OVHcloud

kubernetes

OVHcloud
Kubernetes Tech Lab Spain

Horacio Gonzalez

2022-09-20 - Madrid

2022-09-21 - Bilbao

2022-09-22 - Barcelona

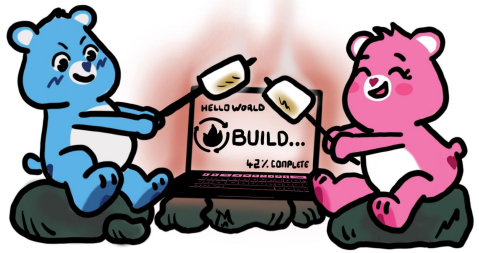
Who are we?

Introducing myself and introducing OVHcloud

Horacio Gonzalez

@LostInBrittany

Spaniard Lost in Brittany



OVHcloud

DevRel Leader

DevFest du
Bout du Monde

Google Developers
Experts
2019
Web Technologies
GDE
Flutter

OVHcloud

- **Web Cloud & Telcom**
- **Private Cloud**
- **Public Cloud**
- **Storage**
- **Network & Security**

30 Data Centers
in 12 locations

34 Points of Presence
on a 20 TBPS Bandwidth Network

2200 Employees
worldwide

115K Private Cloud
VMS running

300K Public Cloud
instances running

380K Physical Servers
running in our data centers

1 Million+ Servers
produced since 1999

1.5 Million Customers
across 132 countries

3.8 Million Websites
hosting

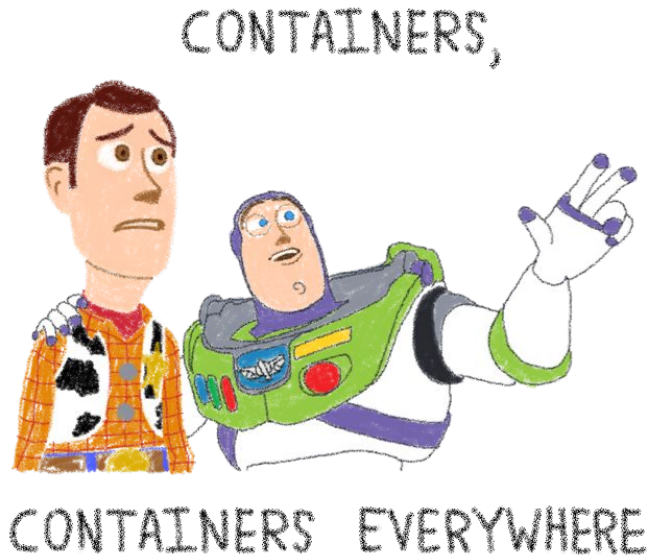
1.5 Billion Euros Invested
since 2016

P.U.E. 1.09
Energy efficiency indicator

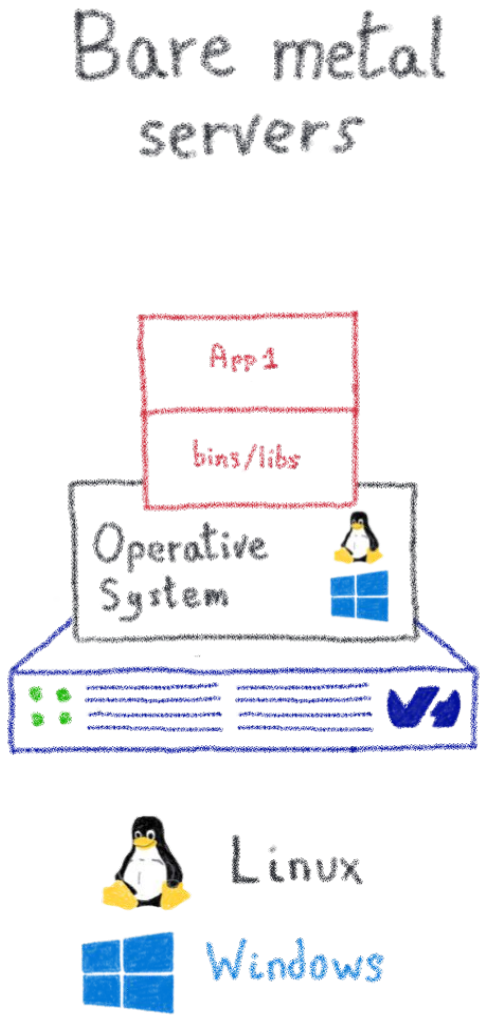
20+ Years in Business
Disrupting since 1999

Why do we need Kubernetes?

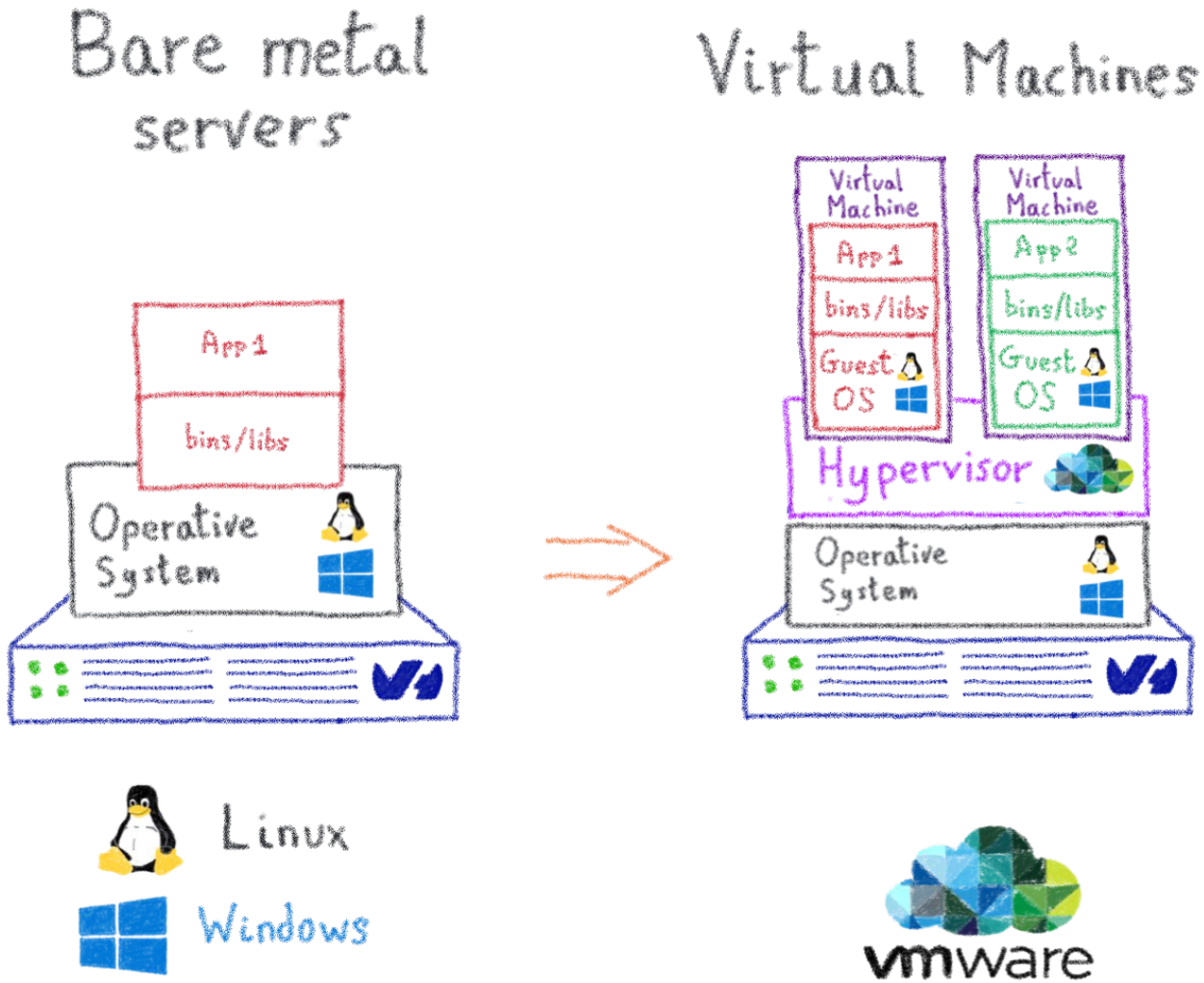
Taming the complexity of operating containers



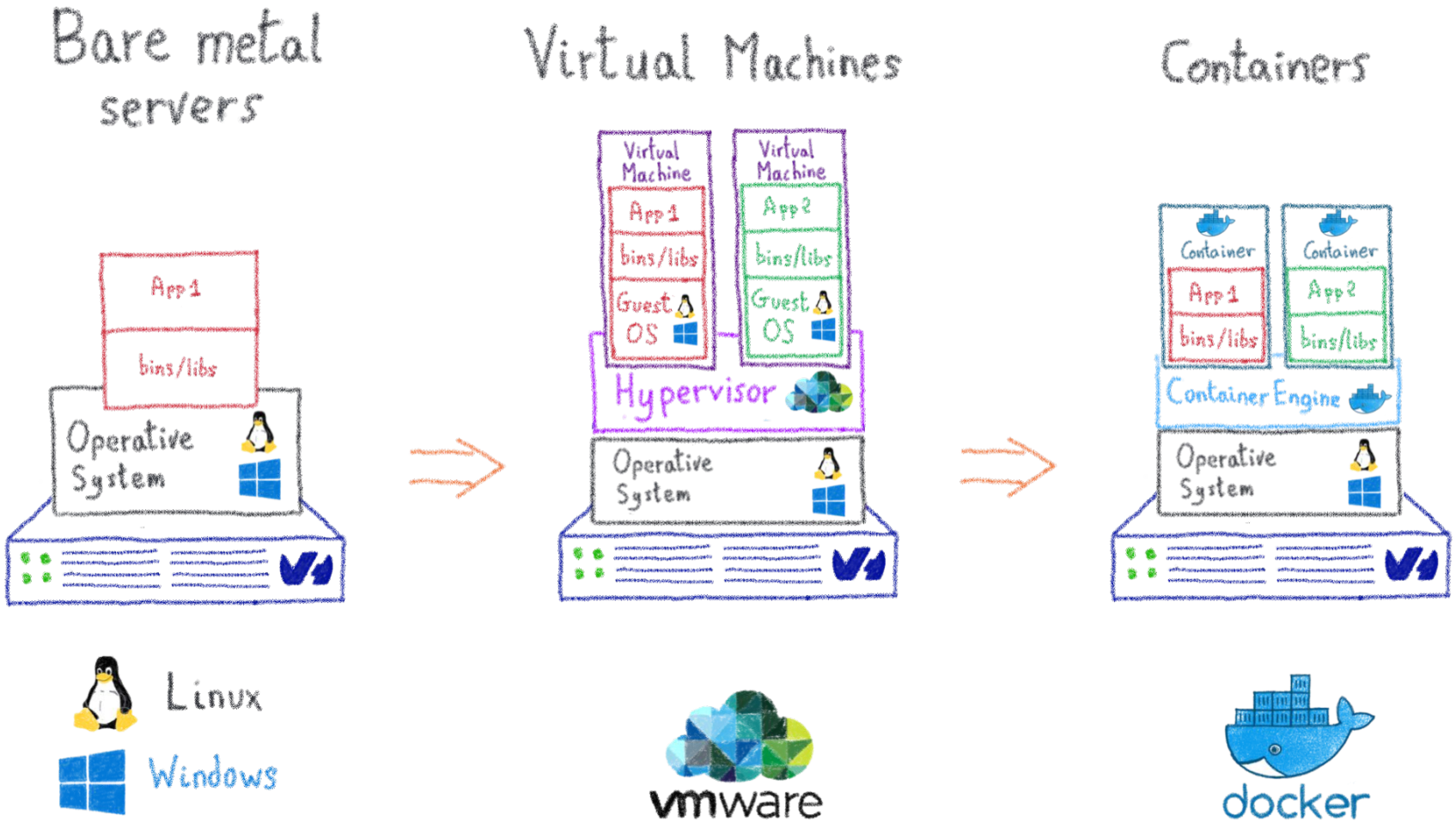
From bare metal to containers



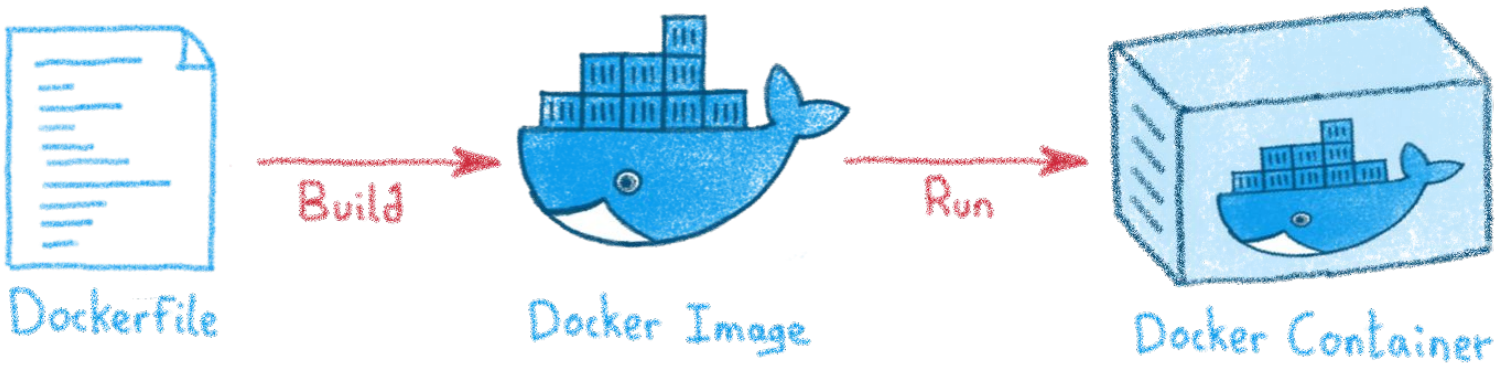
From bare metal to containers



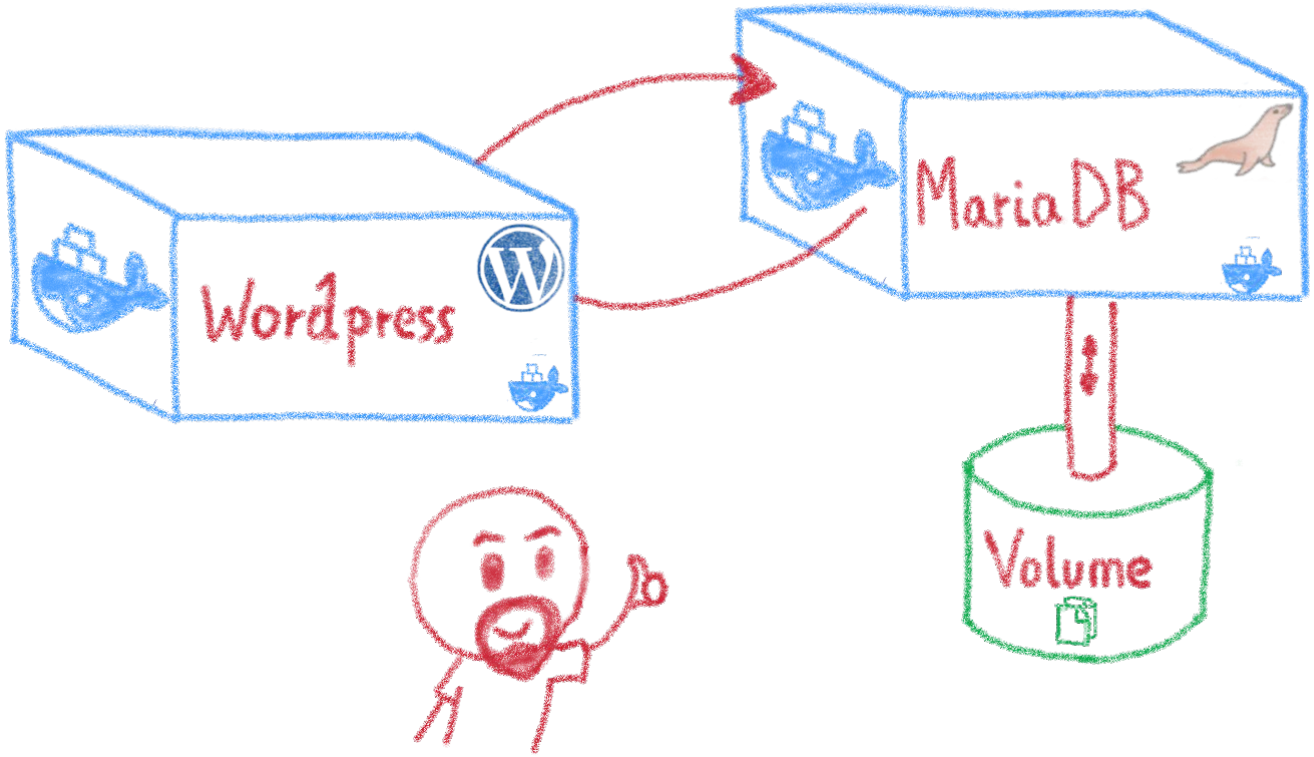
From bare metal to containers



Dockerfiles, images and containers

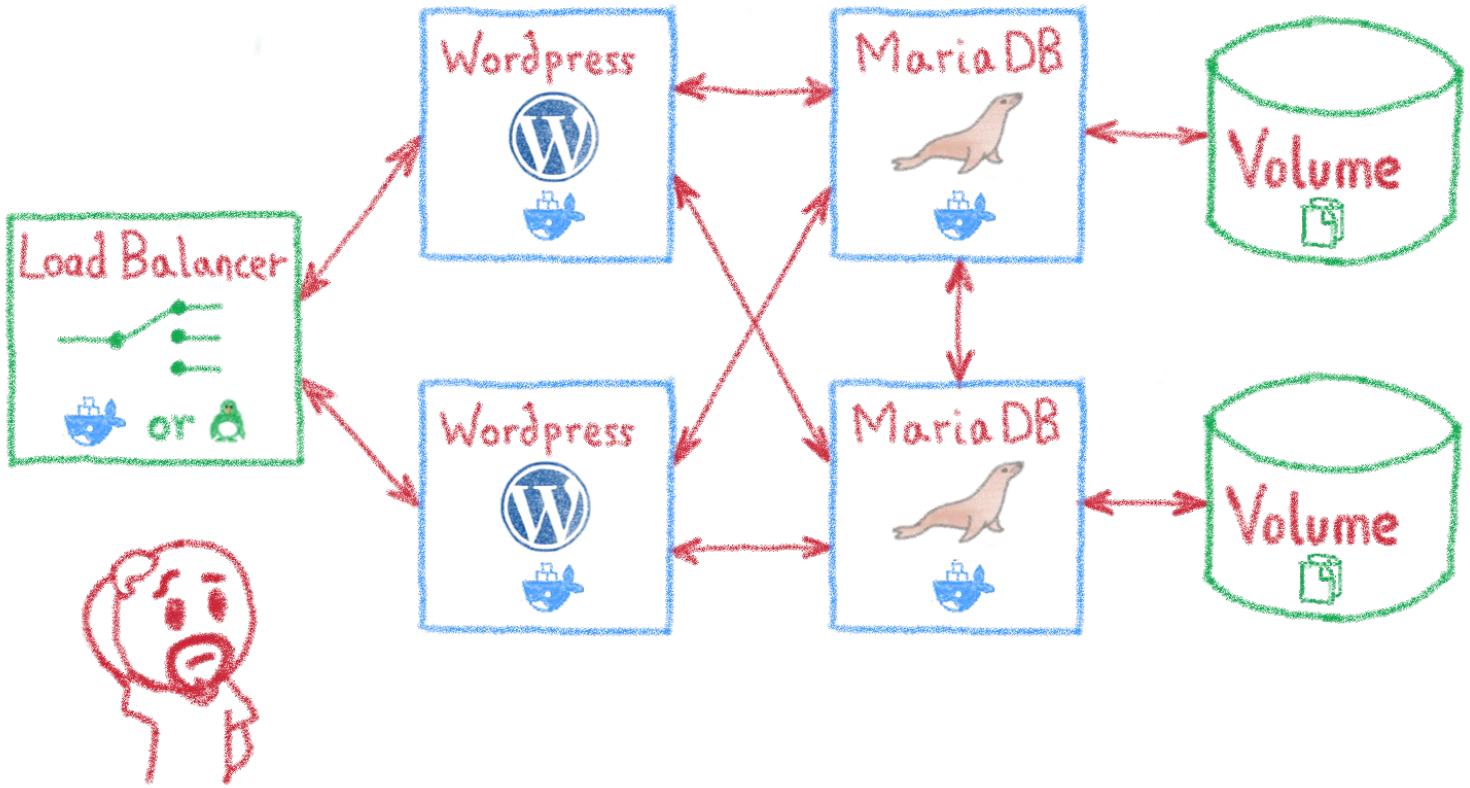


Containers are easy…



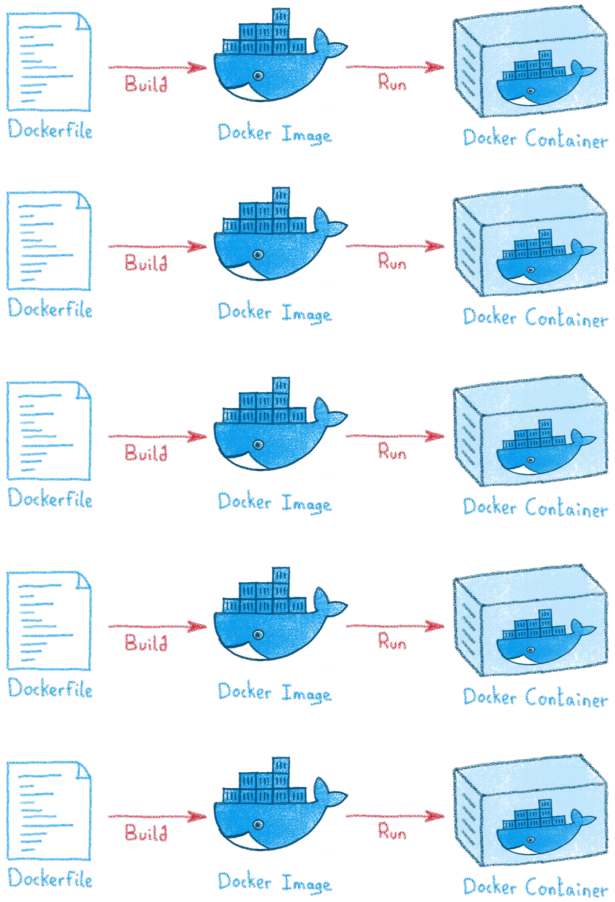
For developers

Less simple if you must operate them



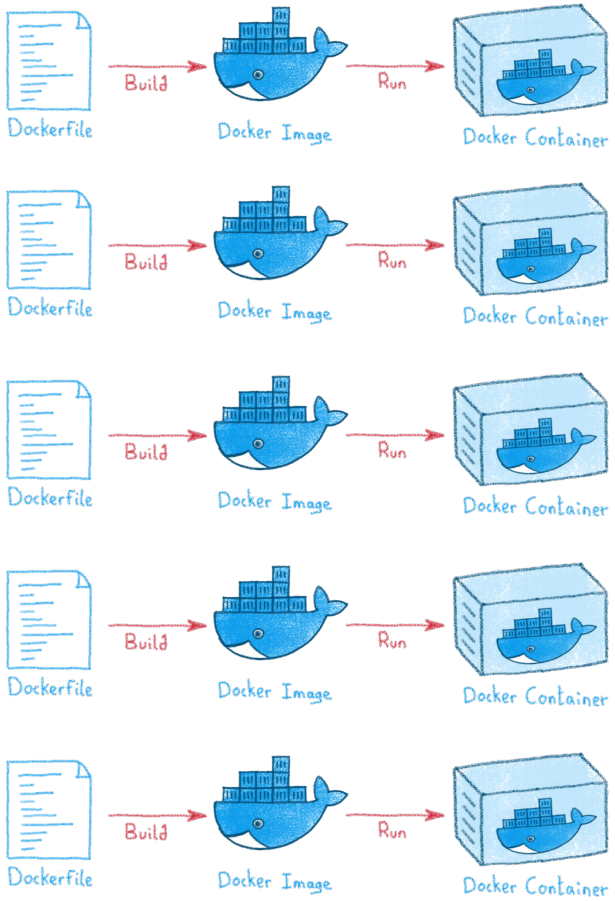
Like in a production context

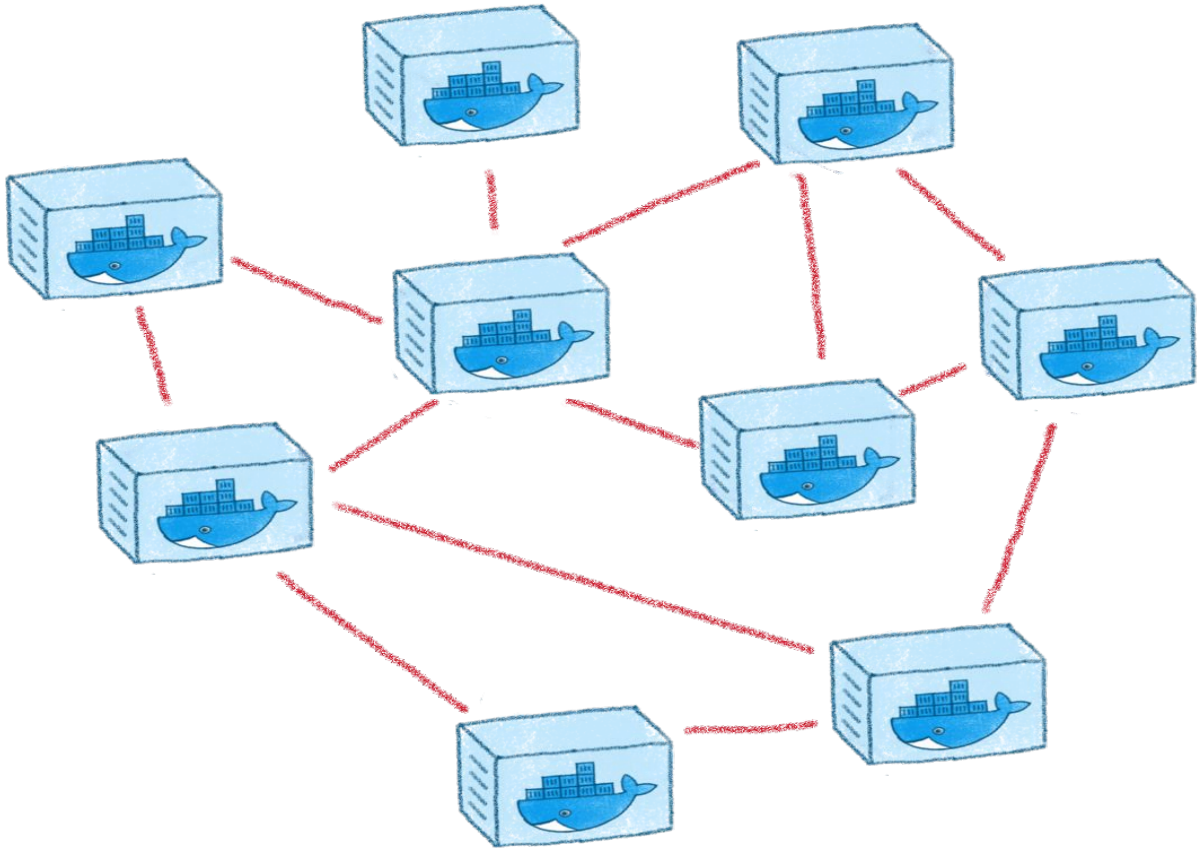
And what about microservices?



Are you sure you want to operate them by hand?

And what about microservices?

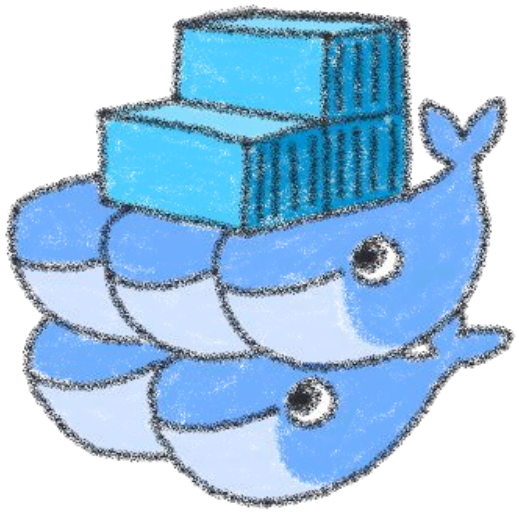




Are you sure you want to operate them by hand?

Helping to tame de complexity

Docker
Compose



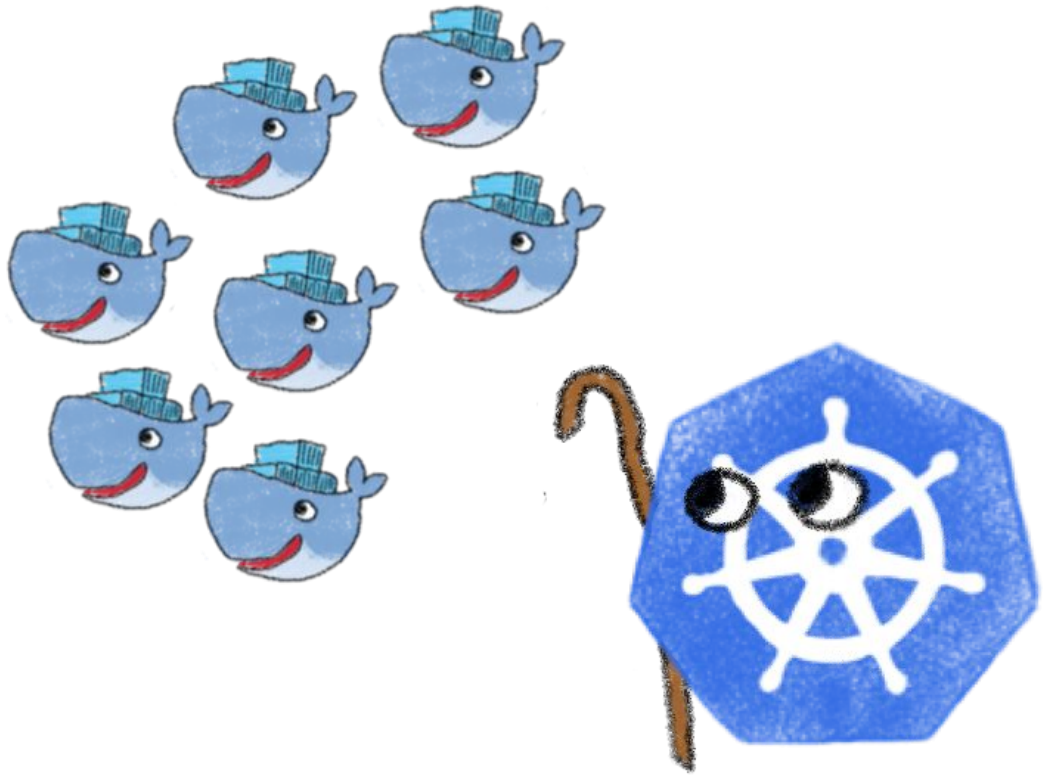
Docker
Swarm

kubernetes

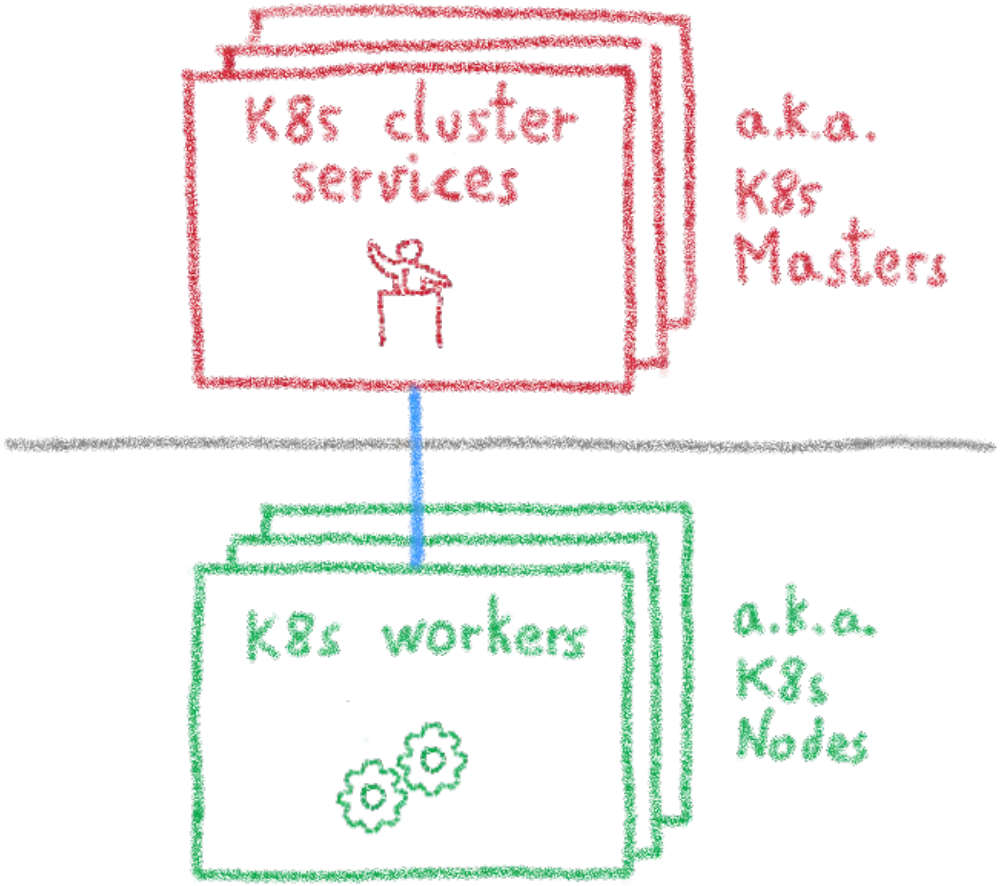
Kubernetes: a full orchestrator

Takes care of:

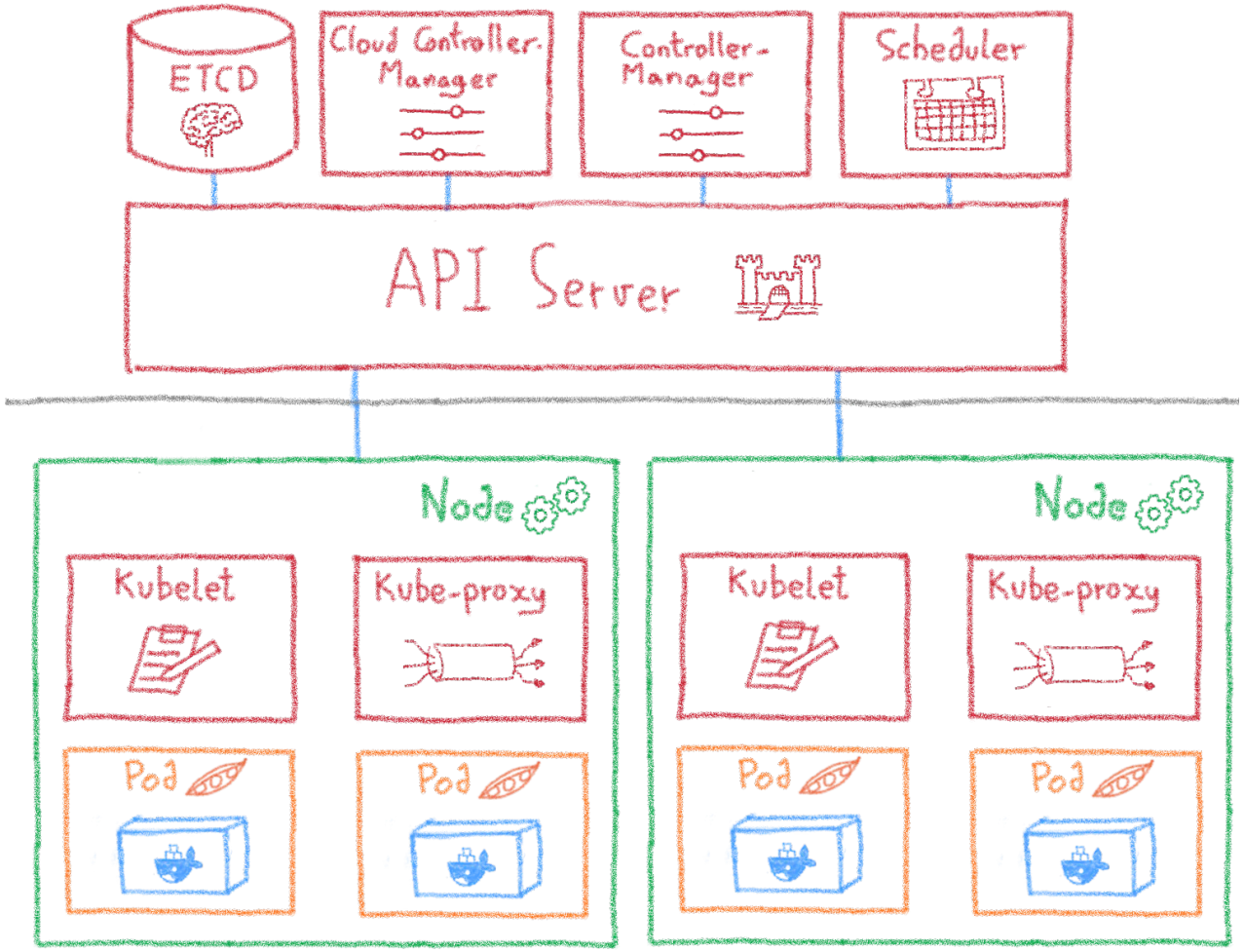
- Deployment
- Scaling
- Monitoring
- Repairing
- Securing
- ...



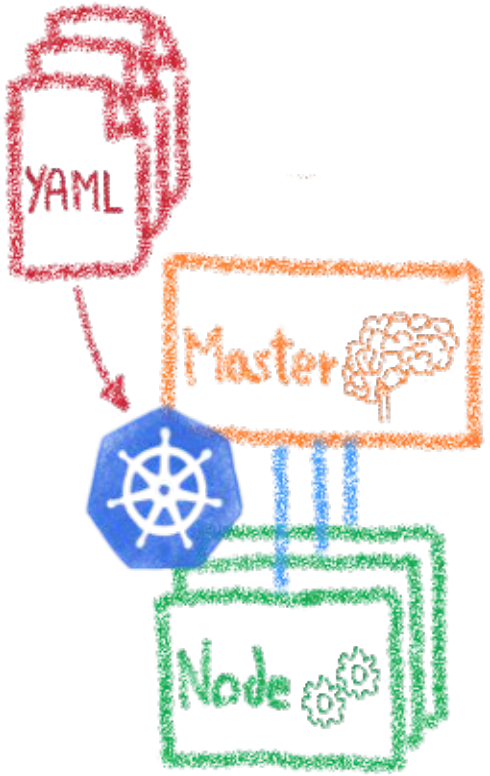
Kubernetes cluster: masters and nodes



Kubernetes cluster: more details



Desired State Management



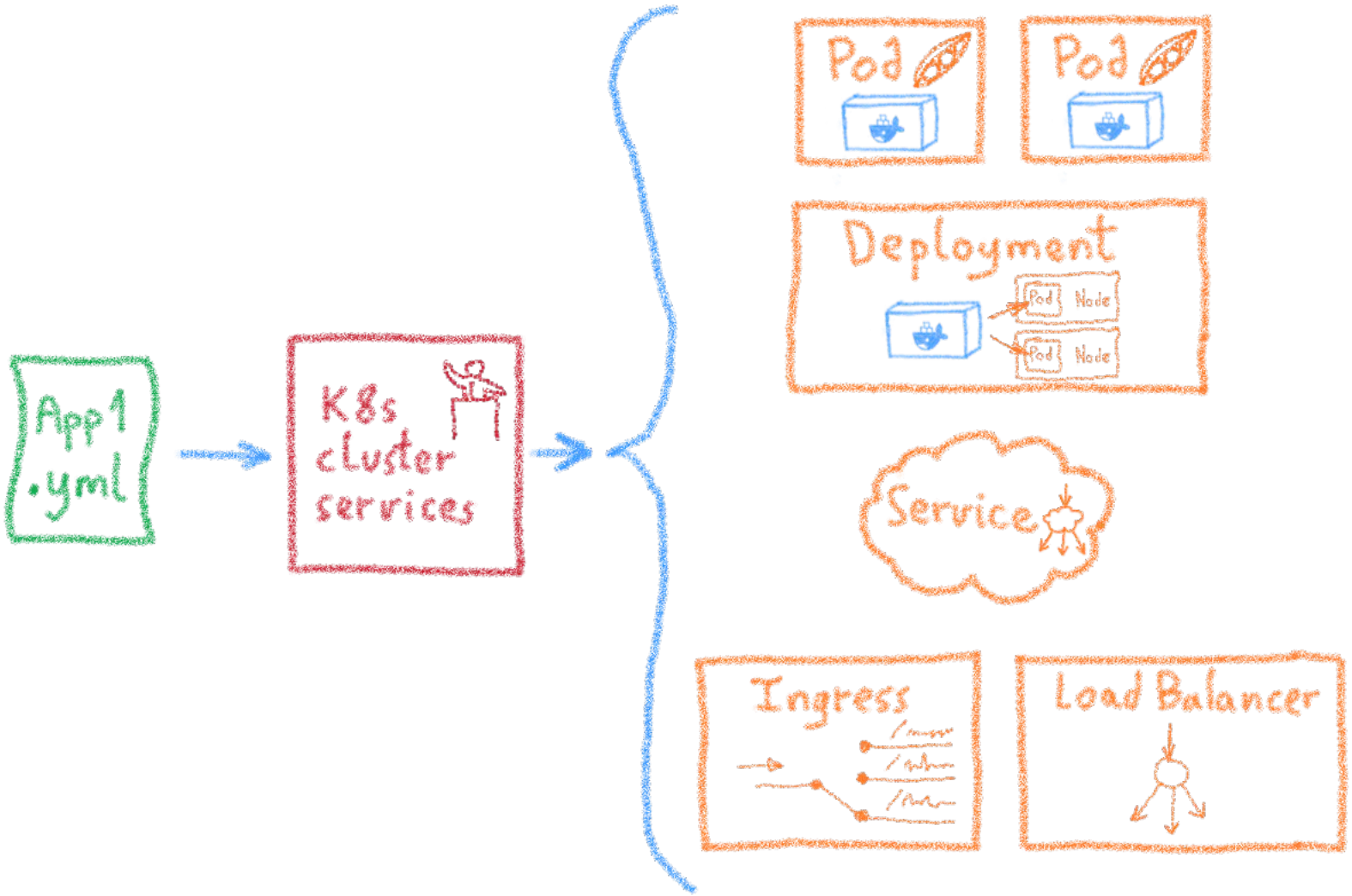
Manifest files:

Text files in YAML format

High-level description of the target architecture

Declarative infrastructure

Desired State Management



Ingress

Services

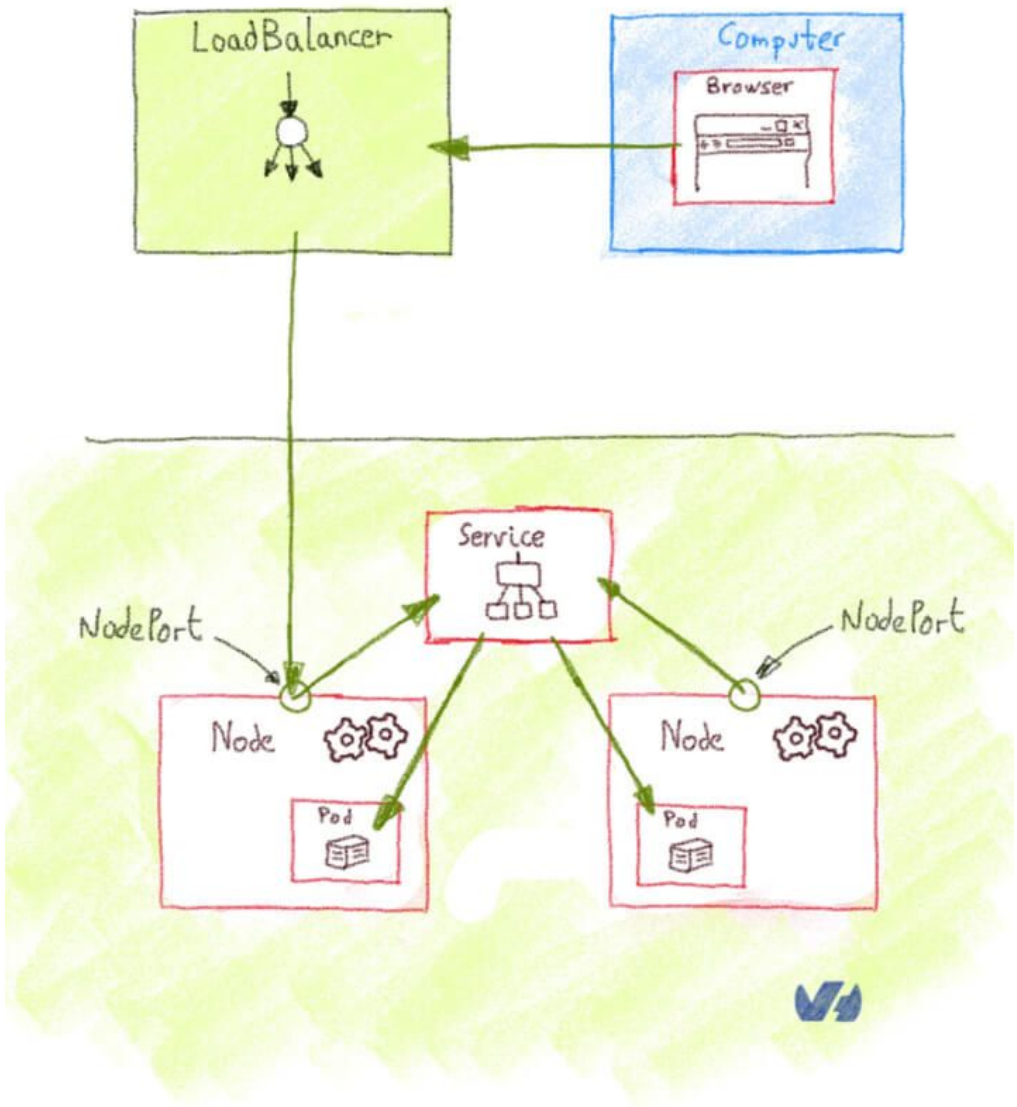
Deployments

Pods

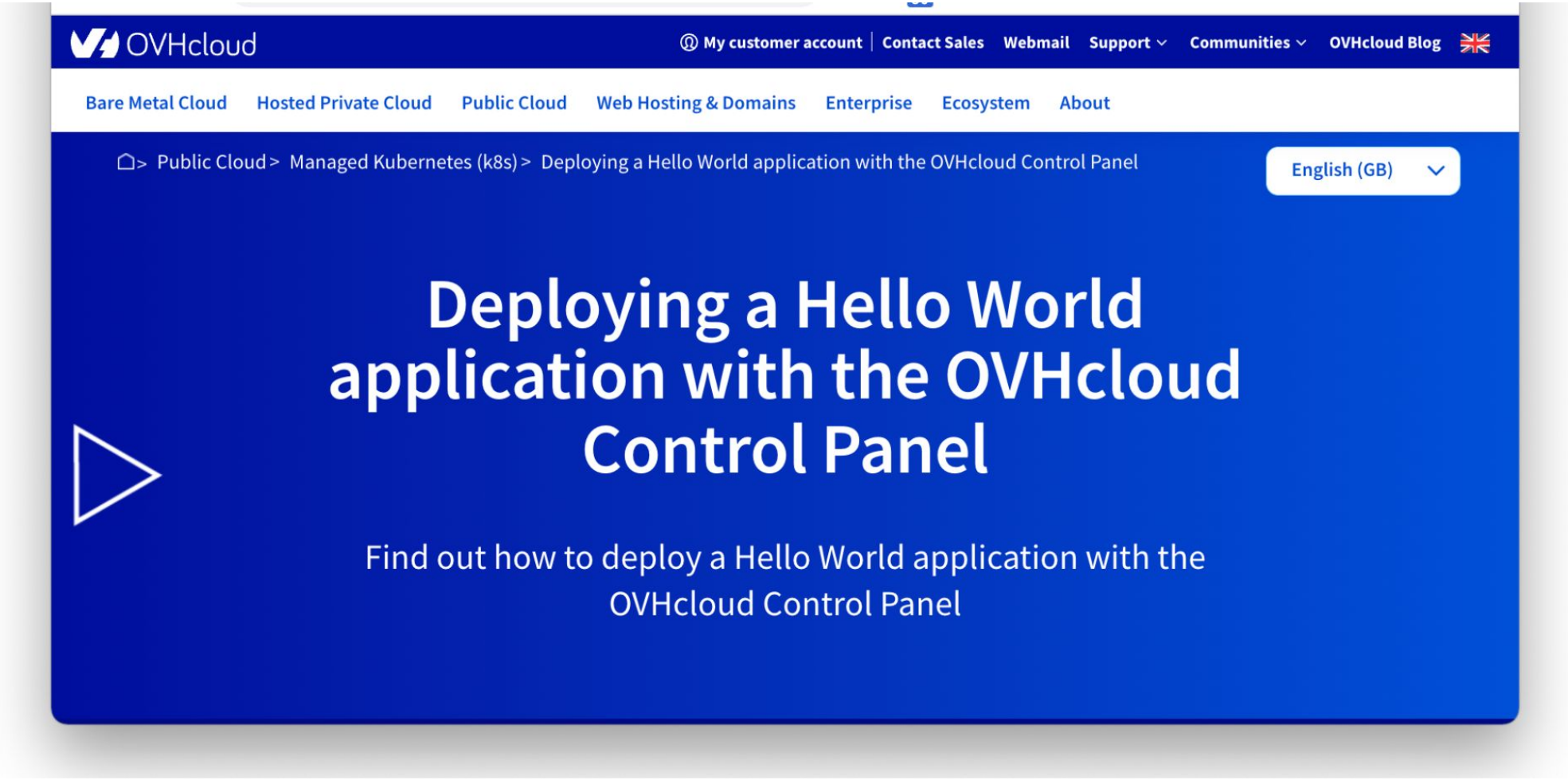
Sidecars

Replica Sets

Let's deploy an application

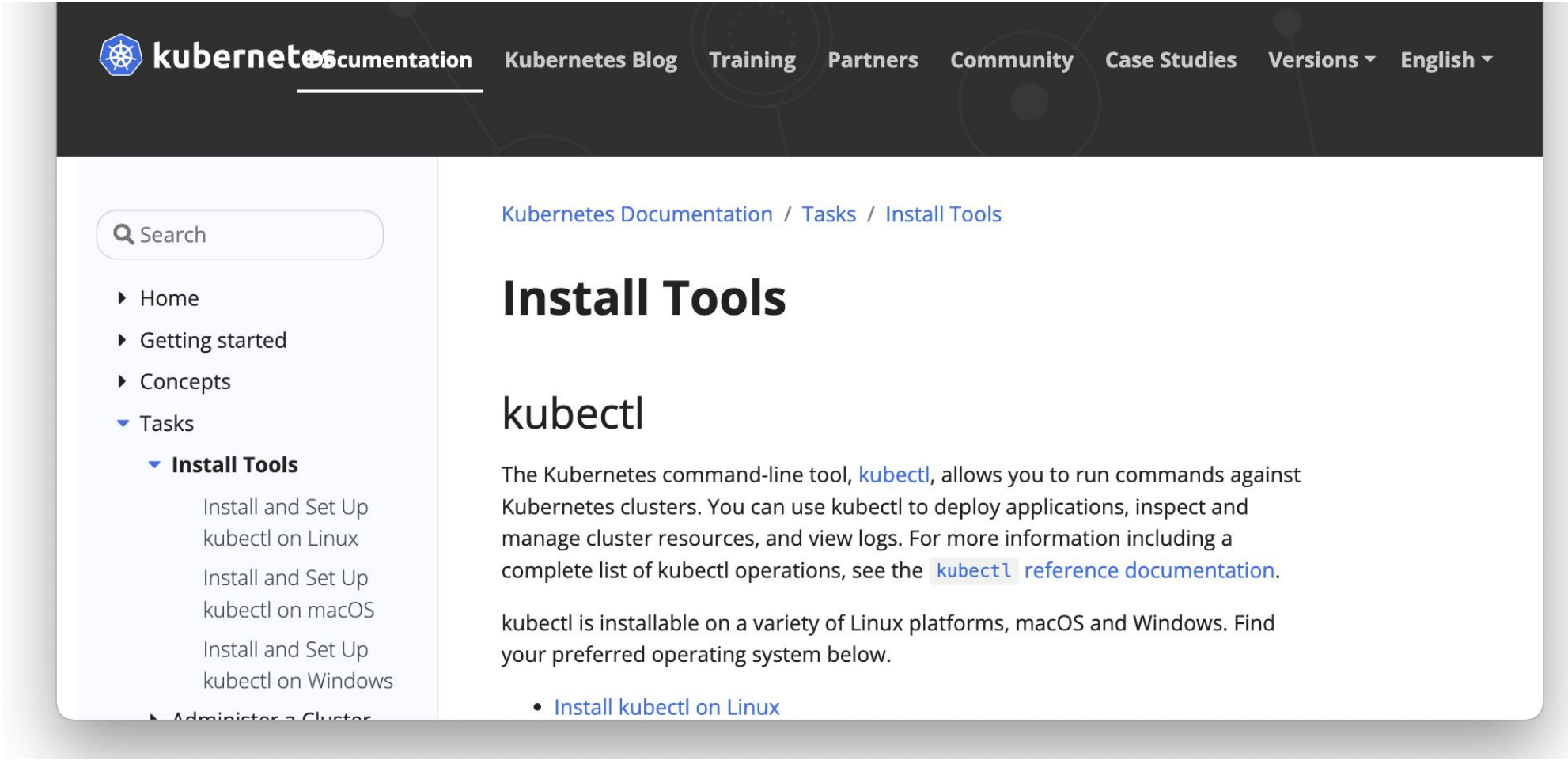


Demo: Hello Kubernetes World

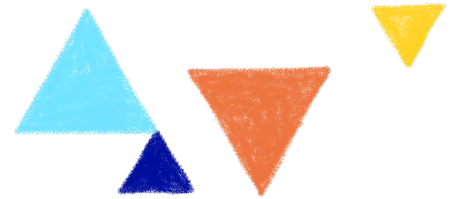


https://docs.ovh.com/gb/en/kubernetes/deploying-hello-world/

Needed tools: kubectl



https://kubernetes.io/docs/tasks/tools/



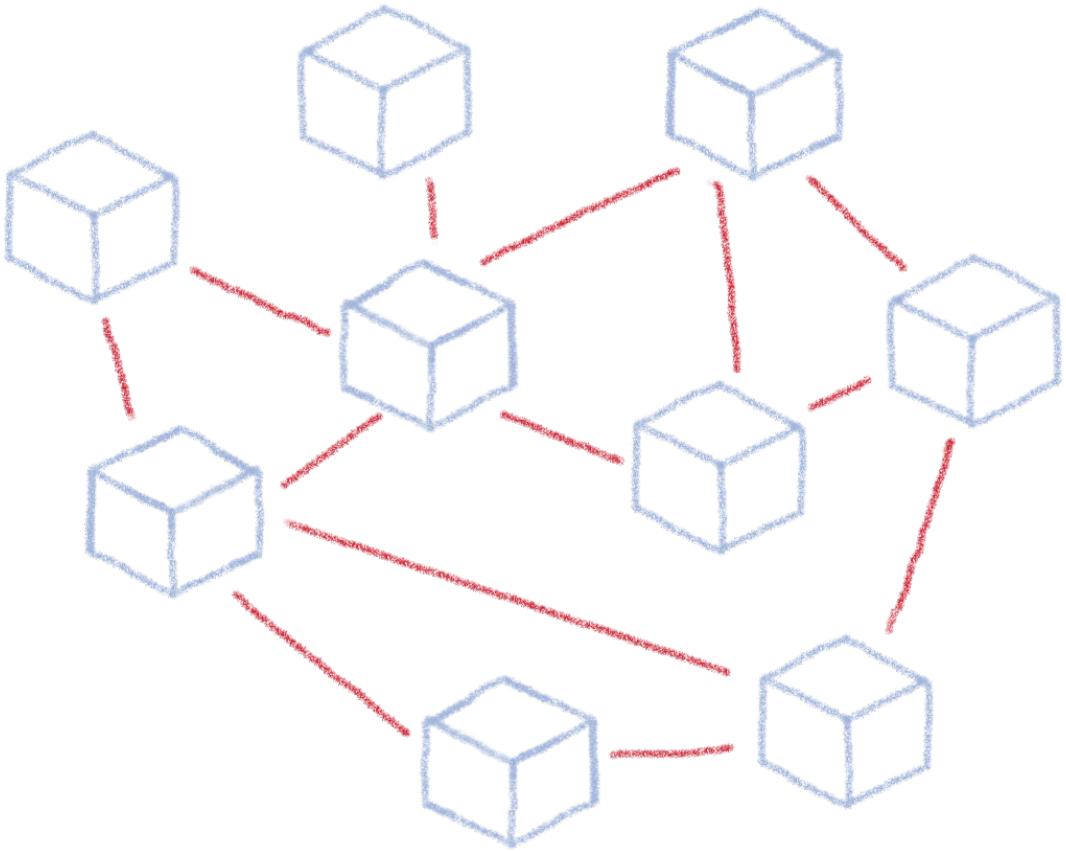
Putting Kubernetes in production

A journey not for the faint of heart

OVHcloud

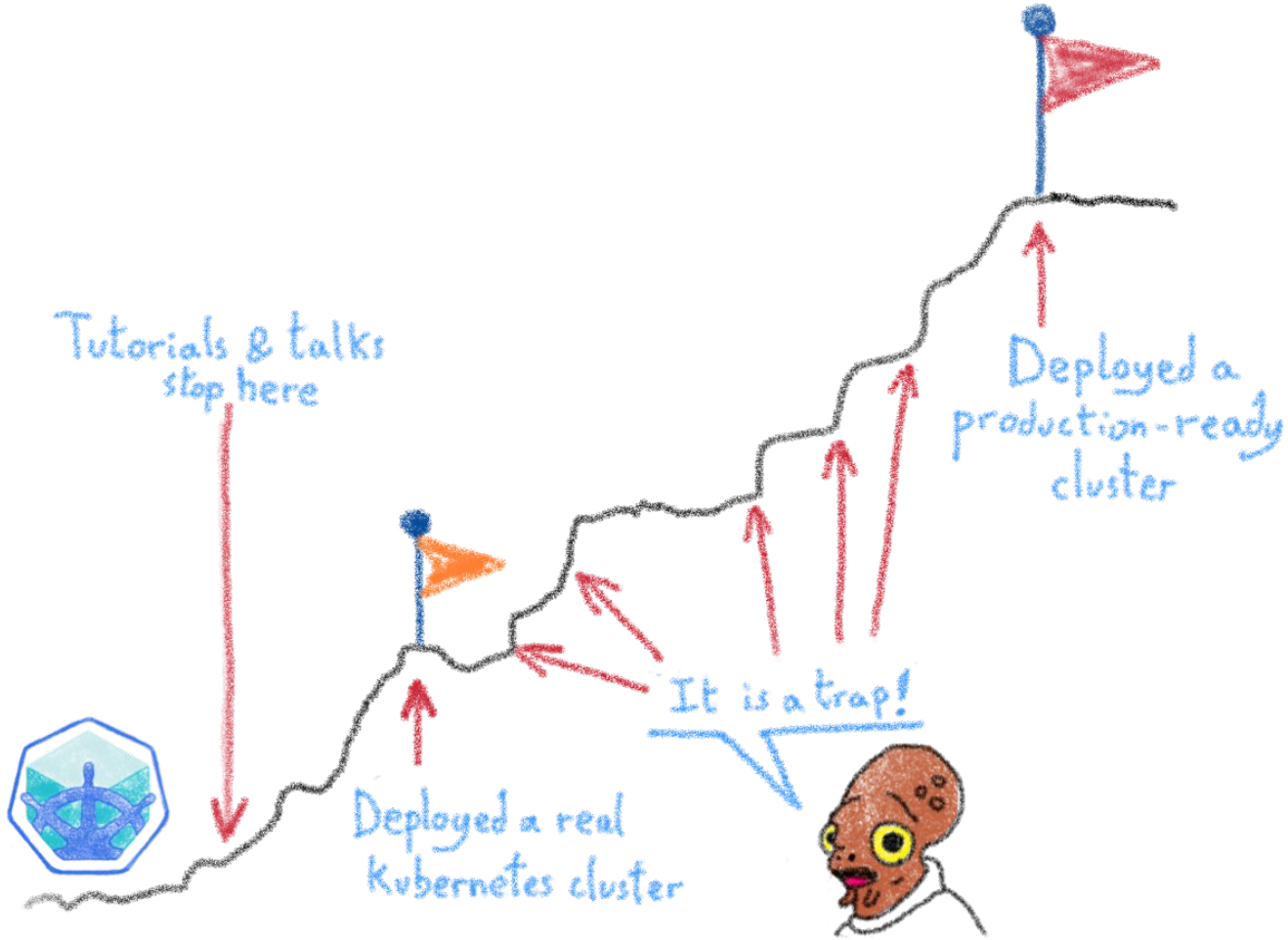
@LostInBrittany

Kubernetes can be wonderful

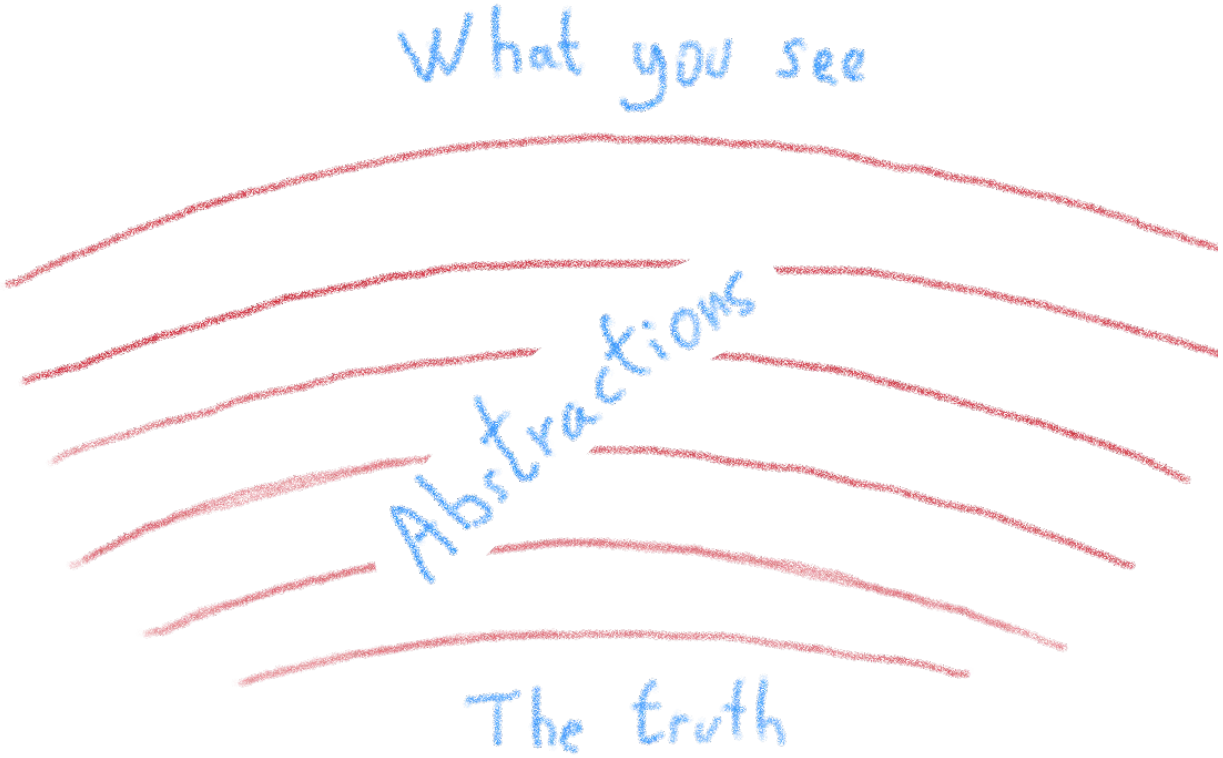


For both developers and devops

The journey from dev to production

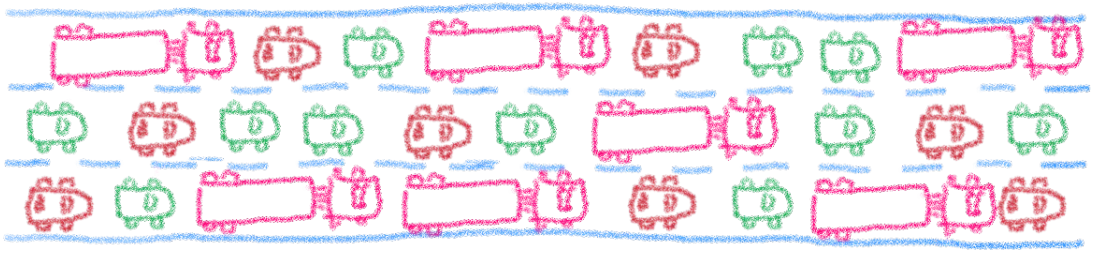


It's a complex technology



Lots of abstraction layers

Kubernetes networking is complex...



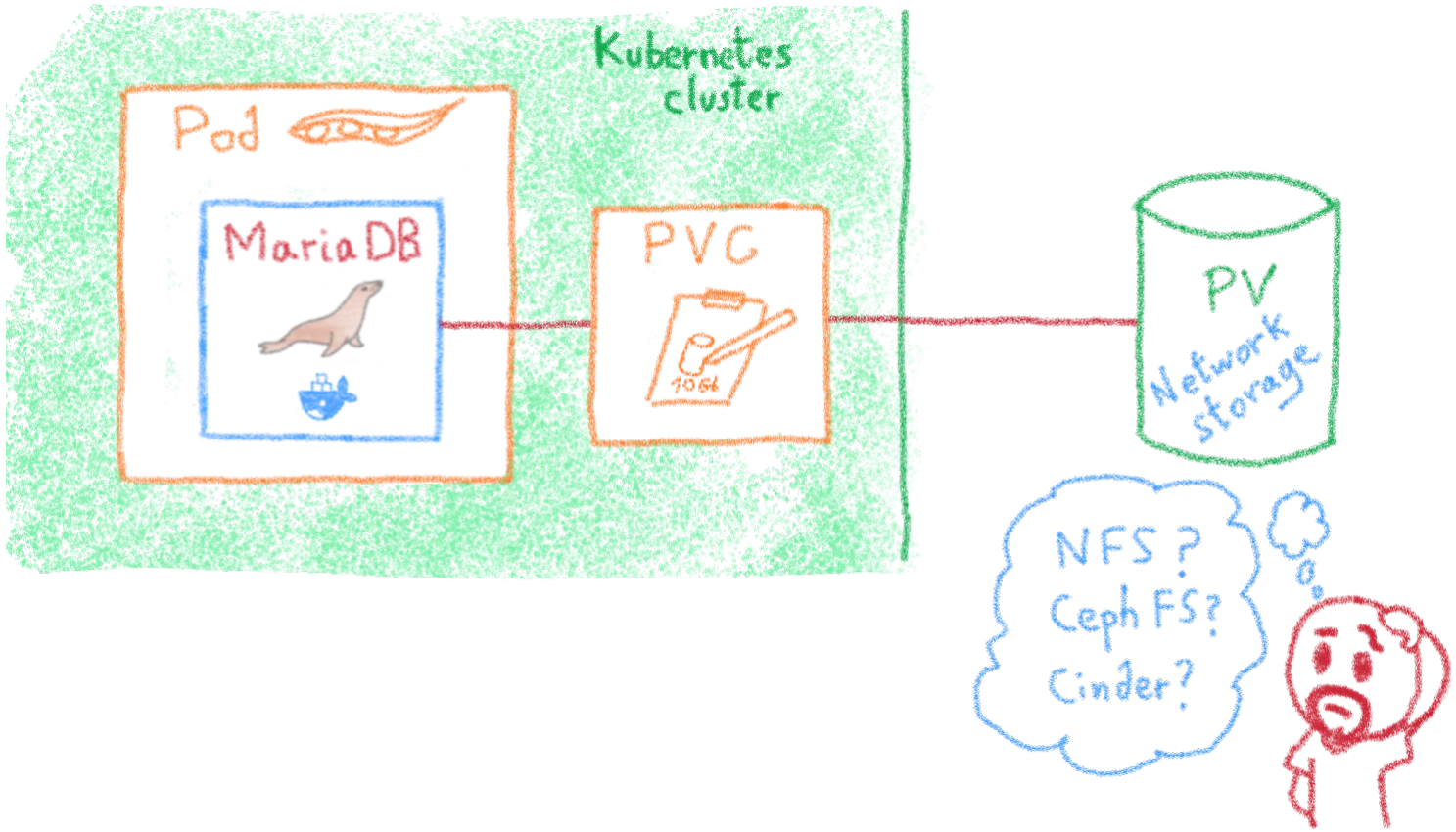
Network plugins (Flannel, Calico, Weave...)

- IPAM
- routing
- iptables
- crossnode networking

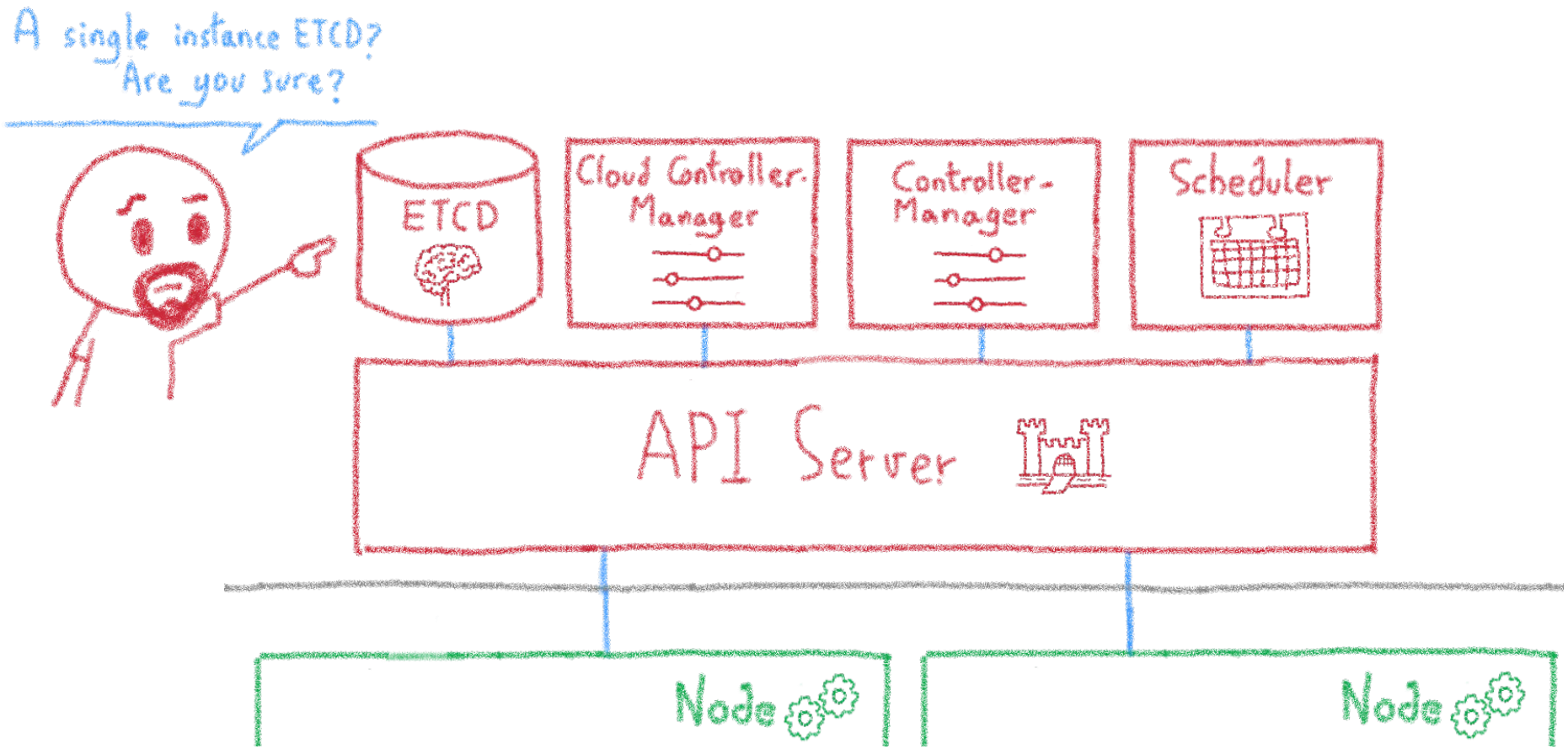
Cluster IP, NodePort, Ingress

Service Meshes, Istio

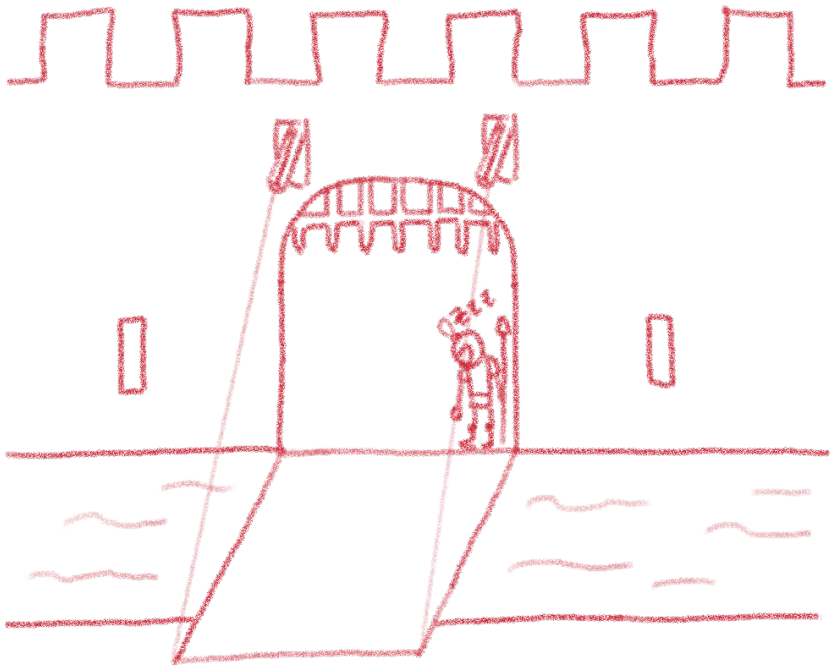
The storage dilemma



The ETCD vulnerability



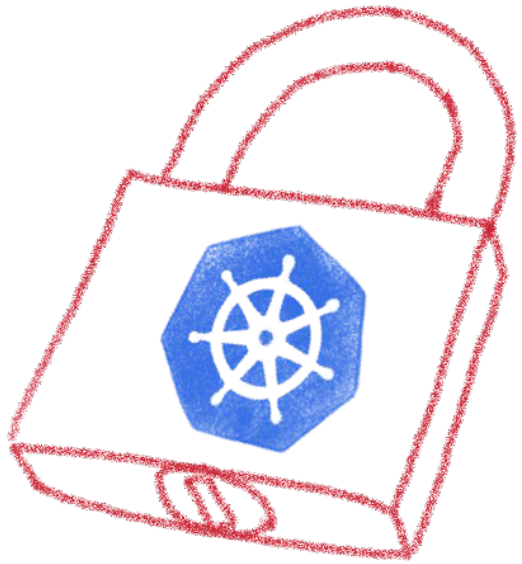
Kubernetes is insecure by design*



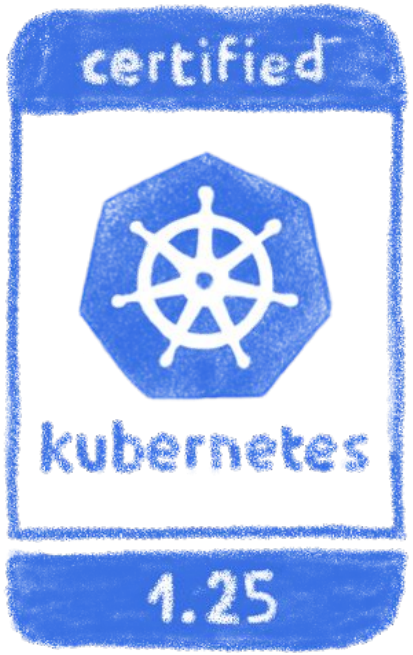
It's a feature, not a bug.
Up to K8s admin to secure it according to needs

Not everybody has the same security needs

Kubernetes allows to enforce security practices as needed



Always keep up to date

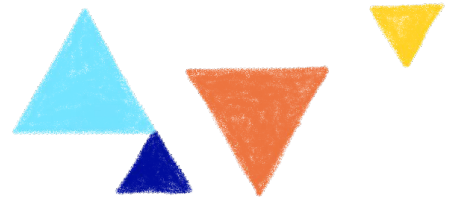


Both Kubernetes and plugins

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...

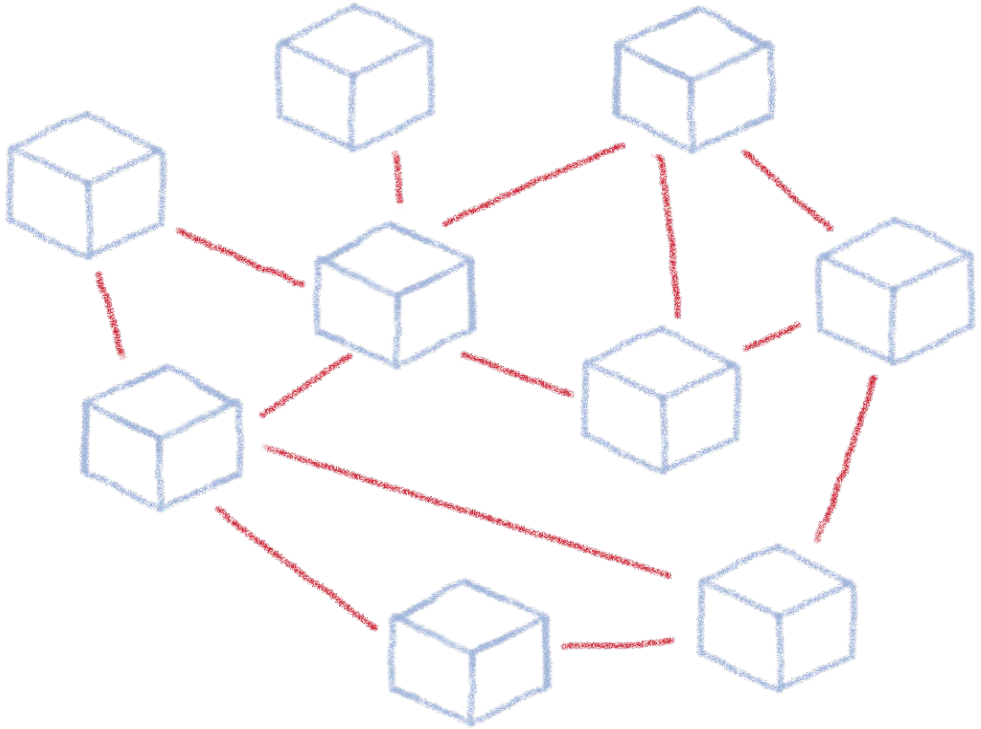
Remain attentive, don't get too confident



A managed Kubernetes

Because your company job is to use Kubernetes, not to operate it!

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

Different roles

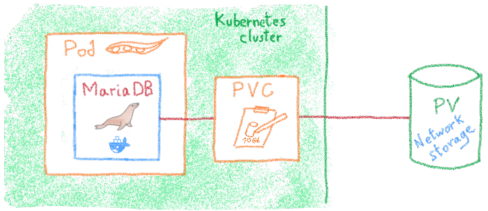
Cluster operator

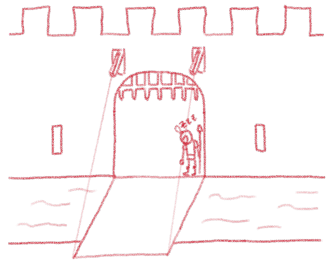
Cluster administrator

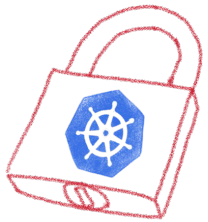
Developer

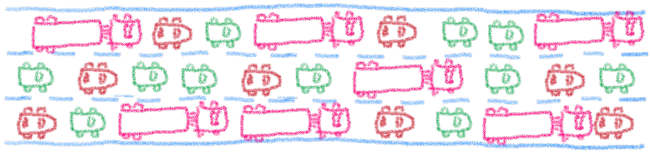
Each role asks for very different knowledge and skill sets

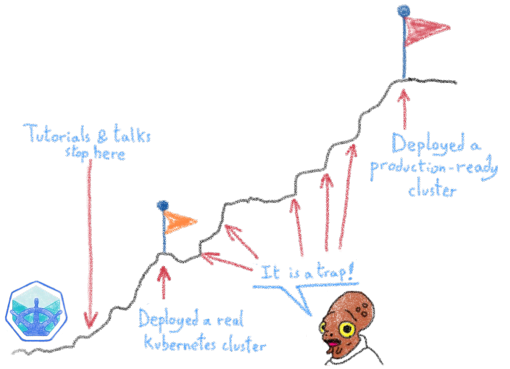
Operating a Kubernetes cluster is hard





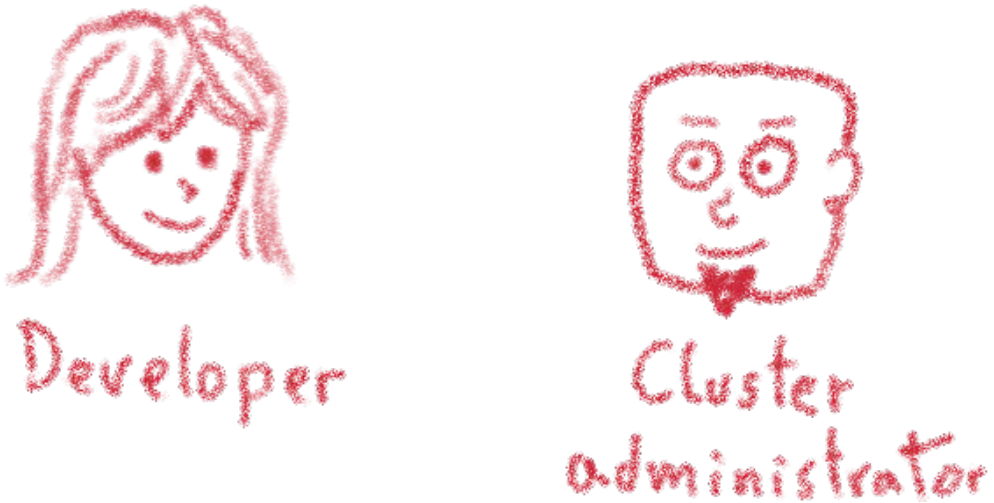






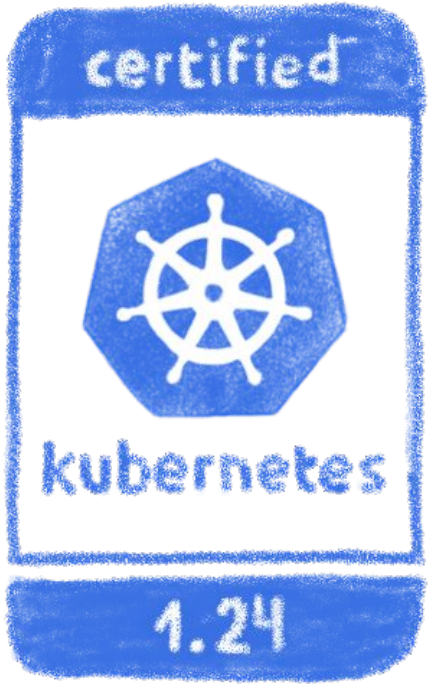
But we have a good news...

Most companies don't need to do it!



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

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

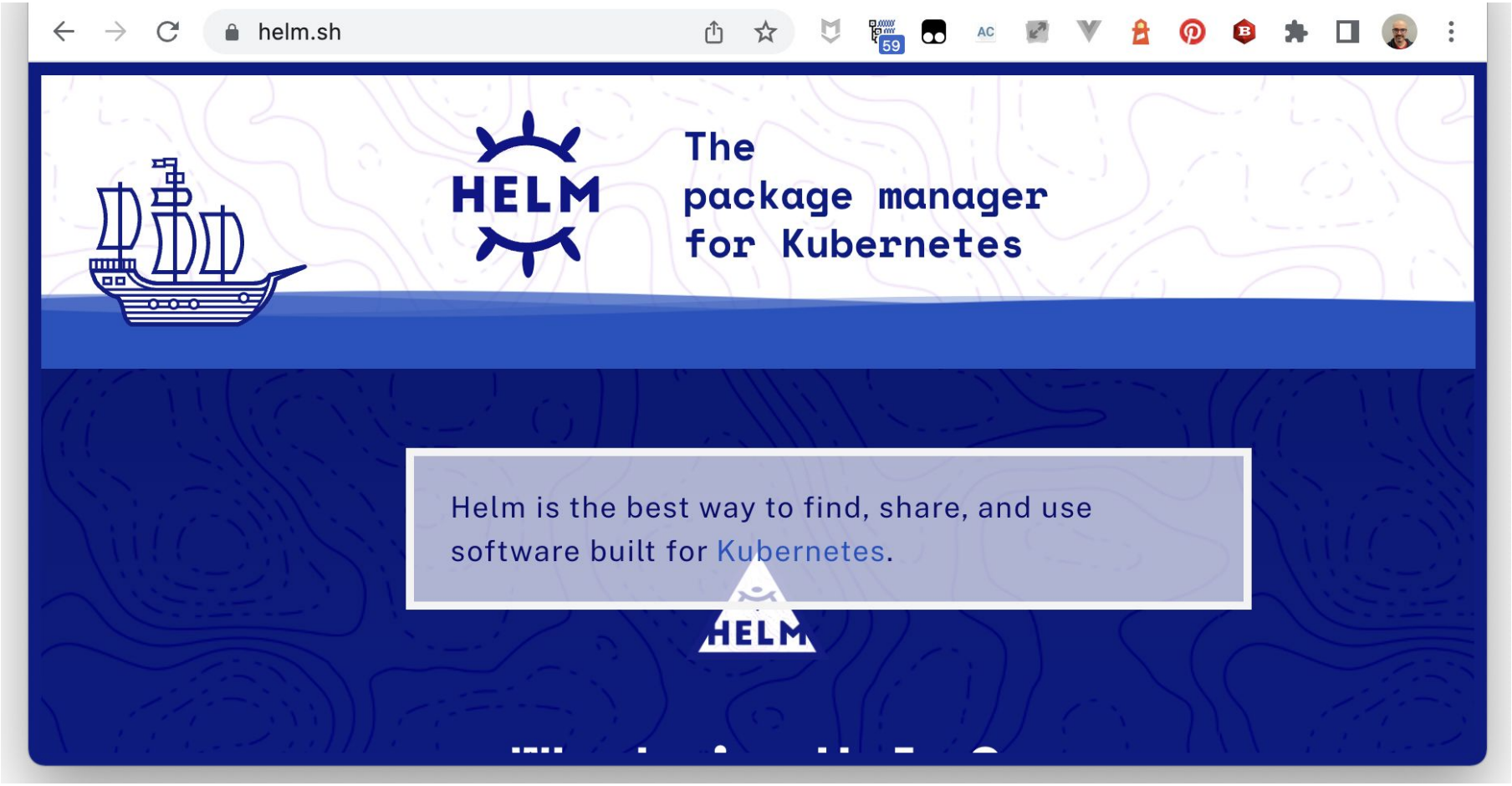


You get the cluster, the operator get the problems

Demo: A complete app - Wordpress

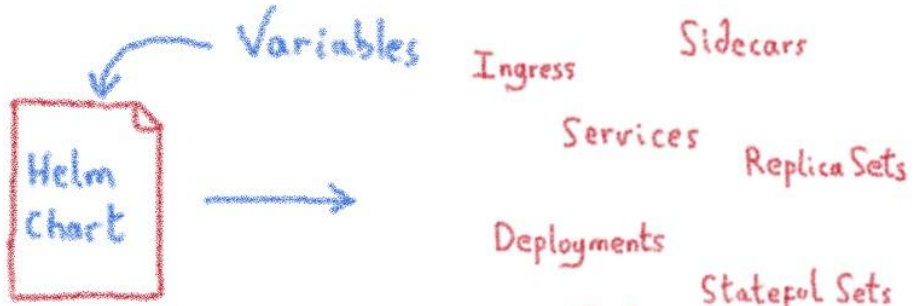
https://docs.ovh.com/gb/en/kubernetes/installing-wordpress/

Needed tools: helm



https://helm.sh/

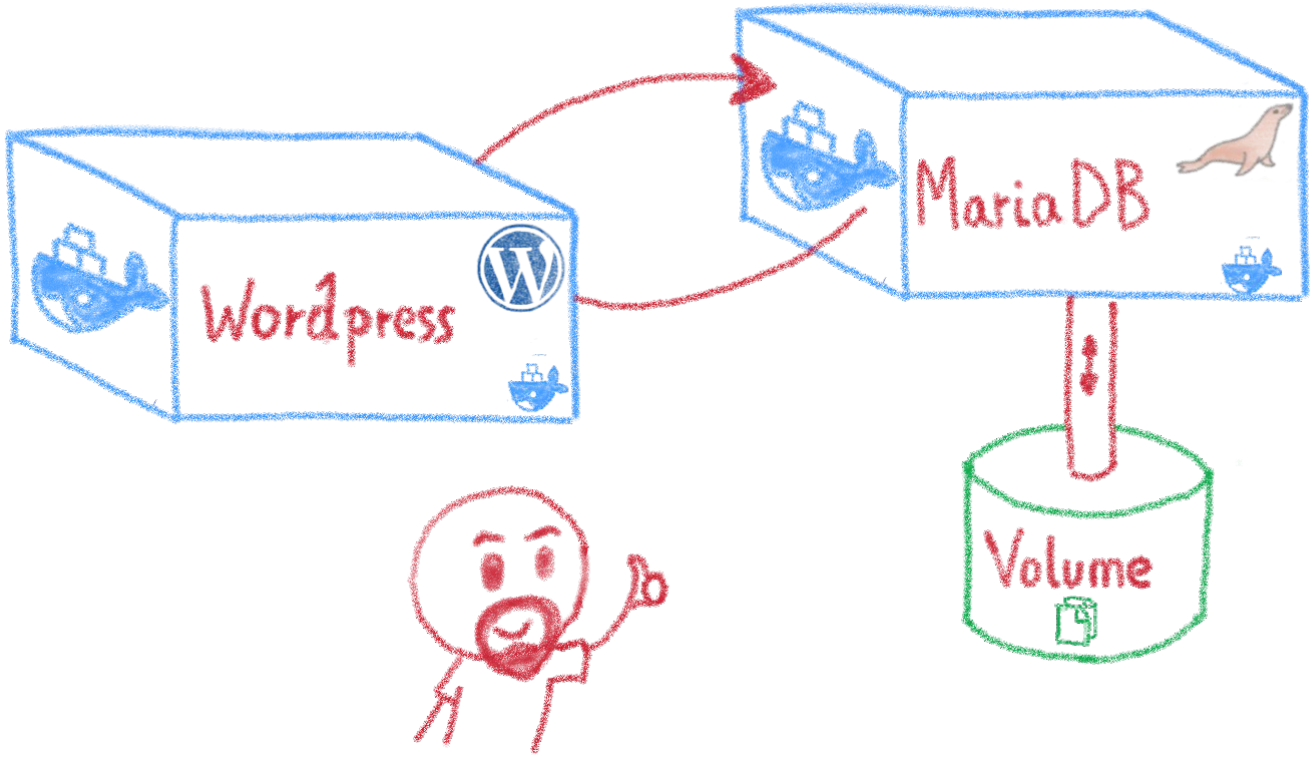
Helm: a package manager for K8s



- Manage complexity
- Easy upgrades

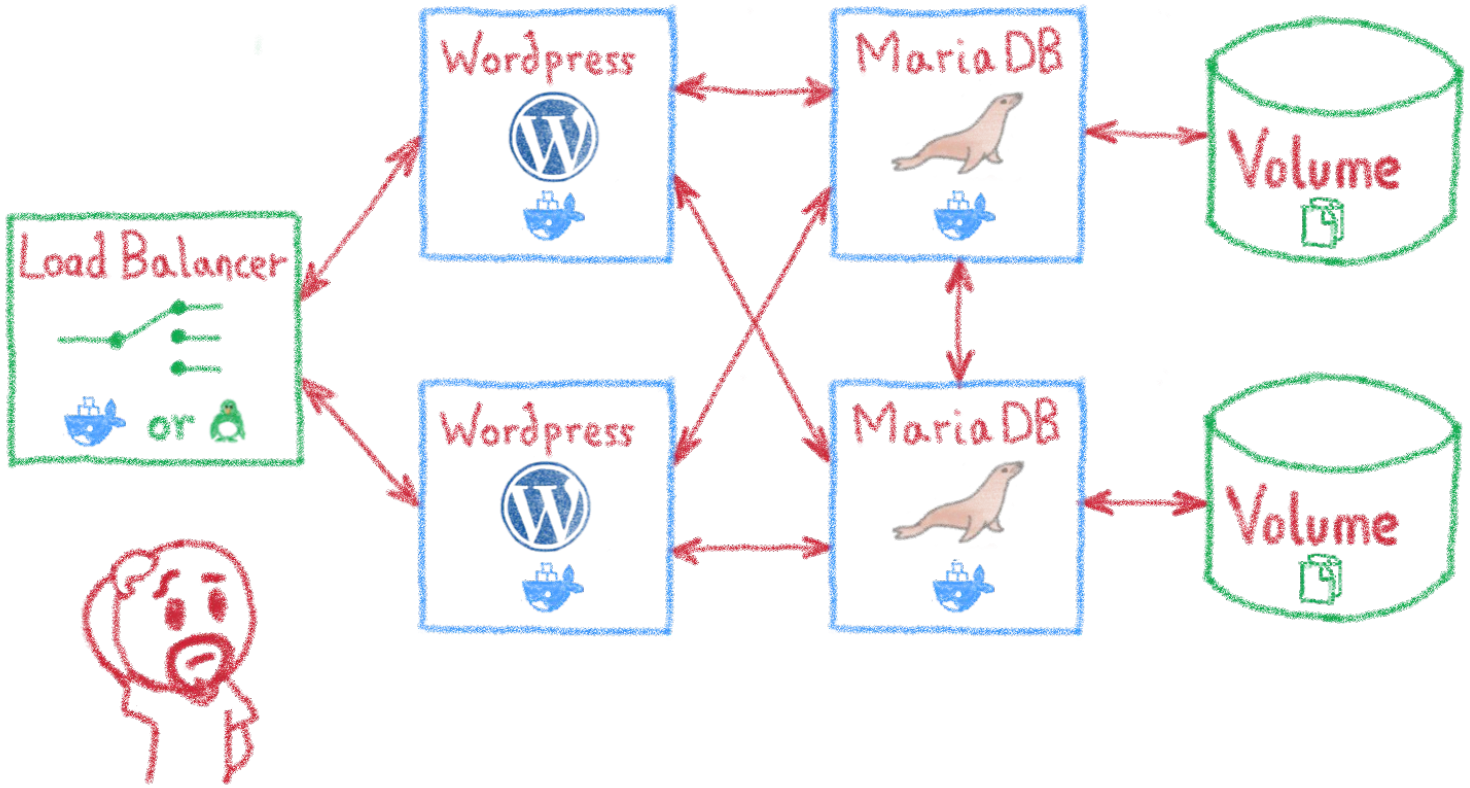
- Simple sharing
- Easy rollbacks

Wordpress is easy...



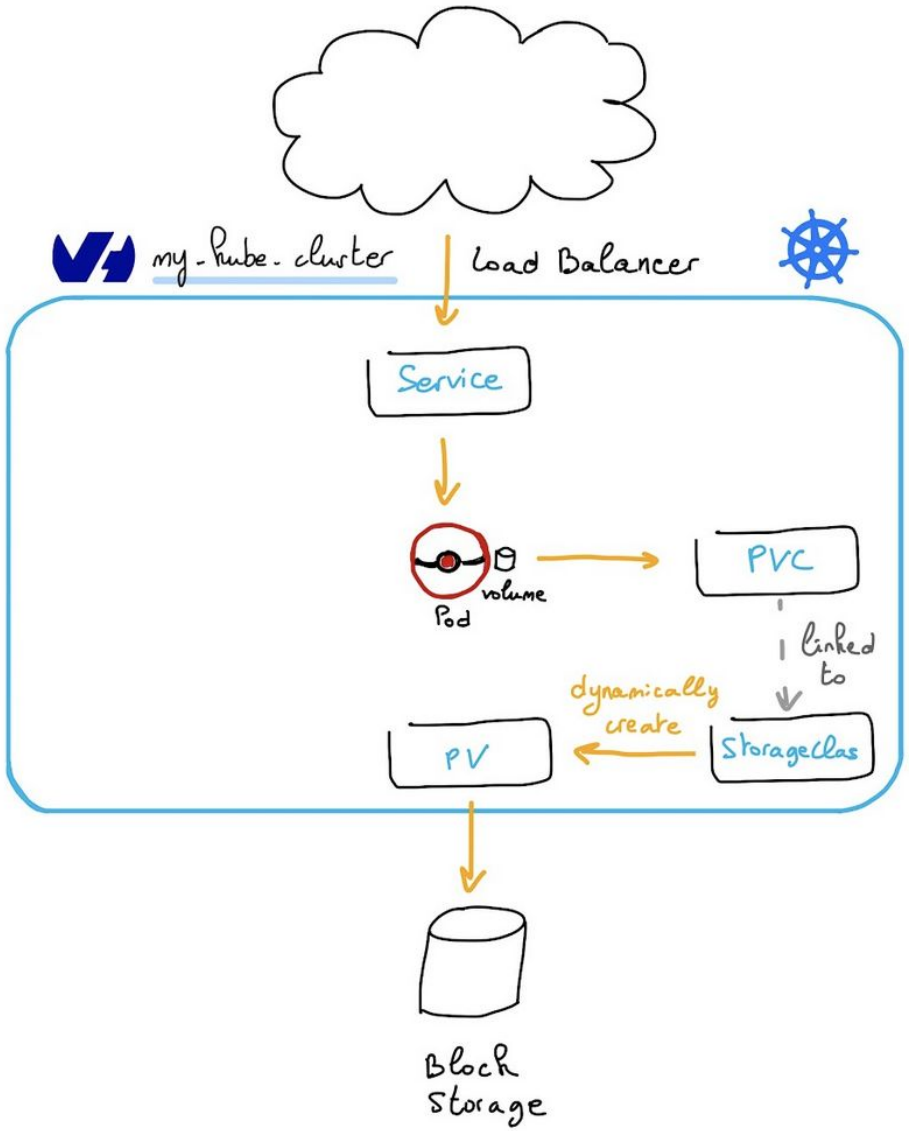
Two pods and a persistent volume

Yet is a complete app



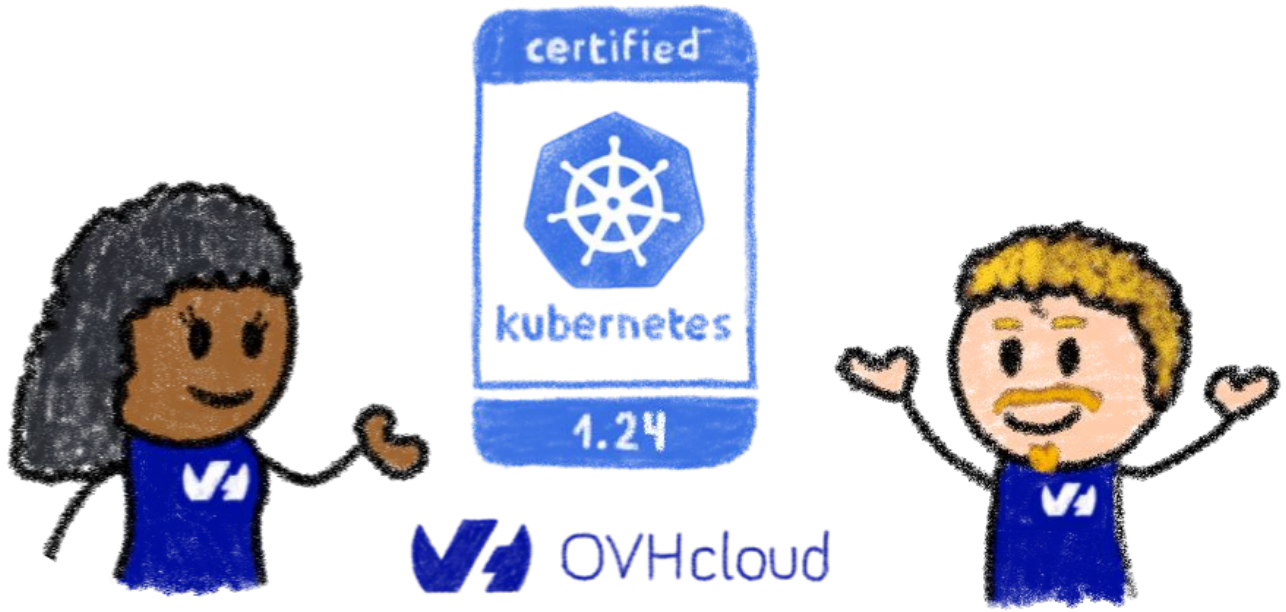
Specially when deployed in production context

Persistent storage in Kubernetes

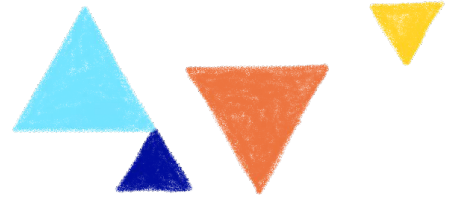


OVHcloud Managed Kubernetes

Why would you choose ours?



Certified Kubernetes platform



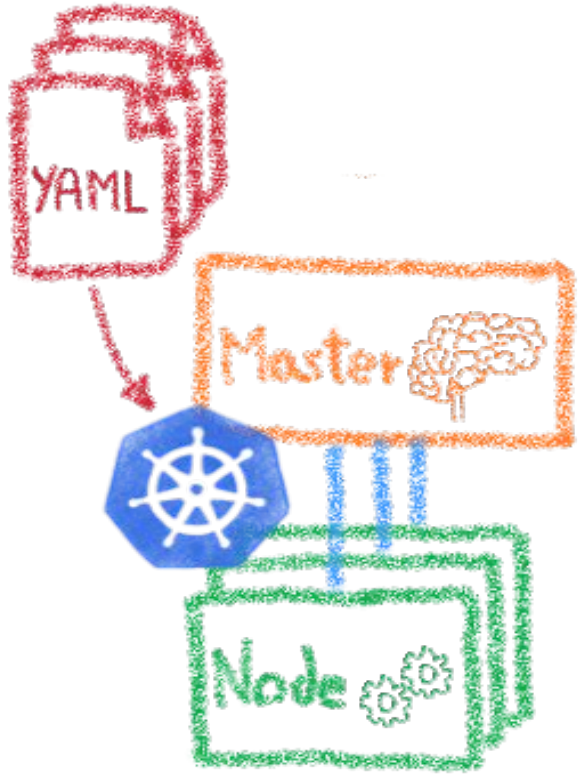
OVHcloud

Managed Kubernetes
certified Kubernetes 1.24

OVHcloud Managed Private Registry

HARBOR

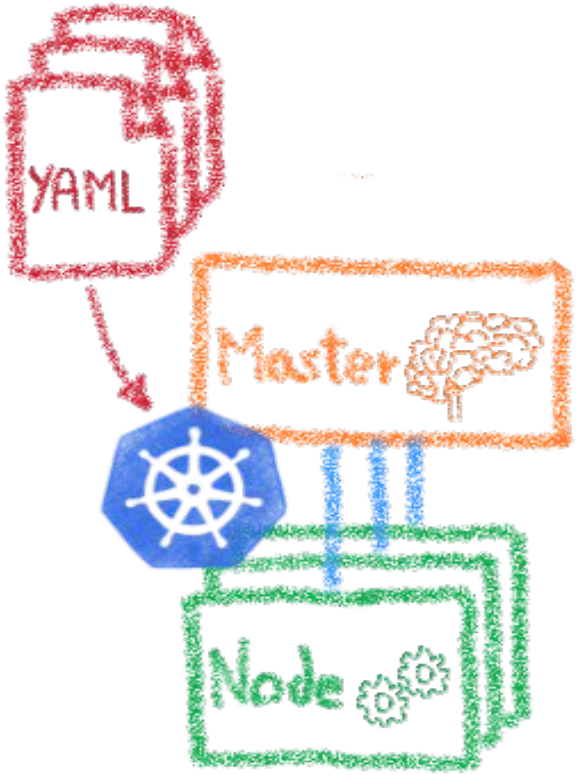
Node Pools



Users can define node pools

controlled from inside Kubernetes

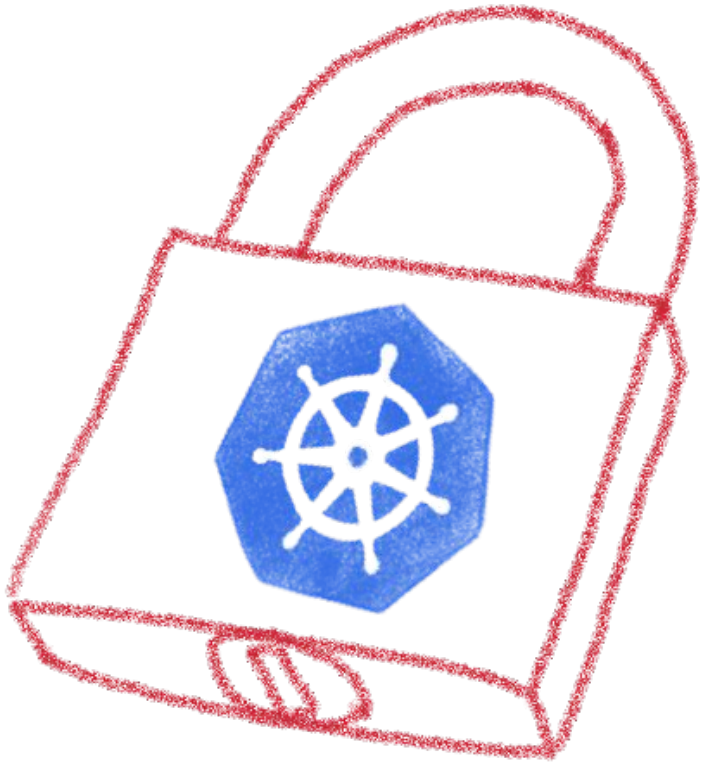
Autoscaling



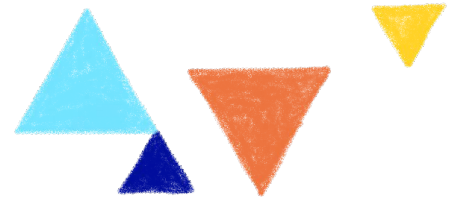
Based on node pools

New instances are spawned or released based on load

Kubernetes in a private network



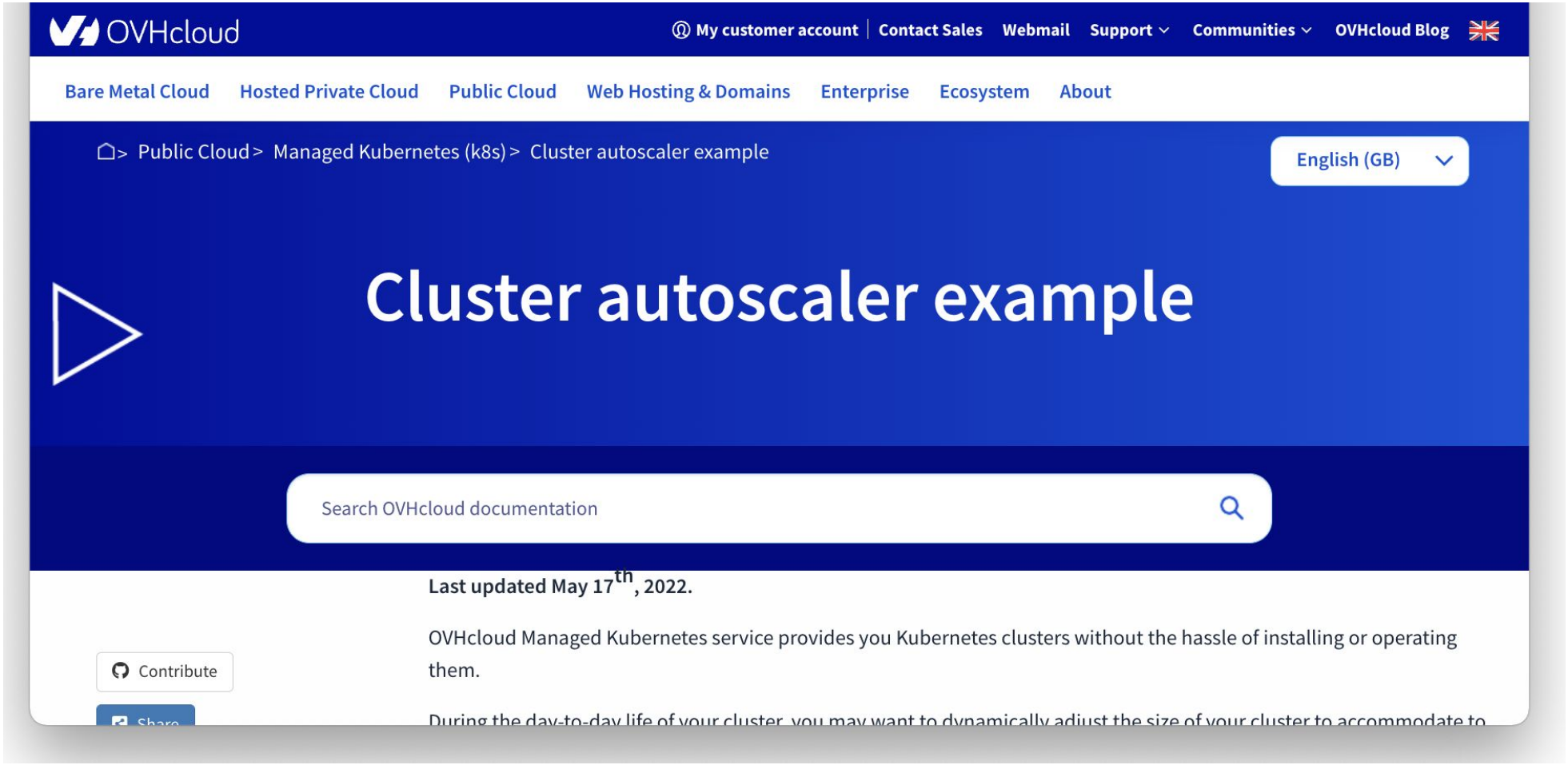
Other features



- Healthcare HDS 1 conformity
- ISO 27001/27701/27017/27018 conformity
- Terraform provider
- Control plane audit logs
- API server IP restrictions
- …

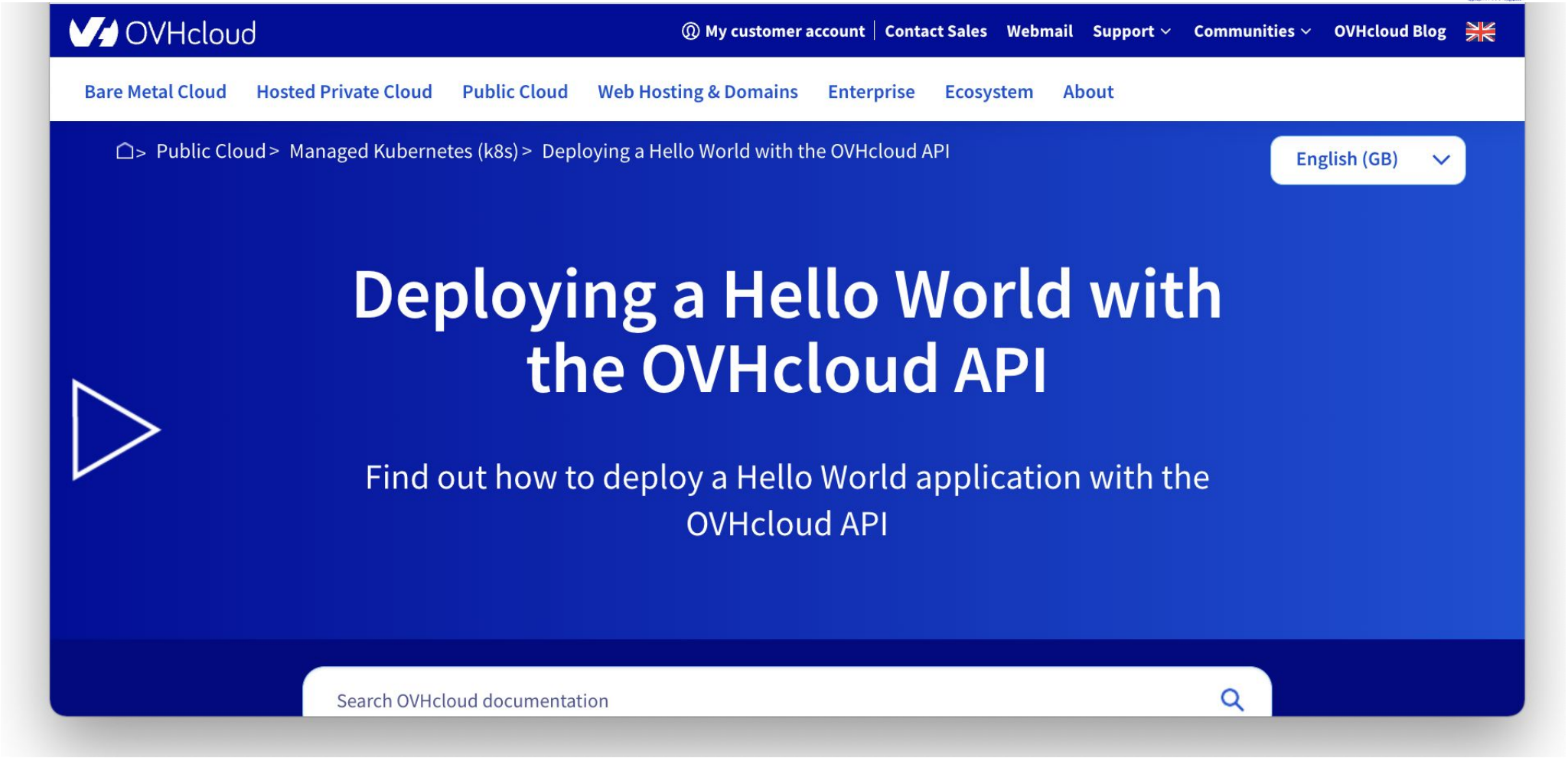
https://github.com/ovh/public-cloud-roadmap/projects/1

Demo: cluster auto-scaling

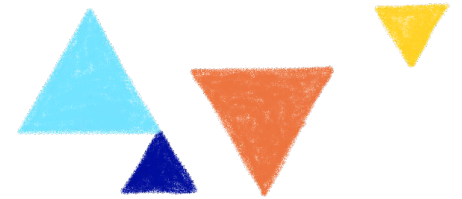


https://docs.ovh.com/gb/en/kubernetes/cluster-autoscaler-example/

Demo: Working with OVHcloud API

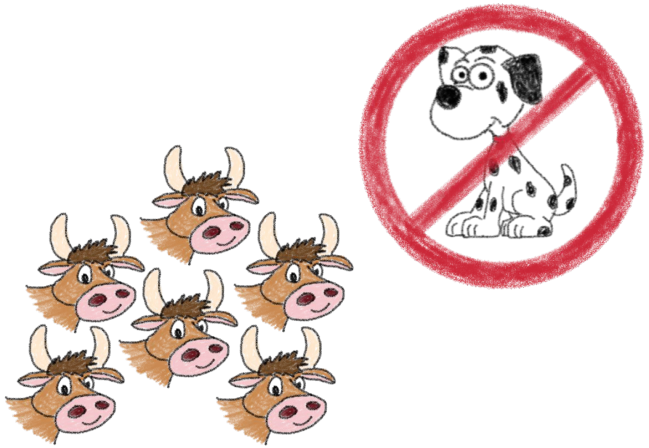


https://docs.ovh.com/gb/en/kubernetes/deploying-hello-world-ovh-api/



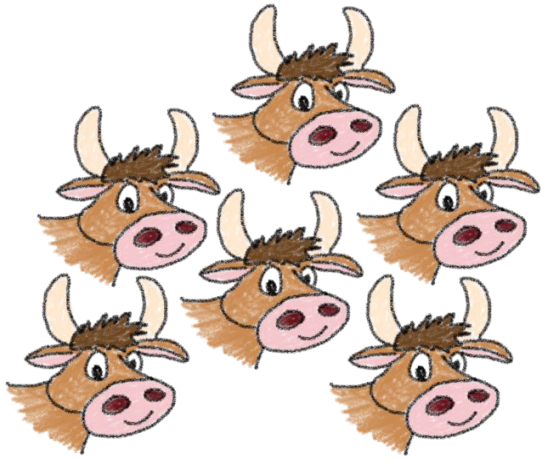
Infrastructure as Code

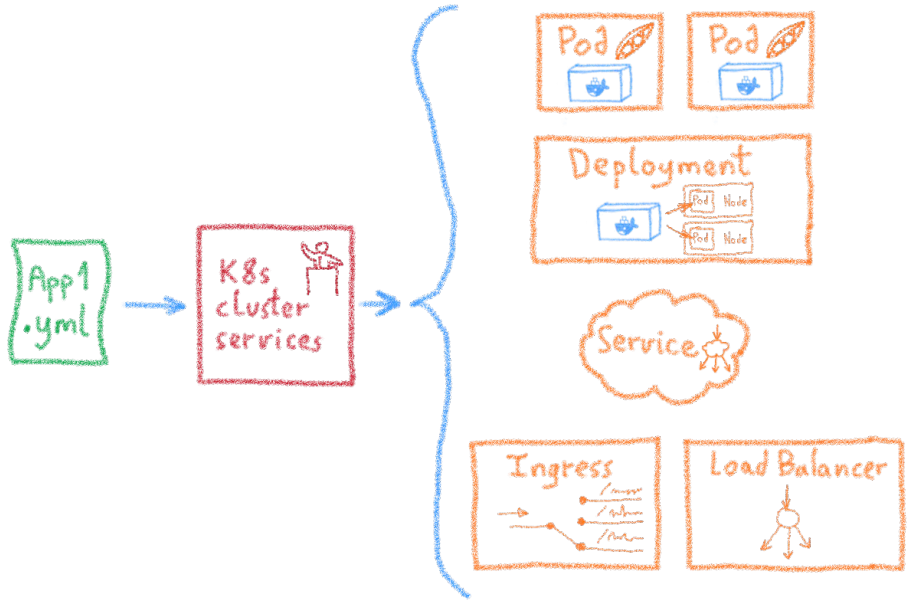
The perfect companion to a cloud



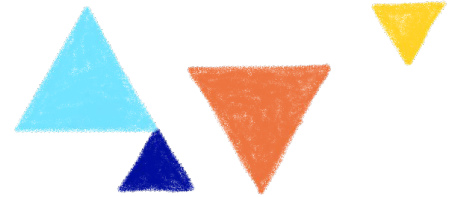
HashiCorp
Terraform

Infrastructure as Code (IaC)





Types of IaC



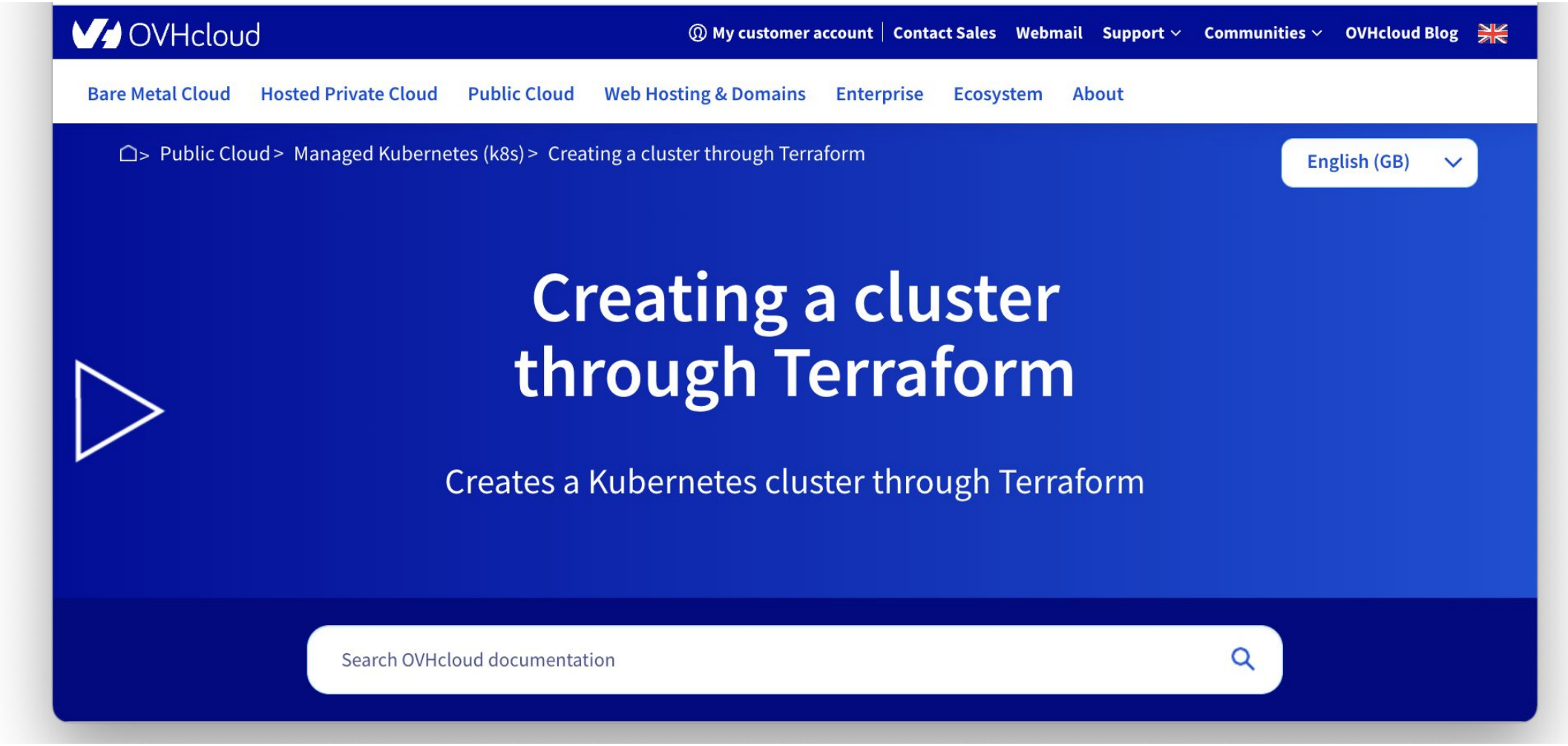
Imperative - Instructions to follow step by step

Declarative - Desired state description

Environment Aware - Intelligent desired state management

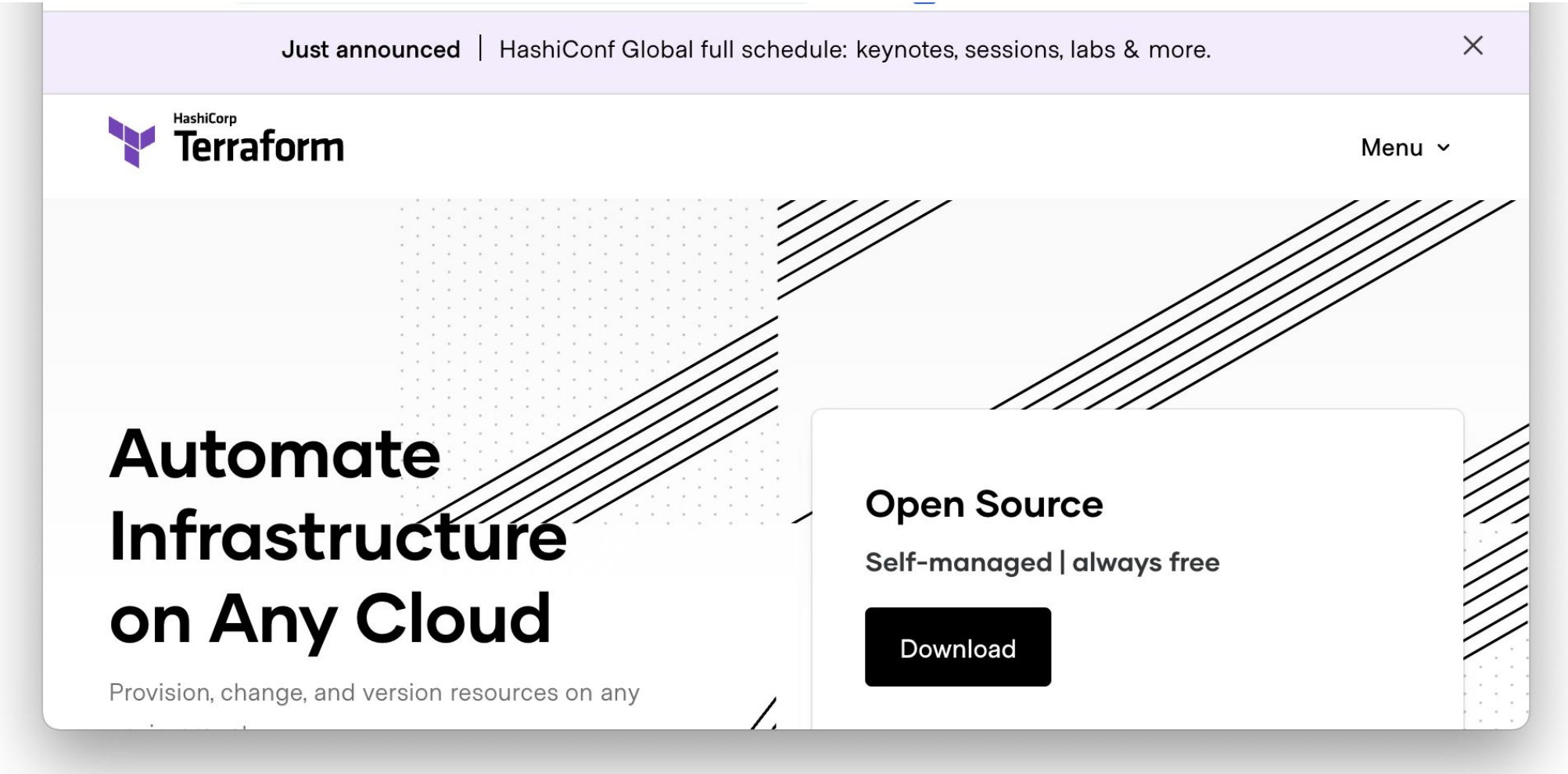
IaC tools

Demo: Using Terraform

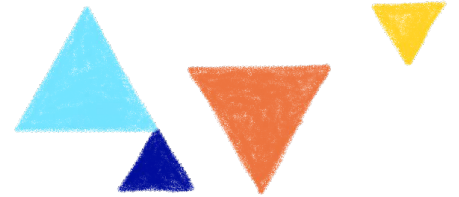


https://docs.ovh.com/gb/en/kubernetes/creating-a-cluster-through-terraform/

Needed tools: terraform

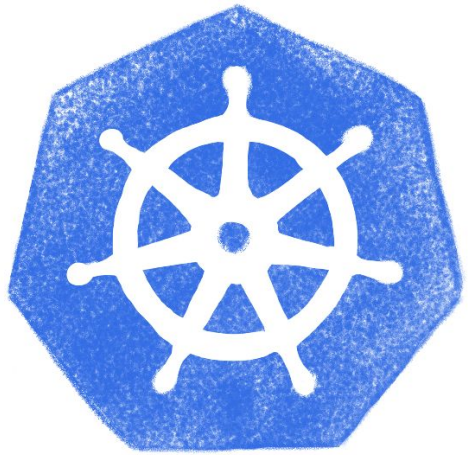


https://www.terraform.io/



Kubernetes Operators

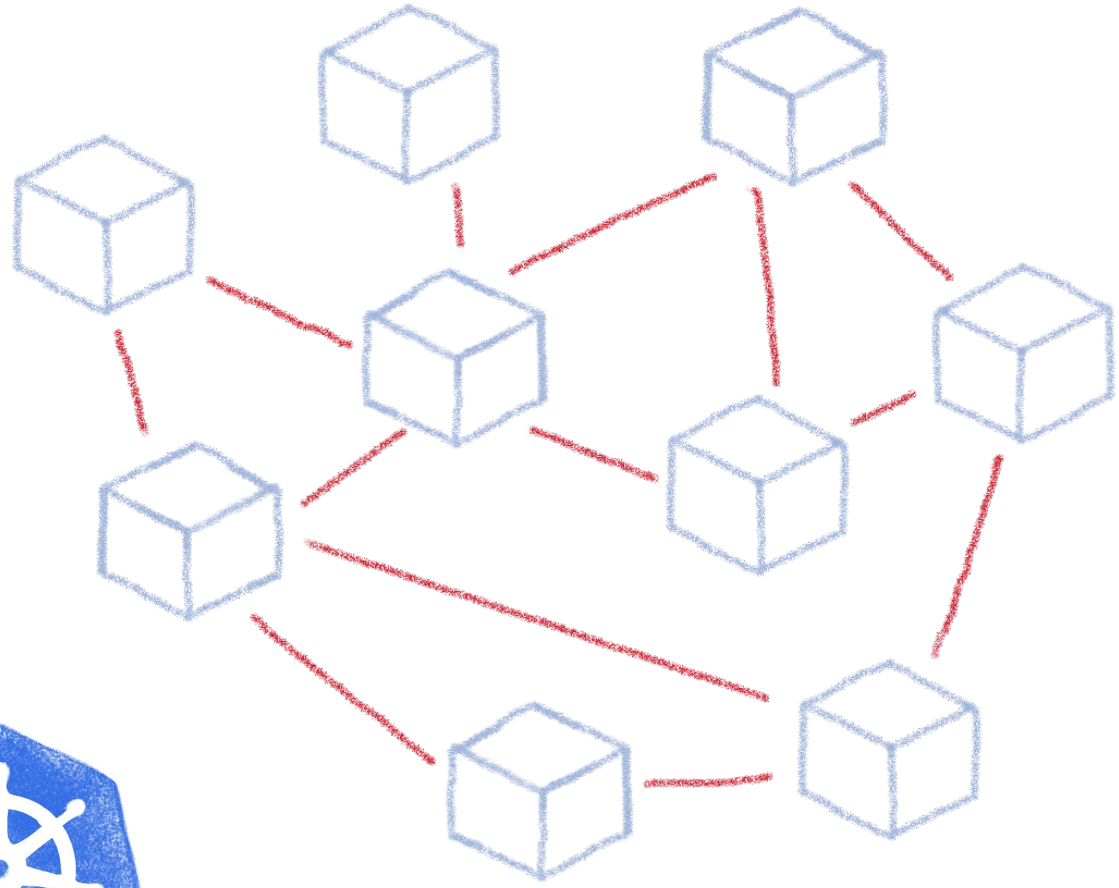
Helping to tame the complexity of K8s Ops



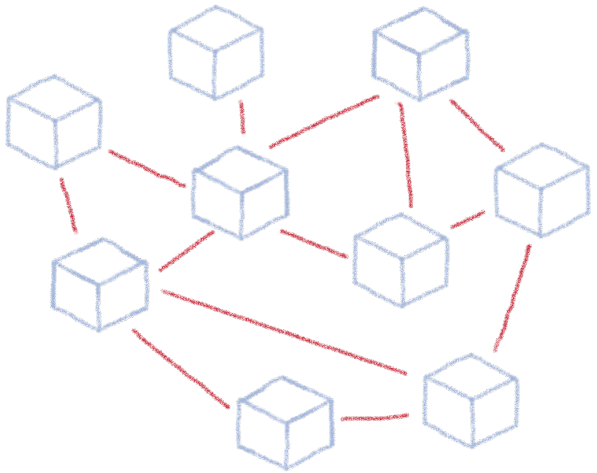
OVHcloud

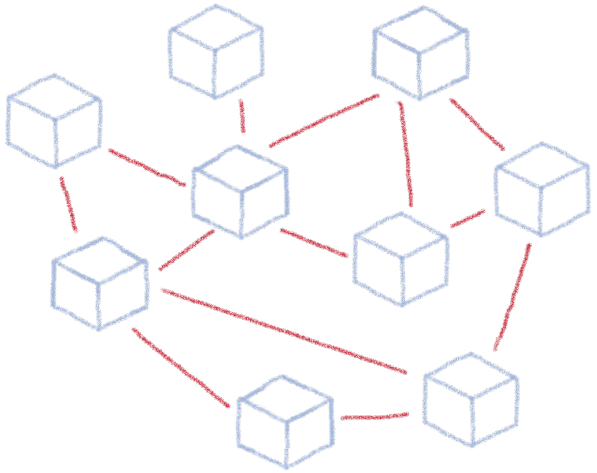
@LostInBrittany

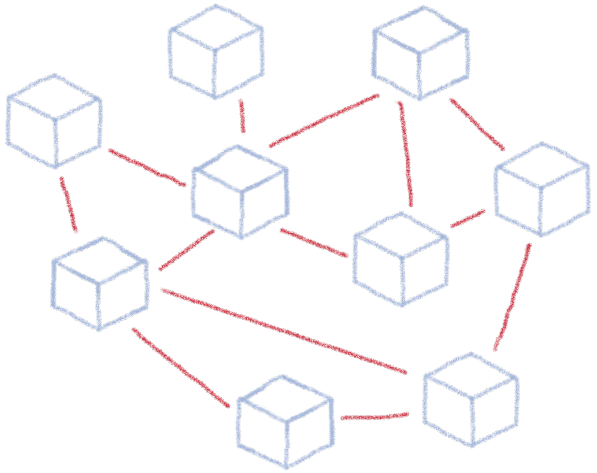
Taming microservices with Kubernetes



What about complex deployments







Ingress

Services

Deployments

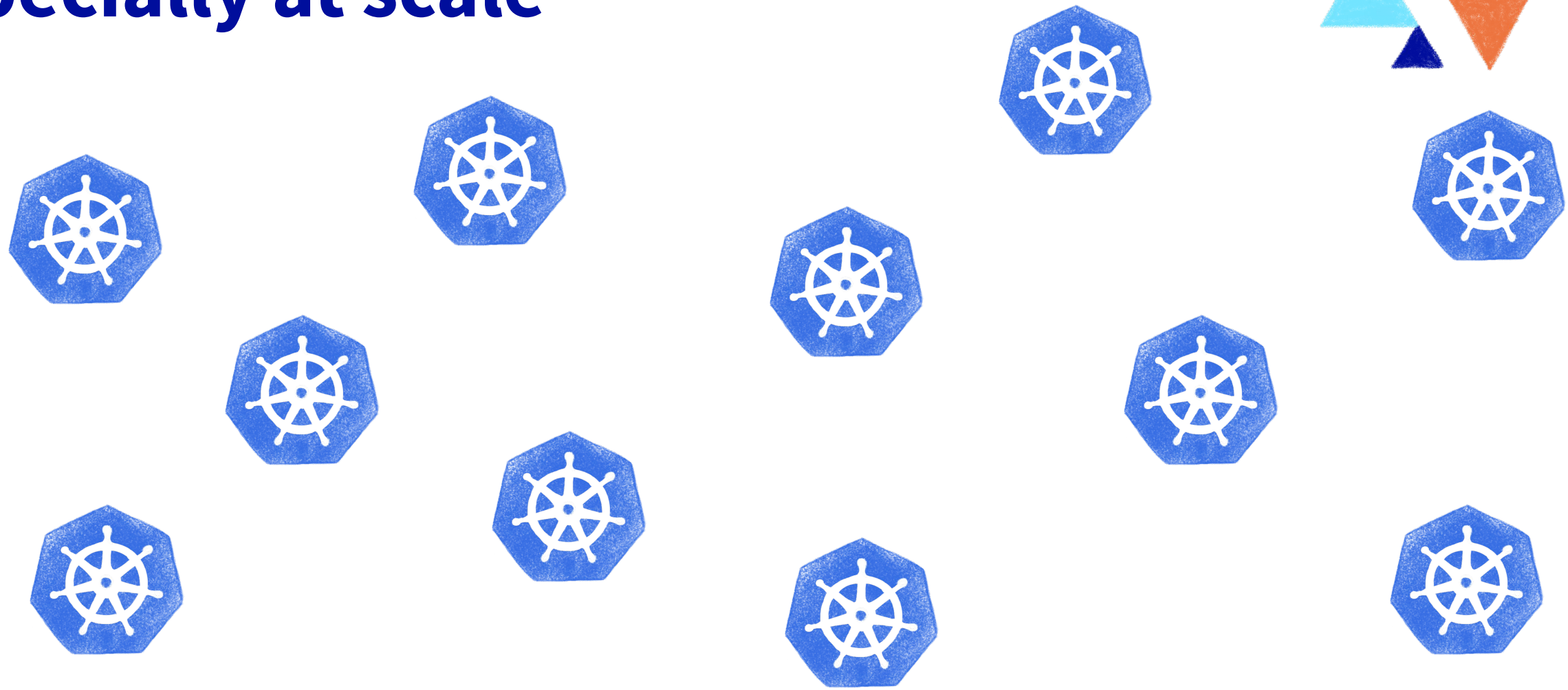
Pods

Sidecars

Replica Sets

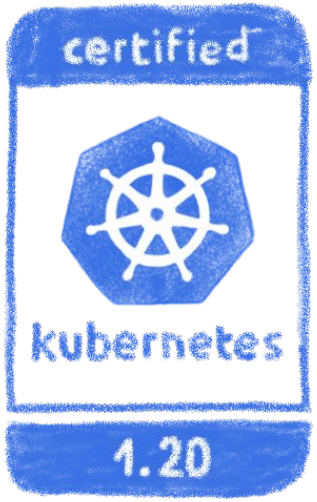
Statepul Sets

Specially at scale



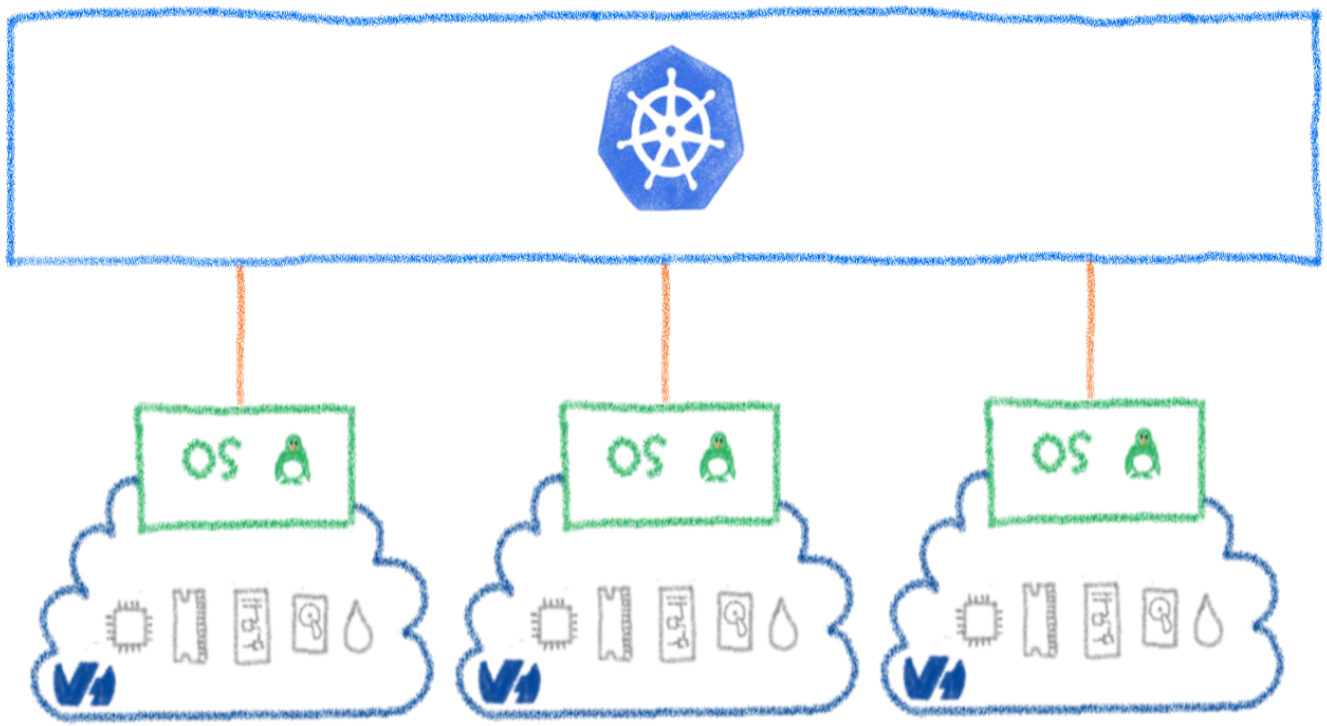
Lots of clusters with lots and lots of deployments

That's just our case

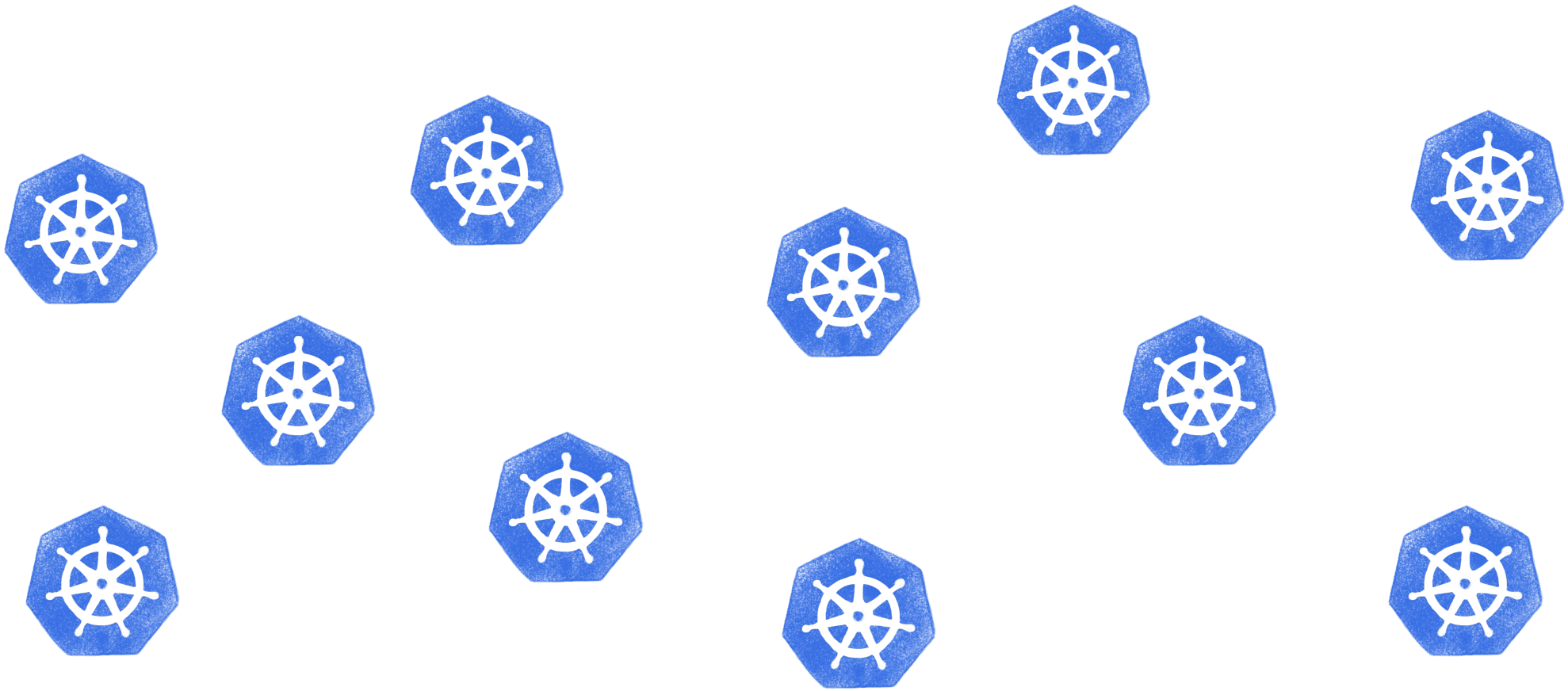


We both use Kubernetes and
operate a Managed Kubernetes platform

Built over our Openstack based Public Cloud

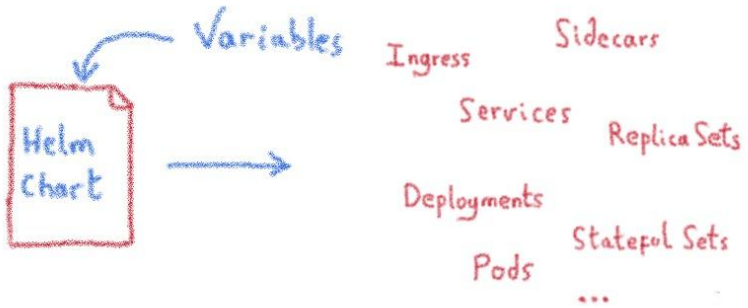


We need to tame the complexity

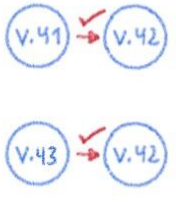


Taming the complexity

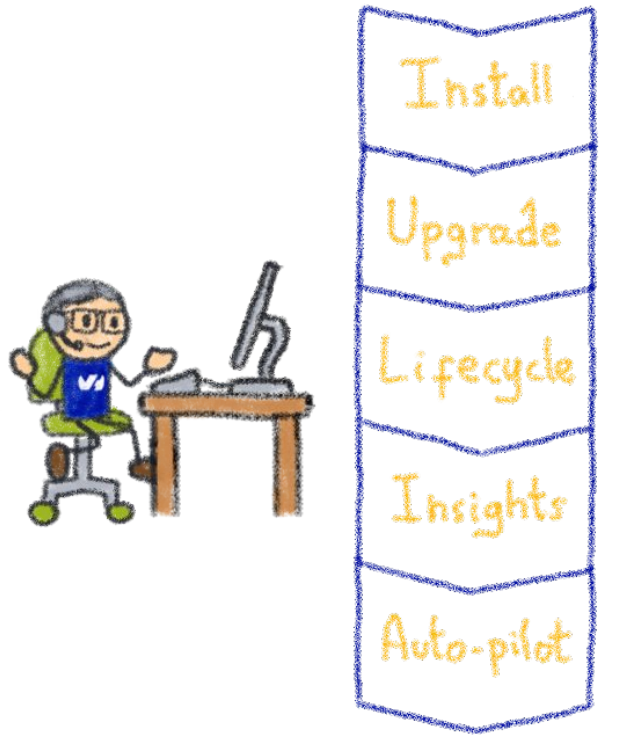
A package manager for Kubernetes



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

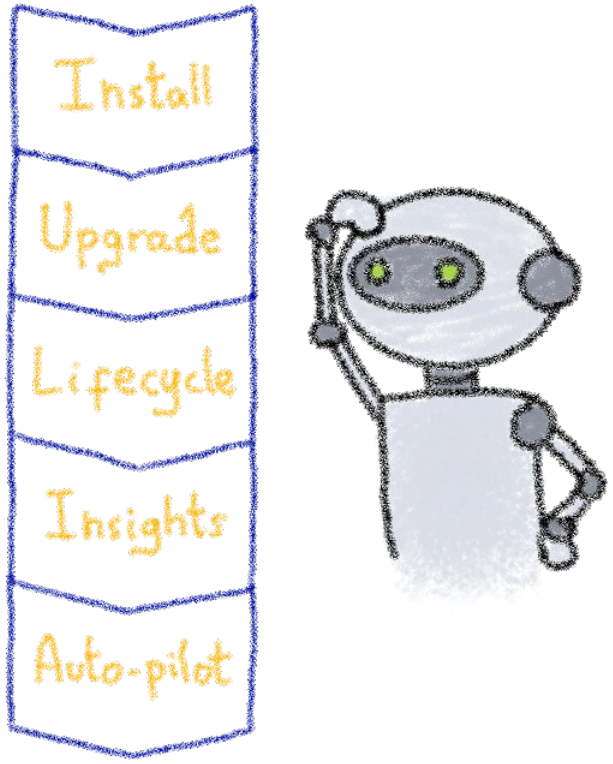


Helm Charts are configuration



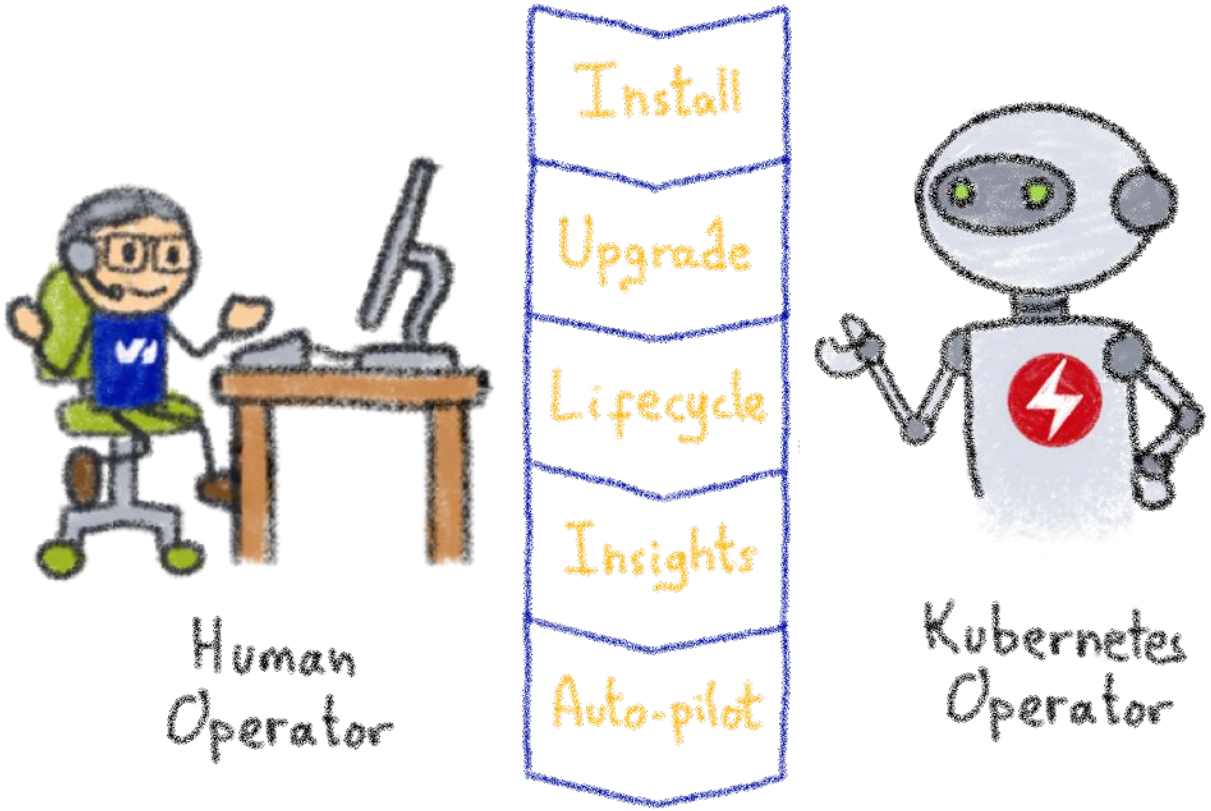
Operating is more than installs & upgrades

Kubernetes is about automation



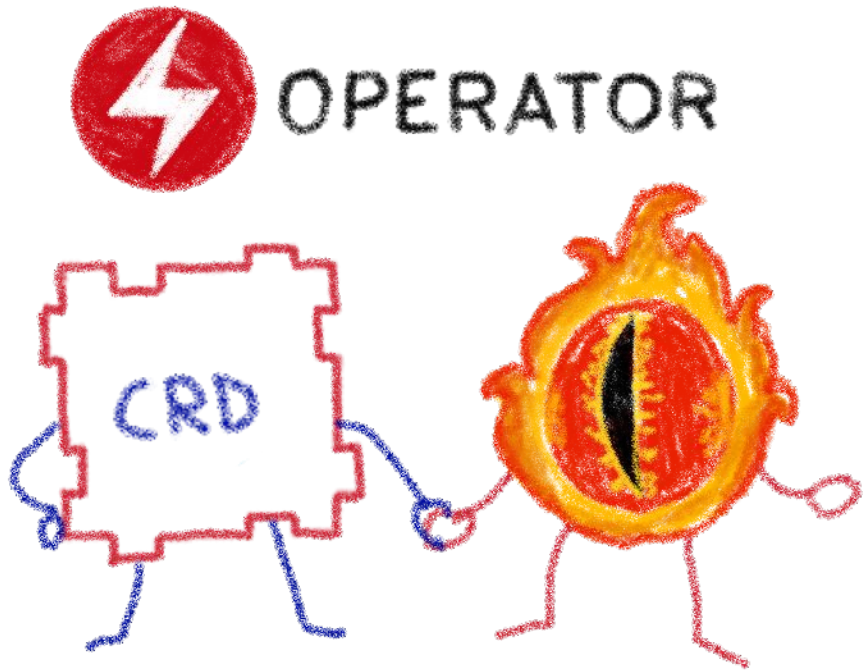
How about automating human operators?

Kubernetes Operators

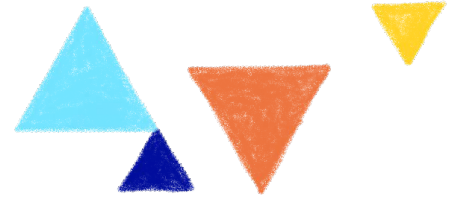


A Kubernetes version of the human operator

Building operators



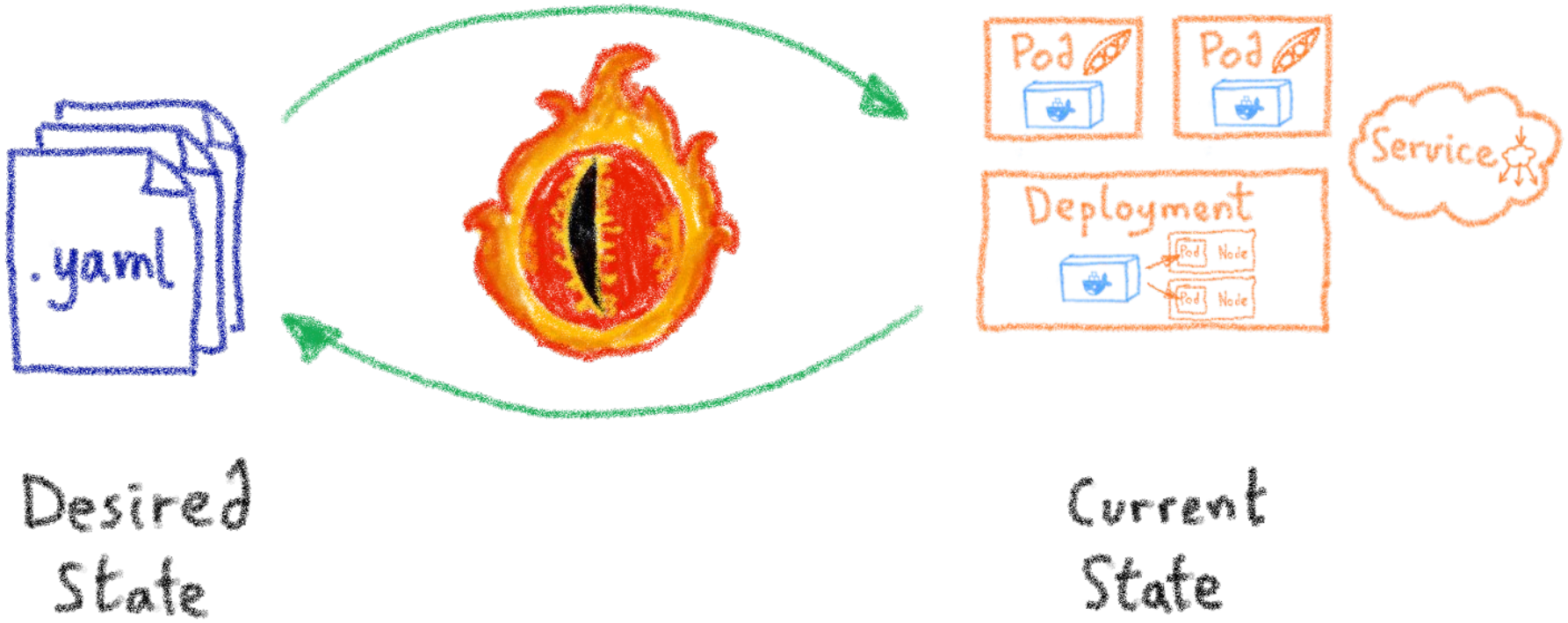
Basic K8s elements: Controllers and Custom Resources



Kubernetes Controllers

Keeping an eye on the resources

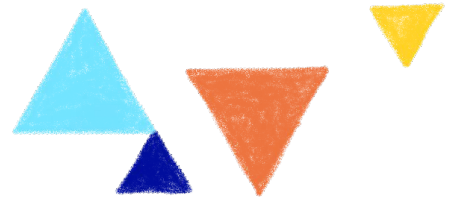
A control loop



They watch the state of the cluster,
and make or request changes where needed

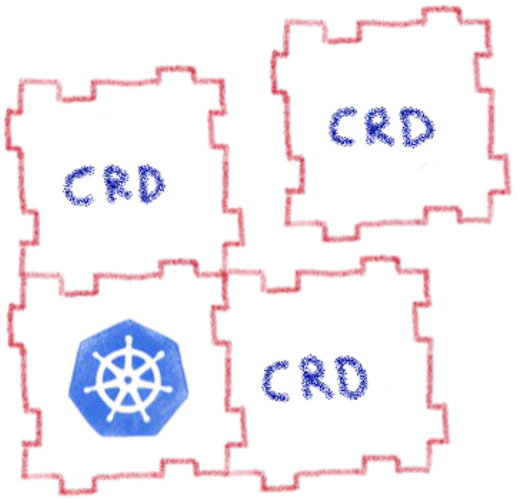
A reconcile loop

Strives to reconcile current state and desired state

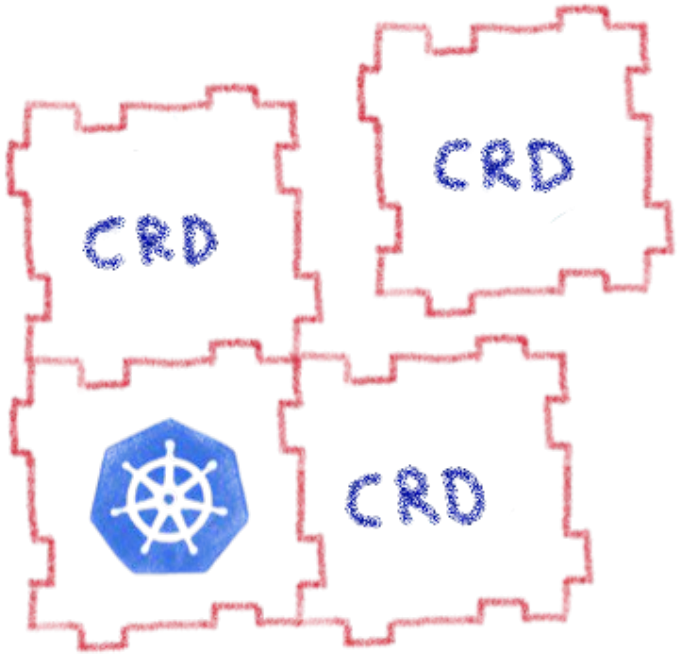


Custom Resource Definitions

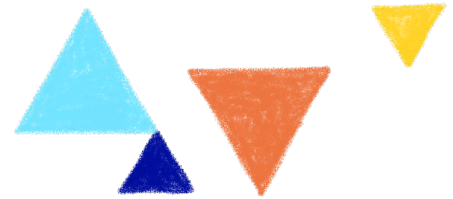
Extending Kubernetes API



Extending Kubernetes API



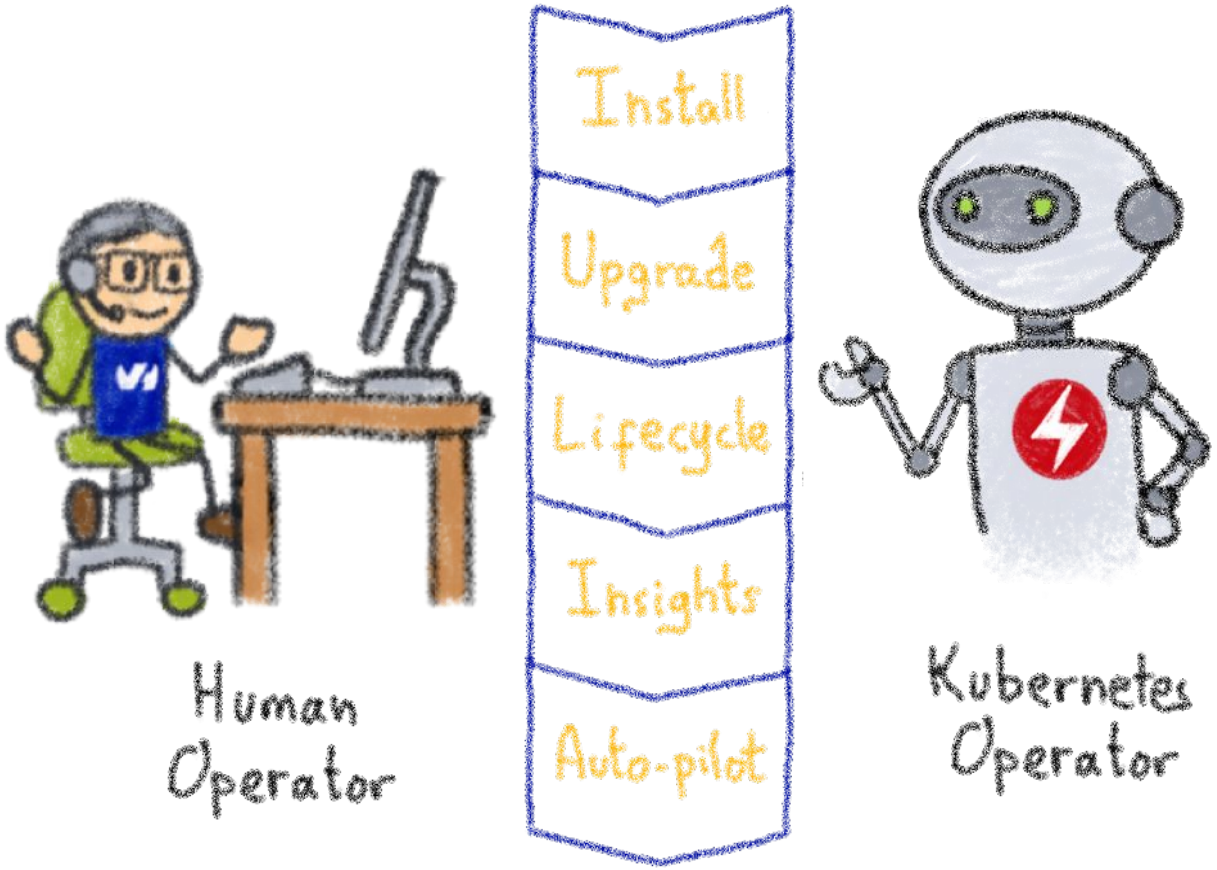
By defining new types of resources



Kubernetes Operator

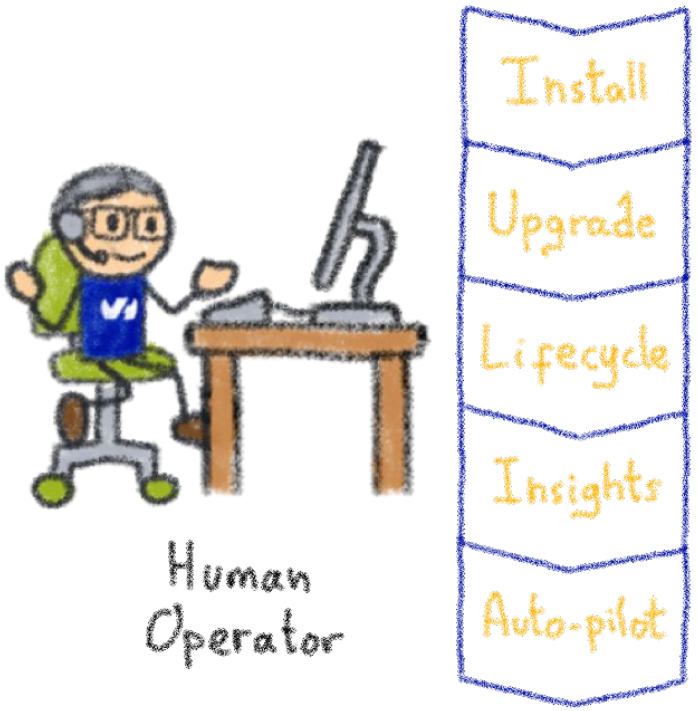
Automating operations

What's a Kubernetes Operator?



An Operator represents human operational knowledge in software to reliably manage an application

Example: databases

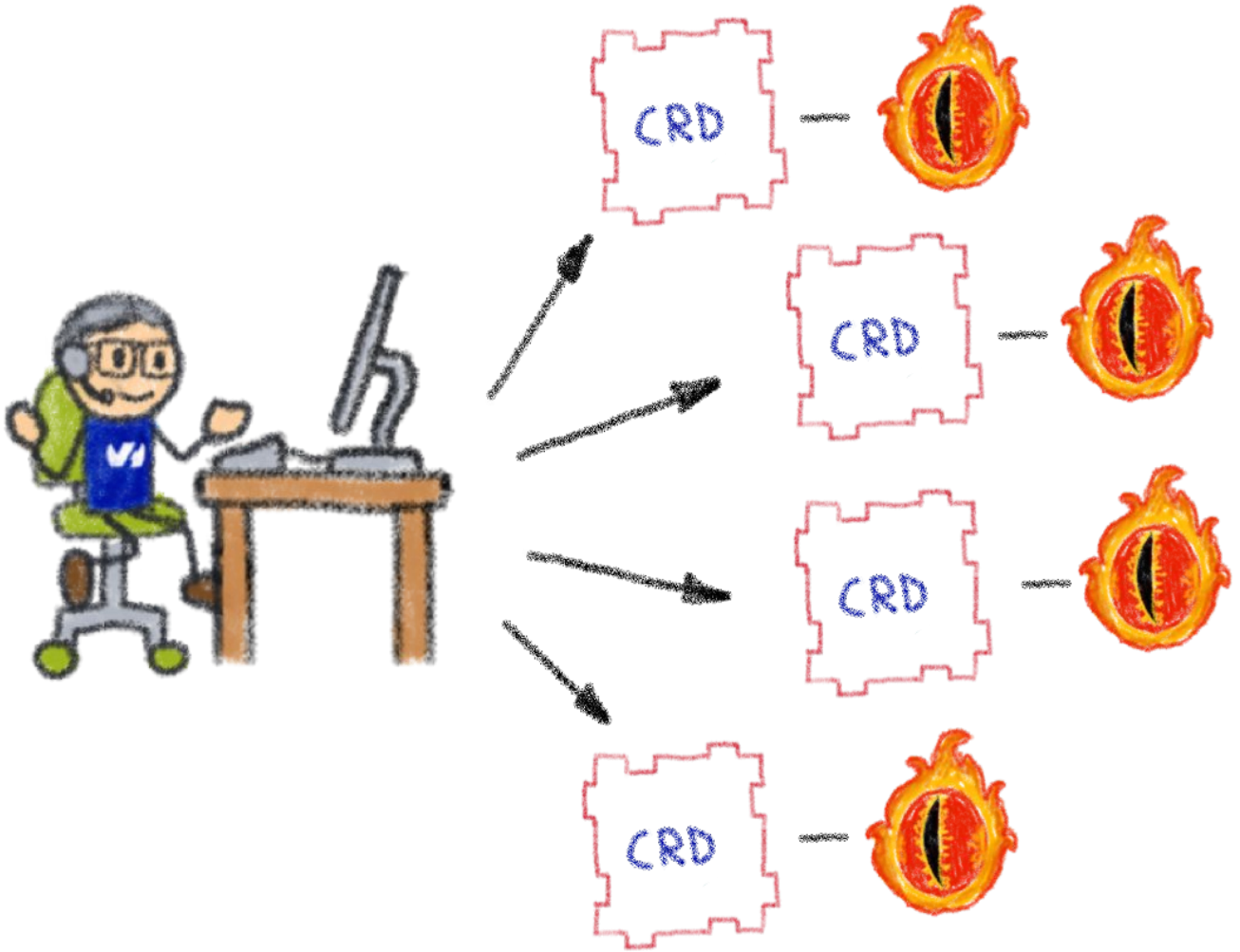


PostgreSQL

redis

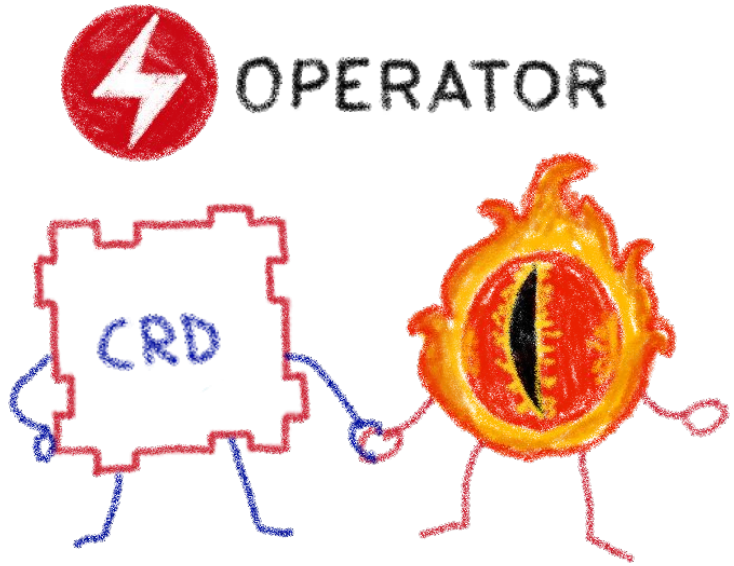
Things like adding an instance to a pool,
doing a backup, sharding...

Knowledge encoded in CRDs and Controllers



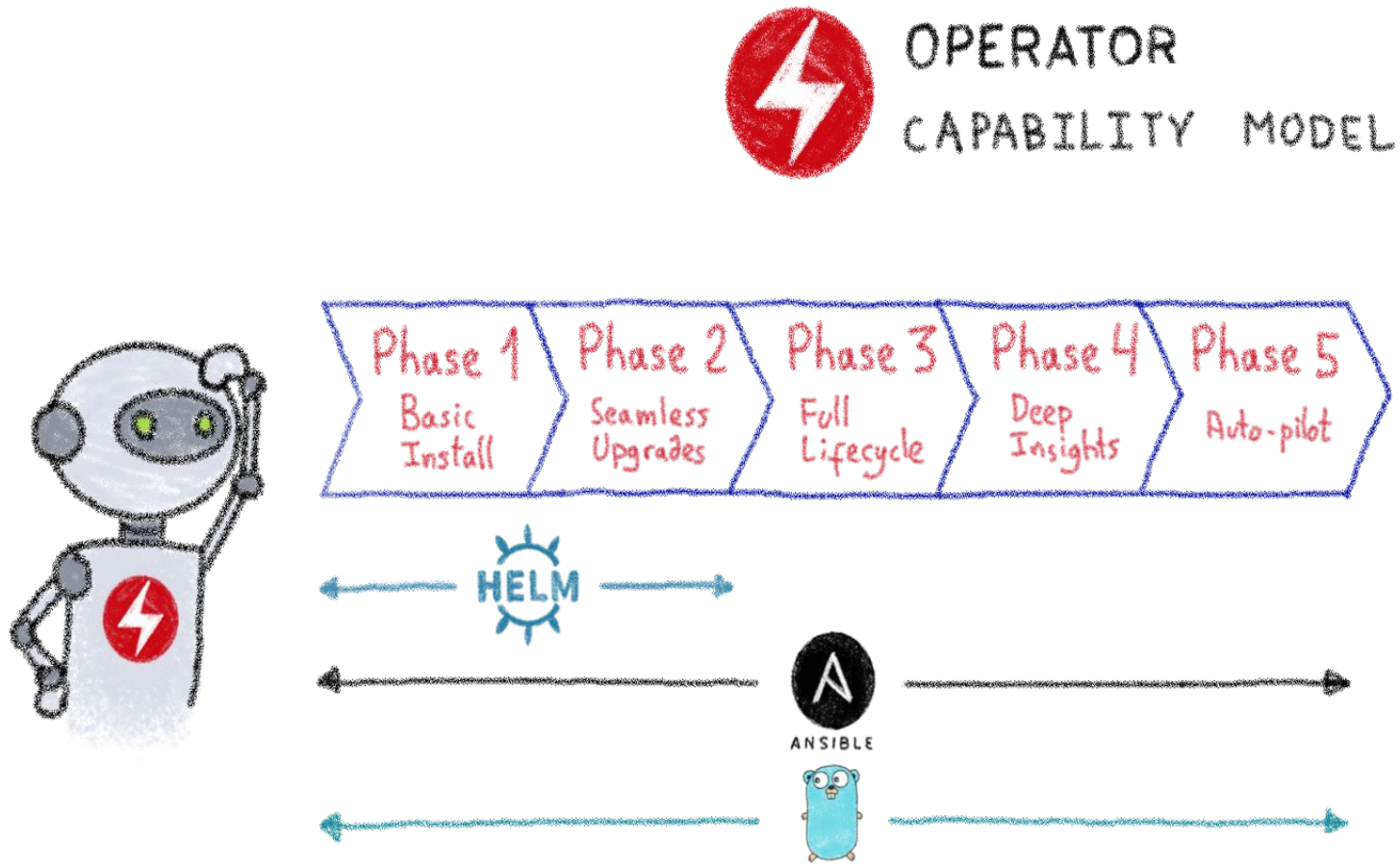
Encapsulating business logic in CRDs & Controllers

Custom Controllers for Custom Resources



Operators implement and manage Custom Resources using custom reconciliation logic

Operator Capability Model



Gauging the operator maturity

That's all, folks!

Thank you all!