

# DevOpsCon

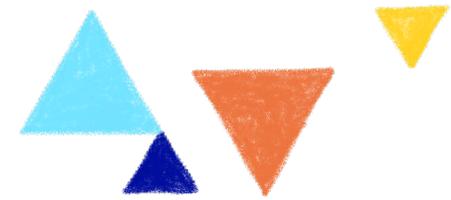
## Kubernetes Operators: Operating Cloud Native services at scale

Horacio Gonzalez

2020-06-30



@LostInBrittany



# Who are we?

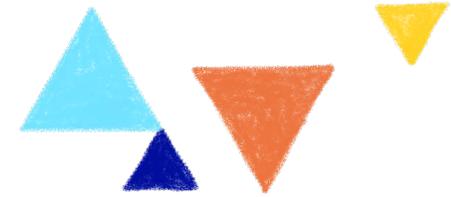
Introducing myself and  
introducing ~~OVH~~ OVHcloud



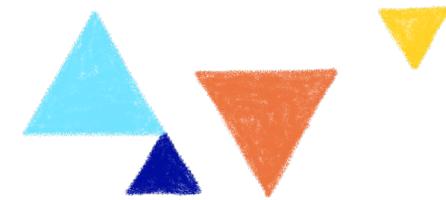
# Horacio Gonzalez

**@LostInBrittany**

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



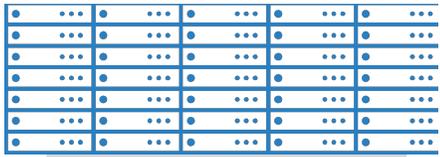
# OVHcloud: A Global Leader



200k Private cloud  
VMs running



**Dedicated IaaS  
Europe**



Hosting capacity :  
**1.3M Physical  
Servers**

**360k**  
Servers already  
deployed



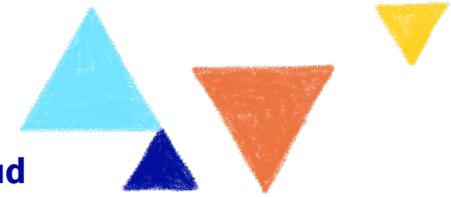
Own  
20Tbps  
Network  
with  
35 PoPs

30 Datacenters

> **1.3M Customers in 138 Countries**



# OVHcloud: 4 Universes of Products



## WebCloud

- Domain / Email**
  - Domain names, DNS, SSL, Redirect
  - Email, Open-Xchange, Exchange
  - Collaborative Tools, NextCloud
- PaaS for Web**
  - Mutu, CloudWeb
  - Plesk, CPANEL
  - PaaS with Platform.sh
- Virtual servers**
  - VPS, Dedicated Server
- SaaS**
  - Wordpress, Magento, Prestashop
  - CRM, Billing, Payment, Stats
  - MarketPlace
- Support, Managed**
  - Support Basic
  - Support thought Partners
  - Managed services

## Baremetal Cloud

- Standalone, Cluster**
  - General Purpose
  - SuperPlan
  - Game T2 >20e
  - Virtualization T3 >80e
  - Storage T4 >300e
  - Database T5 >600e
  - Bigdata 12KVA / 32KVA
  - HCI
  - AI
  - VDI Cloud Game
  - Network
- VPS aaS**
  - pCC DC
  - Virtuozzo Cloud
- Wholesales**
  - IT Integrators, Cloud Storage,
  - CDN, Database, ISV, WebHosting
  - High Intensive CPU/GPU,
- Encrypt**
  - KMS, HSM
  - Encrypt (SGX, Network, Storage)

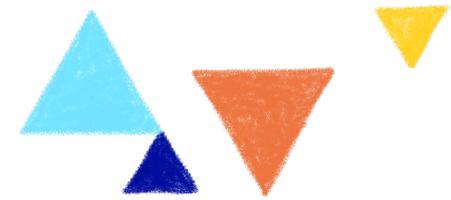
## Public Cloud

- Compute**
  - VM K8S, IA IaaS
  - Baremetal PaaS for DevOps
- Storage**
  - File, Block, Object, Archive
- Databases**
  - SQL, noSQL, Messaging,
  - Dashboard
- Network**
  - IP FO, NAT, LB, VPN, Router,
  - DNS, DHCP, TCP/SSL Offload
- Security**
  - IAM, MFA, Encrypt, KMS
- IA, DL**
  - Standard Tools for AI, AI Studio,
  - IA IaaS, Hosting API AI
- Bigdata, ML, Analytics**
  - Datalake, ML, Dashboard

## Hosted Private Cloud

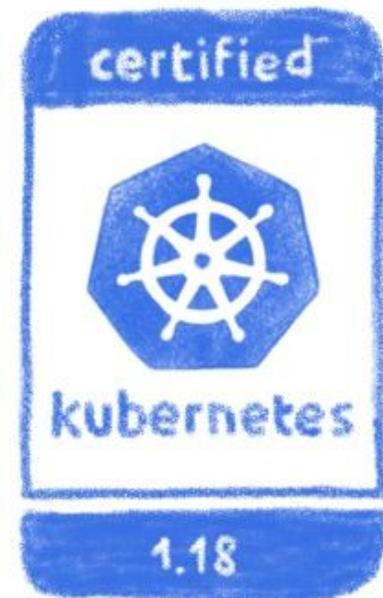
- Hosted Private Cloud**
  - VMware**
    - SDDC, vSAN 1AZ / 2AZ
    - vCD, Tanzu, Horizon, DBaaS,
    - DRaaS
  - Nutanix**
    - HCI 1AZ / 2AZ, Databases,
    - DRaaS, VDI
  - OpenStack**
    - IAM, Compute (VM, K8S)
    - Storage, Network, Databases
  - Storage**
    - Ontap Select, Nutanix File
    - OpenIO, MiniIO, CEPH
    - Zerto, Veeam, Atempo
  - AI**
    - ElementAI, HuggingFace,
    - Deepopmatic, Systran,
    - EarthCube
  - Bigdata / Analytics / ML**
    - Cloudera over S3, Dataiku,
    - Saagie, Tableau,
- Hybrid Cloud**
  - vRack Connect, Edge-DC, Private DC
  - Dell, HP, Cisco, OCP, MultiCloud
- Secured Cloud**
  - GOV, FinTech, Retail, HealthCare



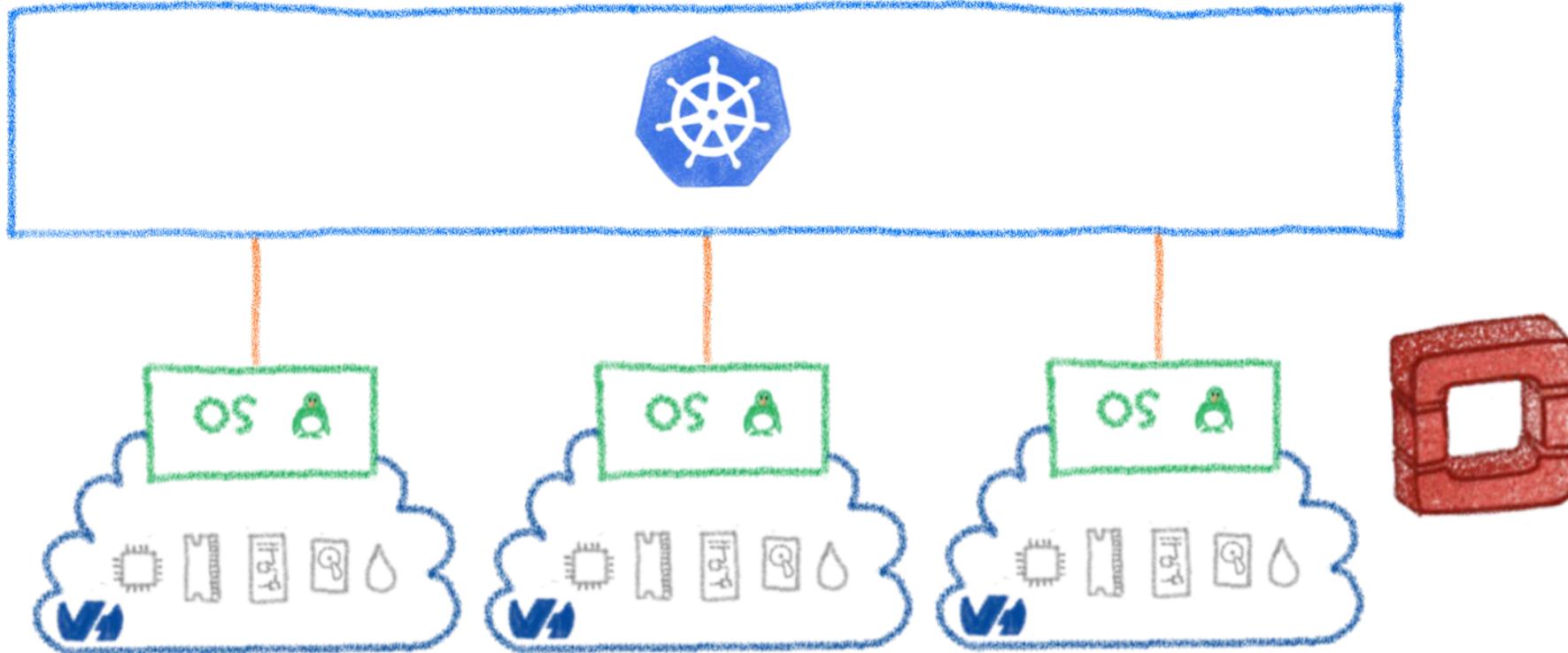
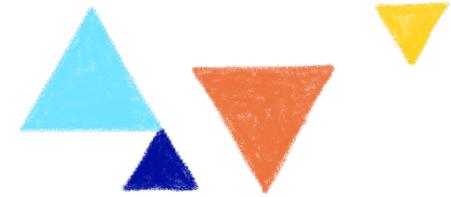


# OVHcloud Managed Kubernetes

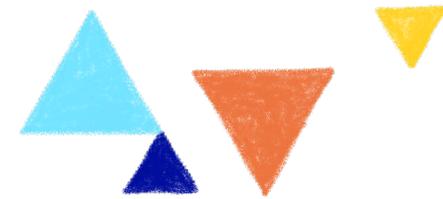
You use it, we operate it



# Built over our Openstack based Public Cloud



# Some interesting features



Fully managed, including version updates

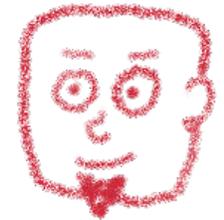
Price/performance ratio, free masters

Large instance range... and more to come

Predictible pricing

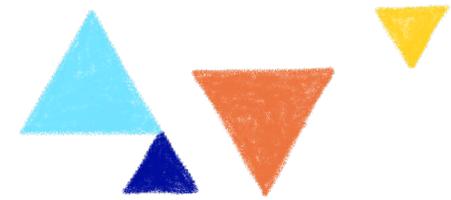


Developer



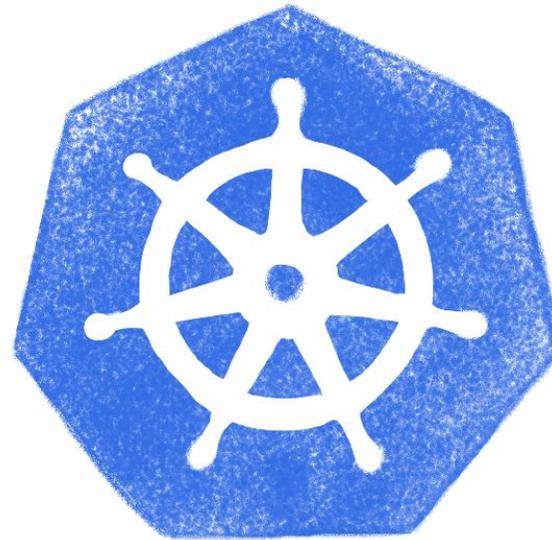
Cluster administrator



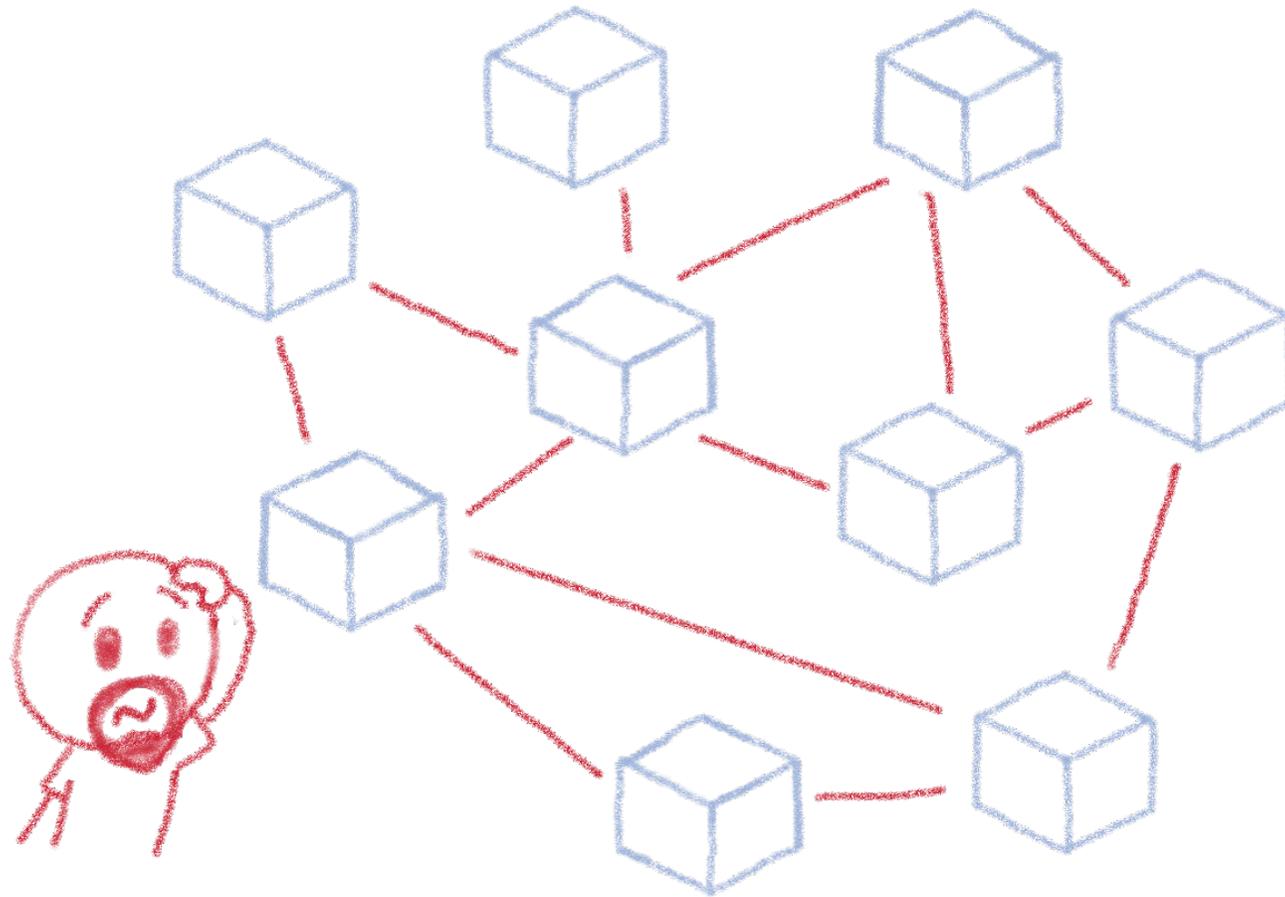
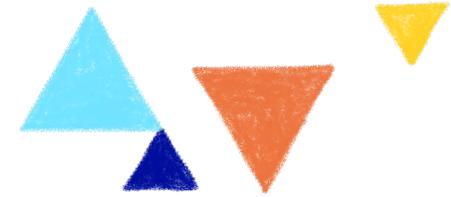


# Operating Kubernetes

Easier said than done



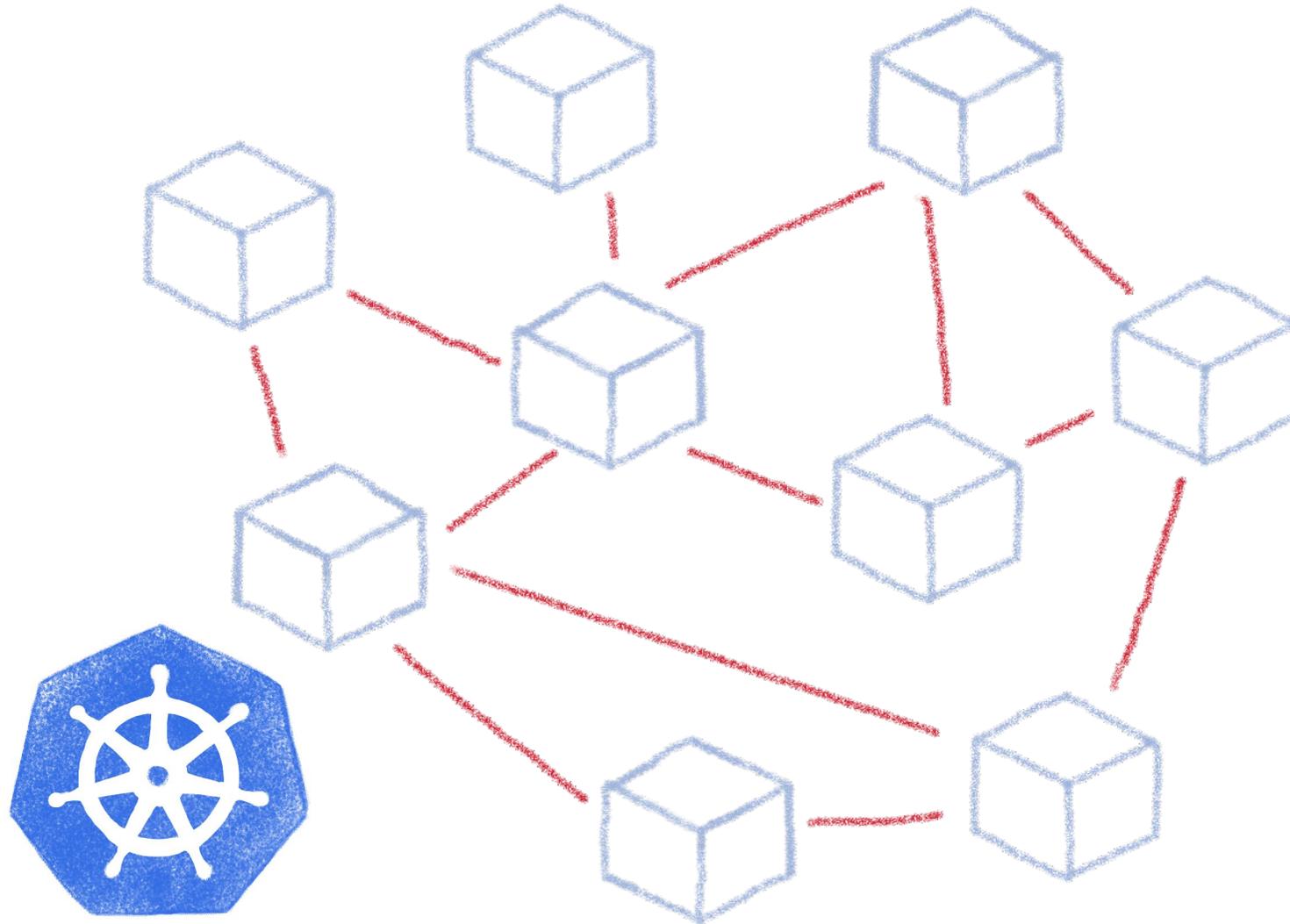
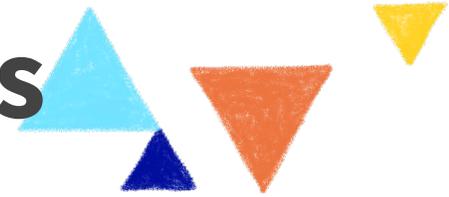
# Operating microservices?



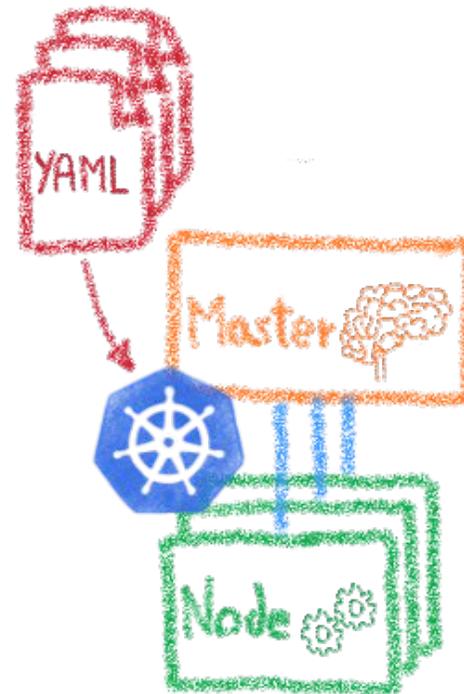
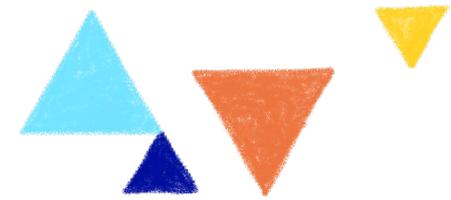
Are you sure you want to operate them by hand?



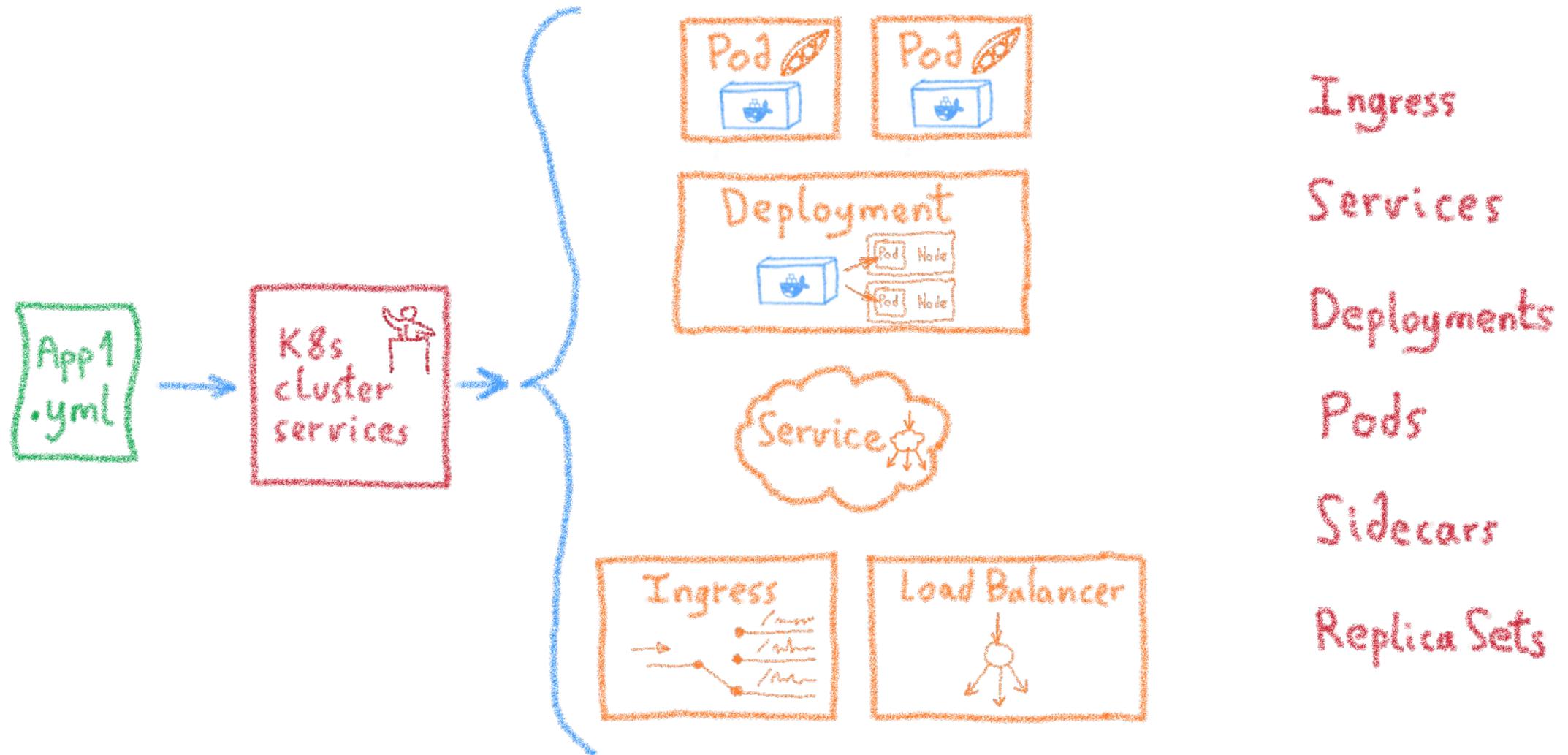
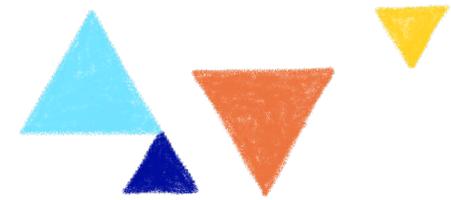
# Taming microservices with Kubernetes



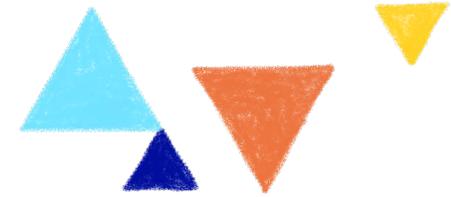
# Declarative infrastructure



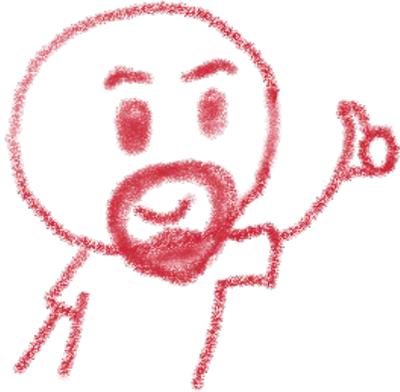
# Desired State Management



# Beyond a simple deployment



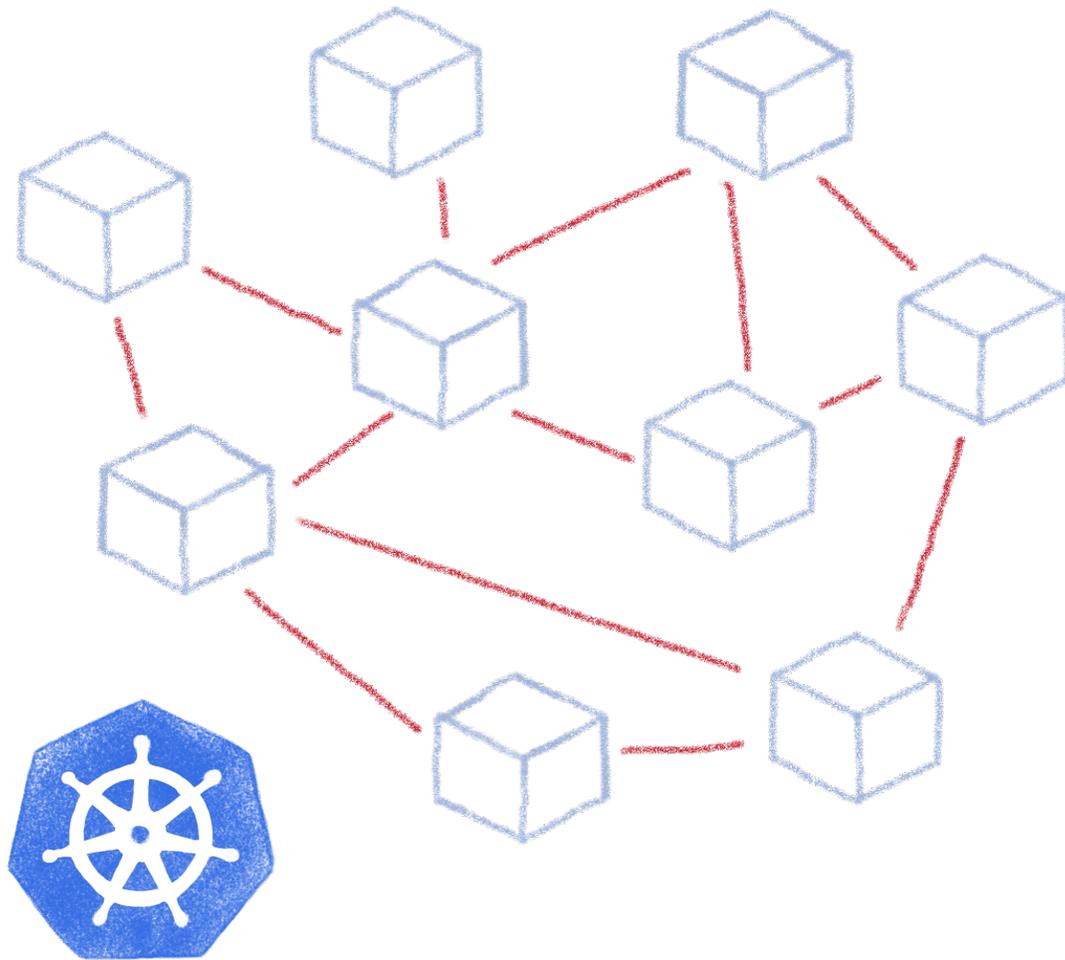
kubernetes



Everything is good now, isn't it?



# Complex deployments



Ingress

Services

Deployments

Pods

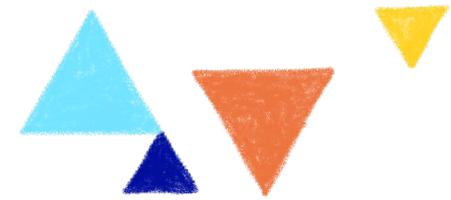
Sidecars

Replica Sets

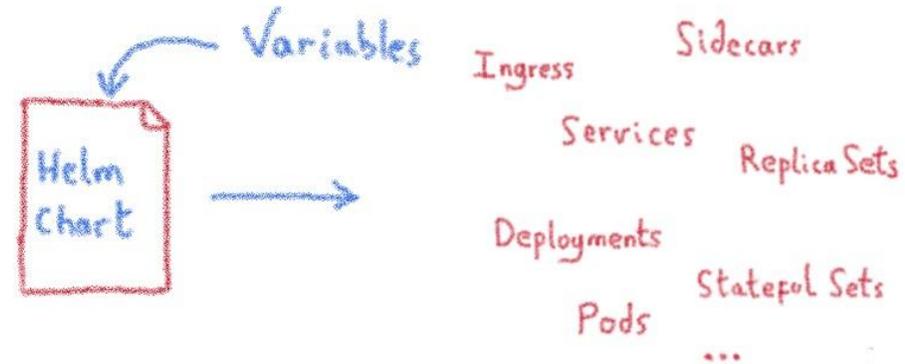
Stateful Sets

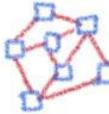


# Complex deployments



A package manager for Kubernetes

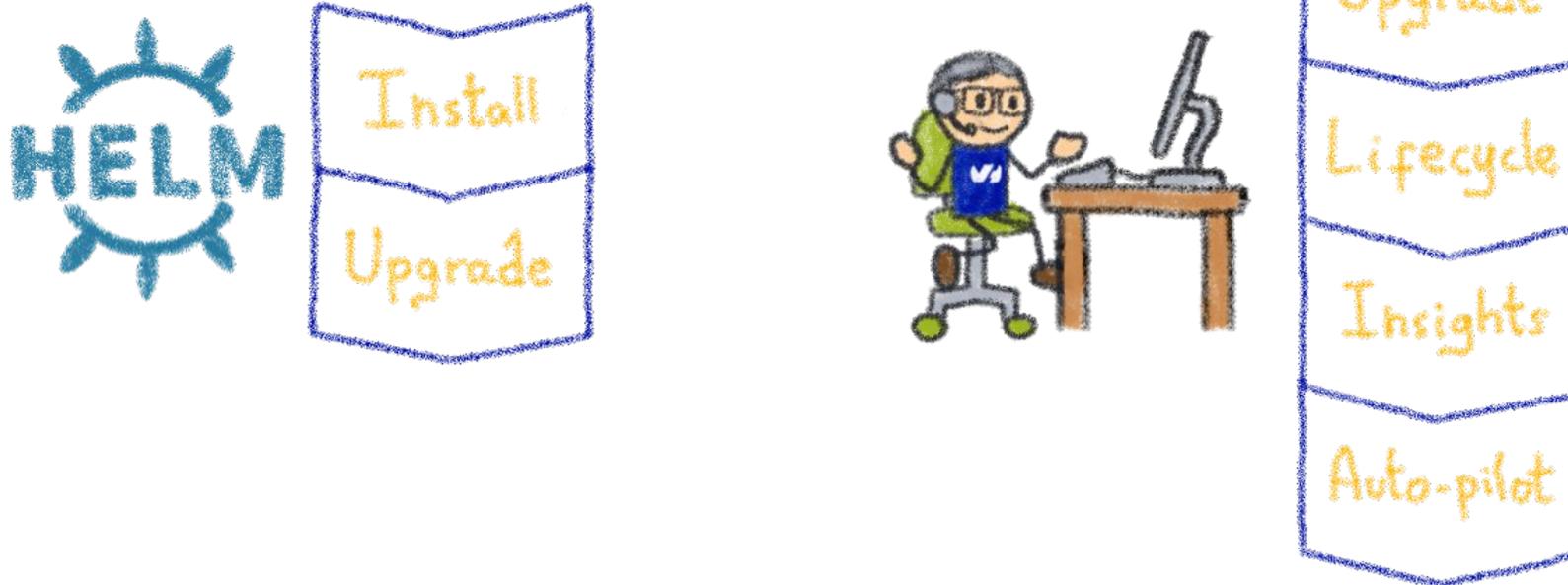
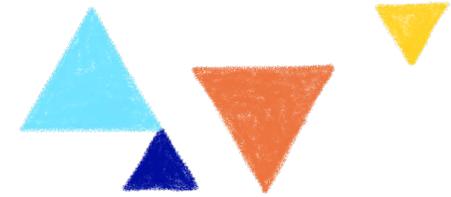


- Manage complexity 
- Simple sharing 

- Easy upgrades 
- Easy rollbacks 



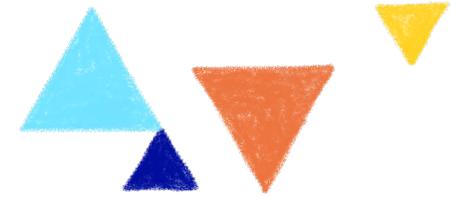
# Helm Charts are configuration



Operating is more than installs & upgrades

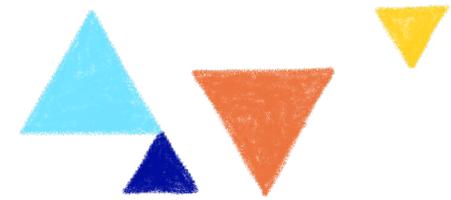


# Kubernetes is about automation



How about automating human operators?



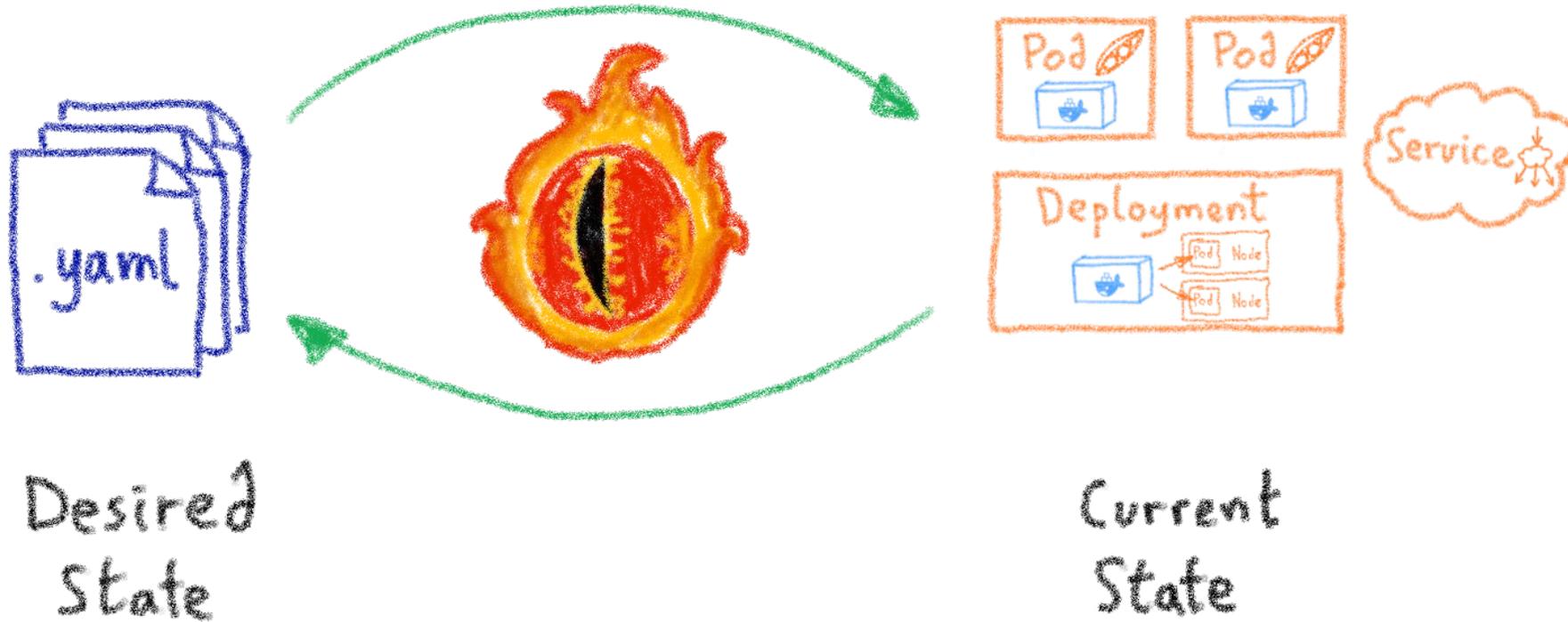
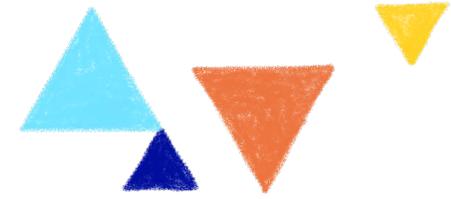


# Kubernetes Controllers

Keeping an eye on the resources



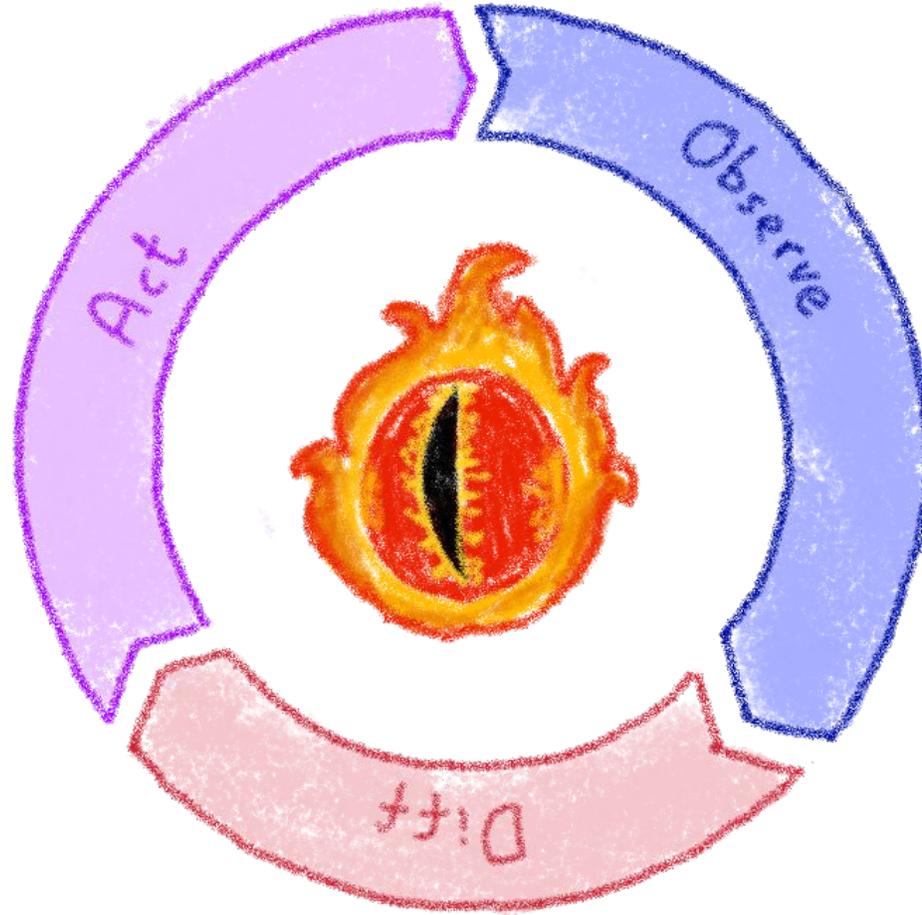
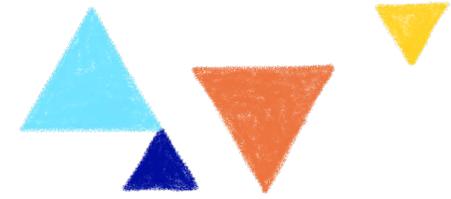
# A control loop



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

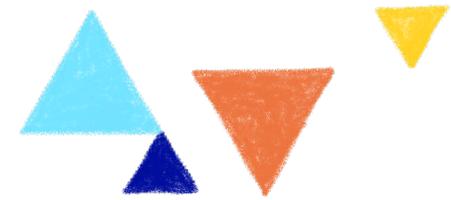


# A reconcile loop



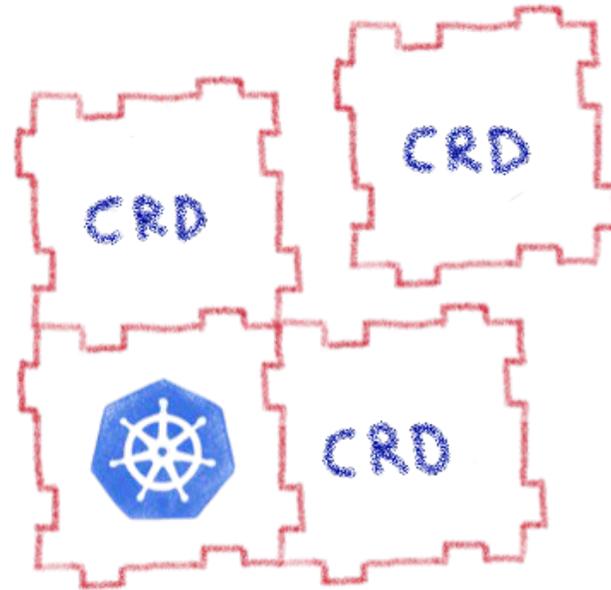
Strives to reconcile current state and desired state



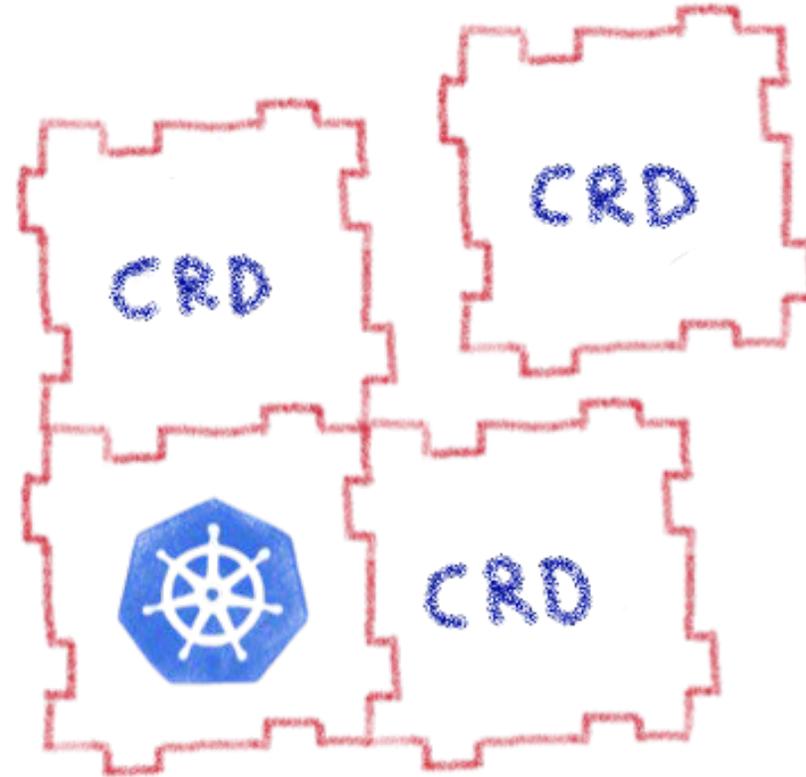
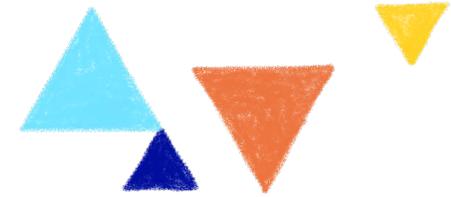


# Custom Resource Definitions

## Extending Kubernetes API

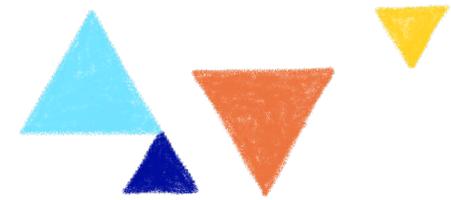


# Extending Kubernetes API



By defining new types of resources



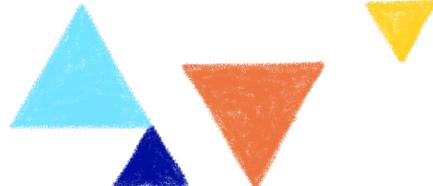


# Kubernetes Operator

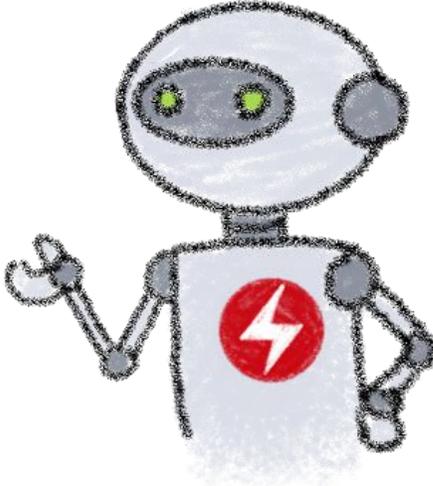
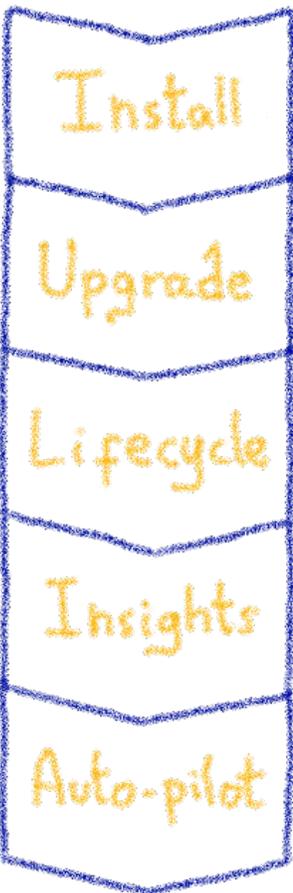
Automating operations



# What's a Kubernetes Operator?



Human Operator

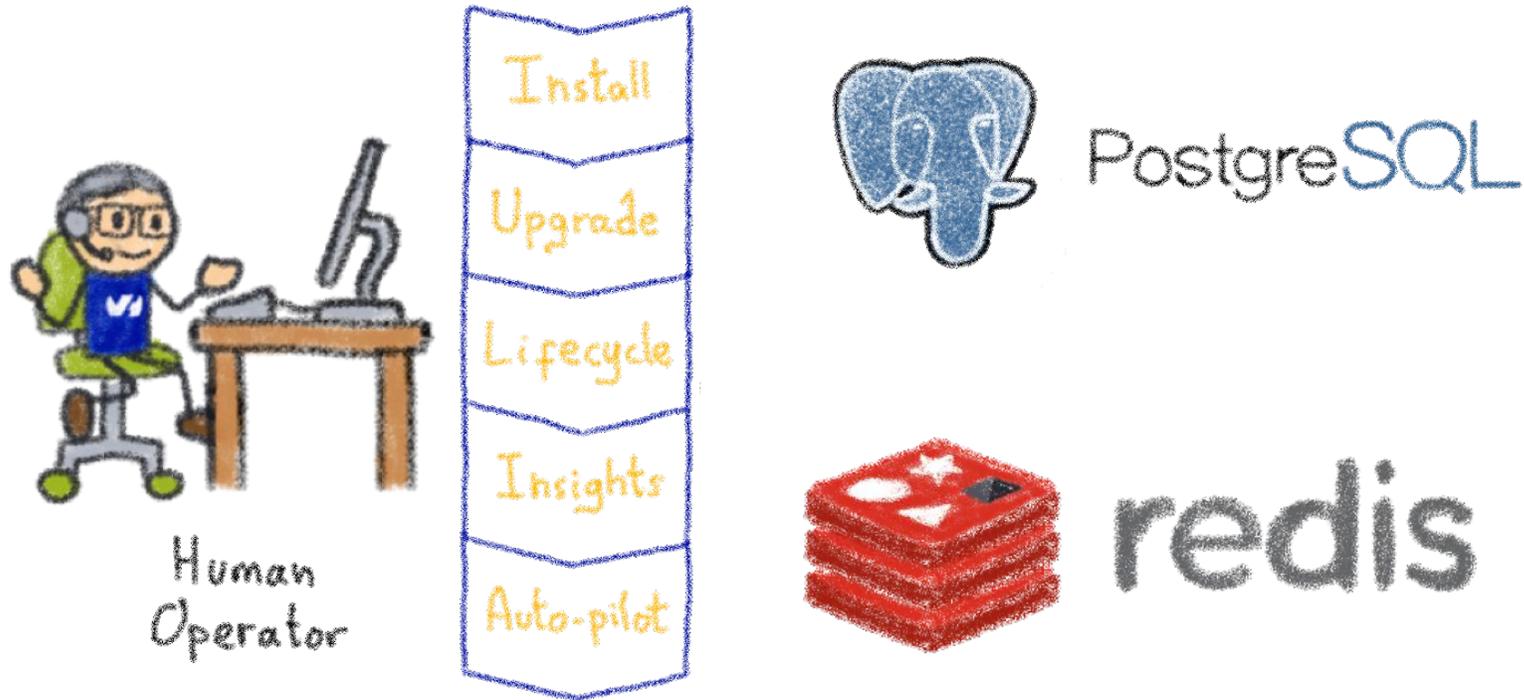
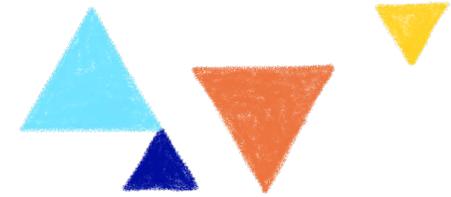


Kubernetes Operator

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



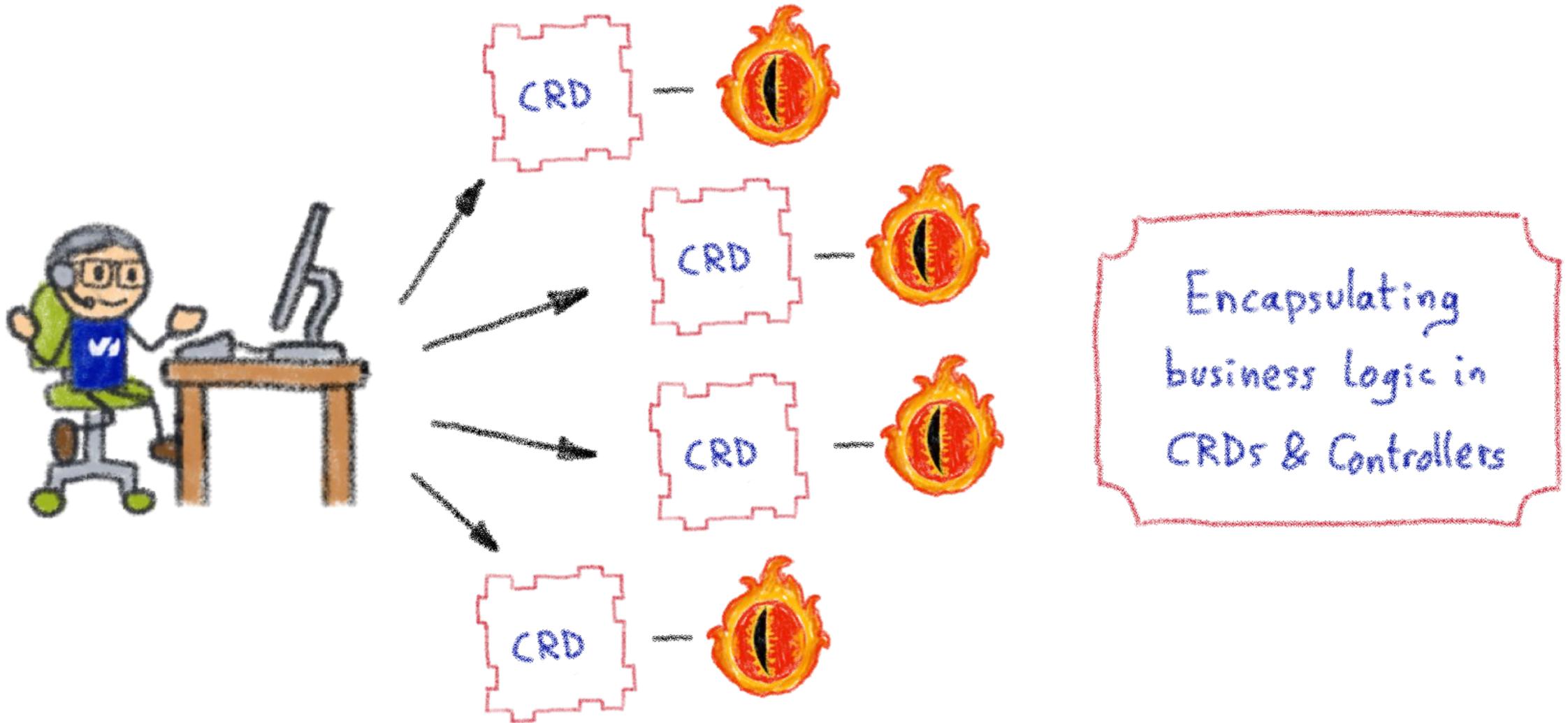
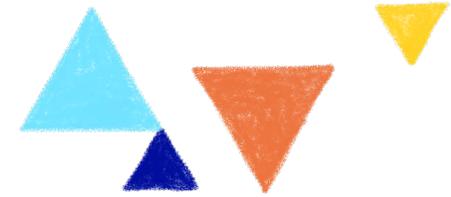
# Example: databases



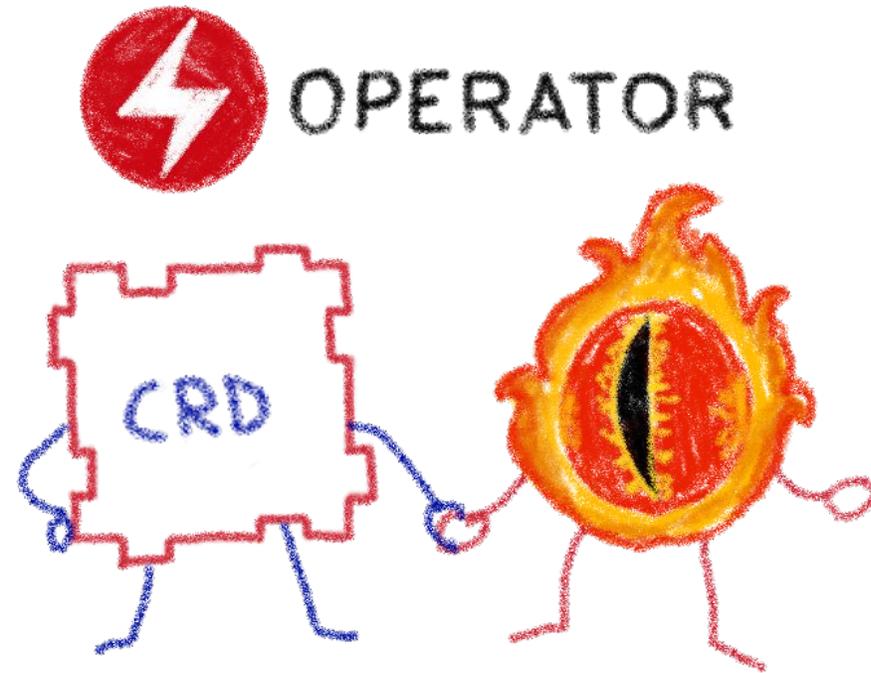
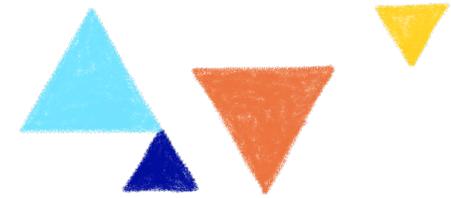
Things like adding an instance to a pool,  
doing a backup, sharding...



# Knowledge encoded in CRDs and Controllers



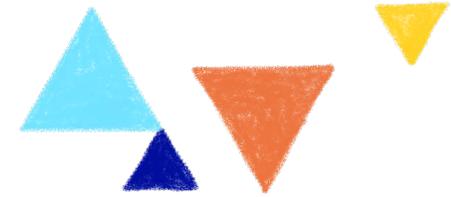
# Custom Controllers for Custom Resources



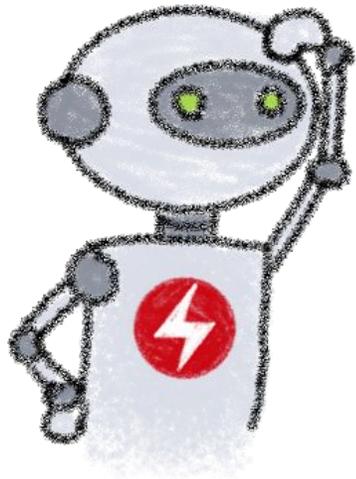
Operators implement and manage Custom Resources using custom reconciliation logic



# Operator Capability Model



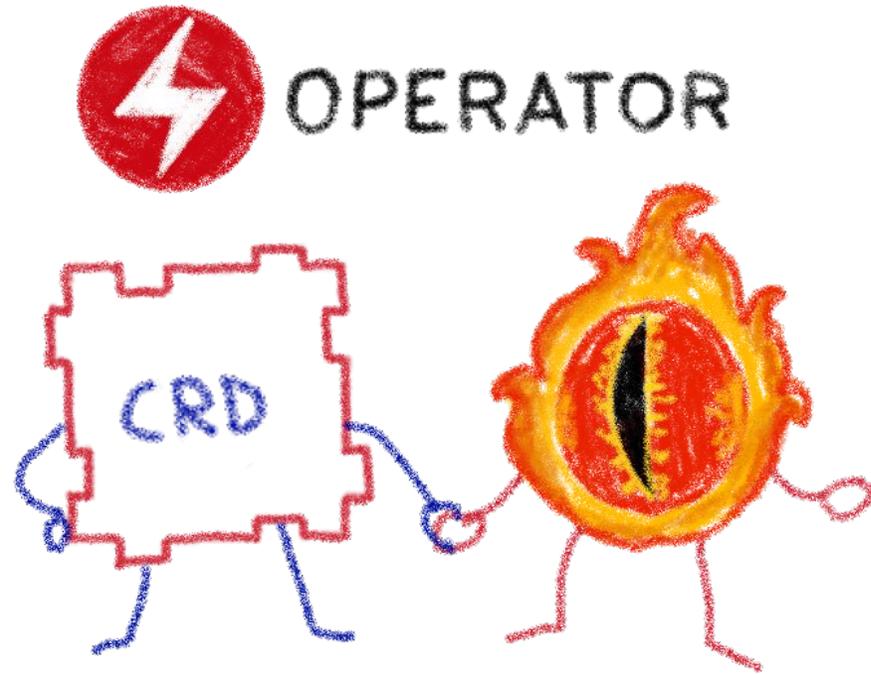
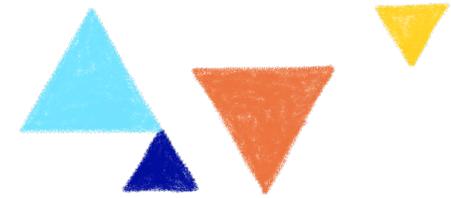
OPERATOR  
CAPABILITY MODEL



Gauging the operator maturity



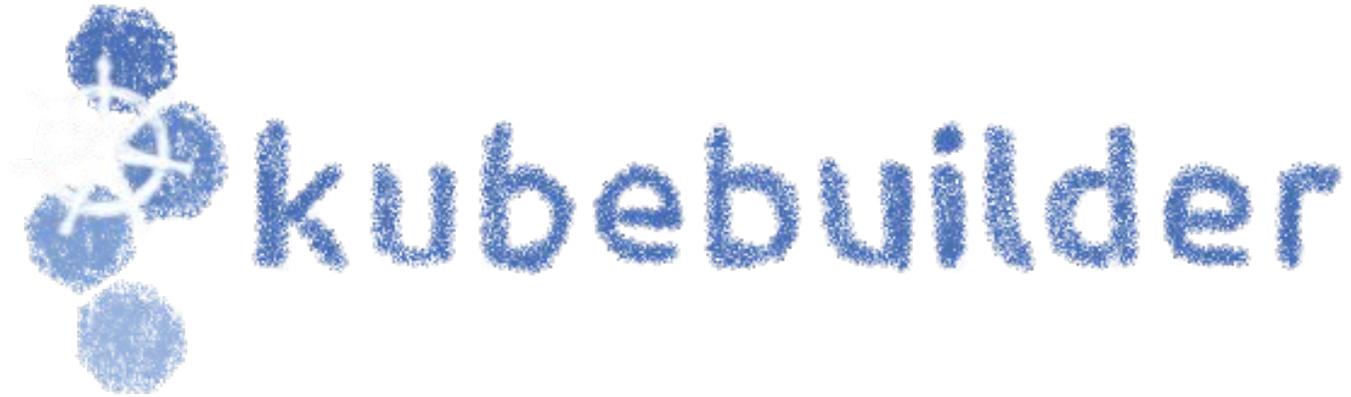
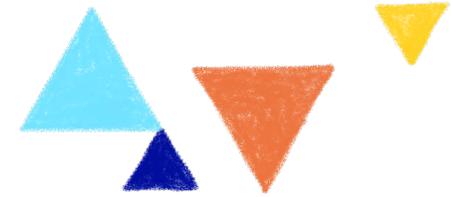
# How to write an Operator



- 1- Create a new project
- 2- Write the CRDs to define new resource APIs
- 3- Specify resources to watch
- 4- Define the reconciliation logic in the Controllers
- 5- Build the Operator



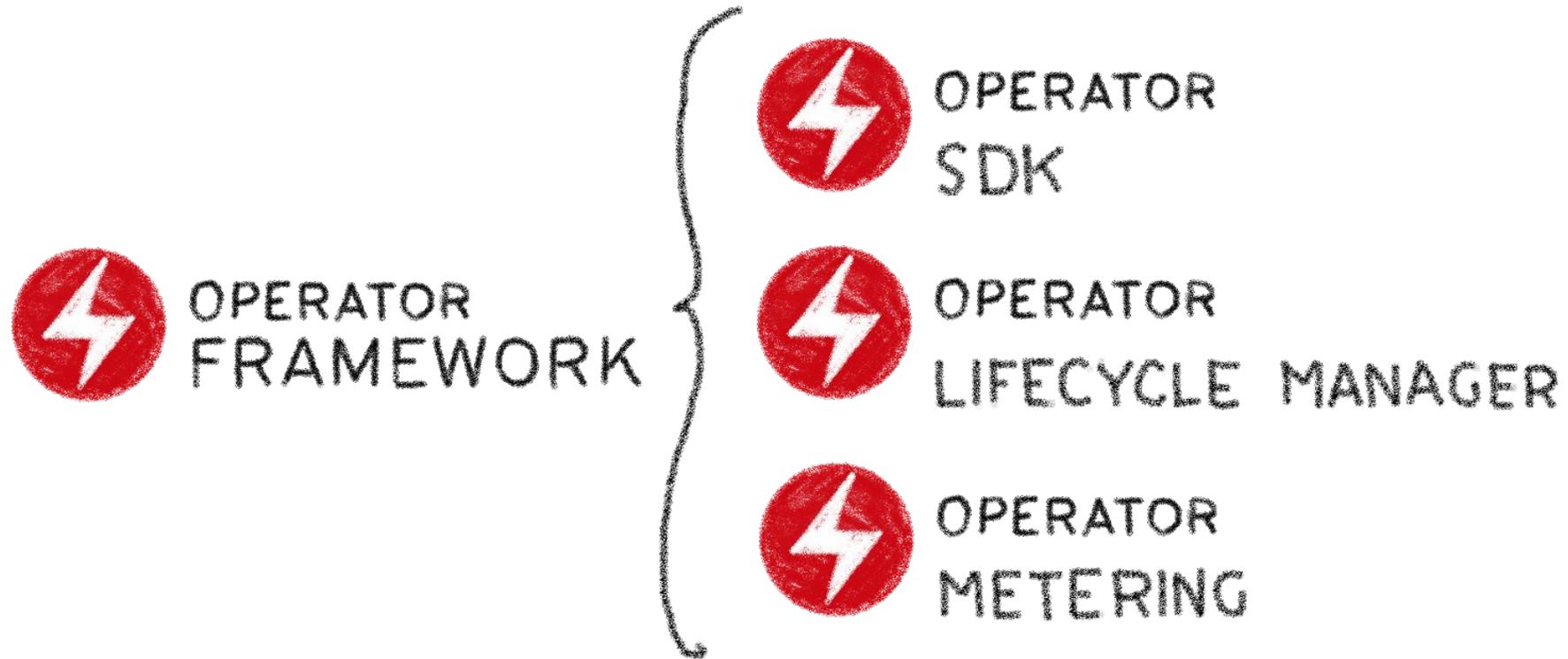
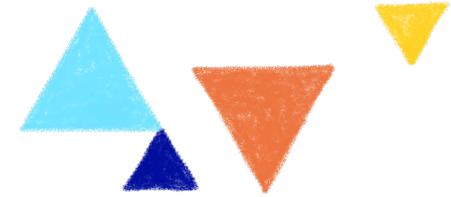
# Kubebuilder



SDK for building Kubernetes APIs using CRDs



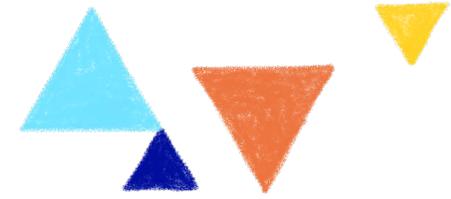
# The Operator Framework



Open source framework to accelerate the development of an Operator

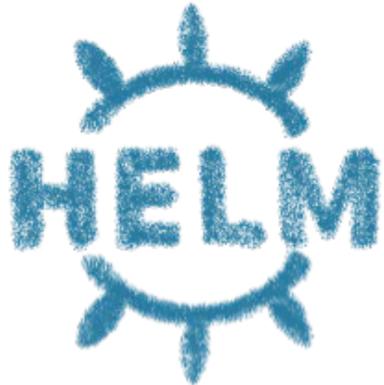


# Operator SDK

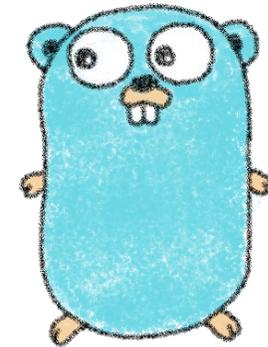


OPERATOR  
SDK

BUILD  
TEST  
ITERATE



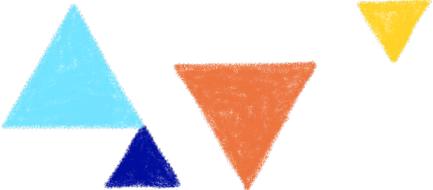
ANSIBLE



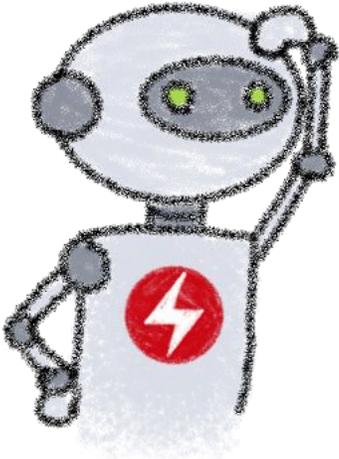
Three different ways to build an Operator



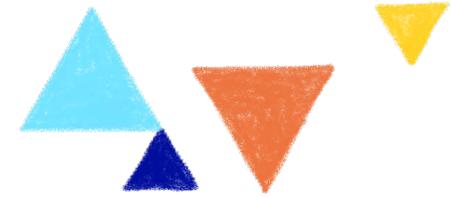
# Operator SDK and Capability Model



OPERATOR  
CAPABILITY MODEL



# Operator Lifecycle Manager

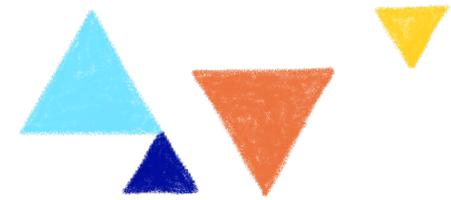


OPERATOR  
LIFECYCLE MANAGER

INSTALL  
MANAGE  
UPDATE





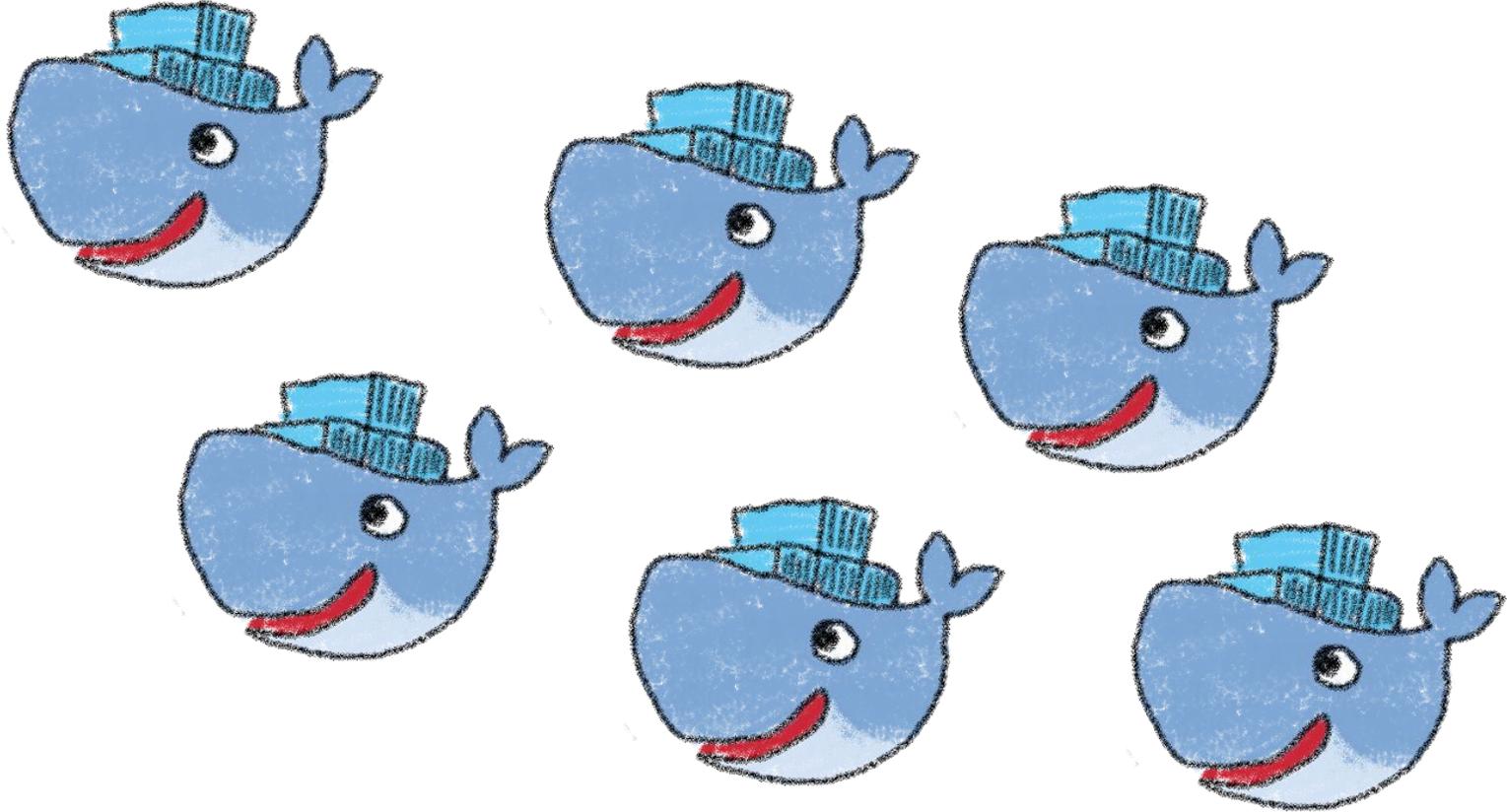
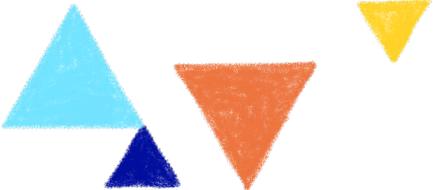


# Harbor Operator

Managing private registries at scale



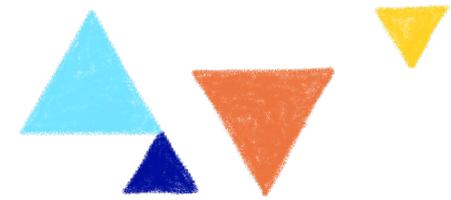
# We wanted to build a new product



OVHcloud Managed Private Registry



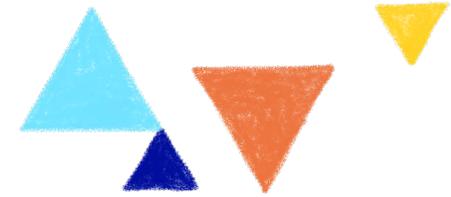
# Looking at the Open Source world



Two main alternatives around Docker Registry



# Harbor has more community traction



★ Star 11.5k

🍴 Fork 3.1k



CLOUD NATIVE  
COMPUTING FOUNDATION



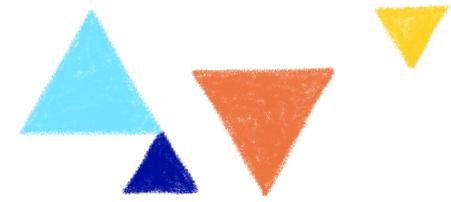
★ Star 2.6k

🍴 Fork 454

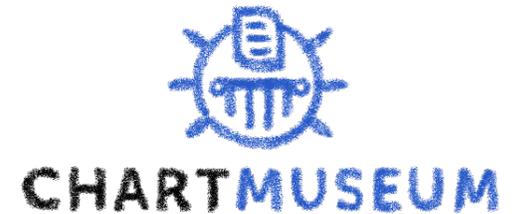
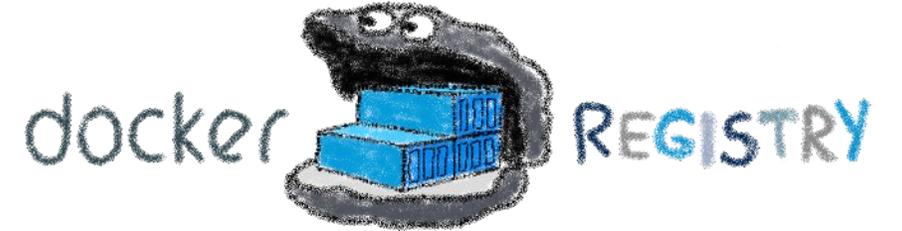
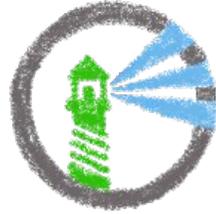
Two main alternatives



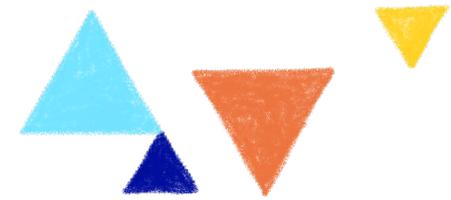
# Harbor has lots of components



NGINX



# But it has a Helm Chart



It should be easy to install, isn't it?

```
$ helm install harbor
```

What about configuration?

Installing a 200 GB K8s volume?

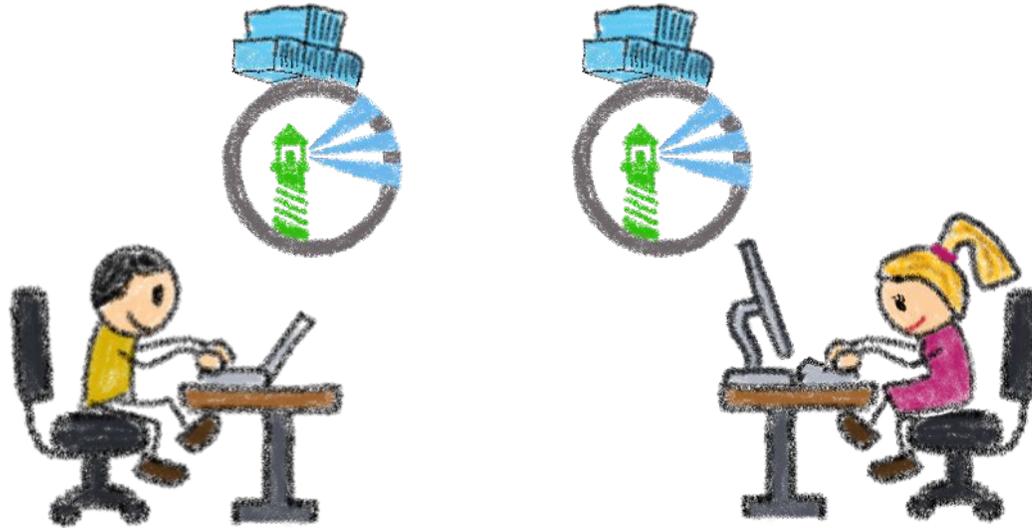
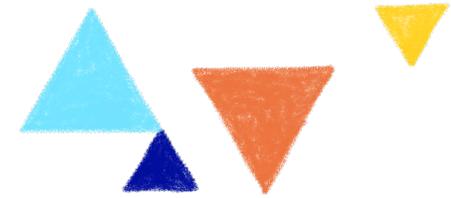
Nginx pods for routing requests?

One DB instance per customer?

Managing pods all around the cluster?



# We wanted a Managed Private Registry



One Harbor instance per customer  
One-click deployment, API  
Shared tooling, isolated data

Ingress controller



redis



PostgreSQL

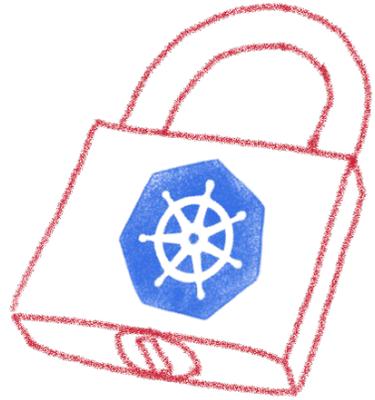
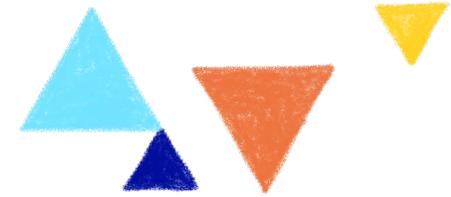
Object Storage

as a Service

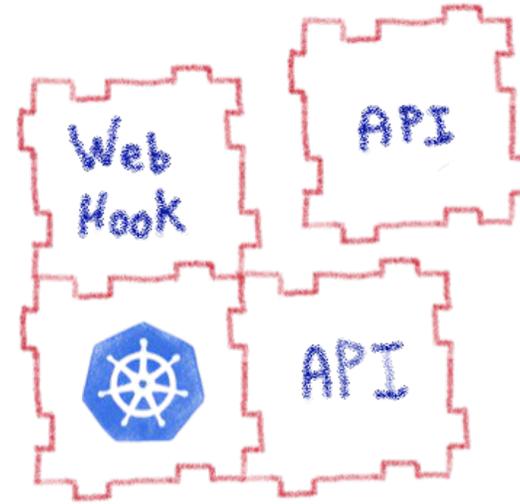
Reusing existing services



# Using the platform



RBAC  
Security policies  
API inputs validation

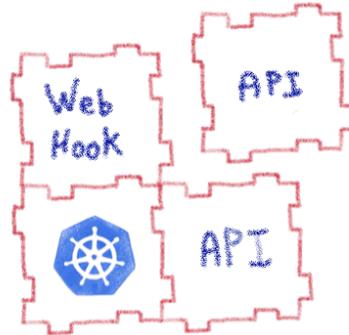
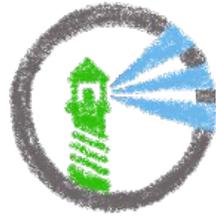
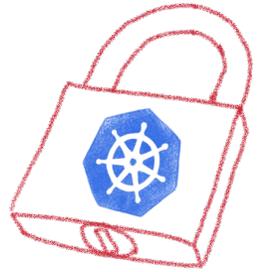
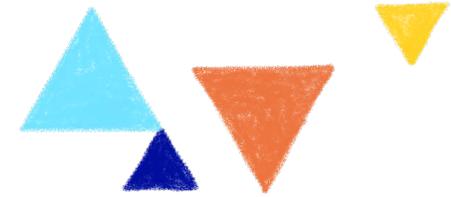


Modularity & Extensibility  
APIception  
Web hooks

Kubernetes tooling to the rescue



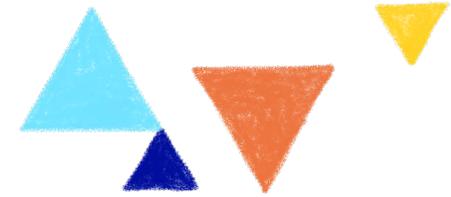
# Let's automate it



We needed an operator... and there wasn't any



# Working with the community



We need an Operator  
for  HARBOR, we are  
coding it. Interested?



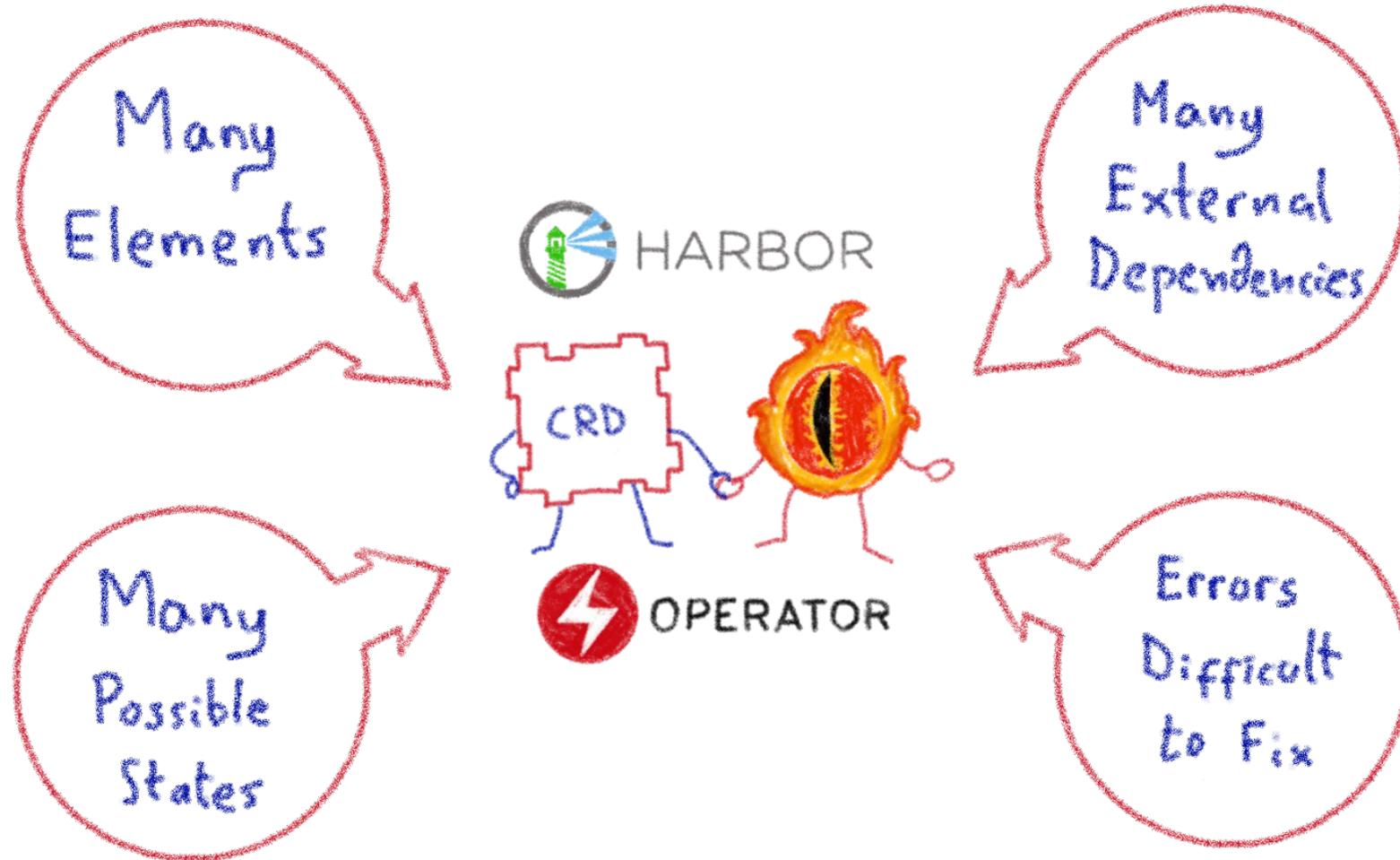
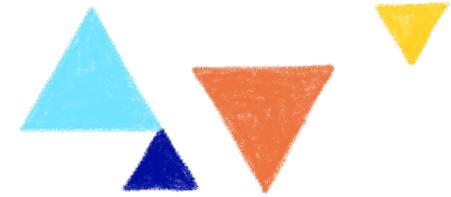
Oh yeah!  
We would love it!



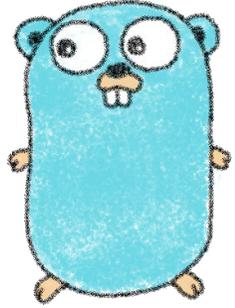
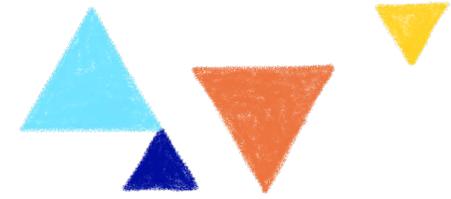
Harbor community also needed the operator



# The challenge: reconciliation loop



# The Harbor Operator

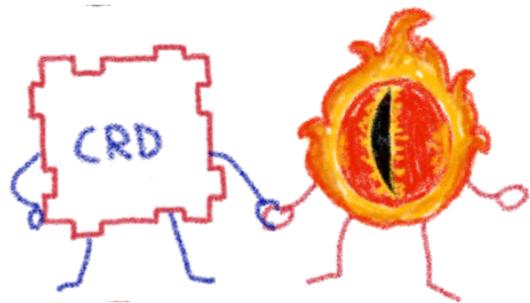


Written in Go



7 Components

- Config Map
- Secrets
- Ingress
- Certificater
- Deployments
- Services



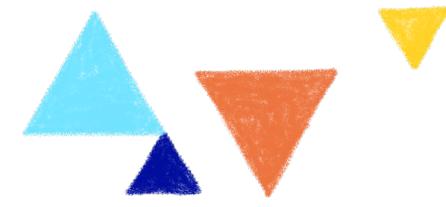
1 CRD & 1 Controller



Uses other operators for specific tasks (e.g. Cert Manager)



# It's Open Source



Donated by  OVHcloud  
to the

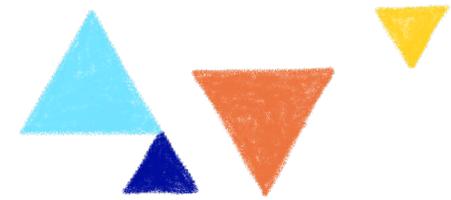


**CLOUD NATIVE**  
COMPUTING FOUNDATION



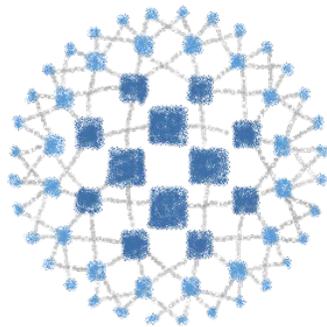
<https://github.com/goharbor/harbor-operator>



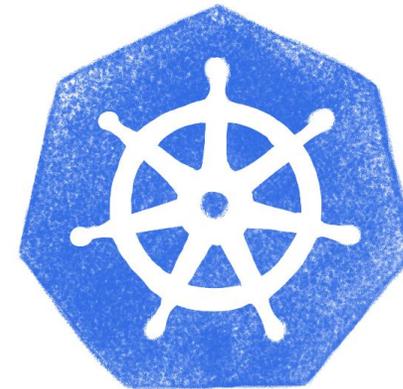


# LoadBalancer Operator

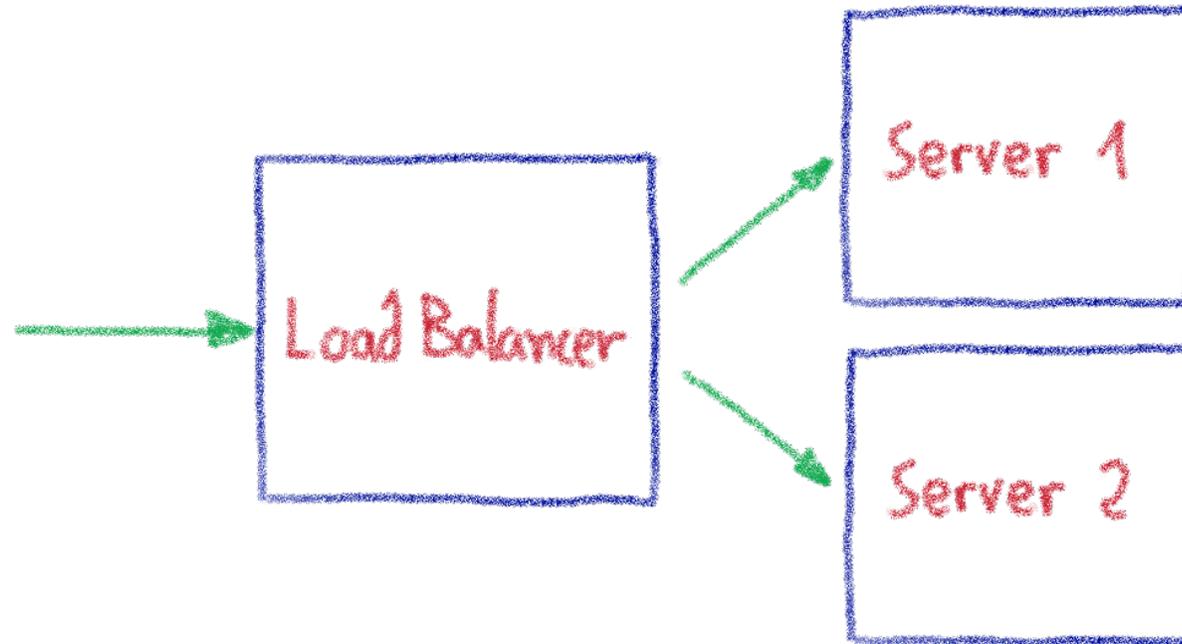
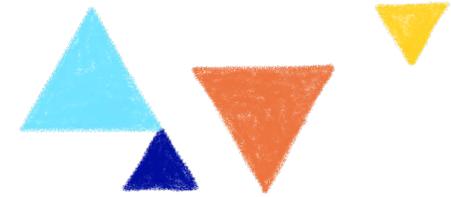
A managed LoadBalancer at scale



HAPROXY



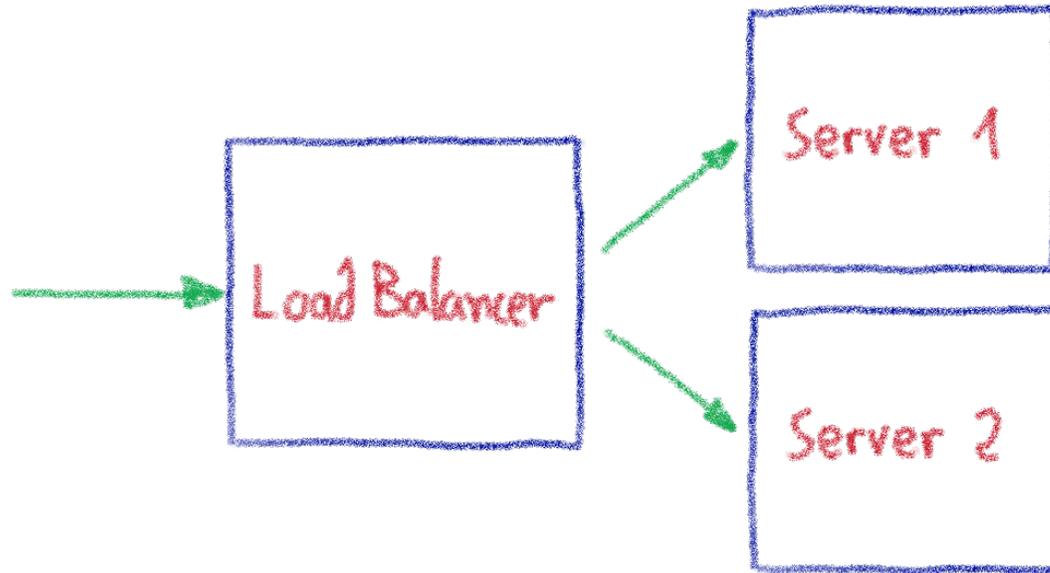
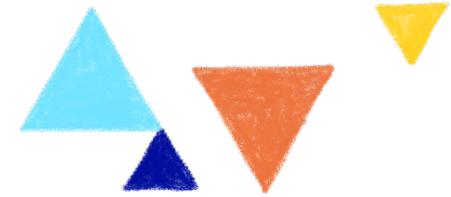
# Load Balancer: a critical cog



Cornerstone of any Cloud Provider's infrastructure



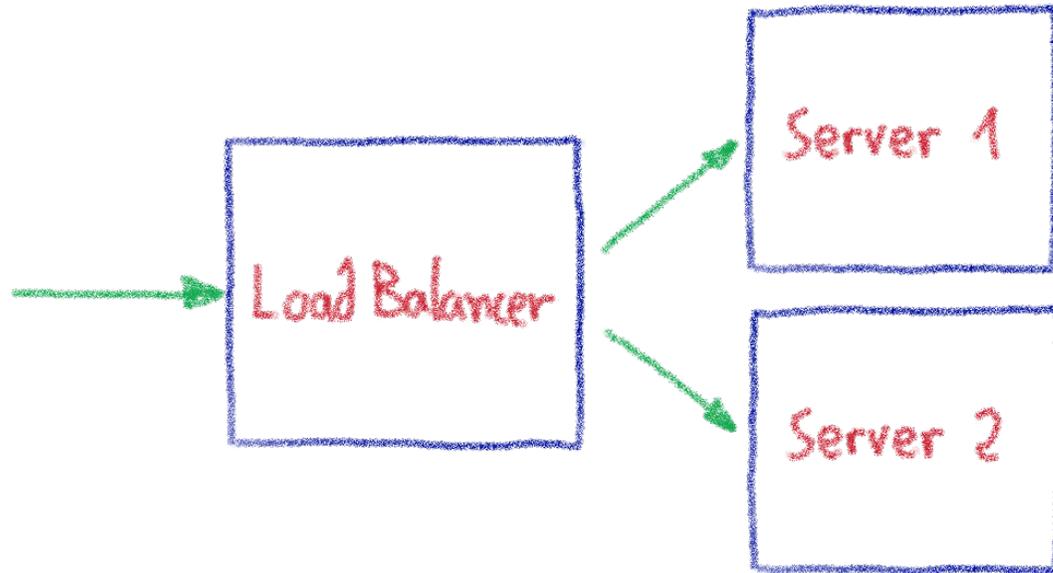
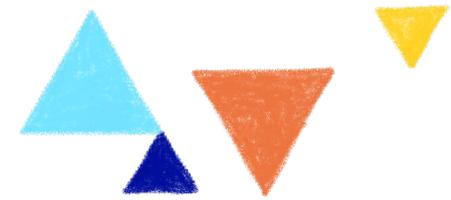
# Our legacy Load Balancer stack



- Excellent performances
  - Built on bare metal servers + BGP
  - Custom made servers tuned for network traffic
- Carry the TLS termination
  - SSL / LetsEncrypt
- Not cloud ready
  - Piloted by configuration files
  - Long configuration loading time
- Custom made hardware
  - Slower to build
  - Needs to be deployed on 30 datacenters



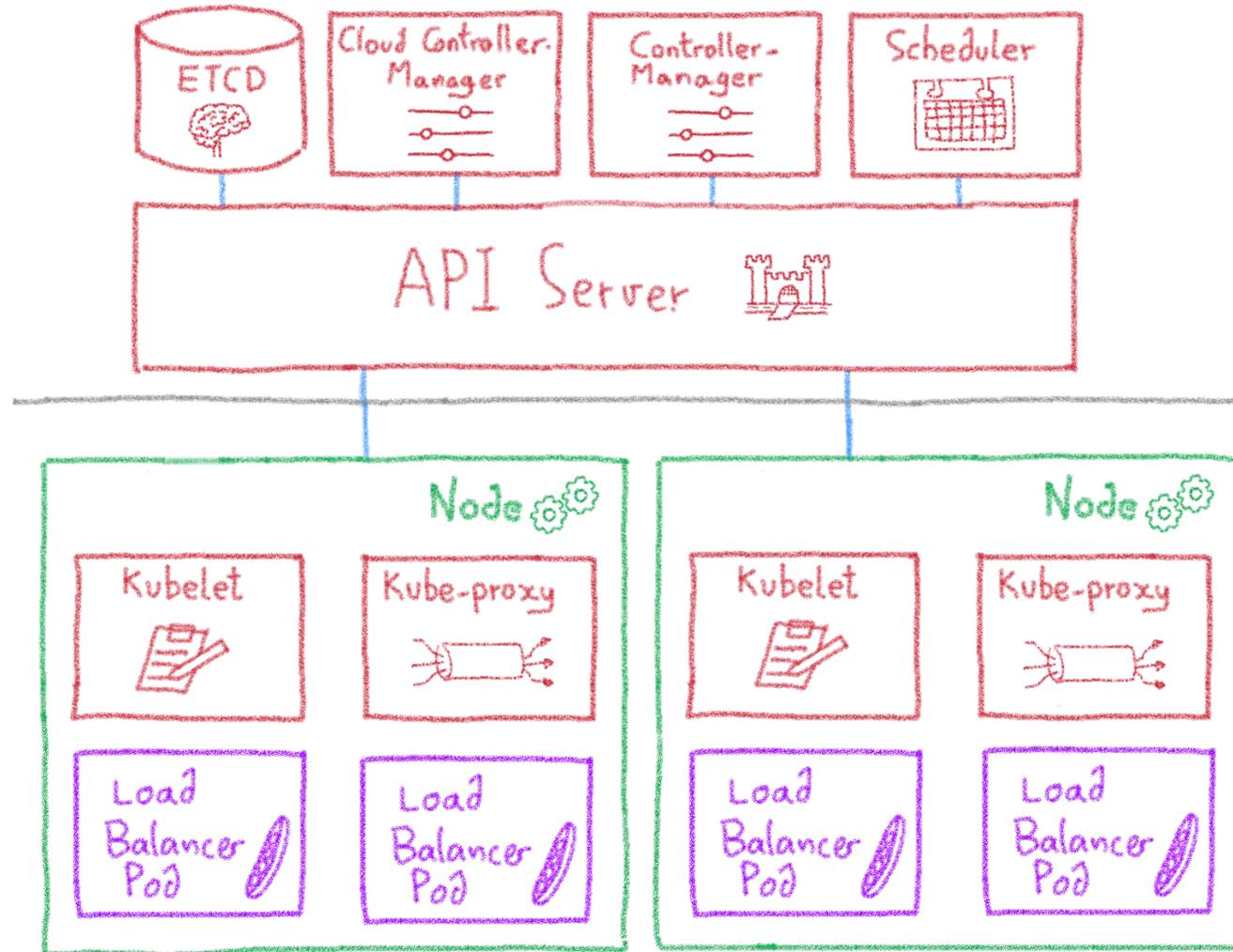
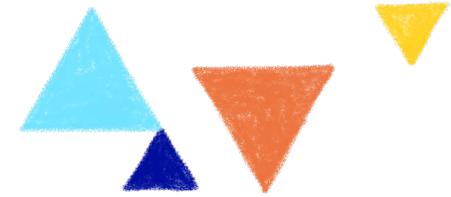
# Our needs for a new Load Balancer



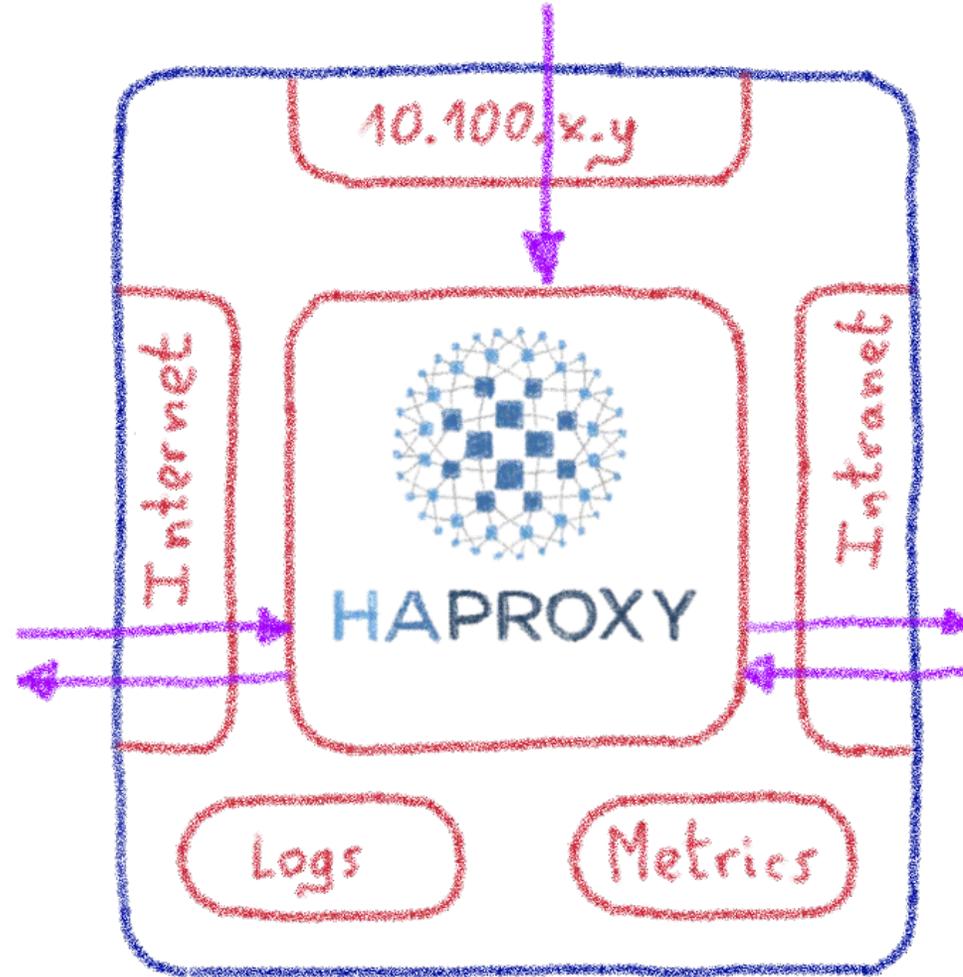
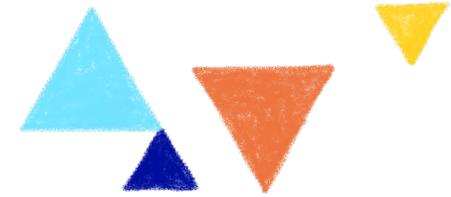
- Supporting mass update
- Quickly reconfigurable
- Available anywhere quickly
- Easily operable
- Integrated into our Public Cloud



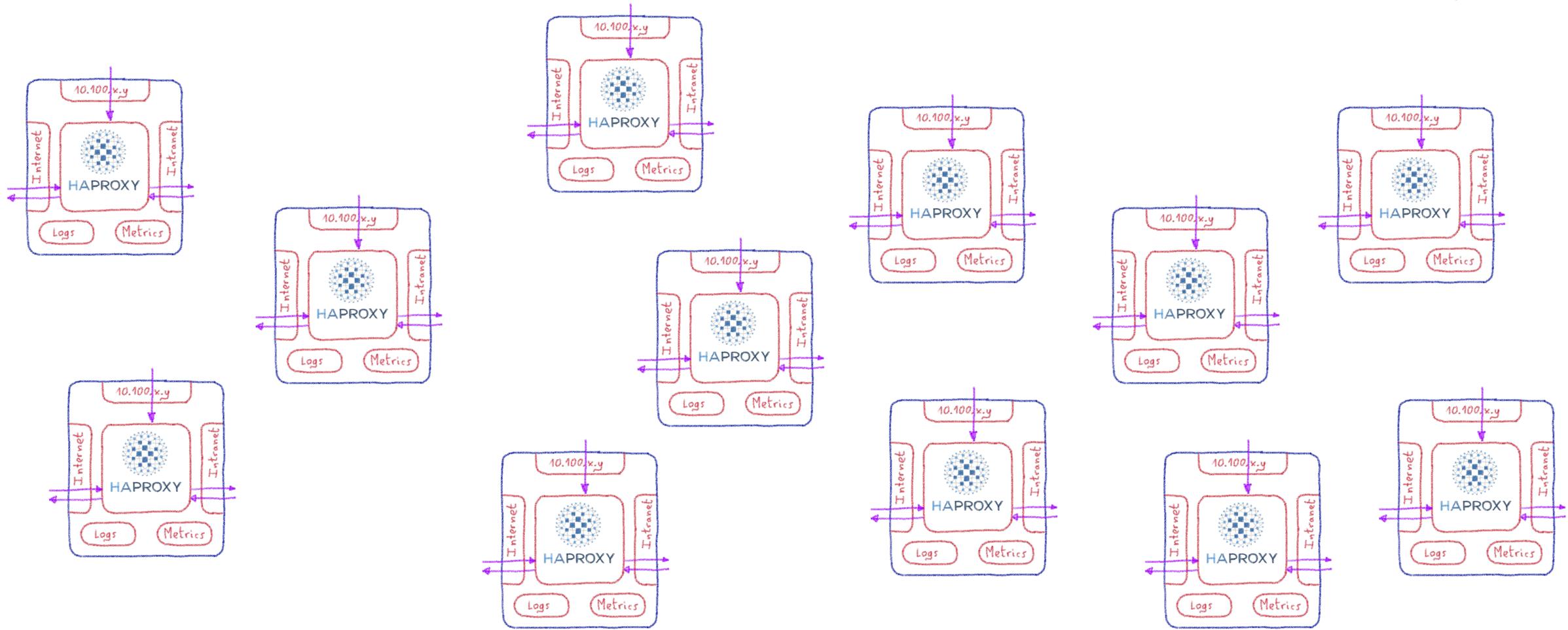
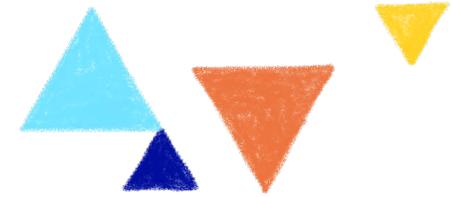
# Building it on Kubernetes



# A Load Balancer in a pod



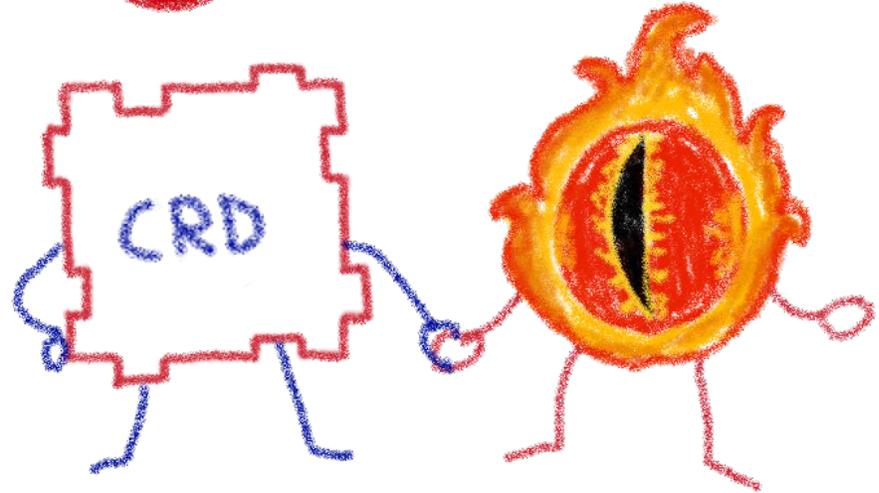
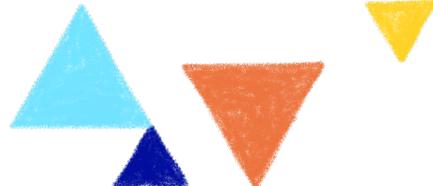
# Orchestrating one million LBs...



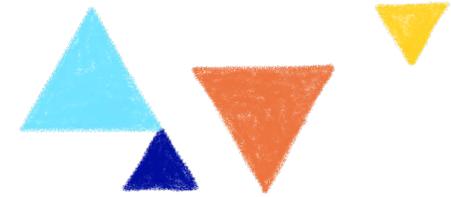
kubectl apply -f lb is not an option!



# We needed an Operator



# Network: multus-cni

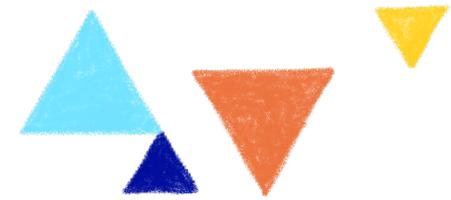


# MULTUS

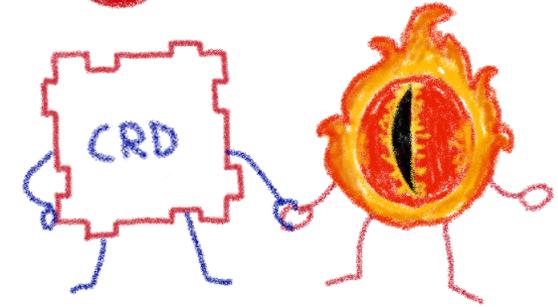
Attaching multiple network interfaces to pods:  
Bridge + Host-local



# Adding network interfaces on the fly



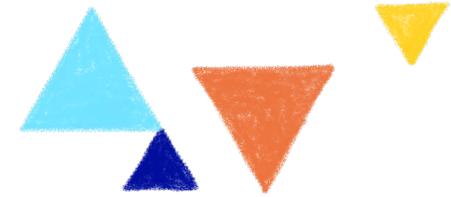
```
Annotations: k8s.v1.cni.cncf.io/networks: 2d9df3f4-9ea4-4494-b16e-eb35ed360d83, 8bee303f-f38f-4a91-b133-1da73fe5bf9c
k8s.v1.cni.cncf.io/networks-status:
  [{"name": "default",
    "interface": "eth0",
    "ips": [
      "10.100.1.133"
    ],
    "mac": "ee:2c:f7:66:c0:4d",
    "dns": {},
    "default-route": [
      "10.100.1.1"
    ]
  }, {
    "name": "2d9df3f4-9ea4-4494-b16e-eb35ed360d83",
    "interface": "net1",
    "ips": [
      "51.89.216.16"
    ],
    "mac": "fa:16:3e:05:87:b6",
    "dns": {}
  }, {
    "name": "8bee303f-f38f-4a91-b133-1da73fe5bf9c",
    "interface": "net2",
    "ips": [
      "51.89.227.253"
    ],
    "mac": "fa:16:3e:fe:f4:12",
    "dns": {}
  }
]
```



Using annotations to add interfaces to pod



# Config management

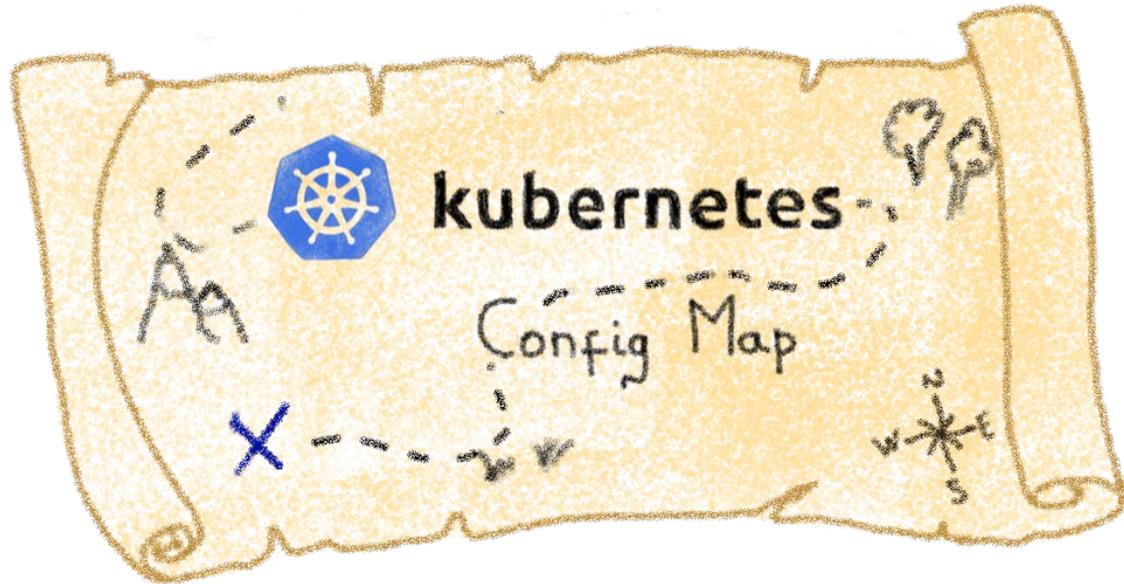


## Using Config Map

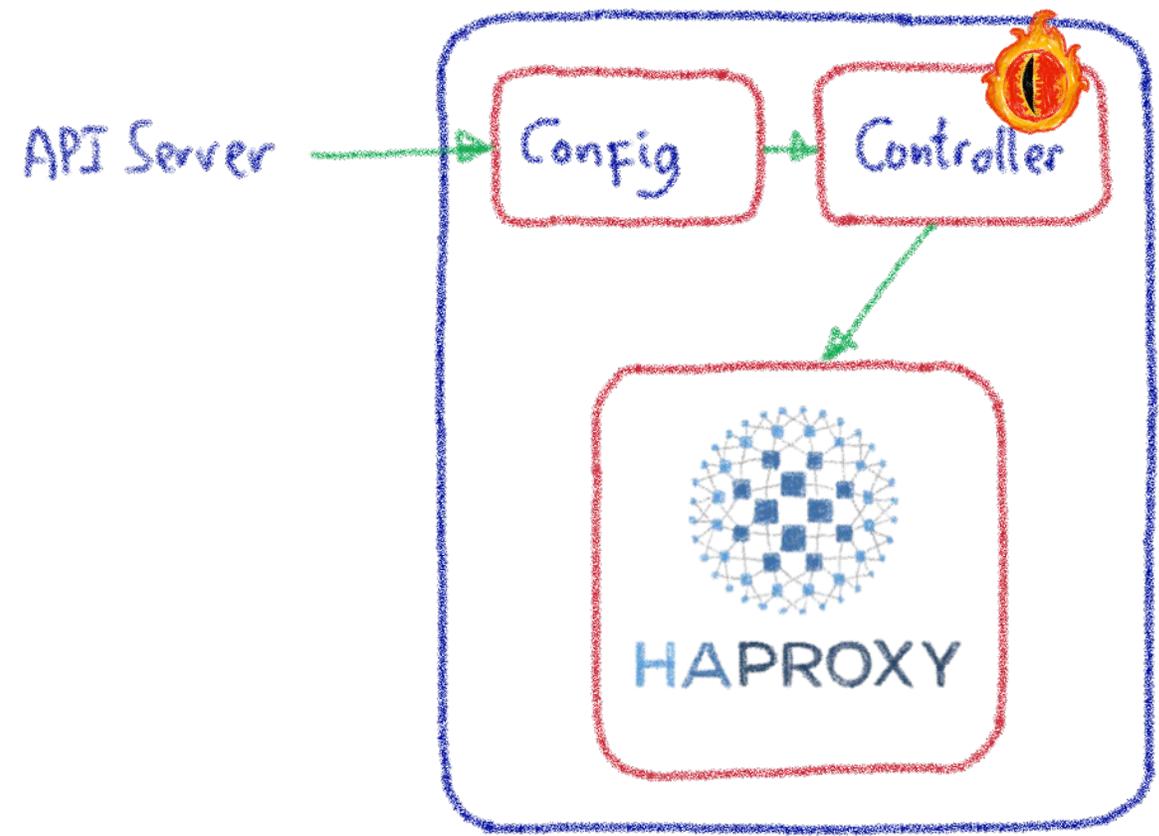
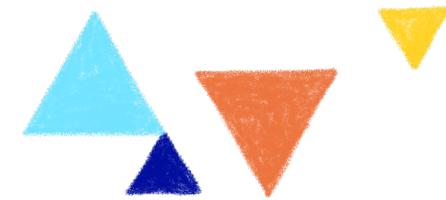
How to detect a change on Config Map files?  
Watch + Trigger?

More information on Config Map working

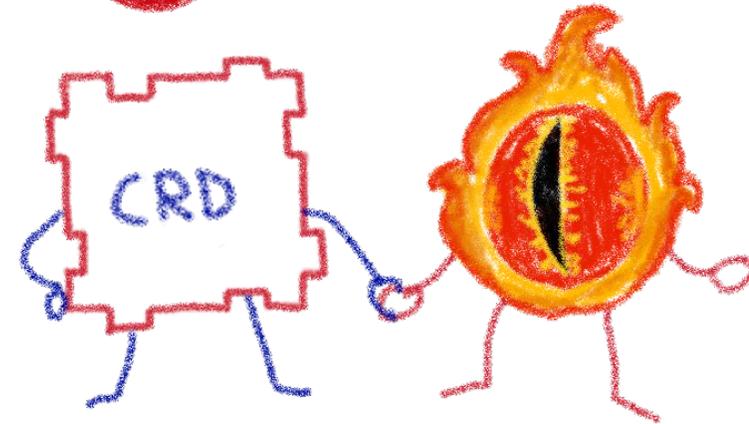
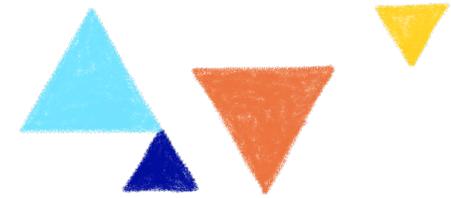
[martensson.io/go-fsnotify-and-kubernetes-configmaps](https://martensson.io/go-fsnotify-and-kubernetes-configmaps)



# A Controller to watch and trigger



# Observability



Tried Prometheus Operator, limited to one container per pod  
Switched to Warp 10 with Beamium Operator



