Containers & Virtualization

Emmanuel Druon

centralelille
IG2I

Global organization

- Module #1
- Virtualization: VM / Clusters / Docker / Network / Storage
- Cloud concepts
- Presentation of different Cloud offers
- Module #2: Instances & Networking
- Lab: Create a simple EC2 instance
- Lab: Build a network infrastructure with a VPC
- Module #3: Cloud Storage & Containers
- Introduction to the main types of storage in AWS
  - Lab: S3 Lab
  - Lab: EBS Lab
- A short introduction to Docker containers
  - Lab: Create a Web Service based on an “nginx” container
- Module #4: A short introduction to DevOps
  - Lab: Working with Ansible

centralelille IG2I

Group work

- In groups, try to define:
  - What cloud computing is?
  - Pros & Cons?
  - Types?
- Timing: 15 minutes

centralelille IG2I

“ Cloud computing is massively based on virtualization technology ”

BUT

Cloud computing is more than virtualization

Cloud Computing?

- Technically, Cloud Computing is based on proposing resource pools through:
  - Abstraction: virtualization,
    - Machine virtualization,
    - Network virtualization,
    - Storage virtualization.
  - Automation: orchestration,
    - Provision,
    - Deprovision,
    - Resize.

centralelille IG2I

“Classical” virtualization vs Cloud computing

- “Classical” virtualization:
  - Abstraction,
  - Administrator oriented,
  - Not self-service,
  - Poor elasticity.
- Cloud computing:
  - Abstraction,
  - Orchestration of resource pools,
  - Self-service.

centralelille IG2I

Cloud computing definition

- National Institute of Standards and Technology's (NIST) definition:
  - *”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."*

centralelille IG2I

Benefits of Cloud Computing

- Agility:
  - Possibility to quickly provision & deprovision resources,
- Resilience:
  - Load-balancing,
  - Duplication.
- Cost:
  - Pay for use,
- Security.

centralelille IG2I

Virtualization in a nutshell

centralelille
IG2I

Virtualization in a nutshell

Machine virtualization

centralelille
IG2I

Machine Virtualization

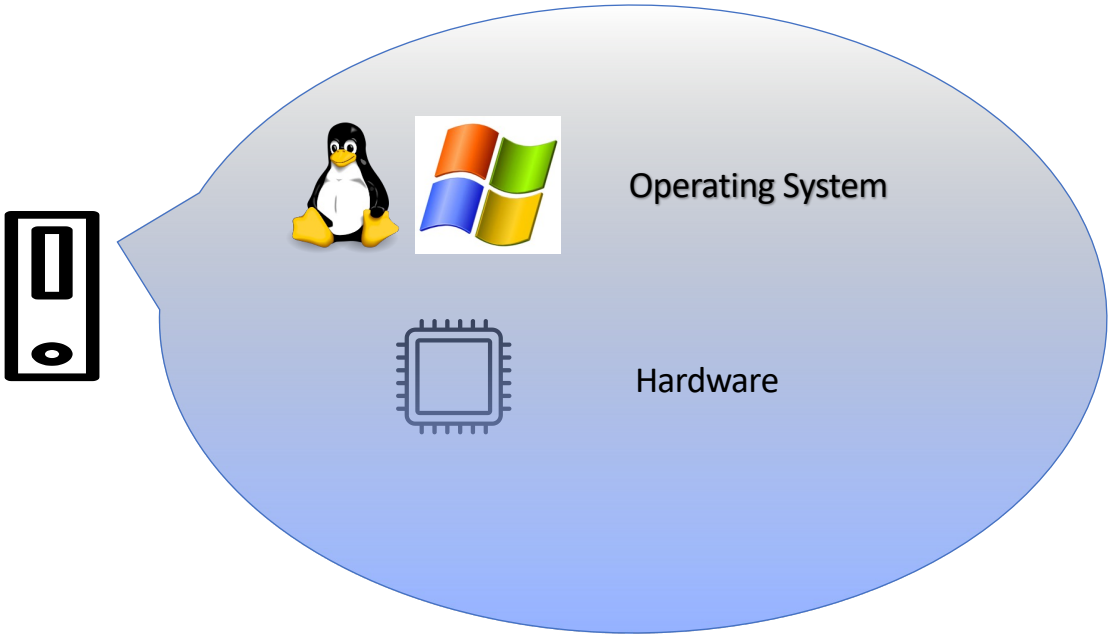
- Two distinct models:
  - Full scale Virtual Machines,
  - Containers.

centralelille
IG2I

Full scale VMs

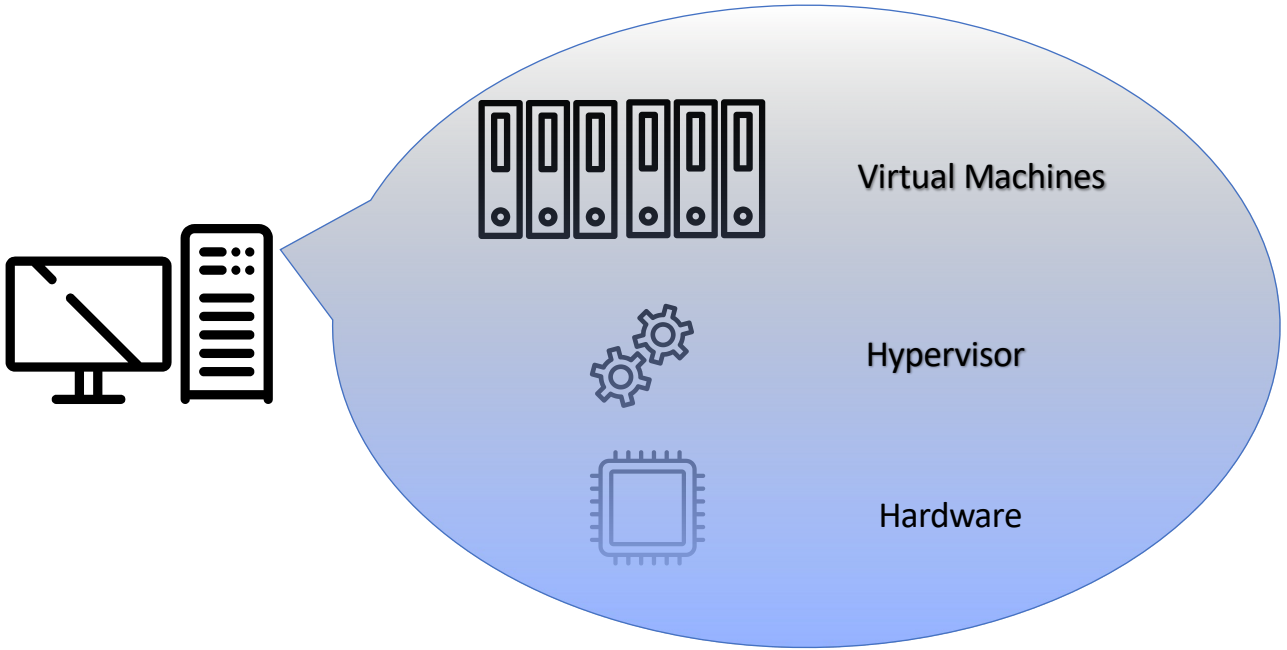
- Principles
  - The hardware is handled by a hypervisor,
  - Virtual hardware is software defined,
  - VMs are “executed” under the supervision of the hypervisor,
  - Each VM has its own OS,
  - VMs are isolated from one another.

Full scale VMs



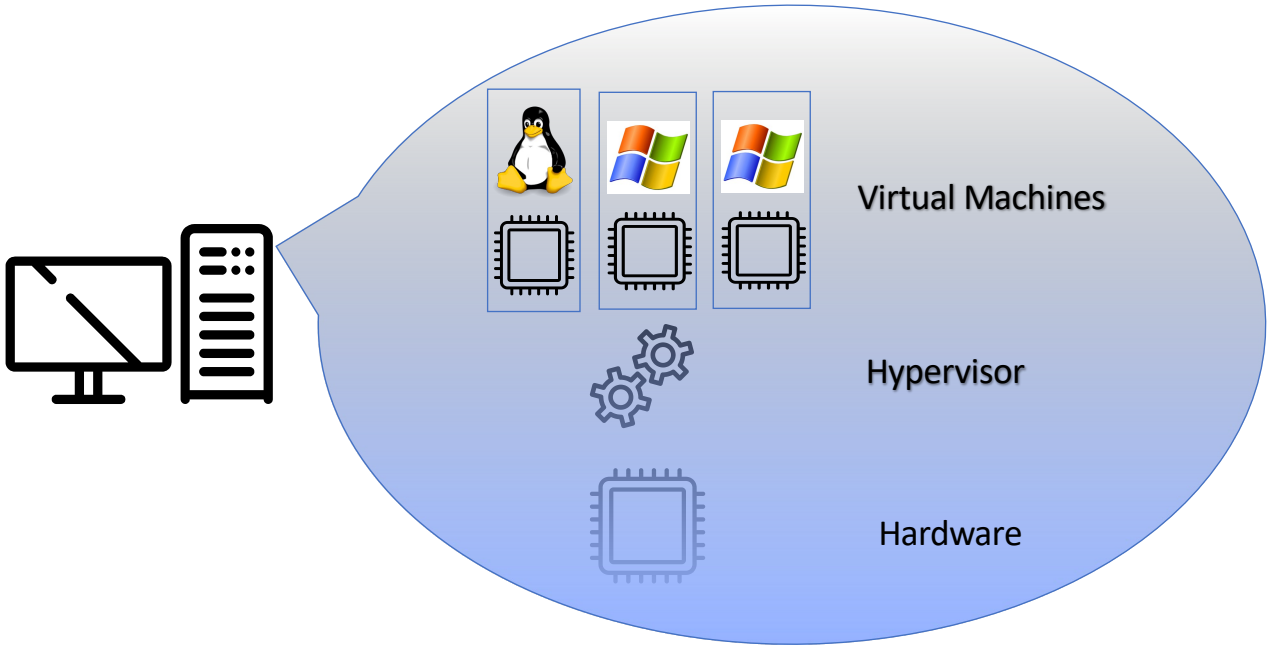
centralelille IG2I

Full scale VMs



centralelille IG2I

Full scale VMs



centralelille IG2I

A VM configuration file example

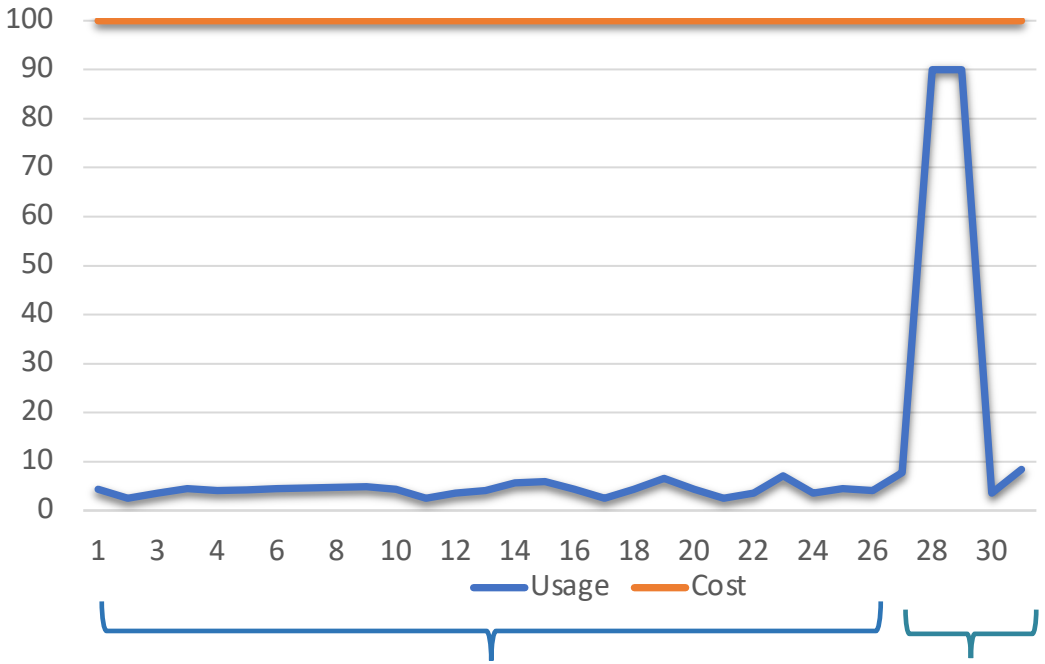

```
edruon@21-3910-001 temp % cat Debian\ 11.x\ console.vmx
usb.vbluetooth.startConnected = "TRUE"
firmware = "efi"
guestOS = "arm-debian11-64"
sound.autoDetect = "TRUE"
sound.virtualDev = "hdaudio"
cpuid.coresPerSocket = "1"
memsize = "2048"
nvme0.present = "TRUE"
nvme0:0.fileName = "Virtual Disk.vmdk"
nvme0:0.present = "TRUE"
sata0:1.deviceType = "cdrom-image"
sata0:1.fileName = "/Users/edruon/Downloads/debian-11.4.0-arm64-netinst.iso"
ethernet0.connectionType = "nat"
ethernet0.virtualDev = "e1000e"
ethernet0.generatedAddress = "00:0c:49:ef:5a:84"
```

Great! I got it!

- But… Do we need virtualization?

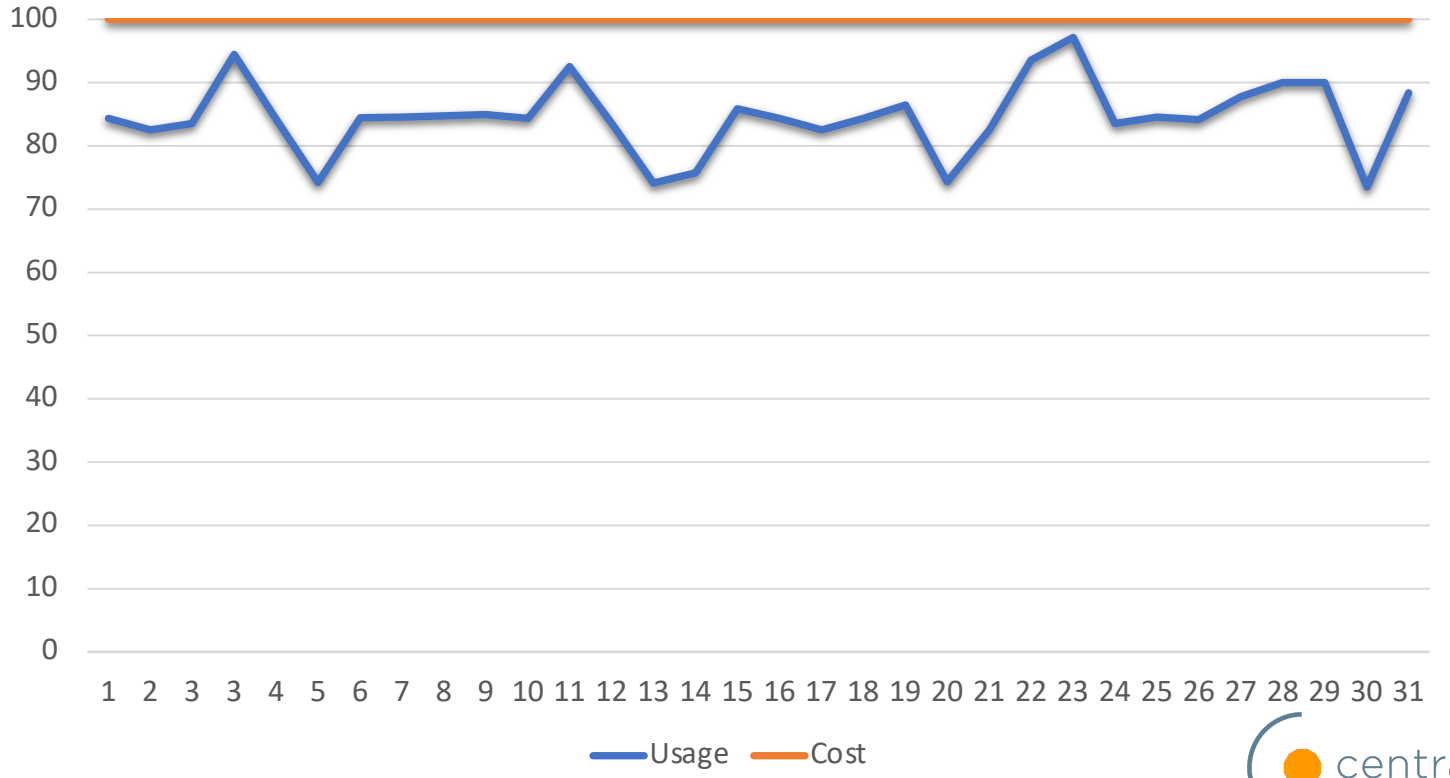
centralelille
IG2I

Why virtualization?



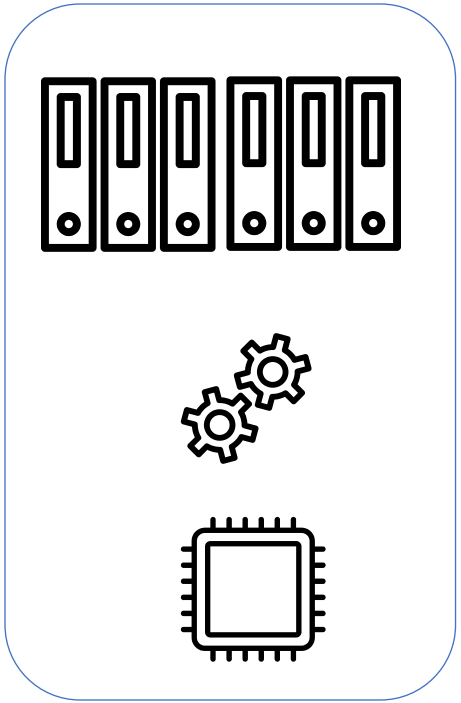
centralelille
IG2I

Why virtualization?



centralelille
IG2I

Some disadvantages of virtualization



- Raw performance loss
- An extra software layer
- A possible unique point of failure
- Additional technical expertise required

centralelille IG2I

Seriously?

- It’s not that good in fact...

centralelille
IG2I

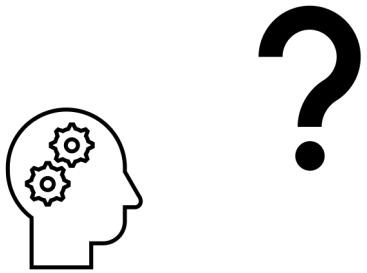
Seriously?

- It’s not that good in fact... Or is it?

centralelille
IG2I

Before moving on… A simple question

What are the “real life components”
of a Virtual Machine?



centralelille
IG2I

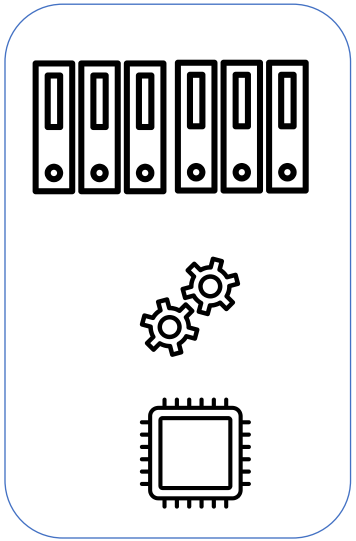
Before moving on… A simple question

- In real life, a VM is basically?
  - A set of files
  - A memory area on the server
- Files represent the non-volatile part of a VM:
  - *.nvram* file saves the BIOS of the VM
  - *.vmx* and *.vmxf* keep the configuration
  - *.vmsd* files are used for snapshots
  - *.vmdk* files are the virtual disks used by the VM
  - Additionally, *.vmem* and *.vmss* files can be found when the VM is suspended or if it crashed. They save the memory pages used by the VM and the VM state when suspended

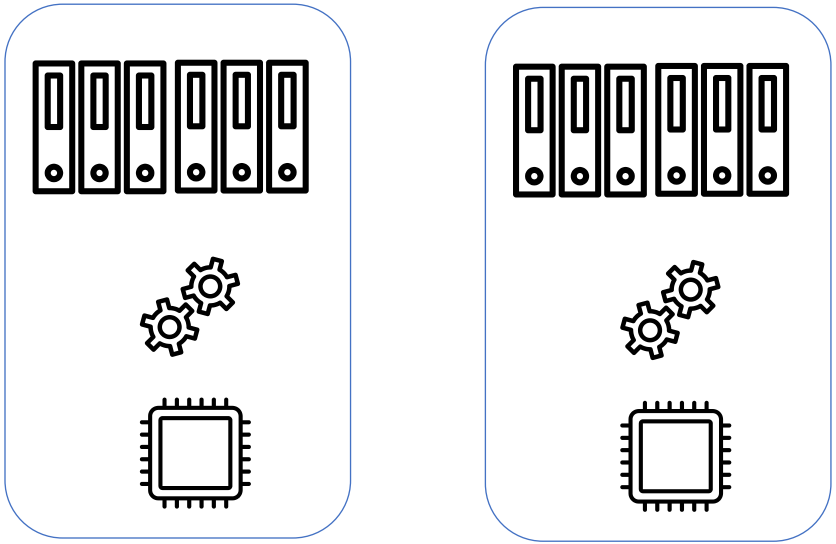
```
Debian 11.x console.nvram
Debian 11.x console.vmsd
Debian 11.x console.vmx
Debian 11.x console.vmxf
Virtual Disk-s001.vmdk
Virtual Disk-s002.vmdk
Virtual Disk-s003.vmdk
Virtual Disk-s004.vmdk
Virtual Disk-s005.vmdk
Virtual Disk-s006.vmdk
Virtual Disk.vmdk
vmware-0.log
```

centralelille IG2I

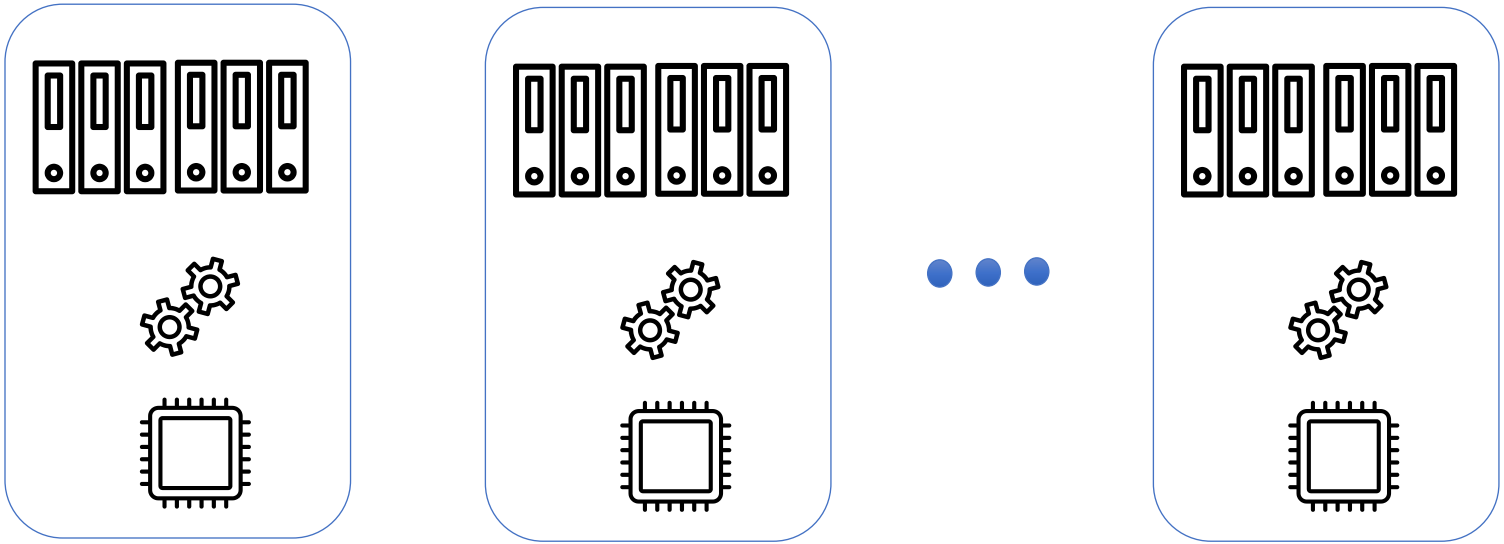
The cluster



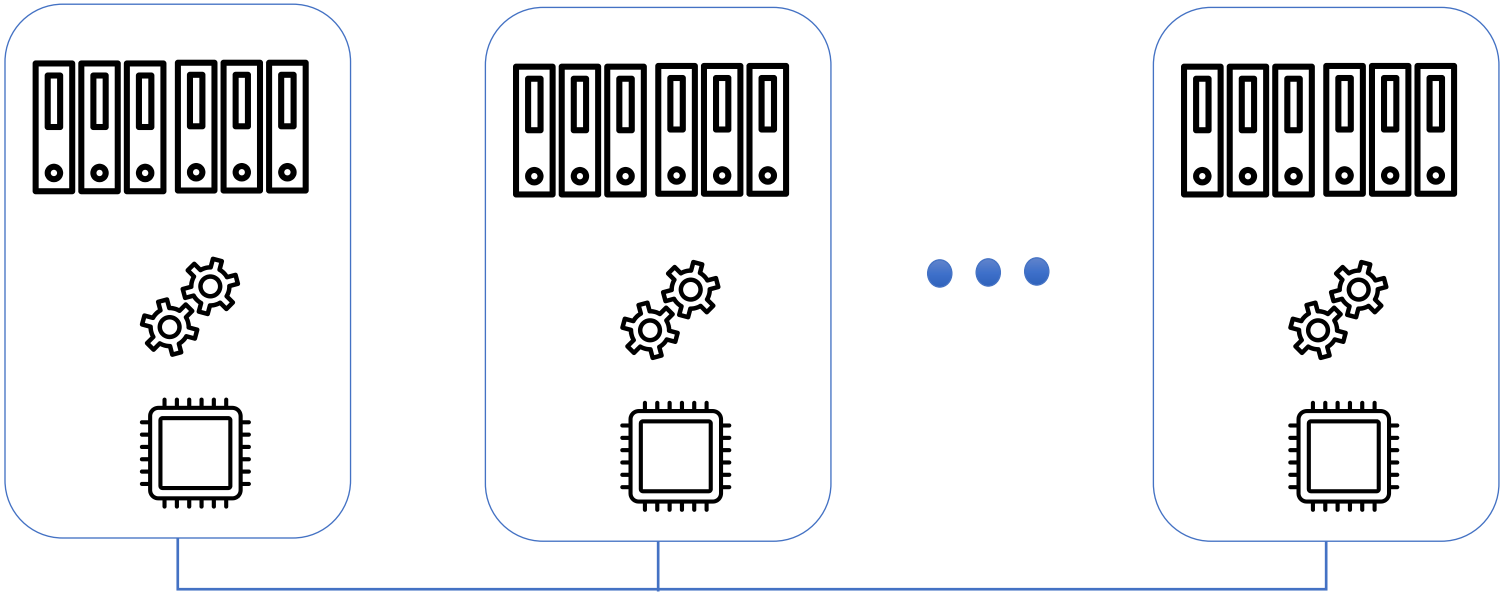
The cluster



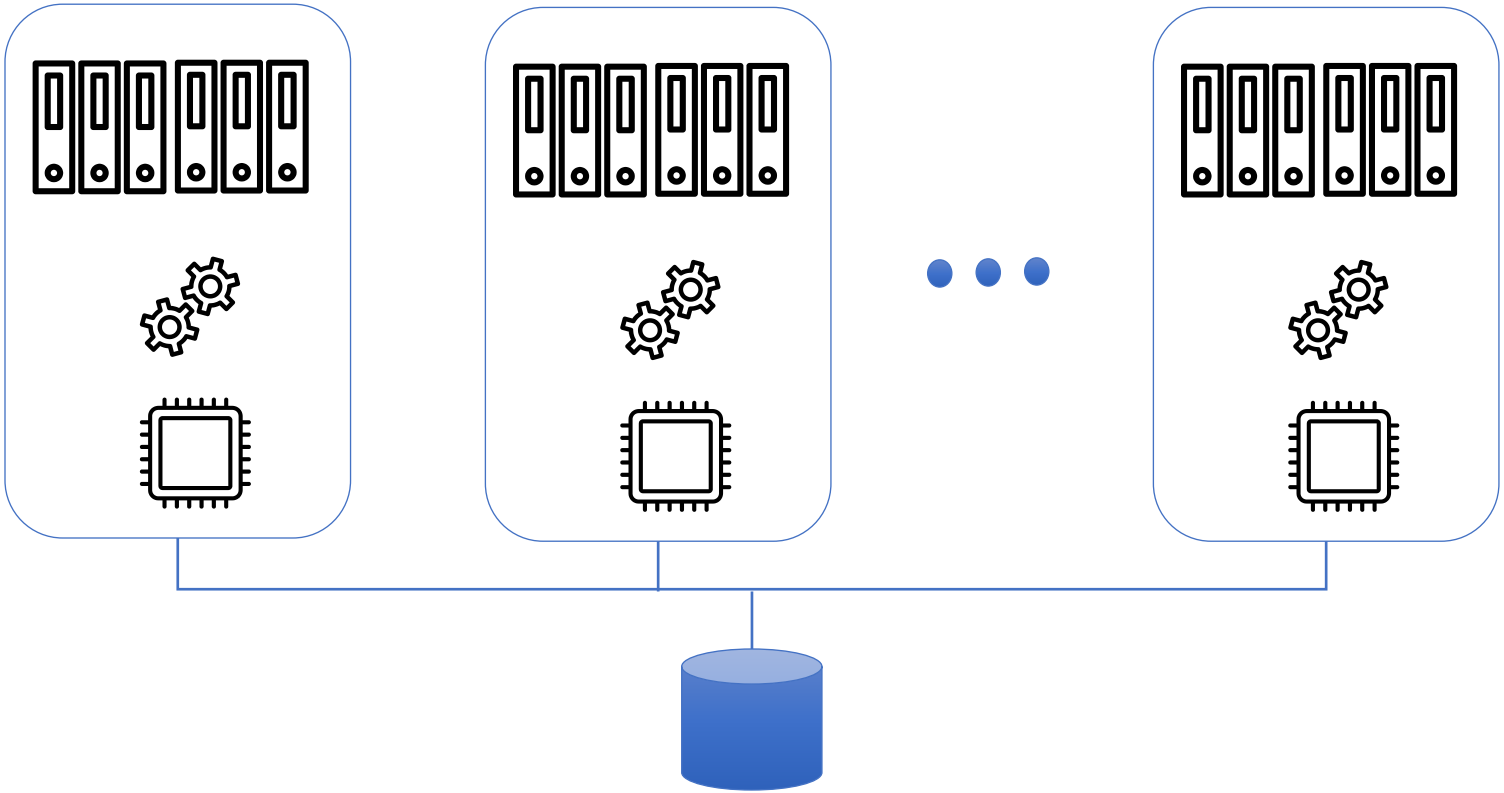
The cluster



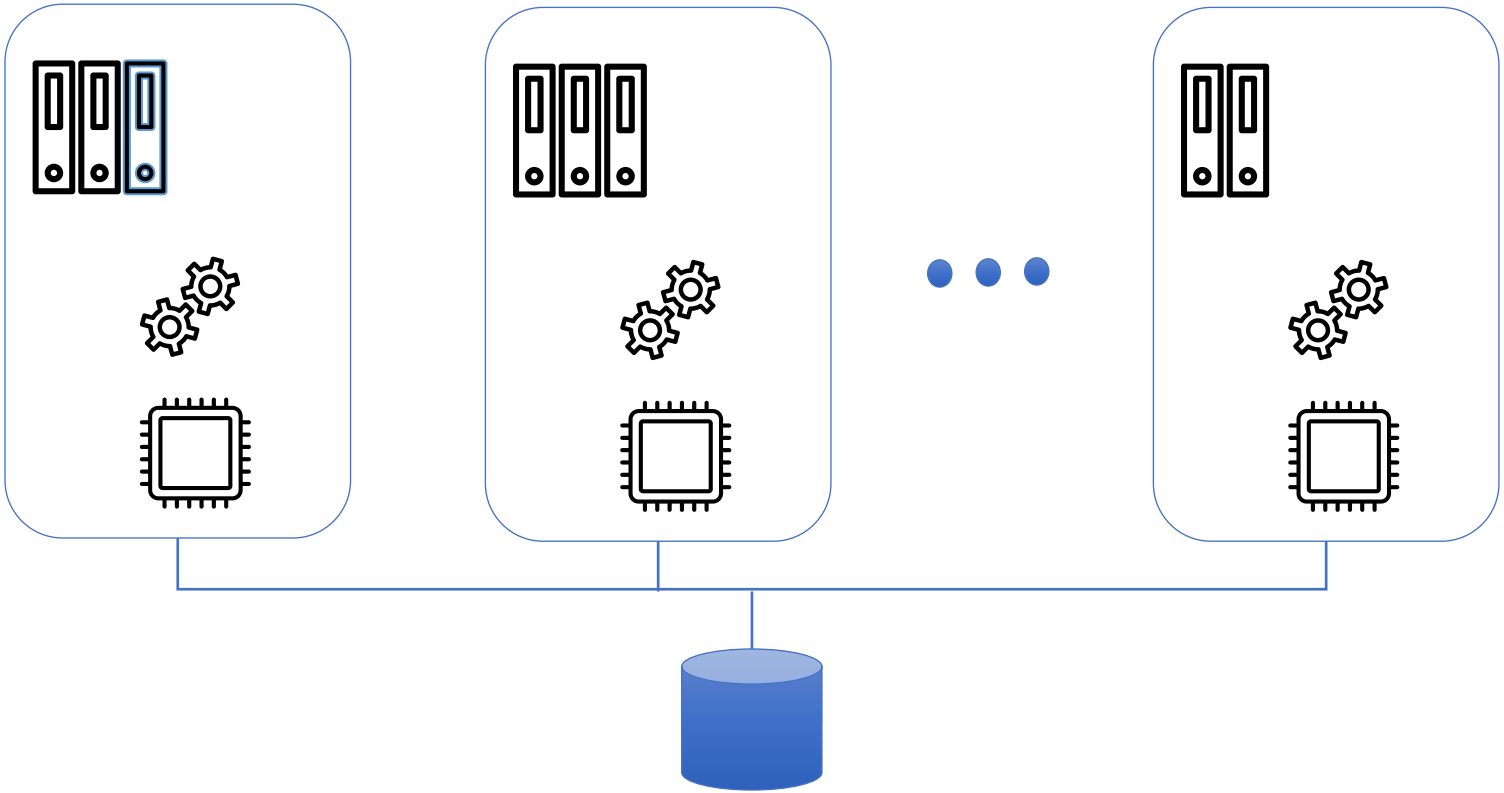
The cluster



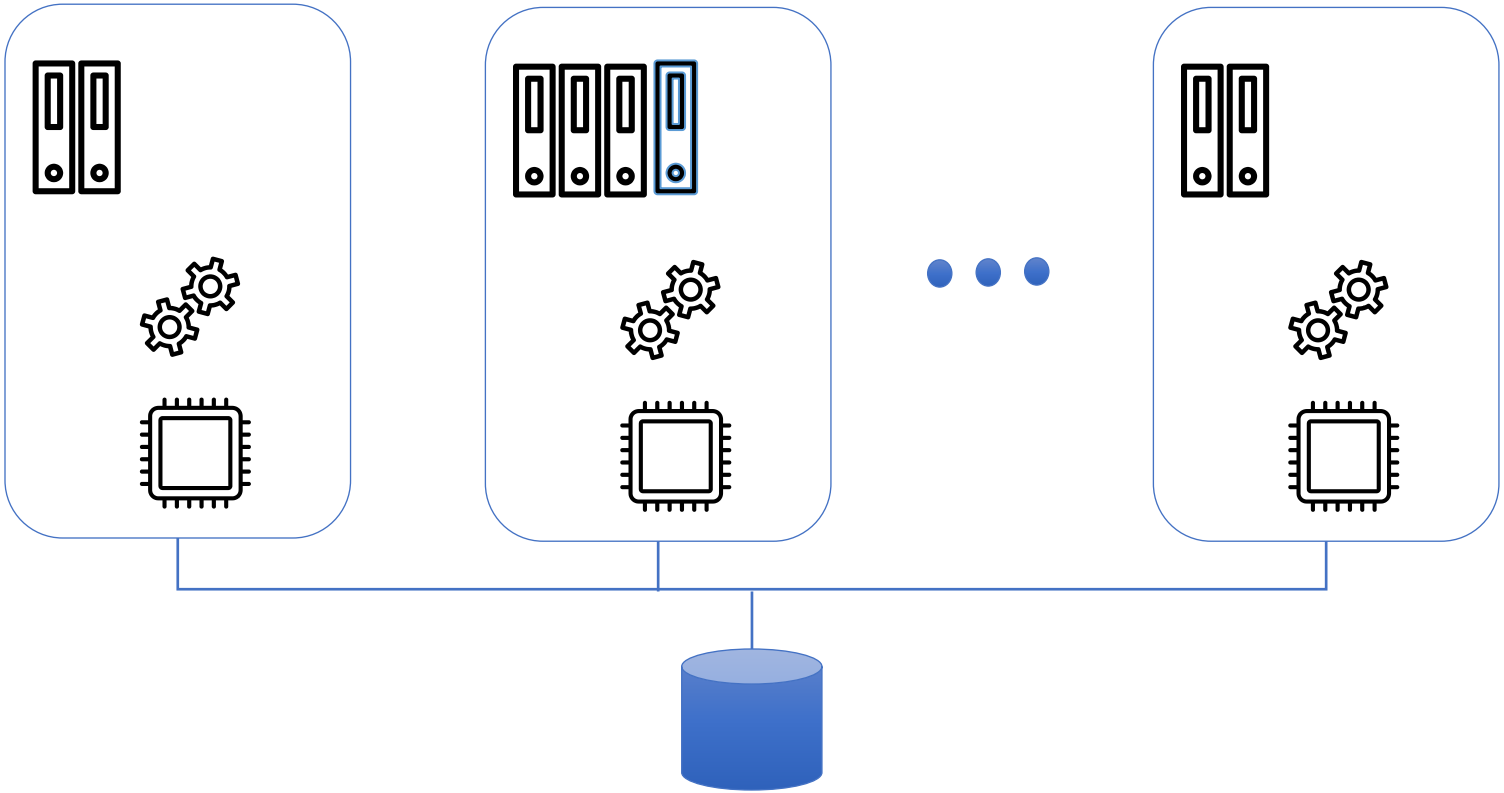
The cluster



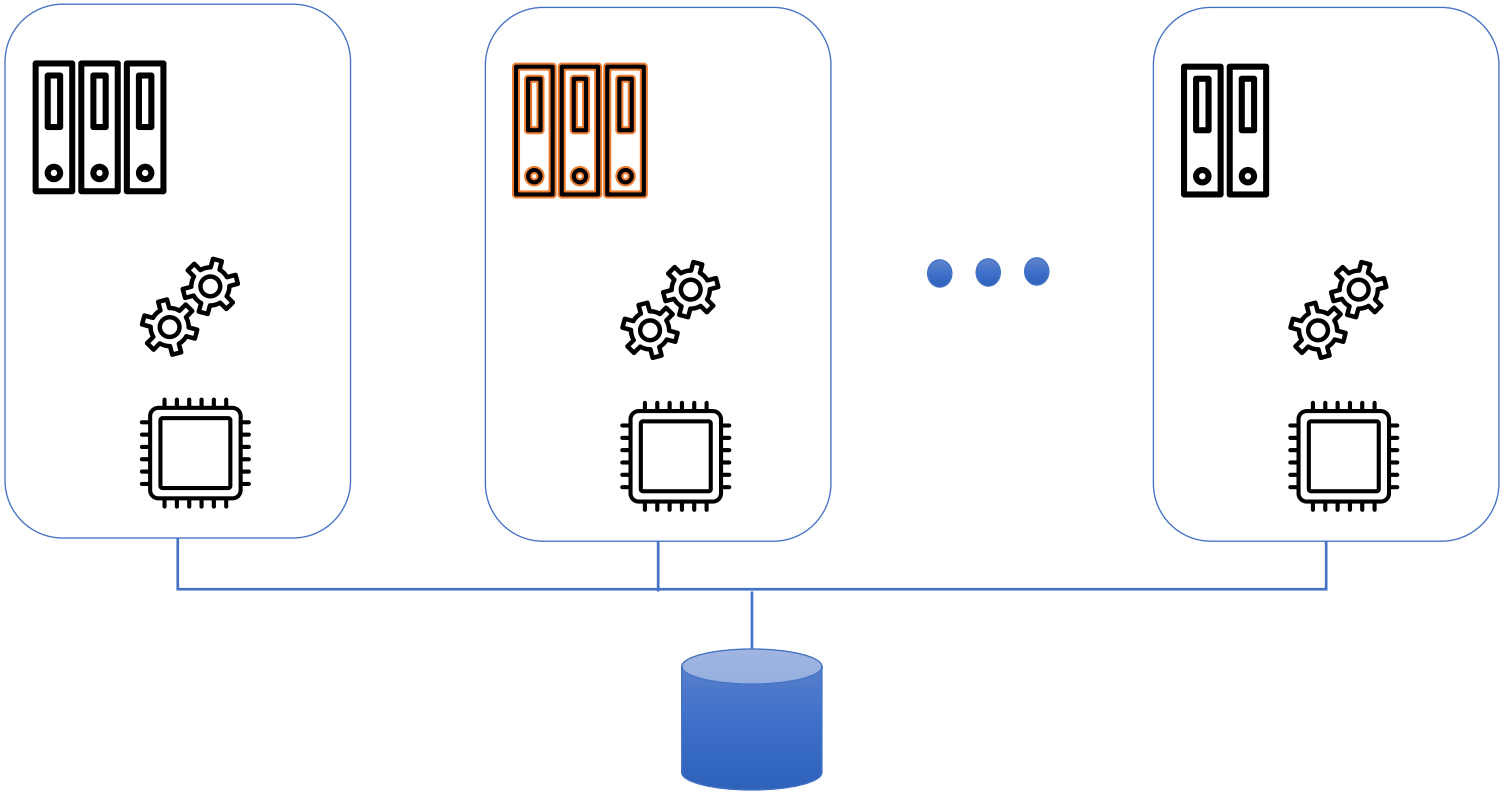
VM migration



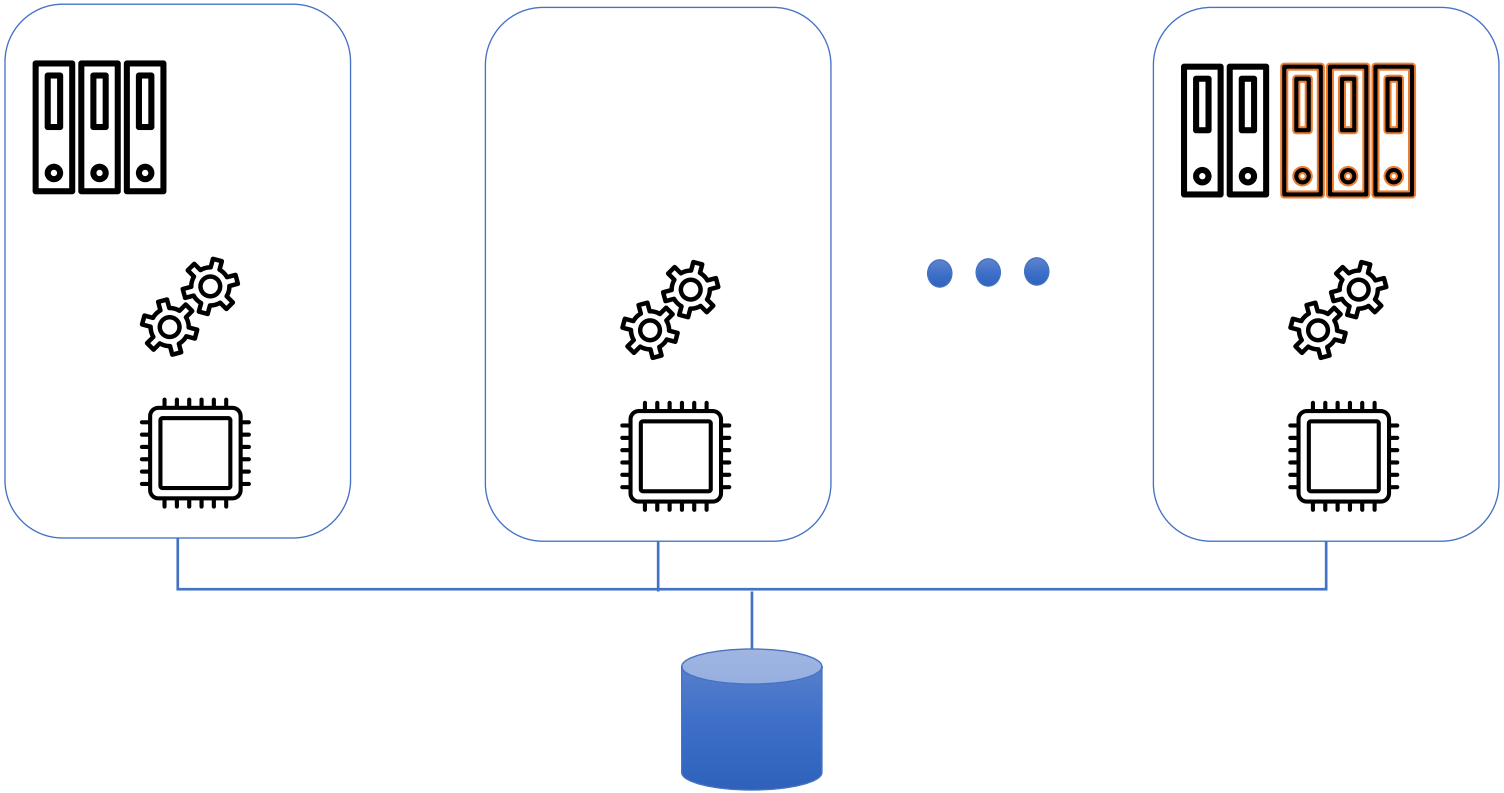
VM migration



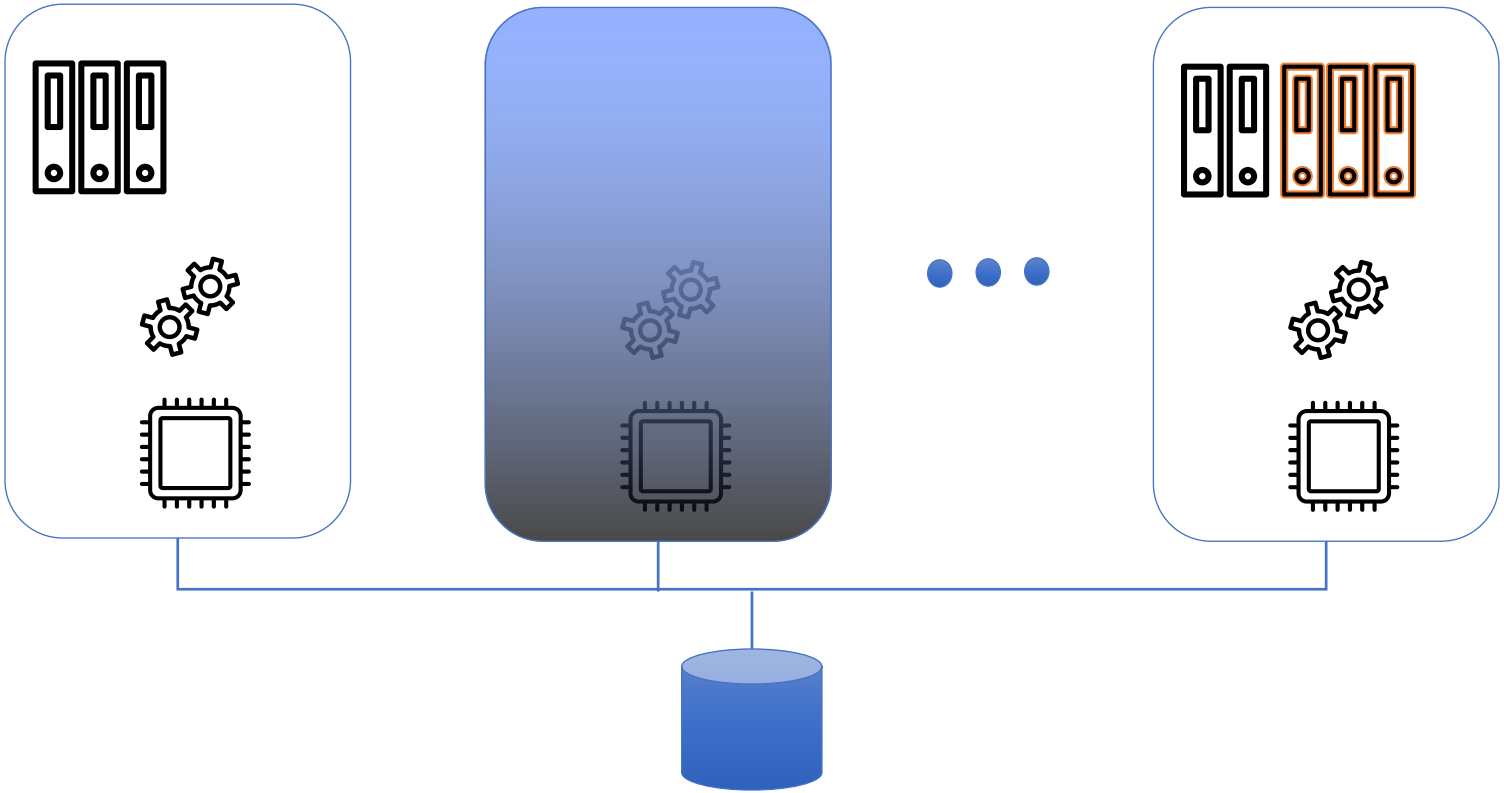
Physical server maintenance



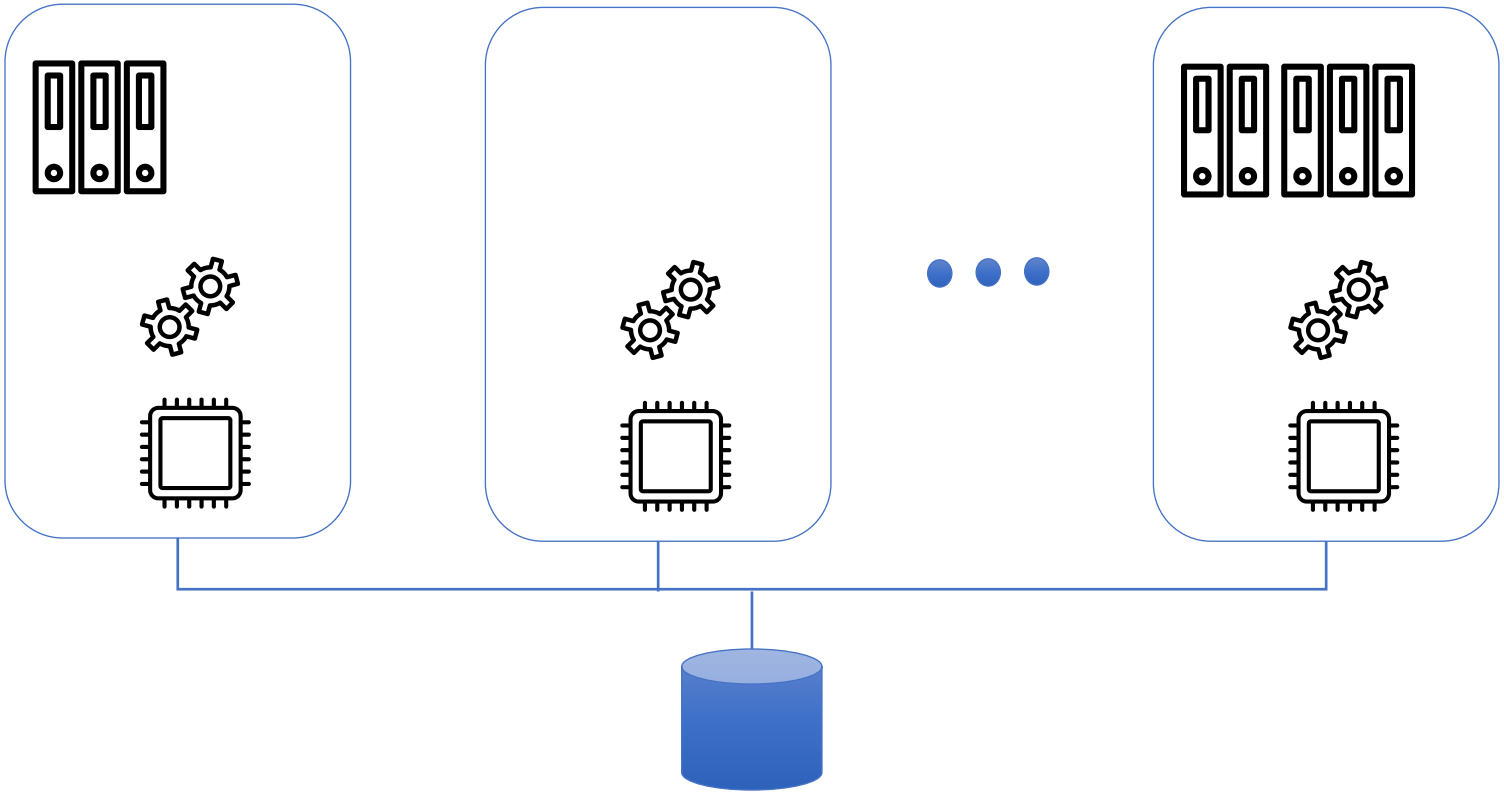
Physical server maintenance



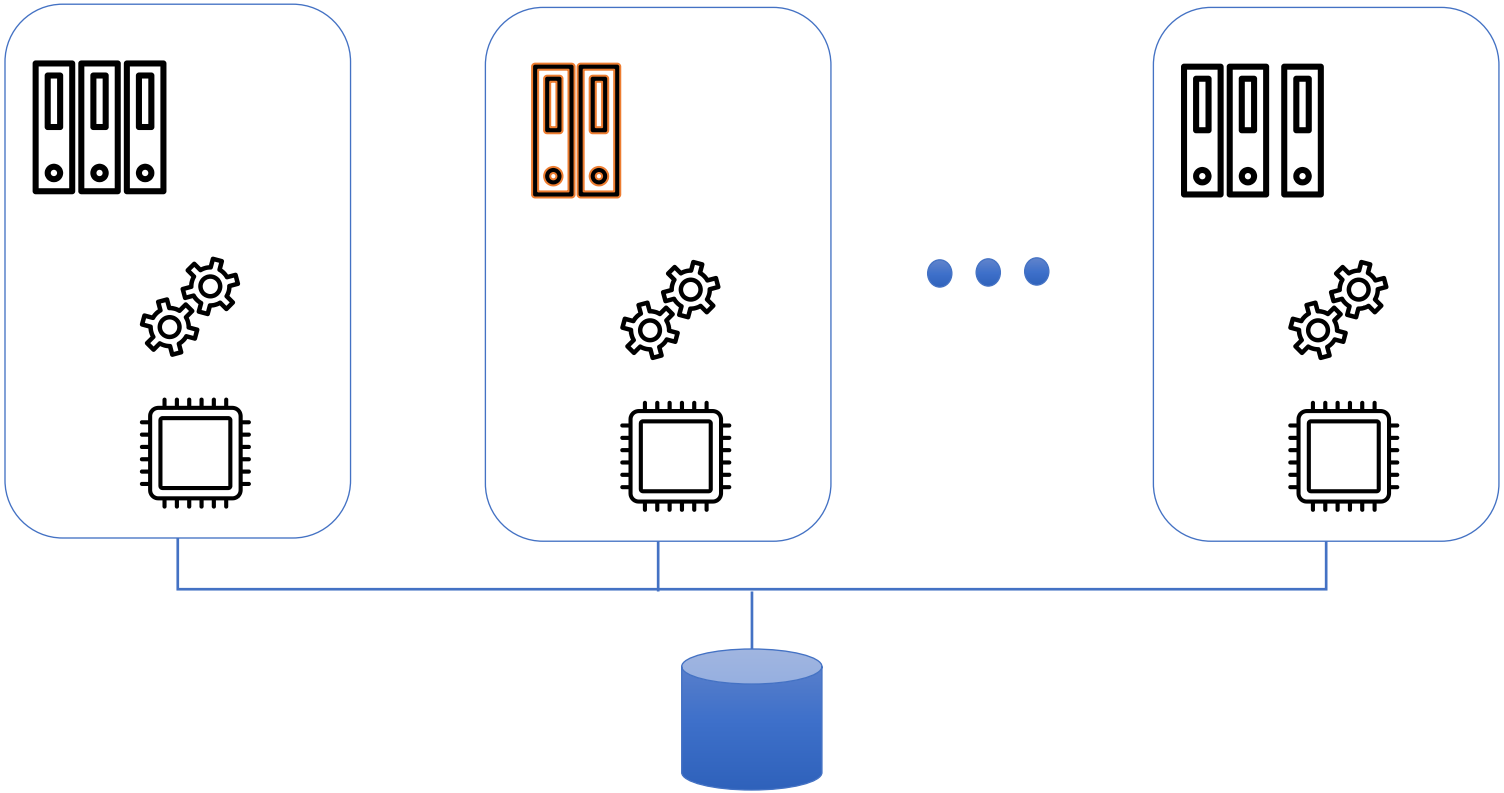
Physical server maintenance



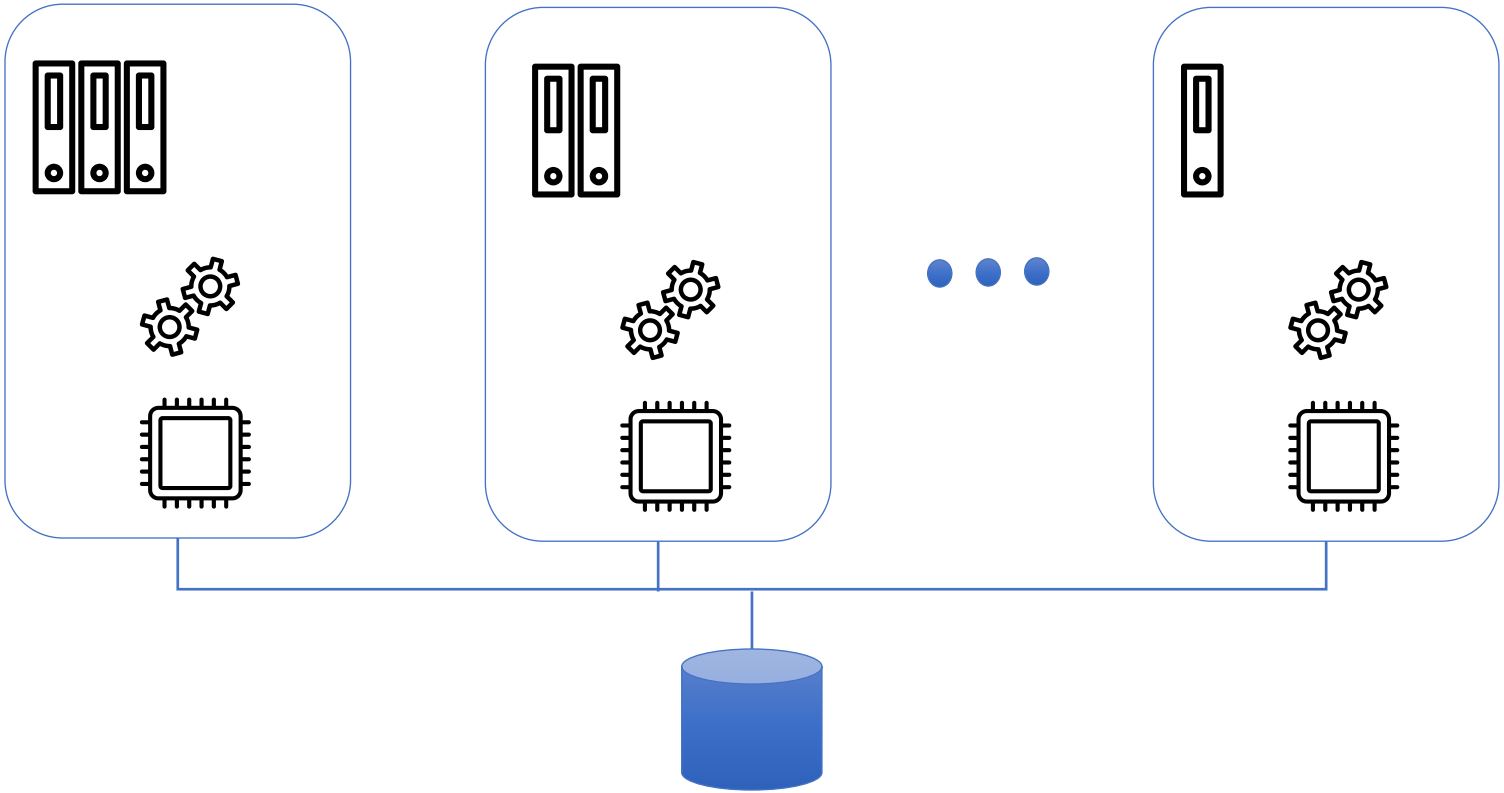
Load-balancing



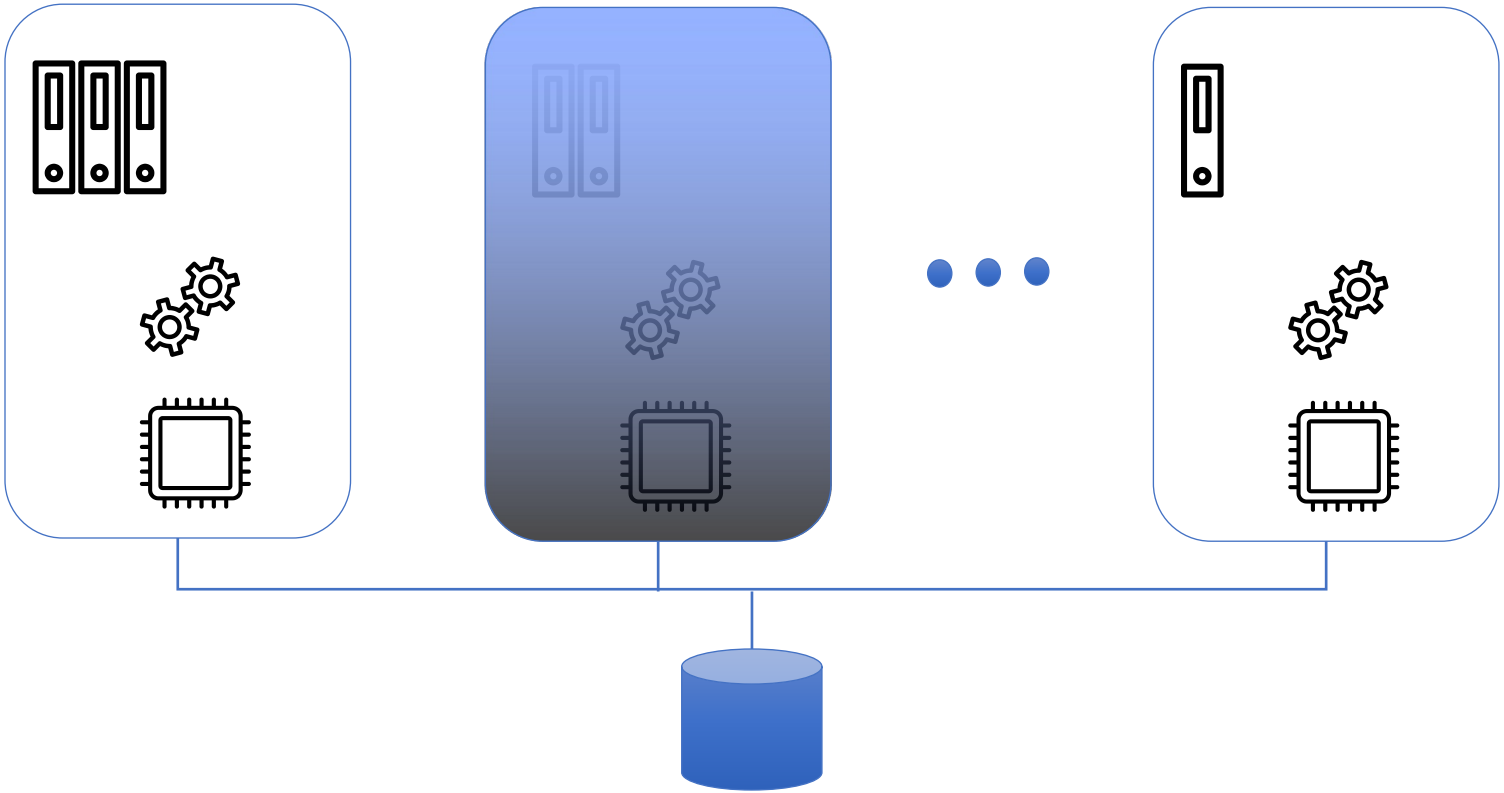
Load-balancing



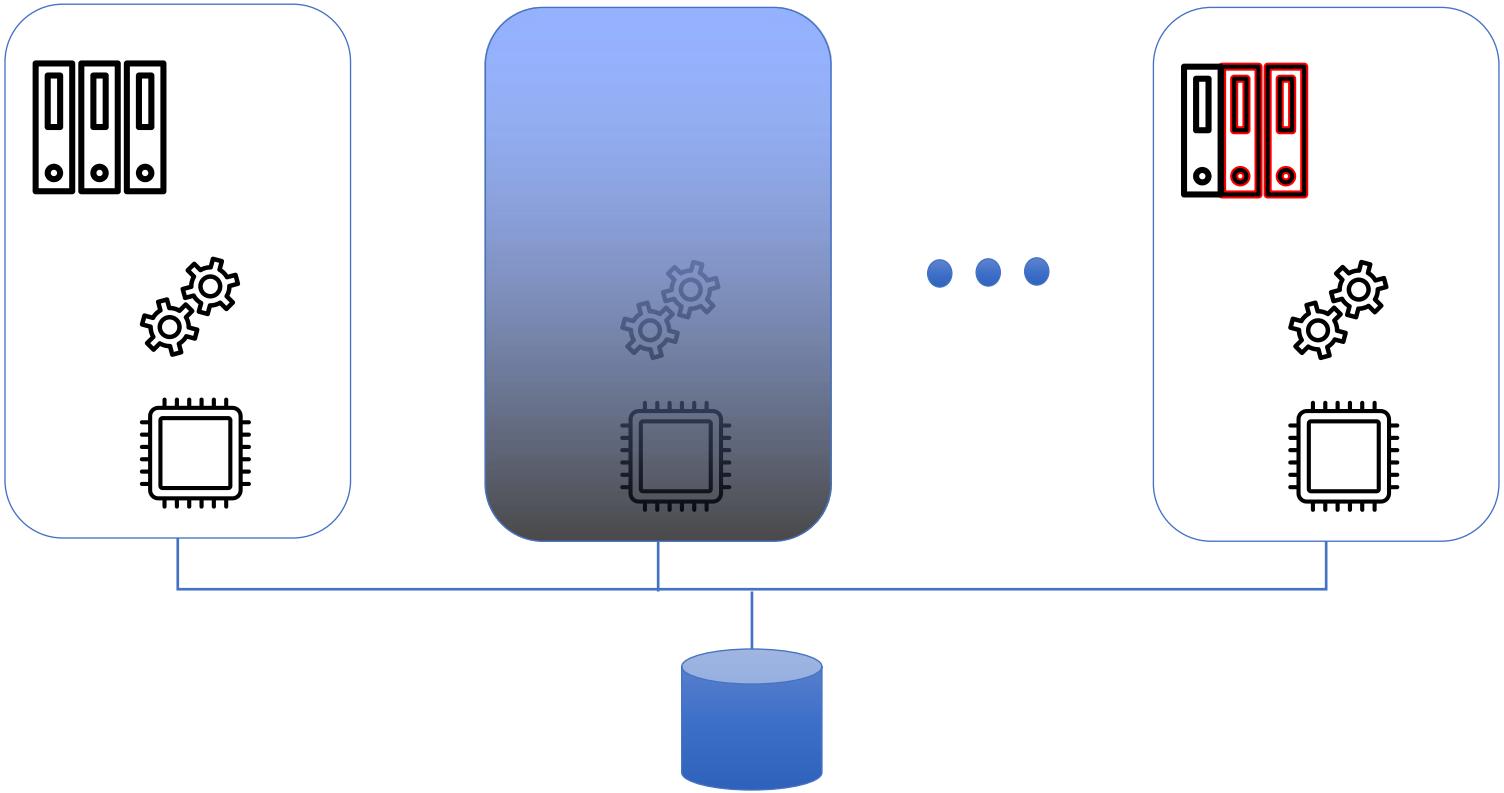
High-Availability (HA)



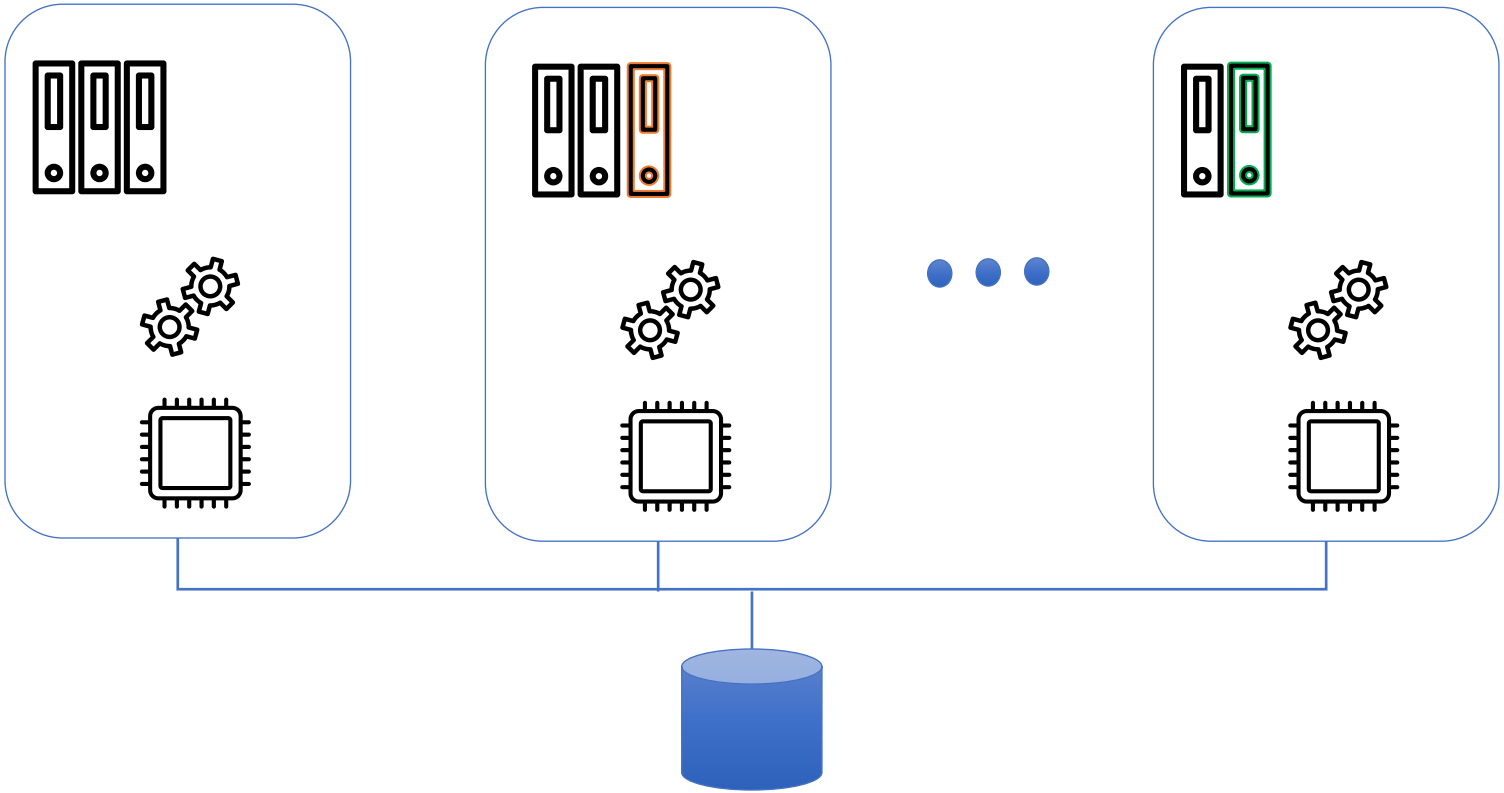
High-Availability (HA)



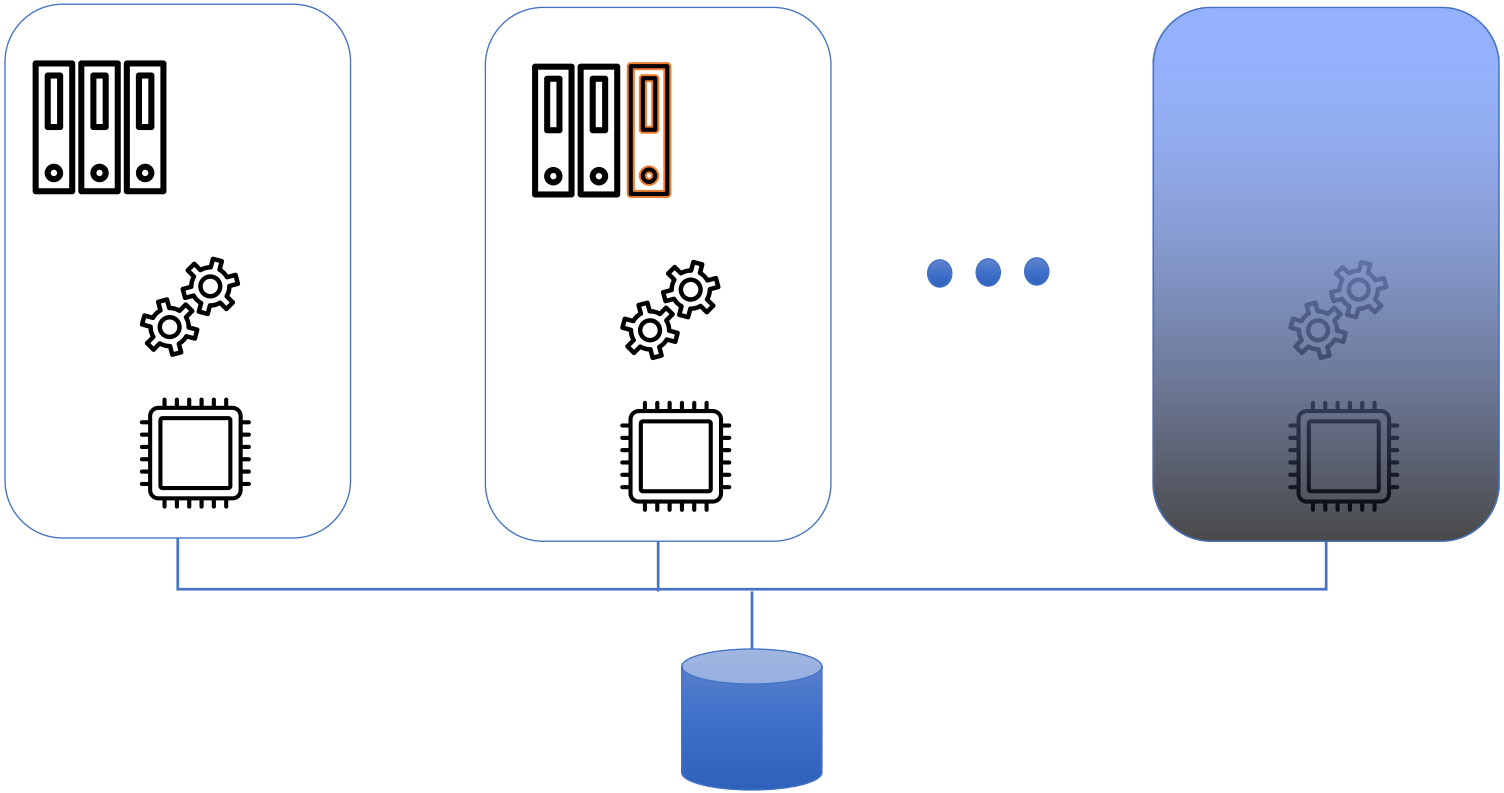
High-Availability (HA)



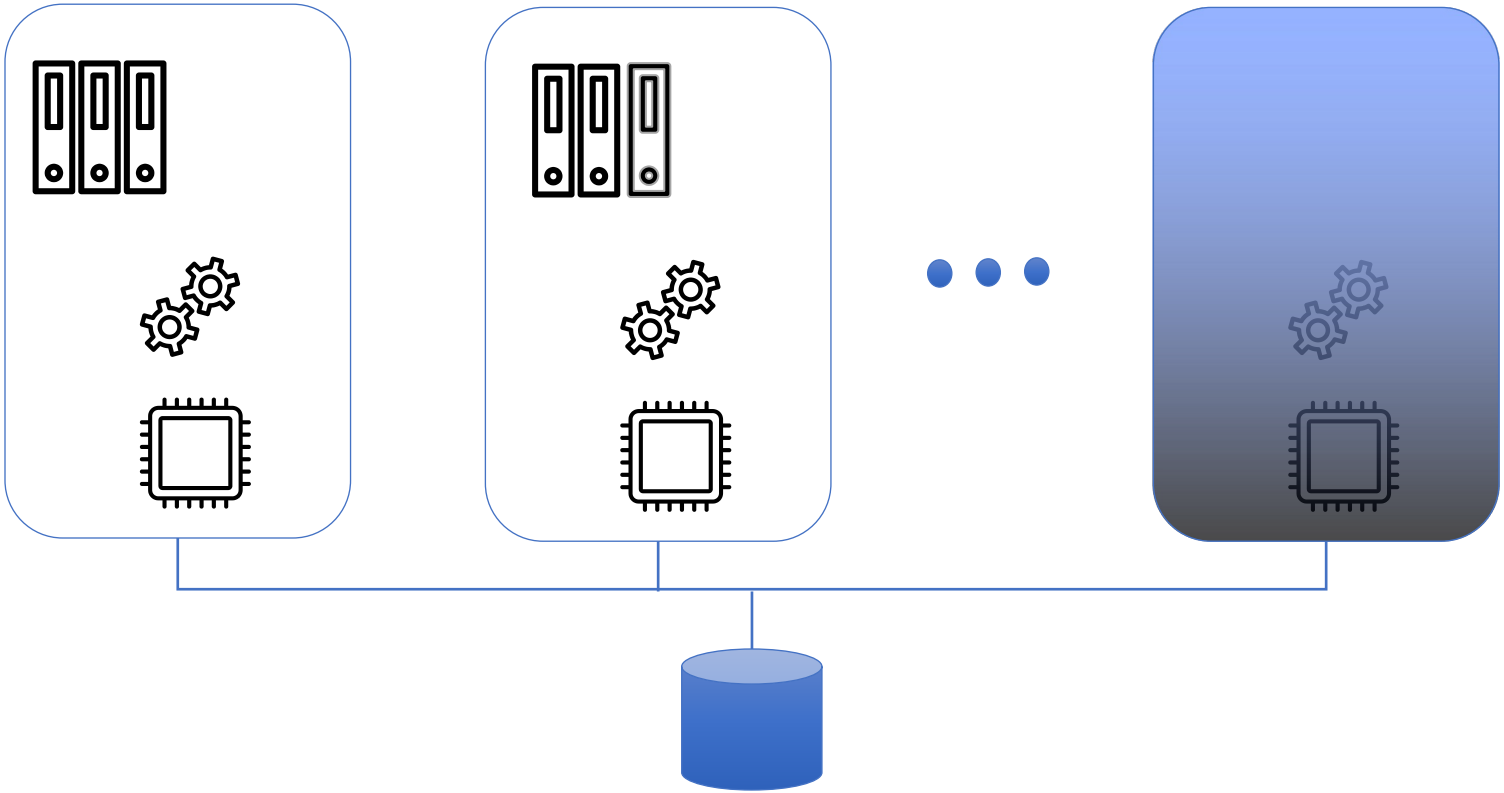
Fault-Tolerance (FT)



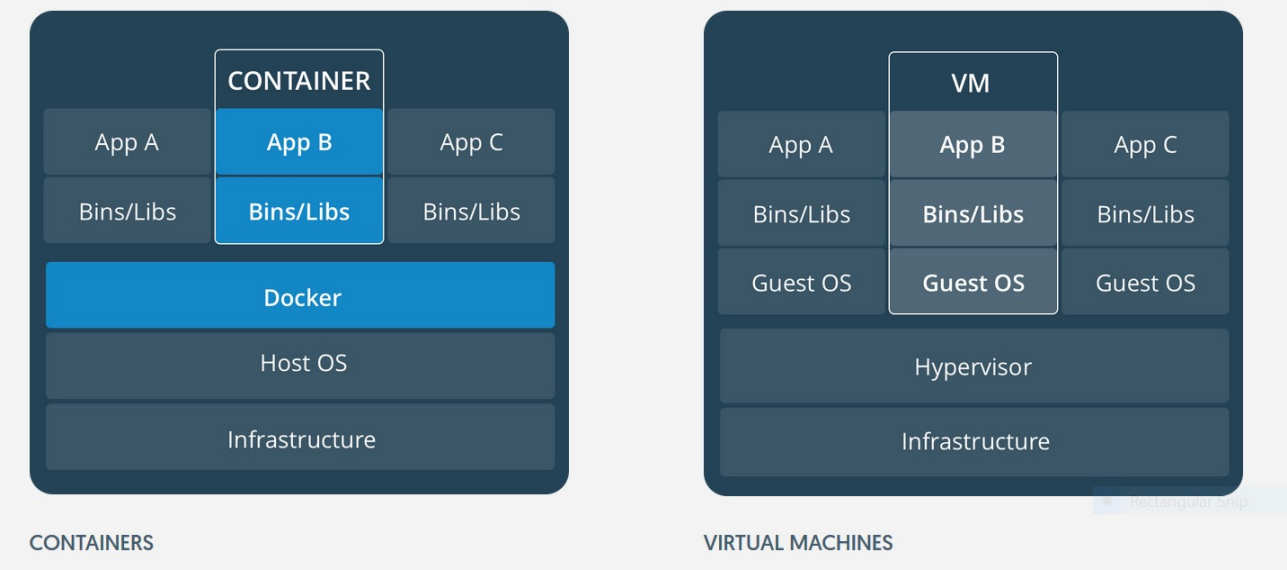
Fault-Tolerance (FT)



Fault-Tolerance (FT)



Containers


CONTAINER
App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Docker
Host OS
Infrastructure
CONTAINERS
VM
App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Guest OS
Hypervisor
Infrastructure
VIRTUAL MACHINES

Full scale VMs vs Containers

- Hypervisors are usually more resource management efficient
- Virtualization clusters have HA & FT features
- Containers are faster to start than VMs
- Containers are usually micro-service orientated
- By default, containers work in a stateless mode
- Containers are portable
- Containers can also be used for local needs (dev environments, sandboxes,…)
- Many orchestration tools exist for containers (Kubernetes, Docker Swarm)
- VMs can be moved between physical host without being stopped
- Web services can be provided through docker images installed in VMs

centralelille IG2I

Virtualization in a nutshell

Network virtualization

centralelille

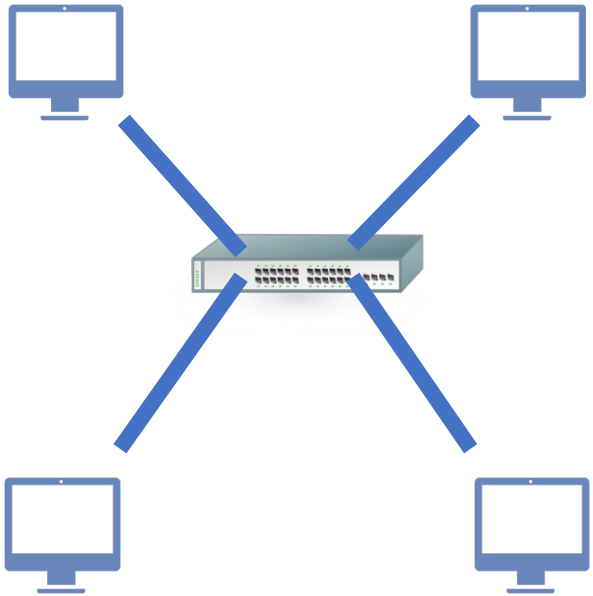
IG2I

Network Virtualization

- Physically based on VLANs
- “Virtually” based on Software Defined Networks (SDN)

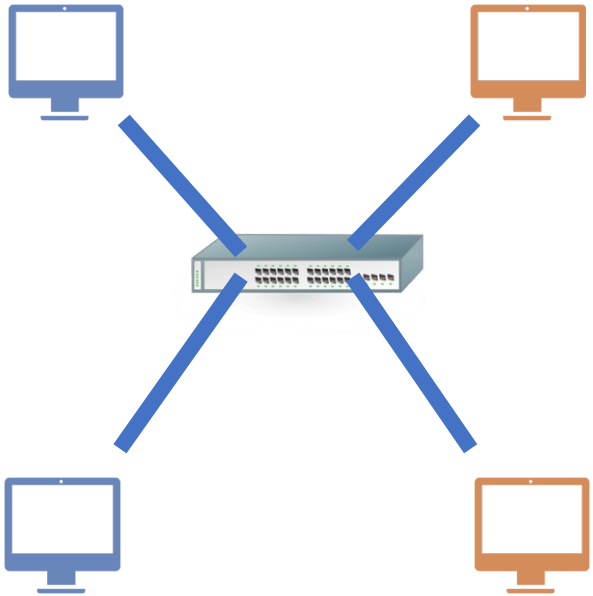
centralelille
IG2I

No VLANs: same L3 network



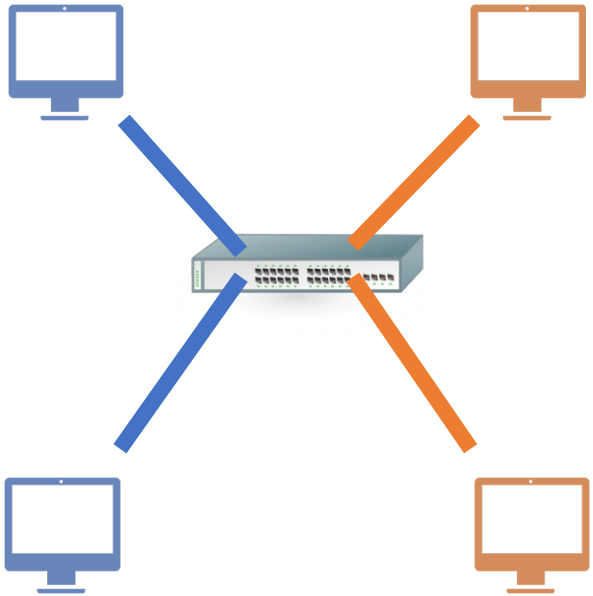
- All machines are
  - In the same L2 network
  - In the same L3 network
- Consequences:
  - They can all communicate
  - They are in the same (L2) broadcast domain

No VLANs: different L3 networks



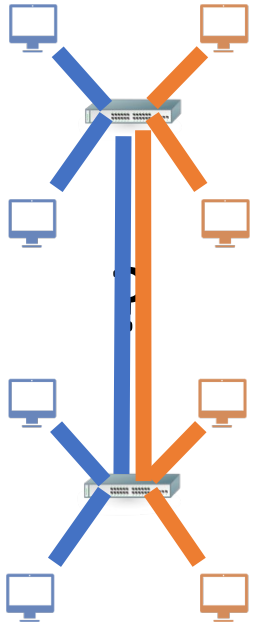
- Machines are
  - In the same L2 network
  - In different L3 networks
- Consequences:
  - They are separated at the L3 level
  - They are in the same (L2) broadcast domain
  - Isolation is only based on L3; no real security

Per-port VLANs



- Machines are
  - In different VLANs (i.e. L2 networks)
  - L3 configuration doesn't matter (unless communication between networks is required)
- Consequences:
  - There are different (L2) broadcast domain
  - Isolation is now based on L2

Per-port VLANs limitations

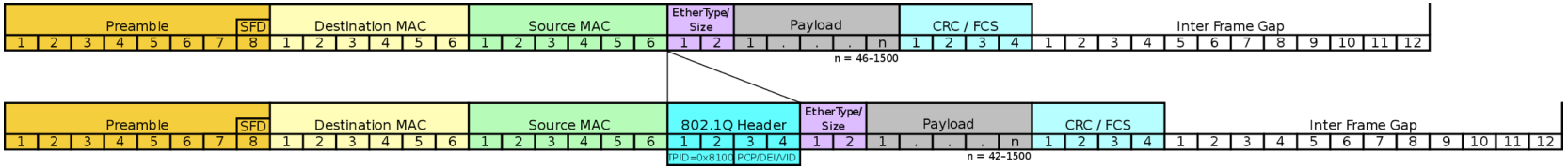


- How to connect different switches preserving VLAN configuration ?

- Answer:
  - Use dedicated ports on each switch

- Problem:
  - When the number of VLANs or the number of switches increases, more ports are "wasted" for the interconnection

802.1Q

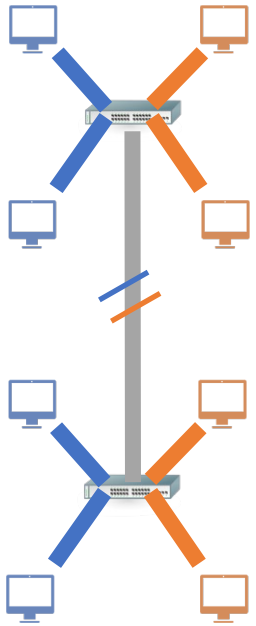


16 bits	3 bits	1 bit	12 bits
TPID 0x8100	TCI		
	Priority Code Point	Drop eligible Indicator	**VLAN-ID 0-4095***

*: 0x000 and 0xFFF are reserved

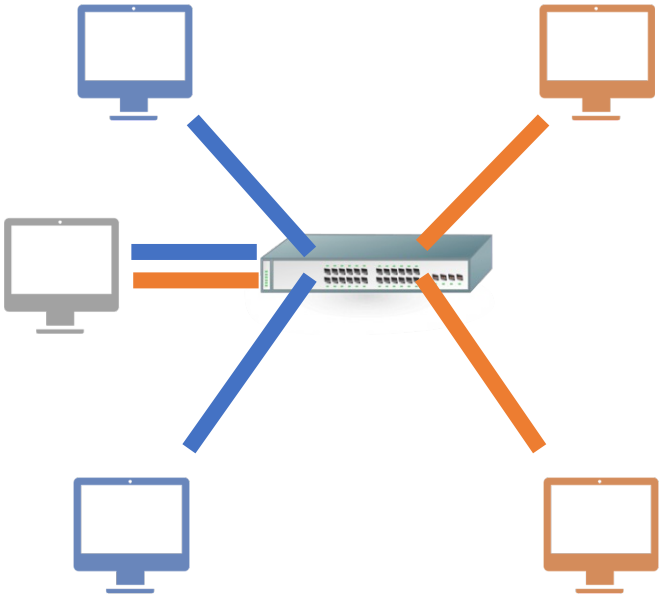
centralelille IG2I

Solution through frame tagging



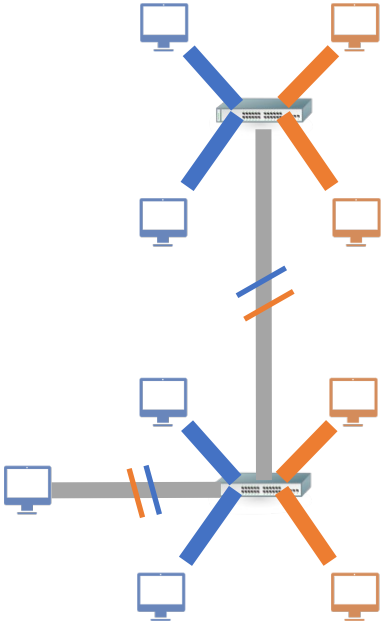
- Ethernet frames are modified to handle "tags" or VLAN-Ids
- Only one wire is required
- Switches have to understand tagging
- Bandwidth bottlenecks can appear

VLAN interconnection without tagging



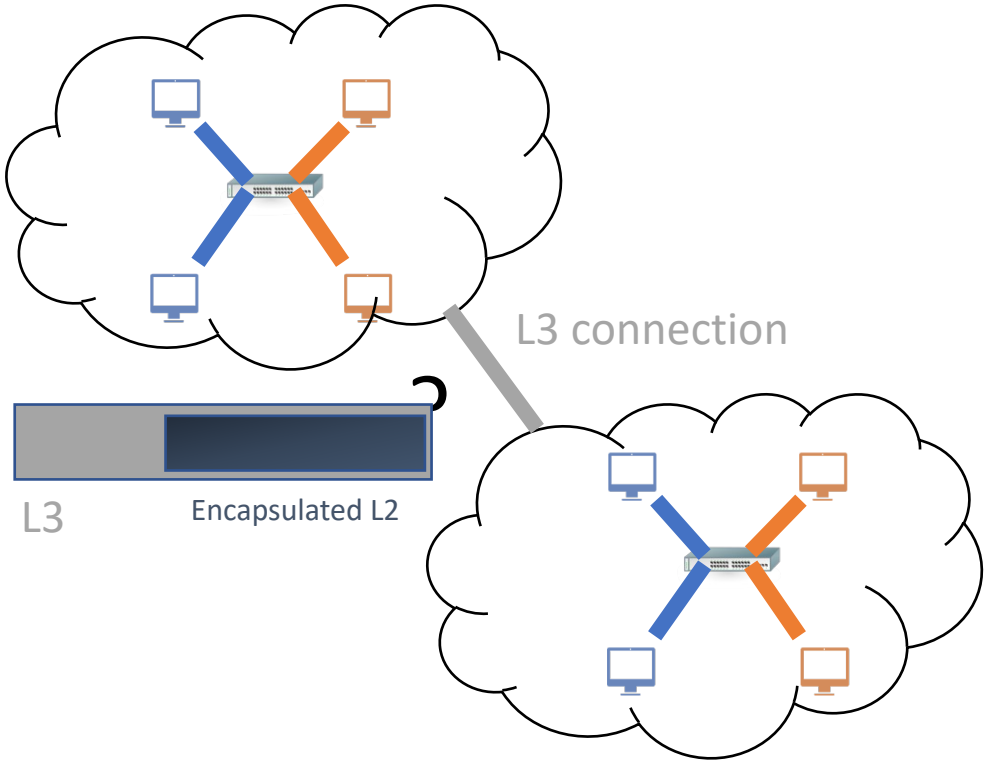
- **How to interconnect two VLANs ?**
- L2 levels are isolated
  - Interconnection at a higher level through routing
  - IP addressing plans must be different
- The same architecture can be used for services installed on both networks
- The router or server and the switch need as many physical cards or ports as there are VLANs

VLAN interconnection with tagging



- **How to interconnect two VLANs ?**
- L2 levels are isolated
  - Interconnection at a higher level through routing
  - IP addressing plans must be different
- The same architecture can be used for services installed on both networks
- The router or server has a single network card, and a single port is used on the switch
- The router or server needs to understand tagging

What about distant L2 networks ?

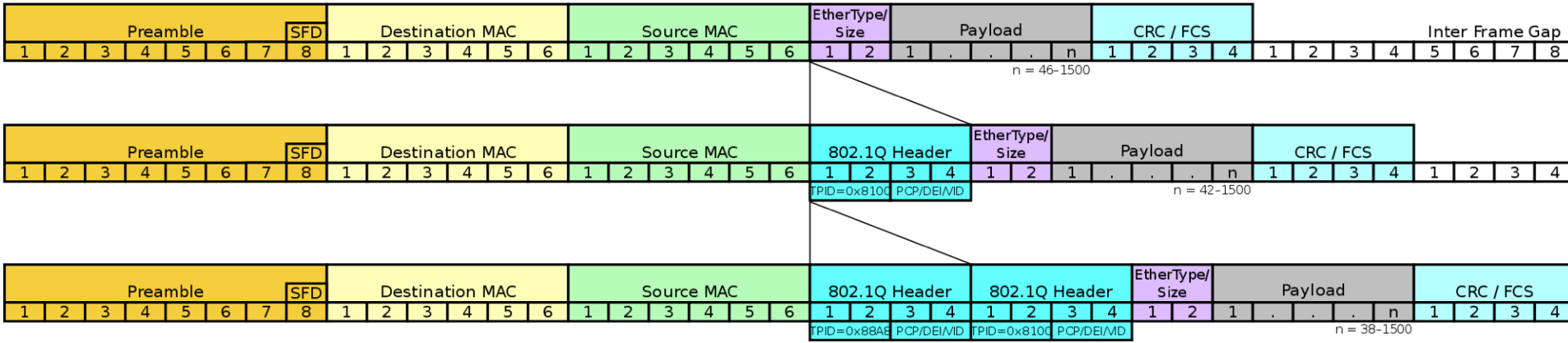


- Interconnection usually exists at the L3 level
- Answer:
  - Encapsulate the original L2 frame into an L3 packet
- Problem:
  - Interconnection points need to understand the encapsulation technique

Other “lower level” approaches

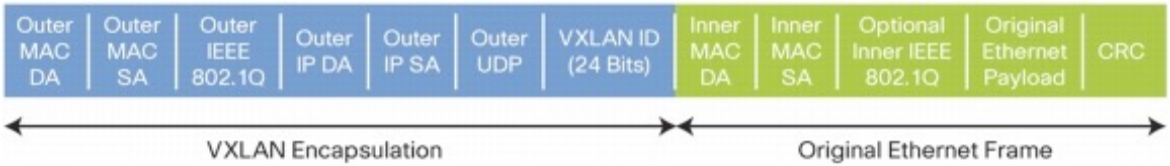
- Problem:
  - Specific VLAN assignment: (theoretically) limited to 2^12 VLANs
- Standard solutions:
  - Q-in-Q
  - VXLAN

L2 Solution: 802.1ad (Q-in-Q)



centralelille IG2I

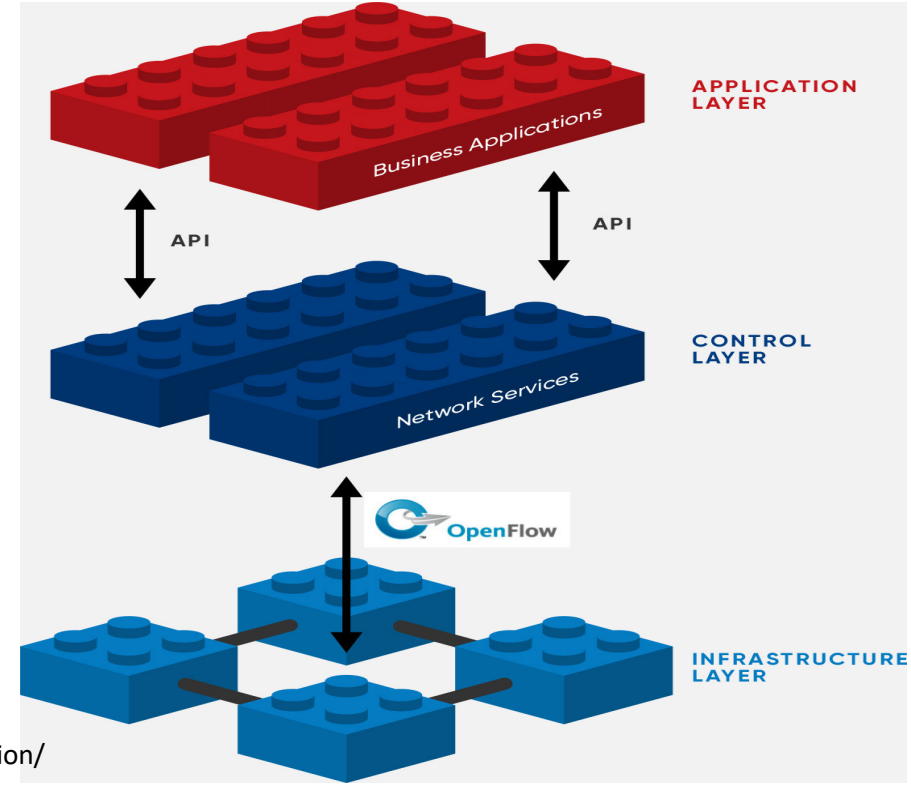
L3 solution: VXLAN



The SDN approach

- Goals:
  - Create programmable network architectures
    - Allows automation,
    - Speeds up deployment
  - Provides agility at the infrastructure level
  - Based on flow routing
  - Allows QoS rules

The SDN approach



Source: https://opennetworking.org/sdn-definition/

centralelille IG2I

The SDN approach

- API enabled
- Provides Firewalling / Security groups
  - Applies rules down to the host from outside the host
- Security policies can be based on tags
- Some SDN solutions & implementations:
  - OpenFlow,
  - Openstack (Neutron),
  - VMWare NSX.

centralelille IG2I

Virtualization in a nutshell

Storage virtualization

centralelille
IG2I

Storage Virtualization

In computer science, **storage virtualization** is the process of presenting a logical view of the physical storage resources to a host computer system

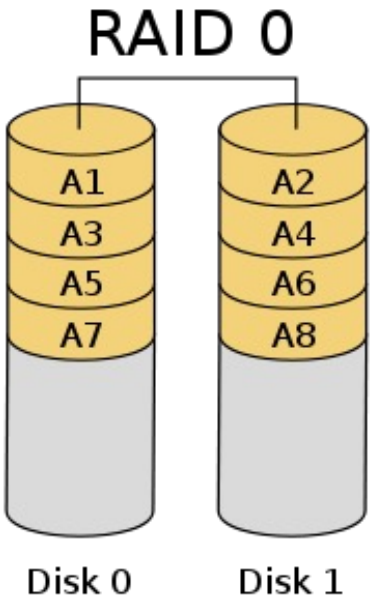
Source: wikipedia

centralelille IG2I

Understanding Storage Technologies

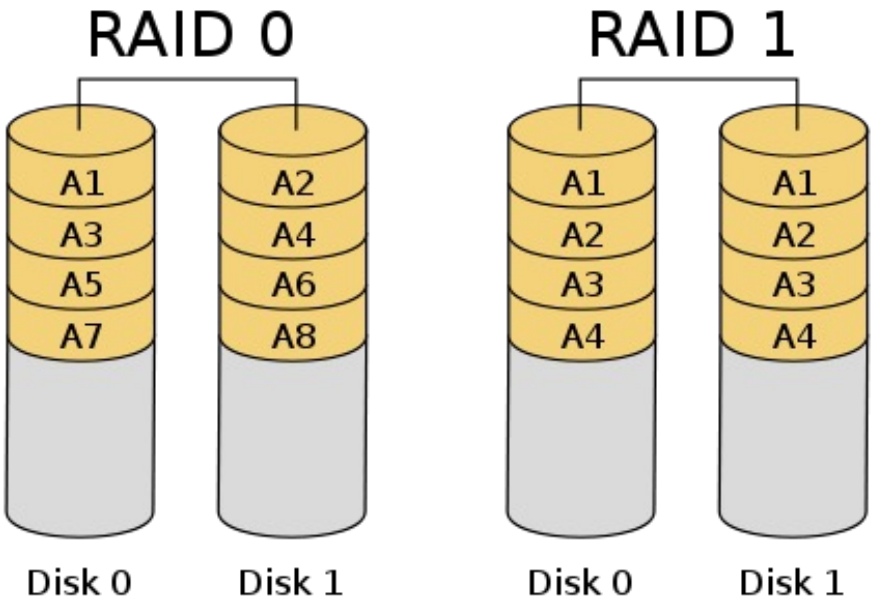
- At the physical level,
  - Example: RAID,
- Block Storage,
  - Examples: LVM, SAN
- File Storage,
  - Examples: NAS
- Object Storage.
  - Example: Ceph

RAID



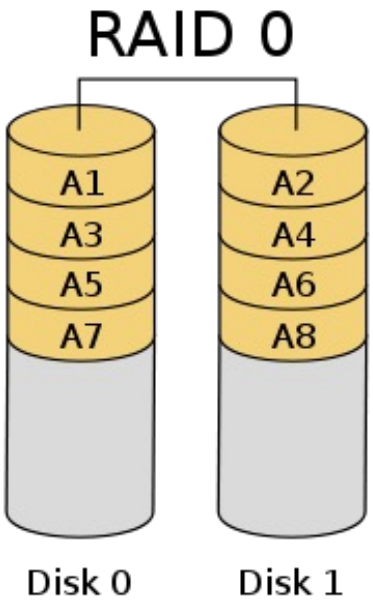
centralelille
IG2I

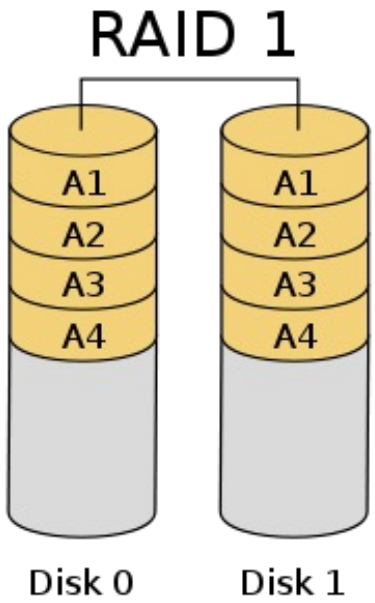
RAID

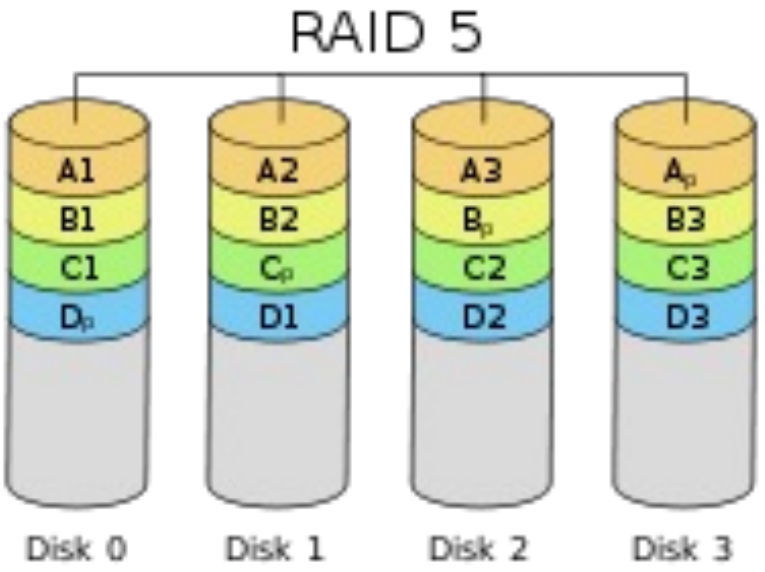


centralelille
IG2I

RAID

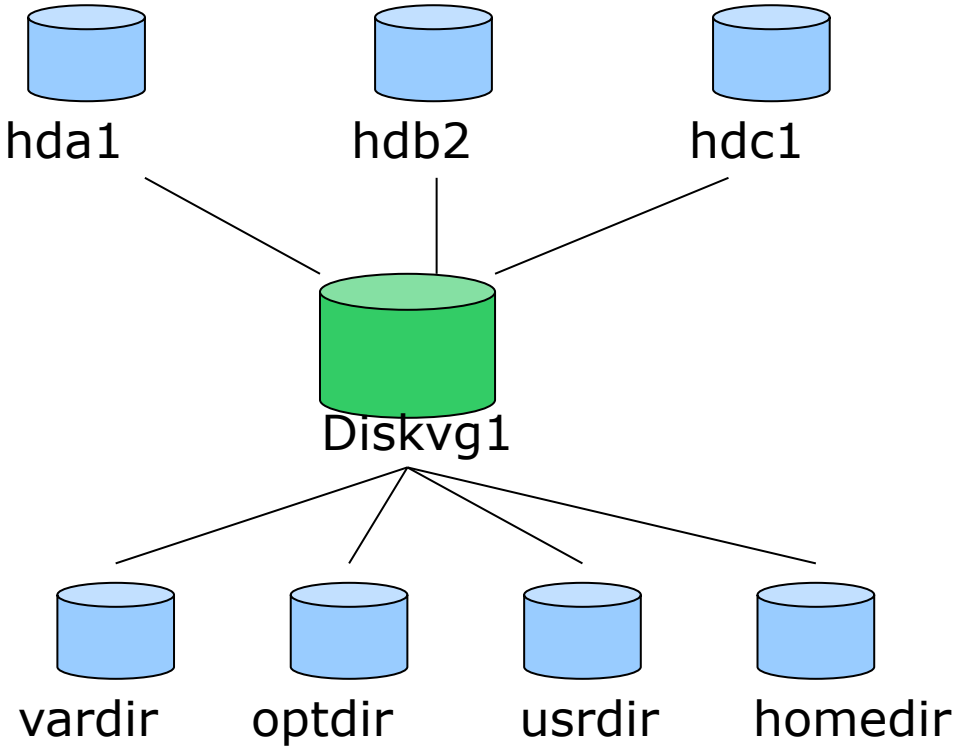






centralelille
IG2I

LVM



Physical Volumes

Volume Groups

Logical Volumes

centralelille IG2I

Network Attached Storage

- File sharing solutions based on IP
- Based on FILES
  - Client systems access files,
  - Files security is enforced at the server level,
  - The real file system is hidden to the client
- Standard solutions:
  - CIFS (or SMB)
  - NFS
  - AFP

centralelille IG2I

Storage Area Network

- Based on BLOCKS
  - The client system “attaches” a remote disk,
  - The client system must install a filesystem,
  - The server controls the access to the shared device,
  - The server doesn’t use the filesystem installed by the client.
- Standard solutions:
  - Dedicated Fiber Channel network,
  - iSCSI on IP networks.

centralelille IG2I

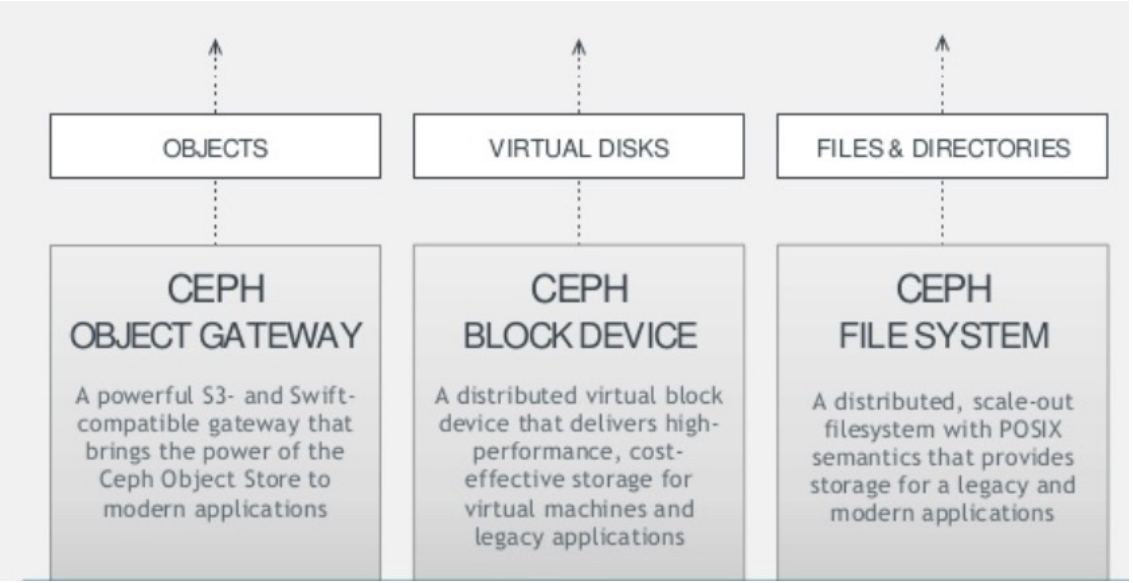
Object Storage

- Basics:
  - Objects are saved “as is” in a flat file system
  - Metadata are associated to objects identified by IDs
- Pros:
  - Scalability,
  - Customizable metadata,
  - High (sequential) throughput performance.
- Cons:
  - Full object manipulation,
  - Can’t easily be used as standard filesystems
- Standard solutions:
  - Ceph Object Storage, Swift, Amazon S3

centralelille IG2I

A focus on Ceph

- Ceph is an opensource, unified, distributed storage system designed for performance, reliability and scalability.



centralelille IG2I

Back to Cloud Computing…

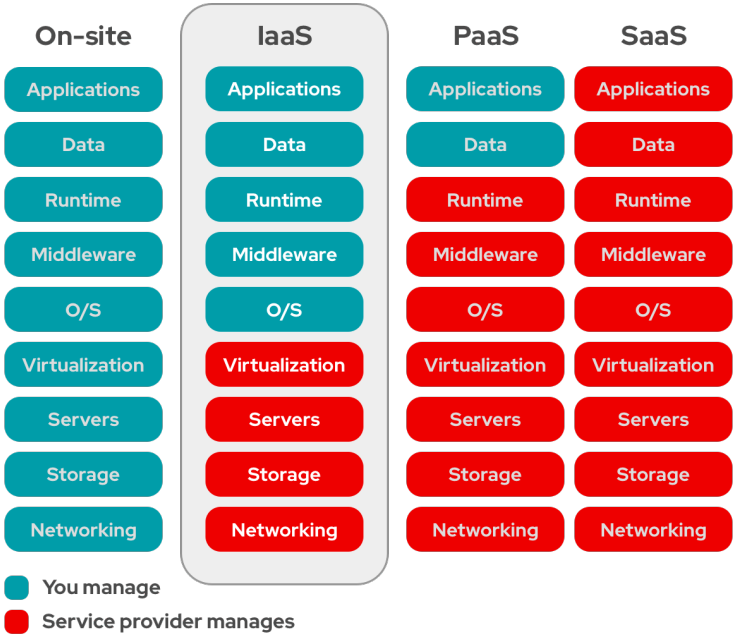
centralelille
IG2I

Cloud Services

- Standard Cloud Services Models – SPI Stack
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)

centralelille
IG2I

IaaS



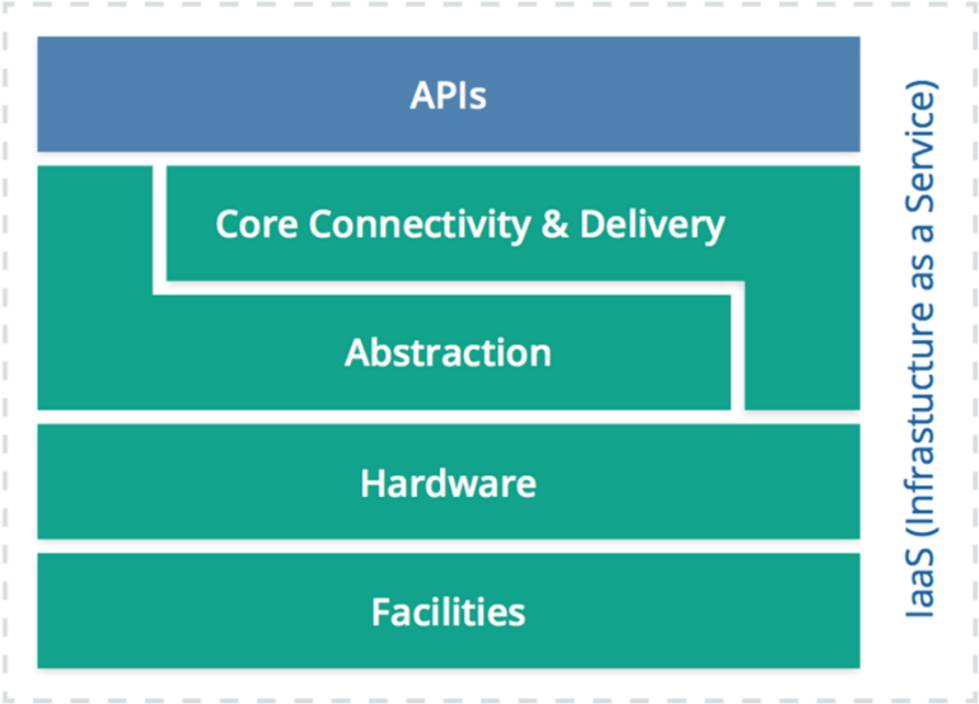
Source: https://www.redhat.com/en/topics/cloud-computing/what-is-iaas

centralelille IG2I

IaaS

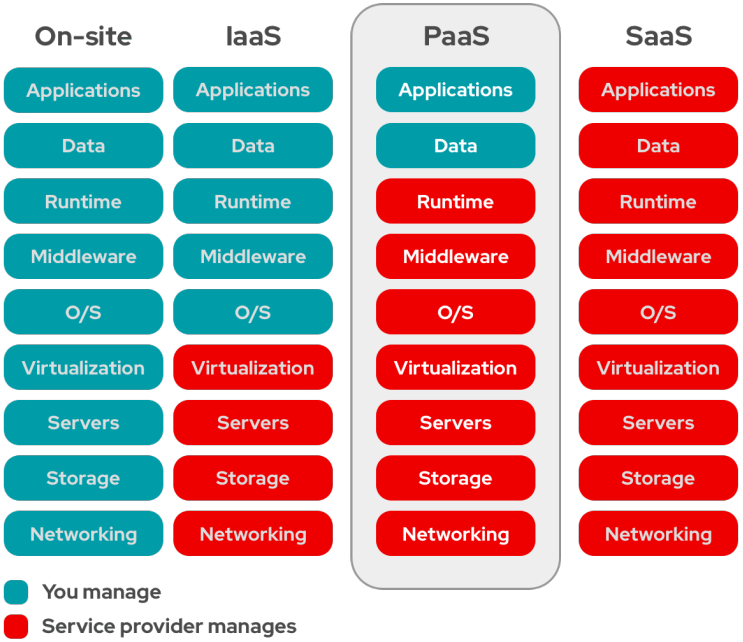
Examples:

- AWS EC2
- Google Compute Engine
- OVHCloud Public Cloud



Source: CSA - Cloud Security Course 1.0 Netacad

PaaS



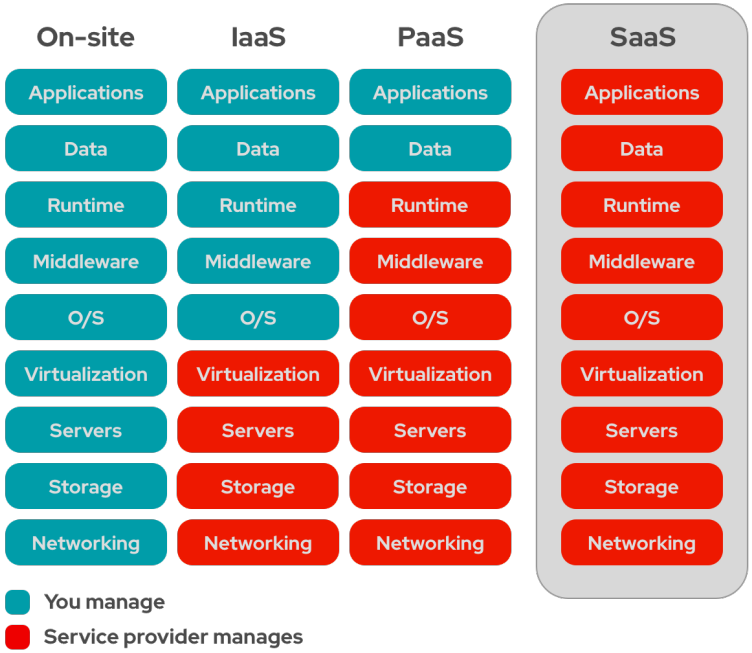
Source: https://www.redhat.com/en/topics/cloud-computing/what-is-paas

centralelille IG2I

PaaS

- Examples:
  - AWS Beanstalk
  - AWS Lambda
  - Google App Engine
  - Microsoft Azure
  - OVHcloud Web PaaS
  - IBM Cloud Foundry

SaaS



Source: https://www.redhat.com/en/topics/cloud-computing/what-is-saas

centralelille IG2I

SaaS

- Examples:
  - Google G Suite
  - Microsoft Office 365
  - Dropbox
  - Slack

Cloud Deployment Models

- The NIST model defines 4 types of Cloud models
  - Public Cloud,
  - Private Cloud,
  - Hybrid Cloud,
  - Community Cloud.

centralelille
IG2I

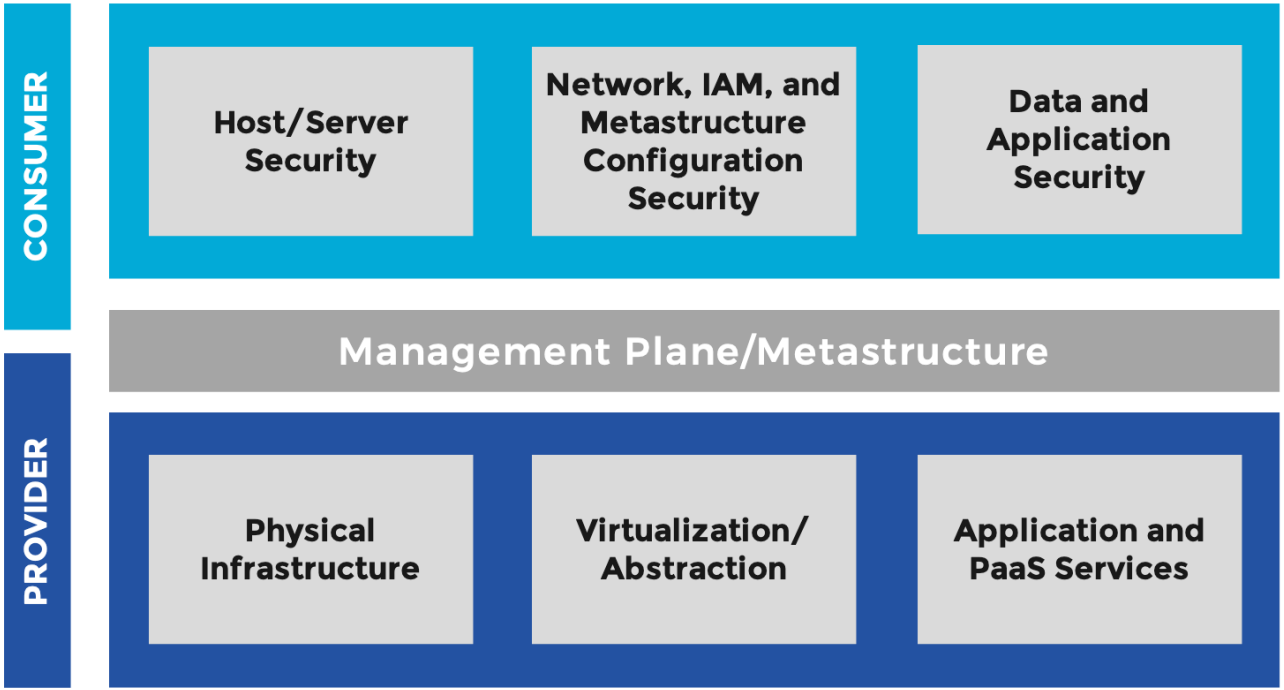
Cloud Deployment Models

	Infrastructure Managed By[1]	Infrastructure Owned By[2]	Infrastructure Located[3]	Accessible and Consumed By[4]
Public	Third Party Provider	Third Party Provider	Off-Premise	Untrusted
Private/ Community	Organization / Third Party Provider	Organization / Third Party Provider	On-Premise / Off-Premise	Trusted
Hybrid	Both Organization & Third Party Provider	Both Organization & Third Party Provider	Both On-Premise & Off-Premise	Trusted & Untrusted

Source: CSA - Cloud Security Course 1.0 Netacad

centralelille IG2I

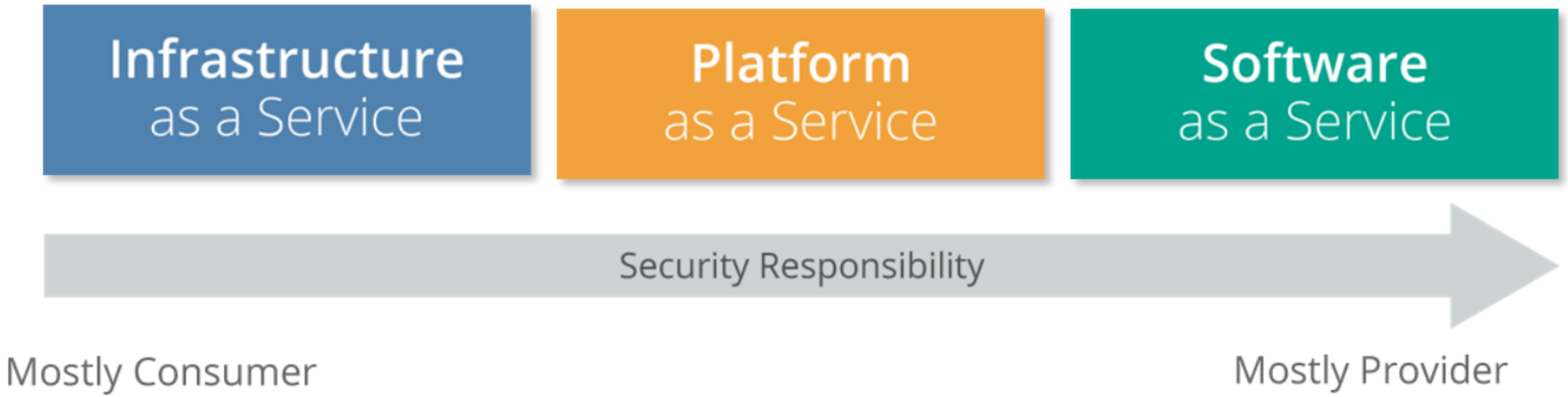
Shared responsibility model



Source: CSA - Cloud Security Course 1.0 Netacad

centralelille IG2I

Responsibility vs. Models



Source: CSA - Cloud Security Course 1.0 Netacad

centralelille IG2I

Cloud Offers

- Case studies:
  - The OVHCloud offer
  - The AWS offer
- Global things to consider:
  - Pricing
  - Availability zones
  - Security features

centralelille IG2I