



We code the future. Together

2020-02-20

WEBINAR



Kubernetes: beyond Minikube

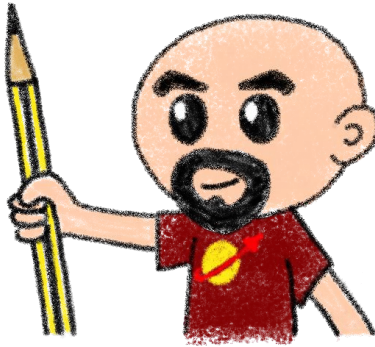


Horacio Gonzalez
@LostInBrittany



Who are we?

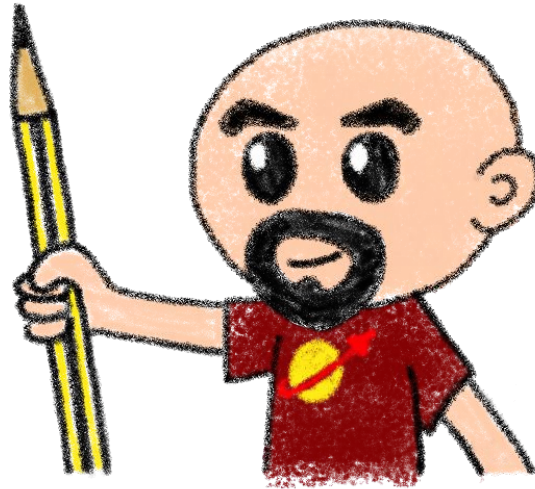
Introducing myself and
introducing ~~OVH~~ OVHcloud



Horacio Gonzalez

@LostInBrittany

Spaniard lost in Brittany,
developer, dreamer and
all-around geek

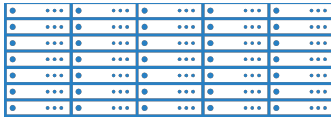


OVHcloud: A Global Leader

250k Private cloud VMs running

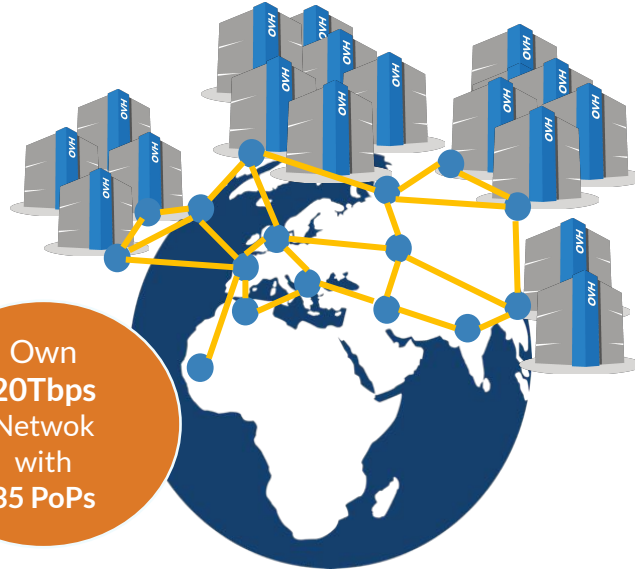


Dedicated IaaS Europe



Hosting capacity :
1.3M Physical Servers

360k
Servers already
deployed







Own
20Tbps
Network
with
35 PoPs

30 Datacenters

> **1.3M** Customers in **138** Countries



OVHcloud: Our solutions

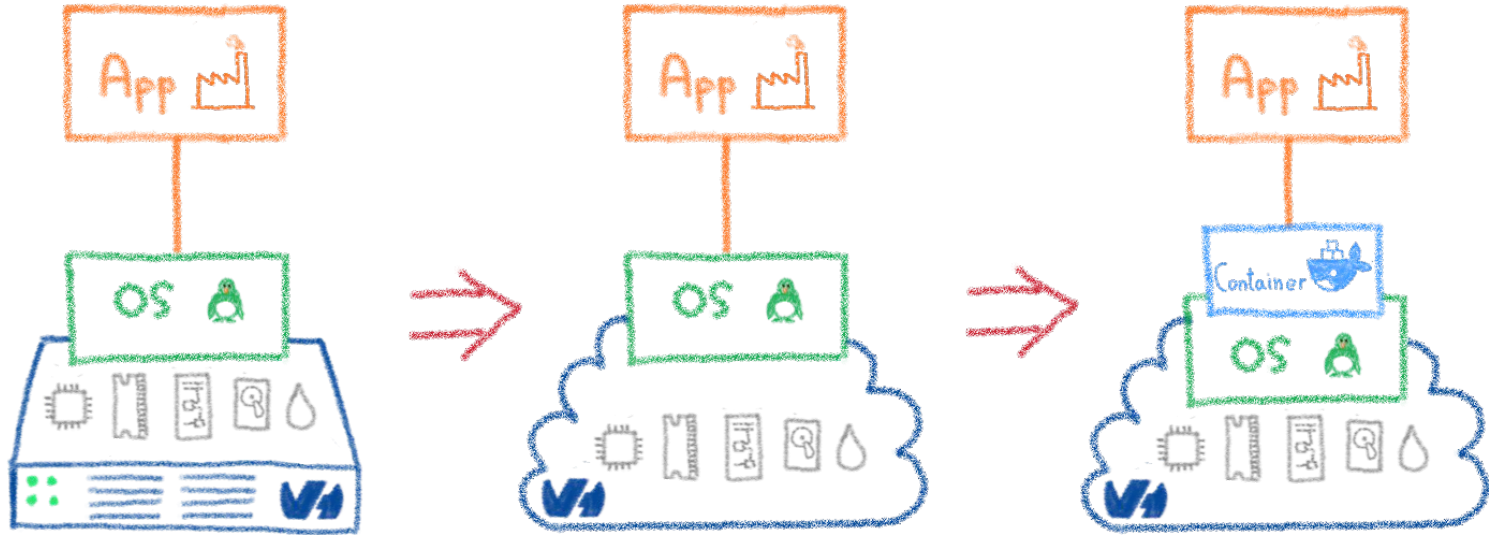
 Cloud <hr/> VPS Public Cloud Private Cloud Serveur dédié Cloud Desktop Hybrid Cloud	 Mobile Hosting <hr/> Containers Compute Database Object Storage Securities Messaging	 Web Hosting <hr/> Domain names Email CDN Web hosting MS Office MS solutions	 Telecom <hr/> VoIP SMS/Fax Virtual desktop Cloud HubIC Over theBox
---	--	---	--

Orchestrating containers

Like herding cats... but in hard mode!

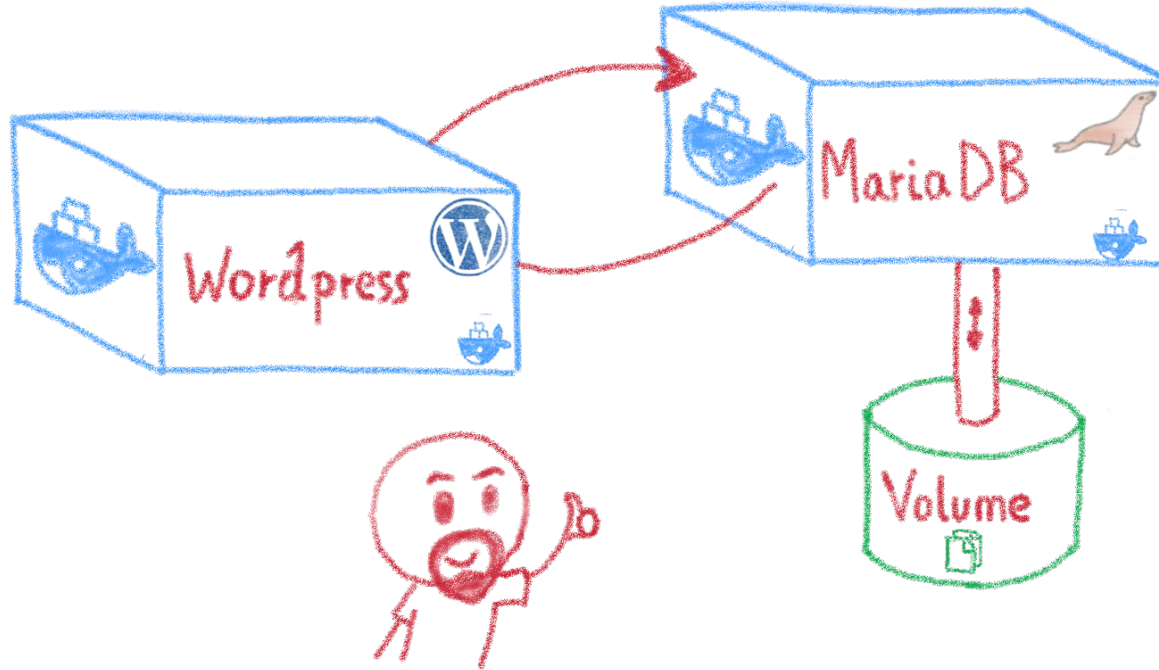


From bare metal to containers



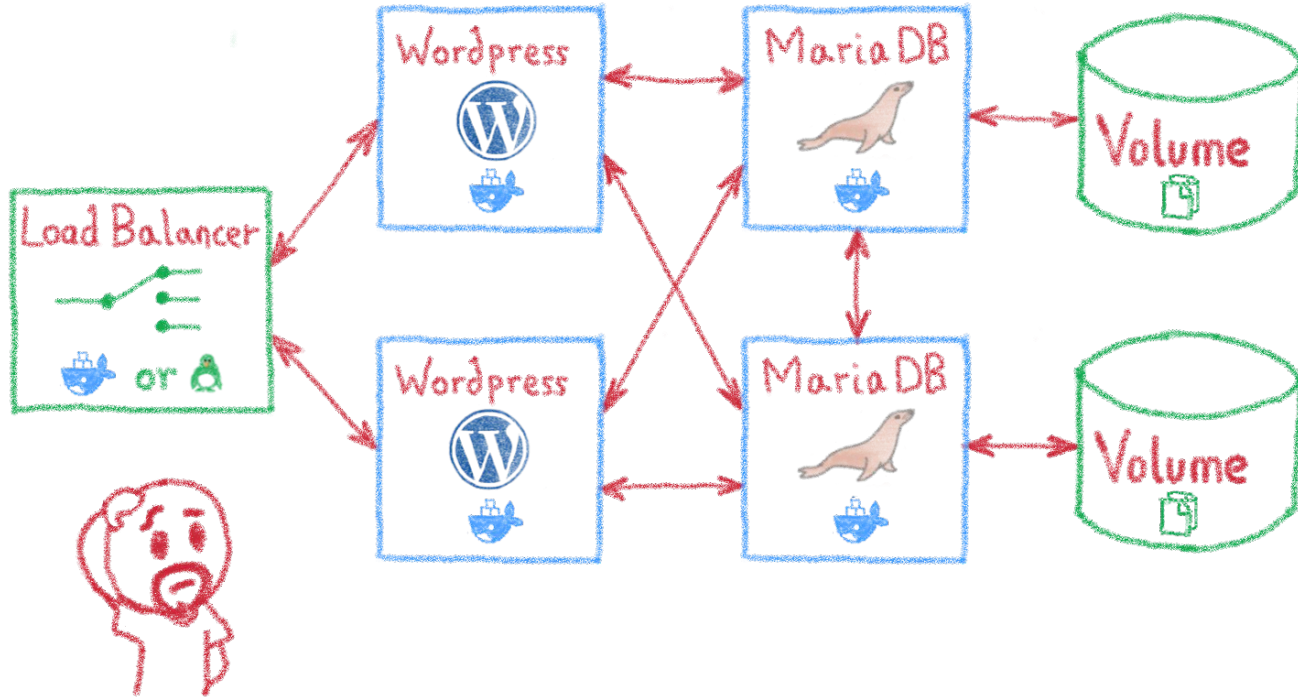
Another paradigm shift

Containers are easy...



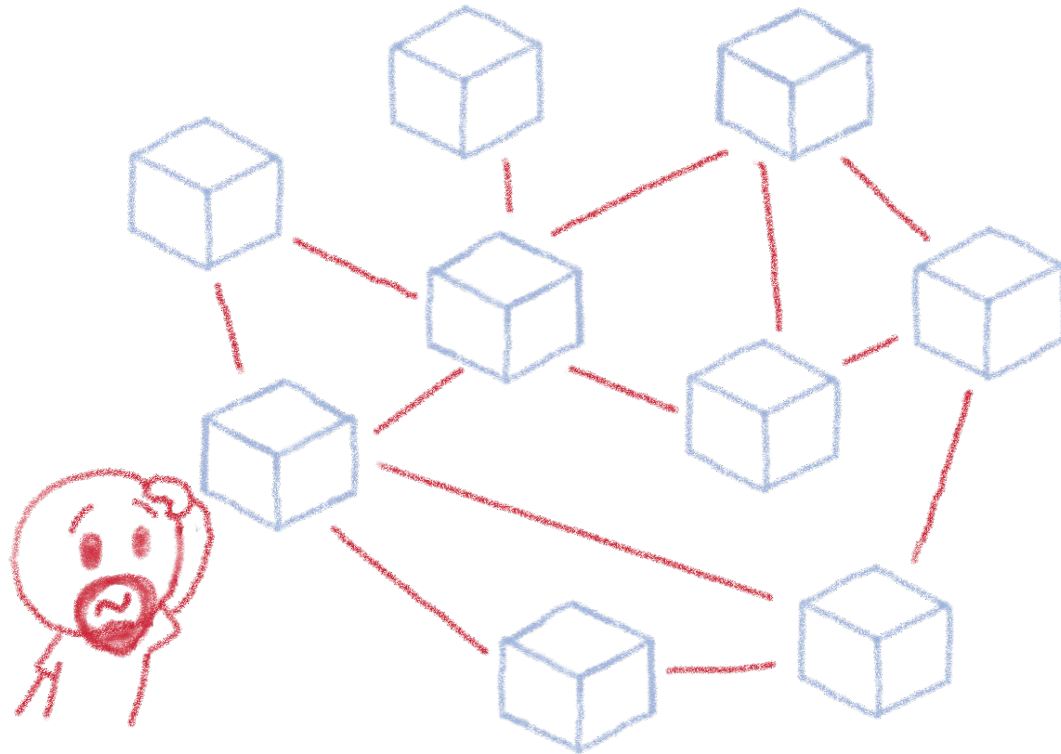
For developers

Less simple if you must operate them



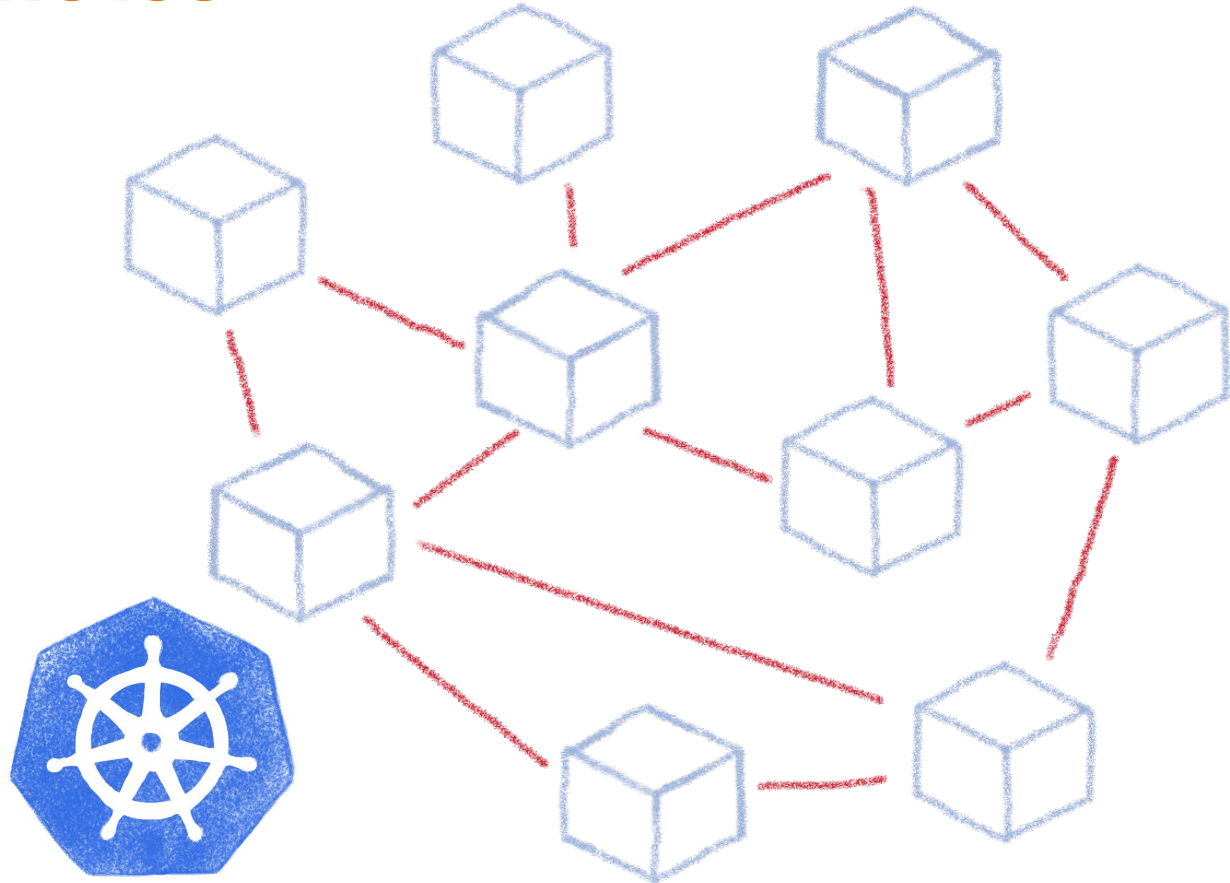
Like in a production context

And what about microservices?



Are you sure you want to operate them by hand?

Taming microservices with Kubernetes

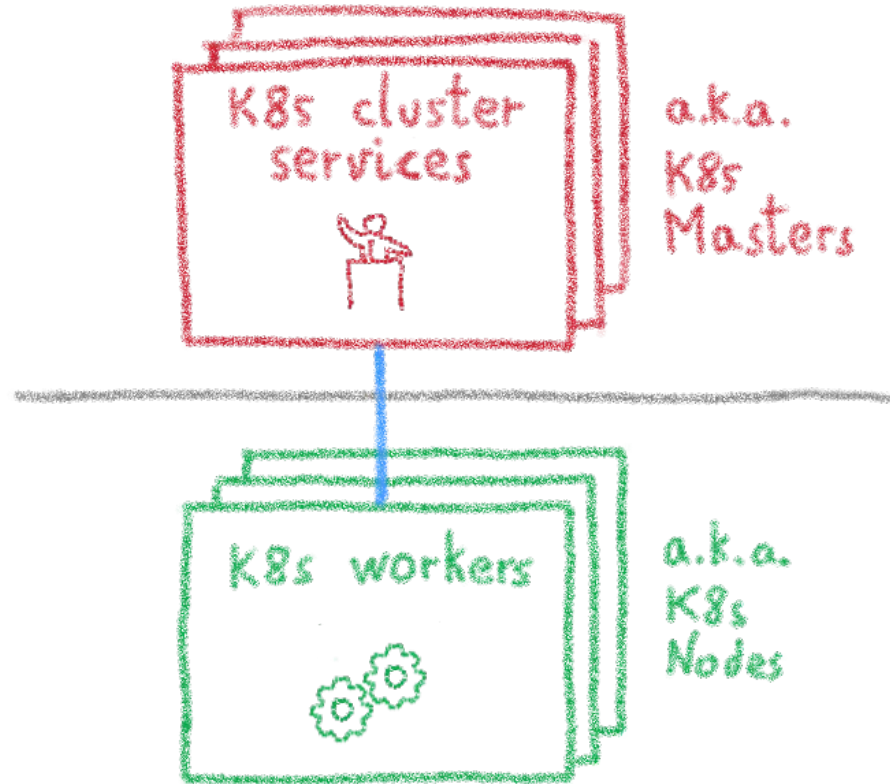


Kubernetes

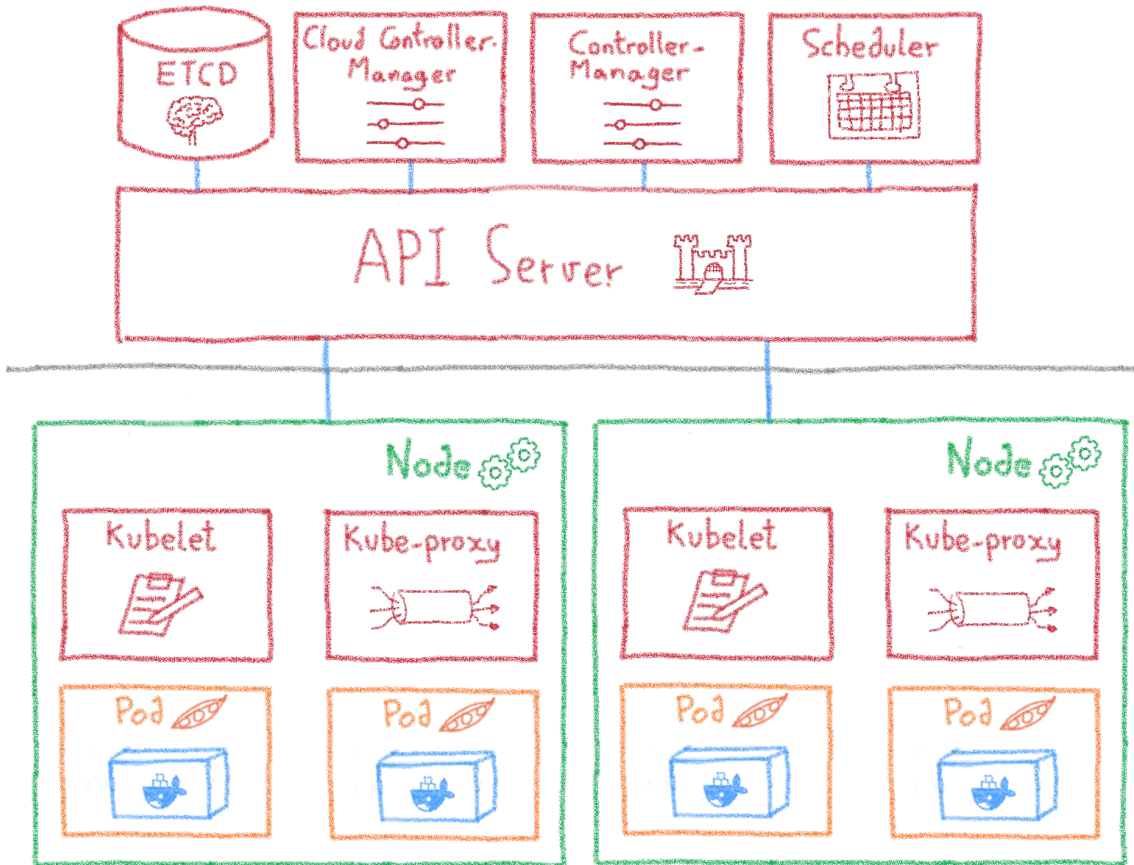
Way more than a buzzword!



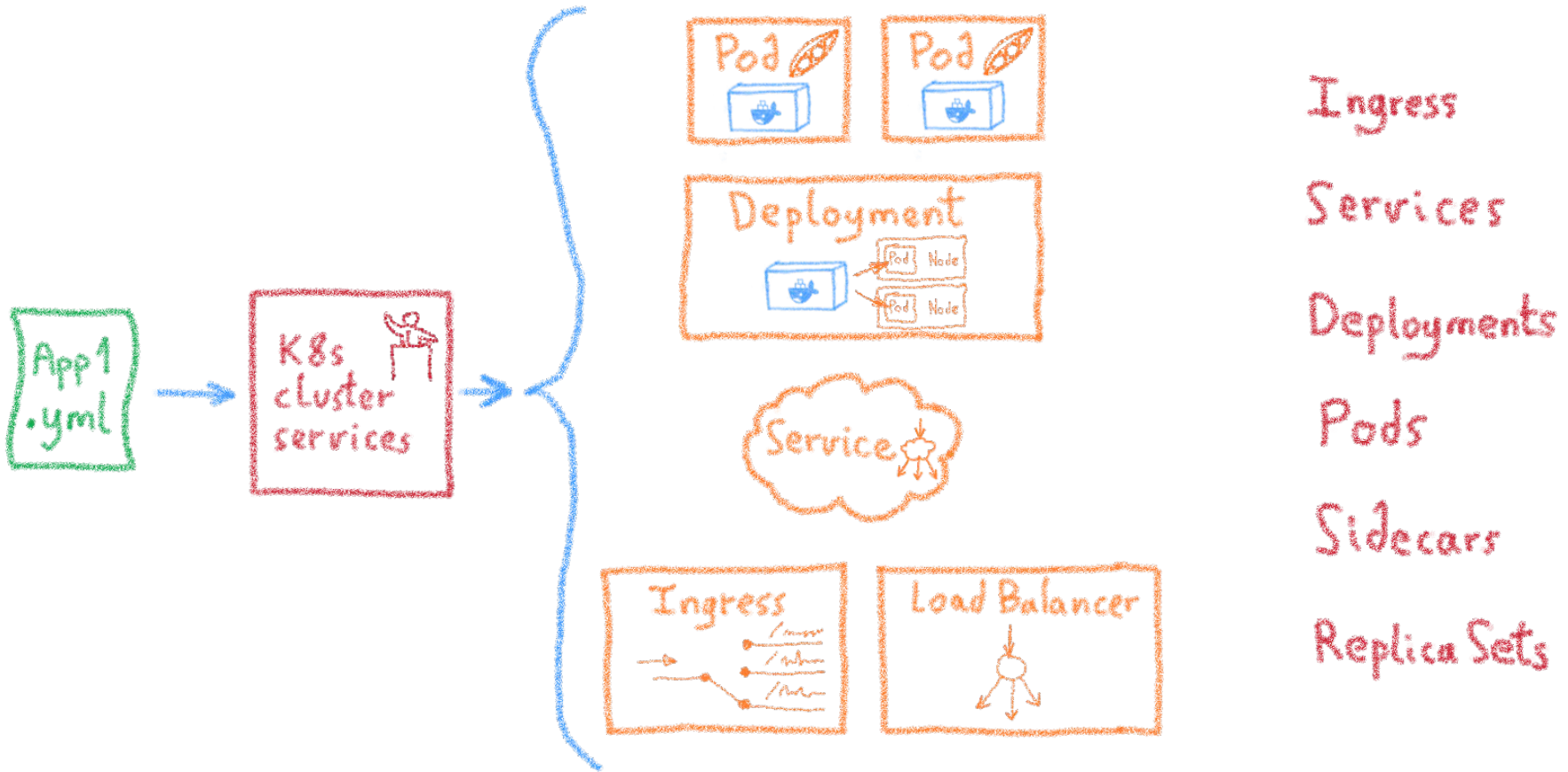
Masters and nodes



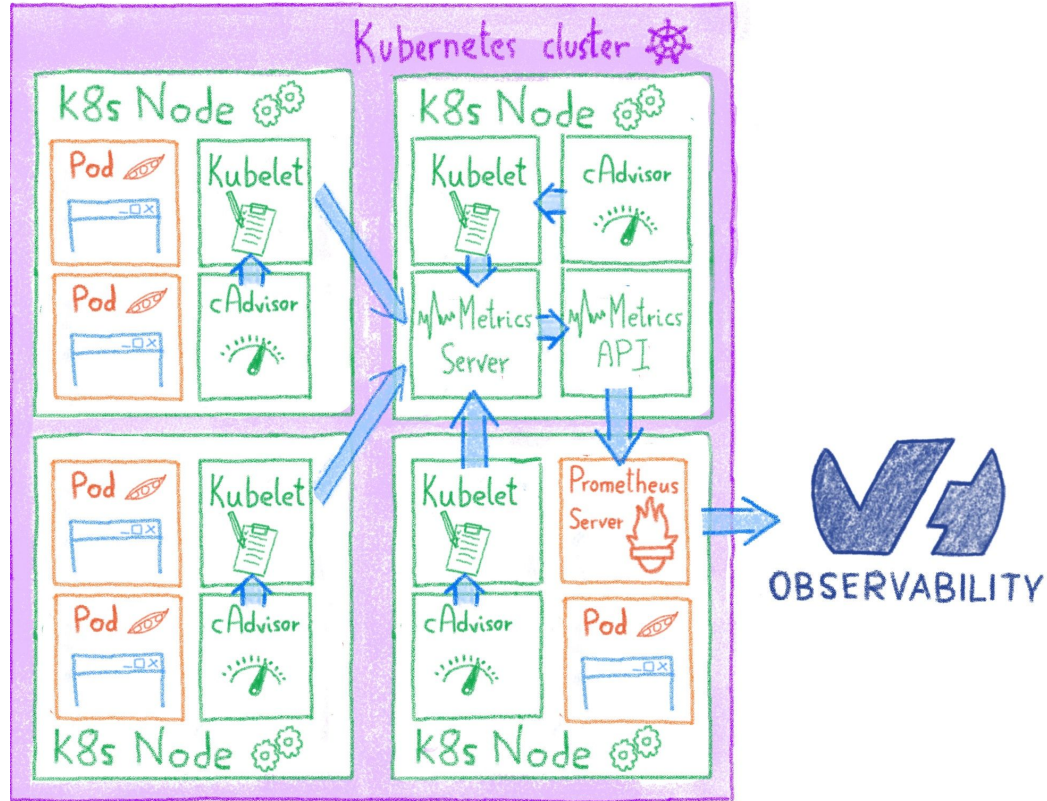
Some more details



Desired State Management

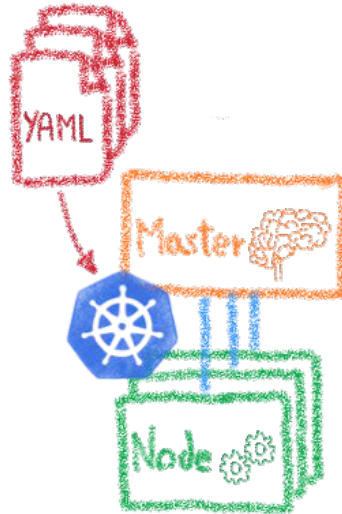


Extending Kubernetes

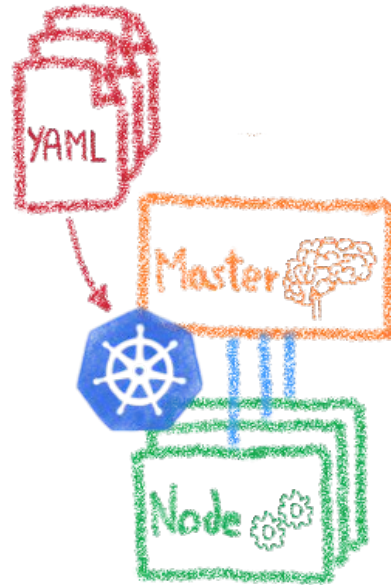


Multi-environment made easy

Dev, staging, prod, multi-cloud...

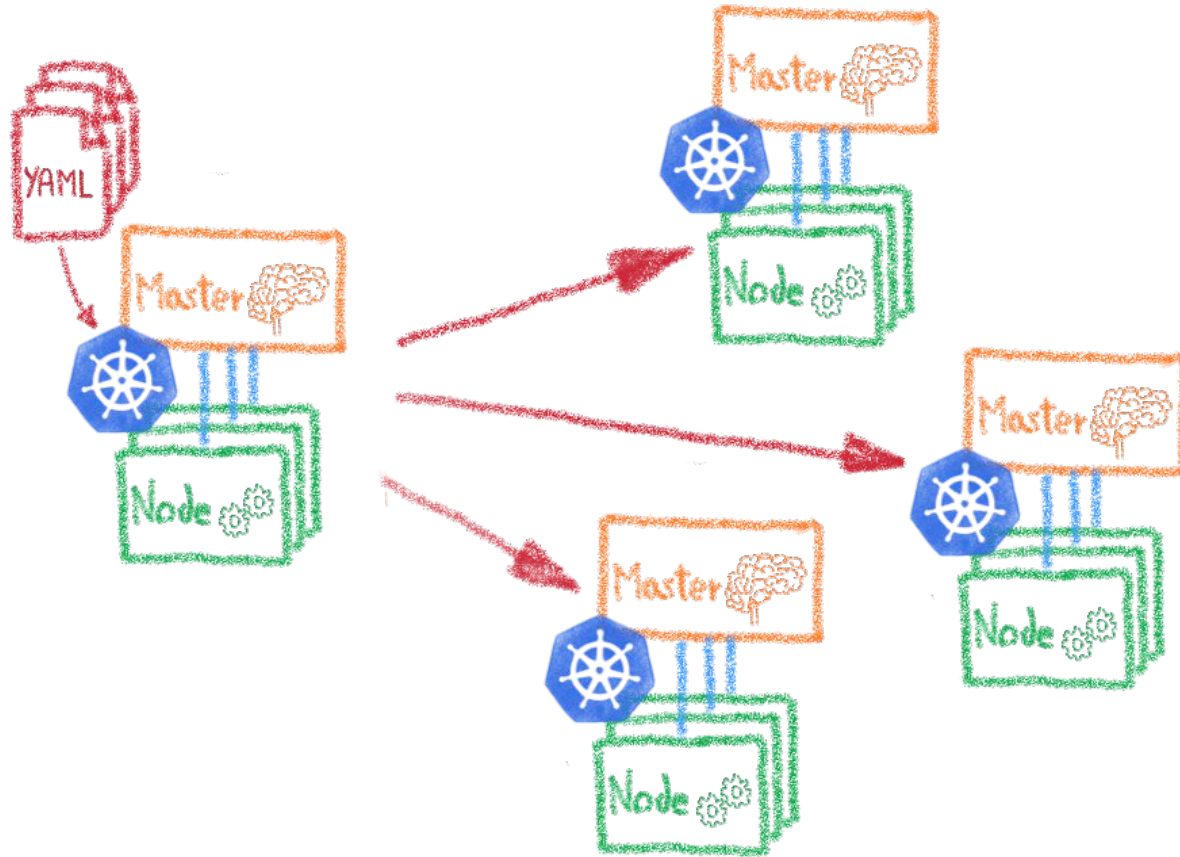


Declarative infrastructure



Multi-environment made easy

Having identical, software defined environments



Dev envs

Staging

Multi-cluster

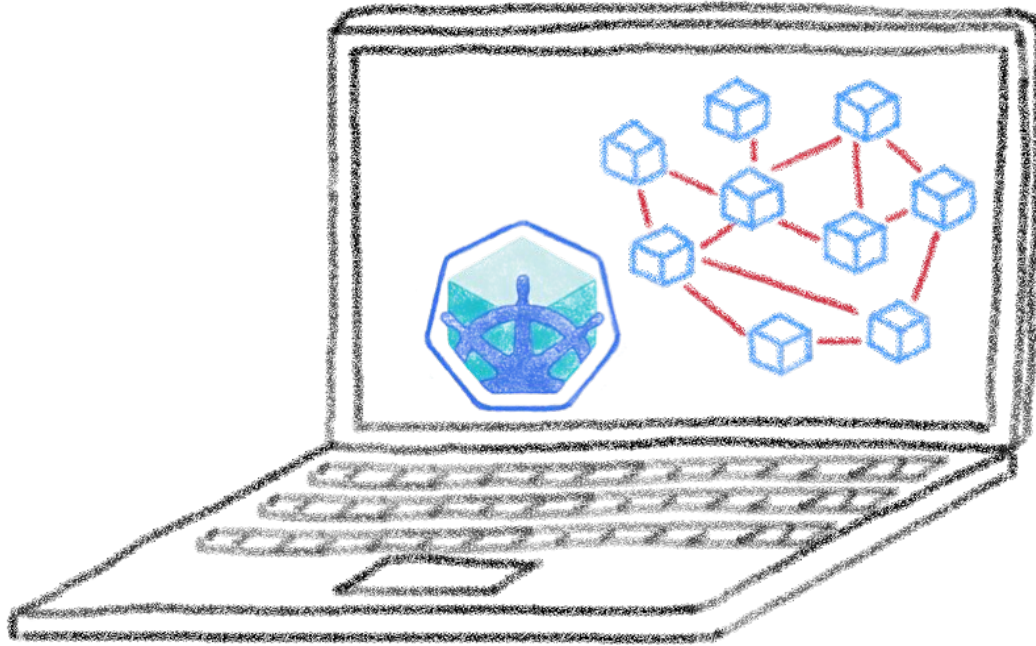
Multi-cloud

I have deployed on Minikube, woah!

A great fastlane into Kubernetes

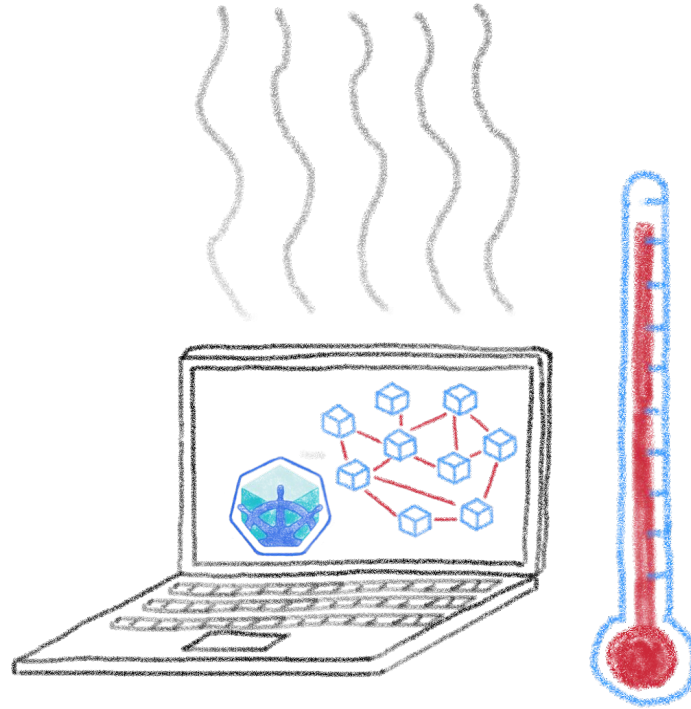


Running a full K8s in your laptop



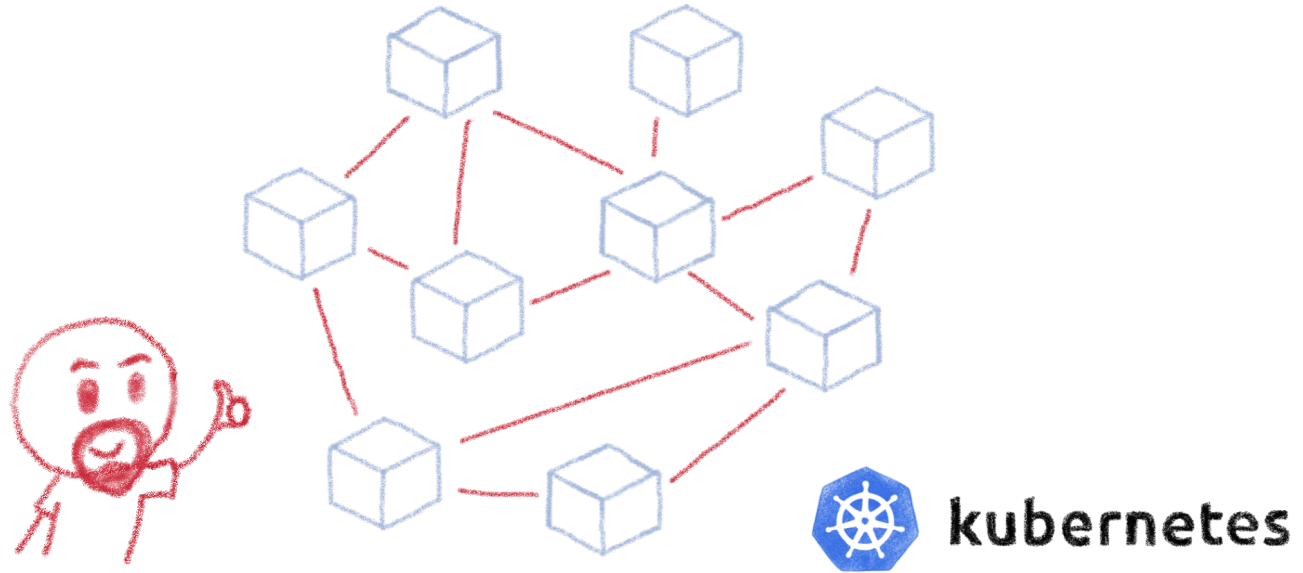
A great learning tool

Your laptop isn't a true cluster



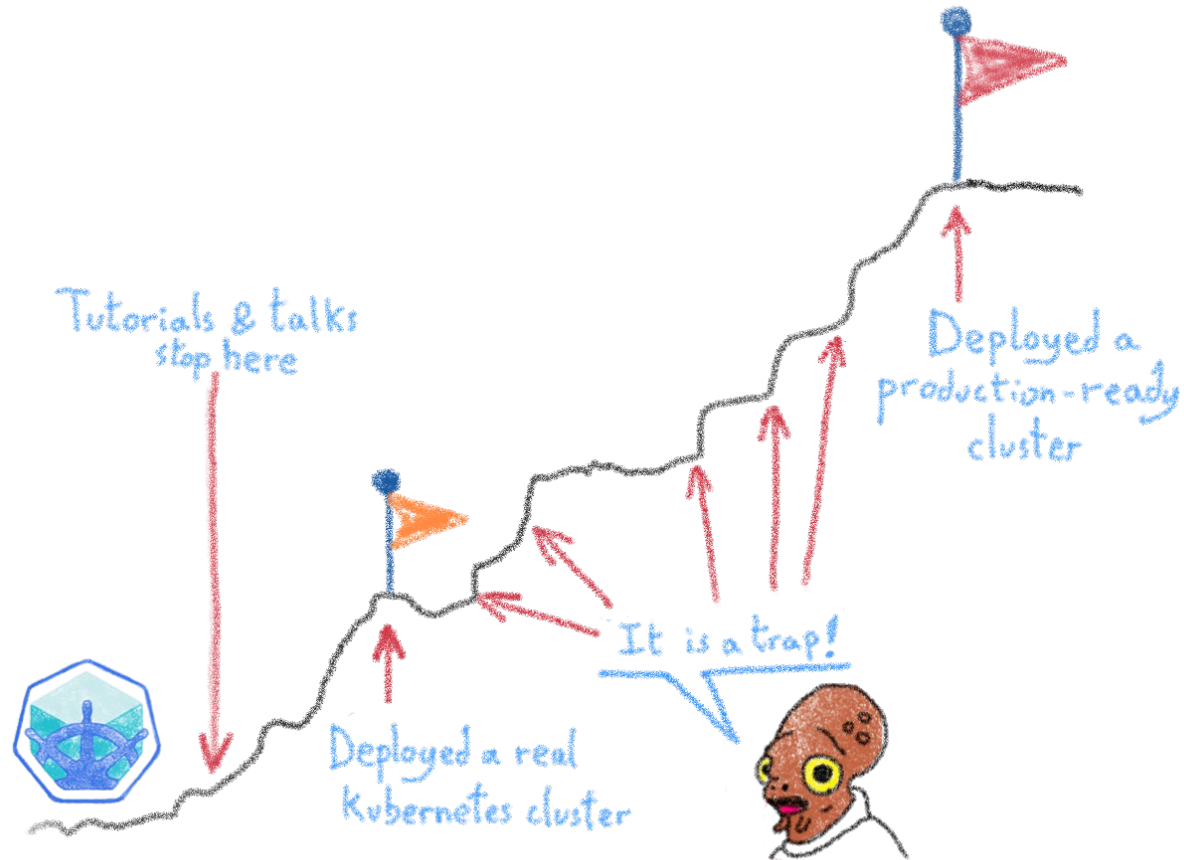
Don't expect real performances

Beyond the first deployment



So I have deployed my distributed architecture on K8s, everything is good now, isn't it?

Minikube is only the beginning

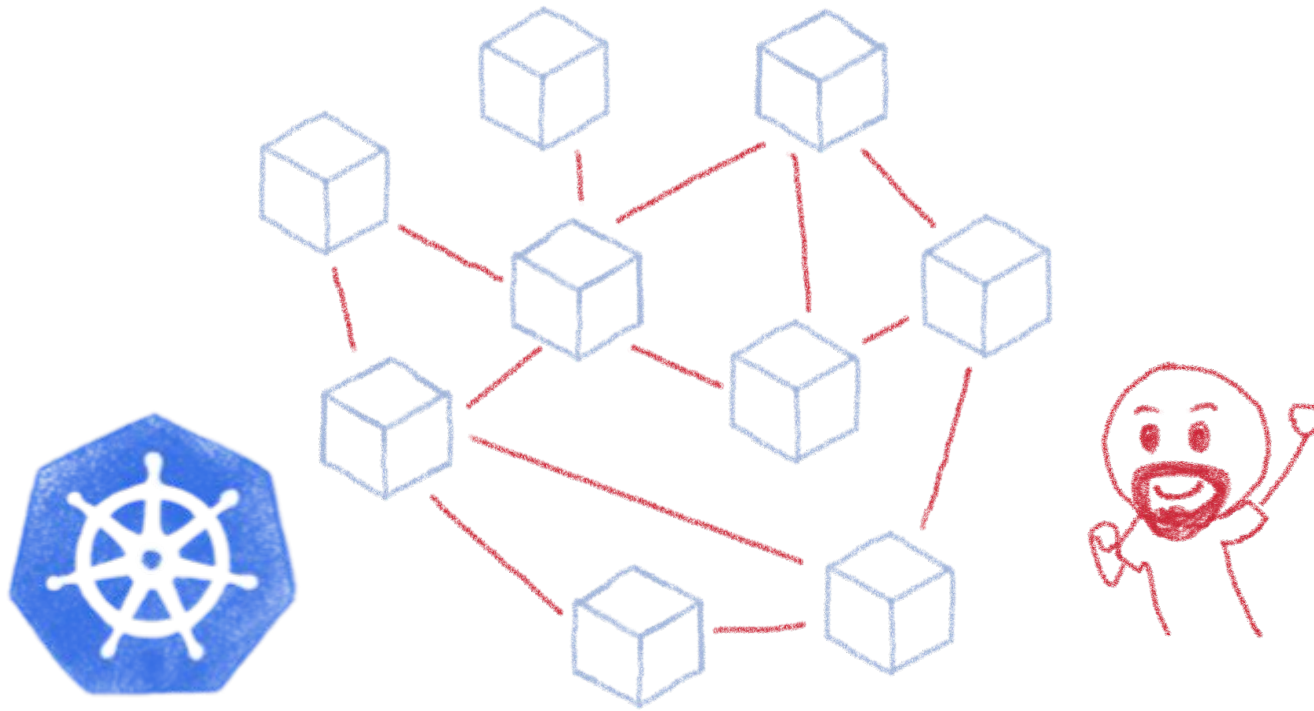


From Minikube to prod

A journey not for the faint of heart

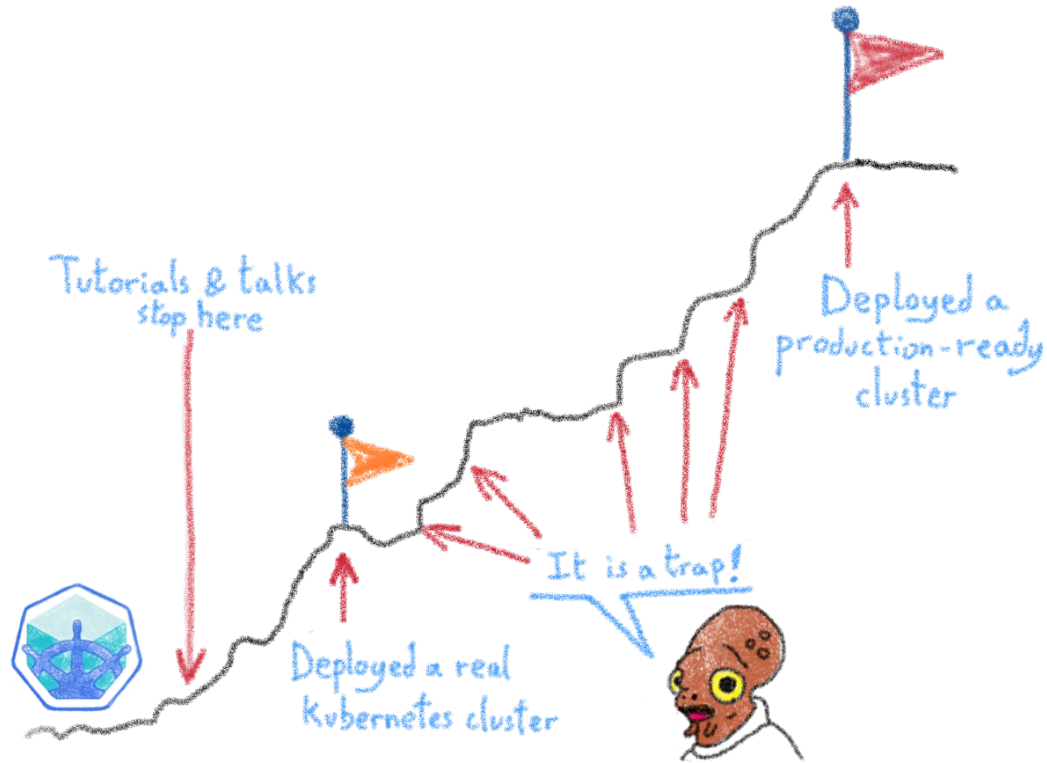


Kubernetes can be wonderful



For both developers and devops

But it comes with a price...



Describing some of those traps

 Security

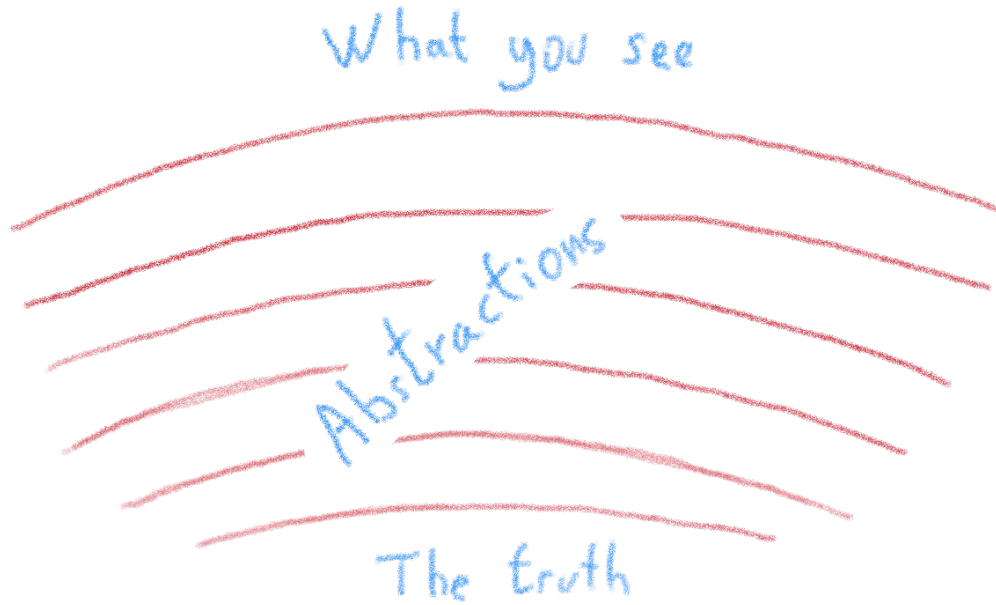
 Deployment

 Monitoring

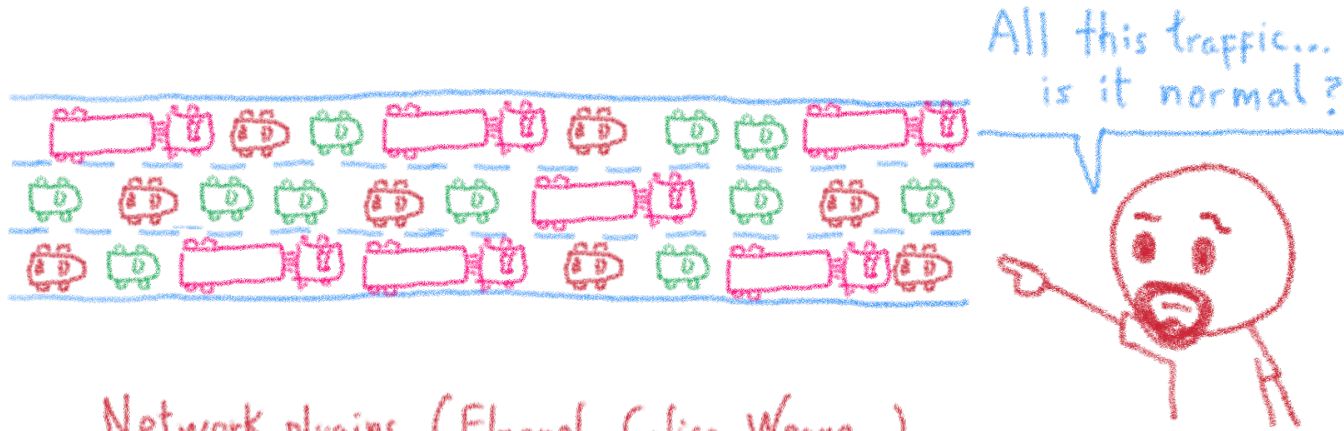
 Backups

To ease and empower your path to production

The truth is somewhere inside...



The network is going to feel it...



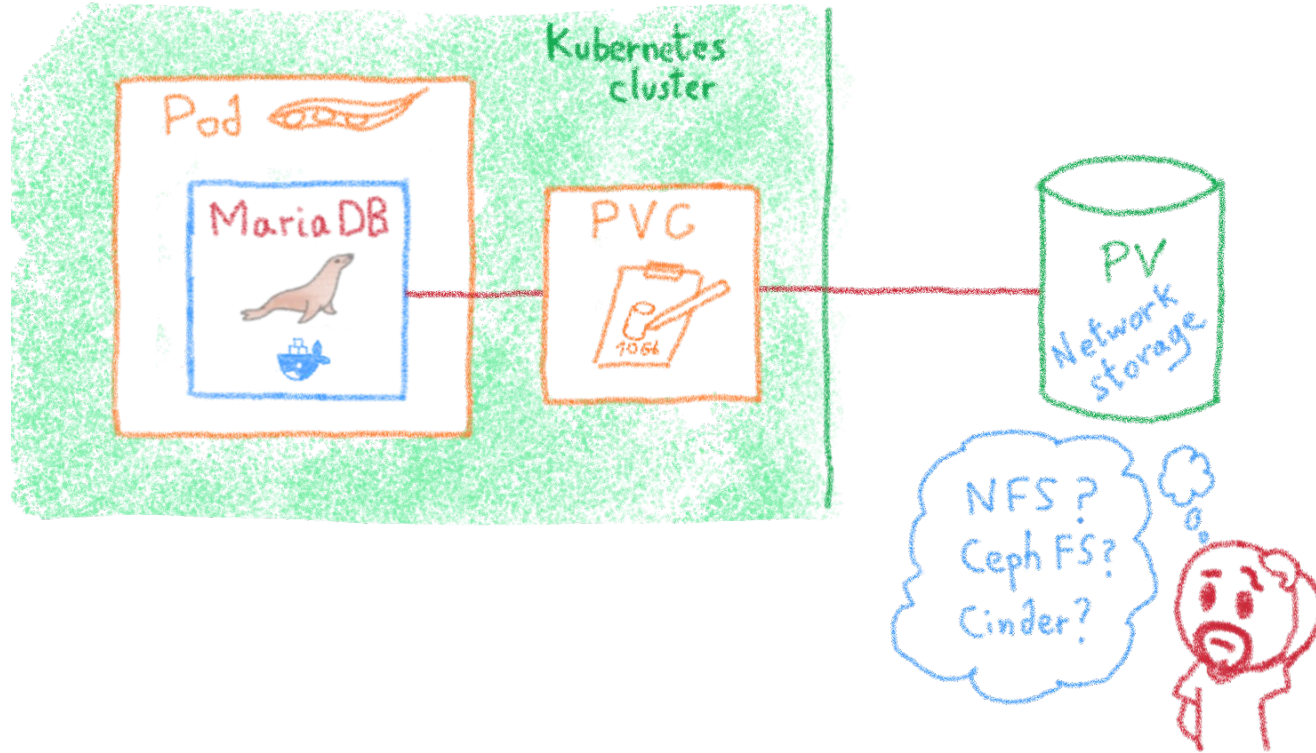
Network plugins (Flannel, Calico, Weave...)

- IPAM - iptables
- routing - crossnode networking

Cluster IP, NodePort, Ingress

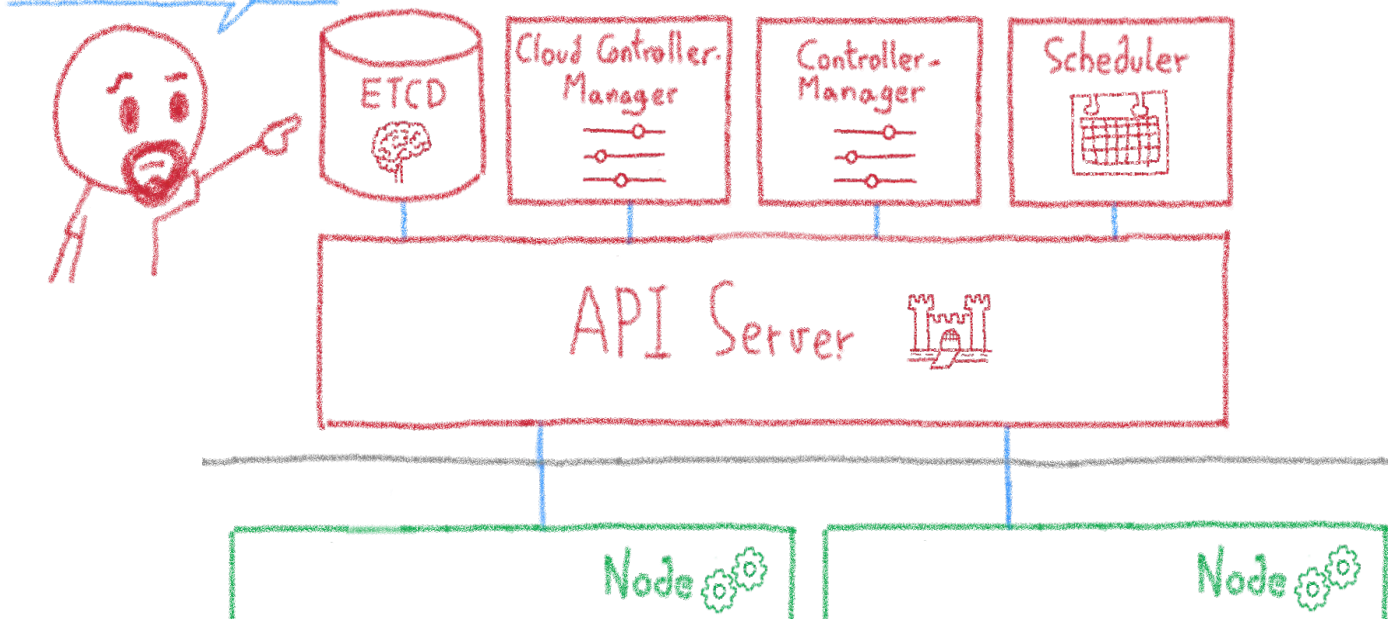
Service Meshes, Istio

The storage dilemma



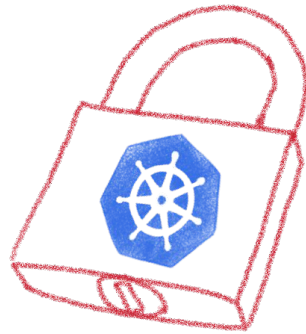
The ETCD vulnerability

A single instance ETCD?
Are you sure?



Security

Hardening your Kubernetes



The security journey

Your security journey

Maturity

- Set up a cluster**
 - Restrict access to kubectl
 - Use RBAC
 - Use a Network Policy
 - Use namespaces
 - Bootstrap TLS
- Prevent known attacks**
 - Disable dashboard
 - Disable default service account token
 - Protect node metadata
 - Scan images for known vulnerabilities
- Follow security hygiene**
 - Keep Kubernetes updated
 - Use a minimal OS
 - Use minimal IAM roles
 - Use private IPs on your nodes
 - Monitor access with audit logging
 - Verify binaries that are deployed
- Prevent/limit impact of microservice compromise**
 - Set a Pod Security Policy
 - Protect secrets
 - Consider sandboxing
 - Limit the identity used by pods
 - Use a service mesh for authentication & encryption

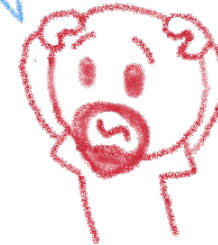
Mattias Gees
@MattiasGees

Your security journey with Kubernetes by @MayaKaczorowski
#GoogleNext18

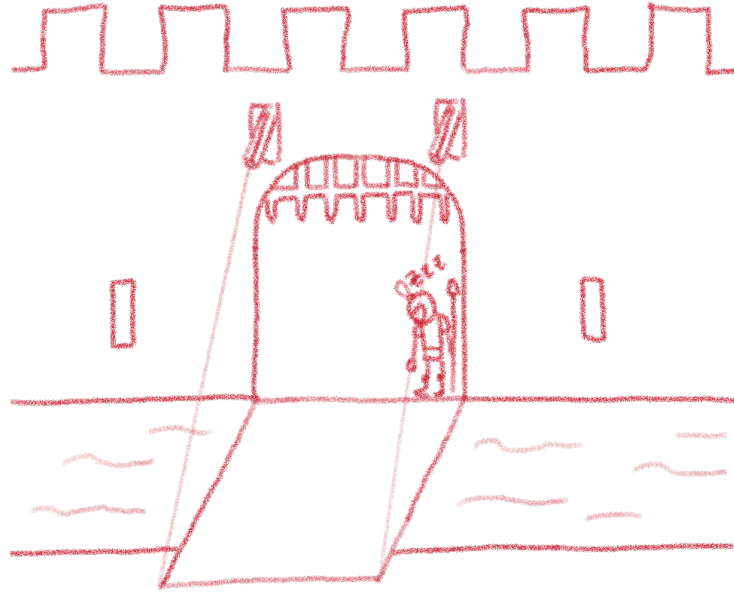
319 12:59 PM - Oct 11, 2018

Open ports (e.g. etcd 2379/TCP)
Kubernetes API (e.g. Tesla hacking)
Exploits (lots of CVEs)
RBAC (e.g. badly defined roles)

Are you kidding me?



Kubernetes is insecure by design



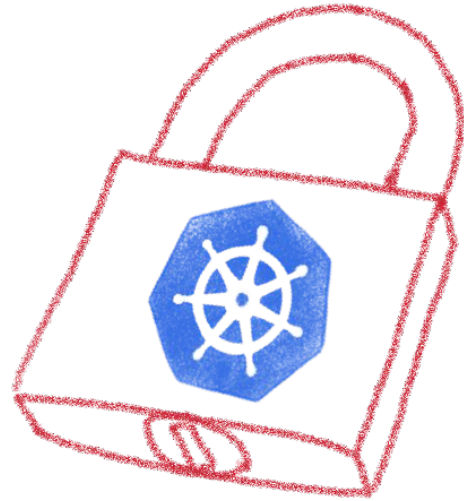
It's a feature, not a bug.

Up to K8s admin to secure it according to needs

Not everybody has the same security needs



Kubernetes allows to enforce security practices as needed



Listing some good practices

- Close open access
- Define and implement RBAC
- Define and implement Network Policies
- Isolate sensitive workloads



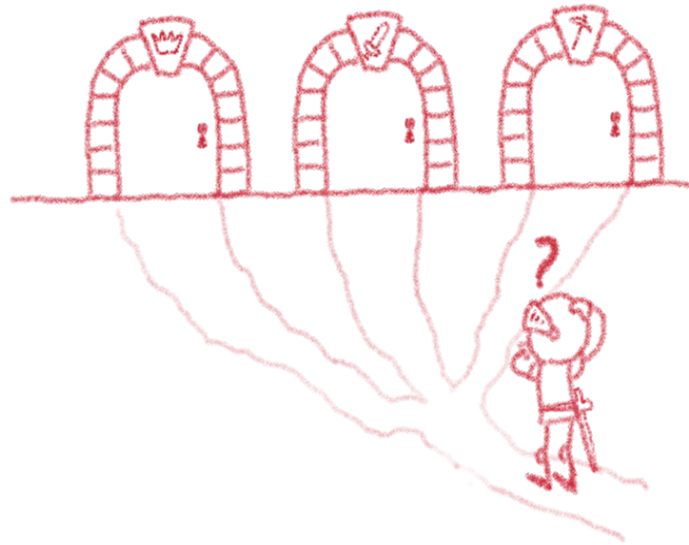
Close open access



Close all by default, open only the needed ports
Follow the least privileged principle

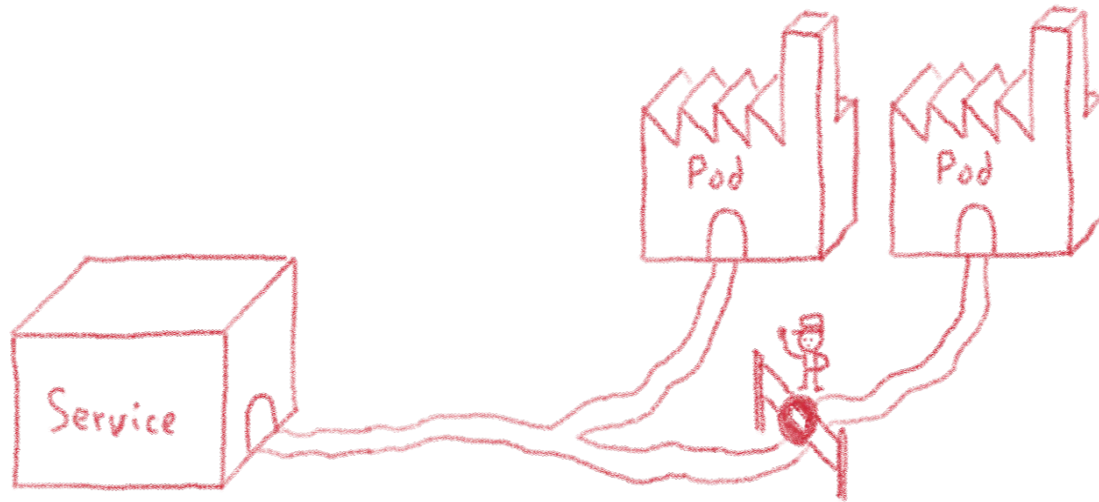
Define and implement RBAC

RBAC: Role-Based Access Control

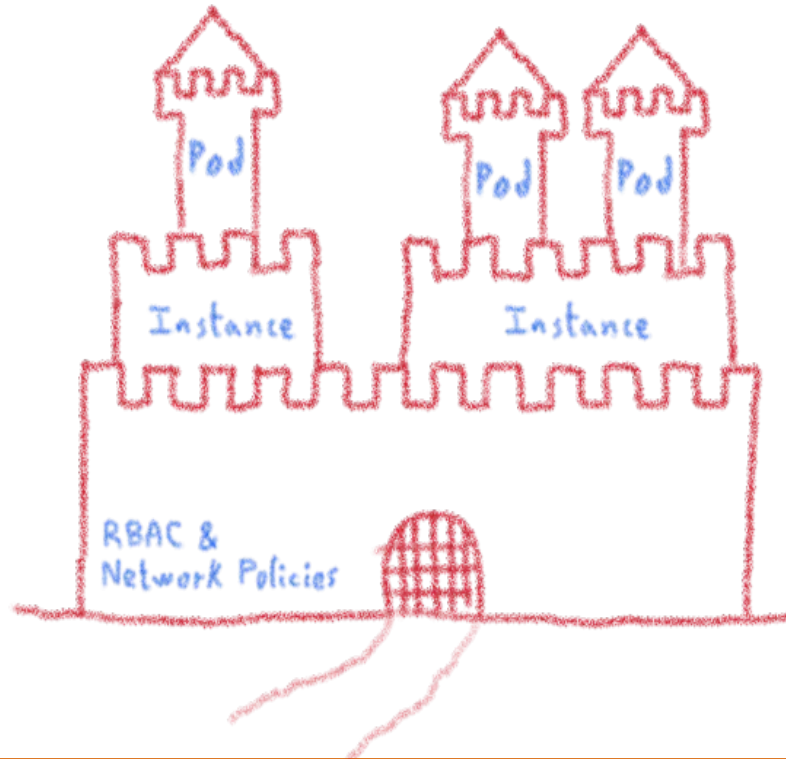


According to your needs

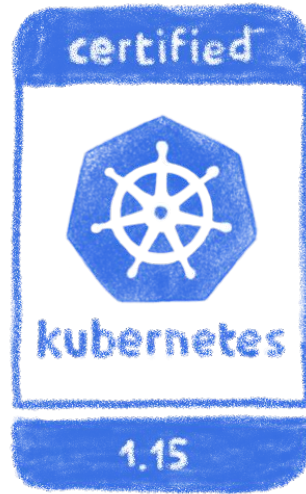
Define and implement network policies



Use RBAC and Network Policies to isolate your sensitive workload



Always keep up to date



Both Kubernetes and plugins

And remember, even the best can get hacked



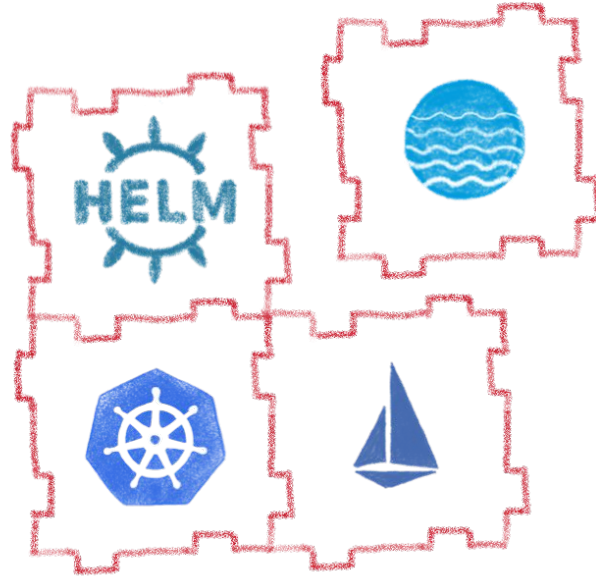
One of Tesla's cluster got hacked
via an unprotected K8s API endpoint,
and was used to mine cryptocurrency...

Remain attentive, don't get too confident

Extensibility

Enhance your Kubernetes

Kubernetes is modular



Fully extensible

- Kubernetes API
- Cluster demons
- Controllers
- Custom resources
- ...

Operators

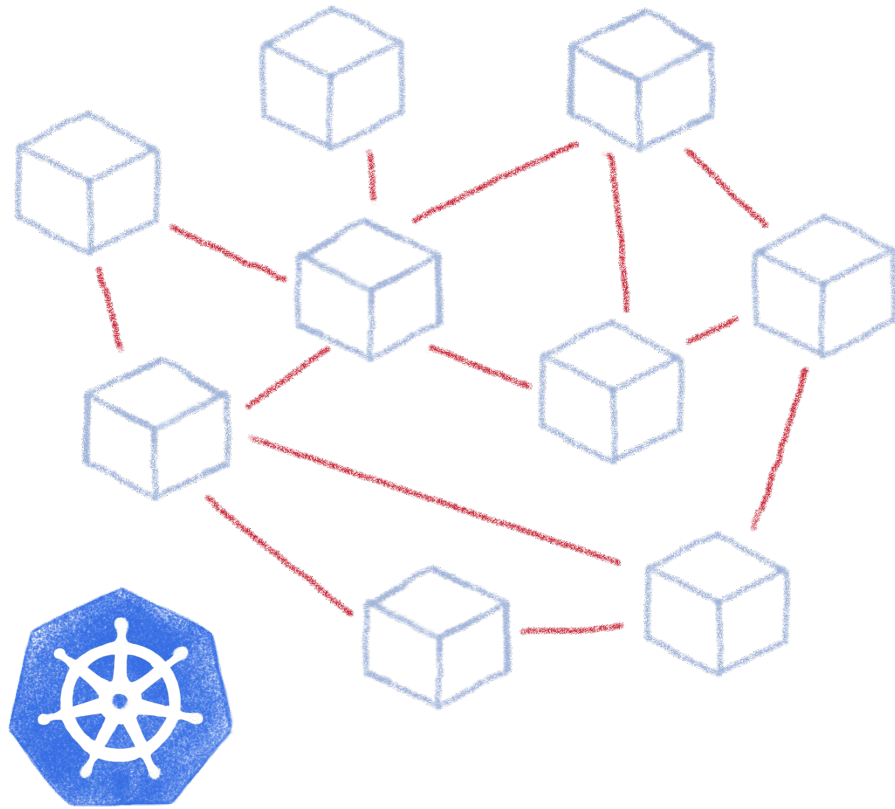
Let's see how some of those plugins can help you

Helm

A package management for K8s



Complex deployments



Ingress

Services

Deployments

Pods

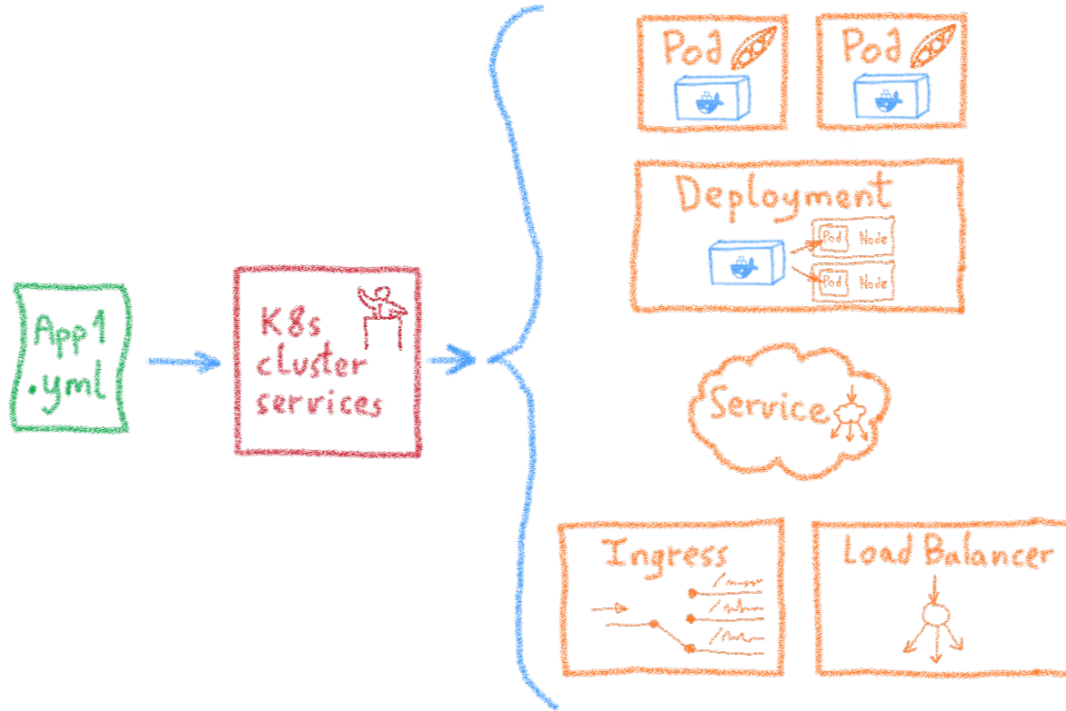
Sidecars

Replica Sets

Stateful Sets



Using static YAML files

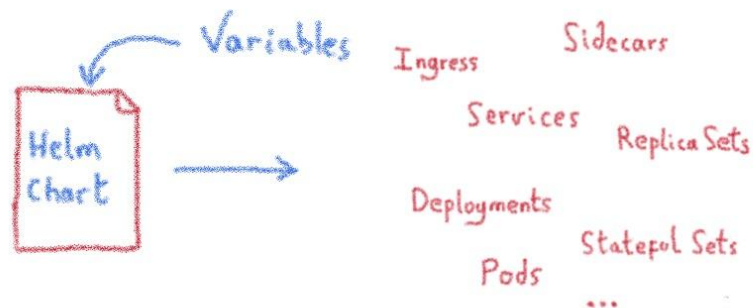


But if I need to customize things?



Complex deployments

A package manager for Kubernetes



- Manage complexity 

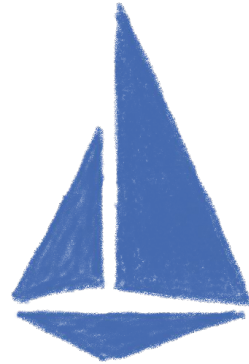
- Simple sharing 

- Easy upgrades 

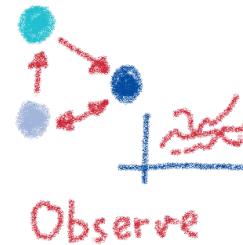
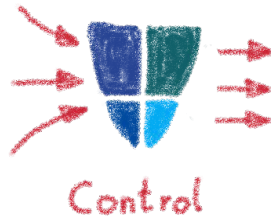
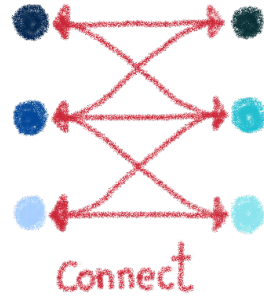
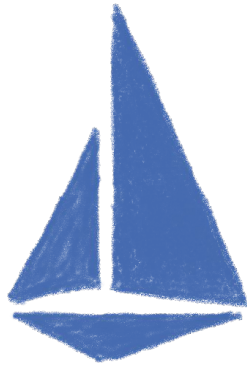
- Easy rollbacks 

Istio

**A service mesh for Kubernetes...
and much more!**



Istio: A service mesh... but not only



Rolling upgrades

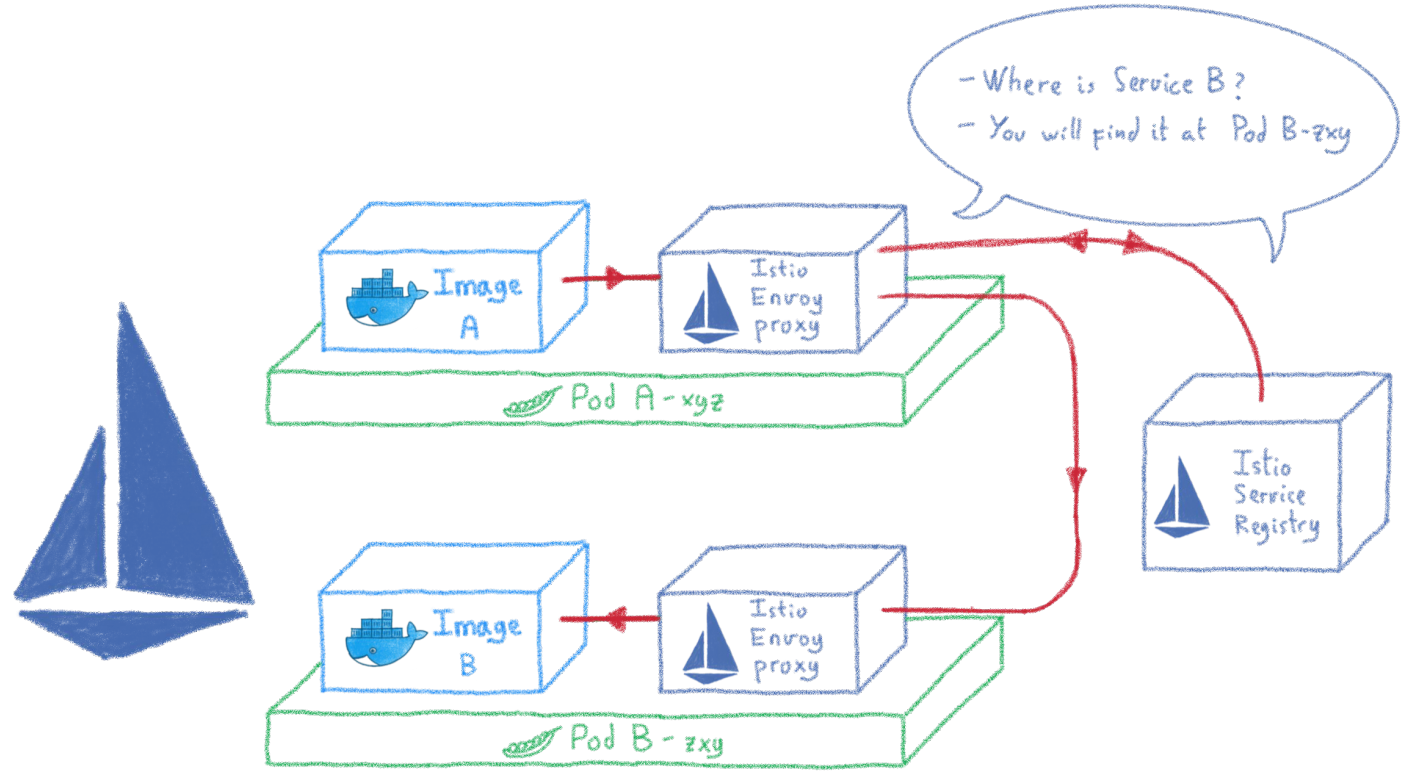
A/B Testing

Canary Testing

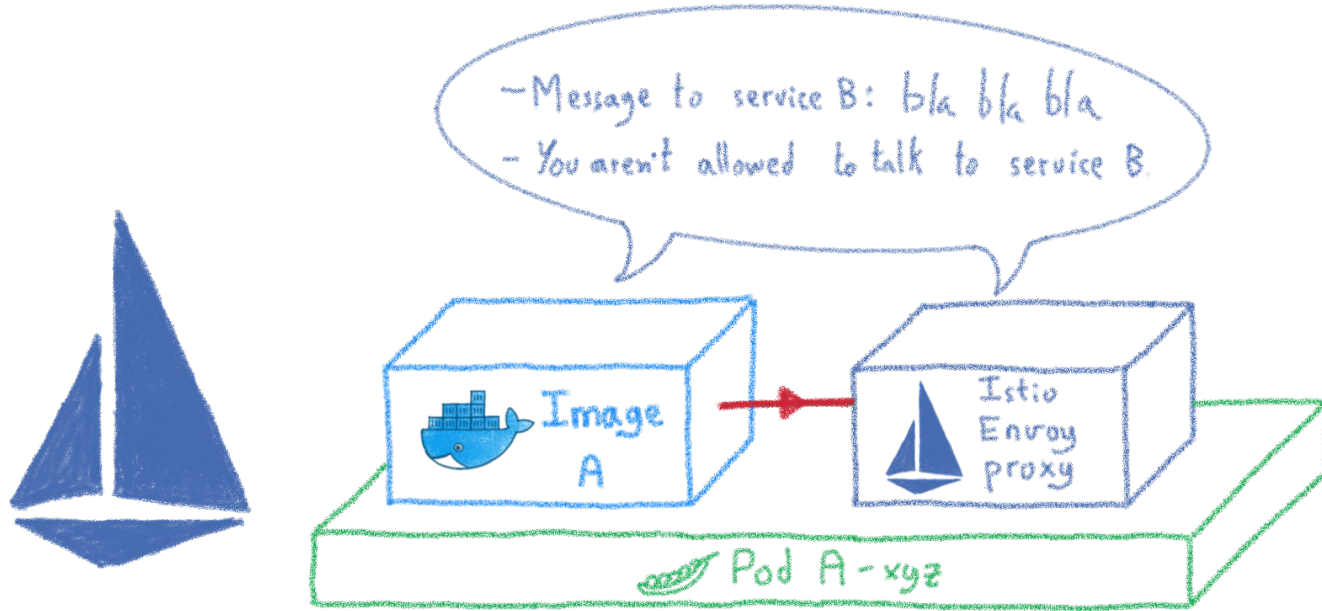
Edge traffic management

Multicloud service mesh

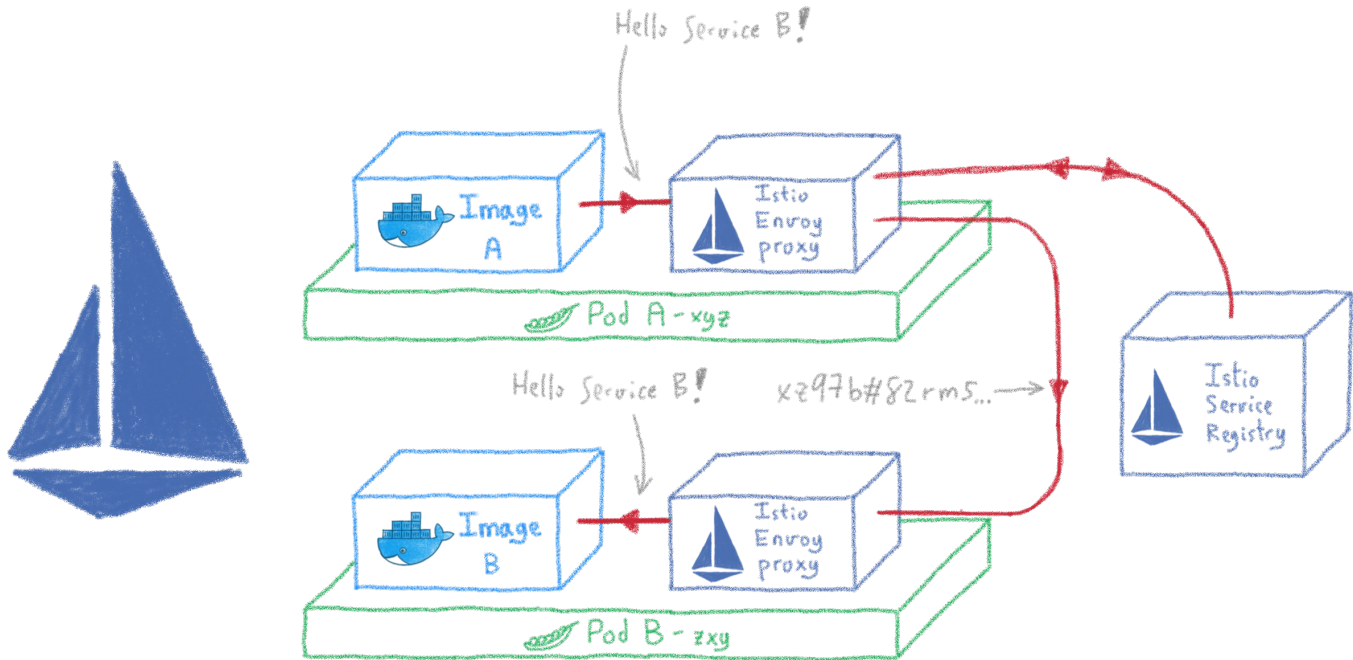
Service discovery



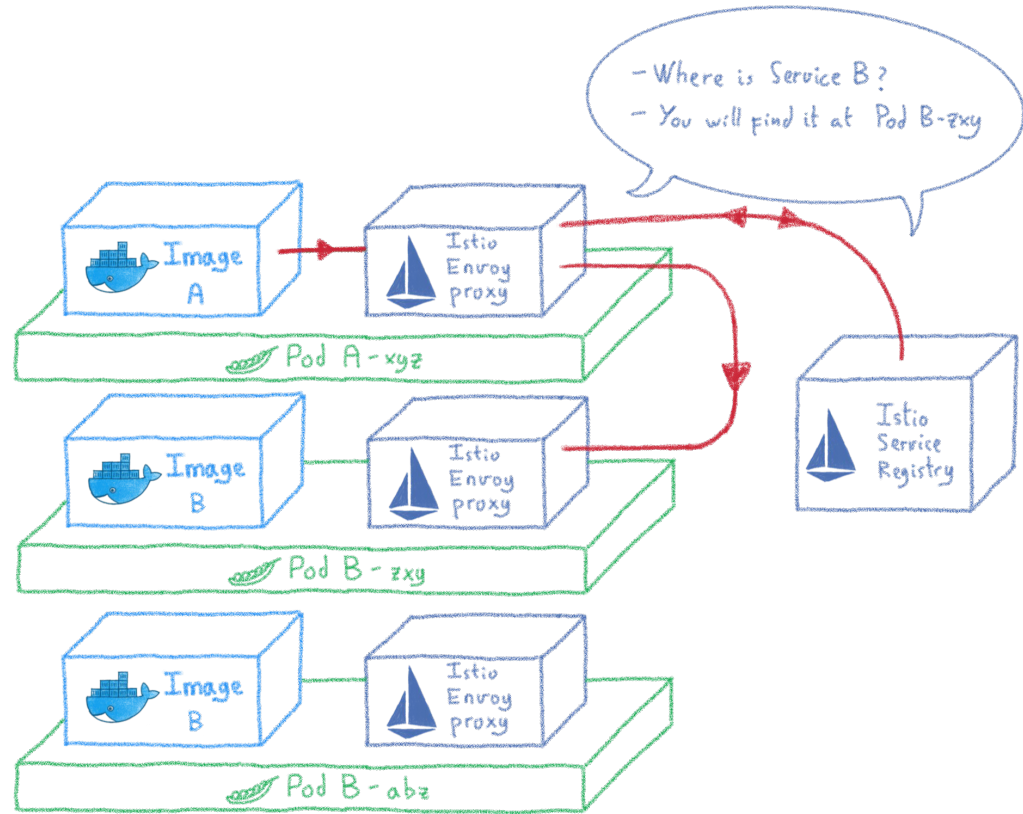
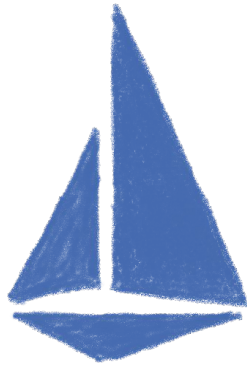
Traffic control



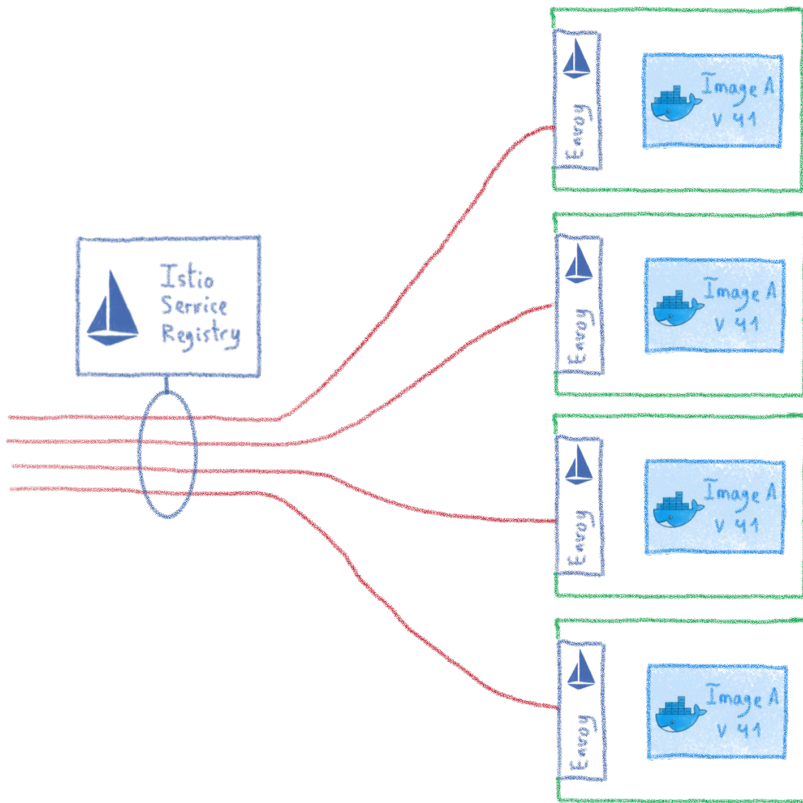
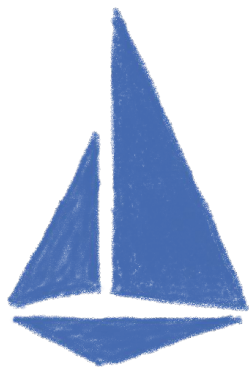
Encrypting internal communications



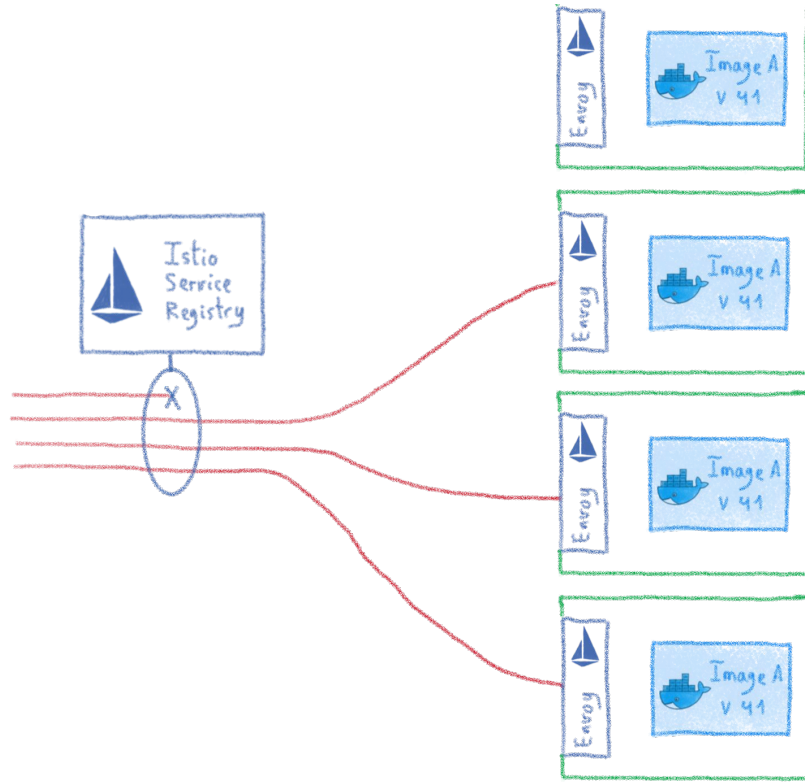
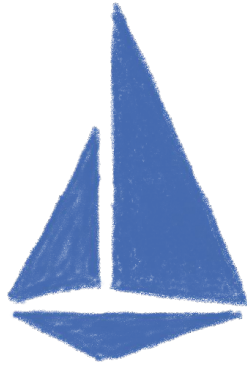
Routing and load balancing



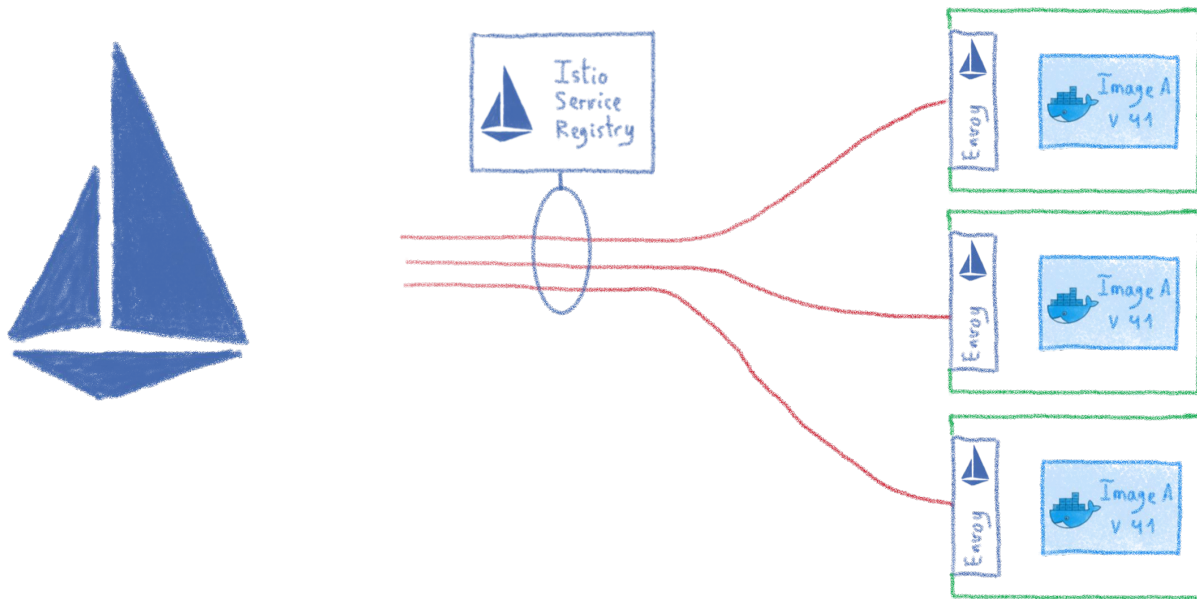
Rolling upgrades



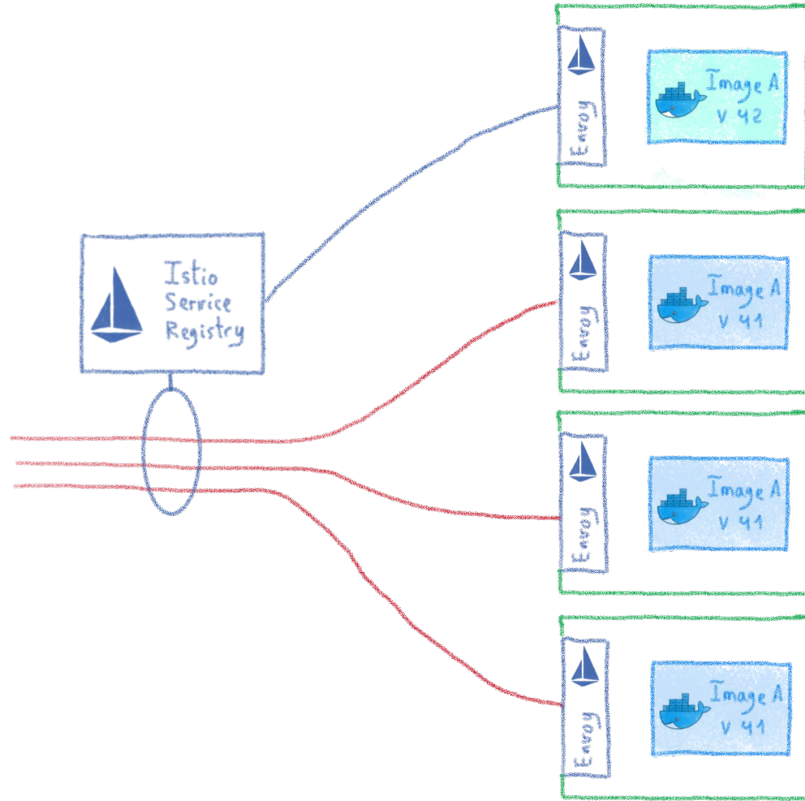
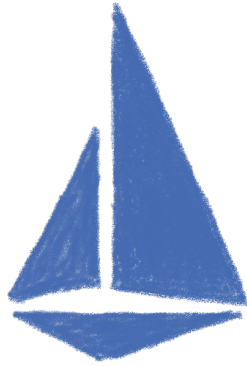
Rolling upgrades



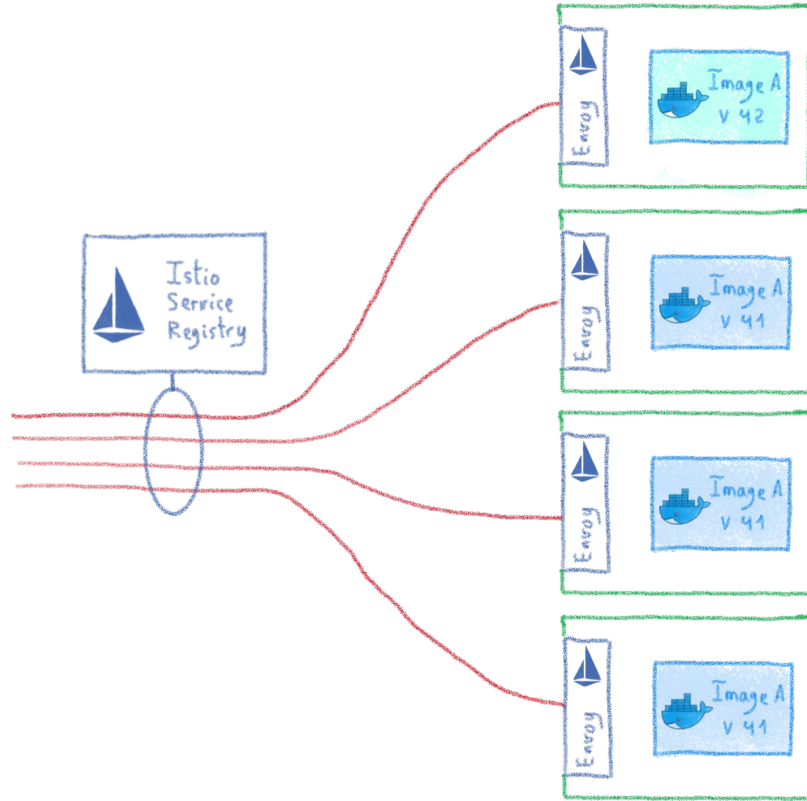
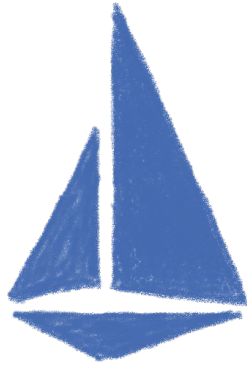
Rolling upgrades



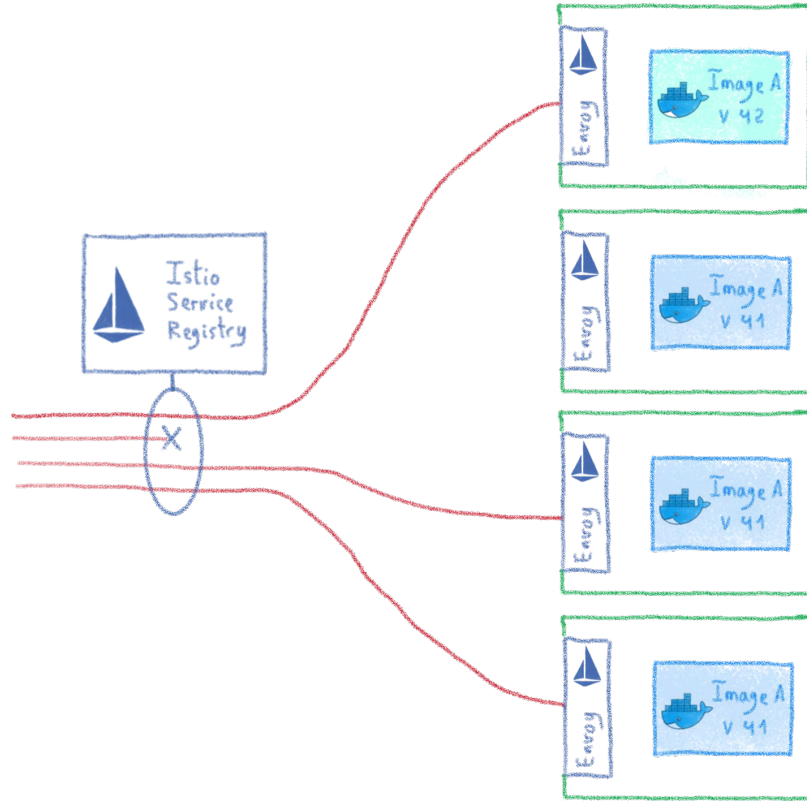
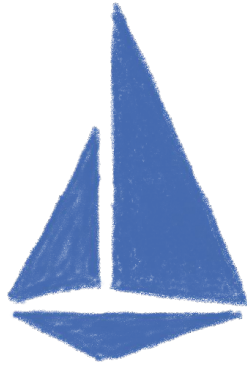
Rolling upgrades



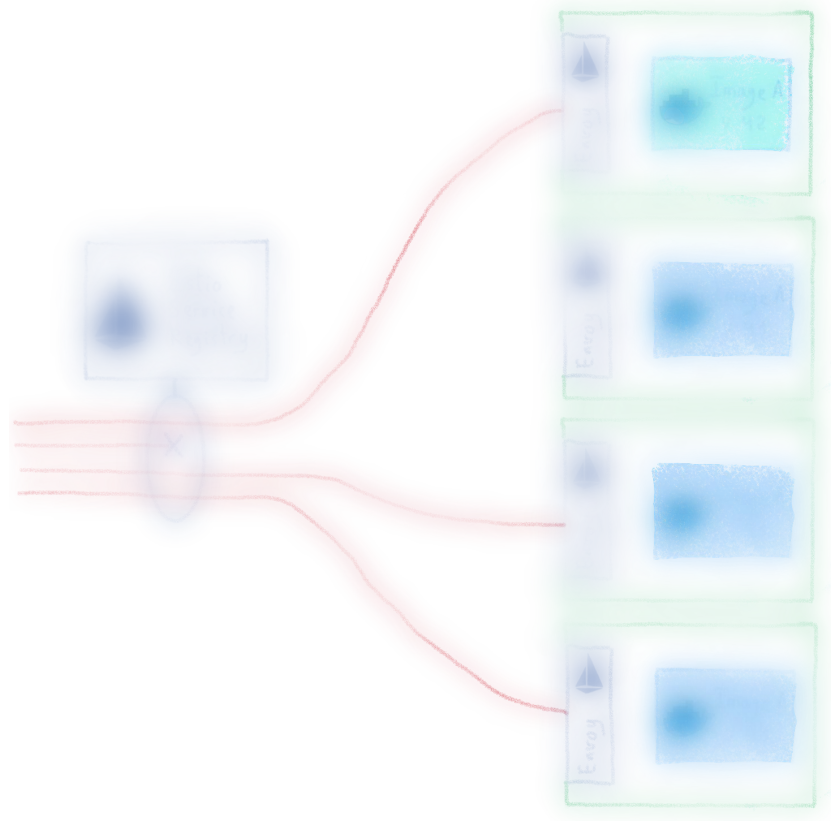
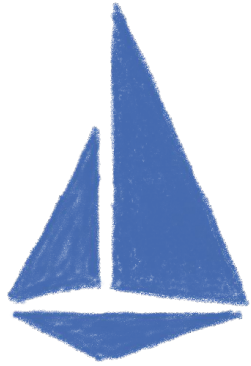
Rolling upgrades



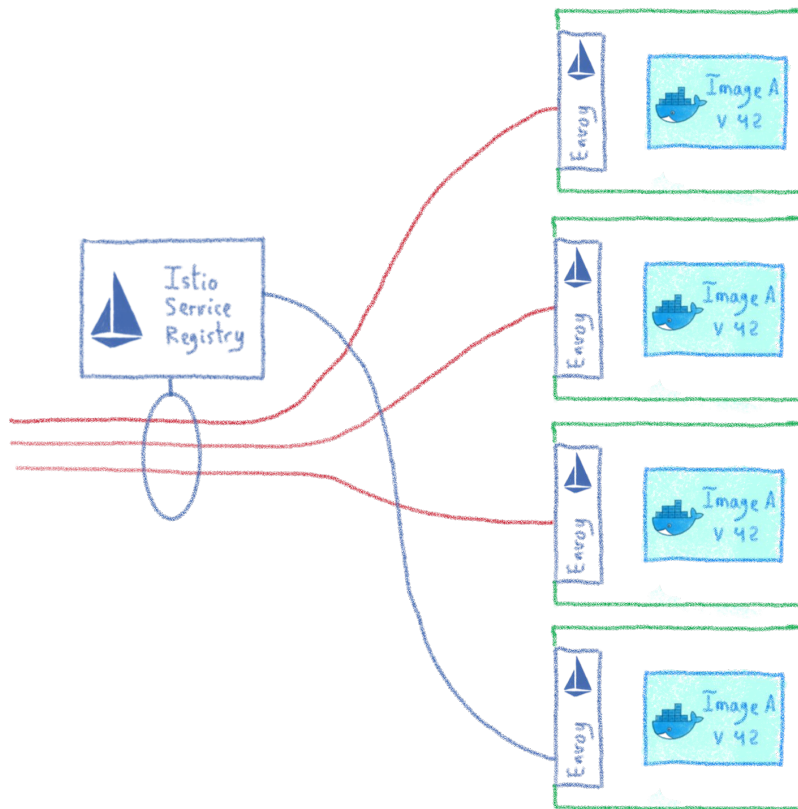
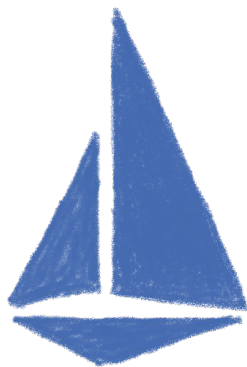
Rolling upgrades



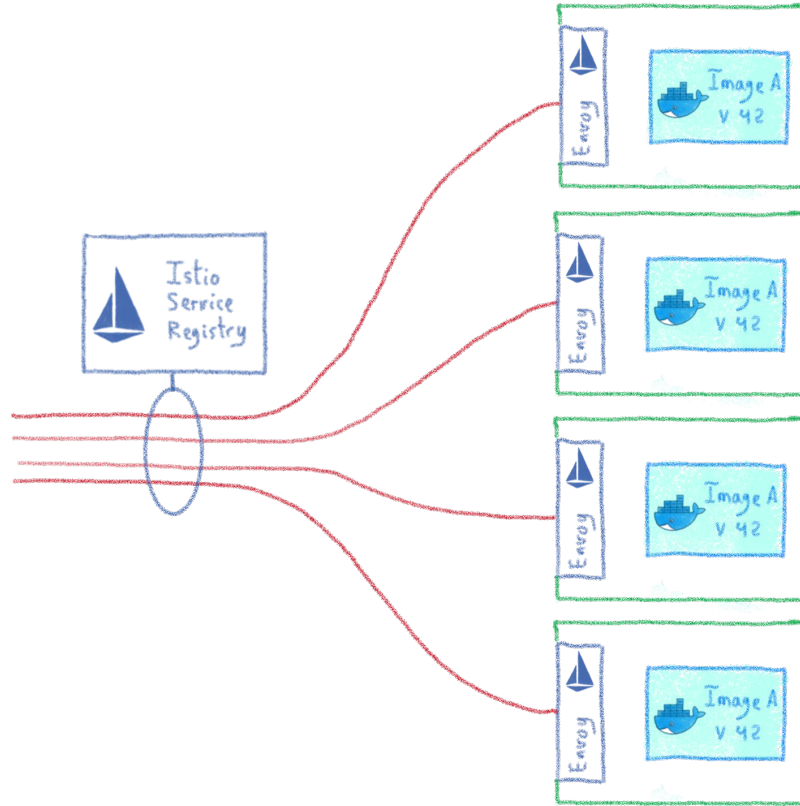
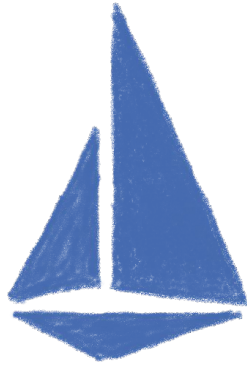
Rolling upgrades



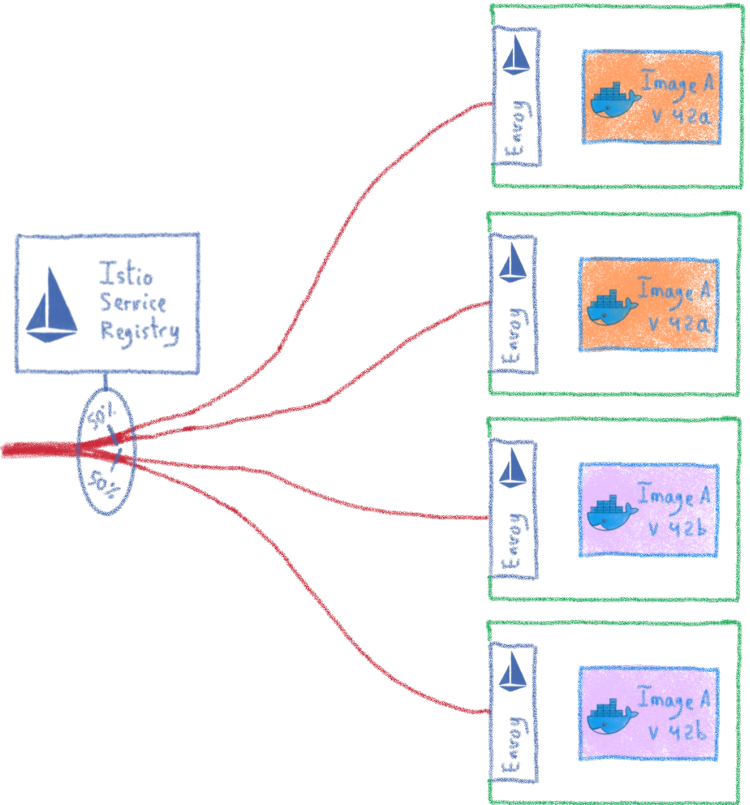
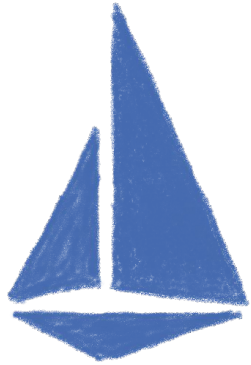
Rolling upgrades



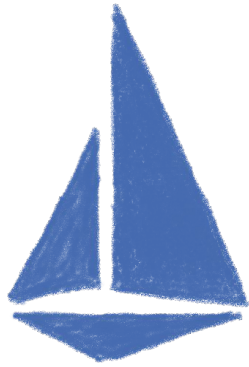
Rolling upgrades



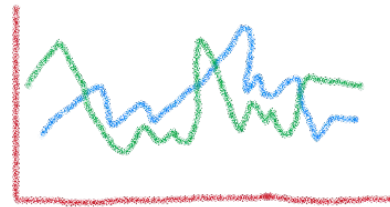
A/B testing



Monitoring your cluster



- Metrics
 - Logs
 - Tracing
- } at {
- Envoy level
 - Control plane level



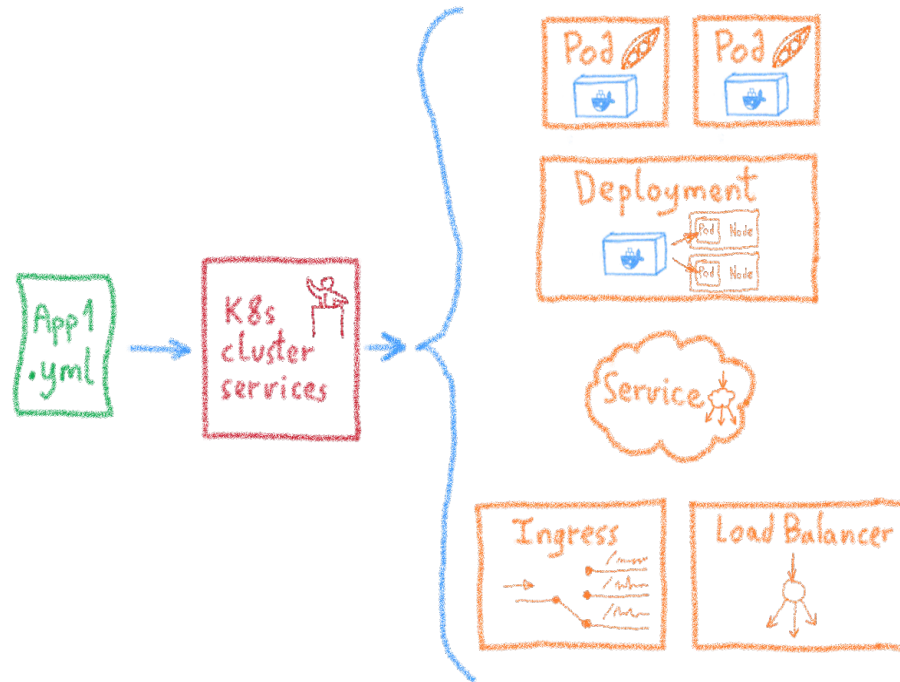
Dashboards

Velero

Backing up your Kubernetes

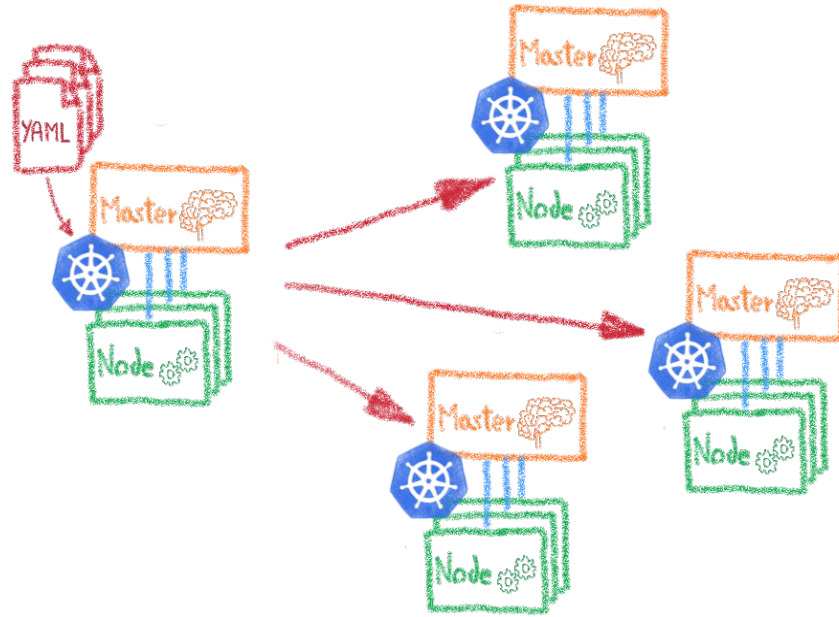


Kubernetes: Desired State Management



- Ingress
- Services
- Deployments
- Pods
- Sidecars
- Replica Sets
- Stateful Sets

YAML files allows to clone a cluster



Dev envs

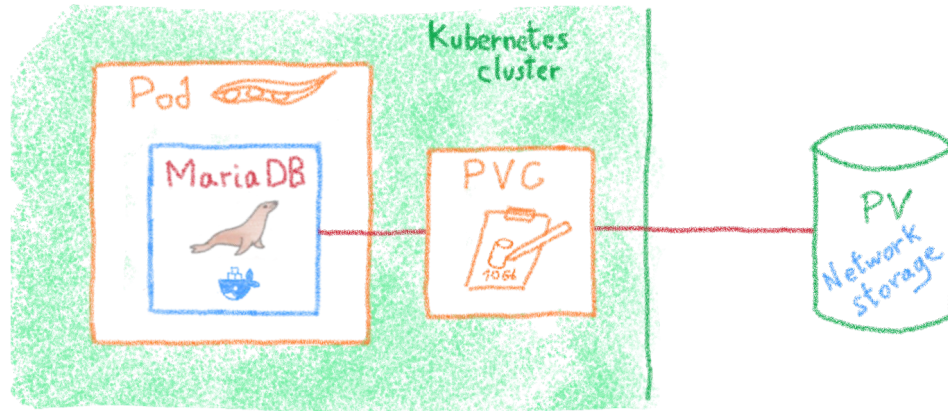
Staging

Multi-cluster

Multi-cloud



But what about the data?

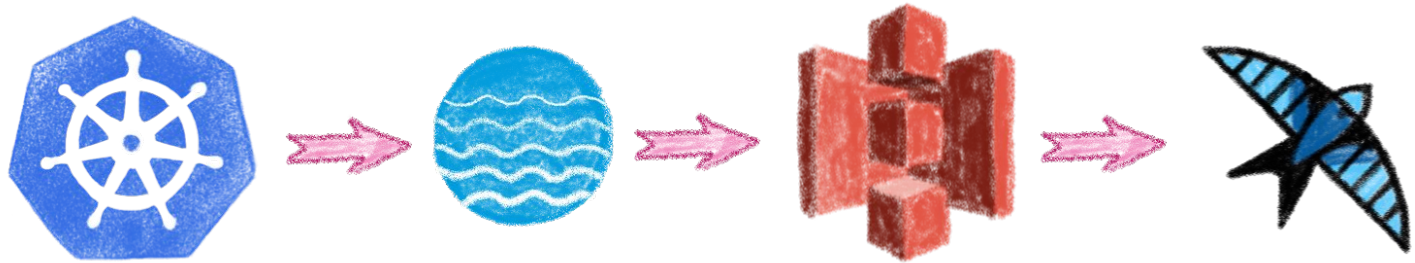


Velero



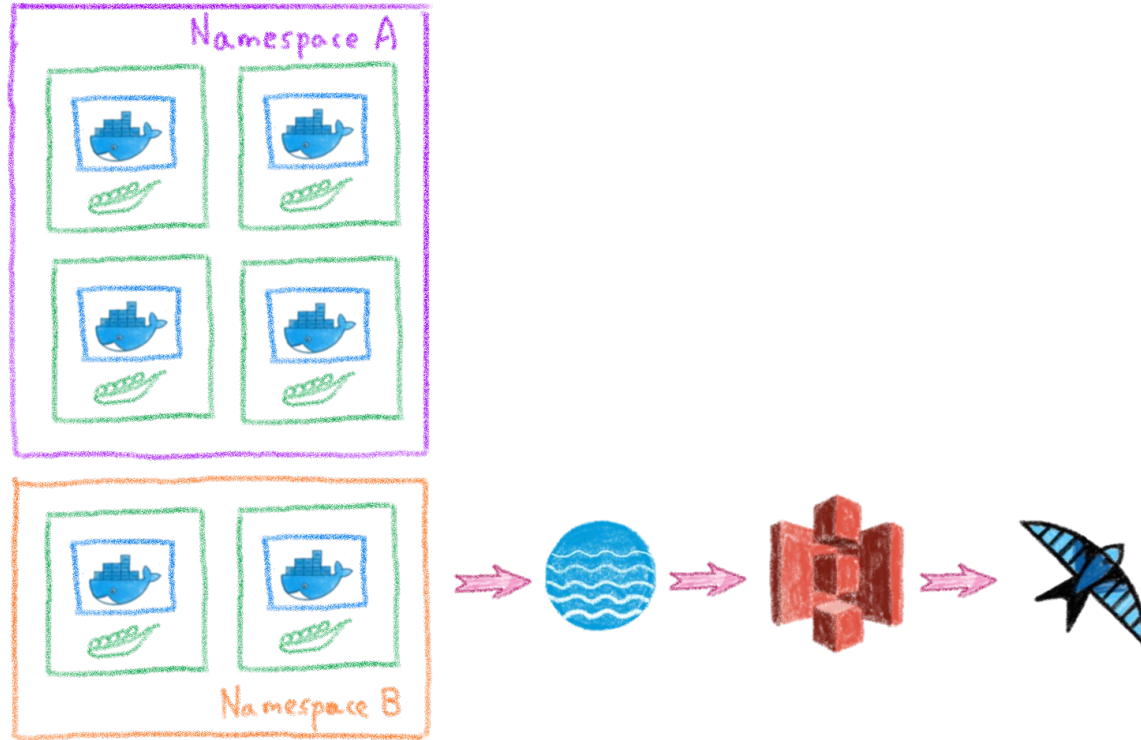
Backup and migrate Kubernetes applications
and their persistent volumes

S3 based backup

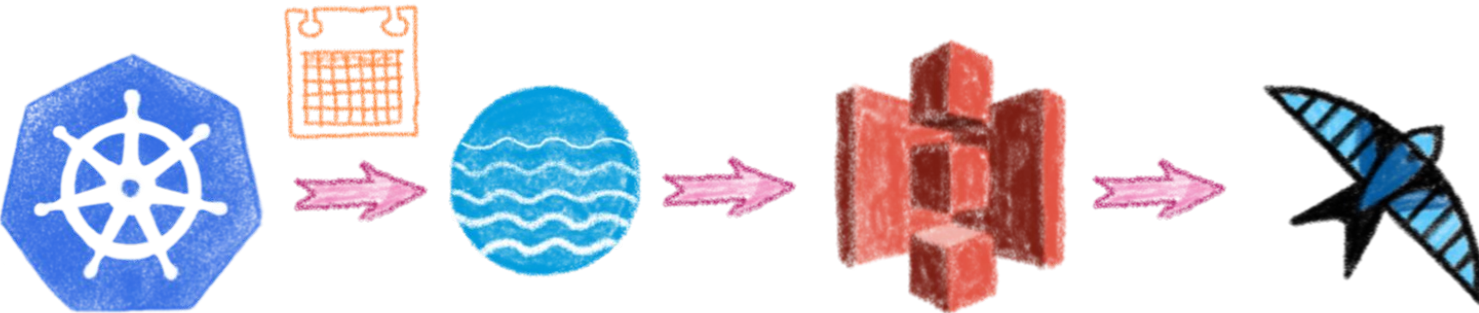


On any S3 protocol compatible store

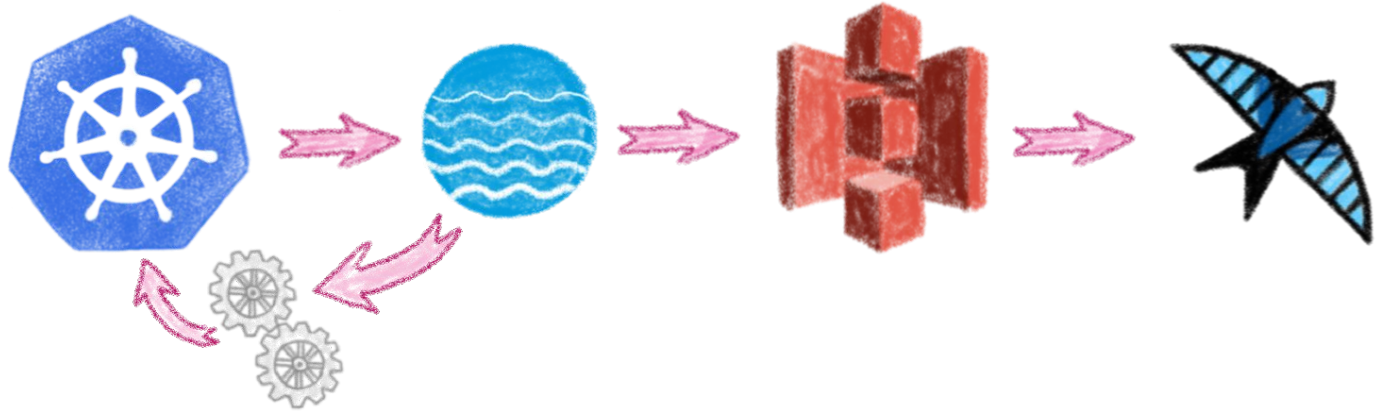
Backup all or part of a cluster



Schedule backups



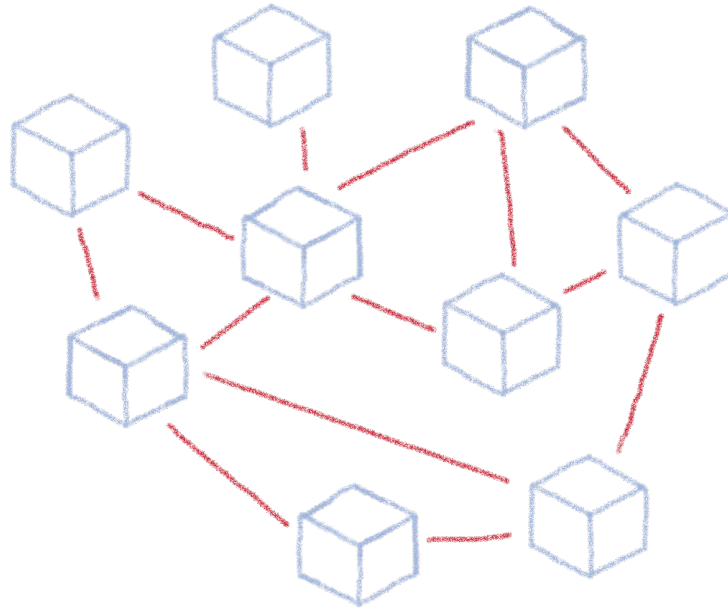
Backups hooks



Conclusion

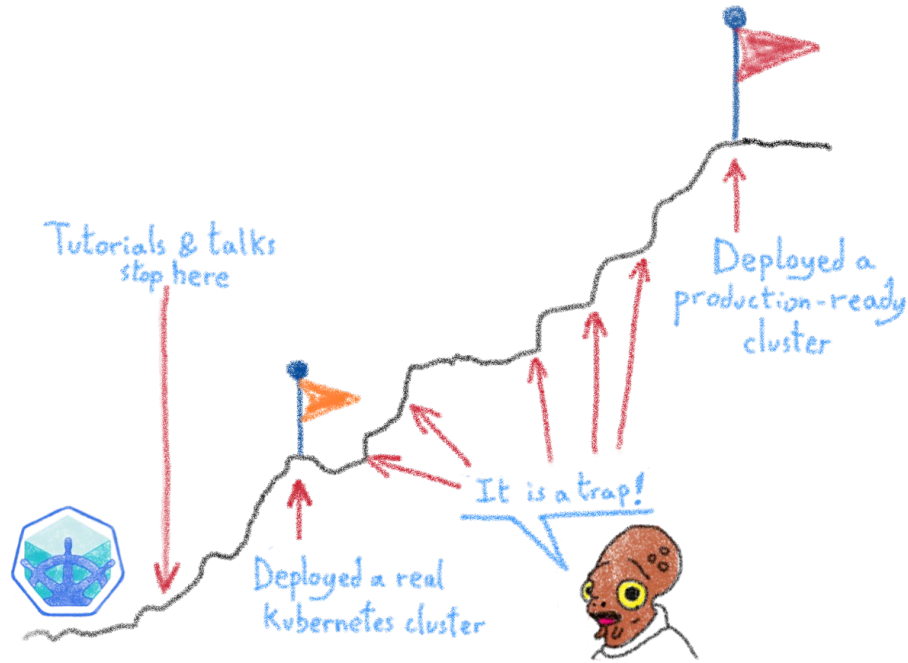
And one more thing...

Kubernetes is powerful







It can make Developers' and
DevOps' lives easier

But there is a price: operating it



Lot of things to think about

We have seen some of them

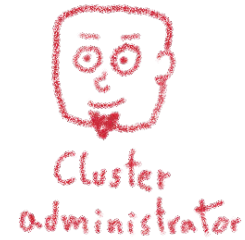
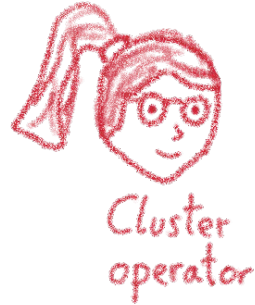
-  Security
-  Deployment
-  Monitoring
-  Backups

One more thing...

Who should do what?



Different roles



Each role asks for very different knowledge and skill sets

Most companies don't need to operate the clusters



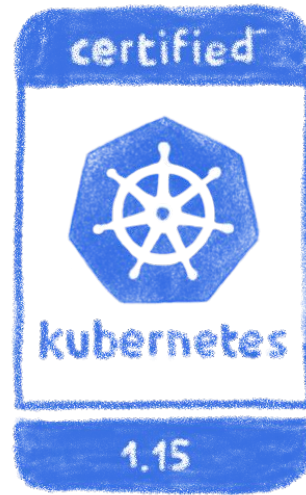
Developer



Cluster
administrator

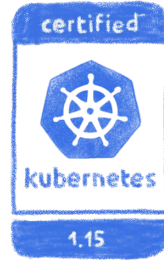
As they don't build and rack
their own servers!

If you don't need to build it, choose a certified managed solution



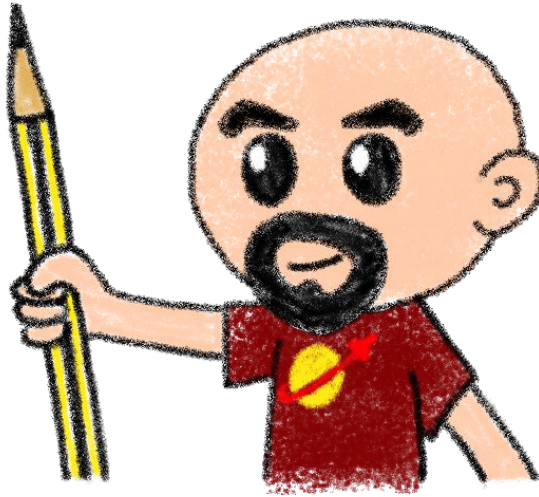
You get the cluster, the operator
get the problems

Like our OVH Managed Kubernetes



Made with  by the Platform team

Do you want to try?



Send me an email to get some vouchers..

horacio.gonzalez@corp.ovh.com

Thank you for listening

