Montpellier - 2020-12-18

Kubernetes for DevOps... and Devs

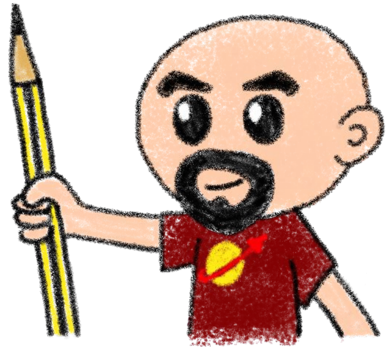
Horacio Gonzalez

@LostInBrittany

OVHcloud

Who are we?

Introducing myself and introducing ~~OVH~~ OVHcloud

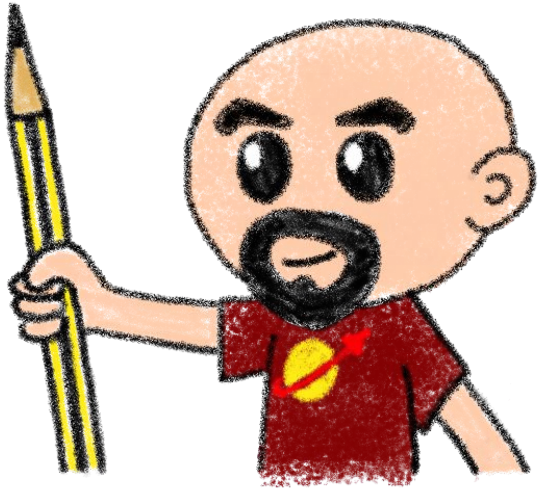


Horacio Gonzalez

@LostInBrittany

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

OVH
Team DevRel



DevFest du
Bout du Monde

Finist
Devs

GDG
Finistère

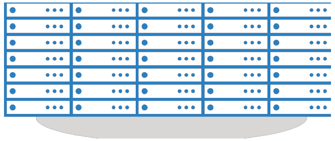
Google Developers
Experts
2019

Web Technologies
GDE
Flutter

OVHcloud: A Global Leader

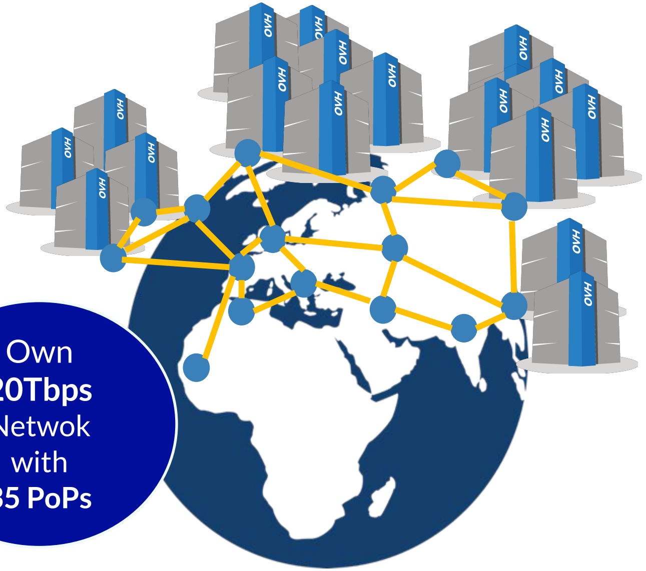
250k Private cloud VMs running

1 **Dedicated IaaS Europe**



Hosting capacity : **1.3M** Physical Servers

360k Servers already deployed



Own **20Tbps** Netwok with **35 PoPs**

30 Datacenters

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

OVHcloud: Our solutions

Cloud

- VPS
- Public Cloud
- Private Cloud
- Serveur dédié
- Cloud Desktop
- Hybrid Cloud

Mobile Hosting

- Containers
- Compute
- Database
- Object Storage
- Securities
- Messaging

Web Hosting

- Domain names
- Email
- CDN
- Web hosting
- MS Office
- MS solutions

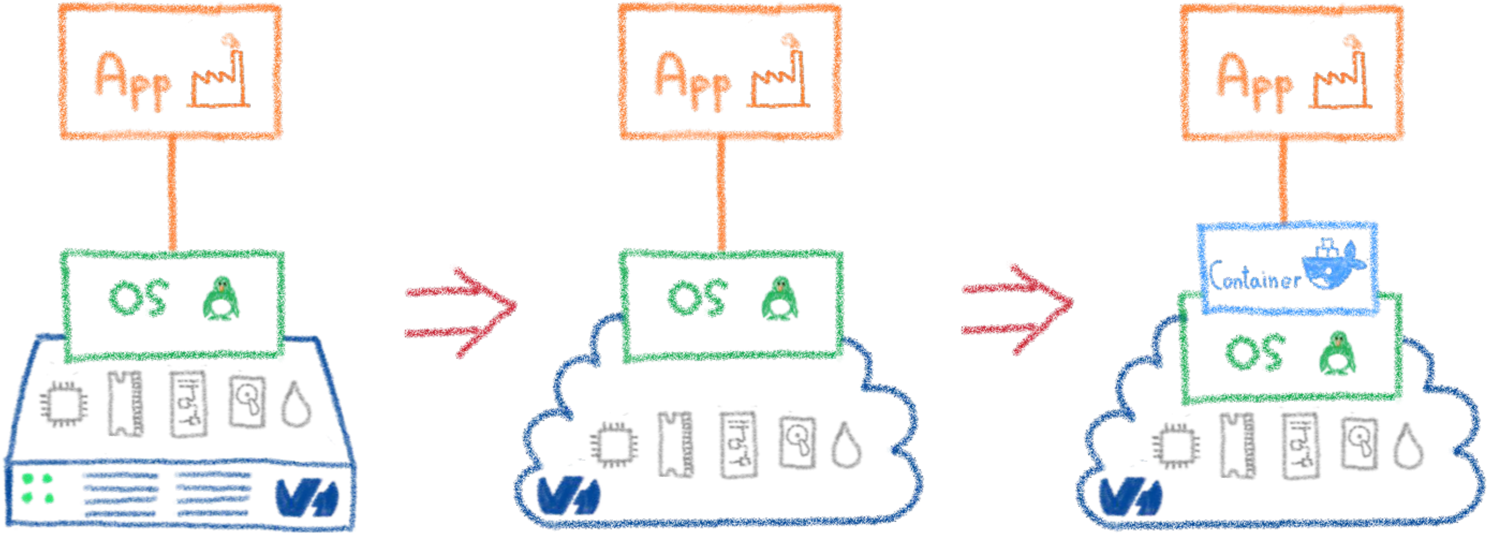
Telecom

- VoIP
- SMS/Fax
- Virtual desktop
- Cloud HubiC
- Over theBox

Orchestrating containers

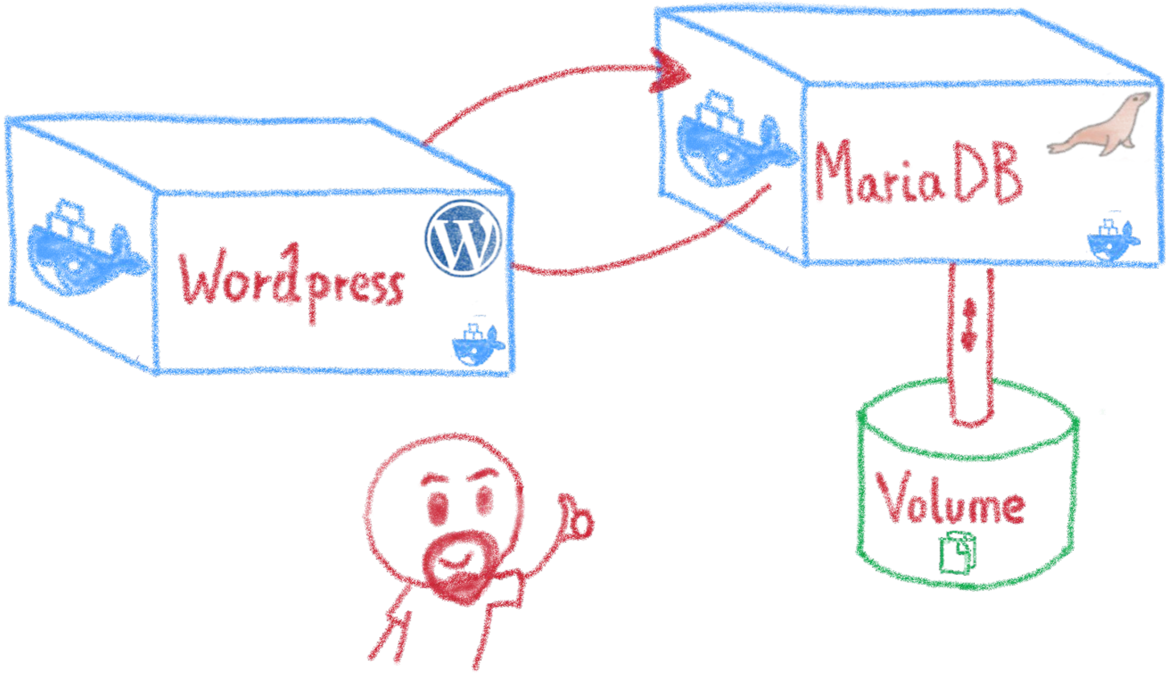
Like herding cats… but in hard mode!

From bare metal to containers



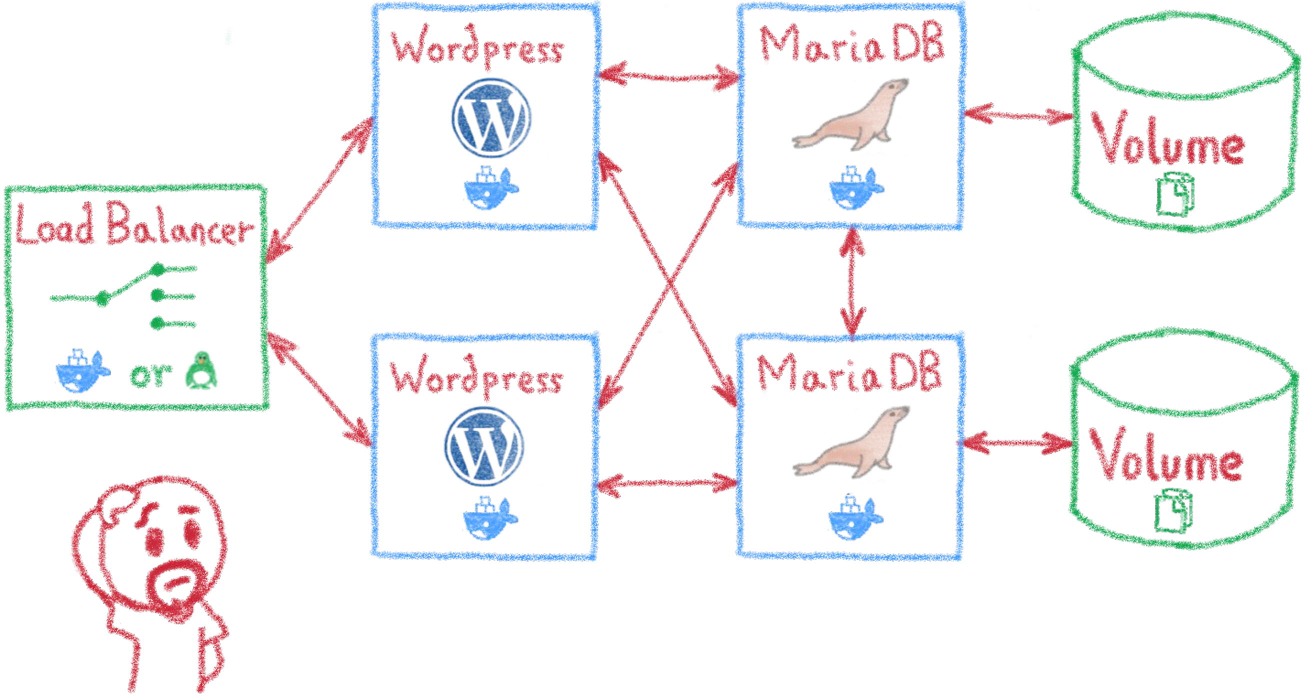
Another paradigm shift

Containers are easy...



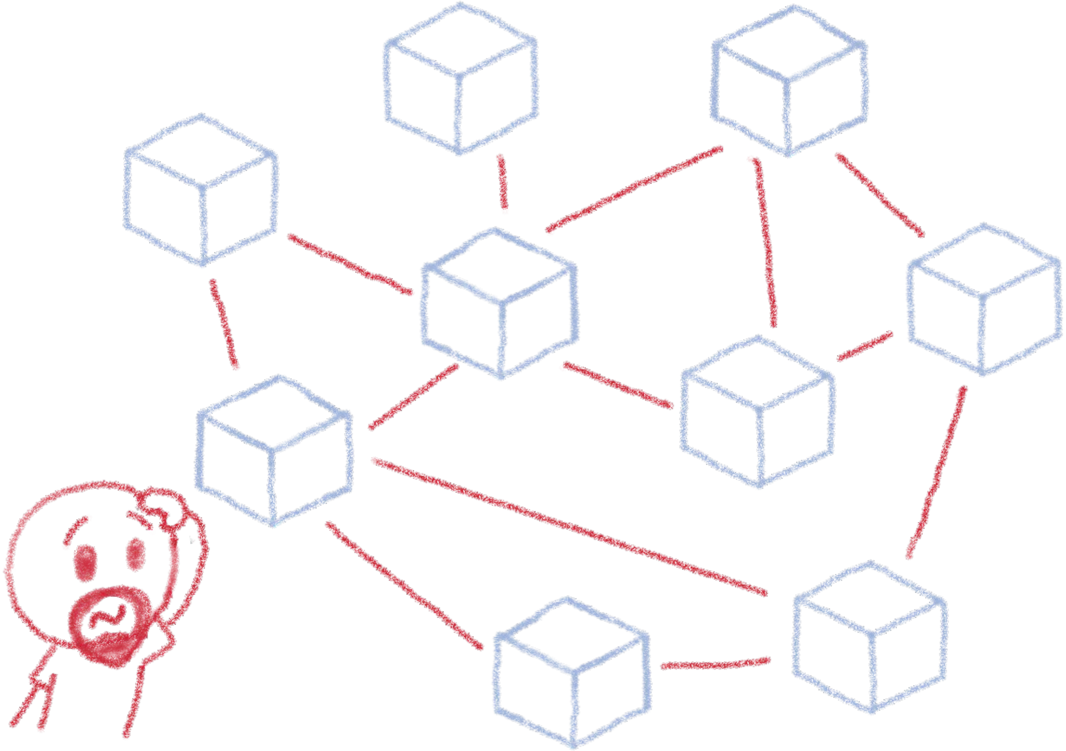
For developers

Less simple if you must operate them



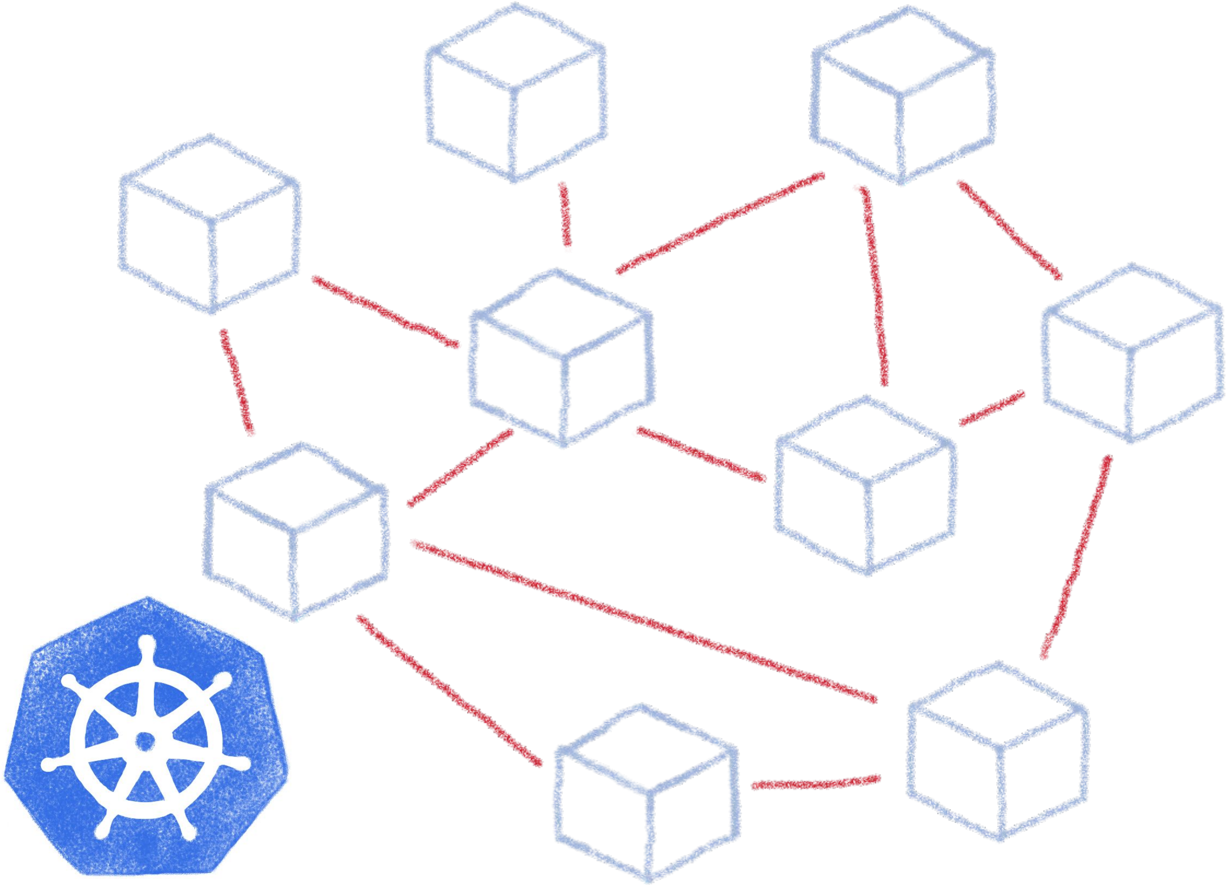
Like in a production context

And what about microservices?



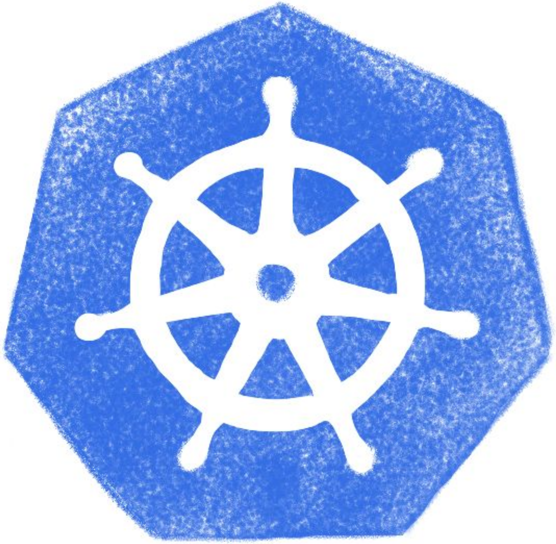
Are you sure you want to operate them by hand?

Taming microservices with Kubernetes

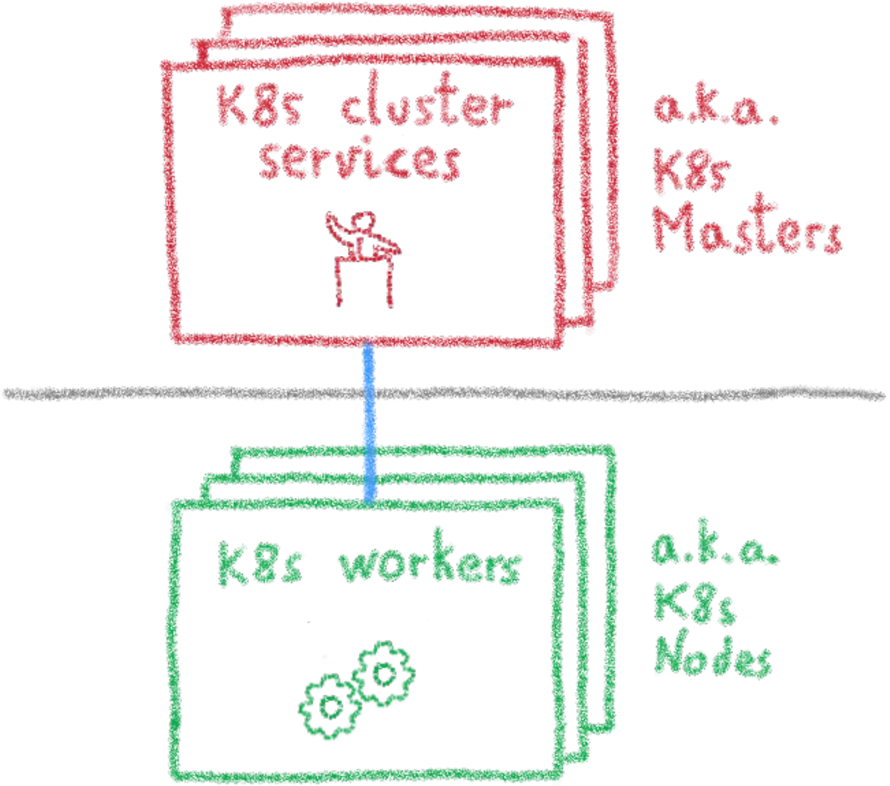


Kubernetes

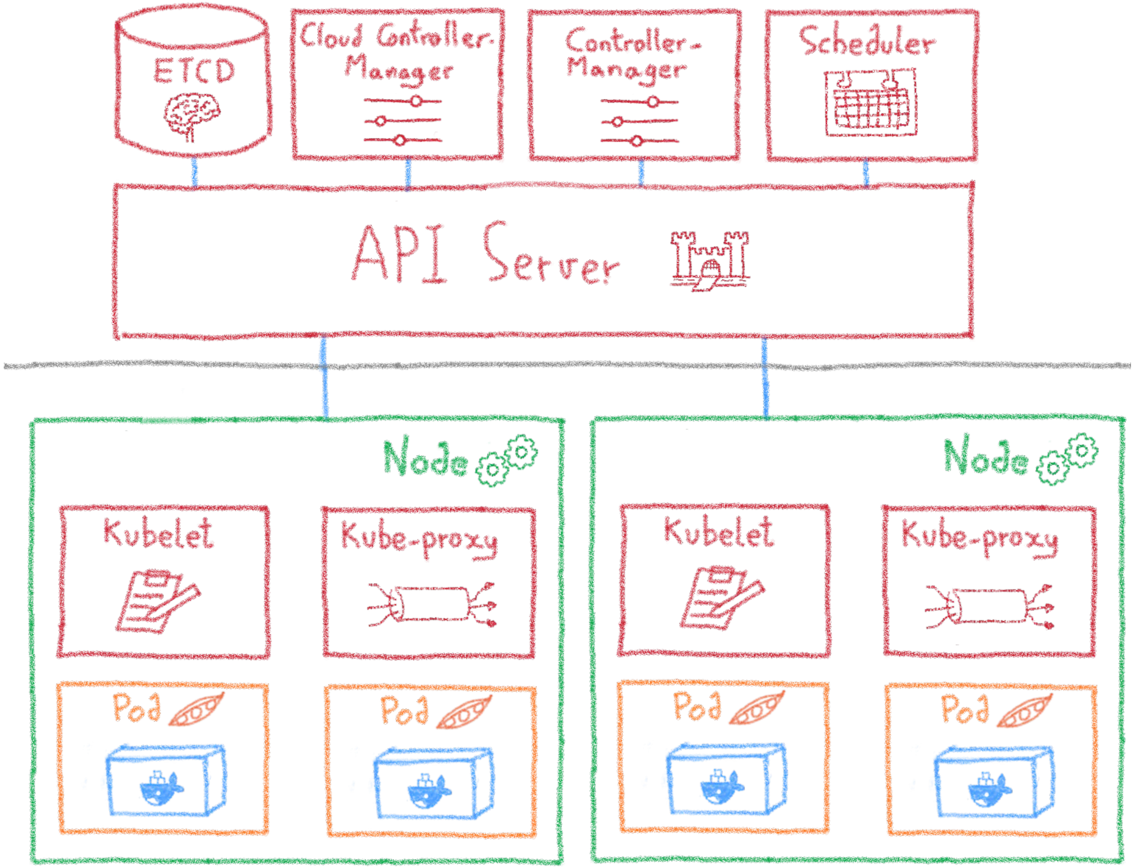
Way more than a buzzword!



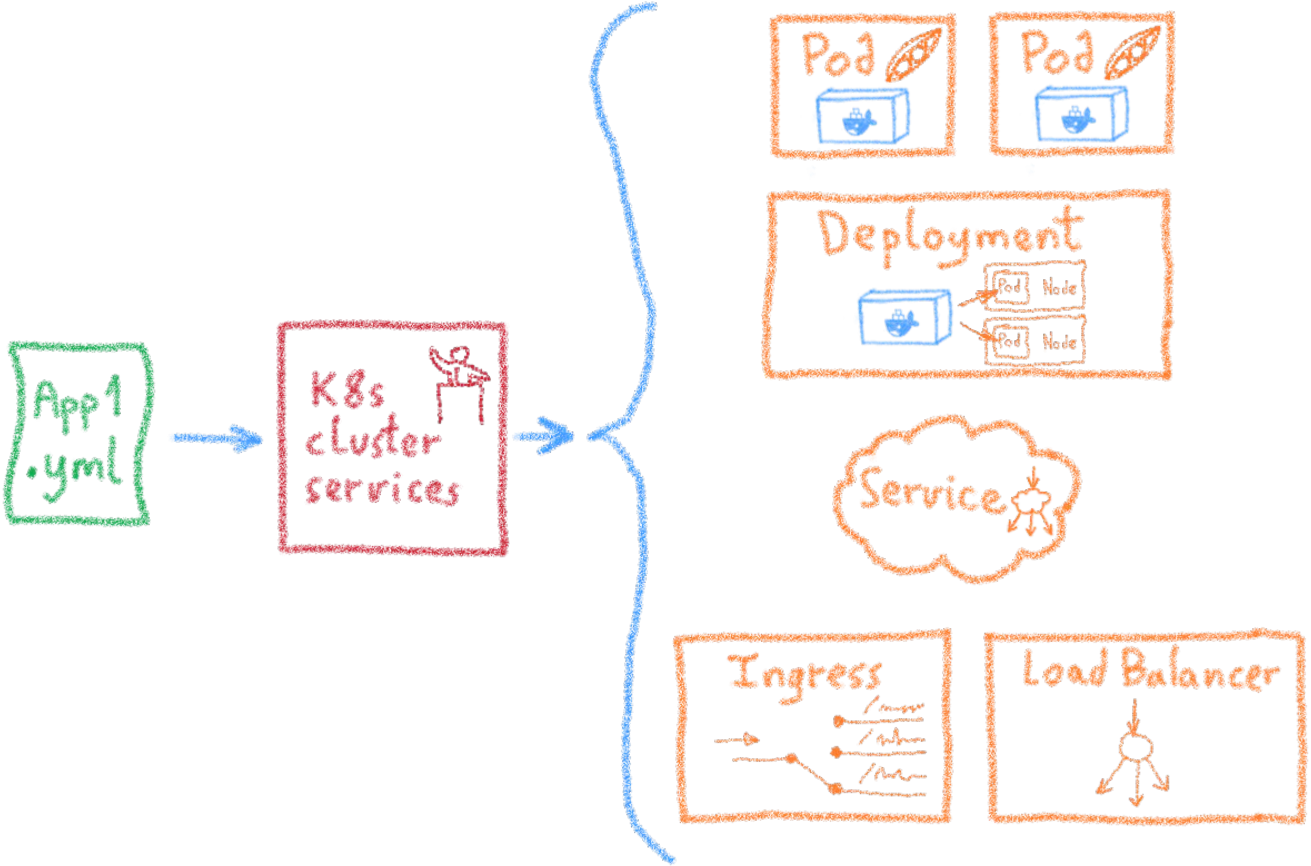
Masters and nodes



Some more details



Desired State Management



Ingress

Services

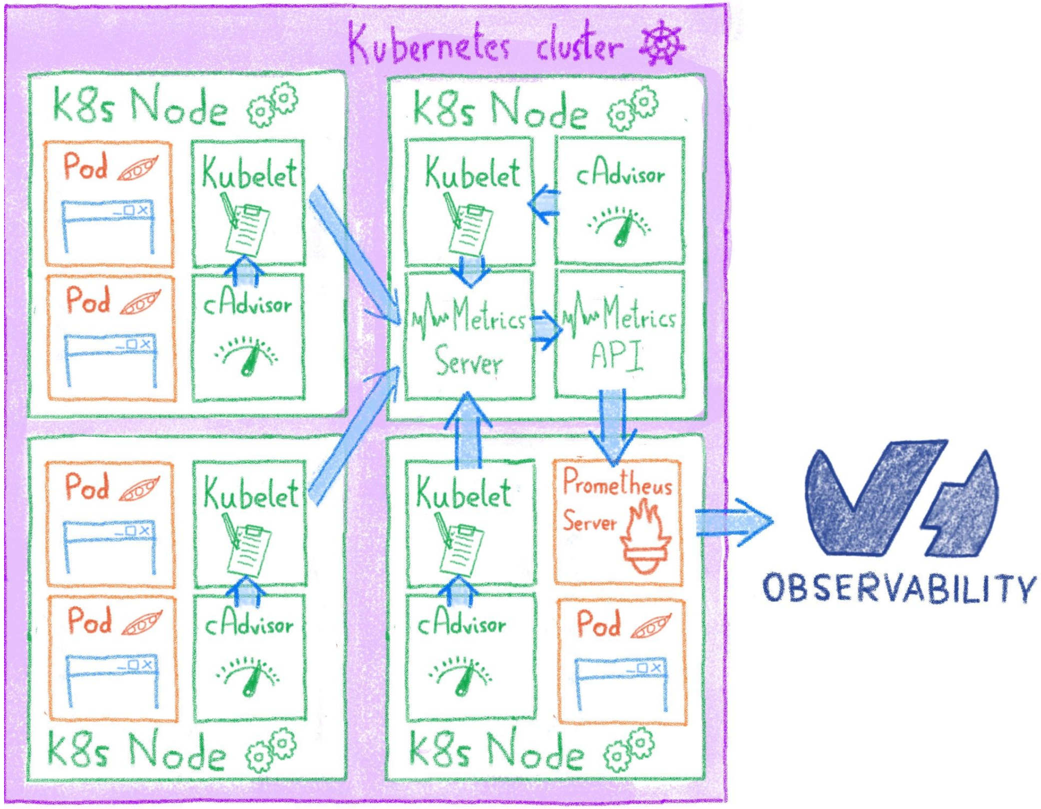
Deployments

Pods

Sidecars

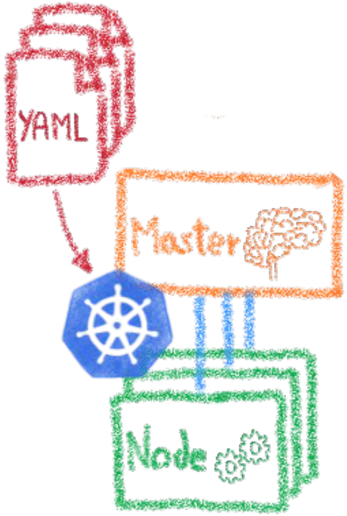
Replica Sets

Extending Kubernetes

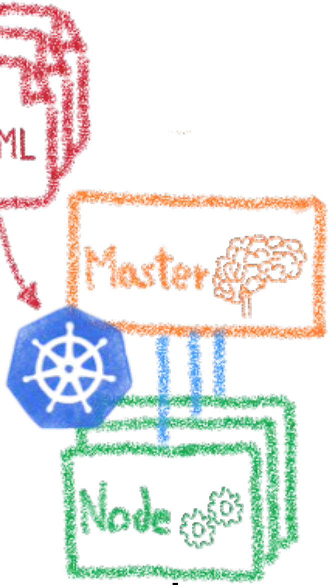


Multi-environment made easy

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

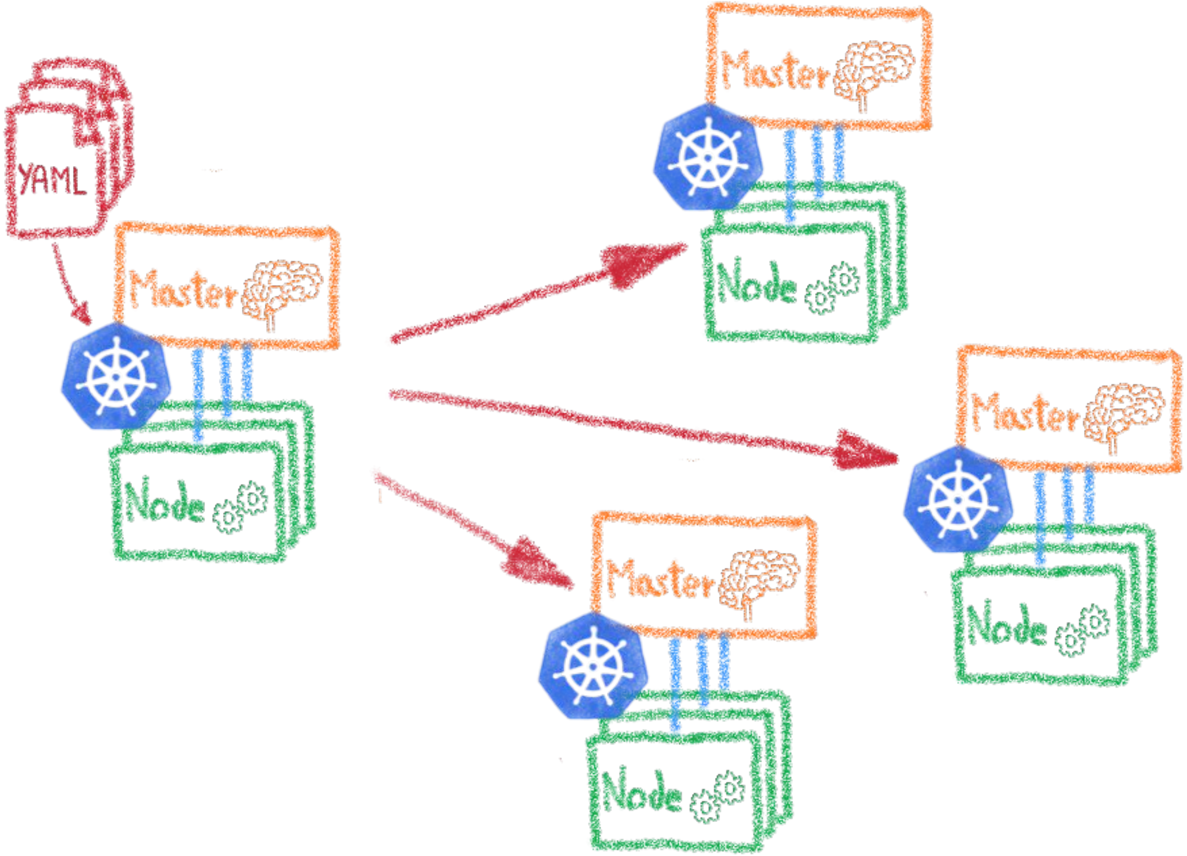


Declarative infrastructure



Multi-environment made easy

Having identical, software defined envs



Dev envs

Staging

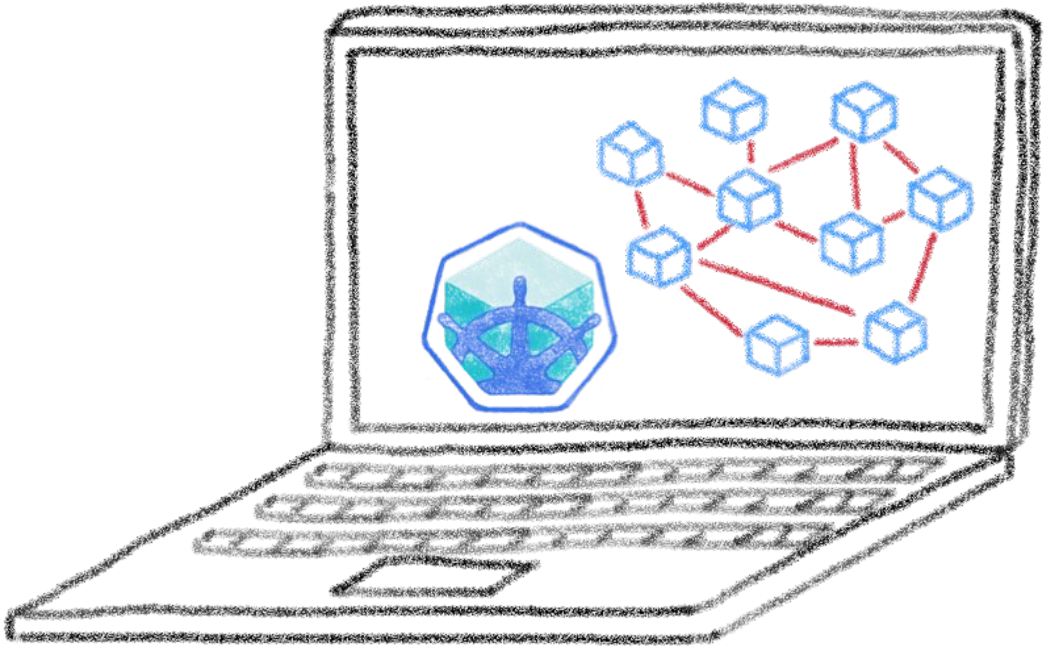
Multi-cluster

Multi-cloud

I have deployed on Minikube, woah!

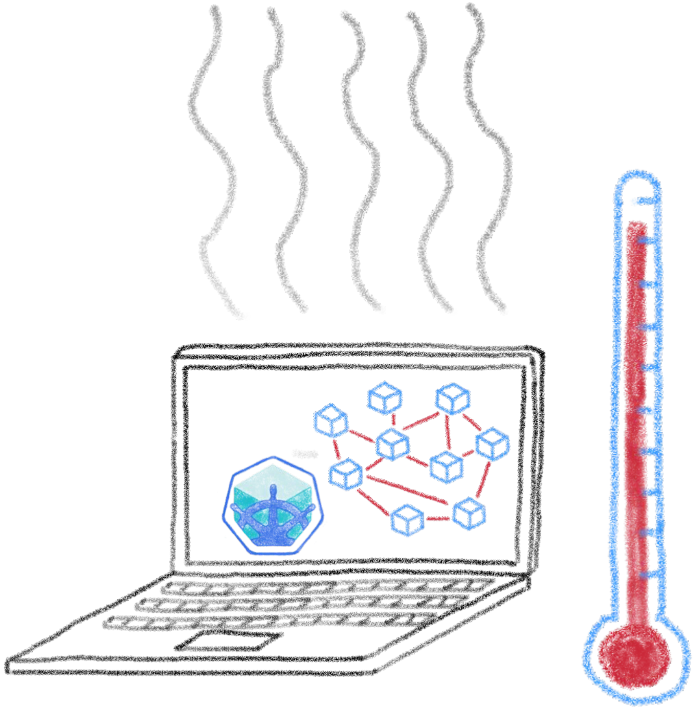
A great fastlane into Kubernetes

Running a full K8s in your laptop



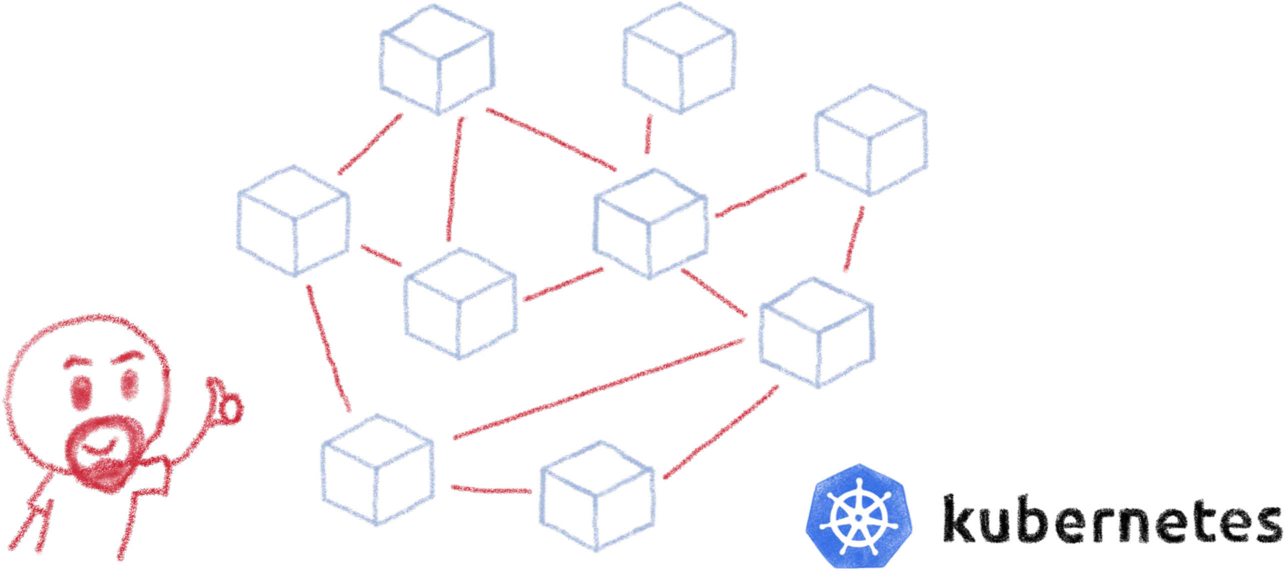
A great learning tool

Your laptop isn't a true cluster



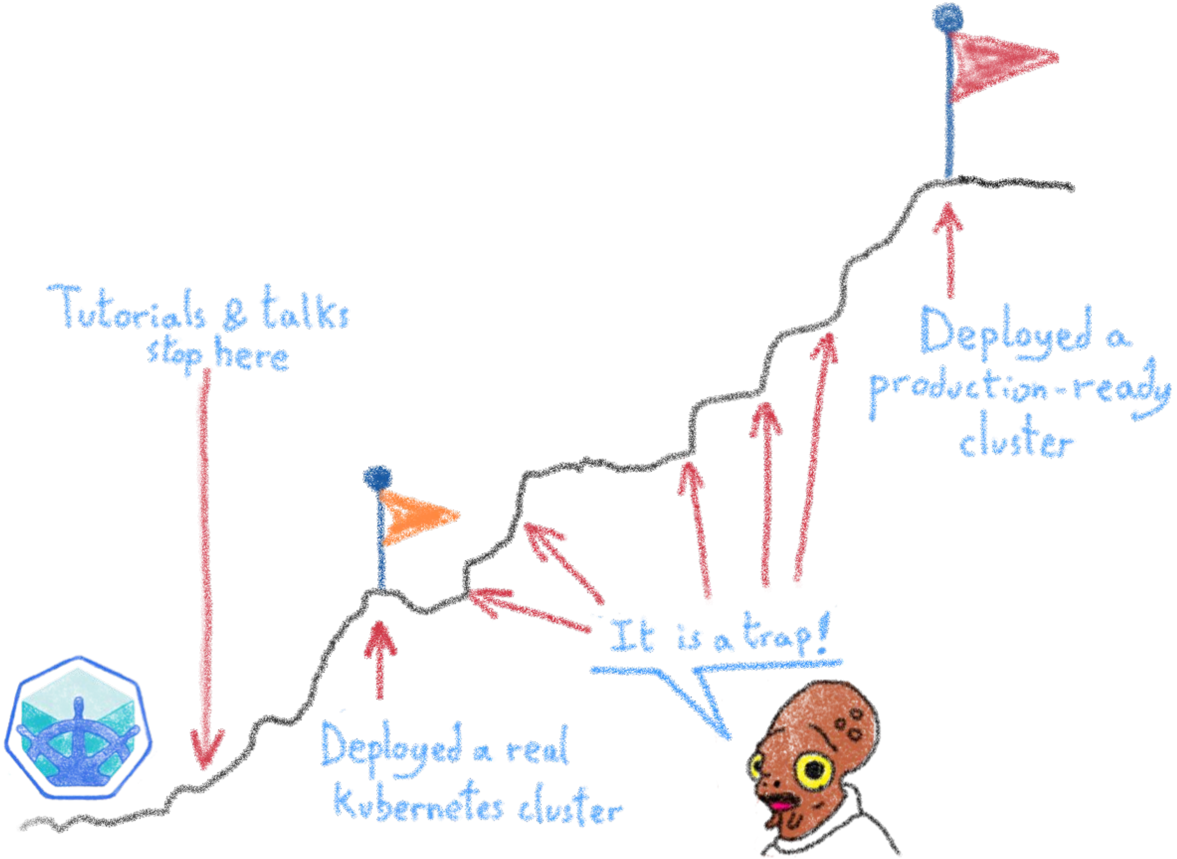
Don't expect real performances

Beyond the first deployment



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

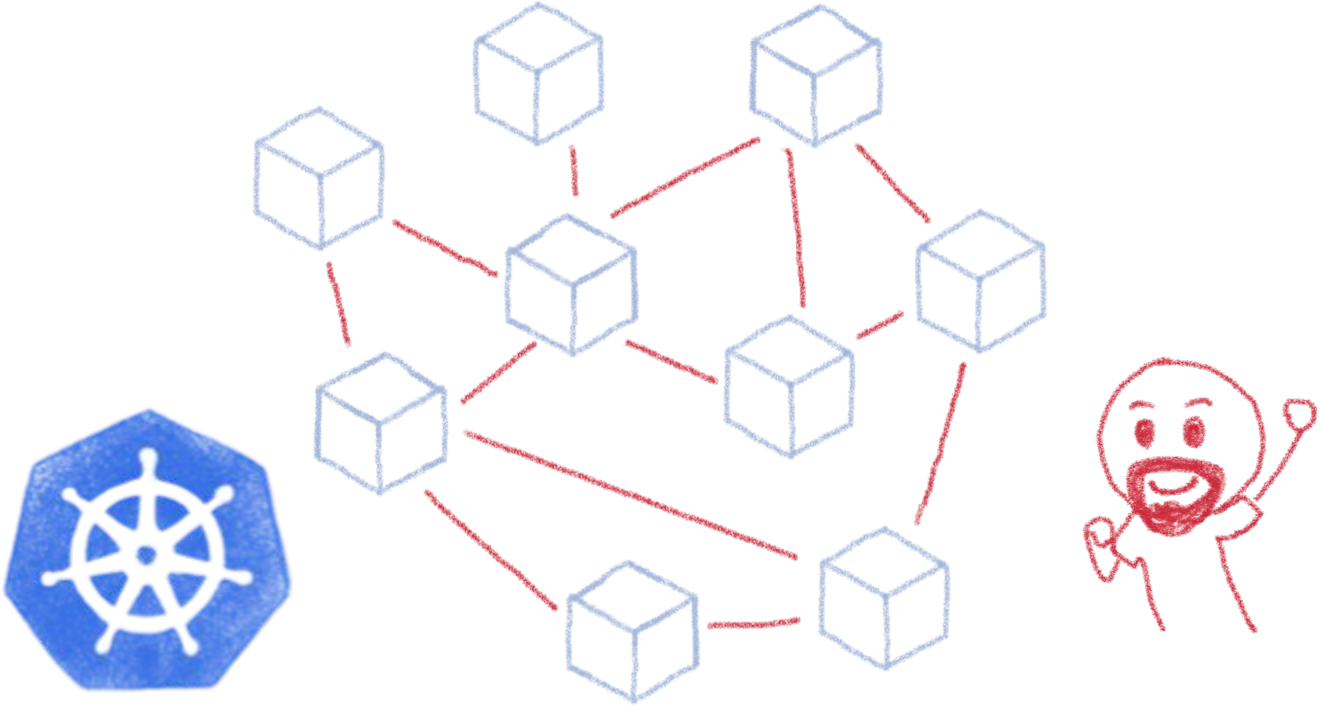
Minikube is only the beginning


Tutorials & talks stop here
Deployed a real Kubernetes cluster
It is a trap!
Deployed a production-ready cluster

From Minikube to prod

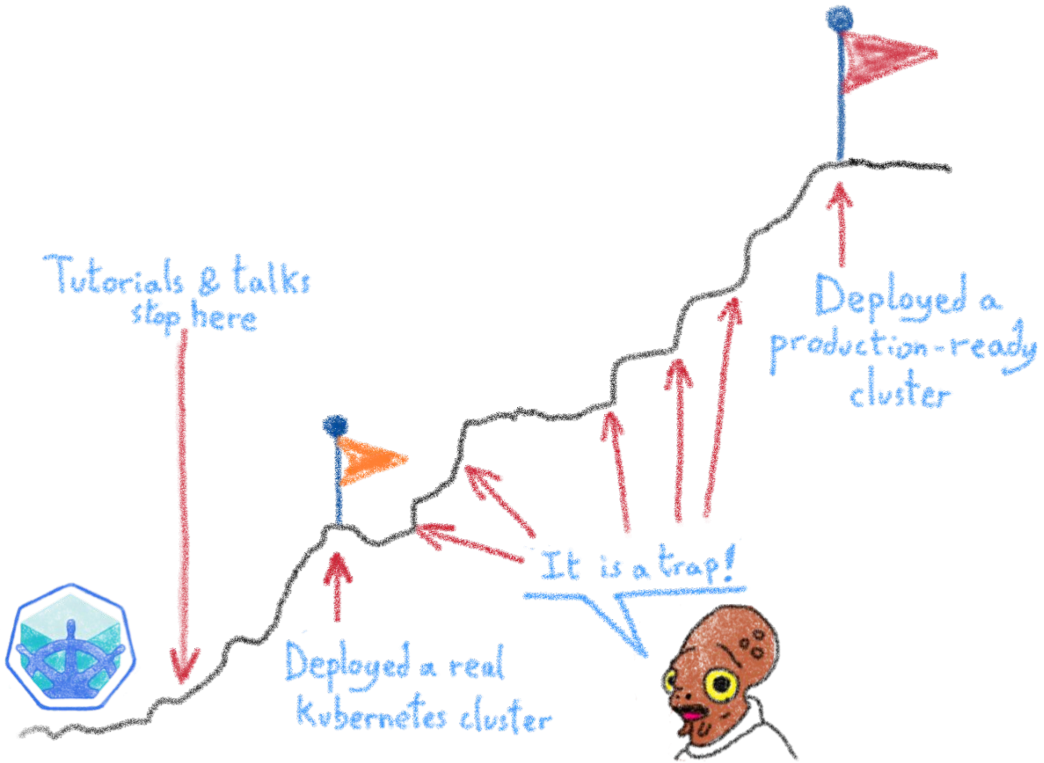
A journey not for the faint of heart

Kubernetes can be wonderful



For both developers and devops

But it comes with a price...

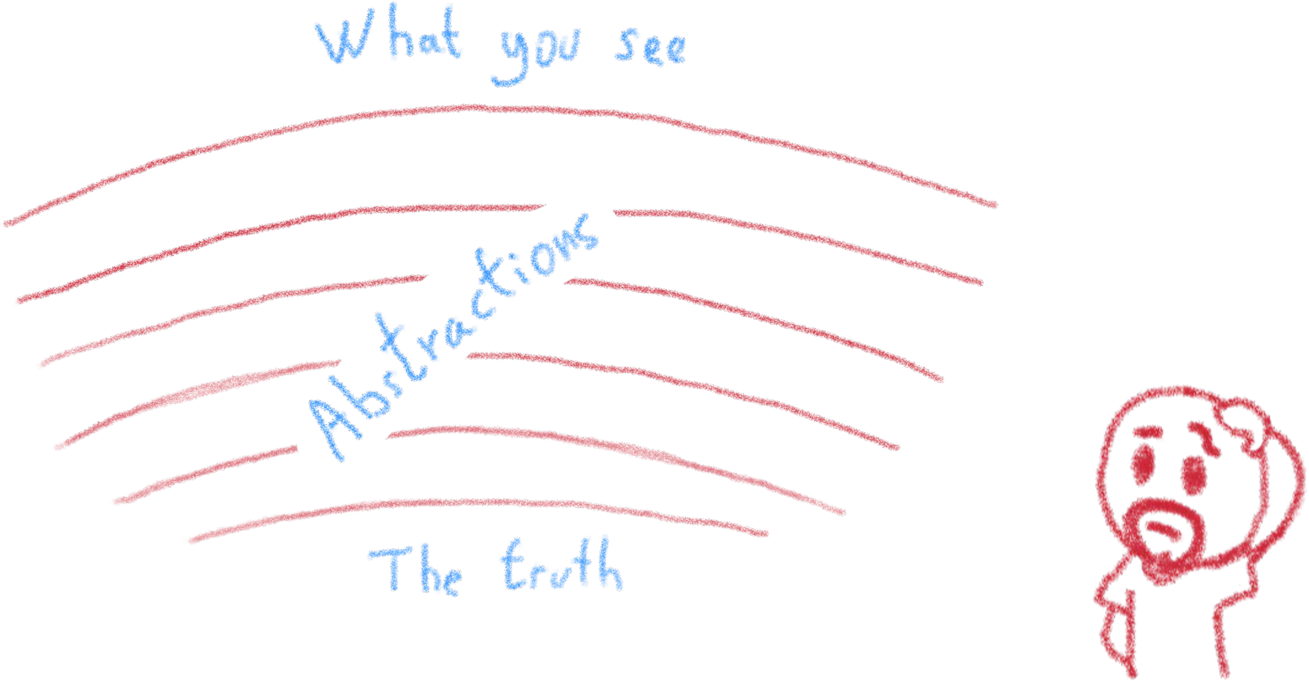

Tutorials & talks stop here
Deployed a real Kubernetes cluster
It is a trap!
Deployed a production-ready cluster

Describing some of those traps

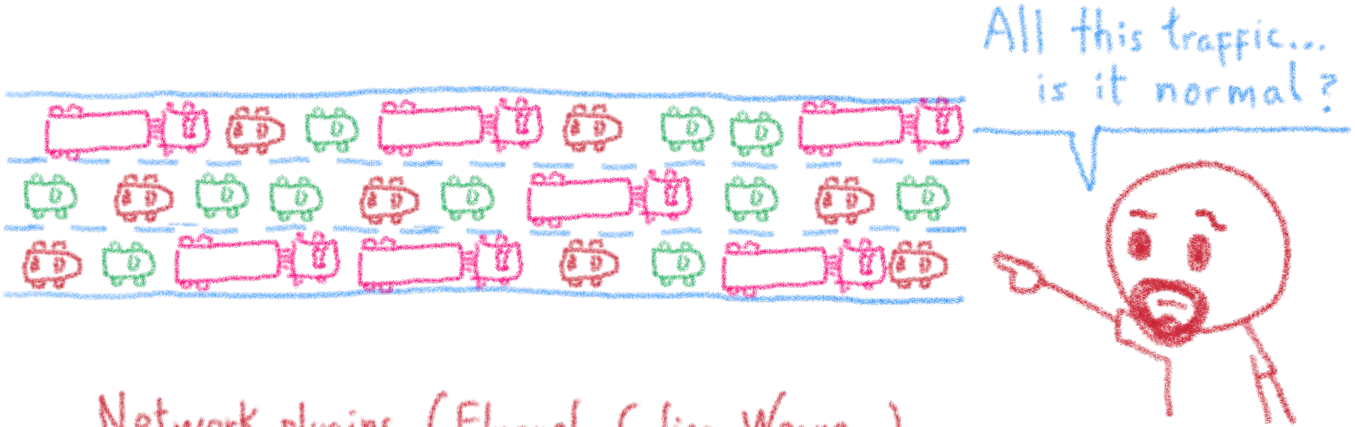
Security
Deployment
Monitoring
Backups

To ease and empower your path to production

The truth is somewhere inside...



The network is going to feel it...



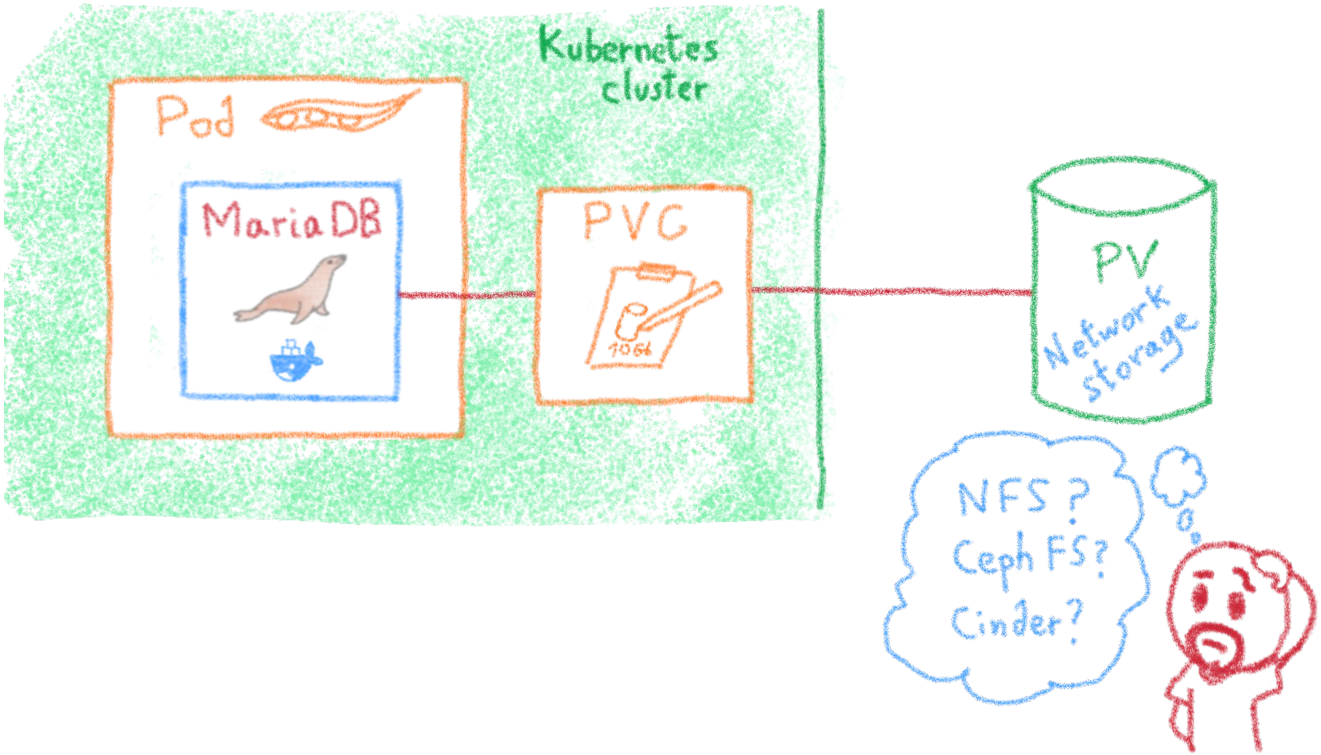
Network plugins (Flannel, Calico, Weave...)

- IPAM
- routing
- iptables
- crossnode networking

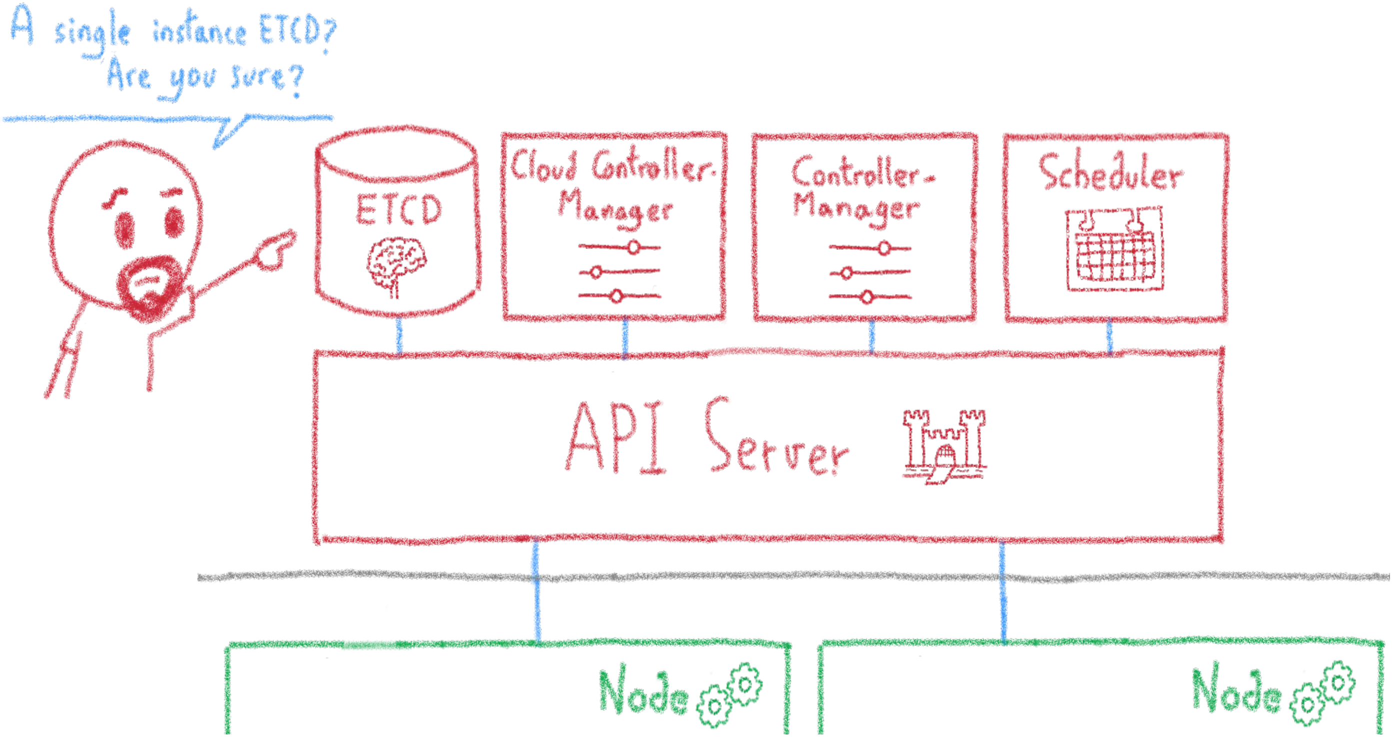
ClusterIP, NodePort, Ingress

Service Meshes, Istio

The storage dilemma

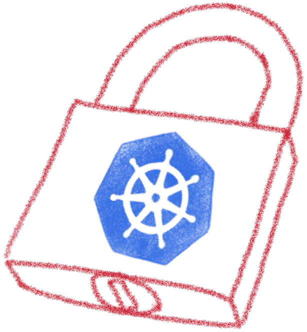


The ETCD vulnerability

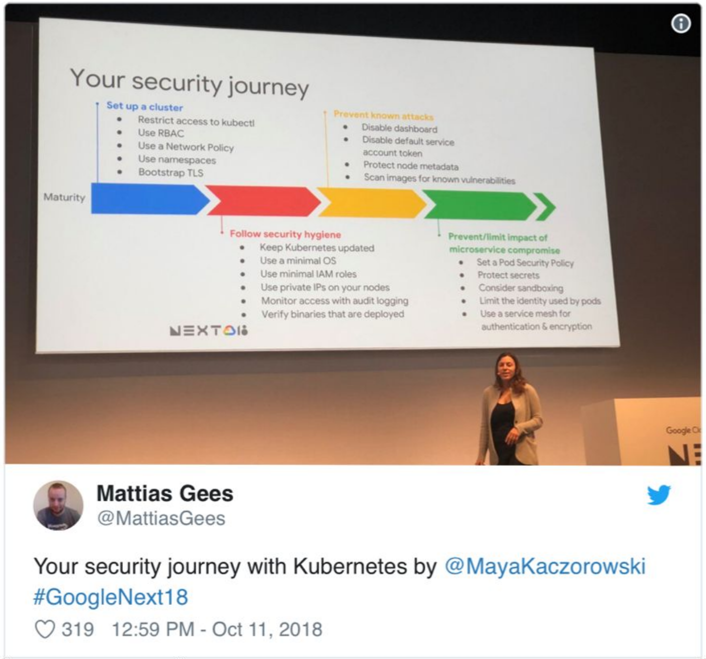


Security

Hardening your Kubernetes



The security journey



Your security journey

Set up a cluster
- Restrict access to kubectl
- Use RBAC
- Use a Network Policy
- Use namespaces
- Bootstrap TLS

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 known attacks
- Disable dashboard
- Disable default service account token
- Protect node metadata
- Scan images for known vulnerabilities

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

Maturity

NEXT 18

Mattias Gees
@MattiasGees

Your security journey with Kubernetes by @MayaKaczorowski #GoogleNext18

♡ 319 12:59 PM - Oct 11, 2018

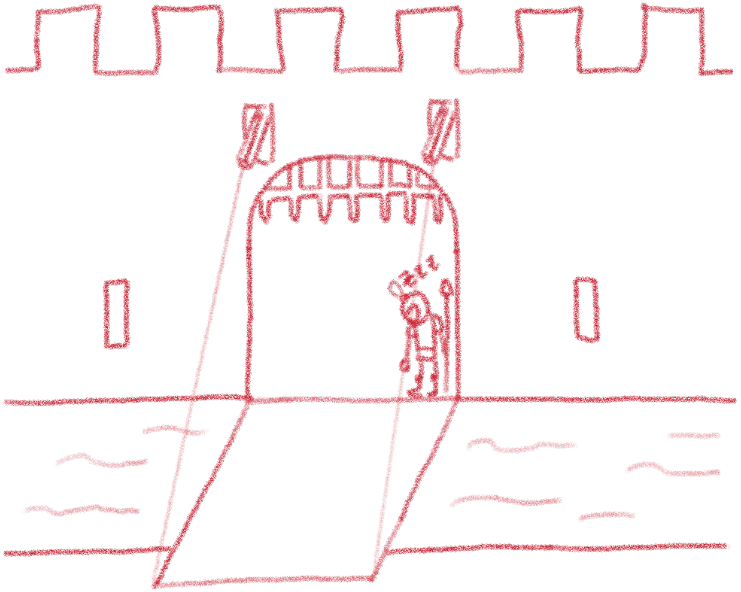
Open ports (e.g. etcd 2379/TCP)

Kubernetes API (e.g. Tesla hacking)

Exploits (lots of CVEs)

RBAC (e.g. badly defined roles)

Kubernetes is insecure by design

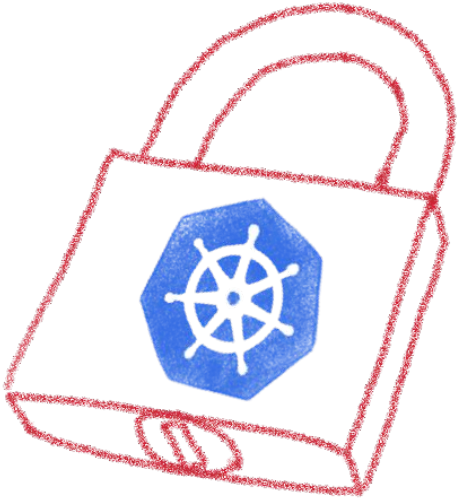


It's a feature, not a bug.

Up to K8s admin to secure it according to needs

Not everybody has the same security needs

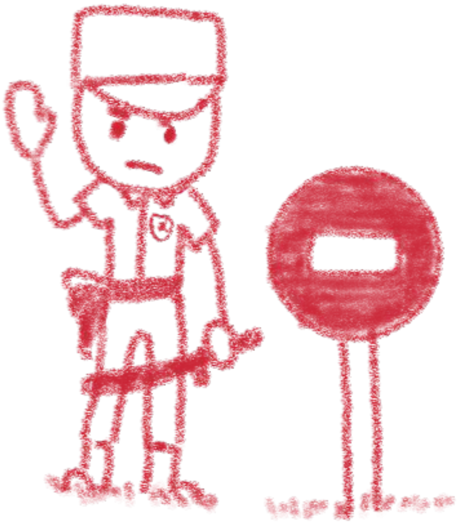
Kubernetes allows to enforce security practices as needed



Listing some good practices

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

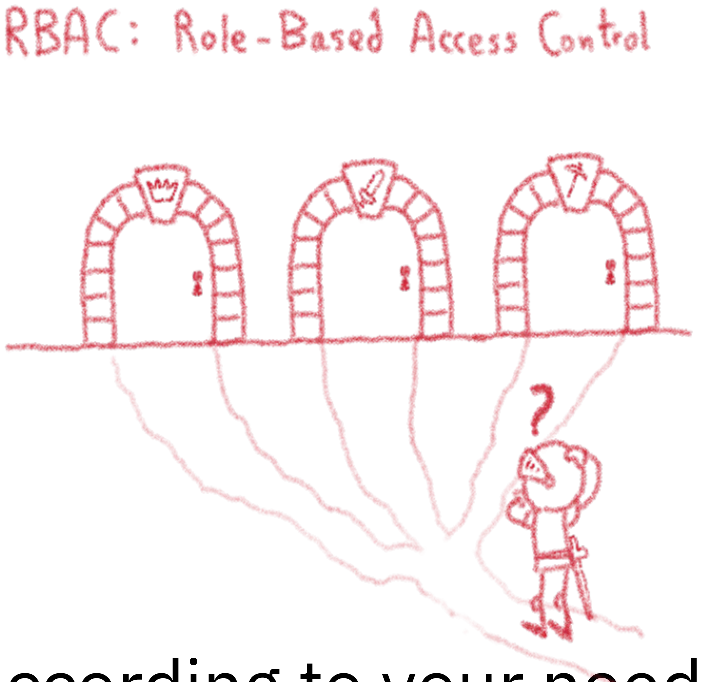
Close open access



Close all by default, open only the needed ports

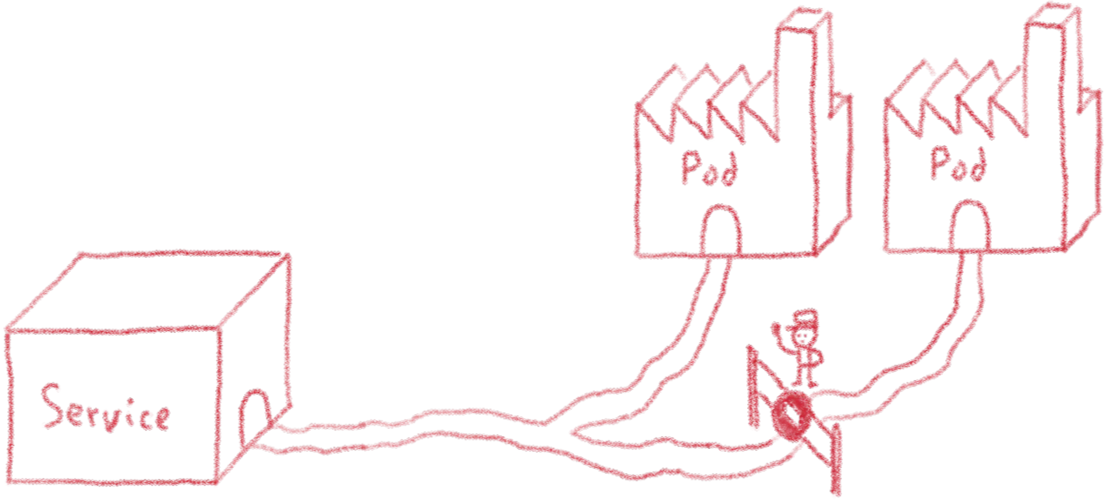
Follow the least privileged principle

Define and implement RBAC

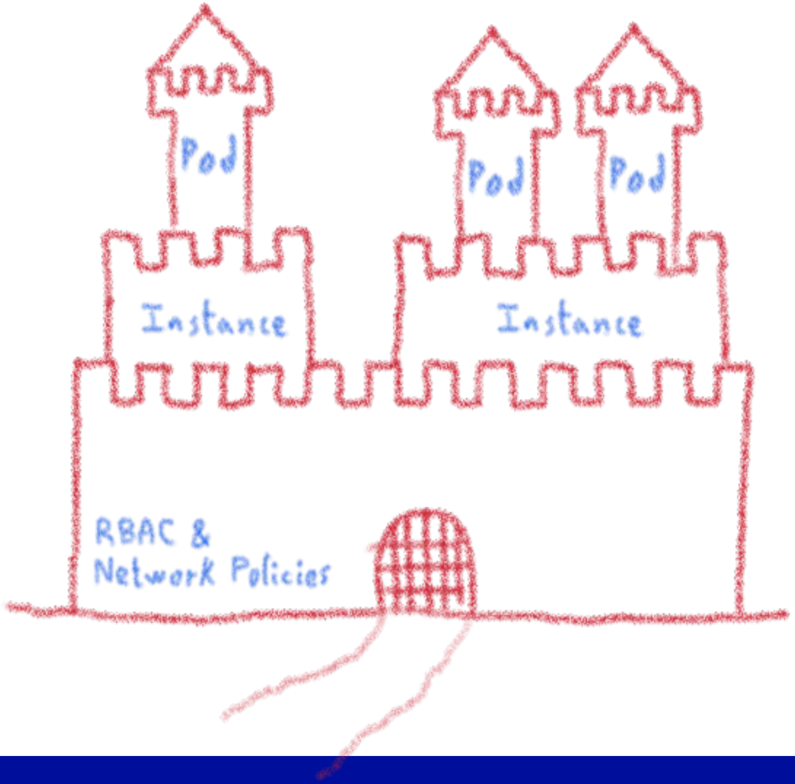


According to your needs

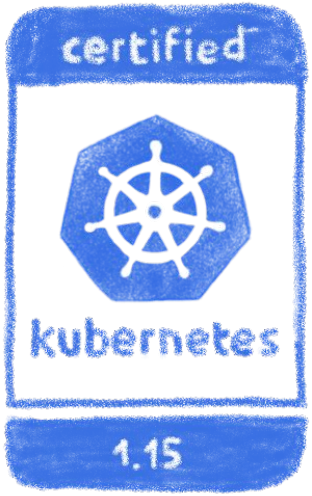
Define and implement network policies



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

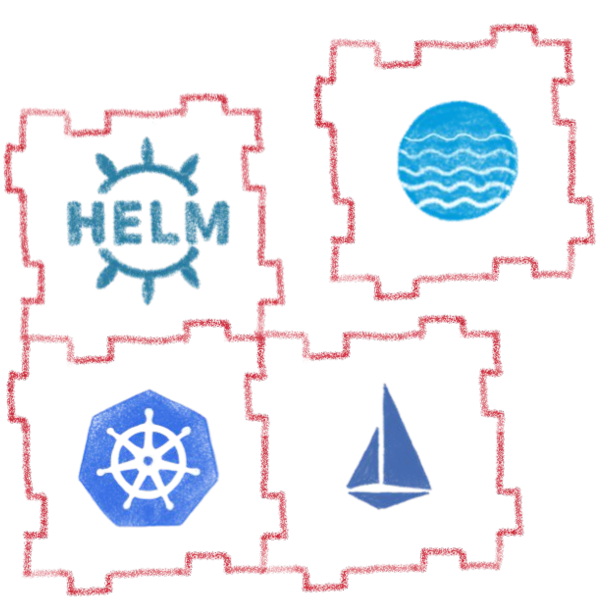
One of Tesla's cluster got hacked via an unprotected K8s API endpoint, and was used to mine cryptocurrency...

Remain attentive, don't get too confident

Extensibility

Enhance your Kubernetes

Kubernetes is modular



Fully extensible

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

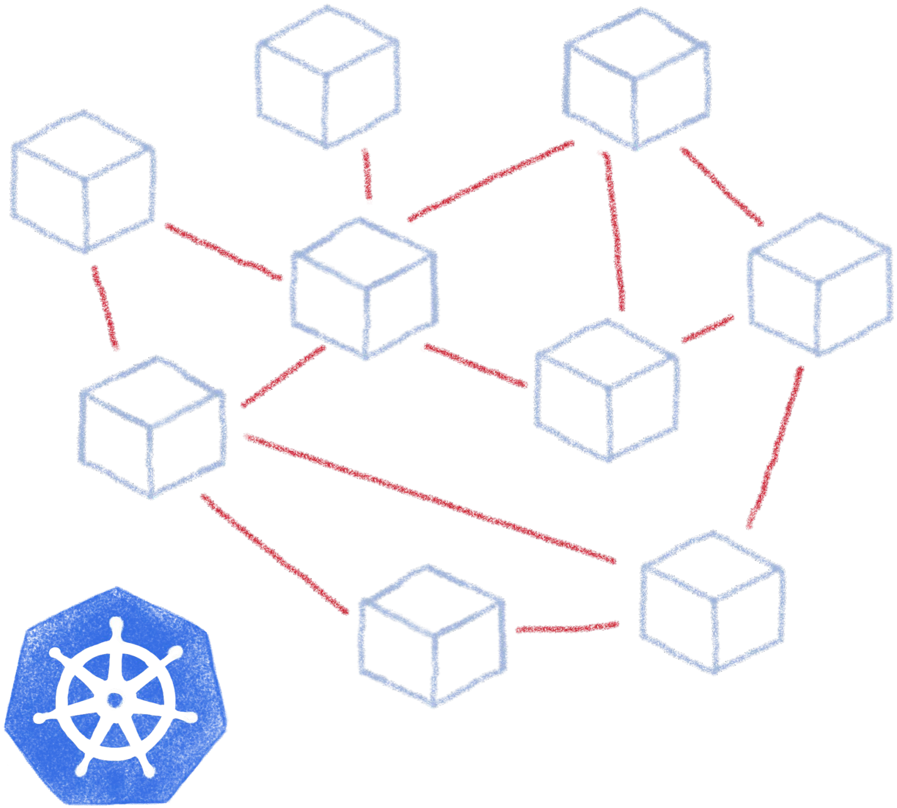
Operators

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

Helm

A package management for K8s

Complex deployments



Ingress

Services

Deployments

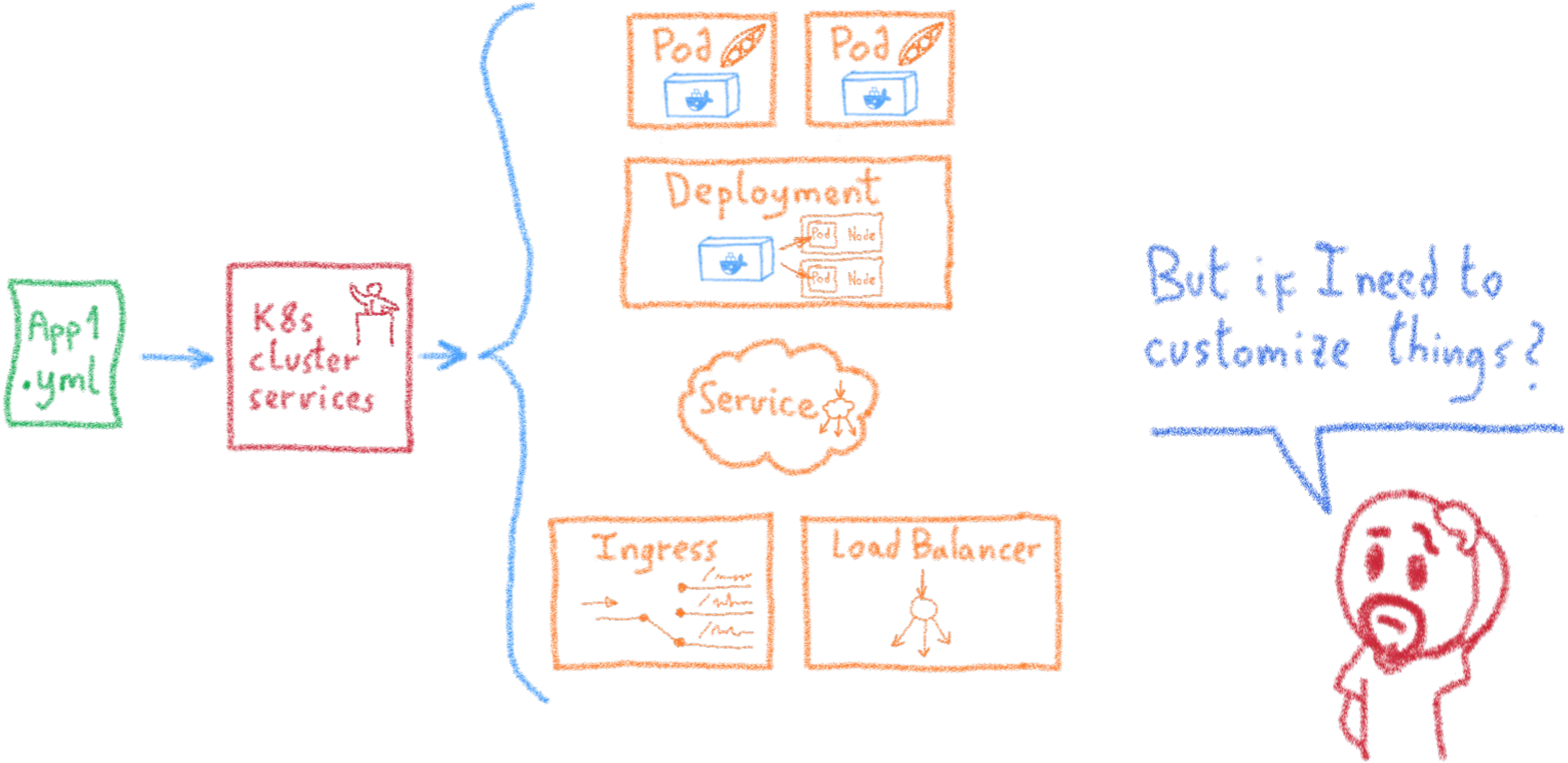
Pods

Sidecars

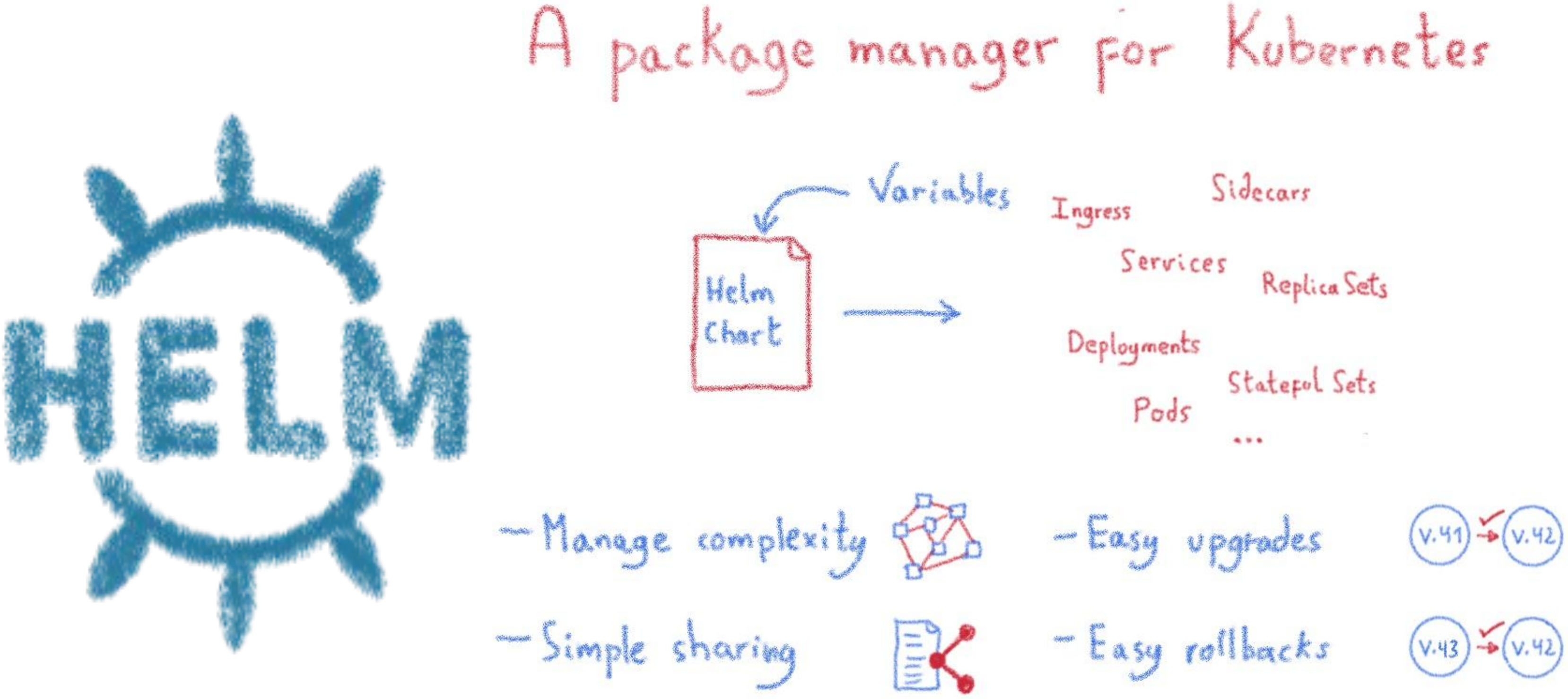
Replica Sets

Stateful Sets

Using static YAML files



Complex deployments

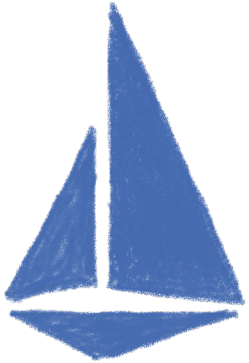


A package manager for Kubernetes

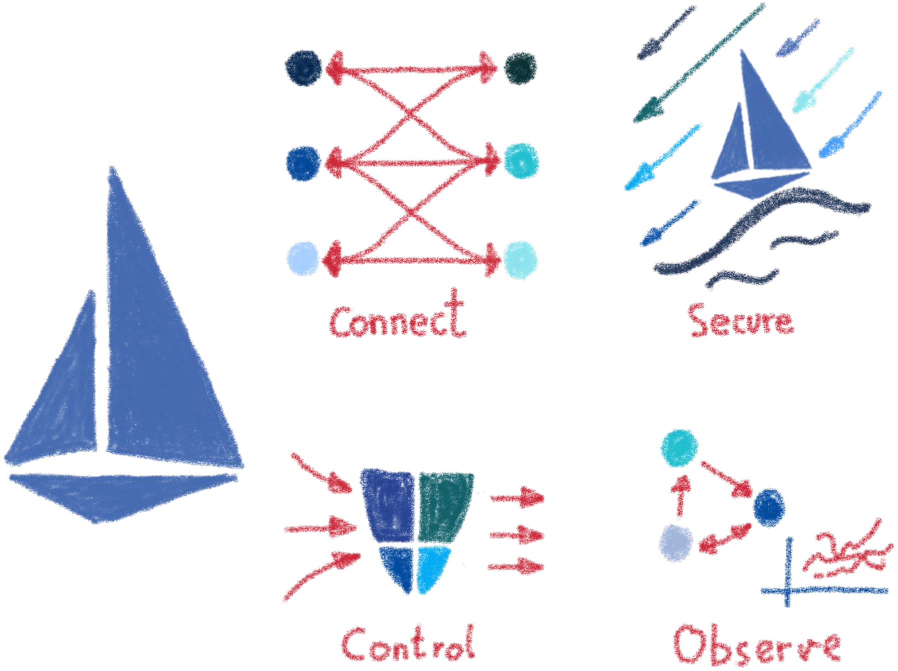
- Manage complexity
- Easy upgrades
- Simple sharing
- Easy rollbacks

Istio

A service mesh for Kubernetes…
and much more!

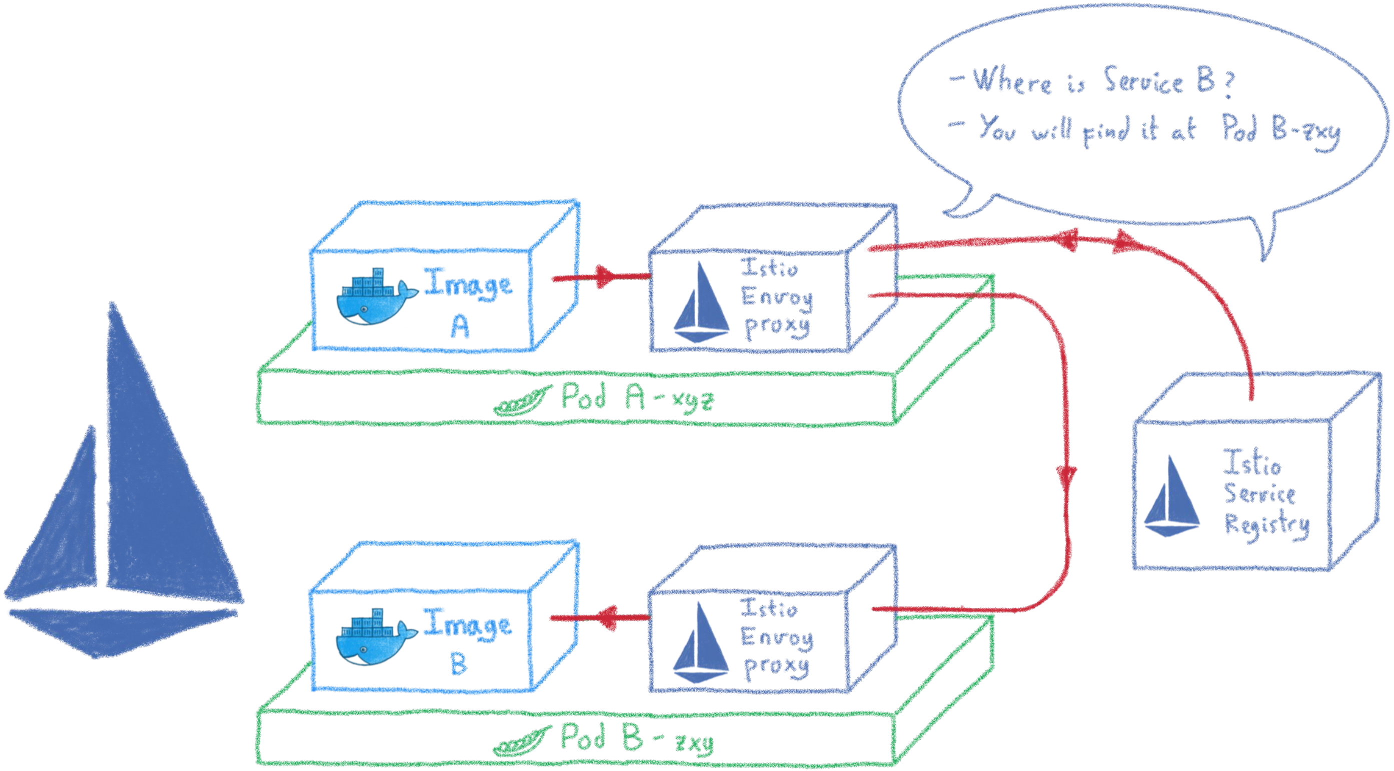


Istio: A service mesh but not only

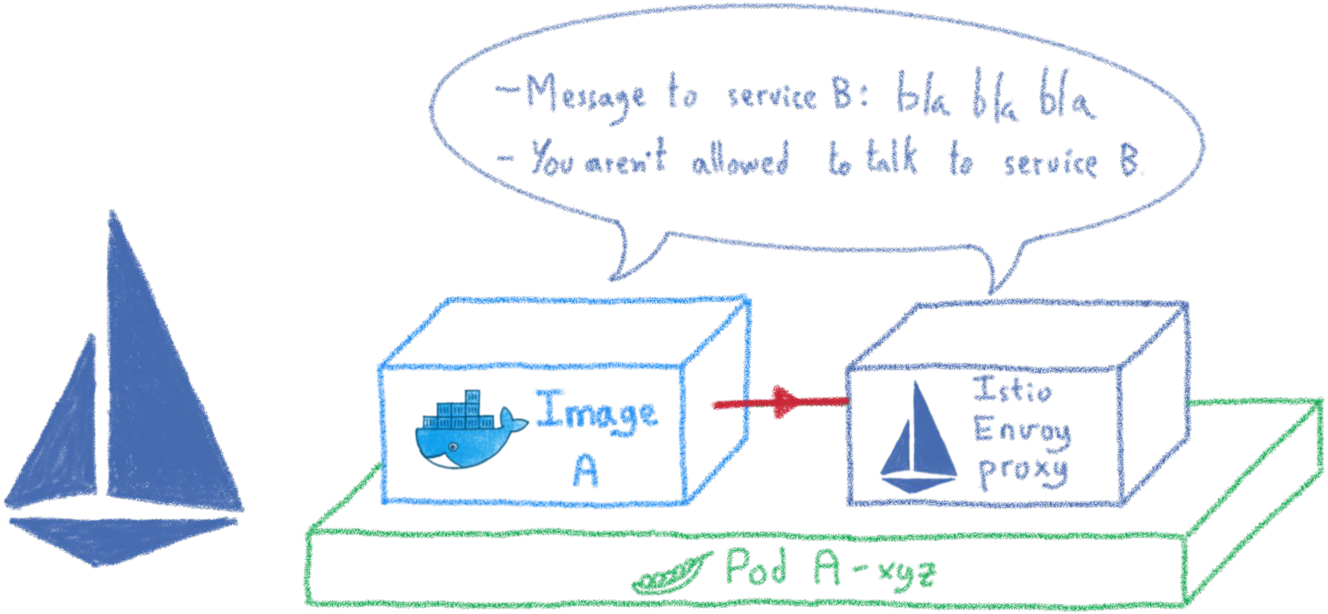


- Rolling upgrades
- A/B Testing
- Canary Testing
- Edge traffic management
- Multicluster service mesh

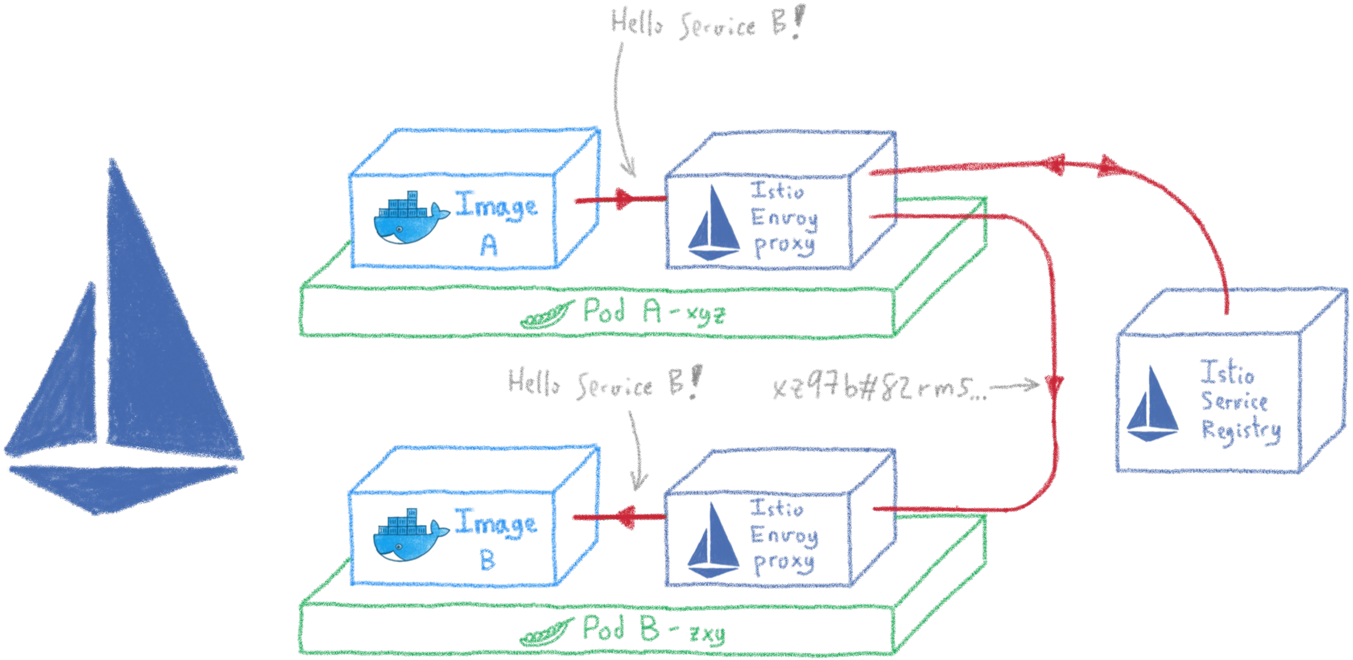
Service discovery



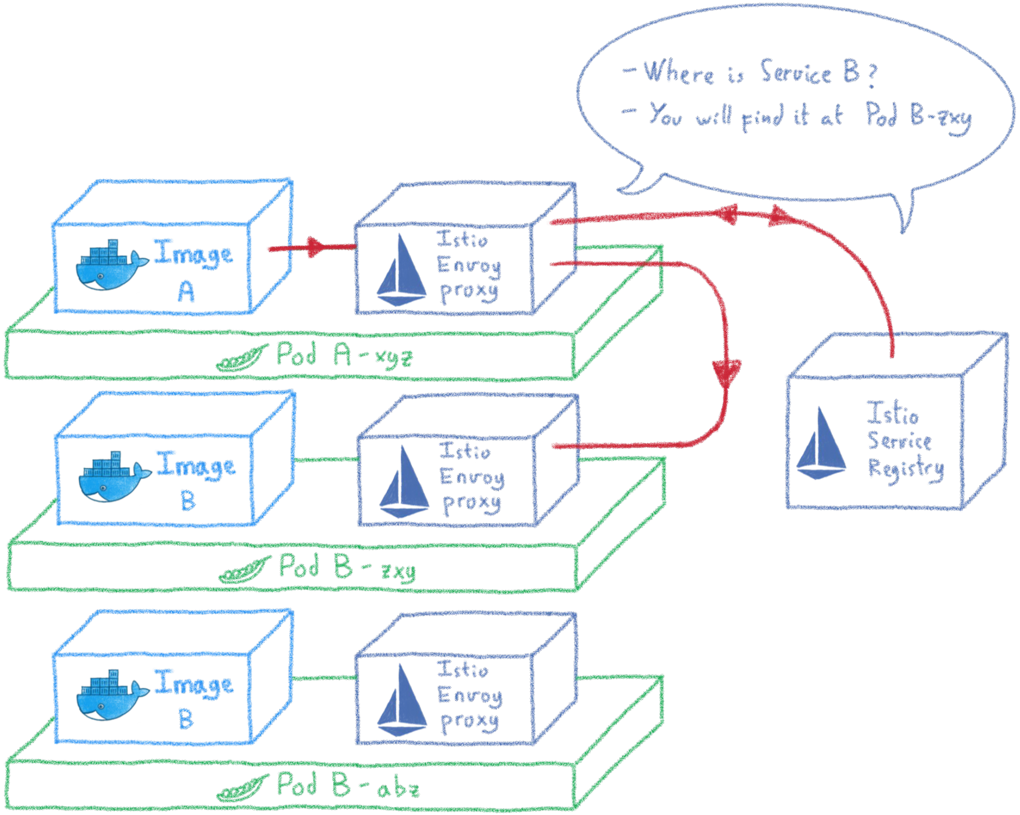
Traffic control


- Message to service B: bla bla bla
- You aren't allowed to talk to service B.
Image A
Istio Envoy proxy
Pod A - xyz

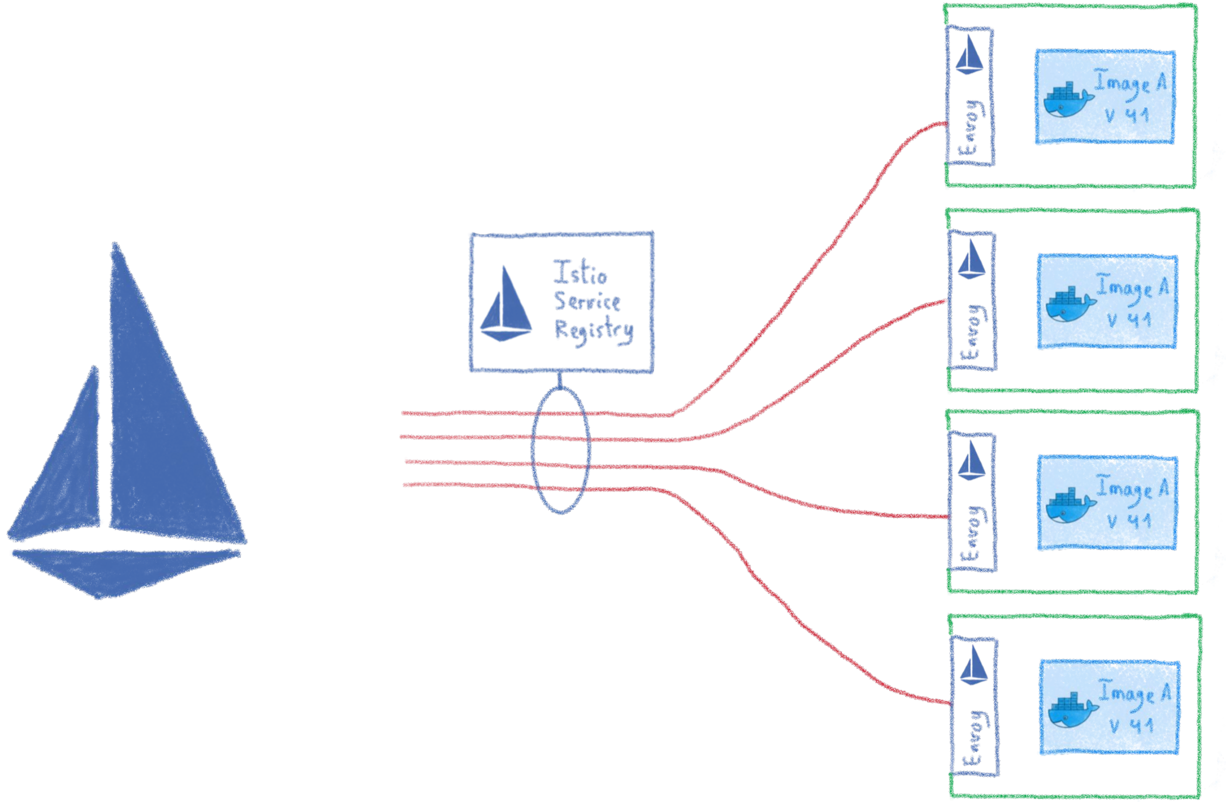
Encrypting internal communications



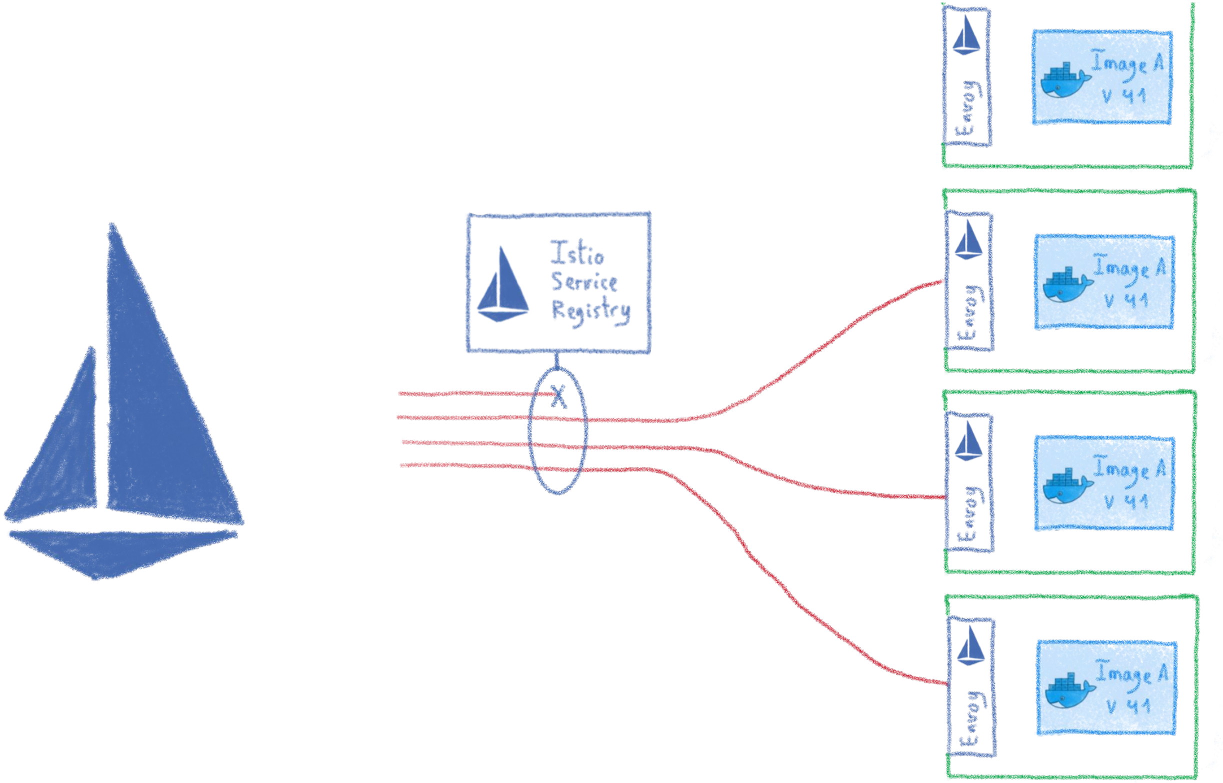
Routing and load balancing



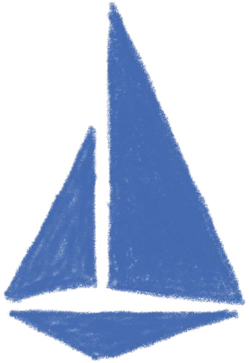
Rolling upgrades

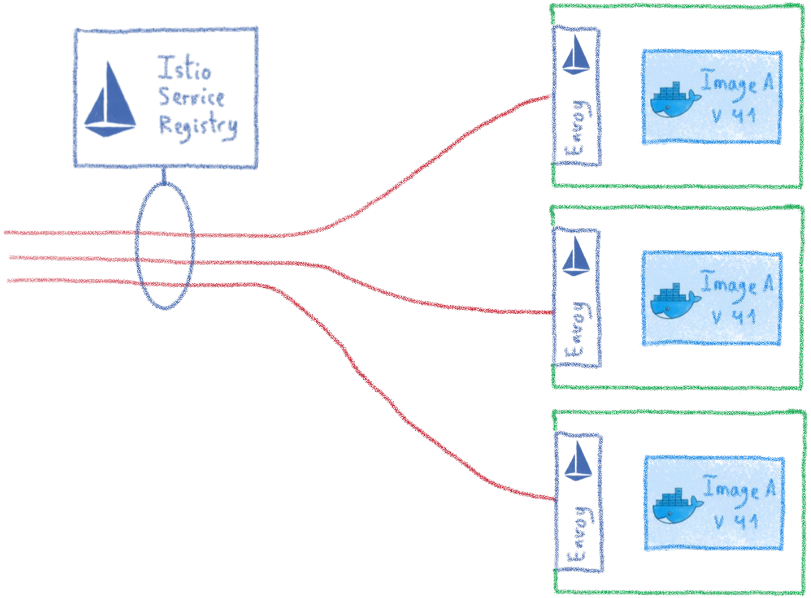


Rolling upgrades

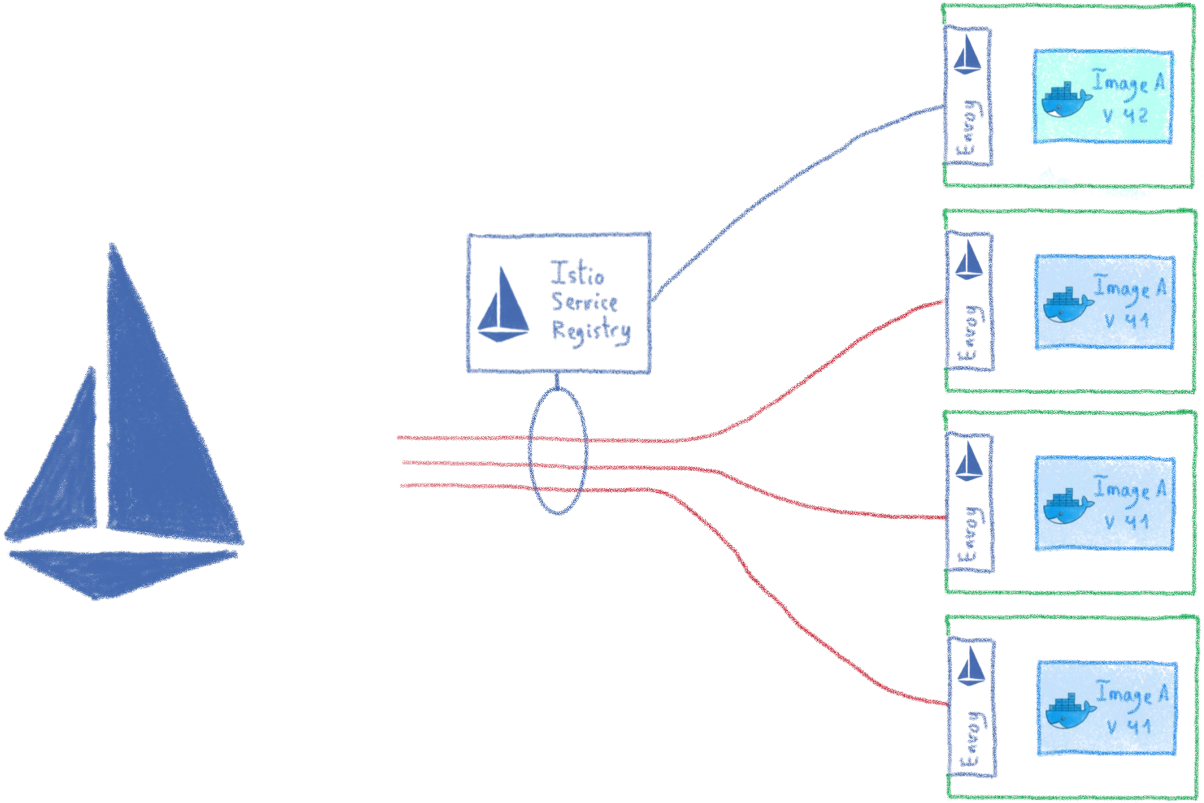


Rolling upgrades

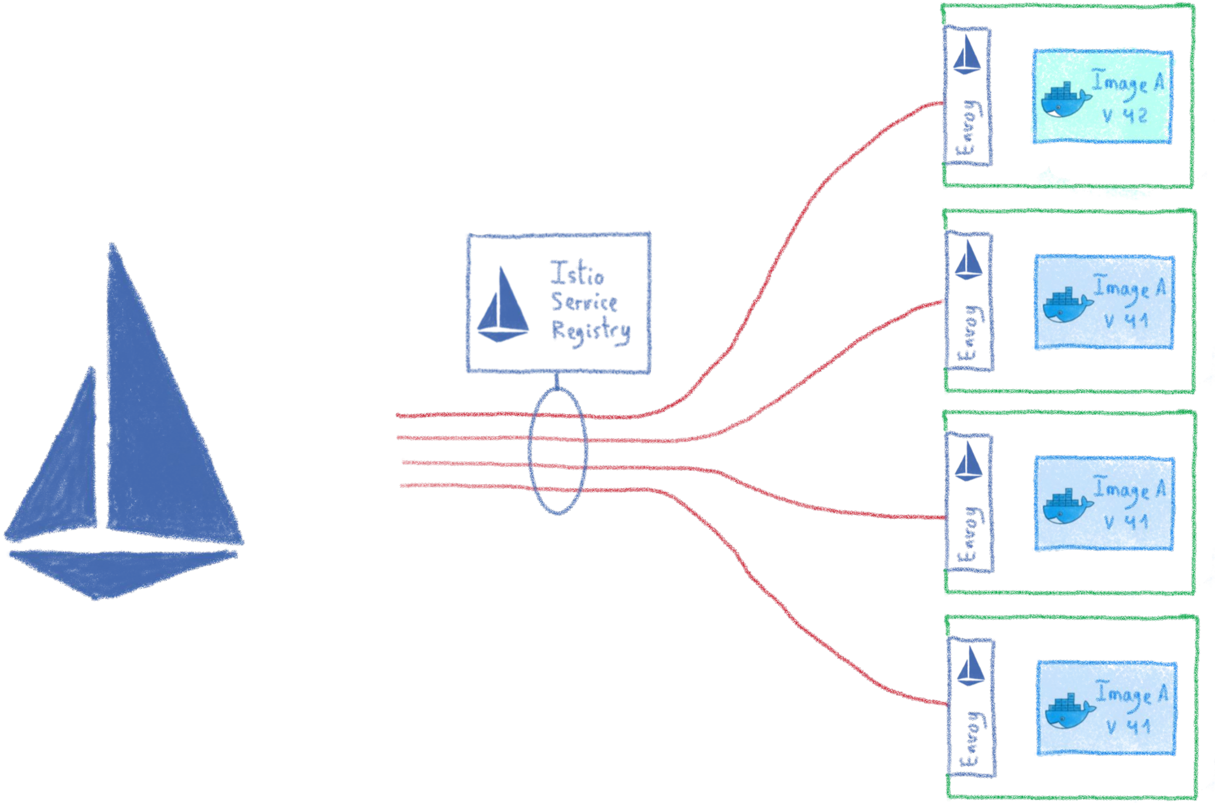




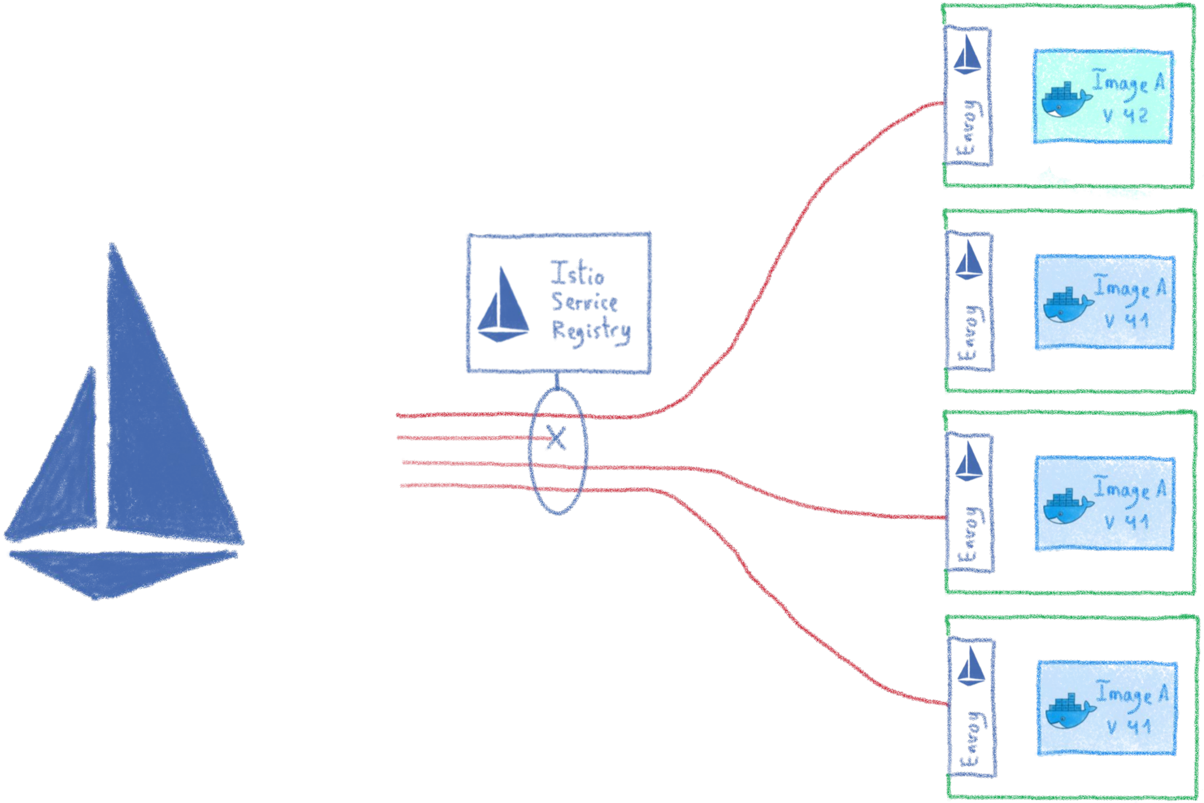
Rolling upgrades



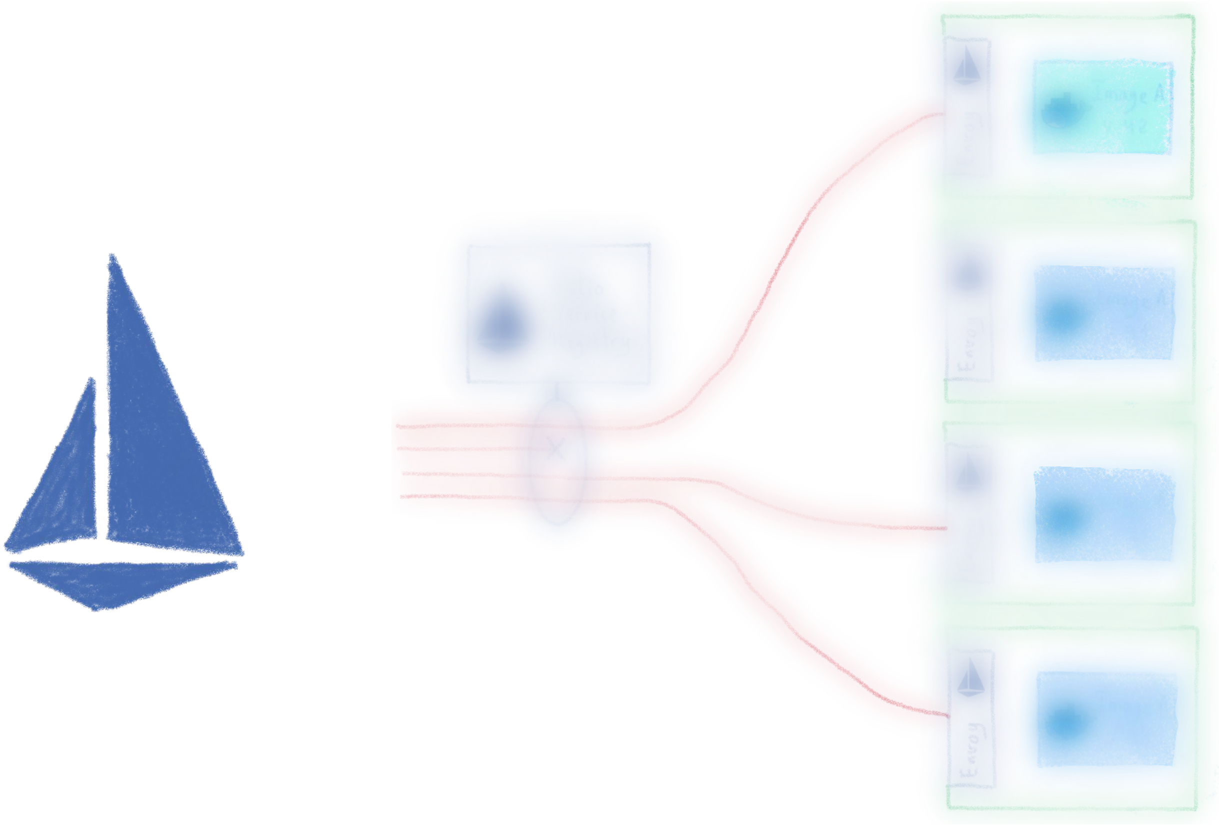
Rolling upgrades



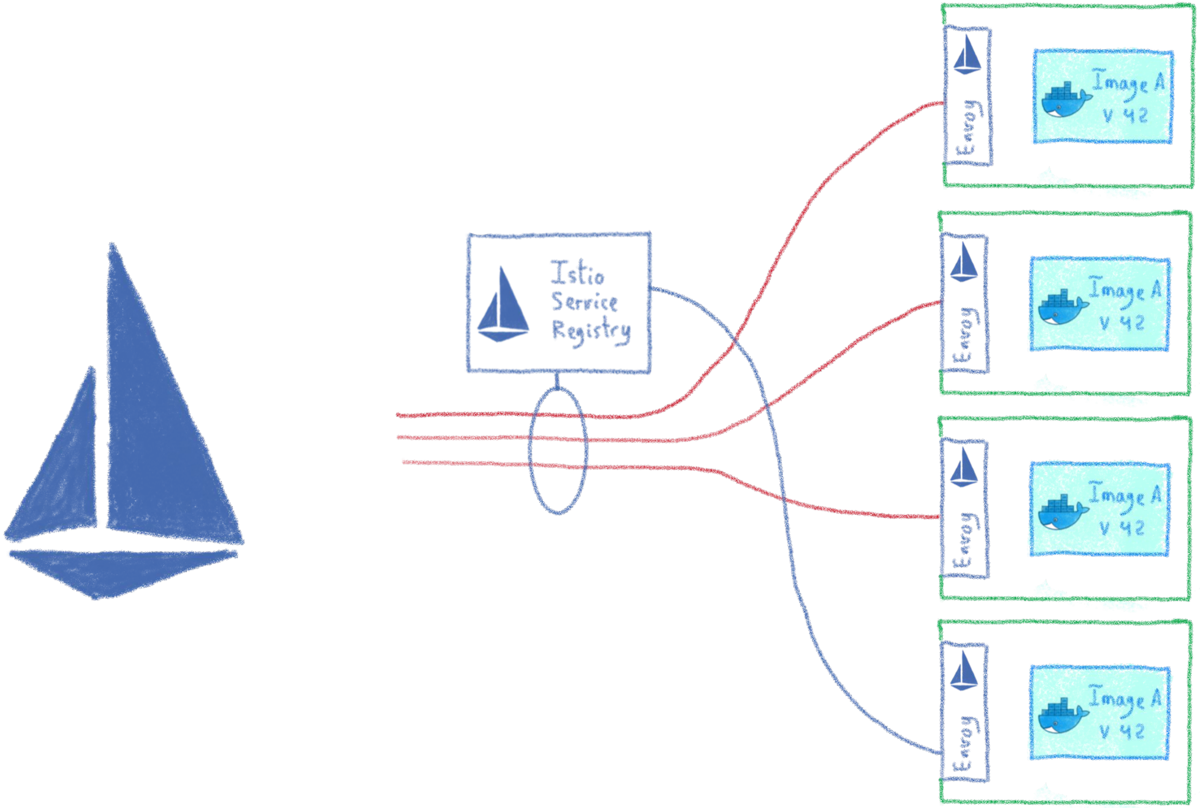
Rolling upgrades



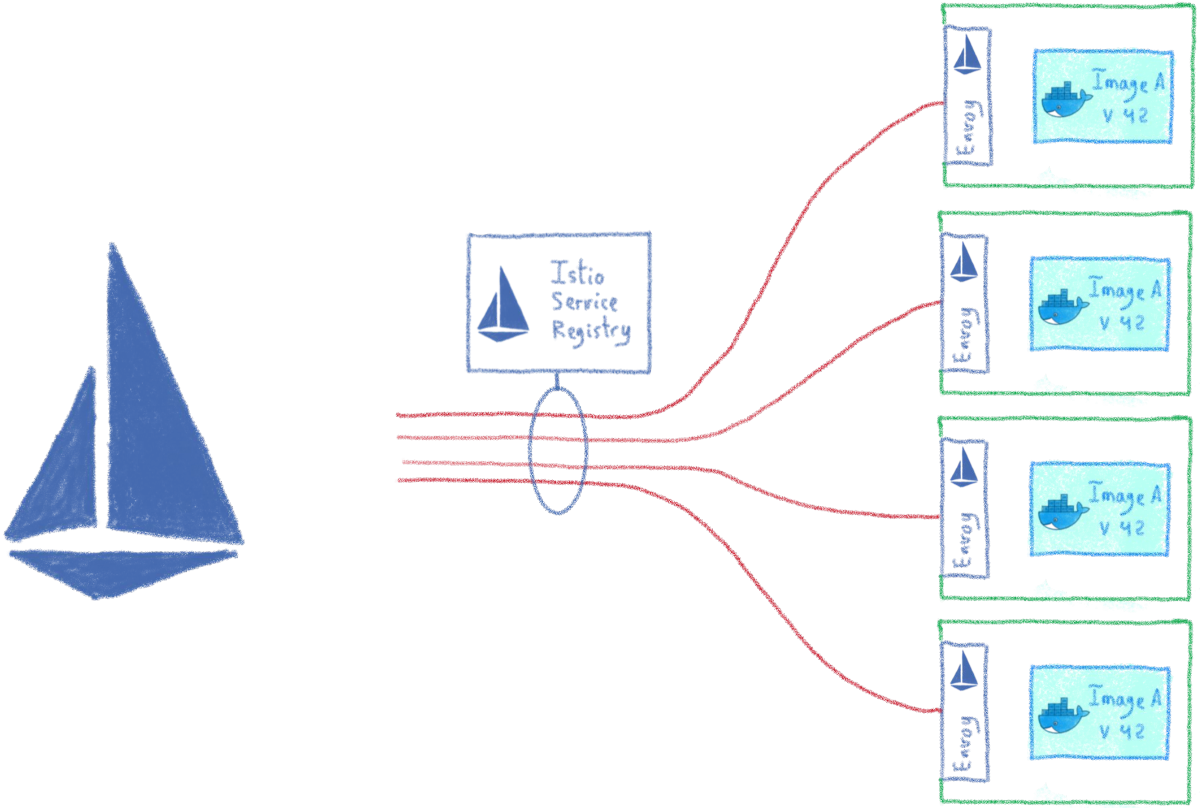
Rolling upgrades



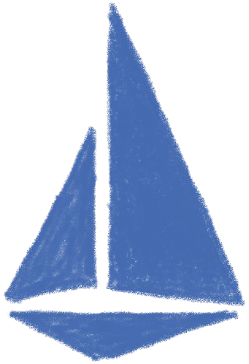
Rolling upgrades

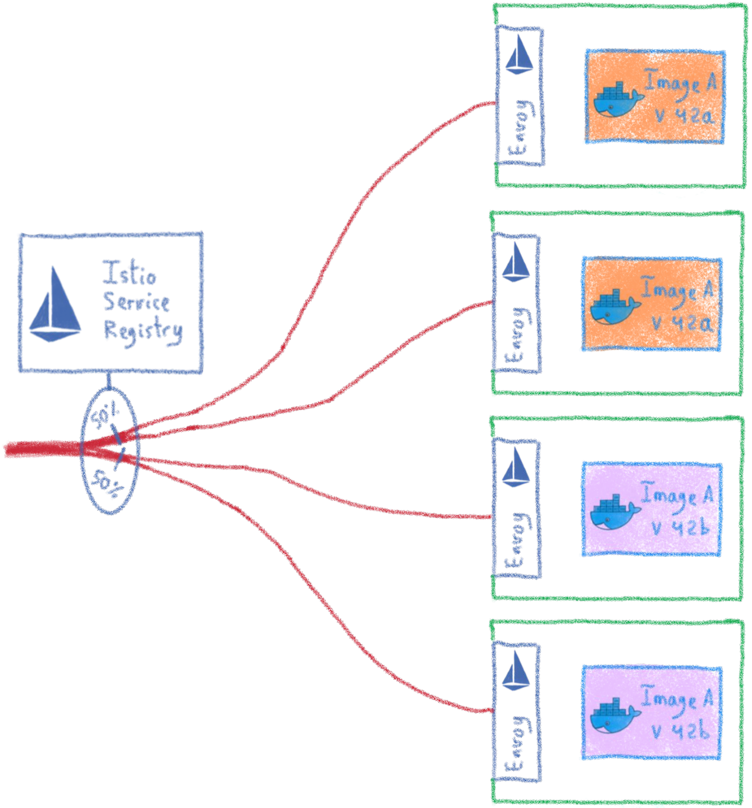


Rolling upgrades



A/B testing



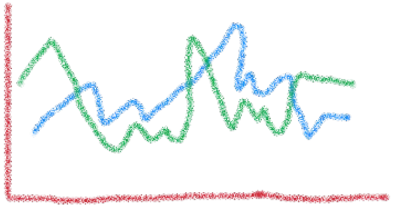

Istio
Service
Registry
50%
50%
Envoy
Image A
v 42a
Envoy
Image A
v 42a
Envoy
Image A
v 42b
Envoy
Image A
v 42b

Monitoring your cluster

- Metrics
- Logs
- Tracing

at

- Envoy level
- Control plane level

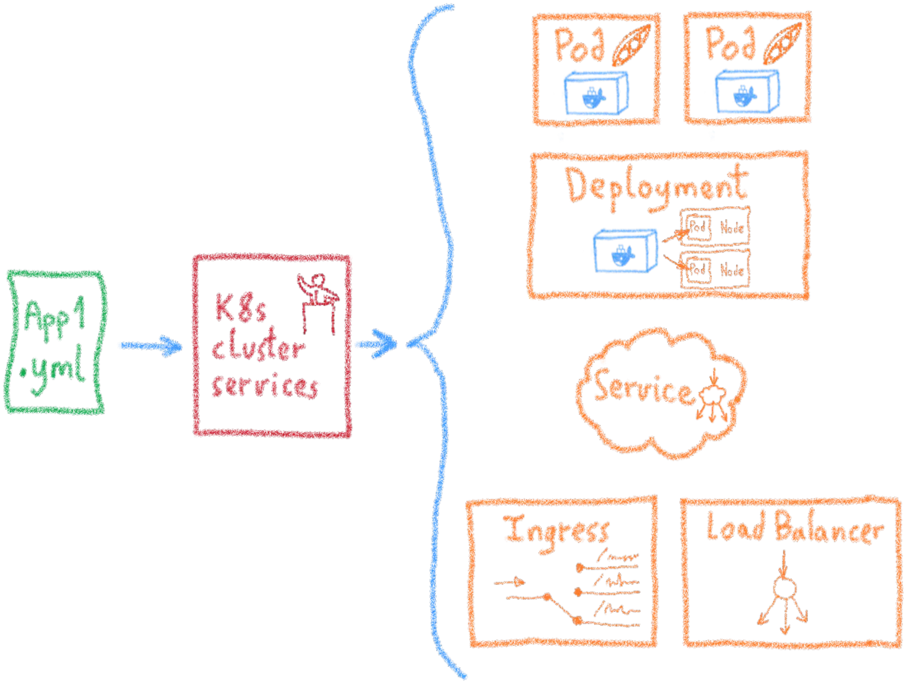


Dashboards

Velero

Backing up your Kubernetes

Kubernetes: Desired State Management



Ingress

Services

Deployments

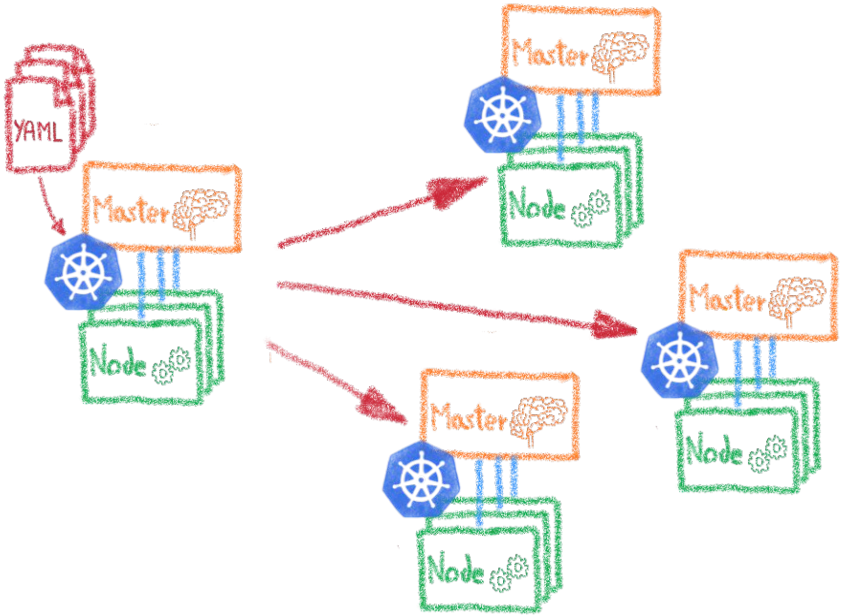
Pods

Sidecars

Replica Sets

Stateful Sets

YAML files allows to clone a cluster



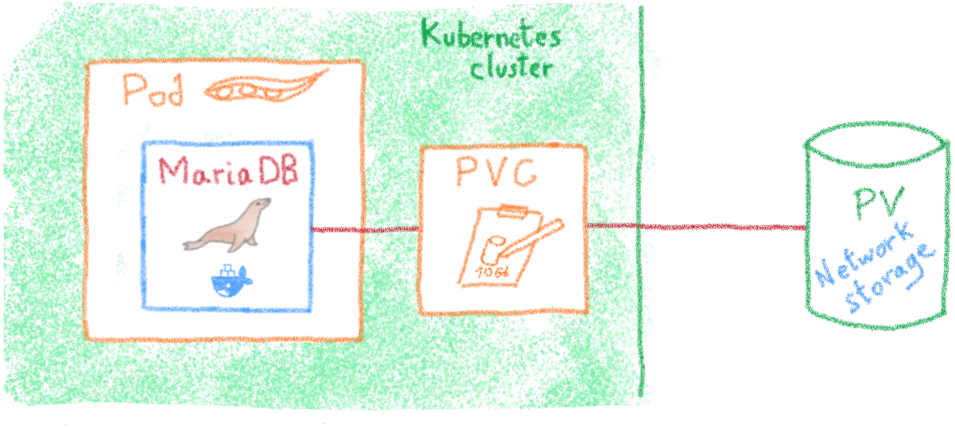
Dev envs

Staging

Multi-cluster

Multi-cloud

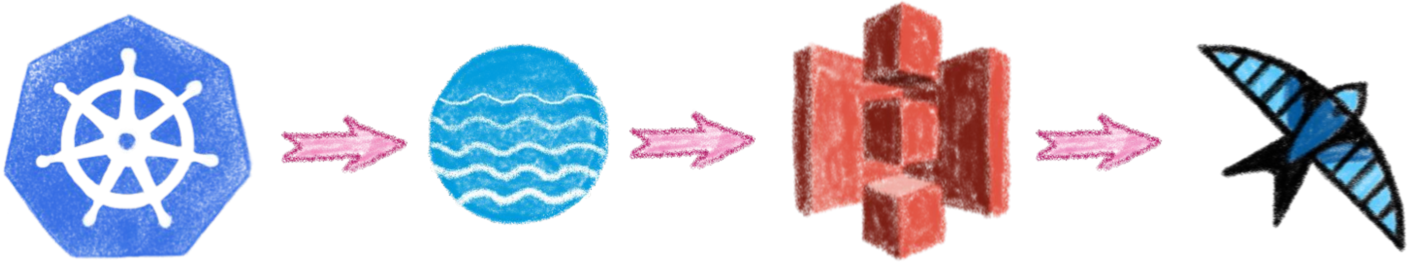
But what about the data?



Velero

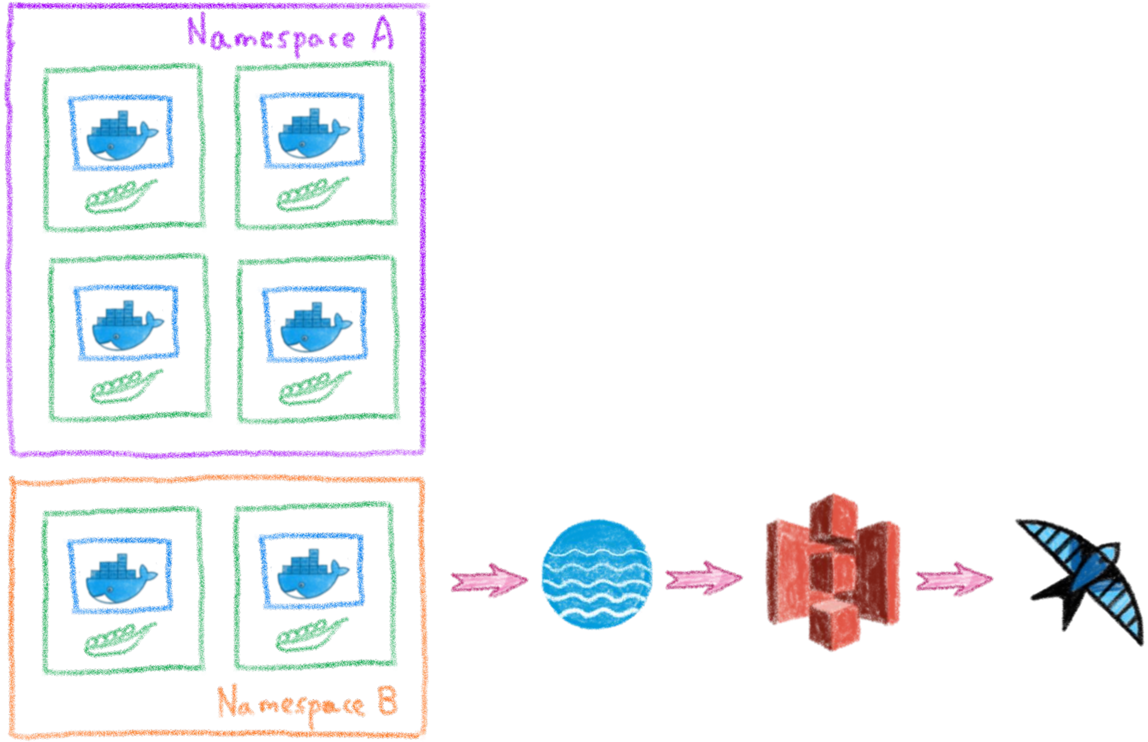
Backup and migrate Kubernetes applications and their persistent volumes

S3 based backup

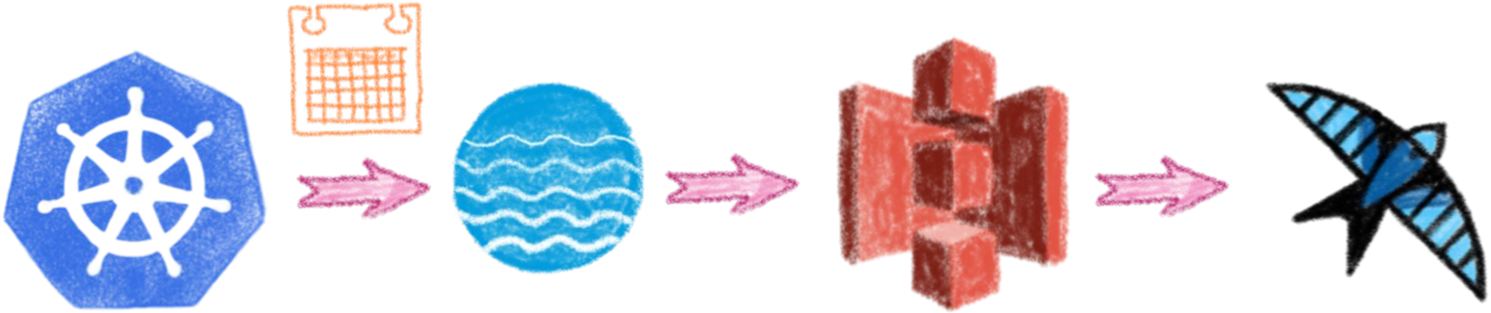


On any S3 protocol compatible store

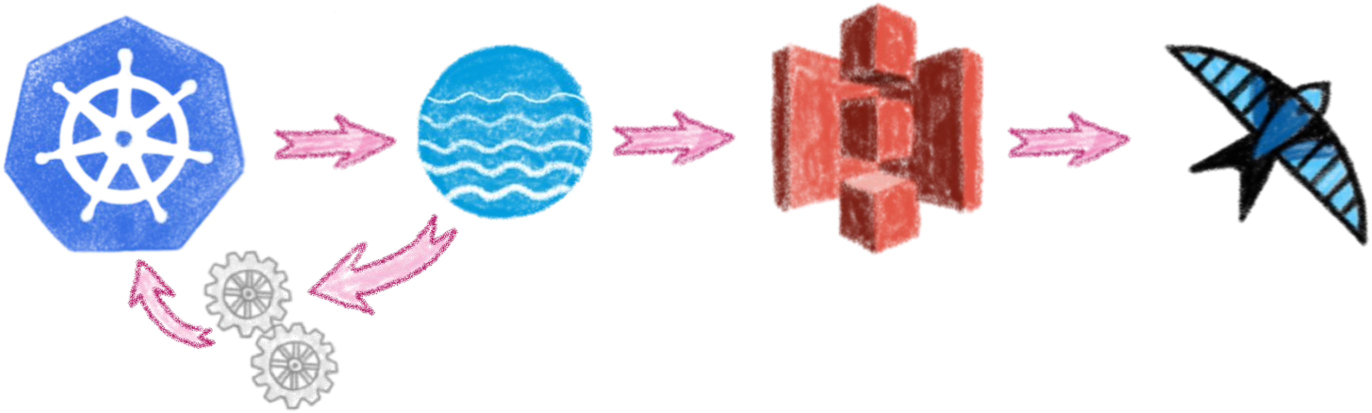
Backup all or part of a cluster



Schedule backups



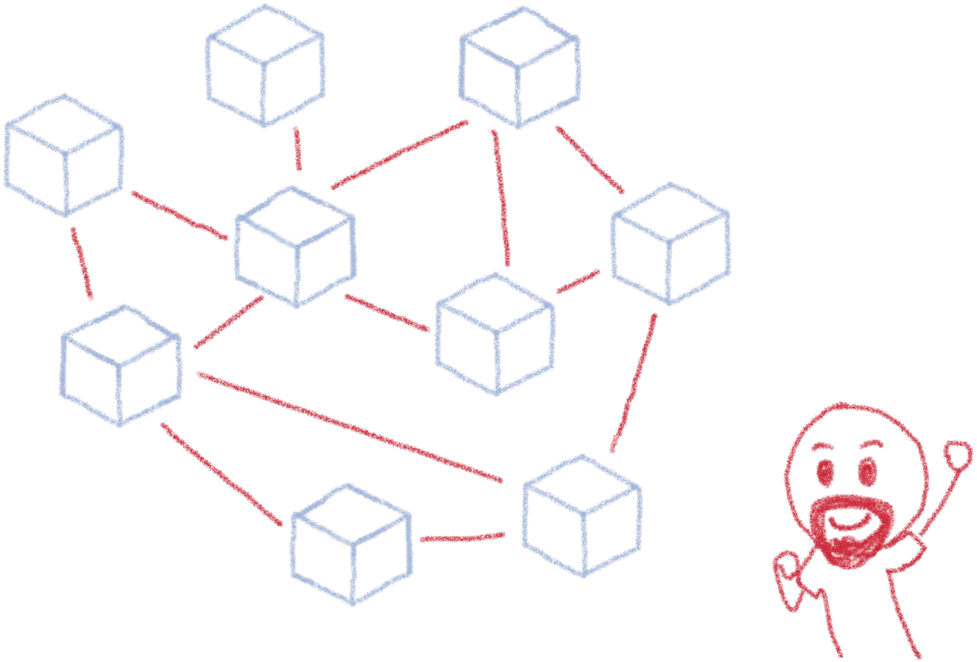
Backups hooks



Conclusion

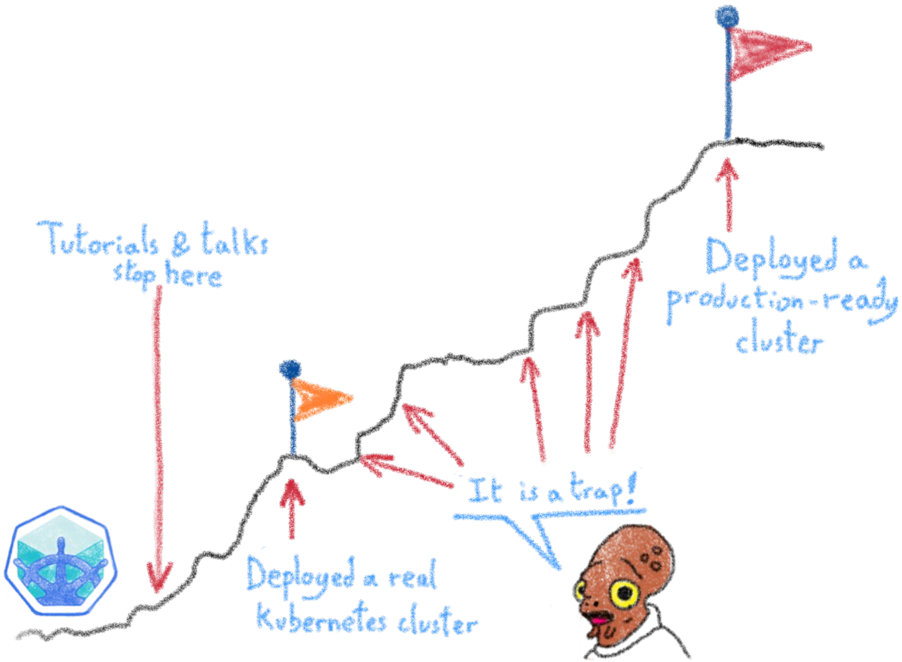
And one more thing...

Kubernetes is powerful



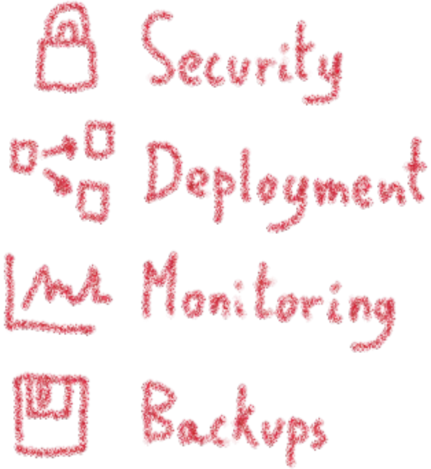
It can make Developers' and DevOps' lives easier

But there is a price: operating it


Tutorials & talks stop here
Deployed a real kubernetes cluster
It is a trap!
Deployed a production-ready cluster

Lot of things to think about

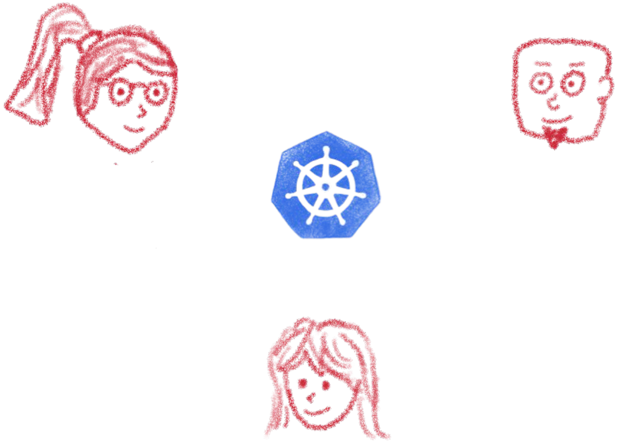
We have seen some of them



- Security
- Deployment
- Monitoring
- Backups

One more thing...

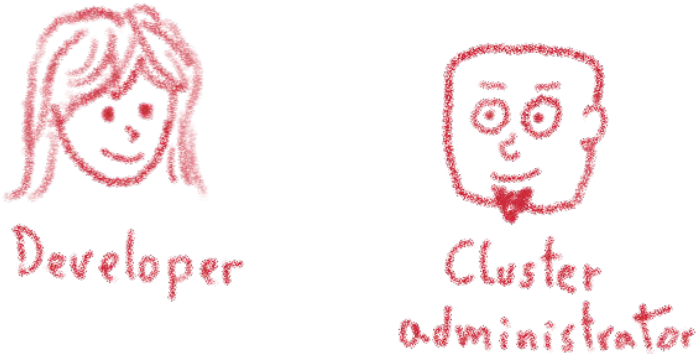
Who should do what?



Different roles

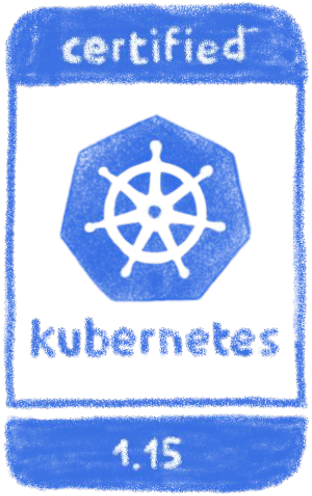
Each role asks for very different knowledge and skill sets

Most companies don't need to operate the clusters



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

OVHcloud

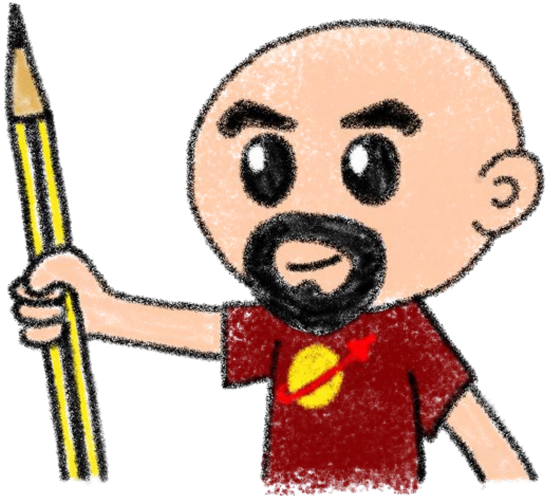
certified

kubernetes

1.15

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

Montpellier - 2020-12-18

GraalVM & Quarkus

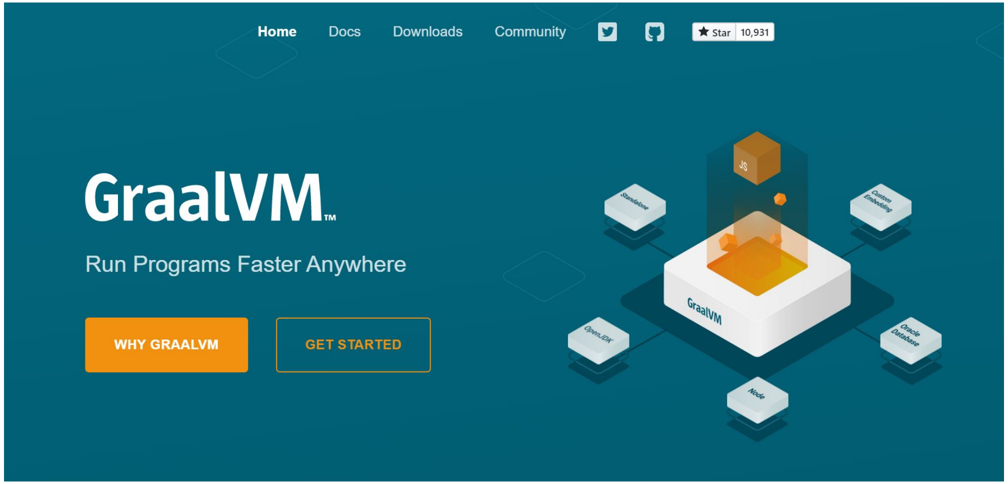
Horacio Gonzalez

@LostInBrittany

OVHcloud

GraalVM

An alternative JVM with a twist



GraalVM™

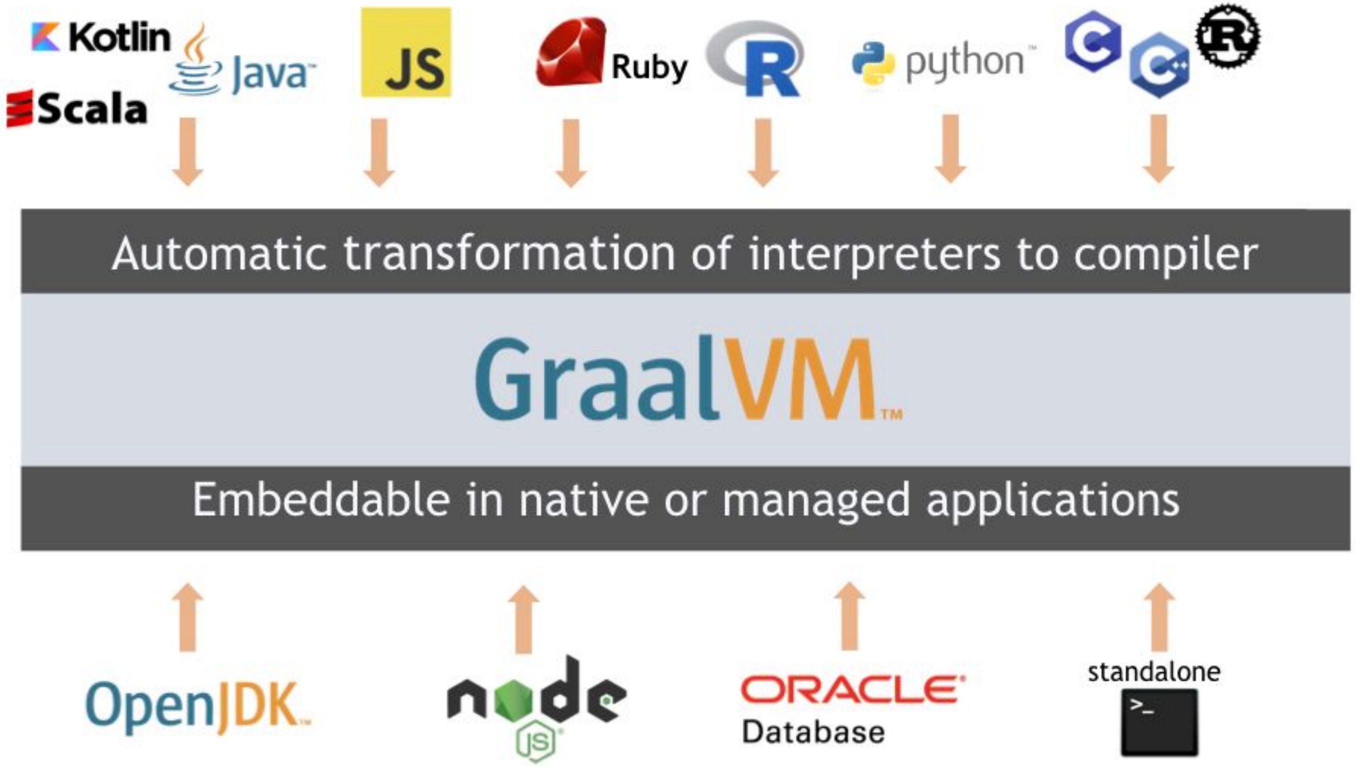
Run Programs Faster Anywhere

What's GraalVM?

A standalone Java Development Kit to execute:

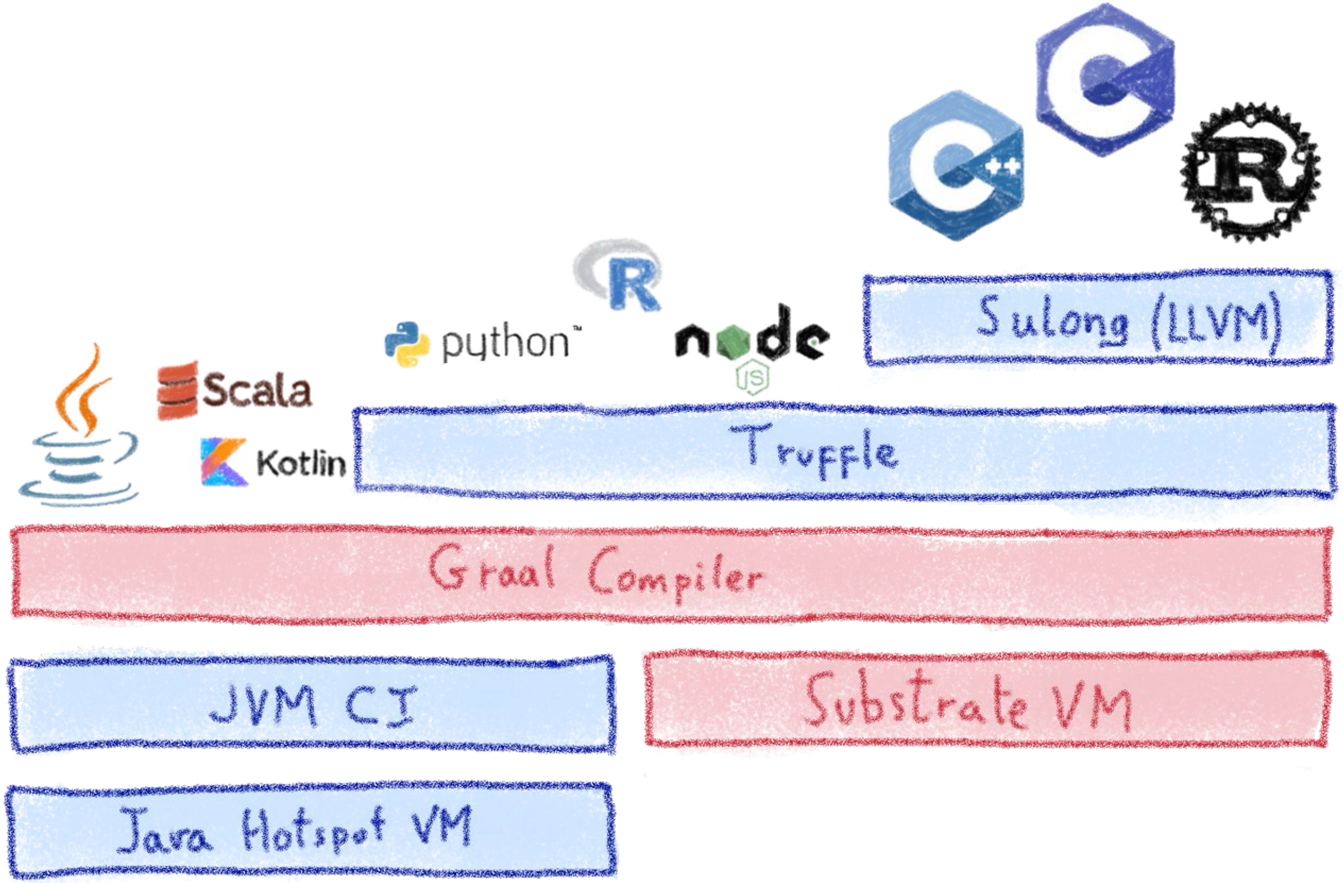
- JVM-based languages
- Dynamic languages
- LLVM-based languages

What's GraalVM?



Kotlin
Scala
Java
JS
Ruby
R
python
Automatic transformation of interpreters to compiler
GraalVM
Embeddable in native or managed applications
OpenJDK
node JS
ORACLE Database
standalone
>_

What's GraalVM?



GraalVM Features

GraalVM lets you:

- Run your code faster and more efficiently
- Interoperate directly with most modern programming languages
- Embed languages with the GraalVM SDK
- Create compiled native images
- Use a single set of tools to monitor, debug, and profile all your code

GraalVM base package

The base installation includes:

- The JVM
- The GraalVM compiler
- The LLVM bitcode interpreter
- The JavaScript runtime

Why GraalVM?

For Java programs:

- Run Java faster
- Make Your Application Extensible
- Create a Native Image

Why GraalVM?

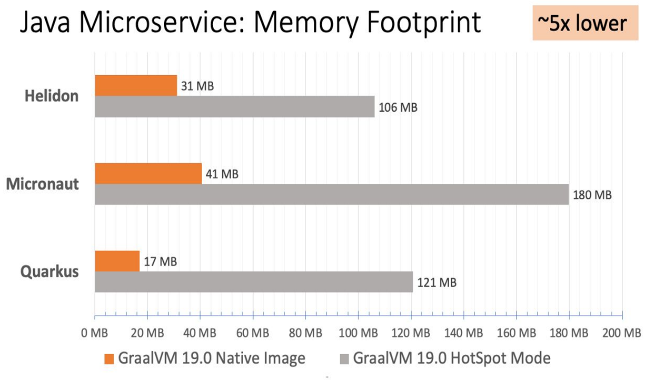
For JavaScript programs:

- Reuse Libraries from Java, R, or Python
- Run with Large Heaps
- Define Data Structures in C/C++

Why GraalVM?

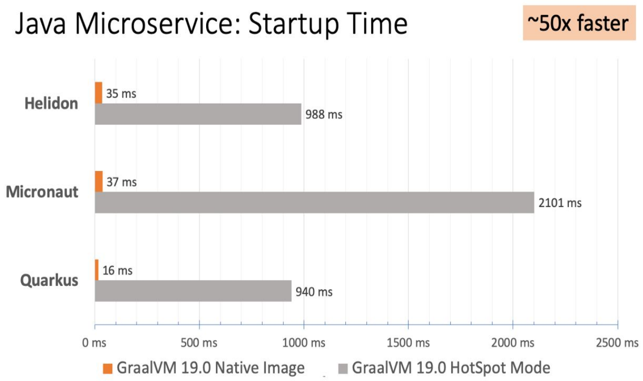
GraalVM native images reduce:

- Runtime memory footprint
- Startup time



Java Microservice: Memory Footprint

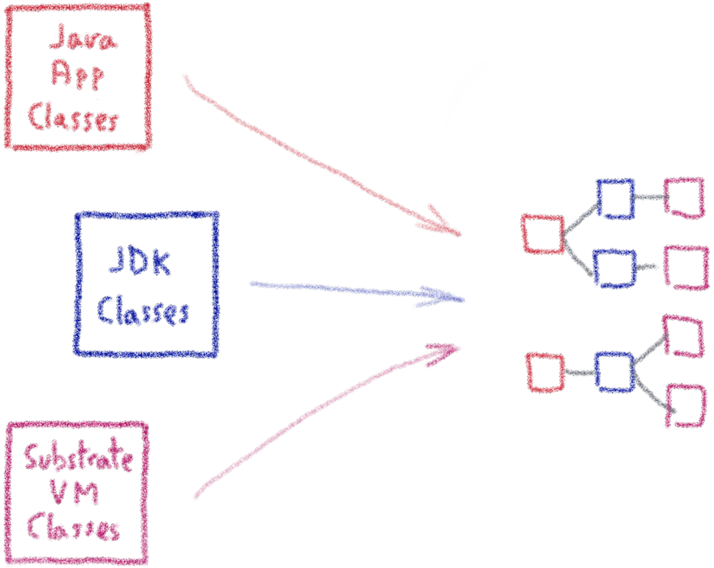
~5x lower



Java Microservice: Startup Time

~50x faster

And how does it compile ?



Dead code elimination

Closed world assumption

Adding some limitations...

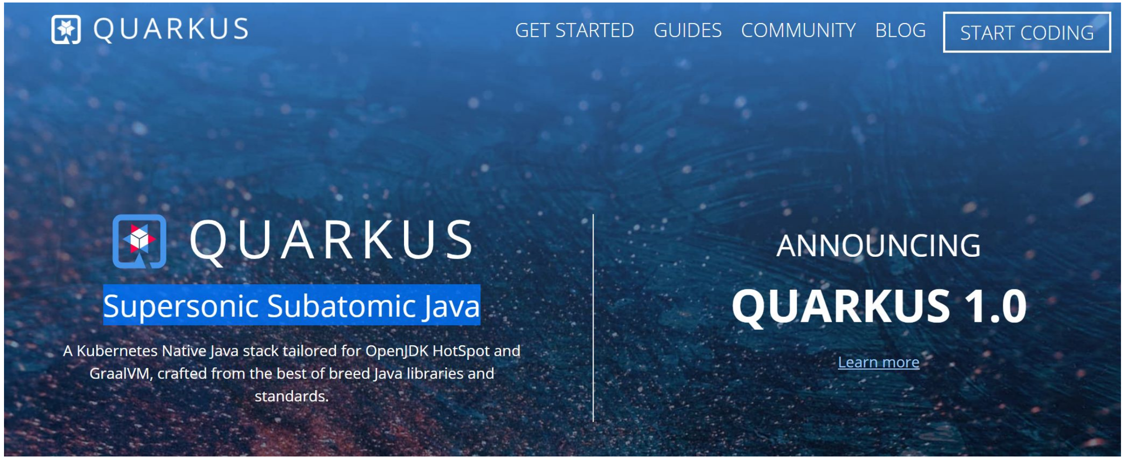
No dynamic classloading 🚫

Manual listing for reflection 📋

Manual listing for dynamic proxy 📋

Quarkus

Supersonic Subatomic Java



Supersonic Subatomic Java

A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards.

ANNOUNCING

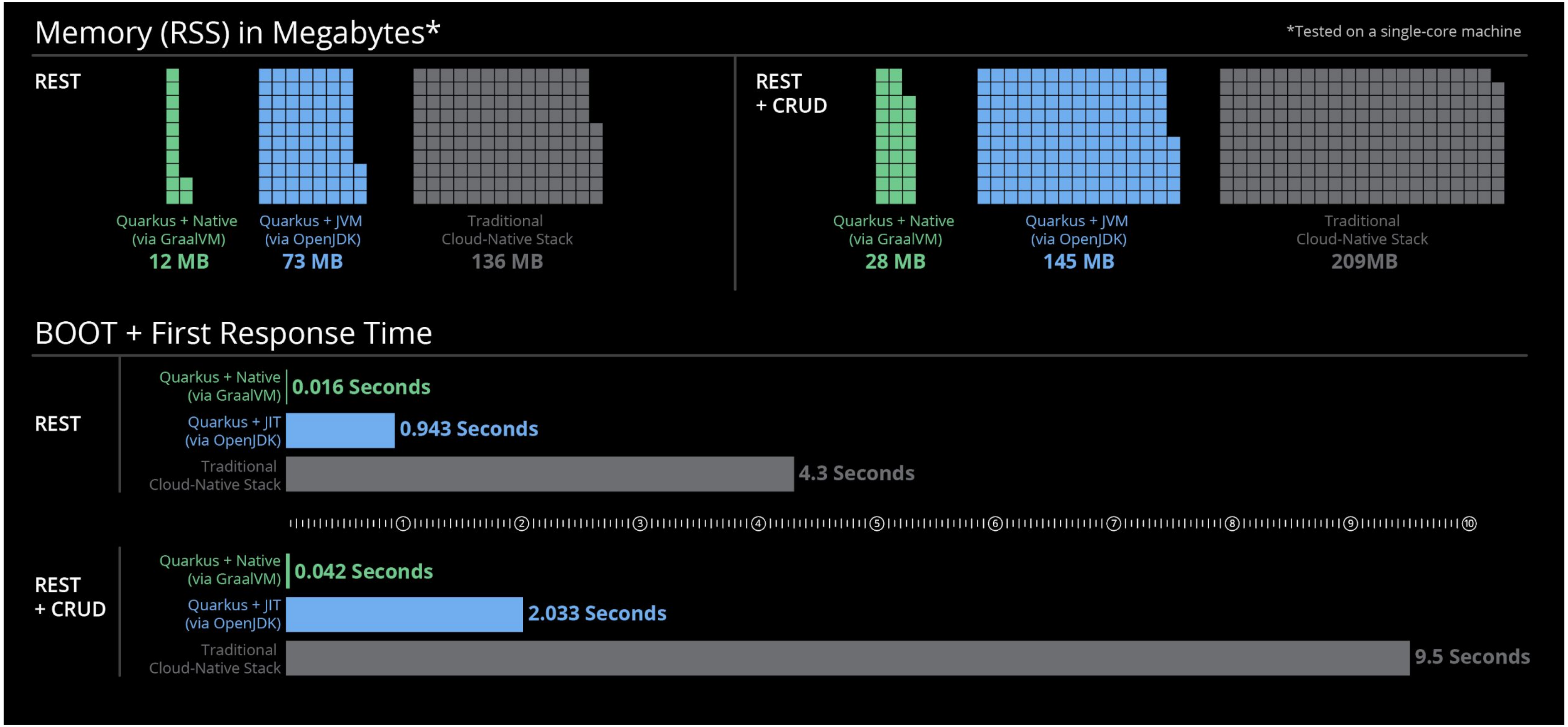
QUARKUS 1.0

Learn more

What's Quarkus?

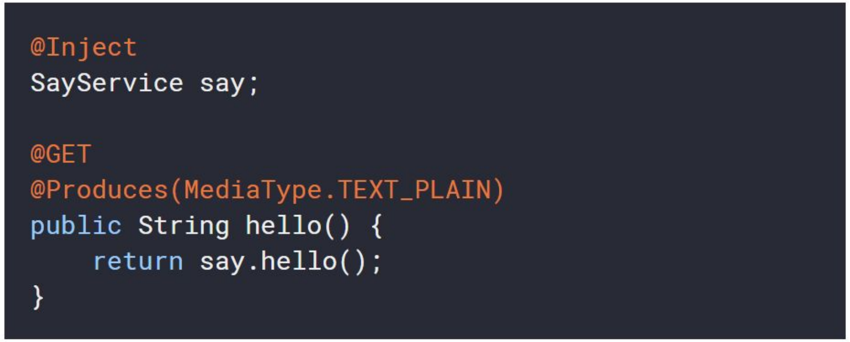
- A Kubernetes Native Java stack
- Tailored for OpenJDK HotSpot and GraalVM
- Crafted from the best of breed Java libraries and standards

Container first



Unifies imperative and reactive

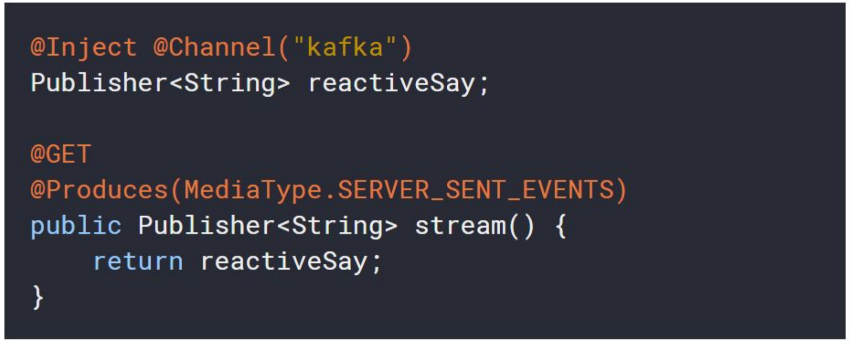
IMPERATIVE



```java
@Inject
SayService say;

@GET
@Produces(MediaType.TEXT_PLAIN)
public String hello() {
    return say.hello();
}
```

REACTIVE



```java
@Inject @Channel("kafka")
Publisher<String> reactiveSay;

@GET
@Produces(MediaType.SERVER_SENT_EVENTS)
public Publisher<String> stream() {
    return reactiveSay;
}
```

Combine imperative code and
the non-blocking reactive style

By developers, for developers

- Unified configuration
- Zero config, live reload in the blink of an eye
- Streamlined code for the 80% common usages, flexible for the 20%
- No hassle native executable generation

Leveraging the ecosystem

Over fifty best-of-breed libraries
wired on a standard backbone

Thank you for listening