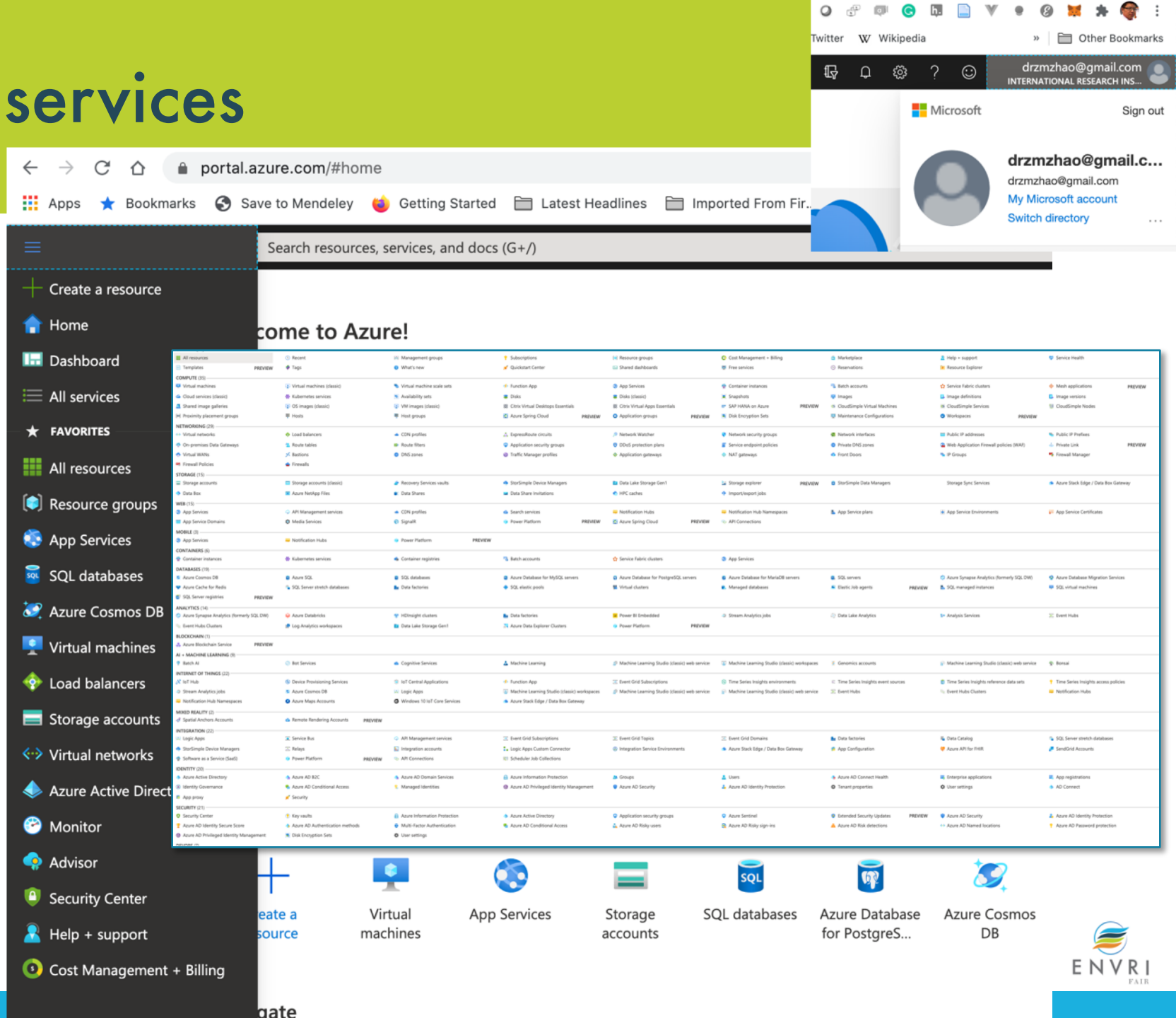
1. What is Cloud computing and the Cloud?
2. How does the Cloud work?
3. **How to use the Cloud?**
   a) **Use basic infrastructure service**
   b) Run a local application in Cloud
   c) Run complex applications in Cloud
   d) Infrastructure automation for operating online services
4. Discussion
5. Summary

# 3.a.1 What should you decide when go to Cloud?

- Which provider?
- Which data center?
- What Cloud services?
- What capacity?
- What budget?
- …

# 3.a.2 Cloud services

- The Cloud Portal (e.g. Azure)
- An account
- Service catalogue
- Order the services and ask for on demand provision



# 3.a.2 The EGI Service Catalogue

www.egi.eu/services

## Compute

### Cloud Compute

Run virtual machines on demand with complete control over computing resources

### Cloud Container Compute BETA

Run Docker containers in a lightweight virtualised environment

### High-Throughput Compute

Execute thousands of computational tasks to analyse large datasets

## Storage and Data

### Online Storage

Store, share and access your files and their metadata on a global scale

### Archive Storage

Back-up your data for the long term and future use in a secure environment

### Data Transfer

Transfer large sets of data from one place to another

## Security

### Check-in BETA

Login with your own credentials

## Applications

### Applications on Demand BETA

Use online applications for your data & compute intensive research

## Training

### FitSM Training

Learn how to manage IT services with a pragmatic and lightweight standard

### ISO 27001 Training

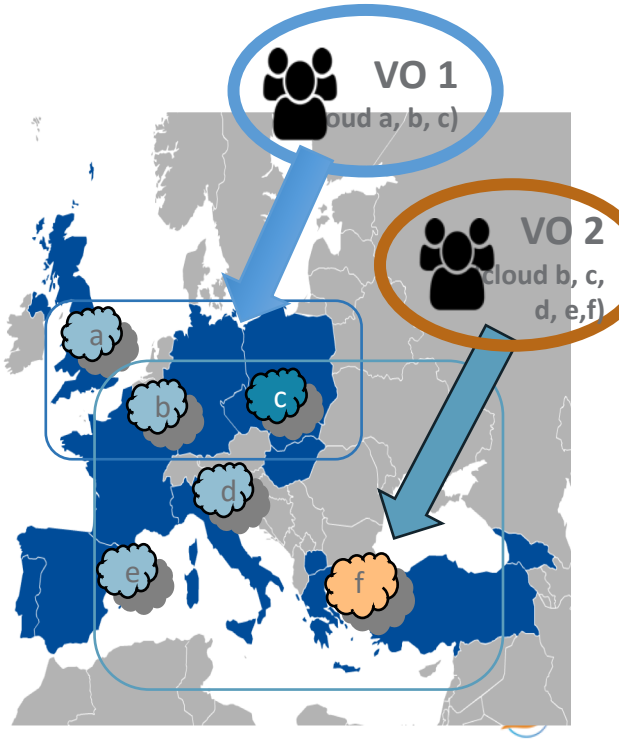
Learn how to manage and secure information assets

### Training Infrastructure

Dedicated computing and storage for training and education

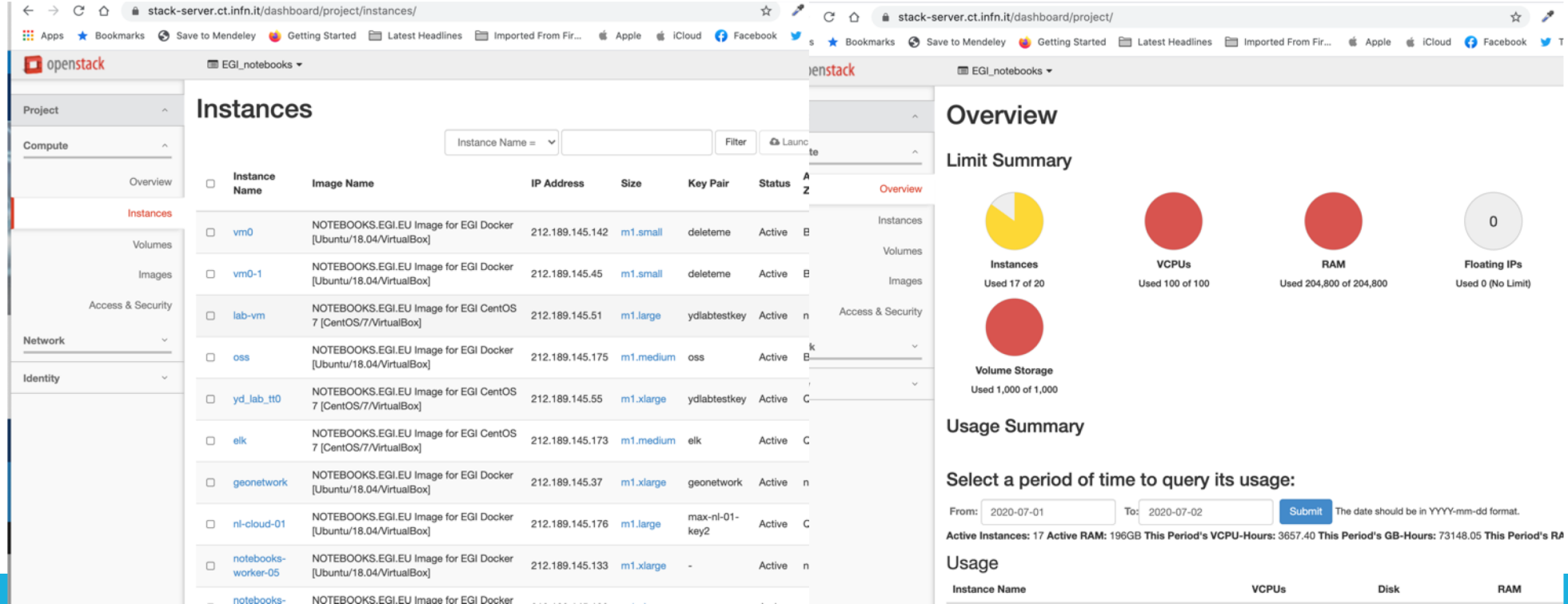
# 3.a.2 EGI Cloud Compute IaaS

- Run Virtual Machines on demand
- Access is based on ‘Virtual Organisations’
  - VO = group of users + cloud providers supporting them
    - Community-specific VOs – e.g. CHIPSTER, EISCAT, etc.
    - Generic VOs – e.g. fedcloud.egi.eu, training.egi.eu
- Diverse providers with common:
  - AuthN and AuthZ
  - VM Image catalogue (applications)
  - Information discovery
  - Accounting
  - Monitoring
  - GUI dashboard



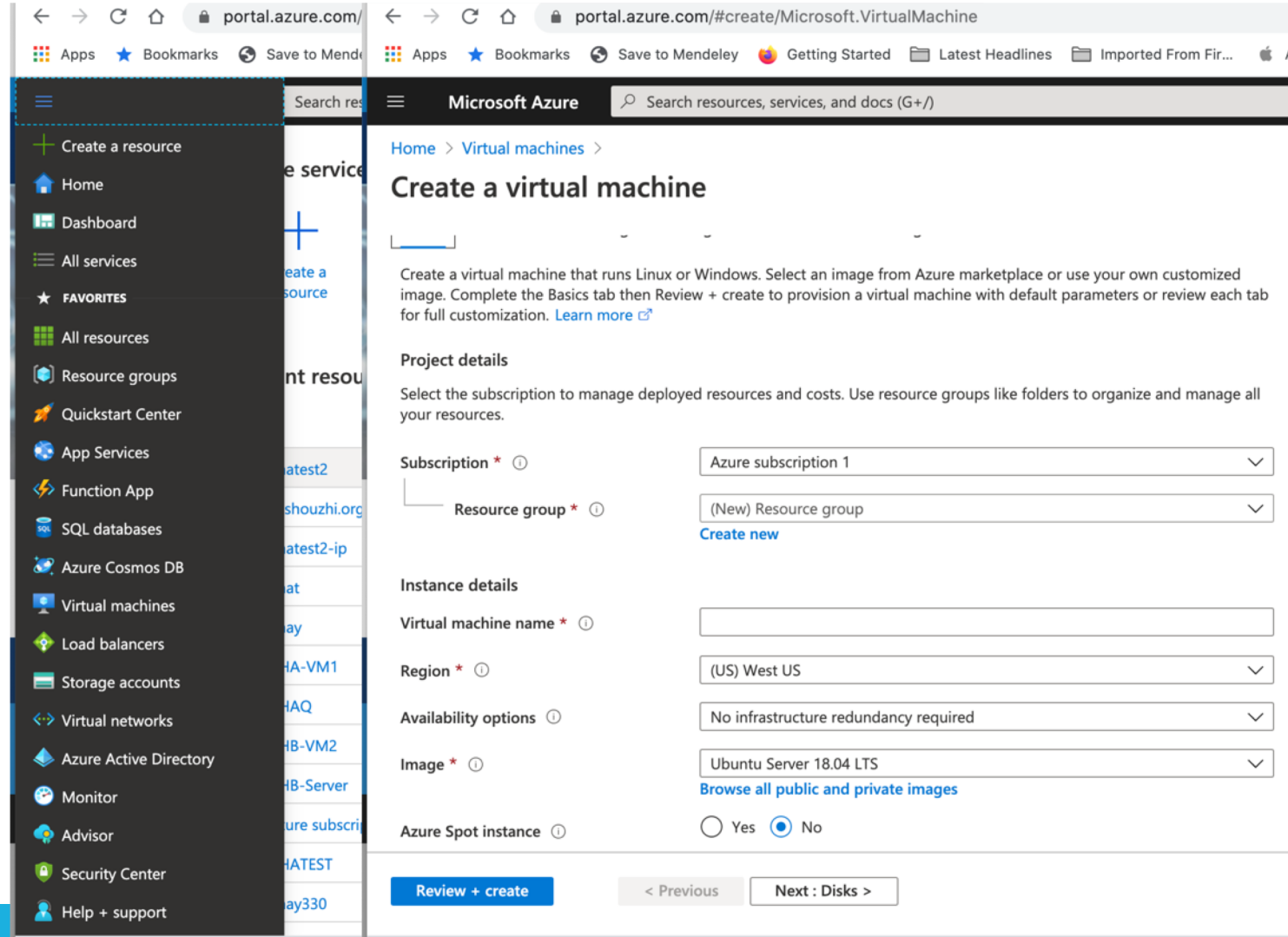
# 3.3 EGI- FedCloud (Steps…)

- Dashboard of the available resources
- Instances
- vCPU
- RAM



# 3.a.3 Azure: a public cloud example

- A big list of services they offer
- Including IaaS, PaaS, SaaS, and other new items
- VM example
  - Configure type
  - Data center
  - OS
  - Disk
  - Network



# 3.a.4 When a VM is created

- You can choose
  - Username/Password pair
  - provide your existing public key, or let the system create new key pair for you
- VM will have both public and private IP address
- Remote log in the created VM, in Linux machines
- Configure network of the VM(s)

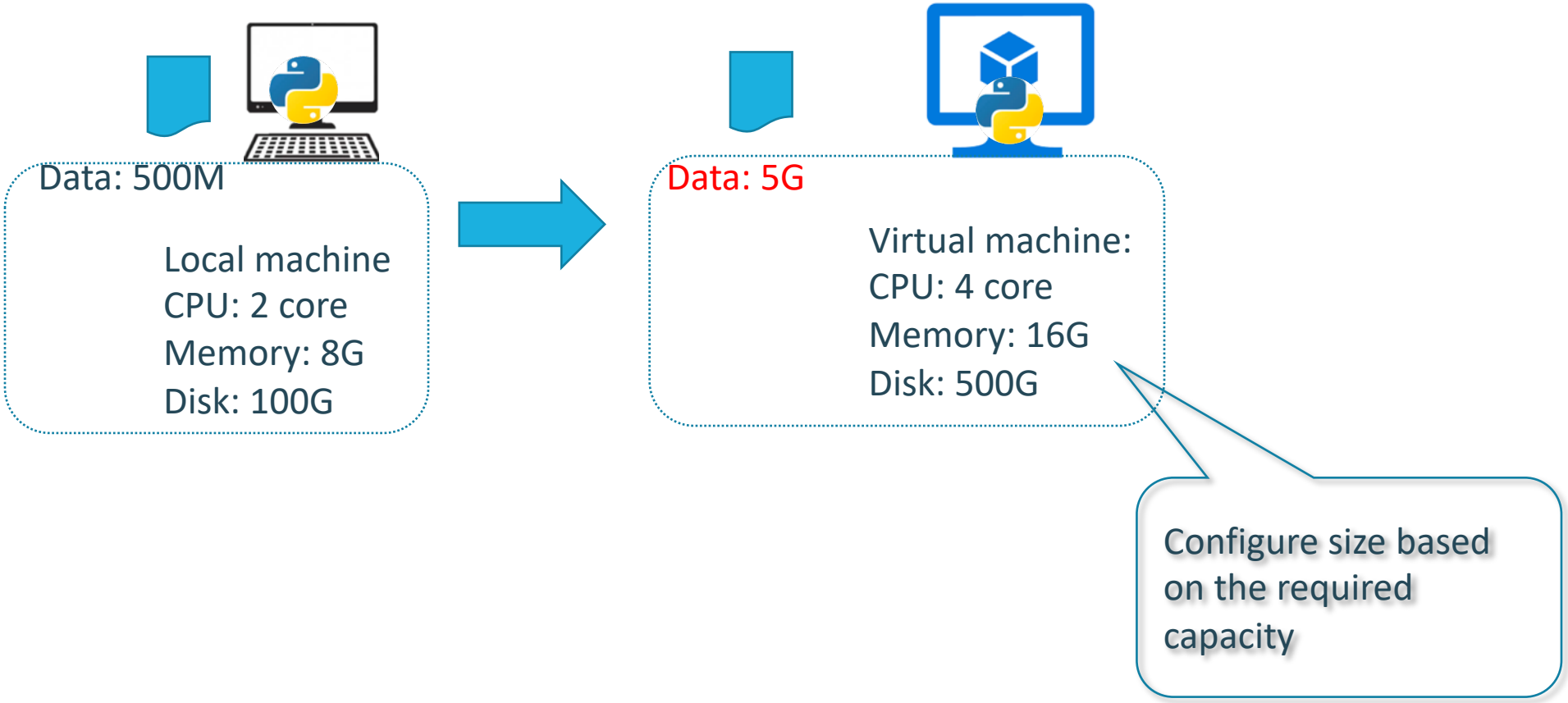
# Outline

1. What is Cloud computing and the Cloud?
2. How does the Cloud work?
3. How to use the Cloud?
   a) Use basic infrastructure service
   **b) Run a local application in Cloud**
   c) Run complex applications in Cloud
   d) Infrastructure automation for operating online services
4. Discussion
5. Summary

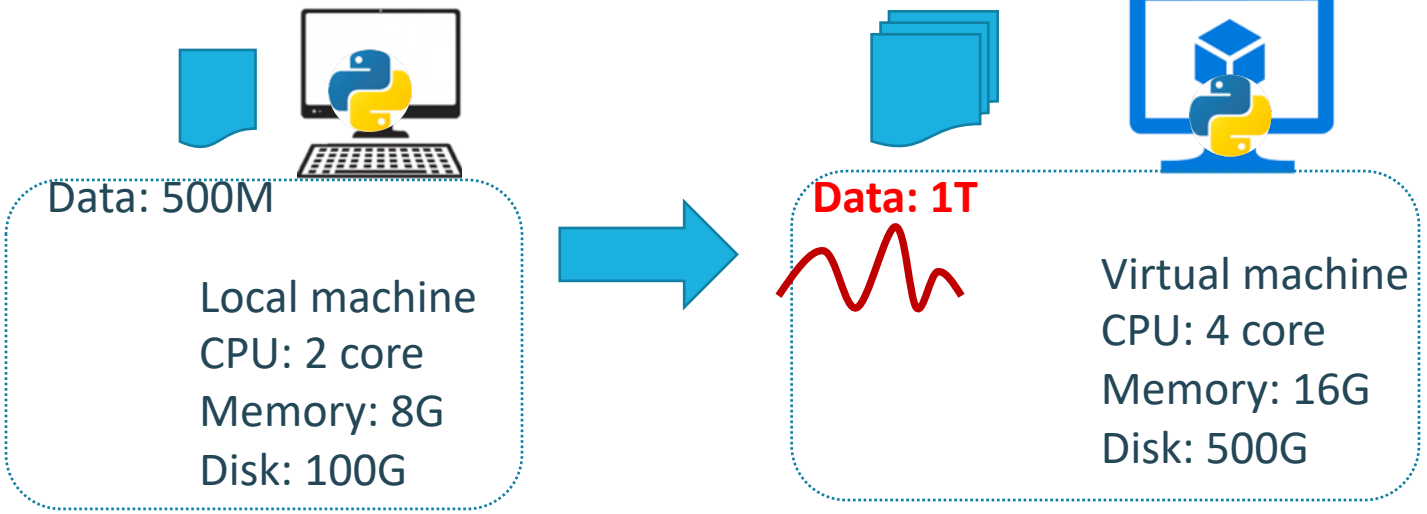
# 3.b.1 Run a local application in Cloud

1. Estimate the capacity of Virtual machine(s)
2. Create Virtual machines
3. Deploy software
4. Run the application

# 3.b.1 Cont.


Data: 500M
Local machine
CPU: 2 core
Memory: 8G
Disk: 100G
Data: 5G
Virtual machine:
CPU: 4 core
Memory: 16G
Disk: 500G
Configure size based on the required capacity

# 3.b.1 Cont.



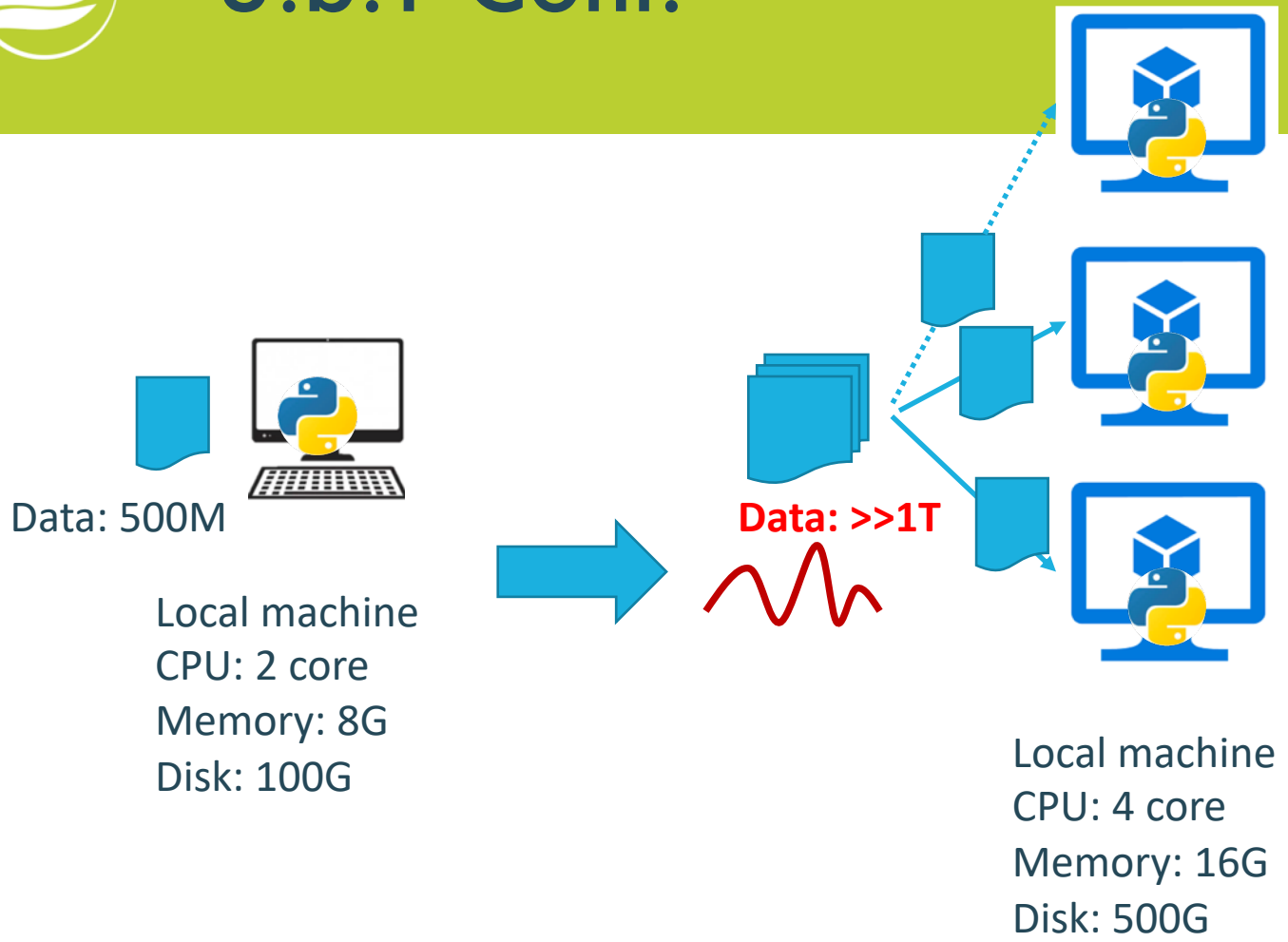

**Vertical scaling**

Scale Up/Down

- Up: increase capacity
- Down: decrease capacity

For VM scale up/down, you need restart the VM instance. It often has high overhead.

# 3.b.1 Cont.



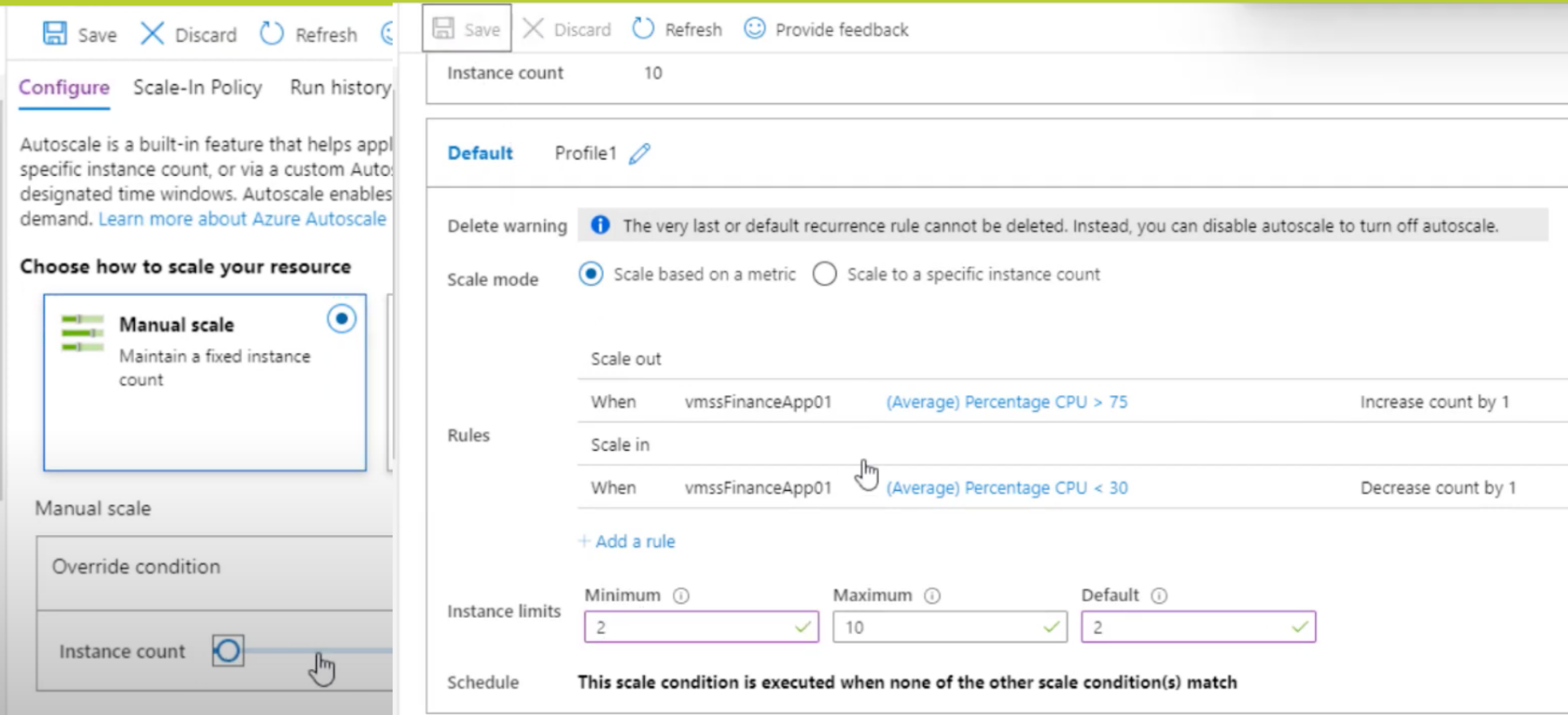

**Horizontal scaling**
Scale Out/In
- Out: increase instances
- In: decrease instances

For VM scale Out/In, require software can be migrated, without high dependencies.

# 3.b.2 Cloud elasticity

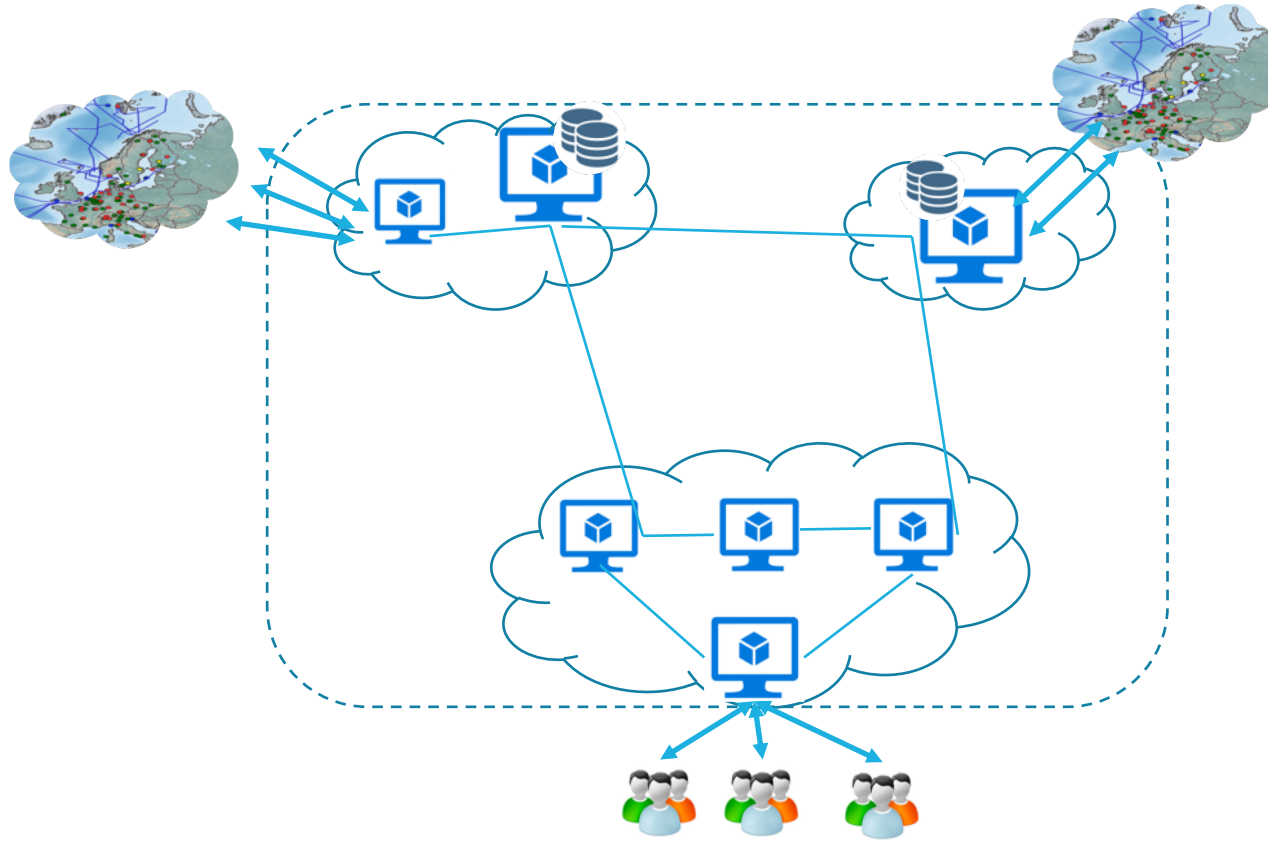
1. By scaling resources, the Cloud provides elastic capacity for computing
2. VM scaling can be provided as:
   a) A set of VM in the pool
   b) Dynamically boot VM instances when needed
   c) as a service by the provider (e.g. Azure, via VM scale set for horizontal scaling)
3. Other types of elasticity
   a) Load balancer
   b) Micro services and Containers

# 3.b.2 VM scaling (Example in Azure)



# 3.b.3 Networked VMs

- Distributed applications:
  - Single VM is not sufficient
  - Large user communities
  - Distributed data sources
  - Cost optimization
- Decide the VM set, and define the network topology among them



# 3.b.3 Other examples

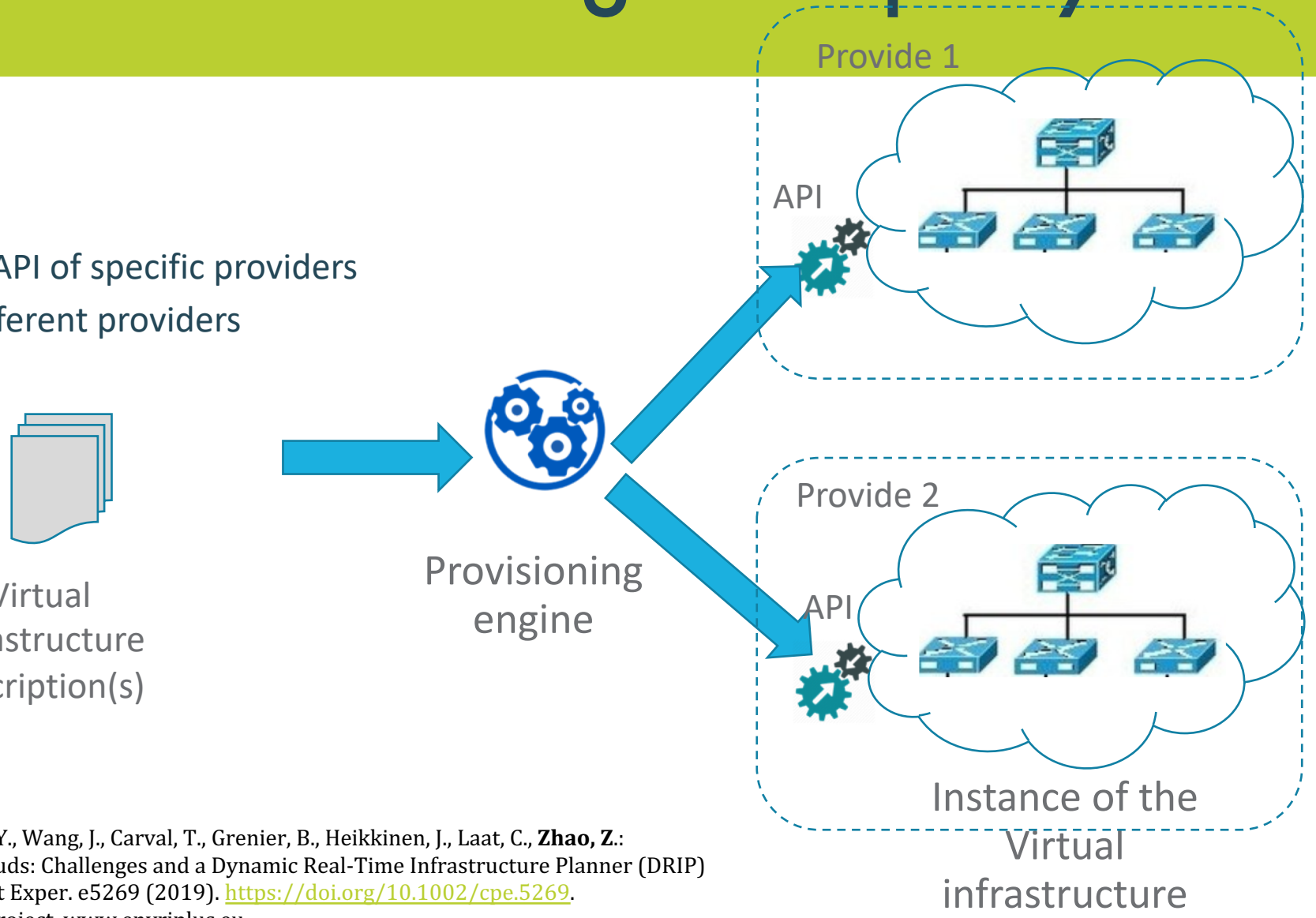
Further reading:

*ACM Computing Surveys (CSUR) Surveys Homepage archive, Volume 51 Issue 1, April 2018 Article No. 22*

| CML | Pragmatics | Target |
|---|---|---|
| Blueprint | Cloud service composition and description of deployment configurations | XaaS |
| Caglar *et al.* | Cloud service simulation and description of deployment plan configurations | IaaS |
| CAML | Cloud application architecture description and refinement of deployment configurations towards target cloud environment | XaaS |
| clADL | Architecture description of interactive cloud services and generation of implementations for the cyber-physical systems domain | XaaS |
| CloudDSL | Description of deployment configurations | XaaS |
| CloudMIG | Application migration to the cloud with emphasis on optimal deployment configurations and their conformance with target cloud environments | PaaS I |
| CloudML-SINTEF | Automated provisioning of multi-cloud applications and re-configuration of provisioned cloud services at run-time | XaaS |
| CloudML-UFPE | Description of cloud services | IaaS |
| CloudNaaS | Description of deployment configurations with emphasis on network aspects | IaaS |
| GENTL | Description of deployment configurations with emphasis on cost-efficient application provisioning | XaaS |
| Holmes | Description of deployment configurations and their automated provisioning | XaaS |
| MOCCA | Optimal (re)arrangement of (existing) deployment configurations for application provisioning to multiple target cloud environments | XaaS |
| MULTI-CLAPP | Application code generation for target cloud environments from component configurations | XaaS |
| Nhan *et al.* | Feature model based software stack (re-)configuration and their automated provisioning | IaaS |
| PDS | Deployment plan generation from described deployment configurations | IaaS |
| RESERVOIR-ML | Description of deployment configurations with emphasis on application-triggered elasticity rules for infrastructure-related cloud services | IaaS |
| StratusML | Generation of executable deployment descriptor and run-time adaptation rule from described deployment configurations | XaaS |
| TOSCA | Description of portable composite cloud applications for their automated provisioning and life-cycle management | XaaS |
| | communication between components | |

# 3.b.3 Provision using third party tools

- Select providers
- SLA
- Provision via the API of specific providers
- Provision over different providers





1. Koulouzis, S., Martin, P., Zhou, H., Hu, Y., Wang, J., Carval, T., Grenier, B., Heikkinen, J., Laat, C., **Zhao, Z.**: Time-critical data management in clouds: Challenges and a Dynamic Real-Time Infrastructure Planner (DRIP) solution. Concurrency Computat Pract Exper. e5269 (2019). https://doi.org/10.1002/cpe.5269.
2. Zhao Z., et al. D7.2, D7.4, ENVRIplus project, www.envriplus.eu

# 3.b.4 Discussion

- Which scaling has minimal influence on the application?
  - Horizontal scaling – in/out
  - Vertical scaling- up/down

**www.menti.com**

**39 25 69**

# Outline

1. What is Cloud computing and the Cloud?
2. How does the Cloud work?
3. How to use the Cloud?
   a) Use basic infrastructure service
   b) Run a local application in Cloud
   **c) Run complex applications in Cloud**
   d) Infrastructure automation for operating online services
4. Discussion
5. Summary

# 3.c.1 Run a local program as part of a workflow in Cloud

1. How to make a program as reusable component ?
2. How to deploy the components in Cloud environments?
3. How to adapt the components at runtime?

# 3.c.1 Run a local program as part of a workflow in Cloud

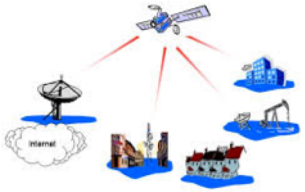
a) Services in Cloud
b) Service deployment
c) Load balancer and autoscaling for service execution

# 3.c.2 What is a service?

- **Services** (also known as "**intangible goods**") include attention, advice, access, experience, and affective labor. [*from wikipedia*]
- **Services** are a new way to deliver software value; from pay per software to pay per use.
- **Services** are used to automate business activities and operations (*)
  - **Is a software program that makes its functionality available via a published API that is part of a service contract**
  - Is comprised of **a body of logic** designed to carry out a collection of capabilities and
  - **a service contract** that expresses which of its capabilities are made available for **public invocation**.
- …


**travel agency**

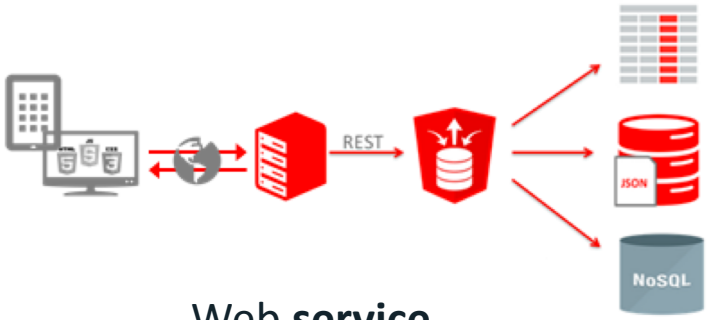
Logistics **service**



Satellite **service**

Communication **service**


Cloud **service**



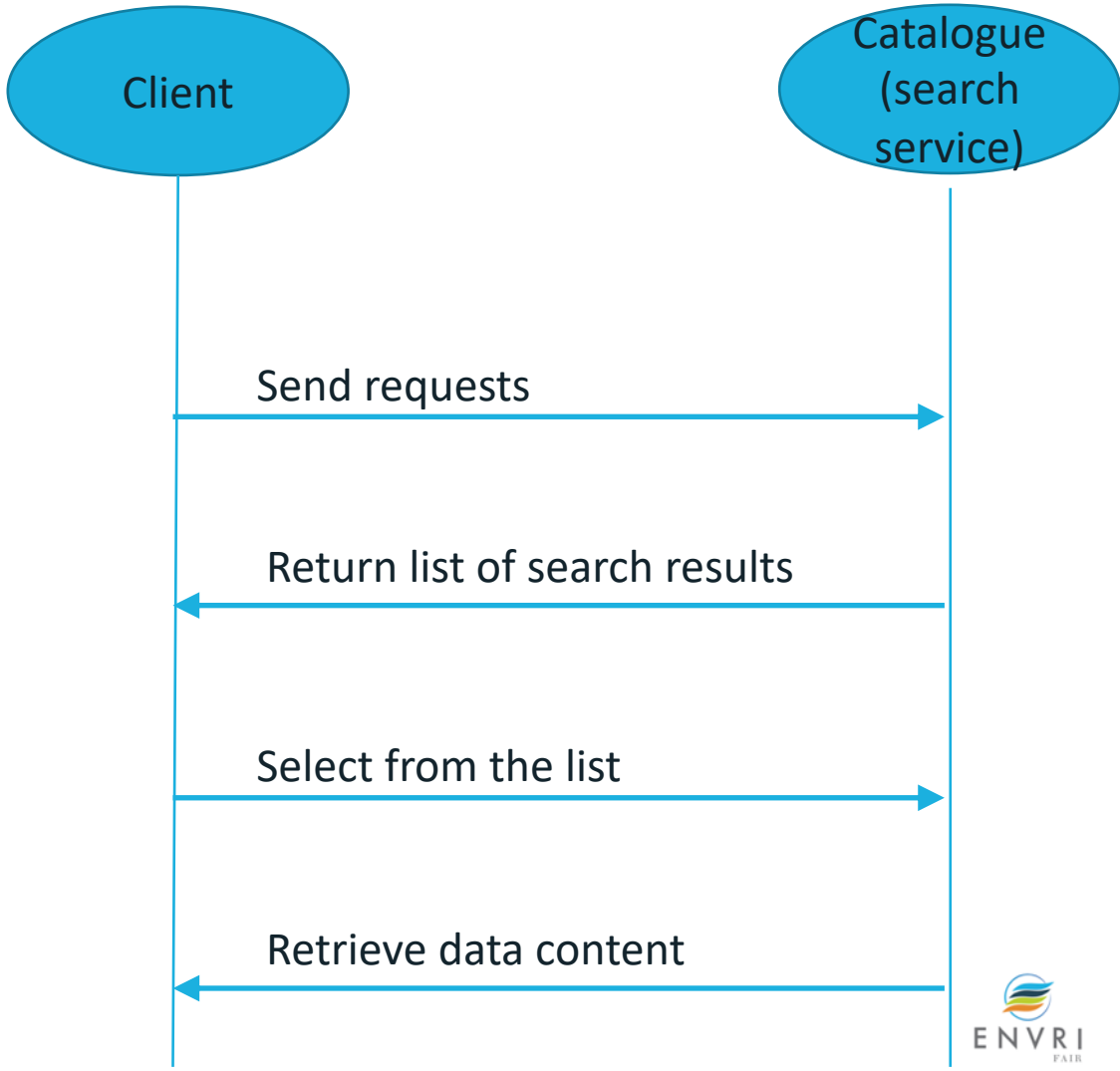

Web **service**

** Thomas Erl,(2016) Service-Oriented Architecture: Analysis and Design for Services and Microservices, 2nd Edition*

# 3.c.2 Services (a catalogue example)

A typical scenario:

- **Send query (e.g. look for data records)**
- **Receive search results**
- **Select result**
- **Retrieve the data content of a selected item**



# 3.c.2 Service implementation

- **Communication protocol**
  - HTTP based: web services
  - Or other protocols
- **Message schema**
  - XML, JSON, or others
- **Interface description**
  - WSDL, WADL, Open API description

# 3.c.2 REST: Representational State Transfer

- **Representational State Transfer (REST)** is a software architectural style for networked system.
- **Web services** that conform to the REST architectural style, termed **RESTful web services （Restful Web API）**

# 3.c.2 Resources on web (via browser)

**Request: Uniform Resource Locator (URL)**

https://www.lifewatch.eu/web/guest/catalogue-of-virtual-labs

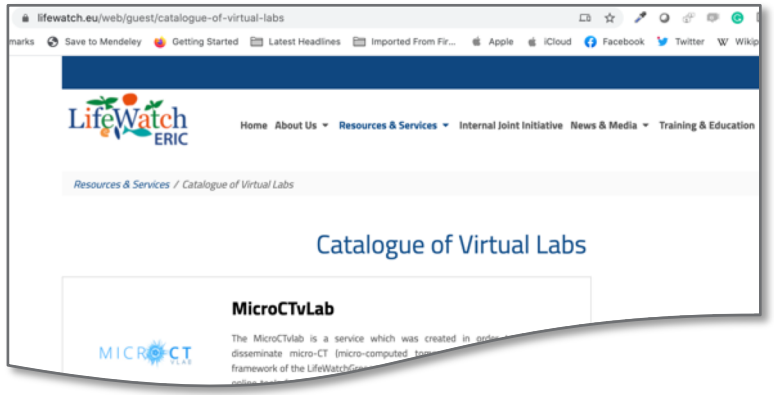
protocol | host | resource

Client

Server (Web)

**Return representation of resources**

HTML, XML, JASON, IMAGE, MEDIA …



# 3.c.2 RESTful API

**getGazetteerRecordByMRGID**

GET /getGazetteerRecordByMRGID.json/{MRGID}/

## Request: verb + URL

**GET**
**POST**
**PUT**
**DELETE**

**URL**

https://www.marineregions.org/rest/getGazetteerRecordByMRGID.json/222/

REST API endpoint URL

resource

Client

Server (Web)

## Return representation of resources

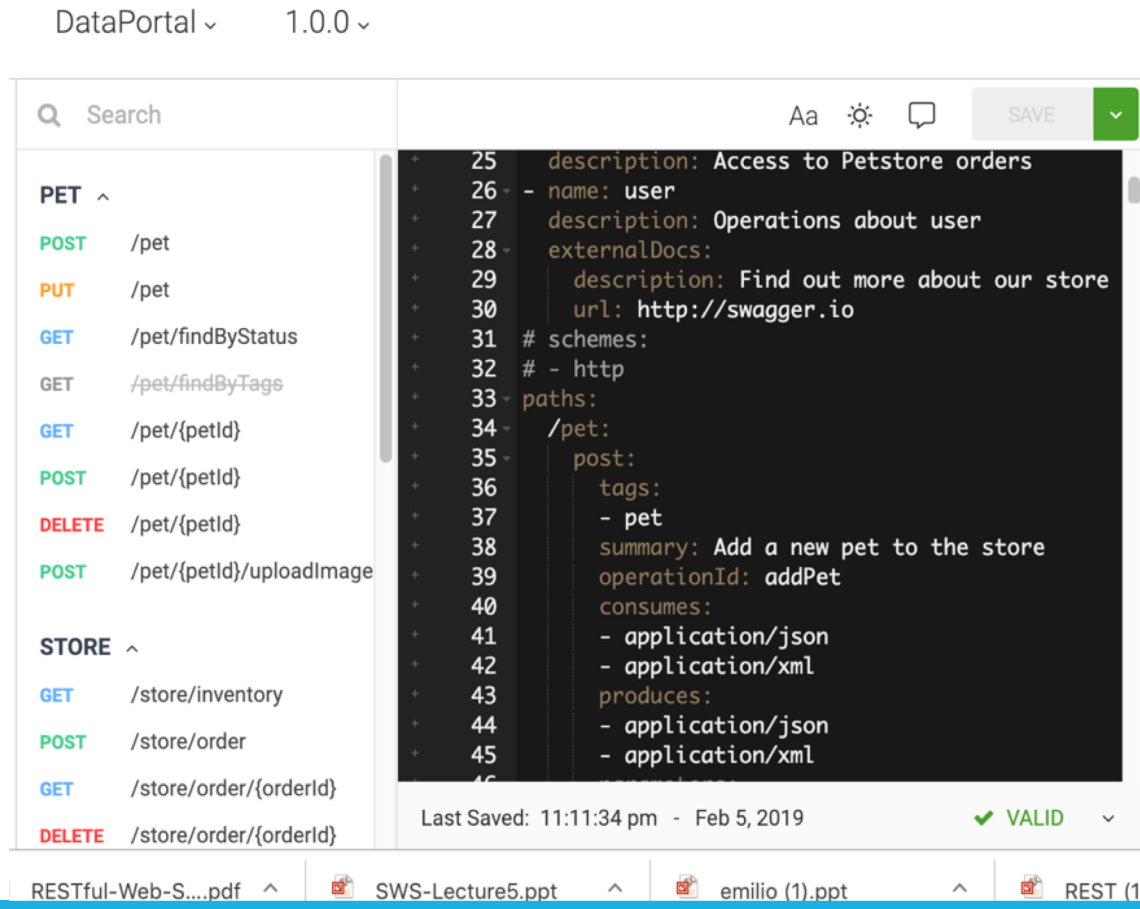
HTML, XML, JASON, IMAGE, MEDIA …


```
{
  "MRGID": 222,
  "gazetteerSource": "SAIL",
  "placeType": "Commune",
  "latitude": 51.56968,
  "longitude": 4.093495,
  "minLatitude": 51.47911,
  "minLongitude": 3.93802,
  "maxLatitude": 51.66024,
  "maxLongitude": 4.24897,
  "precision": 14730.95,
  "preferredGazetteerName": "Tholen",
  "preferredGazetteerNameLang": "Dutch",
  "status": "standard",
  "accepted": 222
}
```

# 3.c.2 RESTful API

- Define API interface
- Generate code
  - For the server side
  - For the client side
- Develop code

- Swagger environment
- Export to the server/client side code



# 3.c.3 Deploy a service in the Cloud

1. Develop and test the service
2. Deploy services in the Cloud environment
    1. Create VM in Cloud
    2. Install software platforms required by the service
    3. Deploy services

# 3.c.3 Deployment of software

- Manual deployment
- Automate the deployment steps
- Include contextual information (runtime environments, libraries and configuration) all in one package

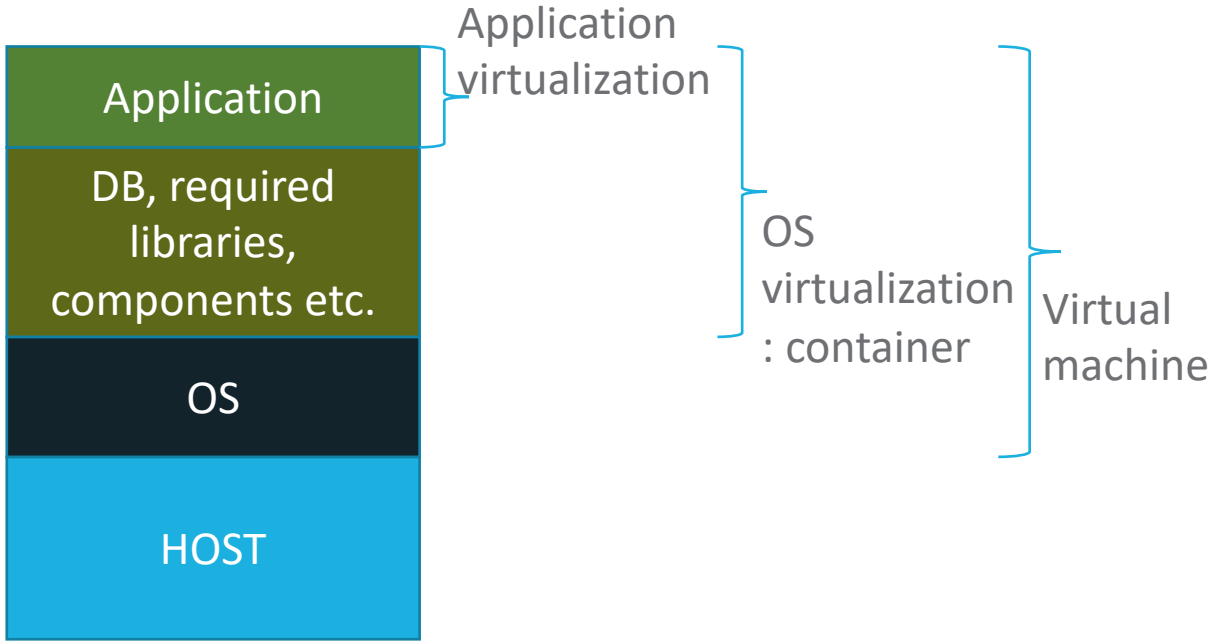
# 3.c.3 Deployment of software

- Manual deployment ➔ time consuming and require profound technical knowledge
- Automate the deployment steps ➔ using tools like Ansible (playbook)
- Include contextual information (runtime environments, libraries and configuration) all in one package ➔ using VM, Containers etc.

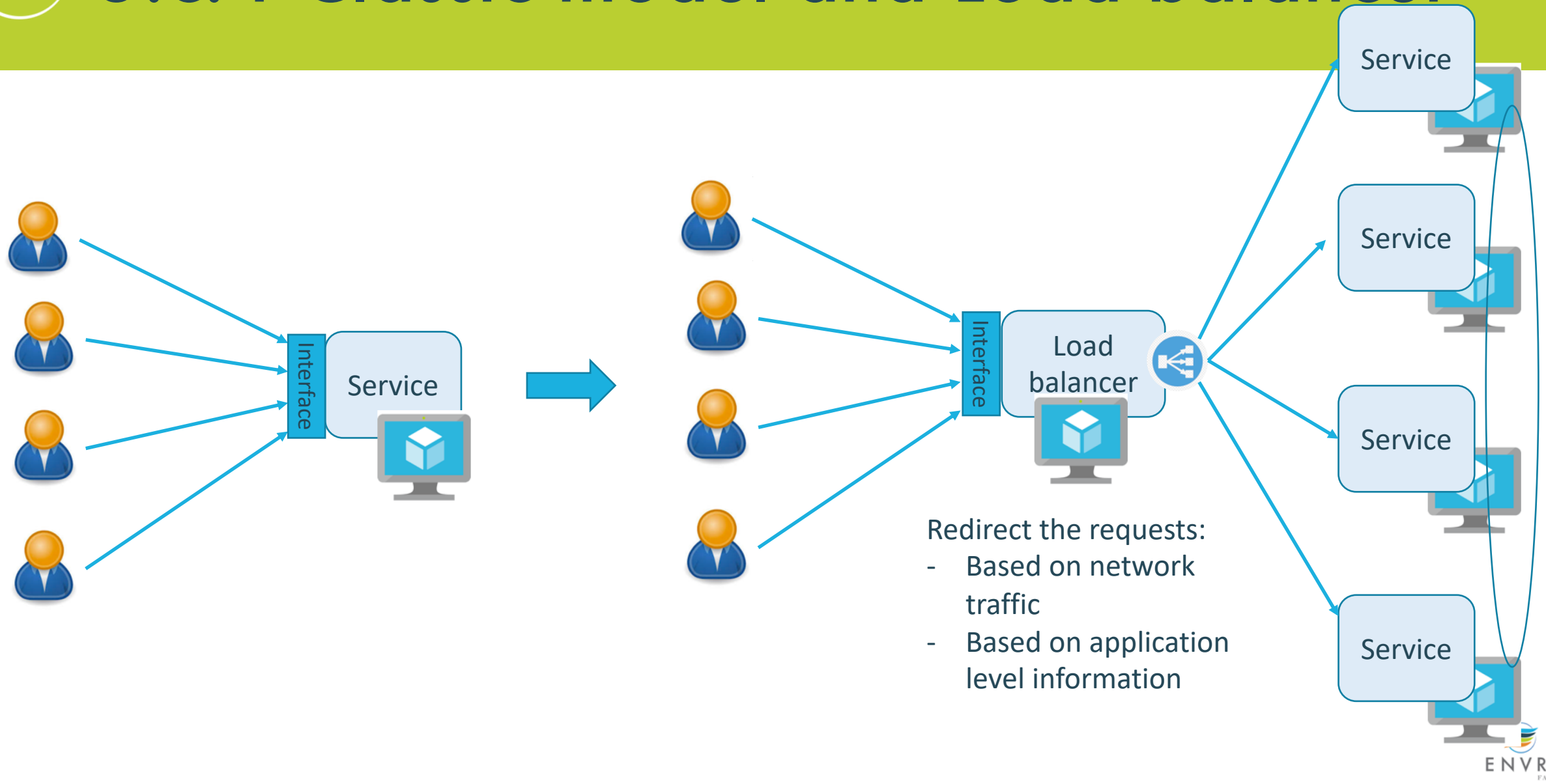
# 3.c.3 Application delivery and deployment

Options:

- Only code
- Built application
- Together with the libraries, components, data bases,
- With Operating system
- With the Host



# 3.c.4 Classic model and Load balancer

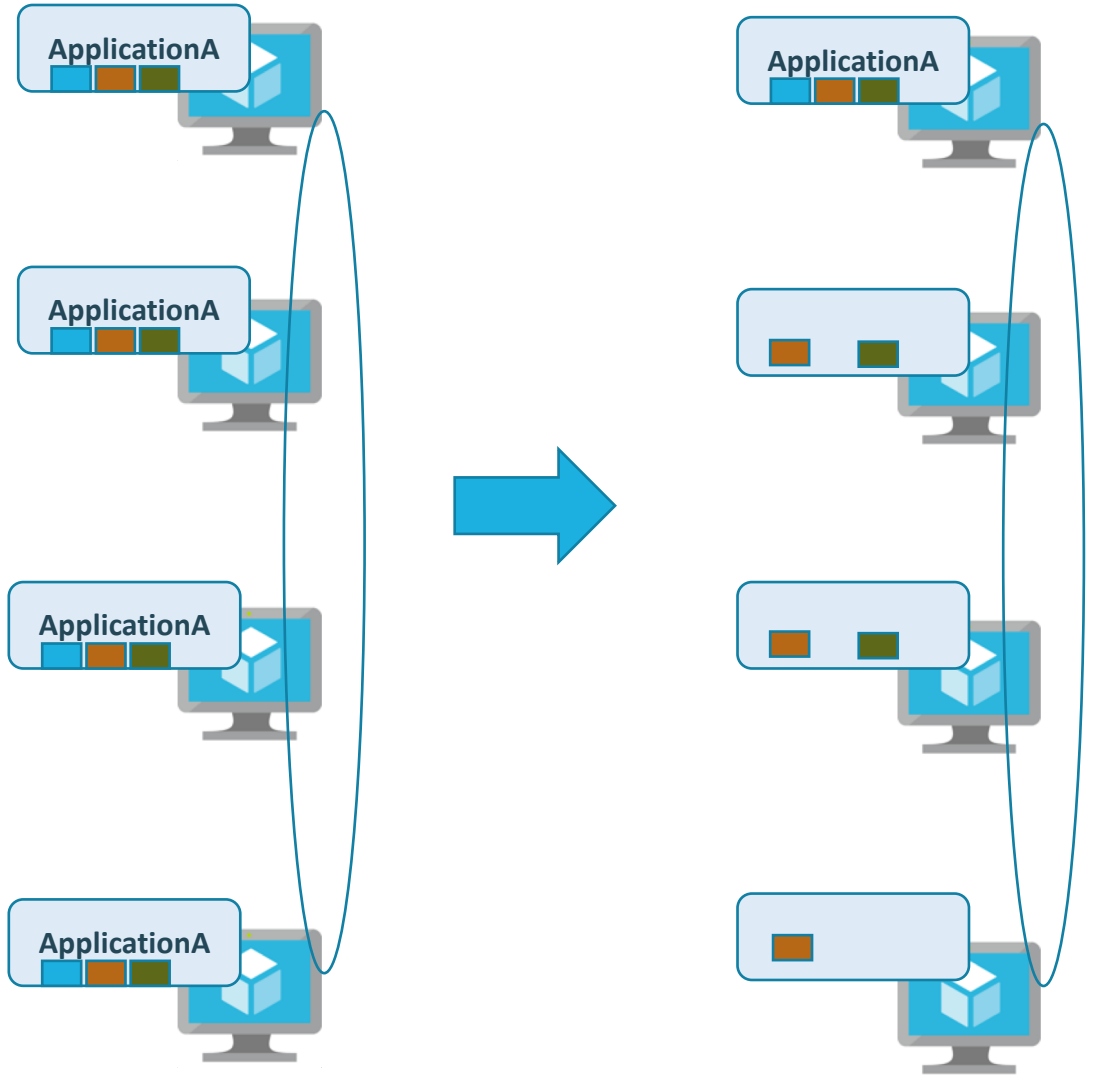



Redirect the requests:

- Based on network traffic
- Based on application level information

# 3.c.4 To enable service load balancer

- From a monolithic application to micro service design
- Suitable granularity
-



# 3.c.5 Exercises after the course

- Create a RESTful service
- Automate the deployment using Ansible
- Wrap as a dock container
- Deploy a Docker on the VM in the cloud
- Deploy a Kubernetes cluster on the VM in the cloud
- Deploy the Docker on the cluster

**http://tiny.cc/phbasz**

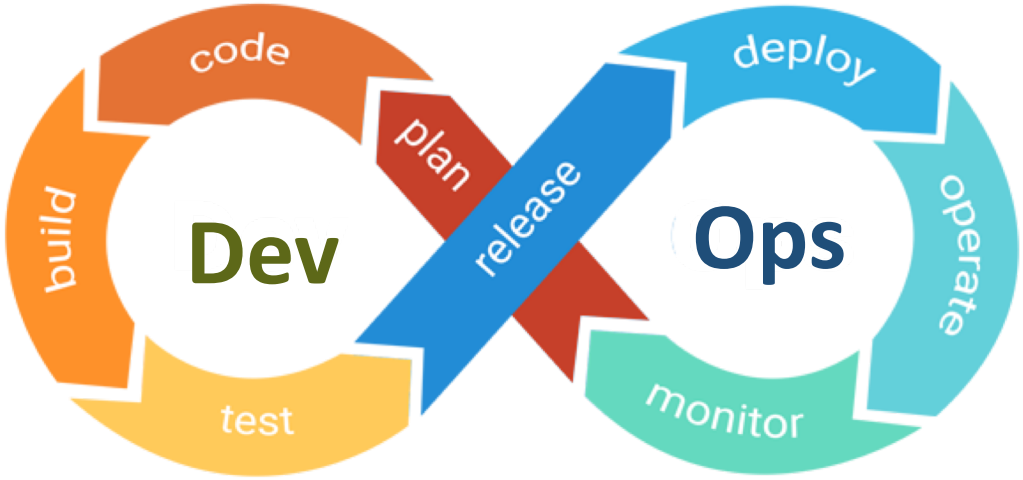
# Outline

1. What is Cloud computing and the Cloud?
2. How does the Cloud work?
3. How to use the Cloud?
   a) Use basic infrastructure service
   b) Run a local application in Cloud
   c) Run complex applications in Cloud
   **d) Infrastructure automation for operating online services**
4. Some Cloud related R&D activities in ENVRI
5. Summary

# Business and ICT aspects of services

**Software implementation of services:** e.g., Service Oriented Architecture (SOA), web services, micro services, business process modelling (BPM), etc.

**Agile**, incremental, iterative




**Business value of services**: e.g., market, number of customers, transactions, etc.

**Continuous** (integration, testing, deployment and operation)

# Outline

1. What is Cloud computing and the Cloud?
2. How does the Cloud work?
3. How to use the Cloud?
   a) Use basic infrastructure service
   b) Run a local application in Cloud
   c) Run complex applications in Cloud
   d) Infrastructure automation for operating online services
4. **Discussion**
5. Summary

# 4.1 Typical questions

"I **manage** a data center, I often **get many requests** for running workfl big data applicatio sometimes I can all those reques

"I have problem to train **a deep neural network** using **large** data sets

**www.menti.com**

**39 25 69**

and I can not ver data products in time"

sever

from my oo big to ocal PC"

# Outline

1. What is Cloud computing and the Cloud?
2. How does the Cloud work?
3. How to use the Cloud?
   a) Use basic infrastructure service
   b) Run a local application in Cloud
   c) Run complex applications in Cloud
   d) Infrastructure automation for operating online services
4. Discussion
5. **Summary**

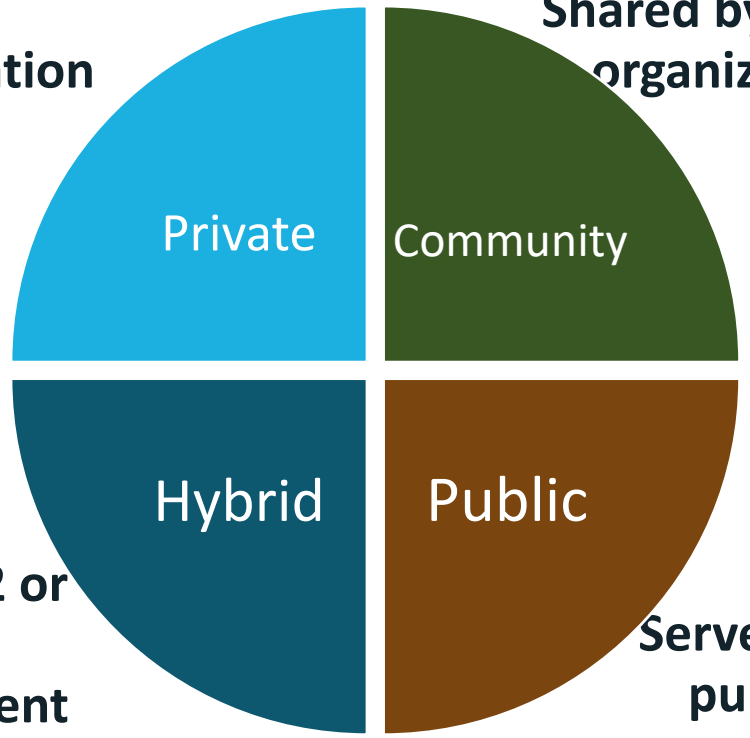
# 5 Summary

Virtual Virtual Virtual

**Single organization**

Private used data center, e.g. for single institute

**Shared by several organizations**

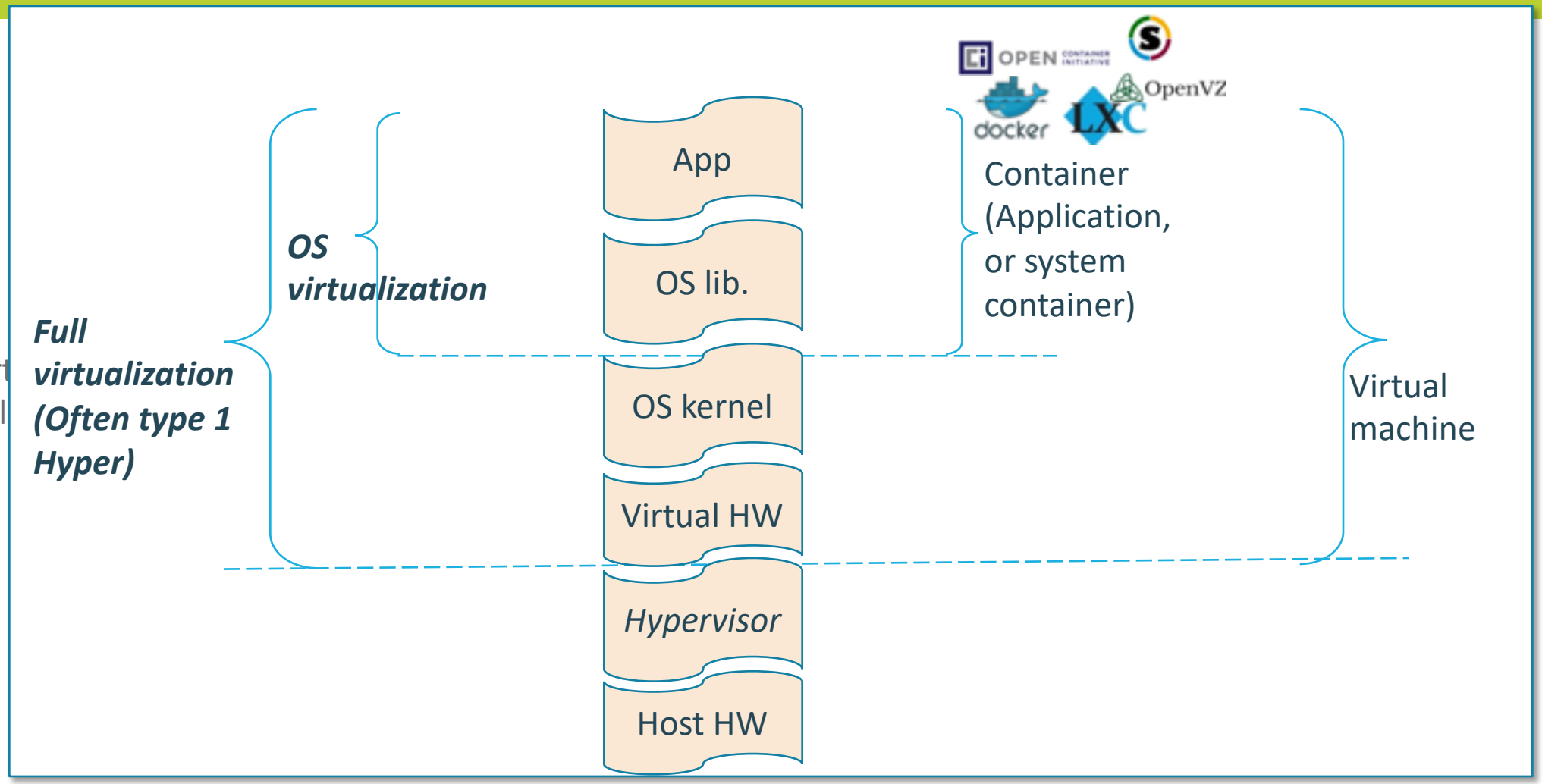



**Mix of 2 or more different types.**

**Serve public use**

See more examples from RIs

# 5 Summary


Vert
scal
Full virtualization (Often type 1 Hyper)
OS virtualization
App
OS lib.
OS kernel
Virtual HW
Hypervisor
Host HW
OPEN
docker
LXC
OpenVZ
Container (Application, or system container)
Virtual machine

# 5 Summary

**Agile**, incremental, iterative

**Software implementation of services**: e.g., Service Oriented Architecture (SOA), web services, micro services, business process modelling (BPM), etc.


**Business value of services**: e.g., market, number of customers, transactions, etc.

**Continuous** (integration, testing, deployment and operation)

# After the course: http://tiny.cc/phbasz

- Practices (with support for one week)
  - Some exercises (try the Cloud IaaS, RESTful, Docker and Kubernetes)
  - Thanks EOSC-HUB and EGI (through the early adoption program) for providing the lab environment
- You are welcome to try:
  - Contact: z.zhao@uva.nl for details

# After the course: http://tiny.cc/phbasz

## Tutorial for the Webinar "An introduction to Cloud computing"

Speaker: dr. Zhiming Zhao, Support: dr. Spiros Koulouzis,
University of Amsterdam, Amsterdam, NL
LifeWatch ERIC, vLab & Innovation Center, Amsterdam, NL

The tutorial is part of the webinar "an introduction to Cloud computing", in the ENVRI community winter school 2020. In this tutorial, you will learn how to define a simple REST service using OpenAPI. You will also learn how to use Ansible and Kubernetes, a.k.a K8s to deploy the RESTful Web Service on a VM in Cloud environments.

We sincerely thank dr. Giuseppe Larocca and dr. Andrea Manzi from EGI to provide the testbed via the EGI training platform. The tutorial is supported by the EOSC early adopter program via ENVRI-FAIR project, and LifeWatch-ERIC. The testbed will be accessible after the webinar for 10 days; during those days we will also provide support for all technical questions.

### 0. Before you Begin

**Install Ansible on local machine (laptop)**

You will need Ansible for the assignment. Please install it on your local computer based on the following instructions:

**What you need:**

1. A laptop
2. Follow instruction to get a VM (from EGI)
3. Follow the tutorials.

**The VM will be available for 10 days after the webinar.**