EPITECH.

PARIS GRADUATE SCHOOL OF DIGITAL INNOVATION

Kubi

Kick-off

T10 - Cloud

T-CLO-902

BARCELONE - BERLIN - BORDEAUX - BRUXELLES - LA REUNION - LILLE - LYON - MARSEILE - MONTPELLIER - NANCY - NANTES - NICE - RENNES - STRASBOURG - TIRANA - TOULOUSE

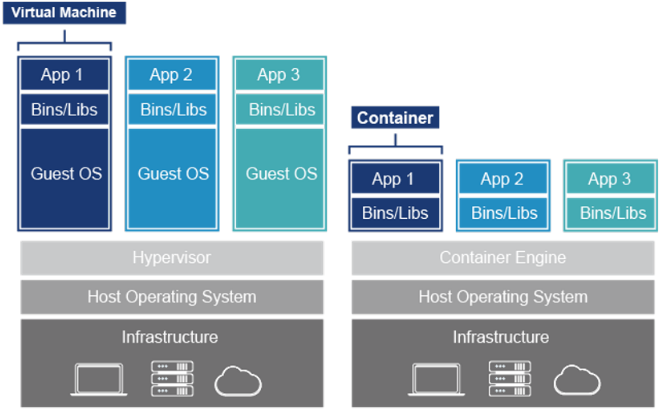
1.2

Rise of containers and Docker

Architecture

- Abstraction layer between OS and application
- You package runtime (ex: NodeJS, libs…) + application into a single container
- Basically a tar.gz

Rise of containers and Docker



Rise of containers and Docker

Use case

- 2 NodeJS app (first in NodeJS v0.12 and second in NodeJS v14)
- Start a Redis instance in 10s without any install
- Ensure you run the same thing in staging and production
- Multiple containers per VM == cheaper
- Build once, run everywhere

Rise of containers and Docker

Isolation at kernel level

- 1 process isolated from neighborhood
- Kernel feature: *namespace* and *cgroups*
- Process don’t see each other
- Execute: “ps aux” inside and outside a container

Rise of containers and Docker

Shared kernel

- Some bottlenecks
- Shared resources
- Security (!)

Rise of containers and Docker

Microservices and Docker containers both grew at the same time.

There is no microservices if infrastructure cannot support it.

Rise of containers and Docker

And you know what?

Docker has been built by **Epitech** alumnis.

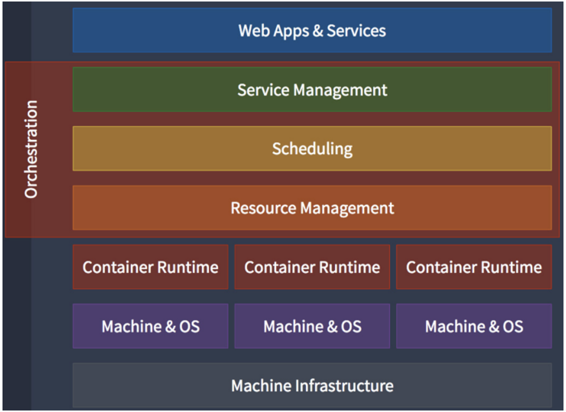
It means a lot ;)

Container orchestrator

An abstraction between

a cluster of **virtual machines** and **containers**

Container orchestrator



Web Apps & Services

Orchestration

Service Management

Scheduling

Resource Management

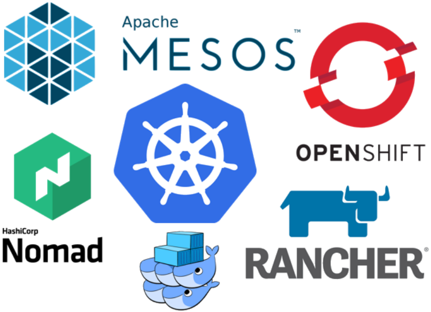
Container Runtime | Container Runtime | Container Runtime

Machine & OS | Machine & OS | Machine & OS

Machine Infrastructure

Container orchestrator

Run in container in Orchestrator, scheduled somewhere



Kubernetes

Examples of scheduling constraints:

- Start X instances of a service
- Spread in multiple bare-metal or datacenters
- Or ensure service A is on the same server as service B
- Resource management (reserve CPU, memory, disk…)

Kubernetes

Examples of scheduling constraints:

- Auto-scaling of services
- Auto-scaling of nodes
- Bin packing
- Node drain

Kubernetes

Examples of lifecycle constraints:

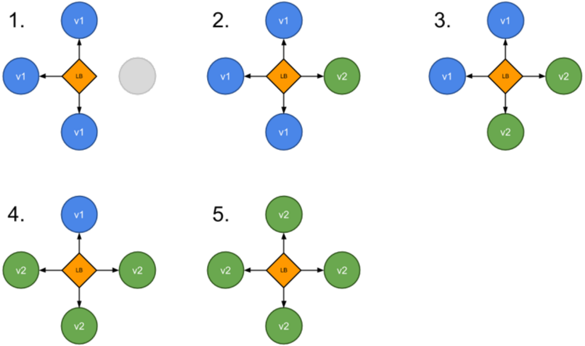
- Start service A after service B’s job is done (pre-start)
- Start twice a day, every 12 hours
- Restart on failure
- Health-check

Kubernetes

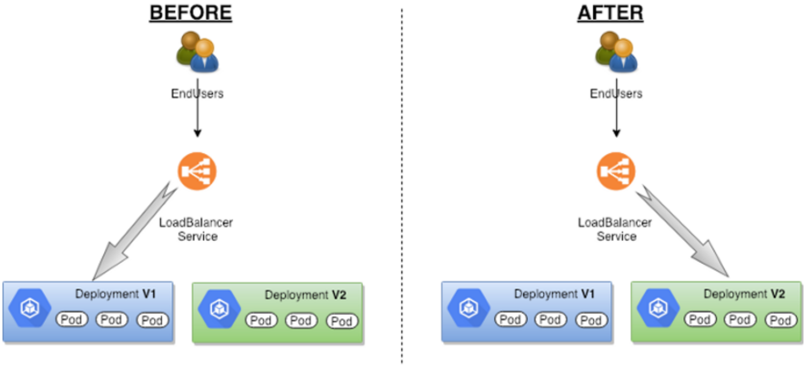
Examples of deployment constraints:

- Zero-downtime deployment
- Auto-healing
- 1-click rollback
- Blue-green, canary or rolling-update

Kubernetes


1.
v1
v1
LB
v1
2.
v1
v1
LB
v2
v1
3.
v1
v1
LB
v2
v2
4.
v1
v2
LB
v2
v2
5.
v2
v2
LB
v2
v2

Kubernetes



BEFORE

EndUsers

LoadBalancer Service

Deployment V1

Pod Pod Pod

Deployment V2

Pod Pod Pod

AFTER

EndUsers

LoadBalancer Service

Deployment V1

Pod Pod Pod

Deployment V2

Pod Pod Pod

Kubernetes

ACL

a user or application must have permission
before carrying out an operation in a k8s cluster

Kubernetes

Other cool things:

- Internal DNS
- Volumes
- Secret management
- Multi-cloud - Cross region

Kubernetes

What about the sysadmin job ?

3 levels:

- infrastructure (hardware) ← cloud provider
- platform (orchestrator) ← system administrator
- application (container) ← developer

Kubernetes

Limits:

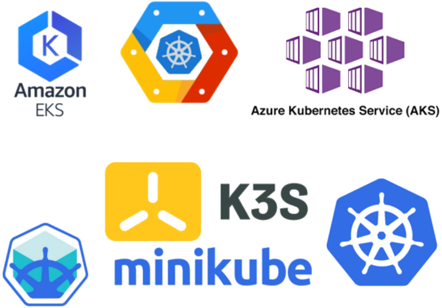
- Stateful services (ex: databases)
- K8s on top of IaaS can be a duplicate
- K8s is a war machine, hard to set up
- A Lot of config files
- Multi-tenant k8s cluster is a security issue

Kubernetes

You should see K8s as an infrastructure framework

Kubernetes

How to run a k8s cluster?



Any questions

?